UNIVERSIDADE D
COIMBRA

Rahul Sharma

Concept Creation with Regulated Activation Networks

Doctoral thesis submitted to the Doctoral Program in Information Science and Technology,
supervised by Professor Bernardete Ribeiro and F. Amílcar Cardoso,
and presented to the Department of Informatics Engineeringof Faculty of Sciences and Technology
of the University of Coimbra.

December 2019

UNIVERSIDADE D
COIMBRA

Concept Creation with Regulated Activation Networks

Author:
Rahul Sharma

Advisors:
Dr. Bernardete Ribeiro
Dr. F. Amílcar Cardoso

Doctor of Philosophy Thesis

in the

Cognitive and Media Systems Group
Department of Informatics Engineering

June 2020

This research work was partially funded by the following two projects:

The Social-Oriented Internet of Things Architecture, Solutions and Environment (SOCIALITE) Project (PTDC/EEI-SCR/2072/2014), co-financed by COMPETE 2020, Portugal 2020 - Operational Program for Competitiveness and Internationalization (POCI), European Union's ERDF (European Regional Development Fund), and the Portuguese Foundation for Science and Technology (FCT).

The Concept Creation Technology (ConCreTe) Project with the financial support of the Future and Emerging Technologies (FET) program within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

"Take up one idea, make that one idea your life, think of it, dream of it, let the brain, muscles, nerves, every part of your body be full of that idea, and just leave every other idea alone. This is the way to success."

Swami Vivekananda

UNIVERSITY OF COIMBRA

Abstract

Faculty of Sciences and Technology
Department of Informatics Engineering

Concept Creation with Regulated Activation Networks

by Rahul Sharma

Concepts are of great value to humans because they are one of the building blocks of our cognitive processes. They are involved in cognitive functions that are fundamental in decision making such as classification and also capacitate us for contextual comprehension. By definition, a concept refers to an idea or a combination of several ideas. In a computational context, a concept can be a feature or a set of features. An individual concept is referred to as a concrete concept, whereas a generalized form of a set of concepts can be perceived as an abstract concept. Computational concepts can be characterized in three broad categories; i.e. symbolic (e.g. Adaptive Control of Thought based approach), distributed (e.g. Neural Networks) and spatial (e.g. Conceptual Space) representations. CLARION, a cognitive architecture, is an example of a hybrid computational framework that combines symbolic and distributed representations. Moreover, the symbolic, distributed, spatial and hybrid representations are mostly used on representing concrete concepts, whereas the notion of an abstract concept is rarely explored.

In this thesis, we propose a computational cognitive model, named Regulated Activation Network (RAN), capable of dynamically forming the abstract representations of concepts and to unify the qualities of spatial, symbolic and distributed computational approaches. Our model aims to simulate the cognitive processes of concept learning, creation and recall. In particular, the RAN's modeling has three learning mechanisms where two perform inter-layer learning that helps in propagating activations from an input-to-output layer and vice versa. The third provides an intra-layer learning that is used to emulate regulation mechanism, which is inspired by biological Axoaxonic synapse where one node in a layer induces excitatory, neutral or inhibitory activation to other nodes in the layer. In this research, two different types of abstract concepts are modeled: first, the convex abstract concepts where the geometrical convexity among the concrete concepts was exploited to create the abstract concept; second, the non-convex abstract concepts where the similarity relationships among the convex abstract concepts were used to capture non-convexity and model it. The RAN uniquely unifies the qualities of symbolic, distributed and spatial conceptual representation, where the model has a dynamic topology, simulates cognitive process like learning and concept creation and performs machine learning operations.

Experiments with 11 benchmarks demonstrated the classification capability of RAN's modeling and provided a proof-of-concept of convex and non-convex abstract concept modeling. In these experiments, the study has shown that RAN performed satisfactorily when compared with five different classifiers. One of the datasets was used to model the active and inactive states of three students. Further, the results of this model of students were analyzed statistically to infer students' psychological and physiological conditions. The recall experiments with RAN demonstrated the cued recall blend retrieval of abstract concepts. Besides cognitive function simulation and machine learning, the RAN's model was also useful in the data analysis task. In one of the experiments, a RAN's model was developed to have 7 layers showing dimension reduction and expansion operations. Additionally, the data visualization of the 1st, 3rd, and 5th layers displayed how deep data analysis with the RAN model unearth the complexities in the data.

The research work involved the study of topics from the fields of Mathematics, Computational Modeling, Psychology, Cognition, and Neurology. Based upon the results of all the experiments and analogical reasoning of RAN's modeling processes, the hypotheses of the research work were demonstrated. The abstract concept modeling was substantiated through classification experiments, whereas the simulations of concept creation, learning, activation propagation, and recall were justified through analogy and empirical outcomes. The research work also helped in discovering new challenges, such as temporal learning and simulation of the cognitive process of forgetting, which will be taken as research projects in the future.

UNIVERSIDADE DE COIMBRA

Resumo

Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

Criação de Conceitos com Redes de Activação Regulada

by Rahul Sharma

Os conceitos são de grande valor para os seres humanos, já que são blocos constituintes dos processos cognitivos. Eles estão envolvidos em funções cognitivas que são fundamentais na tomada de decisões, como a classificação, e também nos capacitam para a compreensão contextual. Por definição, um conceito refere-se a uma ideia ou a uma combinação de várias ideias. Num contexto computacional, um conceito pode ser um recurso ou um conjunto de recursos. Um conceito individual é referido como um conceito concreto, enquanto uma forma generalizada de um conjunto de conceitos pode ser percebida como um conceito abstrato. Os conceitos computacionais podem ser caracterizados em três grandes categorias segundo a forma como se representam; representações simbólicas (por exemplo, abordagem baseada no Controle Adaptativo do Pensamento), distribuídas (por exemplo, Redes Neuronais) e espaciais (por exemplo, Espaço Conceitual). A arquitetura cognitiva CLARION é um exemplo de uma estrutura computacional híbrida que combina representações simbólicas e distribuídas. Além disso, as representações simbólica, distribuída, espacial e híbrida são usadas principalmente para representar conceitos concretos, enquanto a noção de um conceito abstrato raramente é explorada.

Nesta tese, propomos um modelo cognitivo computacional, denominado Regulated Activation Network (RAN), capaz de formar dinamicamente as representações de conceitos abstratos e de unificar as qualidades de abordagens computacionais espacial, simbólica e distribuída. O nosso modelo visa simular os processos cognitivos de aprendizagem, criação e recall de conceitos. Em particular, a modelação da RAN possui três mecanismos de aprendizagem, nos quais dois realizam aprendizagem entre camadas, que ajuda a propagar ativações de uma camada de entrada para saída e vice-versa. O terceiro fornece uma aprendizagem intra-camada que é usada para emular o mecanismo de regulação, que é inspirado pela sinapse axoaxónica biológica, em que um nó de uma camada induz ativação excitatória, neutra ou inibitória sobre outros nós da camada. Neste trabalho, são modelados dois tipos diferentes de conceitos abstratos: primeiro, os conceitos abstratos convexos, em que a convexidade geométrica entre os conceitos concretos é explorada para criar o conceito abstrato; segundo, os conceitos abstratos não convexos, onde as relações de similaridade entre os conceitos abstratos convexos foram usadas para capturar e modelar a não-convexidade. A RAN unifica exclusivamente as qualidades de representação conceptual

simbólica, distribuída e espacial, onde o modelo tem uma topologia dinâmica, simula processos cognitivos como aprendizagem e criação de conceitos, e executa operações de aprendizagem pela máquina.

Experiências com 11 benchmarks demonstraram a capacidade de classificação do modelo da RAN e a constituem prova de conceito da modelação de conceito abstrato convexo e não convexo. Nestas experiências, a RAN demonstrou um desempenho satisfatório quando comparado com cinco classificadores diferentes. Um dos conjuntos de dados foi usado para modelar os estados ativo e inativo de três estudantes. Além disso, os resultados desse modelo de estudantes foram analisados estatisticamente para inferir as condições psicológicas e fisiológicas dos alunos. As experiências de recall com RAN demonstraram a recuperação combinada de recall de conceitos abstratos. Além da simulação da função cognitiva e de aprendizagem automática, o modelo da RAN também foi útil na tarefa de análise de dados. Numa das experiências, o modelo de uma RAN desenvolvido tem 7 camadas, que comportam operações de redução e expansão de dimensão. Além disso, a visualização de dados da 1ª, 3ª e 5ª camadas mostra como uma análise profunda dos dados com o modelo RAN pode revelar as complexidades destes.

Este trabalho de investigação envolveu o estudo de tópicos das áreas de Matemática, Modelação Computacional, Psicologia, Cognição e Neurologia. Com base nos resultados de toda a experiência e raciocínio analógico dos processos de modelação da RAN, foi possível demonstrar a hipótese do trabalho de investigação. A modelação de conceito abstrato foi substanciada através de experiências de classificação, enquanto as simulações de criação de conceito, aprendizagem, propagação de ativação e recall foram justificadas por analogia e resultados empíricos. Este trabalho ajudou também a descobrir novos desafios, como a aprendizagem temporal e a simulação do processo cognitivo do esquecimento, que serão objeto de investigação futura.

Acknowledgments

गुरू गोविन्द दोऊ खड़े, काके लागूं पांय।
बलिहारी गुरू अपने गोविन्द दियो बताय।।

Meaning: *"When a teacher and god are standing together who should be regarded first? It's because of the teacher we have awareness of god, therefore, in such a situation a teacher should be regarded first."*

I would like to express my special gratitude towards Professor Bernardete Ribeiro not only for accepting me as her student but also guiding me from a challenging phase towards the completion of my thesis. Under her supervision, I learned a lot about the international standards needed for machine learning research. Her invaluable teachings helped me to improve my research related skills such as scientific writing. Besides research, her personality taught me the real meaning of the phrase "when the going gets tough, the tough get going." I am grateful to my co-advisor Professor Amílcar Cardoso for encouraging me working on this dissertation while working on research projects at CISUC. I thank him for all the inputs that he gave for my projects and research work especially for financially supporting a part of my graduate research.

I would like to mention about my former advisor Dr. Alexandre Miguel Pinto, I am thankful that he mentored me ever since I entered the field of Machine Learning. This research problem was conceived and a large part of the work was carried out under his advisory. Under his supervision period, during the brainstorming sessions, he passed me his knowledge of computational modeling and helped me to develop an understanding of mathematics in Machine Learning. His conduct and attitude towards solving challenging problems always inspired me and helped me to evolve in my career.

I am very lucky to have Mr. Jorge Ávila as our administrative professional, it's because of him the bureaucracy is far from me. No matter how much he is loaded with work but he always had time to listen to me and solve my problems. I thank my colleagues Ngombo, Mariam, Ali for their invaluable inputs to refine my thesis writing.

Thanks to my friends in Coimbra for their companionship. Especially to Dr. Vivek for discussion on science, technology and politics, these sessions helped in refreshing my mind. I am also thankful to my colleagues Adriano and Amanda for their family-like supports and love. I thank my friends in India Abbas, Abhay, Sudhir, Pavan and Tushar for being there in good and bad times. I thank my sisters and their husbands for their encouragement and support, with special mention of my sister Mukti who not only supported me but also inspired me to be someone. A special mention about my father, mother and fiancée because this Ph.D. is an outcome of their support, love, affection and prayers. Last but not least I thank God for this beautiful life.

Contents

List of Figures

List of Tables

Abbreviations

AA	Actual Activation
AC	Abstract Concept
ACL	Abstract Concept Labeling
AF-H	Afternoon Hours
AI	Artificial Intelligence
ANNs	Artificial Neural Networks
A-S	Active-Subject
AUC	Area Under Curve
BC1	Breast Cancer 669 data
BC2	Breast Cancer 569 data
CA	Credit Approval data
CAC	Convex Abstract Concept
CACC	Convex Abstract Concept Creation
CACUAP	Convex Abstract Concept Upward Activation Propagation
CCI	Convex Concept Identification
CCILL	Convex Concept Inter-layer Learning
CCILW	Convex Concept Inter-Layer Weight
CHC	Concept Hierarchy Creation
CI	Concept Identifier
CRDP	Cluster Representative Data Points
CSRL	Concept Similarity Relation Learning
CSRW	Concept Similarity Relation Weights
DoC	Degree of Confidence
E-A	Expected Activation
E-H	Evening Hours
E-MH	Early Morning Hours

E-R	Expected Recall
GBP	Geometric Back-propagation
GDF	Geometric Distance Function
GI	Glass Identification data
HAR	Human Activity Recognition
ID	IRIS data
IL	Intra-Layer
I-S	Inactive-Subject
MBVCR	Multiple Binary Valued Cue Recall
MCR	Multiple Cue Recall
M-H	Morning Hours
MM	Mammographic Mass data
MP	Mice Protein data
NAC	Non-convex Abstract Concept
NACC	Non-convex Abstract Concept Creation
NACI	Non-convex Abstract Concept Identification
NACIL	Non-convex Abstract Concept Inter-layer Learning
NACL	Non-convex Abstract Concept Labeling
NACUAP	Non-convex Abstract Concept Upward Activation Propagation
NCILW	Non-convex Inter-Layer Weight
O-A	Observed Activation
PoH	Period of Hours
RANs	Regulated Activation Networks
ROC	Receiver Operating Characteristics
RQ	Research Question
SCR	Single Cue Recall
ST	Similarity-Threshold
STF	Similarity Translation Function
TD	Toy-data
WR	Wine Recognition data
WWW	World Wide Web

Dedicated to my Mother & Father

Chapter 1

Introduction

The area of this research work falls under the scope of Mathematics, Computational modeling, Psychology, Cognition and Neurology domains. The conceived research problem of this thesis has its motivations from the field of Computational Psychology and Neurobiology. However, the problem was addressed through a Computational and Mathematical approach. The following sections of this chapter detail the inspirations of this work, the formulated problem and hypothesis for the solution. This chapter also lists the contributions of this research and description of the Research Methods adopted to carry out the research and development activities.

1.1 Motivation

Learning is an ongoing process among all living beings, we perceive our surroundings, observe all the events happening around and build an understanding of all the things as *concepts*. We also develop internal representations (Goldstone et al., 2013) of these concepts which eventually becomes an important part of our recognition process. By dictionary, the term concept refers to an "idea" (Dictionary.com, 2015b). However, concepts are also described as mental categories (Goldstone et al., 2013) that equip us with the abilities such as relative recognition of objects. Concepts are an important part of investigations in cognitive and psychological research. Usually, process-oriented, symbolic or distributed, and knowledge-based conceptual representations describe concepts (Kiefer and Pulvermüller, 2012), where researchers use formal, theoretical, or computational approaches to realize the conceptual representations (Bechtel et al., 1998).

There are several computational modeling techniques (or tools) that simulate cognitive states and represent concepts at symbolic and connectionist levels. At the symbolic level, information is represented by symbols. Rules are defined to manipulate symbols. Within a symbolic representation, the meaning is internal to the representation itself; symbols have meaning only

in terms of other symbols, and not in terms of any real-world objects or phenomena they may represent. Symbolic representations are often associated with Classical AI (Russell and Norvig, 2016), yet symbolic representation in itself does not entail classic AI methodology. An underlying assumption of AI research is that human thinking can be understood in terms of symbolic computation, in particular, computation based on formal principles of logic. However, symbolic systems have proved less successful in modeling aspects of human cognition beyond those closely related to logical thinking, such as perception. At the connectionist level, information is represented by the dynamics over densely connected networks of primitive units. A particular strength of connectionist networks is their ability to adapt their behavior according to observed data. Nevertheless, since the learned behavior is represented as weights between units in the network, they offer limited explanatory insights into the process being modeled. Peter Gärdenfors (Gärdenfors, 2004) proposed a third inter-mediate approach – the Conceptual Spaces theory – whereby concepts are represented as regions within a multi-dimensional space, with the data features representing the dimensions of the space. Under this setting, one particular data instance would correspond to a single point in the space. This approach allows geometric operations over data points and across sub-regions – e.g., to measure the similarity (typically via distance-based measures) between data instances and concepts.

The term concept automatically coins the need to understand the representations of concepts. There are several conceptual representation theories (Kiefer and Pulvermüller, 2012) that not only enable us to understand the various cognitive processes in humans but also psychological ones, like creativity (Kyaga et al., 2013). Each theory has a way to represent concrete concepts through perception, action, emotion, and introspection, but the notion of abstract concepts is debatable (Kiefer and Pulvermüller, 2012). In general, a hierarchical structure defines an organization of concepts where the concrete concepts are placed in the lower level, and the Abstract Concepts occupy the relatively higher levels. Therefore, abstract concepts are seen as the generalization of concrete concepts (Rosch et al., 1976, Tversky and Hemenway, 1984). Abstract concepts are studied mathematically (Saitta and Zucker, 1998), and theoretically (Borghi et al., 2018, 2017). However, they are less investigated computationally (Kiefer and Pulvermüller, 2012). The first aspect of this research project was to investigate concepts, understand how they are perceived and study how they can be computationally represented.

Computational models are useful in understanding the psychological and cognitive phenomena, validate the existing cognitive theories, and helps to formulate fresh ideas related to cognition (Rolls et al., 2008, Kyaga et al., 2013, Braver et al., 1999, O'Reilly, 2006). The representations produced by computational approaches are amodal (symbolic), multimodal (distributed), or hybrid (Hayes and Kraemer, 2017). Adaptive Control of Thought-Rational (ACT-R) (Anderson et al., 1997) is a symbolic architecture where the researcher have attempted to model memory (Lovett et al., 2000), simulate attention (Anderson et al., 2004, 1997, 2004), decision making (Marewski and Mehlhorn, 2011), recognition (Schooler and Hertwig, 2005), and

forgetting (Schooler and Hertwig, 2005). Multimodal approaches such as artificial neural networks (ANN) including Restricted-Boltzmann Machine (RBM) (Hinton, 2012), Deep Neural Networks (Collobert and Weston, 2008) stacked auto-encoders (Vincent et al., 2010) and Convolution Neural Networks (CNN) (Krizhevsky et al., 2017) have significant contribution in feature extraction and distributed memory representation. Besides, hybrid cognitive architectures like CLARION (Sun and Peterson, 1996) simulated scenarios related to cognitive and social psychology. Another motivation of this research was to computationally investigate the dynamic cognitive states in humans and understand concept creation processes. Additionally, studying the association learning among the concepts, how recall of concepts occur and how the concept(s) induce(s) regulatory effect during the retrieval operation were also object of interest.

1.2 Research Goals

The aim of this research was to develop a computational cognitive model capable of dynamically forming the abstract representations of concepts and to unify the qualities of symbolic, distributed and spatial computational approaches. It is also assumed that the model simulates the cognitive processes such as learning, concept creation and recall operations. Besides these qualities, the model should also perform machine learning tasks like classification. This part of the thesis puts forward the problem conceived from the aforementioned motivations. Additionally, the research hypotheses are described along with its research questions.

1.2.1 Problem Statement

There are numerous ways to represent concepts computationally, which are broadly categorized into three classes spatial, symbolic and distributed. The spatial representations of the theory of conceptual space (Gärdenfors, 2004) enable us to view concepts in multi-dimensions, therefore serves as a natural way of contextual learning. Ontologies are a good example of symbolic representations and ANNs belong to a family of distributed representation. Individually, both symbolic and distributed representation have been very beneficial in understanding cognitive functions (such as recognition) and performing machine learning operations (like classification). However, the brain requires both symbolic and distributed representations (Roy, 2011) for contextual reasoning because similarity among concepts plays an important role in deducing context. Separately, all the three approaches are very efficient, although it's not clear how to harmonize the virtues of these distinct approaches for concept representation.

Besides the gap of a unified view of concept representation, the distributed representation of ANNs simulates the Axodendritic synapses behavior of learning and activation propagation. The neuronal behavior responsible for regulating the activations of neurons through Axoaxonic

synapses (Garrett, 2014) is not addressed by ANNs, and leads to an opportunity to emulate these synapses.

1.2.2 Research Hypothesis

Based upon the two aforestated problems and motivation of this work, the objective of this research is to develop a computational approach. To fulfill the objective a hypothesis is developed consisting the following two postulates:

- A computational network where every node symbolically represents a concept (abstract of concrete) and geometry among the concepts is considered in modeling, such a network can be considered a hybrid of symbolic, distributed and spatial representation. A hierarchical network of concepts (nodes) can address the representation of abstract and concrete concepts computationally. If the hierarchy generation is dynamic, the cognitive processes such as concept creation, learning and activation propagation can be simulated.
- The regulation behavior of Axoaxonic synapses can be simulated if the concepts at the same level are able to impact each other's activation. In the process of regulating activation of other concepts, a dynamic state of concepts can be captured similar to dynamic state when recalling a concept.

The research hypothesis has the potential to solve the two research problems of this thesis. To realize the two aspects of the hypothesis following research questions (RQ) are formulated:

- **RQ-1: Model Architecture and Learning**
  Which technique can be useful in viewing concepts as data points in n-dimensional feature hyperspace? How to build a hierarchy of concepts where higher level concepts abstractly represent concepts at lower levels? How to learn the relation between concepts at the same level, and different levels in the hierarchy?
- **RQ-2: Activation Propagation**
  How to propagate activation (signals) from the input-to-abstract level using both types of learning? How to strew activation from abstract-to-input level using the learning? How use learning to have a regulatory effect of activation of one concept over other?
- **RQ-3: Model Evaluation**
  What use cases are suitable for RANs modeling? Which methods are suitable to validate the machine learning capability of the methodology? How to simulate and verify the cognitive behaviors exhibited by the approach and substantiate the processes like recall and blend retrieval of concepts?

1.3 Research Outcome

The research work proposes a novel computational modeling technique, Regulated Activation Networks. In RANs modeling, every data instance is considered as a *spatial* point in an *n*-dimension feature hyperspace. Topologically, RANs modeling is *connectionist* and *symbolically* represents every node in the network as a *concept*. Therefore, RANs modeling unifies the virtues of all the representations of computational modeling, i.e. symbolic, distributed and spatial. The RANs modeling is also able to simulate the cognitive functions such as learning, concept creation and recall operations. In this section, an overview of contributions of RANs modeling is given together with its restrictions. Later, research methods are described that were adopted to achieve the research goals of this work.

1.3.1 Contributions

There are three significant contributions proposed by this research work. First, the RANs modeling technique, which has an evolving topology, learns representations of abstract concepts. The model symbolically identifies every node in the network as a concept and its activation (signal value) as its Degree of Confidence (DoC) of representation. The RANs modeling can build both convex and non-convex representations of abstract concepts. Second, RANs approach is suitable to perform classification operations without supervision. Besides, the data analytics can be performed with RANs modeling by transforming the dimension of data, i.e., dimension reduction and expansion and further visualizing this data for analysis. Third, the emulation to study cognitive and neurobiology process such as dynamic creation of a representation of concepts, learning. RANs modeling can also simulate the inhibitory or excitatory and regulatory neurological impact of concepts during cued recall operations.

Besides these core contributions, the RANs approach has been used to model the Active and Inactive abstract behavior of a group of students. Further, the generated model of students was used to infer the physiological and psychological conditions of a group of students. With RANs modeling there is a liberty of choice in selecting a concept identifier (i.e., clustering algorithm), therefore the basic architecture also becomes flexible. The RANs' convex abstract concept modeling has two types of activation propagation operation: first, upward propagation, from input-to-abstract level, and second, downward propagation from abstract-to-input level. The significance of downward propagation operation is not only simulating recall operation but also retrieving blends of abstract concepts at the input level. This blending operation can be used to retrieve intuitive or non-intuitive combinations of abstract concepts.

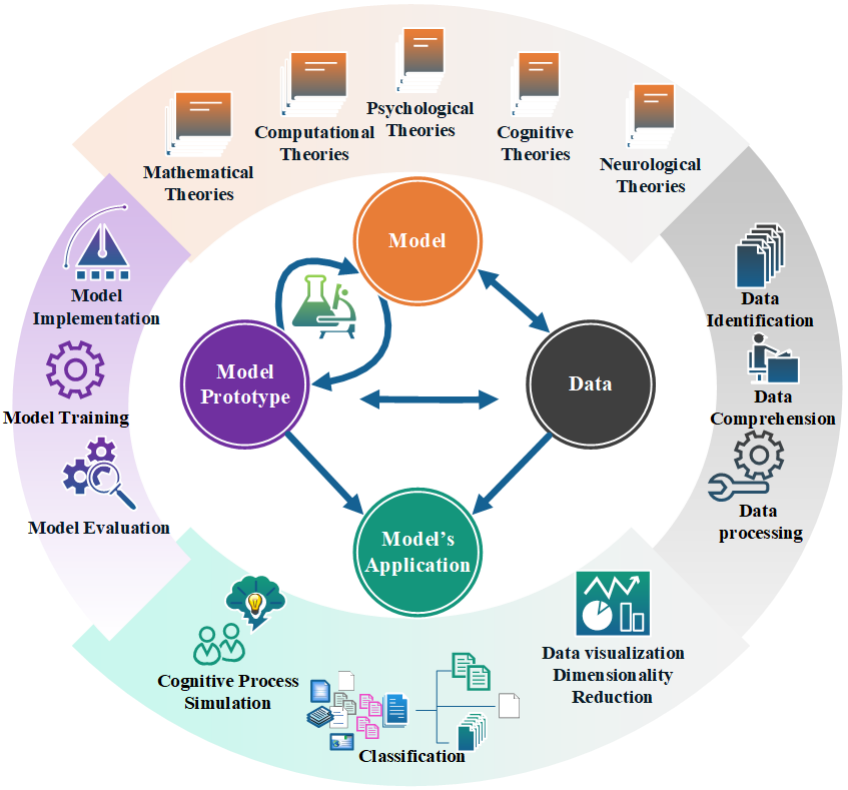


FIGURE 1.1: Research Methodology

1.3.2 Limitations

There are a few restrictions with RANs modeling. First, the training efficiency of the RANs model is linked to the complexity of the concept identifier. For instance, if the Affinity Propagation algorithm is used instead of K-mean for clustering, the training time increases with the size and dimension of the data. Second, RANs modeling accepts only numbers in the range [0, 1] as input. Third, the current versions of RAN is not suitable for image classification tasks. Forth, the present version of RAN works with spatial data only and does not capture temporal characteristics while learning. At last, the Similarity Threshold constants for RANs non-convex Abstract Concept modeling is to be determined empirically.

1.3.3 Research Methodology

The work performed in this research consisted of four major components, *Model*, *Model Prototype*, *Data* and *Model's Application*, as shown in Figure 1.1. The *Model* is the mathematical formulation and an iterative process between Mathematical, Computational, Psychological, Cognitive and Neurological theories, in order to conceive and develop RANs modeling technique. The *data* format acceptable to RANs model was also decided while formulating the RANs model. The *Model* component was essential to seek answers to the research question RQ-1 and RQ-2 (see Section 1.2.2) and to develop theoretical grounds for RANs modeling. The

Model Prototype is another important component where the theoretical aspects of the research questions RQ-1 and RQ-2 were realized computationally. The research question RQ-3 was collectively addressed by the three components *Model*, *Model Prototype*, and *Data*, and test the computational capability of RANs modeling with different use cases.

Experimental Setup

The *Model Prototype* was developed using the Python scripting language. To train the RAN's model, fourteen[1] datasets were identified, understood and pre-processed, and further supplied to the training process of the RANs modeling. Five evaluation metrics (Precision, Recall, F1-Score, Accuracy and ROC-AUC curve analysis) were considered to validate the classifications capability of RANs modeling. The classification operation was also suitable to provide the proof-of-concept of abstract concept modeling. The analogical reasoning made through the RAN's simulations were used to establish the cognitive behavior exhibited by the RAN's Methodology. Further, the *Data* and *Model Prototype* was conjointly used to demonstrate the applications of RANs modeling such as recall process simulation, classification and dimensionality reduction along with data visualization.

1.4 Outline of the Thesis

This research dissertation is organized into six chapters and five supporting appendices. Chapter 1 provides the motivation of this research, proposes a set of research questions and presents the research methodology adopted to solve the problem. Chapter 2 puts forward the state of the art related to this research. Chapter 3 provides the RAN's convex abstract concept modeling, along with the limitations of RANs modeling and its biological justification, it also details the data pre-processing method needed for the RANs modeling. Section 3.3 details the RAN's methodology using a Toy-data problem (Toy-data1). Section 3.4 shows the experiments with two datasets acquired from UCI machine learning repository to exhibit *(1)* flexibility in choosing a suitable concept identifier, *(2)* building a deep hierarchy of abstract concepts, *(3)* automatic association of input-labels to their respective abstract concept nodes. Section 3.5 provides RAN's comparisons with five classifiers and a proof of concept with eight benchmark datasets. At last, Section 3.6 summarizes and concludes the chapter.

Chapter 4 extends Chapter 3 with RAN's non-convex abstract concept modeling. In Section 4.2, the entire RAN's methodology is explained with a Toy-data problem (Toy-data2). Experiments and outcomes with Toy-data, Human Activity Recognition data, Sleep Detection data and four datasets UCI Machine Learning Repository are reported in Section 4.3. Section 4.4 puts forward the conclusions. Chapter 5 also extends Chapter 3 with the induction of back-propagation mechanism in order to exhibit cognitive behavior modeling with RAN. Section 5.2 puts forward

[1]Nine datasets from UCI Machine Learning Repository, one dataset from ISABELLA App of project SOCIALITE, and three artificially generated datasets.

the state of the art related to recall operations. The RANs modeling, the Intra-Layer Regulation algorithm, and Geometric Back-propagation algorithm are detailed in Section 5.3 using a Toy-data problem (Toy-data3). The cued recall demonstration with MNIST dataset is reported in Section 5.8. Section 5.9 concludes the Chapter. Chapter 6 concludes this thesis summarizing the outcome of the research work along with a reflection on future work.

This thesis is complemented with a set of Appendices to help readers understand the founding concepts, to support results for the experiments, and describe Machine Learning application of the work. Appendix A briefly describes the theories that were the building blocks of RANs modeling. Appendix B details the utilities that are used in evaluation process, this appendix also provides the configurations of models used in different experiments in this research. Appendices C and D detail the observations of classification performance for experiments carried out in Chapter 3 and Chapter 4, respectively. Appendix E describes the application of RANs modeling in data manipulation, dimension transformation and data visualization.

Chapter 2

Related Work

This chapter puts forward the State of The Art related to the RANs modeling technique, and is based upon the following International Conferences and part of under review Journal article. Following is the list of articles:

- Ping Xiao, Stuart Battersby, Marko Bohanec, Amilcar Cardoso, Joao Correia, Alberto Dıaz, Jamie Forth, Virginia Francisco, Pablo Gervas, Oskar Gross, Raquel Hervas, Nada Lavrac, Carlos Leon, Penousal Machado, Dragana Miljkovic, Hugo Goncalo Oliveira, Alexandre Pinto, Vid Podpecan, Senja Pollak, Matthew Purver, Rahul Sharma, Tanja Urbancic, Frank van der Velde, Geraint A. Wiggins, Martin Znidarsic, Hannu Toivonen. "Conceptual Representations for Computational Concept Creation" ACM Computing Survey, 2019. (Xiao et al., 2019)

- Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto and Amílcar F. Cardoso, *Exploring Geometric Feature Hyper-Space in Data to Learn Representations of Abstract Concepts.* Applied Sciences, 2020. (Sharma et al., 2020a)

- Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto and Amílcar F. Cardoso, *Learning Non-Convex Abstract Concepts with Regulated Activation Networks*, Journal of Annals of Mathematics and Artificial Intelligence, Springer, 2020. (Sharma et al., 2020b)

- Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto and Amílcar F. Cardoso, *Modeling Abstract Concepts For Internet of Everything: A Cognitive Artificial System.* 13th APCA International Conference on Control and Soft Computing (CONTROLO), Azores, Portugal, 2018. (Sharma et al., 2018b)

- Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto and Amílcar F. Cardoso, *Perceiving Abstract Concepts Via Evolving Computational Cognitive Modeling*, International Joint Conference on Neural Networks (IJCNN), IEEE World Congress on Computational Intelligence (IEEE WCCI), Rio de Janeiro, Brazil, 2018. (Sharma et al., 2018a)

A plethora of computational approaches exist in the area of Artificial Intelligence (AI). These models address a large section of problems of Machine Learning such as clustering, classification, prediction, pattern recognition, deep learning and computational creativity. This Ph.D. dissertation also contributes a computational cognitive model to the AI and Machine Learning domain. In this chapter, the symbolic, distributed, and spatial representations are surveyed to understand how they represent concepts and how their virtues can be unified. This work proposes a modeling technique to represent *abstract concepts*, therefore the computational and conceptual research related abstract concept modeling is also reviewed. Since the model proposed by this Ph.D. work is hybrid and dynamic, a literature survey over recent advancements is also conducted for augmenting and hybrid models.

2.1 Representation of Concepts

Concepts are very important to lives because they are linked to the cognitive functions of relative thinking, reasoning, recognition and comprehension. In psychology, the concepts are referred to "an abstract idea" that is useful to develop understanding about things, objects and processes. Besides perceiving the ideas, knowledge about concepts is also essential for the generation of new or creative ideas (Boden, 1990). On the other hand, in cognitive science concepts are referred to by their *representations*. Therefore, the principal difference of these two senses of concepts is that, in psychology concept is an "idea" whereas in cognitive science "representation of an idea" is a concept.

In *symbolic* approaches, the symbols are not only used to represent concepts but also depicts their association. In *distributed* techniques, a concept is represented by the densely connected network of primitive units. Both techniques have their merits and demerits, and are applied based upon the requirement of the problem being solved. The *spatial* methods represent concepts at a *conceptual level* which lies between *symbolic level* and *distributed levels*. At the spatial level, the concepts are seen as points in geometric space and their associations are established based upon the distance among the concepts. The purpose of this review is to determine the advances in all three type of representations and unify the virtues of these methods in RANs modeling.

2.1.1 Symbolic Representations

Symbol plays a significant role in recognizing concepts and establishing relationships among them. Symbols are used to associate meaning to things, objects or processes such as words makes a language meaningful. This section reviews the computational methods that enable us to represent concepts symbolically.

Associations

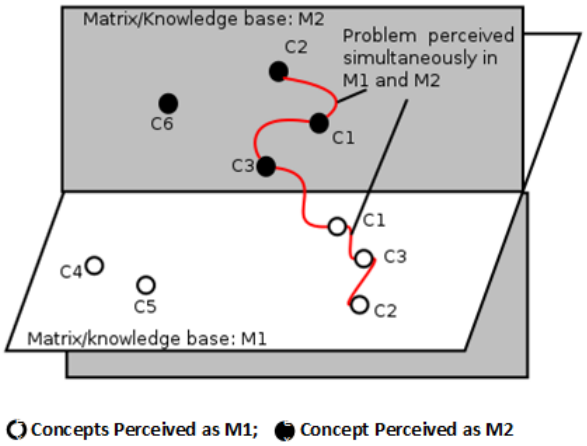


FIGURE 2.1: Koestler's cross-domain association (Koestler, 1964)

Association is "the connection or relation of ideas, feelings, sensations, etc.; correlation of elements of perception, reasoning, or the like" (Dictionary.com, 2015a). If two concepts (C_1 and C_2) are related to one another then, in an association, it is assumed that there exists a connection between them without imposing any condition on the concepts. Understanding of associations among concepts is important and has been used to define creative thinking as a process of combining distant associative element (Mednick, 1962).

Arthur Koestler proposed a model of creative thinking as *bisociation* (Koestler, 1964). Bisociation can be seen as an act of mixing unrelated (often conflicting) information in an innovative manner. This act can also be viewed as the ability to think simultaneously over different dimensions or domain, as shown by Figure 2.1, which depicts two matrices of thought $M1$ and $M2$, as orthogonal planes. $M1$ and $M2$ represent two matrices that belong to different domains. Any event which is perceived simultaneously (shown by the red line in the Figure 2.1) is bisociation of two context M_1 and $M2$. Figure 2.1 also shows six concepts $C1$, ..., $C6$. The concepts $C1$, $C2$ *and* $C3$ are bisociative as they are able to view the problem in two domains. Bisociations generally identify a set of concepts that belong to closely related domains (Berthold, 2012).

Graphs are used to represent bisociations (Dubitzky et al., 2012). The bisociations not only can *bridge* between concepts but also links graphs and structures (Juršič et al., 2012). It has also been shown that bisociation discovery can be tackled using literature mining methods (Swanson, 1990) by finding hypotheses spanning over previously disjoint sets of literature. This can be viewed as determining whether phenomenon x is linked to phenomenon z in an article. However, there is no proof of this association and to associate concepts C_x and C_z an intermediate concepts C_y is identified. This concept C_y connects with C_x in some articles, and with C_x in some others. Putting these connections together and looking at their meaning may provide new insights about C_x and C_z.

Semantic Relations

Semantic relation is termed as any *relationship* established among two or more concepts based upon their meaning. Equation 2.1 shows the formal representation of a semantic relation R among two concepts C_1 and C_2. There are several types of semantic relations in the lexical domain such as *synonymy, homonymy, antonymy, association, causal relation, hyponymy-hypernymy, instance_of relation, locative relation, meronymy-holonymy and temporal relation* (Santos et al., 2016). In linguistics, the words and their meaning conjointly represent concepts and associations among these words are recognized by semantic relations.

$$C_1 \underset{R}{\rightarrow} C_2 \tag{2.1}$$

Semantic relations have been used in text corpora including the electronic dictionaries (Amsler, 1981, Calzolari et al., 1973). However, a large amount of language-related research (Chen and Sharp, 2004, Krallinger et al., 2009, Miljkovic et al., 2012, Ono et al., 2001, Rzhetsky et al., 2004) is based upon the method of discovering discriminating lexical-syntactic patterns (Hearst, 1992). This approach also aids in determining the sequences of words (concepts) from a set of relations (as starting points) which are learned from previous related arguments. There are several Machine Learning approaches that were used to extract semantic relations in text such as the use of classifiers (Girju et al., 2006), clustering (Hasegawa et al., 2004) and Natural Language Processing (NLP) (Craven and Kumlien, 1999).

Symbolically, the semantic relations among the concepts are represented using *logic*, *ontologies* and *semantic networks*. In all these techniques the symbols, graphs or tuples not only represents the concepts but also their relations. There are different types of notations like the conjunction, disjunction, negation, implication, etc., to represent logic. A concept in logic can be one symbol or a composition of several symbols. Consider the logic relation expressed by Equation 2.2, it says "All professors are people", "$\forall$" symbol has its own meaning, "$is\_prof()$" is a relation and "$\rightarrow$" is an implication symbol.

$$\forall x(is\_prof(x) \rightarrow is\_person(x)) \tag{2.2}$$

With logic symbols, we not only can represent spatial concepts but also temporal concepts (Bouzid et al., 2006). Logic-based computational techniques, such as the logic program of Prolog, have also been used to represent concepts and their procedural relations (Hou et al., 2010).

Another symbolic approach that represents concepts are ontologies, by definition, an ontology is a the study of *state of being* (Dictionary.com, 2015c), here the *state of being* can be interpreted as a concept (or category). Ontologies are not only used to studying concepts but also their relations (Poli, 1996). Ontology is also termed as knowledge base representing a hierarchical taxonomy of concepts, relations, functions, Axioms and instances (Slimani, 2015). In information science, *Ontologies* are defined as "a set of representational primitives with which to model

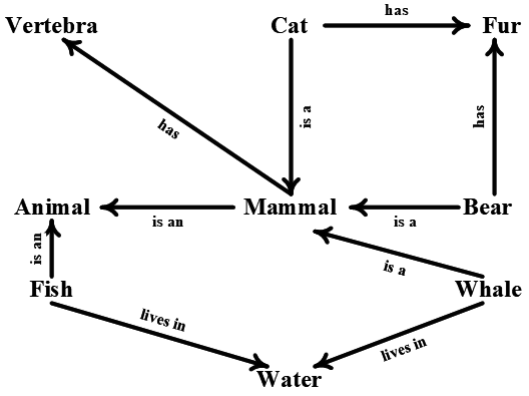


FIGURE 2.2: An example semantic network

a domain of knowledge or discourse. The representational primitives are typically *classes* (or sets), *attributes* (properties) and *relationships* (or relations among the class members)" (Gruber, 2008).

There are three types of ontologies found in information science, upper, domain and hybrid (or middle). The upper and domain ontologies can be compared based upon what they represent. For instance, the upper identifies non-lexical concepts such as objects, entities or situation, whereas, the domain ontology contains concepts that are directly expressed through text or lexicon. The SUMO (Pease and Niles, 2002), DOLCE (Gangemi et al., 2002) and Cyc (Lenat, 1995) are examples of upper-level ontologies. The domain ontologies that are developed specifically to represent a field such as *workflow of an assembly line*. Besides upper and domain ontologies, the hybrid ontologies represent generic concepts, WordNet (Miller, 1998) is an example of this type of ontology.

The representations of ontologies itself can be differentiated into three types formal, prototype-based and terminological (Biemann, 2005, Sowa, 2015). The *formal* ontologies are denoted by symbols of first-order logic, as depicted by Equation 2.2, SUMO is an example of formal ontologies. The *prototype-based* ontologies are a form of terminological ontologies whose classes are discriminated based upon the instance or prototype and are identified based upon the semantic similarity among the concepts. This type of ontology is realized using clustering mechanisms and are easy to construct, however they do not have a label associated with concepts and don't allow interference. The *terminological* ontologies are hierarchical representations of concepts where taxonomy have concept labels and associations are specified by sub-type or super-type (is_a or part_of) relations. The graph-based terminological ontologies are often referred to as semantic networks. The ontologies provide generic information about the concepts and their relations in a knowledge base, whereas the semantic networks are seen as knowledge bases.

Semantic networks (Berners-Lee et al., 2001, Hendler and van Harmelen, 2008) are also a way to symbolically represent concepts and their semantic relations. It is a graph-based approach

(directed or undirected) where the vertices (or nodes) represents concepts and the semantic relation is represented by the edges. The Figure 2.2 shows the relations *"is a", "has", "is an", "lives in"* along with their respective concepts *Animal, Mammal, Fish, Water, Whale, Bear, Fur, Cat and Vertebra*. An example of a semantic network is WordNet, a lexical database of English. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. Some of the most common semantic relations defined are meronymy (A is part of B, i.e. B has A as a part of itself), holonymy (B is part of A, i.e. A has B as a part of itself), hyponymy (or troponymy) (A is subordinate of B; A is kind of B), hypernymy (A is superordinate of B), synonymy (A denotes the same as B) and antonymy (A denotes the opposite of B). ConceptNet (Liu and Singh, 2004) is another example of a semantic network representing a knowledge base of common sense.

Semantic networks are very useful in knowledge representation of databases (Gutierrez et al., 2011), cyber-security (Iannacone et al., 2015) and object serialization in software (Hebeler et al., 2011). Besides, in the World Wide Web (WWW) a massive collection of linked web documents is created through a network of semantic information also known as the semantic web. Semantic Web incorporates ontologies as part of W3C standard stack. An ontology describes formally a domain of discourse. Typically, an ontology consists of a finite list of *terms* and the *relationships* between these terms. The terms denote important concepts (classes of objects) of the domain. For example, in a university setting, staff members, students, courses, lecture theaters, and disciplines are some important concepts. The relationships typically include hierarchies of classes. A hierarchy specifies a class C to be a subclass of another class C' if every object in C is also included in C' (for instance, all faculty is staff members). Apart from subclass relationships, ontologies may include information such as properties (X teaches Y), value restrictions (only faculty members can teach courses), disjointness statements (faculty and general staff are disjoint), the specification of logical relationships between objects (every department must include at least ten faculty members). In the context of the Web, ontologies provide a shared understanding of a domain. In a broad context, the ontologies in semantic web are used to specify standard conceptual vocabularies in which to exchange data among systems, provide services for answering queries, publish reusable knowledge bases, and offer services to facilitate interoperability across multiple, heterogeneous systems and databases.

Several ontology languages conform to semantic web such as Resource Description Framework (RDF) (Lassila and Swick, 1999), RDF Schema (RDFS) (Brickley, 2004), Web Ontology Language (OWL) (McGuinness and van Harmelen, 2004), Terse RDF Triple Language (Beckett et al., 2014) (Turtle) which is a format for expressing data in the Resource Description Framework (RDF) data model. RDF represents information "triples", each of which consists of a subject, a predicate, and an object. The tools for implementing the Semantic Web are designed for encoding data and sharing data from many different sources. The RDF is used to encode information and the RDFS language defines properties and classes and also facilitates using

data with different RDF encoding without the need to convert data to use a different schema. Furthermore, one can make ad hoc RDF statements about any resource without the need to update global schemas.

2.1.2 Distributed Representations

Human intelligence is exceptional because it builds unique combinatorial structures and uses them to perform its cognitive functions (Anderson, 1983a, Newell, 1990, Pinker, 2003, 1999). Therefore, to study cognition it essential to understand the complex structures of neurons and how they function. These intricate network of neurons are studied computationally through distributed (or connectionist) representations of interconnected units. Moreover, these distributed representations of units conjointly depict concepts because the concepts are represented by neural assemblies in the brain (Hebb, 1949). Topologically the distributed models are identical to the graph based approaches like semantic networks and ontologies. However, the operating behavior of the distributer approaches are completely different from graph-based symbolic approaches. In this section, a survey is conducted to determine how concepts are represented through a network of nodes. The review covers three types of distributed representations, i.e., Neural Assembly and Blackboard, Artificial Neural Networks and Deep Representations.

Neural Assembly

In the human brain, a neural assembly represents concepts and the neuronal structure expands or shrinks while carrying out learning operation, e.g., Hebbian form of learning (Hebb, 1949). The neuron assembly formation process can be expressed as "neuron that fire together wire together" (McCulloch and Pitts, 1943, Hebb, 1949). Therefore, the neurons involved in recognizing an object get interconnected, like a set of networked neurons is a representative of words used to recognize an animal (van der Velde, 2015, Ramachandran et al., 1998). Moreover, the whole assembly gets activated upon perceiving that object again and the activation pattern in the network depicts a conceptual representation (Quiroga, 2012). Besides, the neural assemblies can represent concepts grounded in action or emotion (Pulvermüller et al., 2001).

Figure 2.3 shows a neural sentence structure of "The Cat chases the Mouse" in a *Neural Blackboard* architecture (Van der Velde and De Kamps, 2006). A blackboard architecture is a system of interconnected processors (or demons (Selfridge, 1985)) where they communicate with one another through the blackboard (or workbench or bulletin board). Every processor has the ability to performs certain operations such as executing or modifying the information stored on the blackboard (Van der Velde and De Kamps, 2006). The Neural Blackboard consists of network and sub-network of concepts to form conceptual representations of neural assemblies for sentences.

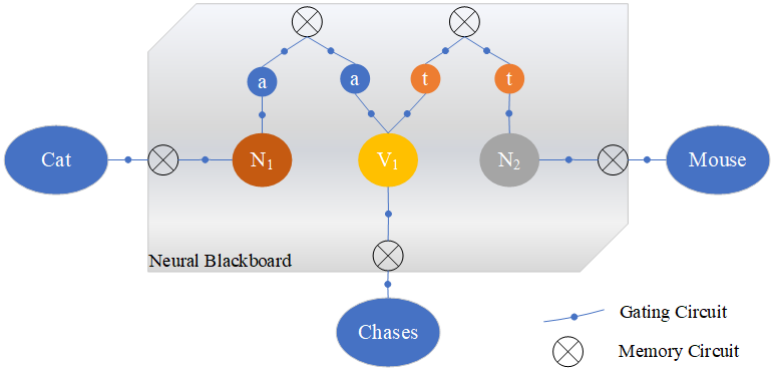


FIGURE 2.3: Illustration of the neural sentence structure of "The Cat chases the Mouse" in the Neural Blackboard architecture.

The neural blackboard in Figure 2.3 shows an assembly that binds the familiar concepts in a combinatorial structure, i.e., links concepts "Cats", "chases" and "Mouse". The words are encoded with the word assemblies. Sentence structure is encoded with "structure assemblies" for noun-phrases (NP assemblies) and verb-phrases (VP assemblies). A structure assembly consists of a main assembly and a number of sub-assemblies, connected to the main assembly by means of gating circuits. The labeled sub-assemblies represent the thematic roles of an agent *(a)*, and theme *(t)*. Binding between assemblies is achieved with active memory circuits. Here, the assembly for the cat is bound to the NP assembly N_1, the assembly for chases is bound to the VP assembly V_1 , and the assembly for the mouse is bound to the NP assembly N_1. N_1 and V_1 are bound by means of their agent sub-assemblies and V_1 and N_2 are bound by means of their theme sub-assemblies. In linguistics, such kind of binding among the concepts provides information about the sentences (Pinker, 2003, Calvin and Bickerton, 2001).

The Neural Blackboard also represents the words (concepts) of the sentence in their temporal form by interconnecting the words assemblies with the sentence structure (Van Der Velde and De Kamps, 2010, Van der Velde and De Kamps, 2006). The Neural Blackboard highlights three aspects of neural assembly structure w.r.t communication (van der Velde, 2015). First, the interconnections between the word assemblies are relational, not associative. Second, the word assemblies are an integral part of the sentence structure to which they are temporarily connected. Third, the construction of the sentence structure proceeds as a "phase sequence" of assemblies. The network structure shown in Figure 2.3 illustrates how combinatorial structures in higher level cognition are learned as conceptual representations, as learned by Hebbian neural assemblies (Hebb, 1949).

Artificial Neural Networks (ANNs)

The ANNs are bio-inspired models developed to behave like the neuron and solve complex problems computationally (Sun, 2008, Fasel, 2003). The computational units (or nodes) and their interconnections can be compared to biological neurons and the neural network. Moreover, the connections of units define the functions of the network. The ANNs can be viewed as a

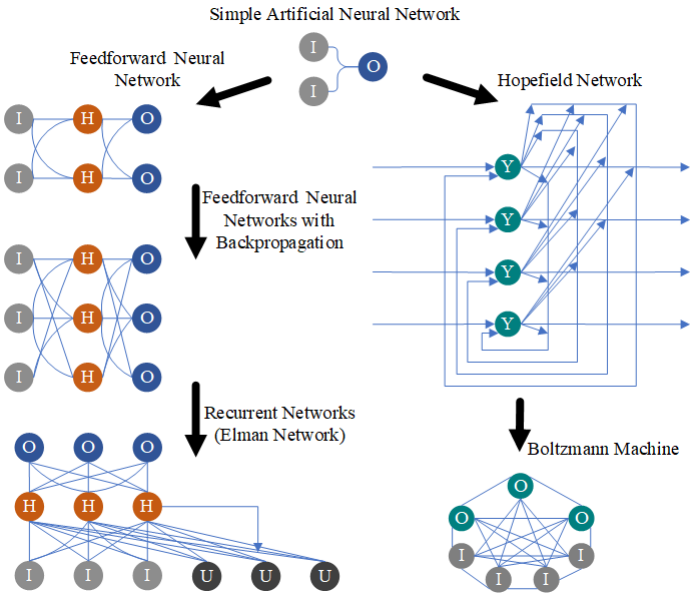


FIGURE 2.4: Different artificial neural networks.

weighted directed graph in which the nodes are *artificial neurons* and the edges are *connections* between these neurons. Figure 2.4 shows a few artificial neural networks developed over decades, all originated from simple *perceptron* based neural network (Rosenblatt, 1958, 1961, 1962). Unlike symbolic approaches, the ANNs don't represent *concepts* explicitly. At input layer, the nodes represent the features of the data and at the output layer, the nodes depict the class of the data. Furthermore, The *concepts* are *implied* by ANNs through the assembly of interconnected nodes, i.e., the nodes and their weights together represent the concepts. The intermediate nodes in a neural network are meaningless individually, although there have been efforts to determine what do these hidden nodes mean (Yosinski et al., 2015, Mahendran and Vedaldi, 2015, Liu et al., 2017).

Under the paradigm of supervised learning the neural networks can be classified into two broad categories, *feed-forward* and *recurrent*, based upon their architectural orientation. The most common form of feed-forward neural network is *multilayer perceptron* (MLP). In MLPs the network is organized into three types of layers the *input*, *hidden* and *output* layers and has a unidirectional connection among them (Haykin, 1994). Besides MLP, the Adaptive Linear Neuron (ADALINE) (Widrow and Hoff, 1960), Adaptive Resonance Theory (ART) (Grossberg, 1987) and Self Organizing Feature Maps (SOFM) (Kohonen, 1982) also belong to the family of feed-forward neural networks. The feed-forward models are static, i.e., an only single output is obtained for an input. They are generally used as classifiers determining classes of inputs features as concepts. The *recurrent* (or *feedback*) neural networks are dynamic systems (Sun, 2008) because of the interconnection of output nodes to input layer nodes acting as a response loop. The Elman Networks (Cruse, 1996), Hopfield Network (Gurney, 2014) and Boltzmann Machine (Ackley et al., 1985, Hinton et al., 1984) are a good example of a recurrent type of networks.

Learning is an essential part of intelligence. In ANNs learning is a process of updating the connection weights between the nodes to carry out the tasks effectively. To understand or design a learning process, one must first have a model of the environment in which a neural network operates, that is, one must know what information is available to the network. We refer to this model as a learning paradigm. There are three main learning paradigms: supervised, unsupervised, and hybrid. In supervised learning there exist input to output pairs that can be used for training. In unsupervised learning there are no outputs available and in hybrid there are some input-output pairs available for training. Second, we must understand how network weights are updated, that is, which learning rules govern the updating process. A learning algorithm refers to a procedure in which learning rules are used for adjusting the weights. There are two basic types of learning rules: Error-Driven, and Hebbian.

Error-Driven Learning In the supervised learning paradigm, the network is given a desired output for each input pattern. During the learning process, the actual output y generated by the network may not equal the desired output d. The basic principle of error-correction learning rules is to use the error signal $(d-y)$ to modify the connection weights to gradually reduce this error. The Back-propagation (Goodfellow et al., 2016, Amari, 1967), Gradient Descents (Barzilai and Borwein, 1988) and Reinforcement Learning (Van Hasselt et al., 2016, Sutton and Barto, 2018) algorithms are based on Error-correction.

Hebbian Learning The oldest learning rule is Hebb's (Hebb, 1949) postulate of learning. Hebbian learning is based upon the following observation from neurobiological experiments: if two neurons on both sides of a synapse are activated synchronously and repeatedly, the synapses strength is selectively increased. Mathematically, the Hebbian rule can be described as

$$w_{ij}(t+1) = w_{ij}(t) + \eta y_j(t) x_i(t)$$

where x and y are the output values of neurons i and j, respectively, which are connected by the synapse with weight w_{ij}, and η is the learning rate. Note that x_i, is the input to the synapse. An important property of this rule is that learning is done locally, that is, the change in synapse weight depends only on the activities of the two neurons connected by it. Hopfield Networks (Gurney, 2014) are an example of a particular ANN model using Hebbian Learning.

Autoencoder is another variant of ANNs learning efficient data coding in an unsupervised manner (Liou et al., 2008, 2014). With autoencoders, it is possible to regenerate a feature (concept) at the output that represents the input feature (Kingma and Welling, 2013). ANNs have been used for concepts generation such as musical melody (Todd, 1992, Eck and Schmidhuber, 2002) and their improvisation (Bown and Lexer, 2006, Smith and Garnett, 2012). ANNs are also helpful in the evaluation of concepts in music (Tokui et al., 2000, Monteith et al., 2010) and images (Baluja et al., 1994, Machado et al., 2008, Norton et al., 2010). Recurrent neural networks was also used to generate concepts such as metaphors (Terai and Nakagawa, 2009b).

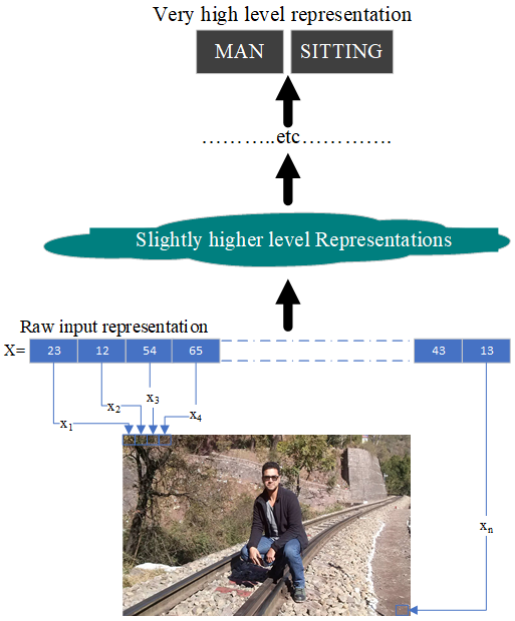


FIGURE 2.5: Levels of abstraction in an image. Inspired and adopted from (Bengio and etal., 2009).

Deep Representations

Deep representations are the most recent member of connectionist computational modeling. Deep representations are based upon the *Deep-learning* (LeCun et al., 2015) methods. The Deep-learning approaches attempt to model high-level concepts (abstract features) in data using deep architectures that are composed of multiple non-linear transformations (Bengio et al., 2013). The human brain is organized in a deep architecture (Serre et al., 2007) with a given input percept being represented at multiple levels of abstraction, each level corresponding to a different area of cortex. Humans often describe such concepts in hierarchical ways, with multiple levels of abstraction. The brain also appears to process information through multiple stages of transformation and representation. This is particularly clear in the primate visual system (Serre et al., 2007), with its sequence of processing stages: detection of edges, primitive shapes, and moving up to gradually more complex visual shapes.

Deep architectures are made up of an intricate structures having multiple layers and performing complex operations, such as in neural nets with many hidden layers (Bengio and etal., 2009). The depth of architecture refers to the number of compositions levels of non-linear operations, in the learned functions. Figure 2.5 exemplifies the core objective of deep architectures for visualizing different concept (feature) at different abstraction levels. In Figure 2.5, at the basic or lowest level of the image, each pixel can be considered as concrete concepts (features). At the intermediate levels concepts like *edges*, *object*, *shadow*, *ground* etc., can be learned. At the highest level features like *man* and *posture* are learned to infer the concept of *man sitting* (Bengio and etal., 2009).

The focus of Deep-learning technique is to discover multiple abstraction levels, from lowest level features to highest level concepts. Majority of the Deep-learning techniques are supervised using two popular approaches back-propagations (Goodfellow et al., 2016, Amari, 1967) and stochastic gradient descent (SGD) (Zhang, 2004a). Deep convolutional neural networks (ConvNet) (LeCun et al., 1990) that uses back-propagation and has achieved landmarks in image processing. ConvNet produced exceptional results in concept (object) recognition (Vaillant et al., 1994, Tompson et al., 2015) such as traffic sign recognition (CireşAn et al., 2012), identification of faces (Garcia and Delakis, 2004, Osadchy et al., 2007), pedestrians (Sermanet et al., 2013) and human bodies (Nowlan and Platt, 1994) in natural images. On the other hand, Deep Belief Network (DBN) (Hinton, 2009, Hinton et al., 2006) is an unsupervised Deep-learning methodology where several Restricted Boltzmann Machines (RBM) (Hinton, 2007) stacked over one another. For image reconstruction operations RBM was also used to build Deep autoencoder (Hinton and Salakhutdinov, 2006). Besides image analysis, deep representations have been used in modeling several domains such as language translation (Jean et al., 2014, Sutskever et al., 2014) and generation (Socher et al., 2013), producing jazz melodies (Bickerman et al., 2010). Deep-learning has also been beneficial in drug discovery (Ma et al., 2015), genomics (Leung et al., 2014) and sentiment analysis (Bordes et al., 2014).

2.1.3 Spatial Representations

The *spatial* representations are perceived based upon the feature-value (or attribute-activation) pairs of concepts. In spatial representations, concepts are viewed as a point in multi-dimensional feature hyperspace (Tversky, 1977, Marr and Nishihara, 1978). The quality dimension (features) of the spatial representations determines the characteristics of the features and the values (activation) of these quality dimensions depicts the coordinates of the concepts. The distance between n-dimensional concepts determines the similarity among them. The similarity relation among the concepts not only plays an important role in categorizing concepts (Goldstone, 1994) but also in modeling cognitive function such as recognition (Tulving, 1981). This review is carried out to understand the importance of viewing concepts in n-dimensional feature space and how the similarity relations are useful in concept learning and concept creation process. The survey covers two state of the art in computational spatial representation related Conceptual Spaces and Vector Space Models (VSM).

Conceptual Spaces

Conceptual Spaces theory (Gärdenfors, 2004) is one of the cognitive approaches that form the basis of RANs modeling. This theory views the concepts as regions within a multi-dimensional space, with the data features representing the dimensions. The *similarity* among the concepts can be identified based upon the geometrical distance between the objects. The Conceptual Spaces, thus, serves as a natural way or tool to capture the similarity relationships among

TABLE 2.1: Notations used in Conceptual Space explanation

Notation	Description
D	Quality dimensions.
v	Vector.
d	Dimension value of vector v.
x	A point in n-dimensional space.

concepts, or objects. Under this setting, one data instance corresponds to a single point in the space. Formally we can say, the *Quality Dimensions*, i.e., a set of $D_1, ...,D_n$, forms the Conceptual Space S. A point in S is represented by a vector $v= \langle d_1, ..., d_n \rangle$, where $\{1,....n\}$ are the indexes of the dimensions. Atomic concepts are *convex regions* –a convex region C having point x that falls between points $x_1 \in C$ and $x_2 \in C$ also belongs to C. The quality dimension is the basic requirement for Conceptual Spaces (Gärdenfors, 2004). An example is a color space with the dimensions Hue, Saturation, and Brightness. Each quality dimension has a geometrical structure. For example, Hue is circular, whereas brightness and saturation correspond with finite linear scales (see Figure 2.6).

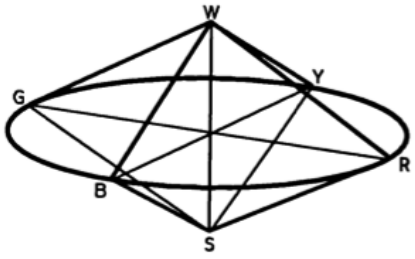


FIGURE 2.6: The color space (Sivik and Taft, 1994)

The theory of Conceptual Spaces also addresses the *prototype theory* of categorization (Rosch, 1975, Mervis and Rosch, 1981, Rosch, 1983). The main idea of prototype theory is that within a category of objects, like those instantiating a concept, certain members are judged to be more representative of the group than others. For example, robins are judged to be more representative of the category "bird" than are ravens, penguins, and emus. If convex regions of Conceptual Spaces describes concepts, then the prototype effect is, indeed, expected, i.e., the most likely central position of a convex region describes an abstract concept. For example, if color concepts in a convex region identified as subsets of the color space, then the central points of these regions would be the most prototypical examples of the color.

Clustering is a suitable way of identifying and learning atomic convex concepts in conceptual spaces. There are several clustering techniques, like hierarchical clustering, subspace clustering (Parsons, 2004), partitioning relocation clustering, density-based clustering, grid-based clustering and many more. Many are frequently used in the statistical and scientific analysis of

TABLE 2.2: Notations used in Vector Space explanation

Notation	Description
D	Document vector.
t	Term vector.
q	Query vector.

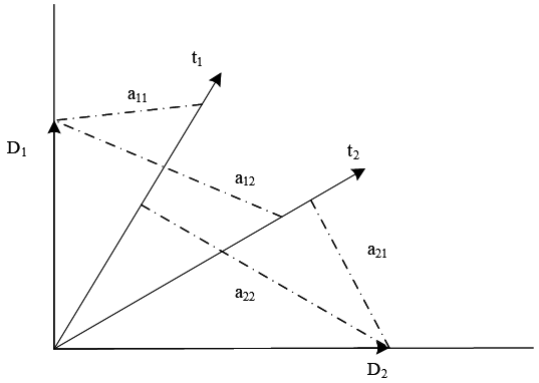


FIGURE 2.7: Components of a 2-dimensional vector space.

data (Livingstone, 2009, Yoon et al., 2001), and in machine learning for the identification of concepts/features (Van Deursen and Kuipers, 1999). On the other hand, the creation of a hierarchy of sub/super-concepts is a way to represent more abstract concepts and their taxonomic-like relations.

Vector Space Model (VSM)

A vector space model is an algebraic approach where concepts (or objects) are modeled as a vector of identifiers. The VSMs are appropriate for computing geometric representations because they provide a framework for similarity related calculations (Gärdenfors, 2004). The SMART information retrieval system (Salton, 1971) is one of the approaches that used VSM to model text documents in three steps namely *document indexing*, *term weighing* and *relevance ranking*. In *document indexing*, the terms related to content are retrieved from the document (Salton and McGill, 1986). Indexing of term can be done based upon their low or high frequency (Luhn, 1958, Rijsbergen, 1979). The *term weighing* was performed to assign weights to the indexed term of the document based upon the co-occurrence relation of the frequency of the index terms (Luhn, 1958). The last step of *relevance ranking* is carried out to rank the documents with respect to the query according to similarity measures. Normally, the similarity is obtained as a coefficient of the cosine of document vector and query vector (Salton, 1971, Salton and McGill, 1986), although, similarity can also be measured using Jaccard and Dice coefficients (Salton, 1988).

$$D_1 = a_{11}t_1 + a_{12}t_2 \tag{2.3}$$

$$D_2 = a_{22}t_1 + a_{21}t_2 \tag{2.4}$$

Figure 2.7 shows as 2-dimensional vector space for document vectors D_1 and D_2 that are expressed as a linear combination of term vectors t_1 and t_2, see Equation 2.3 and 2.4. The a_{11} and a_{12} are components of D_1 along vector t_1 and t_2 respectively, whereas, the a_{22} and a_{21} are components of D_2 along vector t_1 and t_2 respectively.

$$D_r = \sum_{i=1}^{n} a_{ir} t_i, \qquad (r = 1, 2, .., n) \tag{2.5}$$

$$q = \sum_{i=1}^{n} q_i t_i \tag{2.6}$$

$$D_r \cdot q = \sum_{j,i=1}^{n} a_{ir} q_i t_i \cdot t_j \tag{2.7}$$

In general, the document vector can be expressed as Equation 2.5. For *relevance ranking* operation a keyword search can be calculated by comparison of document and query vectors. Equation 2.6 shows the expression of a *query vector* where the q_i are the components and t_i are the terms related to the keyword being searched. The scalar product of D_r and q, see Equation 2.7, aids in determining the similarity between the query and document vectors. Each term t_i in a document D_r has their importance and they are also known as *term weights* of the document vector. Term Frequency - Inverse Document Frequency (TF-IDF) is a common weight schema which considers a term of lesser value if it appears in several documents (Salton et al., 1975). In large documents there are more distinct terms having a higher frequency, therefore, the document vectors are normalized to a unit length (Salton and Buckley, 1988).

In VSM a pairwise orthogonality is assumed between the term vector and the correlation among the terms tends to contradict this relation. However, for this issue of orthogonality, the generalized Vector Space Model (GSVM) provides a solution (Raghavan and Wong, 1986, Wong et al., 1985, 1987). The Latent Similarity Analysis (LSA) uses the VSMs technique to measure the similarity of words or text passages (Landauer et al., 1998). To create a new latent semantic space and discover high order co-occurrences LSA utilizes singular value decomposition (SVD) to a word-context matrix. Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999) is a variant of LSA that uses the statistical theory of model selection and complexity control to determine the dimensions in the new space. LSA a have been applied to build a space from Wikipedia in order to generate pictorial metaphors (Xiao and Blat, 2013). Besides word-context matrix, pair-pattern matrix has also been used to calculate the semantic relations among a word pair (Lin and Pantel, 2001, Turney and Littman, 2003). The similarity relations determined using VSMs have been beneficial in the construction of topic vectors for conceptual blending (Veale, 2012), finding bridging terms of two literature (Juršič et al., 2012) and generating advertising messages (Strapparava et al., 2007). Besides linguistics, VSMs methodology was used in the generation of audio metaphors via audio features (Thorogood and Pasquier, 2013), expressing emotions in virtual humans using RGB vectors (de Melo and Gratch, 2010).

Table 2.3: Summary of review of Conceptual Representations

Approach	Representation	Description	Concepts representative	Tools or Example
Bisociation	Association	Bisociation (Koestler, 1964) can be seen as an act of mixing unrelated (often conflicting) information in an innovative manner.	Graphs based (Dubitzky et al., 2012)	BISOCIATION (Toi.expert, 2019)
Ontologies	Semantic Relations (Santos et al., 2016)	Ontology is a study of the *state of being* (Dictionary.com, 2015c) of concepts and their relations (Poli, 1996).	Formal, prototype-based and terminological (Biemann, 2005, Sowa, 2015)	WordNet (Miller, 1998), SUMO (Pease and Niles, 2002), DOLCE (Gangemi et al., 2002) and Cyc (Lenat, 1995)
Semantic Networks	Semantic Relations (Santos et al., 2016)	A *semantic network* (Berners-Lee et al., 2001, Hendler and van Harmelen, 2008) also represents concepts and their semantic relations symbolically.	Graph and RDF triples (Lassila and Swick, 1999, Brickley, 2004, McGuinness and van Harmelen, 2004, Beckett et al., 2014)	ConceptNet (Liu and Singh, 2004)
Neural Blackboard	Connectionist	The Neural Blackboard consists of network and sub-network of concepts to form conceptual representations of neural assemblies for sentences (Van der Velde and De Kamps, 2006).	Blackboard architecture (Van der Velde and De Kamps, 2006)	Neural Blackboard (Van der Velde and De Kamps, 2006)
Artificial Neural Networks (ANNs)	Connectionist	The ANNs are bio-inspired models developed to behave like the neuron and solve complex problems computationally (Sun, 2008, Fasel, 2003)	Weighted directed graph	MLP (Rosenblatt, 1958, 1961, 1962), ADALINE (Widrow and Hoff, 1960), ART (Grossberg, 1987), SOFM (Kohonen, 1982), Boltzmann Machine (Ackley et al., 1985, Hinton et al., 1984)
Deep learning	Connectionist	The Deep-learning approaches attempts to model high-level concepts (abstract features) in data using deep architectures that are composed of multiple non-linear transformations (Bengio et al., 2013)	Weighted directed graphs	ConvNet (LeCun et al., 1990, Vaillant et al., 1994, Tompson et al., 2015), DBN (Hinton, 2009, Hinton et al., 2006), Deep autoencoder (Hinton and Salakhutdinov, 2006)
Conceptual Spaces	Spatial	Conceptual Spaces theory (Gärdenfors, 2004) theory views the concepts as regions within a multi-dimensional space and captures the similarity relationships among concepts.	Vector of Feature & Value pairs	An example is a color space with the dimensions Hue, Saturation and Brightness (see Figure 2.6)
Vector Space Model (VSM)	Spatial	A vector space model is an algebraic approach where concepts (or objects) are modeled as a vector of identifiers. The VSMs are appropriate for computing geometric representations and similarity related calculations (Gärdenfors, 2004)	Feature vectors	SMART (Salton, 1971), GSVM (Raghavan and Wong, 1986, Wong et al., 1985, 1987), LSA (Landauer et al., 1998), PLSA (Hofmann, 1999)

2.2 Abstract Concept Modeling

Abstract concepts are of immense value in our lives, they enable us in attaining abilities such as relative comprehension and effective decision making (Binder, 2016). abstract concepts are complex when compared with concrete concepts, therefore, they are puzzling and difficult to understand (Schwanenflugel, 1991). This section surveys the research and development work in conceptual and computational domains related to modeling of abstract concepts.

2.2.1 Conceptual Approaches

Investigations related to concepts is an essential domain in cognitive and psychological research. Theoretically, concepts pertain to process-oriented, symbolic or distributed, and knowledge-based conceptual representations (Kiefer and Pulvermüller, 2012). Usually, these conceptual representations are actualized via mathematical, theoretical (conceptual), or computational cognitive modeling approaches (Bechtel et al., 1998). In general, concepts are a hierarchy, where, at a higher level *abstract concepts* are placed, and *concrete concepts* are situated at, relatively, lower levels, thus, it is evident to infer the abstract concepts as the congregation of concrete concepts (Rosch et al., 1976, Tversky and Hemenway, 1984). There have been efforts to study abstract concepts mathematically (Saitta and Zucker, 1998), and theoretically (Borghi et al., 2018, 2017). In medical science, there have been significant efforts to study abstract concepts with the help of technology. One such example is MRI[1], which is being used to inspect the sections of the brain involved in abstract concept identification (Binder et al., 2005). Research in psychology has also reported investigations over abstract concepts, like probing the role of emotional content in processing and representing abstract concepts (Kousta et al., 2011).

There has been a notable contribution from cognitive, and psycholinguists in studying languages through abstract concept modeling and representations. Internally representing abstract concepts via amodal symbols like a feature list, and frames (Barsalou and Wiemer-Hastings, 2005) is among the preliminary research work in linguistics. The association and context were also established, relating abstract and concrete words (Barsalou and Wiemer-Hastings, 2005). Some research reveals that we internally recognize metaphors as abstract concepts (Gibbs Jr, 1996). An interesting work uses Distributional semantic models to study the role of linguistic and effective information while representing abstract words (Lenci et al., 2018). Moreover, abstract words might just be an indirect byproduct of co-occurrence statistics of words, which is consistent with a version of representational pluralism of abstract concepts or words (Dove, 2009, 2014). Embodied multiple representation views (Scorolli et al., 2011) (the second view of representational pluralism) of concepts states that the difference between the concrete and abstract words relies on their relative importance of exemplified and linguistic information. The Language and Situated Simulation (LASS) theory (Barsalou et al., 2008) and the Word as social

[1] Magnetic Resonance Imaging

tools (WAT) theory (Borghi and Binkofski, 2014) proposes a similar view of embodied multiple representations of concrete and abstract concepts.

2.2.2 Computational Approaches

As mentioned earlier, abstract concepts are complex and hence are difficult to understand when compared with concrete concepts. Therefore, in computational modeling realm, the concrete concepts and their representations are mainly addressed as feature recognizer (Ji et al., 2013, Ripley, 2007, LeCun et al., 2015, Anderson, 1996) and the notion of abstract concepts is mostly debated (Kiefer and Pulvermüller, 2012). However, in the field of computational linguistics, some approaches attempted to model abstract concepts in languages. Since, language is also being processed, understood, and comprehended by software through Natural Language Processing (NLP), and attempts are made to build architectures to study abstract representations of spoken language (Hill and Korhonen, 2014). Semantic similarity of nouns with the abstract concept was also investigated using semantic networks and network-based Distributed Semantic Model (Iosif et al., 2013, Iosif, 2013b). An interesting NLP research used Self-organizing Maps (SOM) to automatically extract word taxonomy from corpora (Kanzaki et al., 2004), and by constructing hierarchy by extracting semantic relation between the abstract nouns and adjectives.

Metaphors are internally recognized as abstract concepts (Gibbs Jr, 1996) and there have been efforts in NLP to study and represent metaphors in a text (Shutova, 2010). One interesting research reported the generation of metaphors computationally using recurrent neural networks (Terai and Nakagawa, 2009a) based on statistical language analysis (Kameya and Sato, 2005). In that research the metaphor generation takes place in two stages; First, knowledge structures are estimated as the latent classes of nouns and adjectives (or verbs) using a statistical language analysis. Second, the model of metaphor generation is constructed using a recurrent neural network based on the results of the statistical language analysis. Further, upon providing input sets of expressions consisting of a feature and a target to the model, the model produces the outputs which represent the input expression in the form of "A (target) like B (vehicle)". *Meta4meaning* (Xiao et al., 2016) is a metaphor interpretation method that derives word associations, by applying co-occurrence counts of words from a large text corpus and obtaining a concept's properties and their salience. A computational modeling approach used Latent Semantic Analysis (LSA) knowledge source for nominal metaphor interpretation (Kintsch, 2000). Text classification using maximum entropy (Nigam et al., 1999) is also used discriminate between literal and metamorphic use of lexical items (Gedigian et al., 2006).

Besides linguistics, attempts are made to study abstract concepts as generic or abstract features, such as automatic beat classification on ECG signals (Teijeiro et al., 2018) where a set of signals are obtained using QRS clustering (Castro et al., 2015) as abstract features. A

computational modeling approach is applied to identify abstract concepts as abstract features while processing images via deeply stacked autoencoder (Le, 2013), which is a variation of stacked autoencoder (Bengio and etal., 2009). Fuzzy C-means (Bezdek, 2013, Dunn, 1973) algorithm was used in a work to extract generic features for knowledge discovery and data mining process (Srinivasa et al., 2005). A research studied two main feature-selection measures the CFS measure (Hall, 1999) and the mRMR measure (Peng et al., 2005) and fused them into a generic-feature-selection (GeFS) measure (Nguyen et al., 2010). PCA-derived feature vectors of NORB (Huang and LeCun, 2004) dataset was also used to determine the five abstract features (four-legged animals, human figures, airplanes, trucks, and cars) in NORB data. K Nearest Neighbor (K-NN) and Support Vector Machine (SVM) algorithms were used to perform the classification tasks to identify the abstract features.

2.3 Architectural Orientation of Computational Models

Majority of the computational modeling approaches either have a pre-defined topology (i.e., number of nodes, weight connections etc.) or are symbolic, distributed or spatial representations. However, the biological neural system does not have a static configuration of neurons and synapses (Kappel et al., 2015, Holtmaat et al., 2005, Chambers and Rumpel, 2017), and uses symbolic and distributed representations conjointly (Roy, 2011). This section surveys the computational approaches based upon their architectural characteristics such as networks with dynamic properties and evolving structures, and models that are hybrid.

2.3.1 Dynamic and Evolving Topologies

Architectural variation is mainly related to connectionist computational approaches because their systems are learned as a distributed network of nodes that are connected by weights. As aforestated, a large population of computational techniques preforms modeling with a fixed structure and configurations. However, efforts have been made to develop methodologies evolve while learning and in this part, such methodologies are reviewed.

NeuroEvolution of Augmenting Topologies (NEAT) is a method of evolving the structure and weights of a neural network using genetically encoding technique (Stanley and Miikkulainen, 2002). NEAT was developed to circumvent the problem of competing conventions (Schaffer et al., 1992) also known as the problem of permutations (Radcliffe, 1993) in evolutionary computation. NEAT produces a complex network evolving from a simpler version of the neural network by allowing meaningful crossovers between the individuals having the distinct genetic characteristic. NEAT commences with a minimal population of genotype representing a simple neural network. All the genes having similar origins form the population of alike functions.

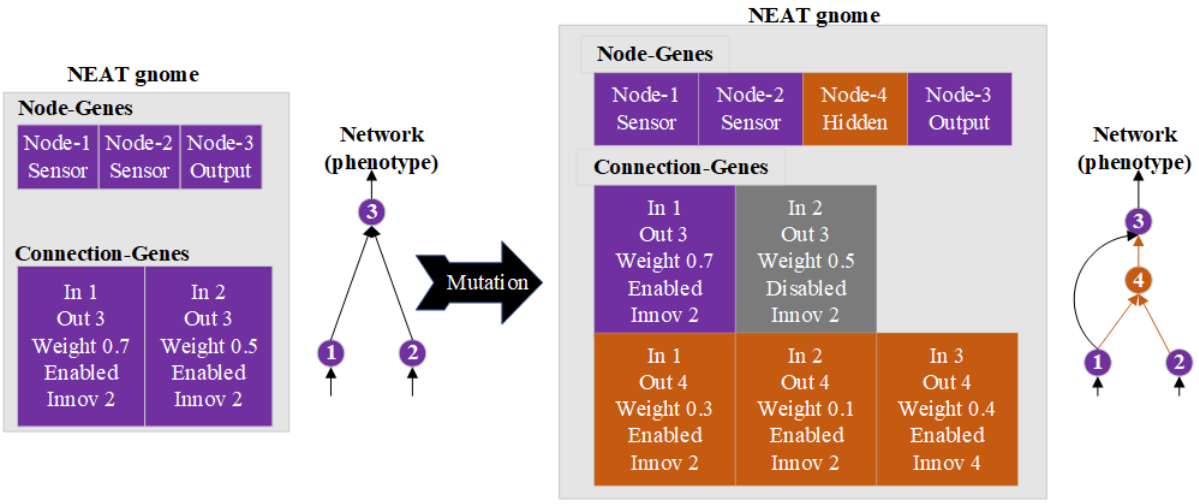


FIGURE 2.8: Structural mutation of in NEAT, inspired by (Stanley and Miikkulainen, 2002).

Every newly created gene is provided a marker (innovation number) that determines the chronological order of the created genes from its evolutionary population. The innovation number helps in finding genes having different lengths, counter competing for convention crossovers and create a sub-population genetic similar genes.

Figure 2.8 shows two types of structural mutation in NEAT. The network begins with three *Node-Genes* and two *Connection-Genes*. Creation of a new *Node-Gene* (i.e., hidden *Node-4*) is the first type of mutation, and the creation of new *Connection-Gene* (i.e., connection weights are shown with orange color in Figure 2.8) is the second type of mutation. As we can see in Figure 2.8, the selective reproductions take place between the individuals (Nodes) of the same subpopulation, i.e., *Node-1* and *Node-2* has marker number *2* and reproduces *Node-4*. This is how the genetic operators modify the genotypes by introducing the new genes and evolve the topology. If the evolved structure is beneficial it is retained and utilized further to augment the network. NEAT has been used in automobile crash warning system (Stanley et al., 2005b), pole balancing (Stanley and Miikkulainen, 2002), computer games (Stanley et al., 2005a, Reisinger et al., 2007) and robot control (Stanley and Miikkulainen, 2004).

A Model of Phenotypic Plasticity for Developing Neural Networks used a growing encoding mechanism to evolve the structure and weights of a neural network (Nolfi et al., 1994) and control a mobile robot (individual) operations. In the experiment, the brain (controller) of the individual consists of *32* neurons forming a genotype. Each neuron had a few properties (expression threshold, physical position, branching angle, weight, bias and neuron type) that manipulated while performing the model development operations. The left-most part of Figure 2.9 shows the neural network arrangements when the robot started exploring the space and growing the axons to other neurons. The middle part of Figure 2.9 depicts the network when the axon reaches a neuron. The functional part of the same network is shown by the right-most part of the Figure 2.9. While evolving the organism, its fitness is increased if it reaches the target

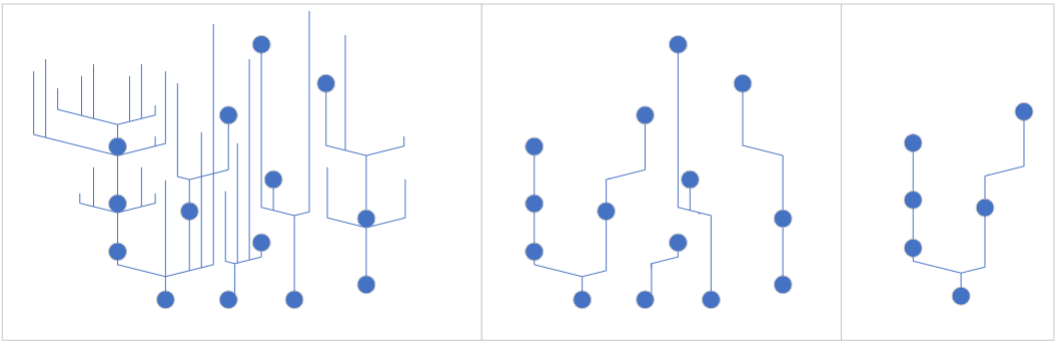


FIGURE 2.9: Neural network development while evolving an organism (robot). The left side of the figure shows the arbitrary growing and branching of axons. The middle part of the figure shows the network after removing the unconnected branches (weights). The right part depicts the functional network. The Figure is inspired by (Nolfi et al., 1994)

area, although the fitness is inversely proportional to the number of steps involved to reach the target.

Cellular encoding (cellular encoding and differentiation process) was also used to genetically encode neural networks (Gruau, 1994b). Here an entire neural network is evolved from a genotype-phenotype mapping of a single cell. In this approach also the genotype is the collection of cells that govern the cell division and transformation process, i.e. the creation of nodes and their connection weights. This technique was applied in the pole balancing problem (Gruau et al., 1996) and controlling robots (Gruau, 1994a). Cascaded-correlation neural networks (Fahlman and Lebiere, 1990) is another kind of evolving supervised neural network where the network is initialized with input and output layer and the nodes in the hidden nodes are added dynamically while learning the network. An interesting undergraduate work (Ringdahl, 2020) tried to improve the performance of the cascaded-correlation neural networks by reducing the depth of the networks and decrease in the over fitting of large networks.

Markov Brain (MB) (Edlund et al., 2011, Gruau, 1994a) is another variant of neural networks with an evolving topology. Figure 2.10 shows the components of the controller making up a Markov Brain. The controller is a type of stochastic Markov network (Koller et al., 2009) where the edges of the networks are formed by simple logic gates or complex computational units. Like ANNs the number of nodes (sensors) and connections (gates) in MB are fixed. However, the feedback mechanism is carried out by the evolved controller that is generated internally as part of MB.

The information stored in the hidden nodes of the MB resembles a *working memory* or *short term memory* because they have to be actively maintained to store information. One can view Markov Brain as an ANN that has a dynamic structure using a boolean logic function. The links, feedback operation, and control mechanisms evolve with time in MB. MBs have been used in decision-making processes (Kvam et al., 2015, Kvam and Hintze, 2018), studying the

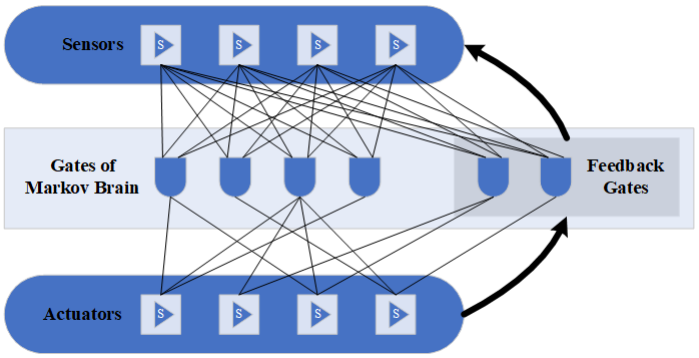


FIGURE 2.10: Components of Markov Brain with feedback gates. Inspired by (Hintze et al., 2017).

behavior of animals (Olson et al., 2013) and performing machine learning prediction (Chapman et al., 2017) and recognition (Olague et al., 2014) tasks.

2.3.2 Hybrid Computational Architectures

Computational architectures (or computational cognitive architectures) are an important part of Artificial General Intelligence (AGI) that aims in modeling human minds and intelligence and apply it to solve real-world problems. In Section 2.1 the symbolic and connectionist representations are explored, and this part reviews the architectures that are hybrid, i.e., they incorporate both symbolic and connectionist representations.

Adaptive Components of Thought-Rational (ACT-R) (Anderson et al., 1997, Anderson and Lebiere, 2003) is a hybrid computational architecture to simulate human cognitive functions and understand the intrinsic mechanisms such as perception, thinking and action. The memory modules, buffers, pattern matchers and perceptual-motors are the main components of ACT-R. There are two memory modules in ACT-R (declarative and procedural memories) realized by both symbolic and connectionist representations. The buffers serve as a volatile memory for inter-module interactions. The pattern matchers perform the comparison between the contents of procedural memory and current buffer state. The perceptual-motor acts an input and output to ACT-R architecture. In ACT-R symbolic constructs (i.e., production of the chunk) are created for each complex operations. When a task is received at input the connectionist processes determine a single rule to map the input to the chunk that handled a similar task. In ACT-R declarative, procedural (Corbett and Bhatnagar, 1997), and associative (Borst and Anderson, 2015) learning are performed. The human-like reasoning behavior is carried out in ACT-R through inherent knowledge and production rules (Nyamsuren and Taatgen, 2014). Besides, ACT-R has been used in modeling memory (Anderson et al., 1998), NLP (Lewis and Vasishth, 2005), games (Sanner et al., 2000) and psychological experiments (Anderson and Douglass, 2001).

The Connectionist Learning Adaptive Rule Induction On-line (CLARION) is a cognitive architecture that has been developed to simulate and understand processes related to cognitive psychology (such as attention, language use, memory, perception, problem solving, creativity, and thinking) and social psychology (Sun and Peterson, 1996). CLARION has a separate symbolic (explicit) and connectionist (implicit) processes and the hybrid nature of the system captures the interactions among the two processes (Sun and Alexandre, 2013, Sun and Zhang, 2004, Sun et al., 2001). CLARION has four modules of memory where each module is made up of a symbolic-connectionist pair of representations. The memory module *action-centered* subsystem processes the regulation of the agent's actions, whereas, the *non-action-centered* subsystem module maintains the generic system of knowledge. The *motivational* subsystem module provides a function to perform operations such as action, cognition and perception. The memory module *metacognitive* subsystem acts as manager to monitor the functioning of the other three modules. The learning in CLARION is declarative when chunking knowledge in declarative memory (Sun et al., 1999) and procedural through inductive inference of explicit rule extraction (Sun et al., 2011). Associative learning is performed using reinforcement methods such as Q-learning (Sun et al., 2001) or supervised methods like back-propagation (Sun, 2006). CLARION have been used to investigate psychological phenomena related to problem-solving and skill learning (Sun and Zhang, 2004) and reasoning (Hélie and Sun, 2014)

The *Learning Intelligent Distribution Agent* (LIDA) is a hybrid cognitive architecture (Franklin et al., 2014) inspired by IDA (Franklin et al., 1998) system that was designed to automate the human resource personnel called detailers (Franklin and McCauley, 2003). LIDA architecture is capable of modeling a large number of cognitive functions such as perception, memory, action selection, metacognition, and conscious type behavior (Franklin and Patterson Jr, 2006). In LIDA architecture the operations take place in three phases; first input phase where the system receives information; second, the attention phase where the information is consciously addressed; third, the action phase where the appropriate response is selected to respond to input stimuli, in this phase learning also takes place. LIDA consists of seven memory modules *sensory, sensory motor, perceptual associative, spatial, transient episodic, declarative* and *procedural* memory (Franklin et al., 2014). LIDA has six learning mechanisms, i.e., *sensory motor, perceptual, spatial, episodic, procedural* and *attentional* types of learning (Franklin et al., 2014). LIDA has been used to realize many cognitive and neuro-psychological theories such as perceptual symbol system (Franklin and Patterson Jr, 2006) and Global Workspace Theory (Franklin et al., 2007).

Besides the aforementioned architectures, The DUAL (Kokinov, 1994), Polyscheme (Trafton et al., 2005, Cassimatis et al., 2004, Cassimatis, 2001), SOAR (Laird, 2012), Novamente (Goertzel et al., 2008), Copycat/Metacat (Hofstadter, 1984, Marshall, 2006) are a few popular hybrid cognitive architectures. All these computational architectures present a kind of blueprint for general intelligence, or try to represent several mental or cognitive functions in order to study

intelligent behavior (Kotseruba and Tsotsos, 2018). All these architecture are hybrid in the sense that they consist of models of agents that are either connectionist or symbolic in nature.

2.4 Machine Learning for Concept Identification

Whenever there is a requirement of identifying groups (of objects, features etc.) based upon the similarity of the elements, we make use of clustering techniques to fulfill the requirements. There exist several clustering techniques, like hierarchical clustering, partitioning relocation clustering, density-based clustering, grid-based clustering and so on, many are frequently used in statistical analysis of data (Yoon et al., 2001). Besides, clustering also finds a significant place in scientific data analysis (Livingstone, 2009), knowledge and information retrieval from databases and big data, analysis of WWW, in biological data analysis, in medical diagnosis etc. Clustering plays a vital role in reducing dimensions (Kaski, 1998) (the curse of dimensionality) of data, though, it introduces sought of loss in the data but reduces the complexity of the data significantly. In this thesis clustering plays an important role because in machine learning clustering aids in identifying the concepts (features or data-points) (Van Deursen and Kuipers, 1999). This chapter surveys three clustering techniques in order to understand their usability as *concept identifiers* for RANs modeling.

2.4.1 Hierarchical Clustering

The Hierarchical clustering, as the name depicts, is a cluster analytics approach where a hierarchy of clusters are build to perform the clustering operations. There are two broad approaches in Hierarchical clustering, i.e., Top-down, and bottom-up. In this section, both type of clustering is surveyed for concept identification. Agglomerative clustering is one of the hierarchical cluster analysis technique (HierarchicalClusteringWiki, 2015). It uses a bottom-up approach to construct the hierarchy, here, at the lowest level each node represents a cluster of its own, for each new layer these nodes group together pair-wise to form a new node. The grouping operation stops when a single node is left representing the highest level. To decide upon which nodes (or clusters) should be combined, a measure of dissimilarity (usually Euclidean distance) is applied, and a linkage criterion (like Centroid link, average link, single link, complete link (Olson, 1995, Murtagh, 1984)) is used to determine the pair-wise linkage between two set of observations. The Figure 2.11 shows a dendrogram of a sentence *"HOW ARE YOU"* at the basic level each letter is a cluster itself and at each level above the pairs are formed either by the pair of the two words, or by the pair of word and *linkage* of another pair of word, or between the pair of *linkage* of letters. At the highest level, the whole sentence is represented by a cluster.

The example depicted by Figure 2.11 can be seen as a hierarchical representation of concepts (or features), where the at the lowest level features are the alphabets, and the above level

represents the abstract representation of the lower level concepts (or features), the top-most level represents the most abstract form of the concept (or feature). The agglomerative clustering is widely used with spatial data to detect features, a system to recognize faces (Barbu, 2013) uses the technique for detecting faces via Scale-invariant feature transformation (SIFT) characteristics and agglomerative clustering. An algorithm named CURE (Cluster Using REpresentatives) (Guha et al., 1998) is compatible with numeric feature values of low dimension spatial data. This agglomeration algorithms has distinctive features like the novelty in dealing with outliers through a separate label assignment stage. The linkage among the cluster pair is established using the centroids of the clusters. ROCK (Robust Clustering Algorithm for Categorical Data) proposed by (Guha et al., 1999) was capable of working with categorical data. The hierarchical agglomerative clustering approach named CHAMELEON was proposed by (Karypis et al., 1999b). It consists of two phases: in the first, small clusters of closely related nodes (concepts or features) are formed; in the second, phase the agglomerative clustering takes place. The dynamism in the model is brought-up by using the relative inter-connectivity and closeness of the centers of the clusters. The level of hierarchy is to be provided by the user as threshold. Agglomerative clustering not only helps in feature detection for spatial data but also can be used to identify temporal patterns (sequences). A temporal model (Karypis et al., 1999a) for learning and recognition uses agglomerative clustering along with Hidden Markov Models (HMMs) (George, 2008) to detect sequences. Agglomerative clustering in an essence is a prominent tool of machine learning and is very useful in achieving objectives like knowledge discovery, feature recognition (or detection), classification (or categorization), concept learning and so on.

Similar to agglomerative clustering (bottom-up approach) there exists a top-down clustering approach named Divisive Clustering (HierarchicalClusteringWiki, 2015). In this clustering technique, the process starts with a set of data, then the data is split to form clusters representing specific (or concrete) features (or data), this process continues until a stopping criterion. Top-down clustering is conceptually more complex than bottom-up clustering since we need a second, flat clustering algorithm as a "subroutine". It has the advantage of being more efficient if we do not generate a complete hierarchy all the way down to individual document leaves. For a fixed number of top levels, using an efficient flat algorithm like K-means, top-down algorithms are linear in the number of documents and clusters. So they run much faster than HAC algorithms, which are at least quadratic.

There are evidences that divisive algorithms produce more accurate hierarchies than bottom-up algorithms in some circumstances (HMMWiki, 2015). Bottom-up methods make clustering decisions based on local patterns without initially taking into account global distribution. These early decisions cannot be undone. Top-down clustering benefits from complete information about global distribution when making top-level partitioning decisions.

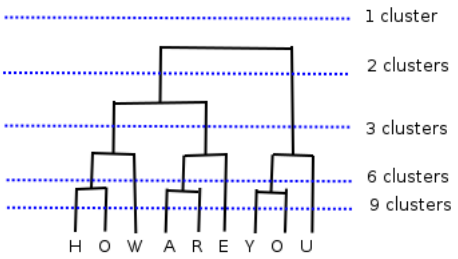


FIGURE 2.11: Dendrogram showing bottom-up Agglomerative clustering

2.4.2 K-means Clustering

Suppose there are n data points $\mathbf{x}_i$, i=1...n that have to be partitioned in k clusters. The goal is to assign a cluster to each data point. K-means is a clustering method that aims to define k centroids of k cluster that minimizes the distance of centroids and data points of every cluster. K-means clustering solves:

$$\arg\min_c \Sigma_{i=1}^{k} \Sigma_{x \in c_i} \; d(x, \mu_i) = \arg\min_c \Sigma_{i=1}^{k} \Sigma_{x \in c_i} \parallel x - \mu_i \parallel_2^2$$

where c_i is the set of points that belong to cluster i, k is number of clusters, and μ is the cluster mean. The K-means clustering uses the square of the Euclidean distance $d(x, \mu_i) = \parallel x - \mu_i \parallel_2^2$ This problem is not trivial, so the K-means algorithm only tends to find the global minimum, with a possibility of getting stuck into local optimum.

The K-mean (Hartigan and Wong, 1979) algorithm (also known as Lloyd's algorithm (Lloyd, 1982)) partitions the data into k mutually exclusive clusters. K-means treats each observation in the data as a feature (or object) having a location in space. It intends to find a partition in which feature within each cluster such that they are closest to other feature in the same cluster and far from the objects from other clusters. Further K-means determine the centroid of each cluster and uses an iterative procedure to minimize the sum of distances between each object with its cluster. The centroids can also be identified using Forgy's algorithm (Forgy, 1965), in this, initially all the points are assigned to nearest centroid then re-compute the centroid for the newly formed cluster. Figure 2.12 shows K-means cluster convergence (RbloggersKmeansplots, 2015) for an iris data set, it also shows the centroids. Though K-means is an efficient approach there is a drawback to this approach, the count of clusters is to be provided as input to the algorithm, this may produce inappropriate results with a wrong assumption for the cluster count, several approaches (Bardley and U, 1998, Babu and Murty, 1993, Chiu et al., 2001) suggest some measures that help to make proper assumptions for the cluster counts.

K-means algorithm work with numerical data (Jain and Dubes, 1988) as it minimizes the cost function of the means of the clusters. This limits the usage of the K-means algorithm for categorical values, but an approach (Ralambondrainy, 1995) used binary values to depict the presence of category with K-means, apart from this HUANG (Huang, 1998a) proposed an algorithm using K-means paradigm to cluster categorical data. Usually k-means algorithm is used to cluster real-valued data and not considered suitable for binary data as they consider distance as a metric to form clusters. A technique to cluster binary data streams with K-means is proposed in (Ordonez, 2003) where: first, a sparse distance computation takes place by pre-computing the distance of the null transaction (i.e., zero's on all dimensions) with the centroids; and afterward the computed distance is used to identify the clusters.

K-means algorithm is not only easy to implement but also can be applied to large data sets. It finds its application in market analysis, geostatistics, signal processing, computer vision, etc. Apart from this, it is very helpful in performing machine leaning task such as feature (or concept) learning (Coates and Ng, 2012) and entity recognition (Lin and Wu, 2009). In case of object identification K-means is found competent to other feature learning techniques like Restricted Boltzmann Machines.

A variation of K-means algorithm is K-mode (Huang, 1997) and K-prototype (Huang, 1998b) algorithm. K-mode extends K-mean in essence that it aims to cluster categorical data. It uses modes instead of mean and frequency-based approach to updating modes. K-prototype is an integration of K-means and K-mode in an attempt to cluster data based upon both numeric and categorical values. A similar approach in clustering related to K-means is the K-medoids algorithm. Both K-means and K-medoids attempts to minimize the distance between cluster members (points or features) and the centroid. K-medoids differs from k-mean because K-medoids chooses data-points as centers (medoids or exemplars) and works with an arbitrary matrix of distances between data-points. It is more robust to noise and outliers as compared to K-means because it minimizes a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances. Mainly used variation of K-mediods are PAM (Partitioning Around Mediods) and CLARA (Clustring LARge Applications) (Kaufman and Rousseeuw, 1990). PAM operates iteratively to optimize the mediods as a combination of the relocation of points between the perspective clusters. CLARA is a combination of a sampling procedure and PAM.

2.4.3 Fuzzy Clustering

Several clustering techniques have been introduced until now, a few have been discussed in the previous sections, and these techniques can be broadly classified into two categories i.e. *soft (or fuzzy)* and *hard (or crisp)*. *Hard* Clustering techniques mostly correspond to classical set theory satisfying the property of disjoint sets, i.e. the clusters should be a mutually exclusive set, satisfying the property $A \cap B = \emptyset$, (A *&* B are some arbitrary sets). *Fuzzy* clustering

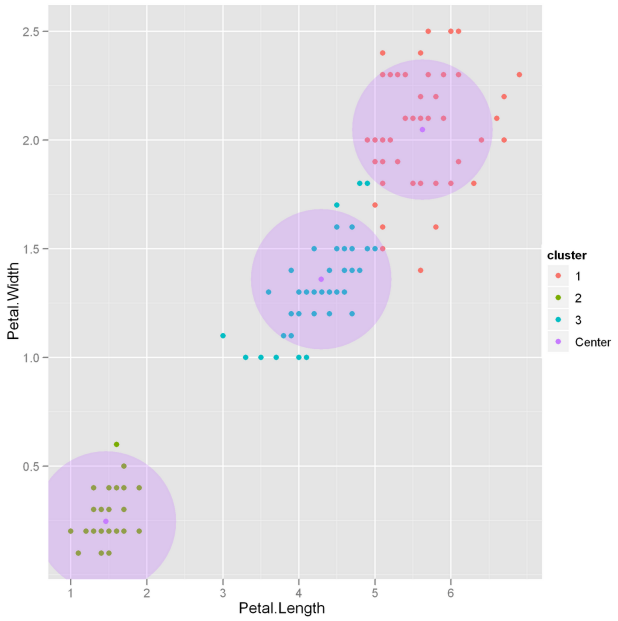


FIGURE 2.12: K-means convergence (KmeanClusterImage, 2015)

techniques are based upon fuzzy set theory (Zadeh, 1965) and make use of fuzzy set properties (Mitsuishi et al., 2001) allowing an element (of the set) to have a membership of different clusters. This technique is very suitable for real-world applications because the outliers and the boundary elements of a cluster can attain membership of more than one clusters. The *fuzzy* clustering requires a *fuzzy* type partitioning mechanism to form clusters. The *fuzzy* partitioning space for a data set is:

$$M_{fc}= \{ \mathbf{U} \in \mathbb{R}^{c\times N} \mid \mu_{ik} \in [0,1], \forall k; \Sigma_{i=1}^{c}\mu_{ik} = 1, \forall k; 0 < \Sigma_{k=1}^{N}\mu_{ik} < N, \forall i\}$$

U is the partition matrix, c is number of clusters (assumed, c to be known in advance), μ is the membership function that decides the existence of each node (feature or element), N is set of observations, $i \Rightarrow 1 \leq i \leq c$; and $k \Rightarrow 1 \leq k \leq N$. For example, consider the Figure 2.13 showing six points $x_1 ..... x_6$ and data set z_1={1.0, 1.0, 1.0, **0.5**, 1.0, 1.0} and z_2={1.0, 1.0, 1.0, **0.5**, 1.0, 1.0}, the data point x_4 has membership in both the clusters with the value **0.5**. The work proposed in (Bellman et al., 1966, Ruspini, 1969) introduced the application of *fuzzy* sets in performing classification and cluster analysis. *Fuzzy* clustering can be based upon *fuzzy* relations as well as *fuzzy* objective functions (Yang, 1993).

Fuzzy clustering is very helpful extracting features and representing patterns (Mernberger et al., 2011), apart from this, the inclusion of fuzzy partition or membership functions of the elements (feature or concepts) makes the machine learning techniques to deal with ambiguity in choosing

elements while forming clusters. The most popular algorithm for fuzzy clustering is *Fuzzy c-means* (Bezdek, 2013, Dunn, 1973) algorithm. It has been successfully applied in the field of pattern analysis (Pham et al., 1997, Bezdek et al., 1992) and analyzing geographic-specific data (Rignot et al., 1992, Chumsamrong et al., 2000). The main objective of *fuzzy c-mean* is to minimize the objective function:

$$J_m(Z;U,V) = \Sigma_{i=1}^{c}\Sigma_{k=1}^{N}\mu_{ik}^{m} \parallel z_i - v_k \parallel_A^2$$

where

$$\mathbf{U} = [\mu_{ik} \in] \; M_{fc}$$

is a fuzzy partition matrix of Z,

$$\mathbf{V}= [v_1, v_2, ..., v_c], v_i \in \mathbb{R}^n$$

is a vector of cluster prototypes(centers), which have to be determined,

$$D_{ikA}^2 = \parallel z_k - v_i \parallel_A^2 = (z_k - v_i)^T \mathbf{A} (z_k - v_i)$$

is a squared inner-product distance norm (A is a norm-inducing matrix), and
$1 \leq m \leq \infty$ is a parameter which determines the fuzziness of the resulting clusters.

The *fuzzy c-means* algorithm minimizes the previously mentioned objective function making use of a Lagrange multiplier, and obtain the updated membership μ for the elements and their mean using a *fuzzy c-mean* algorithm (Bezdek, 2013, Yang, 1993, Dunn, 1973). To use the *fuzzy c-means* algorithm, a few parameters must be taken care of, like specifying number of clusters initially, the fuzziness exponent, initialization of fuzzy partition matrix, termination criterion, and a norm-inducing matrix for distance calculation. *Fuzzy c-means* algorithm has been applied to cluster objects based upon the biased-observations of the observer (Fazendeiro and Valente de Oliveira, 2015) to deduce meaningful clusters. A clustering architecture (Pedrycz, 2002) also makes use of *fuzzy c-means* algorithm as inspiration to find common structures, for a given subset of patterns. A fuzzy clustering algorithm named *DifFUZZY* (Cominetti et al., 2010) is an efficient clustering algorithm to handle larger datasets that are curved, elongated or clusters having different dispersion, this technique utilizes the technique diffusion process in graphs.

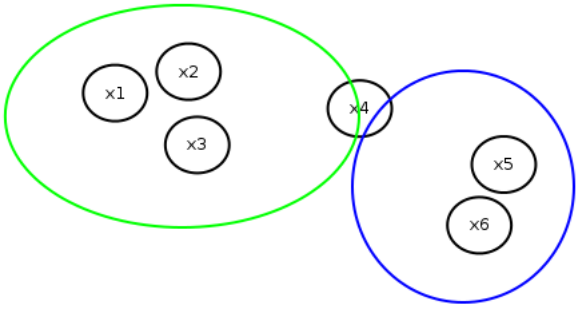


FIGURE 2.13: *Fuzzy* partitioning of data

Chapter 3

Convex Abstract Concept Modeling with Regulated Activation Networks

This chapter describes the RAN's convex concept modeling technique and is based upon the following International Conferences and Technical report, and part of under review Journal article. Following is the list of articles and the Technical reports:

- Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto and Amílcar F. Cardoso, *Exploring Geometric Feature Hyper-Space in Data to Learn Representations of Abstract Concepts.* Applied Sciences, 2020. (Sharma et al., 2020a)
- Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto and Amílcar F. Cardoso, *Modeling Abstract Concepts For Internet of Everything: A Cognitive Artificial System.* 13th APCA International Conference on Control and Soft Computing (CONTROLO), Azores, Portugal, June 2018. (Sharma et al., 2018b)
- Rahul Sharma, Alexandre Miguel Pinto, Vivek Kumar Singh. *Energy and Resource Usage-Aware Building Via Cognitive Internet of Things Agents.* 3rd Energy For Sustainability (EFS) Conference Madeira, Portugal, 2017. (Sharma et al., 2017a)
- Rahul Sharma, B. Ribeiro, Alexandre Miguel Pinto, and A. F. Cardoso, *Computational concept modeling for student centric lifestyle analysis: A technical report on socialite case study*, (Sharma et al., 2017b)
- Alexandre Miguel Pinto, Rahul Sharma, *Regulated Activation Network*, deliverable D2.4 Project Concept Creation Technology (ConCreTe), Small or Medium-Scale Research Project ICT- Future and Emerging Technology (FET) `http://conceptcreationtechnology.eu/?q=node/43`. Technical Report, 2016.

Table 3.1: Notations

Notation	Description
W	Inter-Layer weight matrix.
A	Output Activation vector.
a	Input Activation vector.
n	Size of a vector.
n_a	Number of elements in input vector at Layer l.
n_A	Number of elements in output vector at Layer $l+1$.
l	l'th Layer representative.
d	Normalized Euclidean distance
C	Cluster center or Centroids.
i, j, k	Variables to represent node index for input-level, abstract-level, and arbitrary node index in either of the levels, respectively.
t	Iterator variable.
$f(x)$	Transfer function to obtain similarity relation.
D-n	n^{th} dimension value of an input data instance.
H-n	n^{th} dimension value of an input data header vector.
N	Abstract node label.
S	Input node label.

3.1 Assumptions, Limitations and Data Pre-Processing

The boundary related to input data is, data value should be between "0" and "1" (both inclusive), this limitation has its inspiration from biological neurons. A value "0" indicates neuron (or node) is inactive, whereas "1" shows the neuron is highly active. The model is, by design, applicable only to multidimensional data sets where each feature takes a real value between 0 and 1 – it works as well for discrete data sets where the variables take either 0 or 1. If the user data is in a different format, the user must define the transformation and inverse transformation of the data. The following are a few possibilities of such alterations for some of the most common kinds of data:

- If a variable in the input data is categorical, e.g., *blue*; *green*; *red*, transform the data using One Hot Coding technique. For example, transform *blue* into (1 0 0), *green* into (0 1 0), and *red* into (0 0 1);
- If a variable in the input data is numerical, bounded within a minimum and a maximum value it can be normalized into $[0, 1]$, e.g., via $\frac{value-min}{max-min}$;

The user must implement these transformations and their inverse transformation functions to interpret the results obtained from RAN's model. Since RANs modeling technique is designed to work with multivariate datasets, where each data value is a point in Conceptual Space, it is assumed that the data, being used, is compatible with the requirements. Though images are a form of multivariate data, pictures are not ideal candidates to be interpreted as points in conceptual spaces, (discussed in Section 2.1.3). For this reason, RANs modeling will, most probably,

Table 3.2: Input Data Format for implemented RANs Modeling

Header	**H-1**	**H-2**	**.....**	**H-n**
Data Instances	D-1	D-2	.....	D-n
	D-1	D-2	.....	D-n
	.	.	.....	.
	.	.	.....	.
	.	.	.....	.
	D-1	D-2	.....	D-n

underperform on image processing tasks against other models that are, individually, designed for this kind of data, such as deep representations built with Convolutional Networks (Eigen et al., 2014, Kavukcuoglu et al., 2010, Sermanet et al., 2014); RANs modeling technique is preferably suitable for understanding and simulating cognitive processes like abstract concept Identification.

The implemented RANs modeling tool in python takes input data in a specific format (shown in Table 3.2). Besides the n-dimensional sample data (D-1, D-2, ..., D-n), the inputs require a header as the first row stacked over the original data. Each header element, $[H-1, H-2,$ ....., $H-n]$, is the Maximum value possible for their respective column (feature, or dimension). The objective of adding a header is to use it in normalizing the data between 0 and 1 while preprocessing. It is assumed that the minimum value of the column is zero, if it is not the data must be transformed between zero and the maximum positive.

3.2 Introduction

The term *concept* itself has a lot to say about itself. Anything can be seen as a concept, whether it is a living being, or a thing, or an idea. The appellation concept automatically coins the need to understand their representations. There are several conceptual representation theories (Kiefer and Pulvermüller, 2012) like *modality-specific*, *localist-distributed*, *experience-dependent* (Xiao et al., 2019). Such representations not only enable us to understand the various cognitive processes in humans but also the psychological ones, like creativity. Each theory has a way to represent concrete concepts through perception, action, emotion, and introspection, but the notion of abstract concepts is debatable (Kiefer and Pulvermüller, 2012). In RAN's modeling, this issue is addressed computationally by simulating and studying the formation of convex abstract concepts.

Computational models provide us algorithmic specificity, conceptual clarity, precision. Besides, they empower us to perform simulations that can either be useful to test and validate psychological theories or to generate new hypotheses about how the mind works – this has turned them into an indispensable tool to study the human brain. The literature (Braver et al., 1999,

O'Reilly, 2006, Rolls et al., 2008) shows that this ambitious goal is not out of reach of computational cognitive modeling. Furthermore, this type of computational tools with the ability to capture cognitive phenomena has also the potential to simulate and study some mental states and processes such as those linked to creativity (Kyaga et al., 2013). Several computational

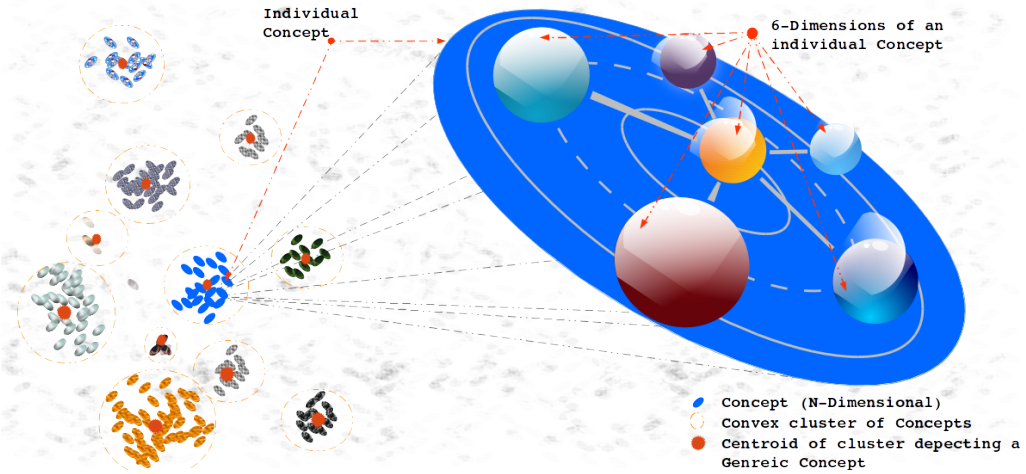


FIGURE 3.1: The universe of concepts in six-dimensional feature hyperspace. The ovals in the diagram depict an individual concept. Each *individual concept* is described by their defining 6-dimensions. The cluster of concepts shows the groups formed by similar concepts represented by *convex cluster of concepts*, and the *cluster centers* depict the most generic concept of the cluster.

modeling techniques (or tools) simulate cognitive states and represent concepts at symbolic and connectionist levels. Symbols represent information at a symbolic level. Rules are defined to manipulate symbols. Within a symbolic representation, the meaning is internal to the description itself; symbols have sense only regarding other symbols, and not regarding any real-world objects or phenomena they may represent. Adaptive Control of Thought-Rational (ACT-R) (Anderson et al., 1997) is an example of symbolic approaches, with contributions in, almost, all field of AI (such as language processing, perception, and attention, decision making, etc.). At the connectionist level, information is represented by the dynamics over densely connected networks of primitive units. A particular strength of connectionist networks is their ability to adapt their behavior according to observed data.

The weights among the units of a distributed network represent the learned behavior, they offer limited explanatory insights into the process, being modeled. Bioinspired Artificial Neural Networks (ANN) such as Restricted Boltzmann Machine (RBM) (Hinton, 2012), Autoencoders (Vincent et al., 2010), Deep Neural Networks (Collobert and Weston, 2008) are some excellent such examples with a significant contribution toward classification and perception (or recognition). A third way constitutes a hybrid view of connectionist, and symbolic methods. Connectionist Learning with Adaptive Rule Induction Online (CLARION) (Sun and Peterson, 1996) is a methodology that is hybrid, and capable of simulating scenarios related to cognitive and social psychology. All these methodologies either require a predefined structure or have a

fixed topology that imposes a limitation of having supervision, and inflexibility while modeling the concepts.

This chapter proposes a computational method named Regulated Activation Network (RAN) which unifies the virtues of symbolic, distributed and spatial representations to represent concepts (both concrete and abstract). RAN has a graph-based topology hence it is distributed, every node in the graph (network) identifies an entity, therefore, it's symbolic, and every node (or entity) is viewed in an n-dimensional feature space, hence, it's also spatial. The spatial view of concepts as points in multidimensional geometric feature space (see Figure 3.1 for 6-dimensional View of Concepts) is inspired by the theory of conceptual spaces (Gärdenfors, 2004). The RAN's modeling has an evolving topology that enables it to build a model depicting a hierarchy of concepts. The geometrical associations among concepts aid in determining the convex abstract concepts. Further, the representatives (nodes) of the abstract concepts form a new layer dynamically, where each node acts as a convex abstract concept representative for the underlying category. Symbolically, the concepts at (relatively) lower level in the hierarchy are identified as concrete concepts and the concepts at (relatively) higher level are seen as abstract concepts.

The first experiment in this chapter, demonstrates the RAN's methodology with the help of an artificial dataset (Toy-data1) and describe how the three cognitive function (i.e. concept creation, learning and activation propagation) are simulated. The second experiment in the chapter, shows the two distinct aspects of RANs modeling: first, flexibility in choosing a clustering algorithm as concept identifier; second, deep model generation. The third experiment of the chapter shows how RANs modeling can instinctively identify of generic concepts in data and build a representation for it. The proof of concept of RANs classification capability is provided by testing RAN's approach with eight UCI benchmark datasets. The generated models were evaluated using metrics precision, recall, F1-score, accuracy and Receiver Operating Characteristic (ROC) curve analysis. The RAN's classification performance was also compared with five machine learning techniques, Multilayer Perceptron (MLP) (Rumelhart et al., 1986), Logistic Regression (LR) (Freedman, 2009), K Nearest Neighbors (K-NN) (Altman, 1992), Stochastic Gradient Descent (SGD) (Zhang, 2004b) and Restrict Boltzmann Machine (Hinton, 2012) pipelined with Logistic Regression (RBM+).

The chapter is organized as follows: RAN's methodology is detailed using a Toy-data problem in Section 3.3; Section 3.4 shows the experiments with two datasets from UCI machine learning repository to exhibit: *(1)* flexibility in choosing a suitable concept identifier, *(2)* a deep hierarchy of abstract concepts, and *(3)* automatic association of input-labels to their respective abstract concept nodes; Section 3.5 provides RAN's comparisons with five classifiers and proof of concept with eight benchmark datasets; Finally, Section 3.6 summarizes and concludes the chapter advising links for Chapter 4.

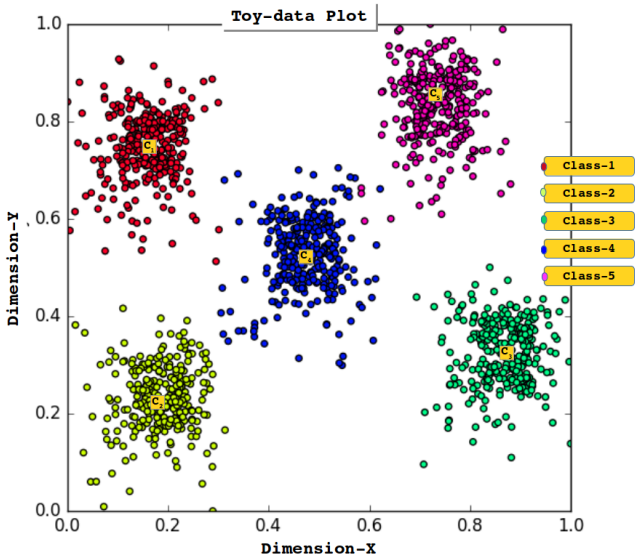


FIGURE 3.2: Plot of Toy-data, a 2-D artificially generated data. The plot shows five classes along with their cluster centers

3.3 Abstract Concept Modeling with RANs

The proposed approach models convex abstract concepts through four core steps (i.e., *convex concept identification, convex abstract concept creation, convex concept inter-layer learning and convex abstract concept upward activation propagation*), along with one optional step (i.e., *Abstract Concept Labeling*). The RAN's methodology is illustrated using an artificially generated 2-dimensional Toy-data with 1500 instances that are evenly distributed into five categories (see Figure 3.2). At the end of this section the model evaluation and comparisons are reported, by performing thirty iterations for each experiment of RANs modeling. Every experiment applied nine Research Designs (RD) (see Table B.1 in Appendix B), where, in every RD a 10-fold cross-validation procedure was applied. The main objective of the experiment with Toy-data1 demonstrate RANs modeling process and to simulate the cognitive process of *concept creation*, *learning*, and *activation propagation*. For this experiment, it was hypothesized that the created convex abstract concepts symbolically represents the 5 classes in the Toy-data1. To prove the hypothesis classification operations were performed using RAN's modeling with Toy-data1 the results are reported in section 3.3.5.

To use the RAN's approach the data is provided to the model with an additional header stacked over the data as already mentioned. The size of the header is the same as the dimension of the input data vector, and each header element holds the largest value of their corresponding input data attribute. See Section 3.1 for elaboration.

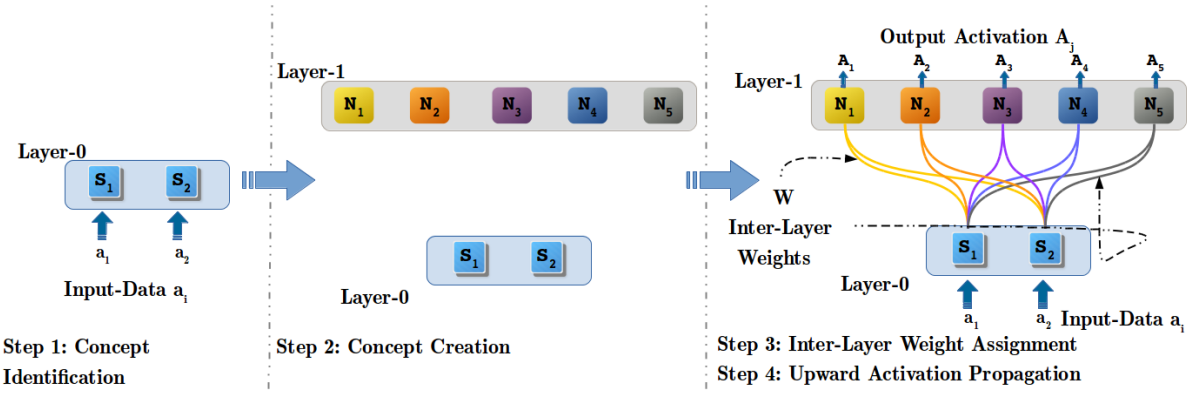


FIGURE 3.3: Steps in the model generation with Regulated Activation Networks

3.3.1 Step 1: Convex Concept Identification (CCI) Process

Concept identification is the first step in RANs modeling. The objective of the CCI procedure is to appropriately identify each instance within the data as a distinguished member of various underlying convex groups. This is realized by categorizing the input data based upon their geometrical relationship, i.e., distance, conforming to the theory of conceptual spaces (see Section 2.1.3). Here, we also recognize data points that are the most probable representative of each identified group, complying with the prototype theory (see Section 2.1.3).

The process of CCI instantiates after preprocessing the input data. Initially, an input layer is formed, with dimension equal to the size of input data feature vector. Step 1 in Figure 3.3 shows the Layer-0 with two nodes which is like the magnitude of the input vector of 2-Dimensional Toy-data. At Layer-0, clustering methods are used to determine geometrical relation among the several input data instances and identify the underlying categories within the data. Thus, K-means (Hartigan and Wong, 1979) clustering algorithm is applied to Toy-data to identify five classes (Class-1, ..., Class-5) by assigning a value 5 to 'K' in the K-means clustering algorithm. Figure 3.2 shows the plot of 2-D data points obtained after performing a concept identification operation using the K-means algorithm. Figure 3.2 also displays the centroids (C_1, ..., C_5) of all the clusters, recognized as Cluster Representative Data Points (CRDP) of all five classes.

Any clustering algorithm can act as a concept identifier in RANs modeling if it suffices two basic requirements. First, the algorithm is able to determining convex categories based upon their geometric relationship among the data instances. Second, the algorithm recognizes CRDPs of all the identified clusters. This flexibility of choosing a suitable method for concept identification process in RANs modeling is demonstrated by a separate experiment using Affinity propagation (Frey and Dueck, 2007) clustering algorithm, in Section 3.4.1.

3.3.2 Step 2: Convex Abstract Concept Creation (CACC) Process

Concept creation is a cognitive process to create representation of a newly identified concept. In RANs modeling this process is simulated by CACC operation by forming a new layer dynamically, where each constituent node in the new layer acts as an abstract representative of their respective categories identified in the CCI process. The Step-2 in Figure 3.3 shows the newly created layer (Layer-1), that has five nodes (N_1, ..., N_5), corresponding to five classes (see Figure 3.2), identified in CCI operation with Toy-data.

Besides abstract representation of underlying categories, the activation of nodes in newly created layer discloses the degree of confidence (DoC)[1] indicating the certainty of identification of a class by its representative node in the new layer (for a given input data instance). For example, if a node (say N_1) gets an activation of 0.85, it can be stated that with a confidence of 85% the input data belongs to the category being represented by node N_1. Thus, for all input data instances, the obtained ⟨*feature, value*⟩ pair of ⟨*abstract-node, Activation-value*⟩ at new layer adds more meaning. For instance, in Figure 3.3, Step-2, at Layer-0 input vector is [0.1, 0.21] it means that the dimensions S_1 and S_2 have activation 0.1, and 0.21 respectively. For the, aforementioned, input vector, [0.13, 0.32, 0.89, 0.16, 0.05] vector of activation is observed at all nodes (N_1, ..., N_5) respectively, at Layer-1. The observed activation vector itself describes that the input data belongs to Class-3 with a DoC of 89%.

3.3.3 Step 3: Convex Concept Inter-Layer weight (CCILW) Assignment

Learning is an important cognitive process that create a relationship among the concepts. In RAN's convex concept modeling the inter-layer layer learning is the relationship established between two adjacent layers. After the generation of the new layer dynamically, the next step in RANs modeling is to establish a relation between these two layers (input-layer and new layer). As aforestated in Section 3.3.2 that each node in the new layer is an abstract representative of categories identified in CCI process, thus we learn association among the two-layer such that it substantiates the abstract representation by the nodes at the new layer. Since discovered CRDPs (see Section 3.3.1) are the most apparent choice as an abstract representative of a cluster (and adhere to the inspiration from prototype theory); consequently, the CRDPs are assigned as an association between the two layers.

Equation 3.1 shows the CRDPs in the form of a matrix, where W is the learned convex concept inter-layer weight (CCILW) between node j at new layer (i.e., Layer-1 in Figure 3.3) and node i at input layer (i.e., Layer-0). The set of ILWs, from one node j at new layer to all input nodes i, are the values of CRDP of j^{th} cluster center (i.e., C_j) identified in the CCI process. For instance, cluster center C_1 (see Figure 3.2) forms the scaler weight vector [$W_{1,1}$, $W_{1,2}$] (CCILWs

[1]Calculating DoC of a node is explained in detail with *upward activation propagation* operation.

showed by two yellow lines in Step 3 Figure 3.3) between the node N_1 at Layer-1 and both input nodes S_1 and S_2 at Layer-0.

$$W = \begin{bmatrix} W_{1,1} & W_{1,2} & \dots & W_{1,i} \\ & \dots & & \\ W_{k,1} & W_{k,2} & \dots & W_{k,i} \\ & \dots & & \\ W_{j,1} & W_{j,2} & \dots & W_{j,i} \end{bmatrix} = \begin{bmatrix} C_1 \\ \dots \\ C_k \\ \dots \\ C_j \end{bmatrix} \tag{3.1}$$

Weight matrix W for the sake of easy notation input nodes are represented as i, and abstract concept layer nodes are represented as j. Where i=$1, 2, ..., n_a$ and j=$1, 2, ..., n_A$.

The distance between the learned weight vector of one node j (at Layer-1) and activation of all input nodes S_1 and S_2 (at Layer-0), is used to determine how strongly the input vector represents the node N_j at the new layer. Thus, enable us to identify the convex abstract concepts for the input instance (elaborated in Section 3.3.4).

3.3.4 Step 4: Convex Abstract Concept Upward Activation Propagation (CACUAP) Process

The upward activation propagation is a geometric reasoning operation, i.e., a non-linear projection of an i-dimensional input data vector a_i, into a j-dimensional output vector A_j (see Step 4 in Figure 3.3). The CACUAP operation is carried out in two stages, in the first stage the geometric distance operation takes place, and in the second stage, geometric distance is translated to establish a similarity relation.

Geometric Distance Function (GDF)- Stage 1 In the first phase of the CACUAP mechanism we determine the geometrical distance between the learned weight vectors (see Equation 3.1) and an input instance a_i. The numerator of Equation 3.2 shows a function to calculate the Euclidean distance between the j^{th} weight row vector and input vector a. The denominator of Equation 3.2 shows the relation that normalizes[2] the calculated distance between [0, 1].

$$d_j = \frac{\sqrt{\sum_{i=1}^{n_a}(W_{j,i} - a_i)^2}}{\sqrt{n_a}} \tag{3.2}$$

And consequently, j normalized Euclidean distances d_j are obtained between all j weight vectors and input instance a_i.

[2] In RANs modeling the activation values are, by definition, real values in the [0, 1] interval – in an n-dimensional space the maximal possible Euclidean distance between any two points is $\sqrt{\sum_{i=1}^{n}(a_i - 0)^2}=\sqrt{n}$, where a_i=1.

Similarity Translation Function (STF)- Stage 2 In the second phase the calculated normalized distance is transformed to obtain a similarity relation such that the following requirements are fulfilled:

- $f(d = 0) = 1$, i.e. when distance is 0 similarity is 100%.
- $f(d = 1) = 0$ i.e. when distance is 1 similarity is 0%.
- $f(d = x)$ is continuous, monotonous, and differentiable in the $[0, 1]$ interval.

$$f(x) = (1 - \sqrt[3]{x})^2 \tag{3.3}$$

In RANs modeling Equation 3.3 is used as Similarity Translation Function to determine the similarity relation of the previously calculated distance. The non-linearity of STF is depicted in Figure 3.4, indicating that the similarity value reduces drastically when the normalized Euclidean distance is larger than 0.05 (or 5% dissimilar).

Algorithm 1: Convex Abstract Concept Upwards Activation Propagation algorithm

Input: Vector $[a_1, a_2, ..., a_{n_a}]$ as input at layer l
Output: New activation vector $[A_1, A_2, ..., A_{n_A}]$ in layer $l + 1$
foreach *Node* A_j *in layer* $l + 1$ **do**
  Calculate Normalized Euclidean Distance:
    $d_j = \frac{\sqrt{\sum_{i=1}^{n_a}(W_{j,i} - a_i)^2}}{\sqrt{n_a}}$
  Transform d_j through STF Equation 3.3:
    $A_j = f(d_j^2)$

Where:
i= $[1, 2, ..., n_a]$.
j= $[1, 2, ..., n_A]$.
$W_{j,i}$ is ILW see Equation 3.1.

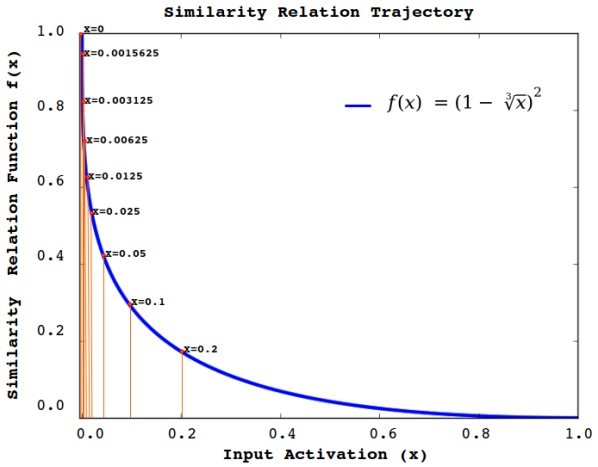


FIGURE 3.4: Plot of Similarity Translation Function with respect to varying input values in range $[0, 1]$

The first three steps generate the RAN's model (see Figure 3.3), later, in the fourth step, this model is used via CACUAP operation by propagating the input activation (a_i) upward and obtaining activation (A_j) at convex abstract concept layer (inspired by the theory of spreading activation see Appendix A.3). Algorithm 1 describes the convex abstract concept upward activation propagation operation, showing how the inputs and interlayer learning weights W are used to calculate similarity relation to generating output activation at each abstract concept representative nodes.

3.3.5 RAN's Performance with Toy-data1

The metrics Precision, Recall, F1-Score, and Accuracy, are used to evaluate the performance of RAN's model with Toy-data. To carry out the evaluation operation *True-labels*, and *Test-labels* are determined via abstract concept labeling (ACL) operation of RANs (see Section B.1 for ACL's description in Appendix B). Further, these labels were used to form a multi-class confusion matrix for the five classes of Toy-data, upon which, the four evaluation metrics were calculated.

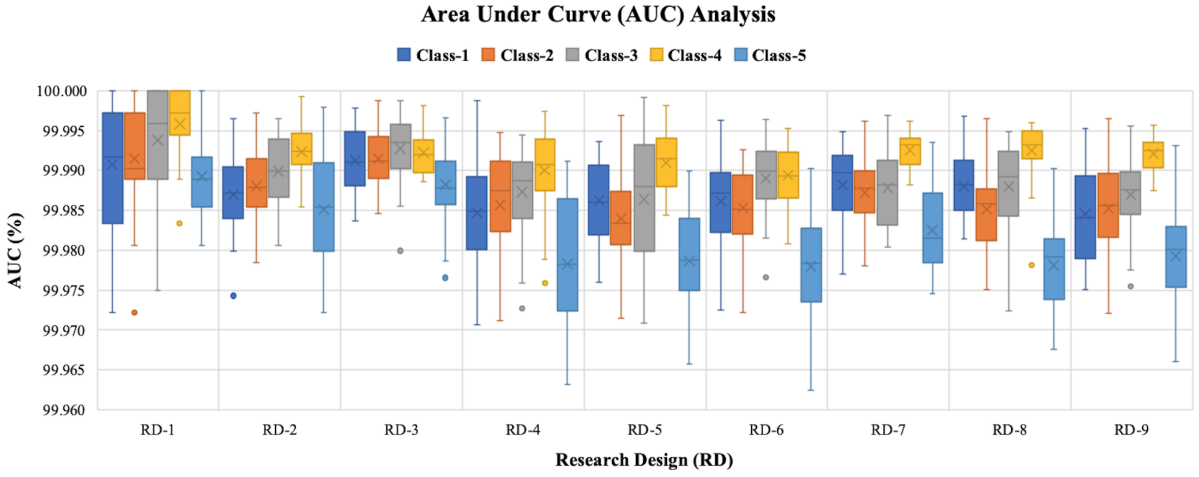


FIGURE 3.5: Area Under Curve for five classes of Toy-data for nine Research Designs (RD) of varying Test and Train data sizes

Receiver Operating Characteristics (ROC) curves for five classes were plotted to determine the operating characteristic of the RAN's model with Toy-data. The input True-labels were converted into five vectors of binary labels. Each vector has '1' for the label that maps to an abstract node the vector represents, and '0' otherwise (see Section B.3 in Appendix B for the node-wise binary transformation of input True-label). Further confidence-score (see Section B.3 in Appendix B for node-wise confidence-score calculation) is calculated for each abstract node. These binary-labels, confidence-score are used to plot ROC curves for the five classes and determine the Area Under Curve (AUC) for each class as well. Figure 3.5 shows AUC retrieved for all five classes w.r.t. different Research Designs (Refer Appendix B.4 for the definition of Research

Table 3.3: RAN's Comparative Study for Toy-data

Model	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
RBM+	90.87 ± 1.26	85.25 ± 2.61	82.34 ± 3.85	85.25 ± 2.61
K-NN	99.96 ± 0.08	99.95 ± 0.11	99.94 ± 0.12	99.95 ± 0.11
LR	99.65 ± 0.07	99.64 ± 0.07	99.64 ± 0.07	99.64 ± 0.07
MLP	95.62 ± 11.18	96.82 ± 7.56	96.02 ± 9.95	96.82 ± 7.56
RANs	99.12 ± 0.09	99.12 ± 0.09	99.12 ± 0.09	99.12 ± 0.09
SGD	96.00 ± 2.81	95.25 ± 2.86	94.57 ± 3.76	95.25 ± 2.86

Design). Table 3.3 shows the RAN's comparison with five different types of methodologies based upon their classification capabilities. It is observed that RANs performed satisfactorily.

According to the Table 3.3 the K-NN, LR and RANs performed equally well, although, the performance of the K-NN algorithm was the best. The model generated with RBM+ displayed the lowest performance. The models obtained from RBM+, SGD and MLP not only performed lower than the other three techniques but also displayed higher variance. Moreover, the MLP showed the highest variance but performed better than RBM+. The results of classification also proves that the hypothesis the RANs experiment with Toy-data1 which symbolically identifies the 5 nodes in Layer-1 as convex abstract concepts representative of 5 classes in the Toy-data1.

In RAN's algorithm the 4 operations has different complexities: (1) the concept identification process is expressed as *O(f(n))* where *f(n)* is the complexity of the concept identifier (or clustering algorithm); (1) the concept creation has complexity of *O(k)* where k is the number of clusters; (3) the inter layer learning also has complexity of *O(k)* because it is an assignment operation and is equal to number of identified cluster centers; (4) the upward activation operation has the complexity of *O(n)* when n is the number of data instances. The overall complexity of the RANs modeling for creating a single layer is expressed by equation 3.4.

$$T(n) = O(max\{O(f(n)), O(n)\}) \tag{3.4}$$

$$f(n) = O(n^{(k+2/p)}) \tag{3.5}$$

where: k is number of clusters; p is number of features

The time complexity of the K-means algorithm is given by equation 3.5 and when K-means is chosen to be the concept identifier is, therefore, the $T(n)=O(n^{(k+2/p)})$. Table 3.4 lists the time complexities of all the algorithms used in this article including the RANs time complexities with both K-means and Affinity Propagation algorithms. In Table 3.4 the complexities of K-means and Affinity Propagation algorithms, in fact, are the complexities of the RANs modeling because their complexities are greater than *O(n)*.

TABLE 3.4: Time Complexities of Models used in the Article

Algorithm	Time Complexity	Description	Source
K-means	$O(n^{k+2/p})$	n: n_samples; k: n_clusters; p:n_features	Pedregosa et al. (2011)
Affinity Propagation	$O(n^2)$	n: n_samples	Pedregosa et al. (2011)
MLP	$O(n \cdot m \cdot h^k \cdot o \cdot i)$	n: n_samples; m: features; k: no. of hidden layers; h: number of hidden neuron o: output neuron; i: no. of iterations	Pedregosa et al. (2011)
RBM	$O(d^2)$	d: max(n_components, n_features)	Pedregosa et al. (2011)
KNN	$O(m \cdot n \cdot i)$	m: n_components; n: n_samples; i: min(m, n)	Pedregosa et al. (2011)
LR	$O(n \cdot m^2)$	n: n_samples; m: n_features	Pedregosa et al. (2011)
SGD	$O(k \cdot n \cdot \bar{p})$	n: n_samples; k: n_iterations; $\bar{p}$: the average number of non-zero attributes per sample	Pedregosa et al. (2011)

3.4 Behavioral Demonstration of RANs

This section exhibits two distinct aspects of RANs modeling via experiments with two datasets: IRIS and UCI's Human Activity Recognition data. Both investigations present a different view of RAN's methodology, highlighting the capabilities of the RAN's approach.

3.4.1 Experiment with IRIS dataset

There are two objectives of this probe, first is to demonstrate flexibility in choosing an appropriate methodology for concept identification operation in RANs modeling (see Section 3.3.1). Second is to show how RANs modeling can be used to build a deep hierarchy of convex abstract concepts dynamically. This experiment uses Affinity Propagation (Frey and Dueck, 2007) clustering algorithm as a concept identifier to support the claim of independence in selecting a suitable clustering method for CI process in RANs modeling. Unlike the K-means algorithm (used to describe the RANs methodology in Section 3.3), with the Affinity Propagation algorithm, the number of clusters within the data need not to be known beforehand. Furthermore, Affinity Propagation conforms to the basic requirements (see Section 3.3.1) for being a concept identifier in RANs modeling.

The second prospect of this experiment is to illustrate the dynamic topology of RAN's approach where the network grows to form several layers representing convex abstract concepts. For this demonstration, an algorithm is developed, named concept hierarchy creation (CHC) algorithm (see Algorithm 2). The CHC algorithm streamlines all four steps of RANs modeling (i.e., CCCI, CACC, CCILW and CACUAP) and uses these steps iteratively to build a hierarchy of convex abstract concepts as described through Algorithm 2. This experiment was conducted using the IRIS dataset obtained from the UCI machine learning repository (Lichman, 2013).

IRIS dataset consists of 150 instances, each having four dimensions/attributes corresponding to features of a flower: petal width, petal length, sepal width, sepal length. The original data set values are normalized to fall within the interval of $[0, 1]$. The dataset represents three

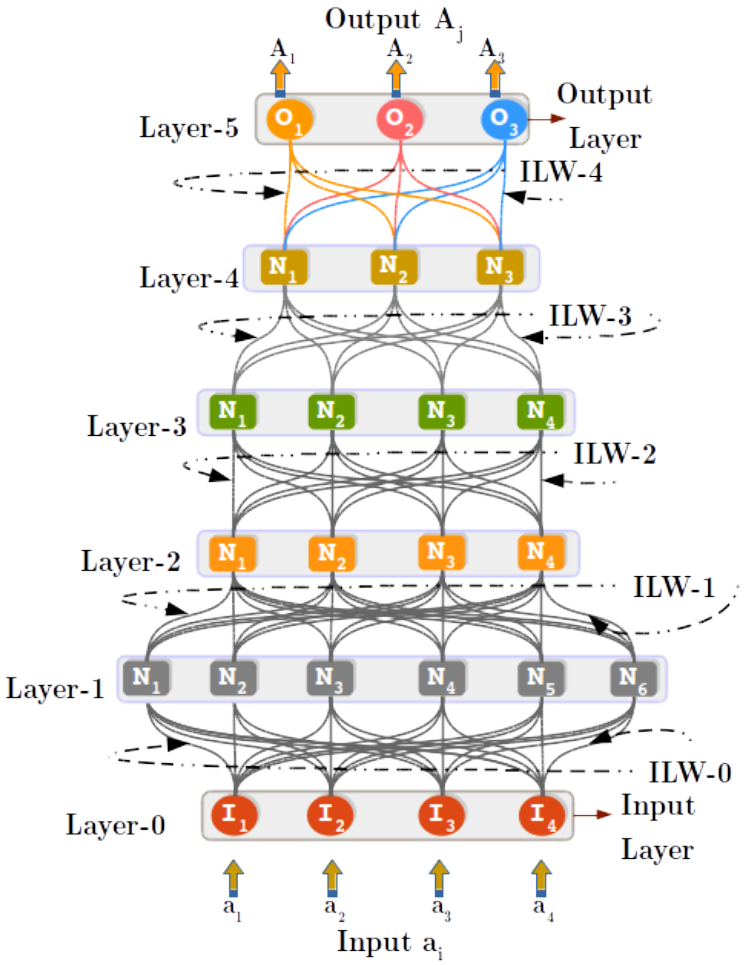


FIGURE 3.6: Model generated with 90% stratified IRIS data using concept hierarchy creation Algorithm. Layer-0 is created while initializing the CHC algorithm. The algorithm grew to a *Desired-depth* of six Layers (including input Layer-0), and in each iteration of CHC algorithm a new layer is created dynamically and the Interlayer weights (CCILW) are learned between the existing layer and a newly created layer above it.

categories of flowers Iris-Setosa, Iris-versicolar, and Iris-verginica. In our concept hierarchy creation algorithm, we initialize the Affinity Propagation clustering algorithm with the following parameters: (1) damping_factor (DF) = 0.94 for layers below level 3, DF = 0.9679 for the layers at level 3 and above; (2) convergence_iteration=15; (3) max_iteration=1000.

Input layer-0 was created, with four nodes (equal to the dimension of IRIS data), and the RAN's hierarchy generation is carried out according to Algorithm 2. The model obtained from CHC process is depicted in Figure 3.6, the model was initialized to grow six layers deep. Therefore, hierarchy augmentation terminates at Layer-5, with Layer-5 identified as most abstract layer consisting three nodes acting as abstract representatives of three categories of flowers of IRIS dataset. To evaluate the obtained RAN's model, *True-labels*, and *Test-labels* were retrieved using an abstract concept labeling procedure (see Section B.1 in Appendix B). A confusion matrix (see Figure 3.7) was generated using the True and Test labels. With the aid of the confusion matrix, Precision, Recall, F1-Score and Accuracy were calculated to evaluate the

model. The model performed quite decently with an observed accuracy of 93% (ca.), the results of precision, recall and F1-Score are reported in Table 3.5.

Algorithm 2: Concept Hierarchy Creation Algorithm

Input: Input Multi-variate data with values between [0,1]
Output: OutputSet of layers of Concepts – concept hierarchy
Initialization: Create input layer layer-0 having dimension equal to that of input data
Set *Current-layer-size* $CLS = i$, a dimension of *input-data* vector
Set *Layer-count* $L = 0$
Set *Desired-depth*= 6
Select Clustering algorithm and initialize
Set *current-data* = *input-data*
repeat
  Run *clustering algorithm* on *current-data* to identify set of cluster centers C
  Create a *new-layer* above *current-layer*, with no nodes
  foreach *cluster center* $C_j \in C$ **do**
    Create new node j in *new layer* $l+1$
    foreach *node* i *in* current-layer **do**
      Create a new weighted connection $W_{c_j,i}$ between c_j and i such that $W_{c_j,i}$ is the coordinate of c along the i dimension
  Set *new-data* = empty data set
  foreach *each* datum *in* current-data **do**
    Inject *datum* in *current-layer*
    Propagate activation from *current-layer* to *new-layer* using algorithm 1
    Add activation pattern produced in *new-layer* to *new-data*
  Set $L = L + 1$
  Set CLS = number of clusters in *current-layer*
  Set *current-data* = *new-data*
  Set *current-layer* = *new-layer*
until CLS=1 ***OR*** Desired-depth= L;

		Predicted Labels			
		Class-0	Class-1	Class-2	
True Label	Class-0	100%	0%	0%	5
	Class-1	0%	100%	0%	5
	Class-2	0%	20%	80%	5
		5	6	4	15

Figure 3.7: Confusion Matrix generated to validate RAN's model with IRIS data (having 9 : 1 *train, and test* data ratio) for Class-0 (Setosa), Class-1 (Verisicolour), and Class-2 (Virginica).

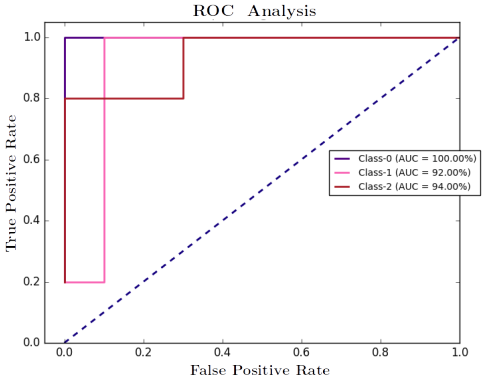


Figure 3.8: ROC curve analysis with IRIS dataset (having 9 : 1 *train, and test* data ratio), for Class-0 (Setosa), Class-1 (Verisicolour), and Class-2 (Virginica)

The ROC curve analysis of the RAN's model, as shown in Figure 3.8, displays the various operating characteristic and the observed Area Under Curve for all the classes of IRIS data. In

TABLE 3.5: Evaluation of RANs Model generated through IRIS data

Class	Precision (%)	Recall (%)	F-1 Score (%)	Support
Setosa	100.0	100.0	100.0	5
Versicolour	83.33	100.0	90.91	5
Virginica	100.0	80 .00	88.89	5
Avg/Total	94.44	93.33	93.26	15

this experiment, it is worth mentioning the application of RANs modeling for data dimension transformation and data visualization. In Figure 3.6 we can observe that the dimension of Layer-0 is four, whereas the size of the other layers either expands or reduces when the network grows. This dimension transformation operation is helpful in addressing the issue of the curse of dimensionality. Besides, the transformed data can be plotted to extract useful information from the data (See Appendix E for details of this application of RANs).

3.4.2 Experiment with Human Activity Recognition Data

This experiment aims to show the ability of the RAN's approach to building the representation of generic concepts, such as identifying abstract form made up of concrete features, for instance concrete concepts food ingredients blends to produce an abstract concept a recipe. The experiment uses UCIHAR (Anguita et al., 2013) dataset for home activity recognition using the smartphone, obtained from the UCI machine learning repository. The data captured six activities Walking, Walking_upstairs, Walking_downstairs, Sitting, Standing, and Laying. With this demonstration, we show how the RAN's approach can be used to model two abstract concepts, i.e., characterizing Mobile and Immobile behavior from the six activities as mentioned earlier.

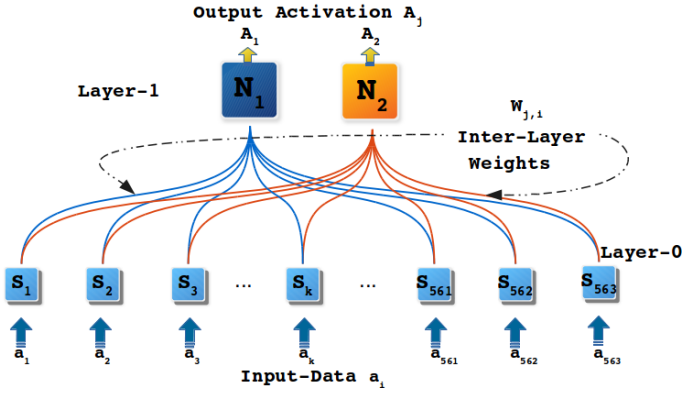


FIGURE 3.9: Model generated with the RAN's approach. Nodes N_1 and N_1 at Layer-1 represent either of the two abstract concepts, i.e. *Moving* and *Stationary*. Each node at Layer-0 represents individual dimensions of the input data vector

The True-label and Test-label obtained through ACL operation were used to form the confusion matrix, which is later referred to calculate Precision, Recall, F1-Score, and Accuracy for evaluating the generated model. Node-wise binary labels and confidence scores were determined (as

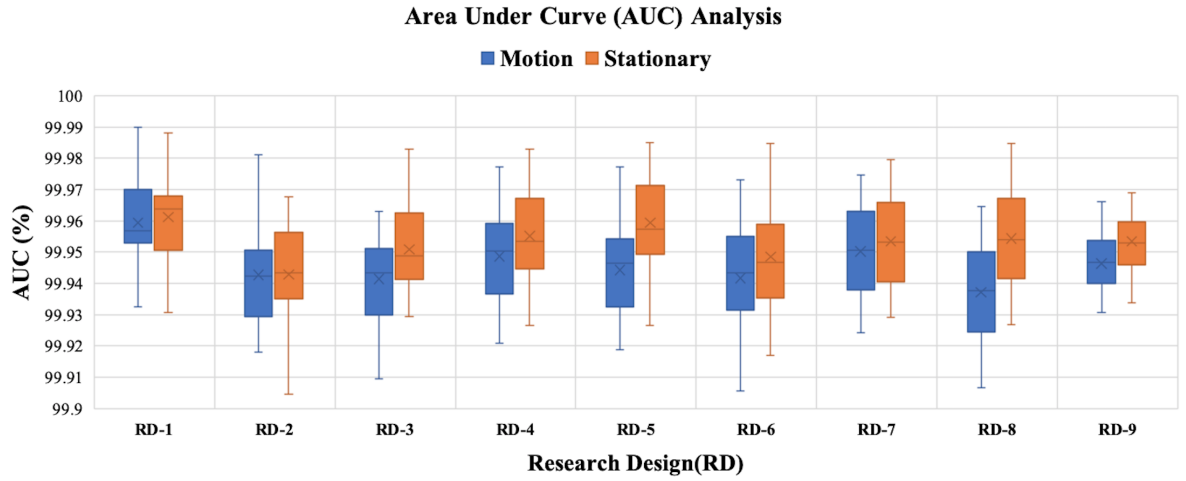


FIGURE 3.10: Area Under Curve observed during ROC curve analysis of UCIHAR data in order to determine operational points of two abstract concepts (i.e. *Motion* and *Stationary*) for all nine Research Designs (RD)

TABLE 3.6: RAN's Comparative Study for UCIHAR dataset

Model	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
RBM	99.68 ± 0.14	99.68 ± 0.14	99.68 ± 0.14	99.68 ± 0.14
K-NN	99.96 ± 0.02	99.96 ± 0.02	99.96 ± 0.02	99.96 ± 0.02
LR	99.97 ± 0.02	99.97 ± 0.02	99.97 ± 0.02	99.97 ± 0.02
MLP	99.96 ± 0.02	99.96 ± 0.02	99.96 ± 0.02	99.96 ± 0.02
RANs	99.85 ± 0.01	99.85 ± 0.01	99.85 ± 0.01	99.85 ± 0.01
SGD	99.98 ± 0.01	99.98 ± 0.01	99.98 ± 0.01	99.98 ± 0.01

described in Section B.3) for both abstract nodes at Layer-1. Figure 3.10 shows the Area Under Curve (AUC) observed during the ROC curve analysis of all 10-Folds in different Research Designs. With both of these evaluations it is deduced that, apart from building the representation of abstract concepts, the model generated with RANs performed satisfactorily.

The UCIHAR dataset consists of 563 attributes, where each attribute has data-values from embedded sensors of a smartphone. Before initiating the modeling procedure, the dataset set was preprocessed, such that, all data values are normalized in the range [0, 1] (refer Section 3.1 for normalization procedure). Further, the modeling process is carried out with the CHC algorithm (see Algorithm 2). The processed data assigned as input-data, further an input-layer was created with 563 nodes (equal to the dimension of input-data). In this investigation, we hypothesized that there are two abstract concepts related to motion and addresses them as *Mobile* and *Immobile* behaviors, that pertain to the subject's movement-related activities. Consequently, the K-means algorithm is chosen for aoncept identification and configured to identify two clusters. The *Desired-depth* was also set to '1' such that modeling terminates after growing one layer above the input layer (Layer-0). Having fulfilled the initialization part of the CHC algorithm modeling is performed, generating a two-layered model as depicted in Figure 3.9. In Figure 3.9 Layer-0 shows *input-layer* and Layer-1 corresponds to *abstract concept layer* where both nodes

(N_1, and N_2) represents each of the two abstract concepts (i.e. *Motion* and *Stationary* abstract concepts).

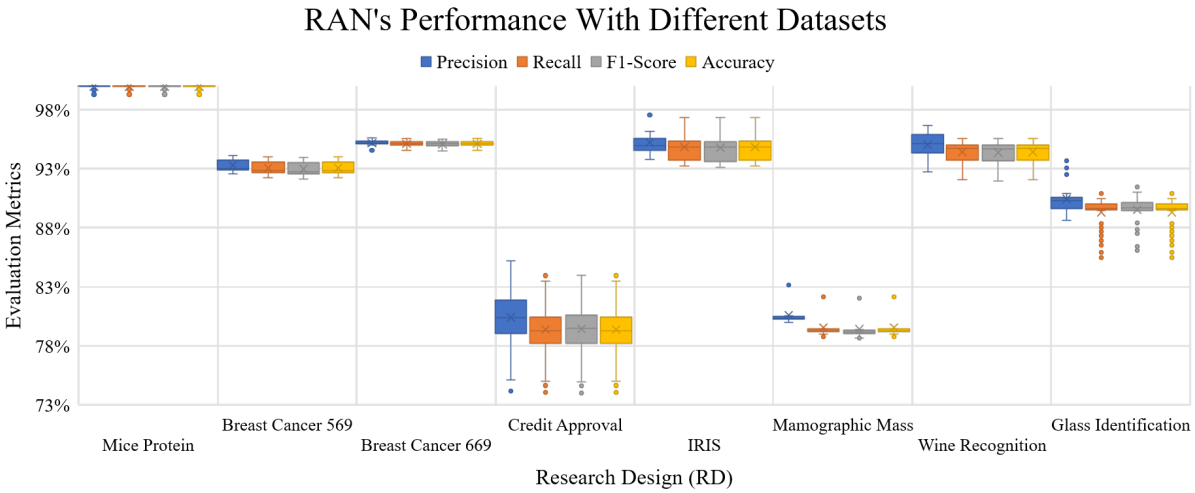
Among captured six activities (Walking, Walking_upstairs, Walking_downstairs, Sitting, Standing and Laying), Walking, Walking_upstairs, Walking_downstairs are the actions of motion, whereas the three remaining ones represent static states. Based upon these two facts, we expect that one of the abstract nodes in Layer-1 conjointly represents Walking, Walking_upstairs and Walking_downstairs as one class. The other node in Layer-1 takes the other three categories (i.e., Sitting, Standing and Laying) together. Upon performing the labeling of nodes at Layer-1 through ACL procedure (see Section B.1 for ACL process elaboration), it was observed that Walking, Walking_upstairs, Walking_downstairs classes were mapped to one node of Layer-1. Whereas, the labels Sitting, Standing and Laying was traced to the other node in Layer-1. Interestingly, this outcome commensurates with the expectation of this experiment and shows the RAN's capability to identify abstract concepts in an unsupervised manner naturally.

The RANs modeling was compared with five different approaches based upon their classification operation. To carry out the comparative study it was essential to transform the six Labels into binary Labels. Therefore, the Labels of the dataset were merged to form two groups, i.e., Walking, Walking_upstairs, Walking_downstairs in Class-1, and Sitting, Standing, and Laying in Class-2. Later, the modeling was performed followed by validation and evaluation. Table 3.6 displays the comparison of all five approaches with RANs modeling. It is observed that RANs approach is competent to five techniques with one advantage that it is an unsupervised approach, moreover it demonstrate the ability to build representations of abstract concepts.

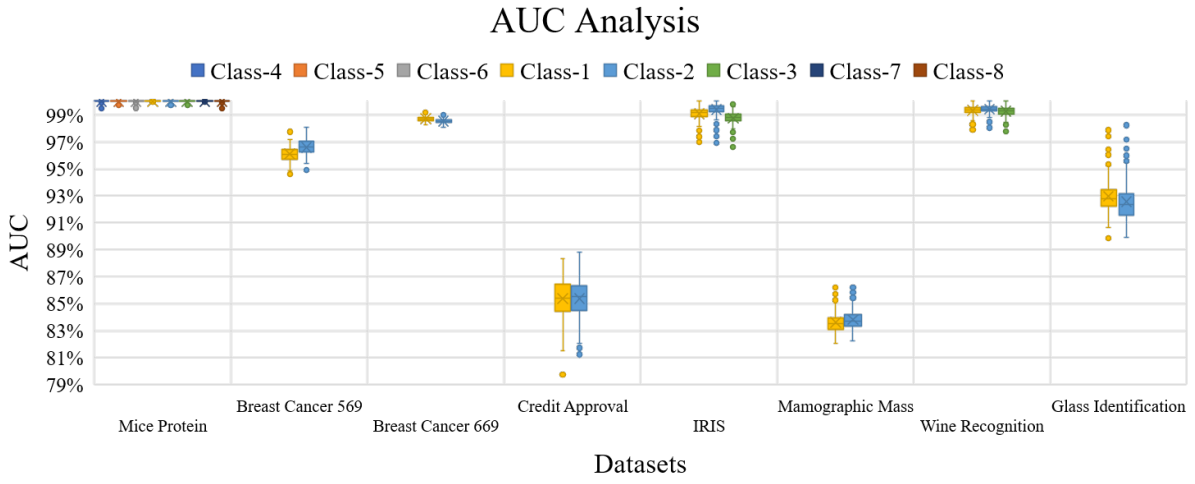
3.5 RANs Applicability

This section highlights the scope of RANs modeling as a classifier w.r.t. distinct domains. To support it's usability, experimental results are reported using eight datasets concerning different areas. A comparative study was also carried out using these datasets to match RANs classification ability with five different classifiers.

Among the eight datasets, the *Mice Protein* (Higuera et al., 2015), *Mammographic Mass* (Elter et al., 2007), *Breast Cancer 569* and *669* (Street et al., 1993, Bennett and Mangasarian, 1992) data pertain to the medical field, *Glass Identification* (Evett and Spiehler, 1987) data representing forensic science, *Credit Approval* (Quinlan, 1999) represents economic data, *Iris* (Fisher, 1936) is a botanical data, and Wine Recognition (Forina et al., 1988) is a data for chemical composition analysis. The experiments performed with these datasets were akin to the investigations done with Toy-data1 (in Section 3.3), and UCIHAR data (in Section 3.4.2), i.e., K-means algorithm used as concept identifier, where K is the number of class labels of each dataset, the hierarchy is set to have a depth of two layers (one Input and one abstract concept layer). For



(A) RAN's performance with eight different datasets depicting RAN's appositeness with data belonging to distinct domains



(B) Observed Area Under Curve (AUC) while performing ROC curve analysis for RAN's model generated with eight different datasets. The graph shows the plot of percentage AUC for classes 1 to 8. For each dataset class labels of the graph is serially mapped as: *Mice protein* {*c-CS-s [Class-1], c-CS-m [Class-2], c-SC-s [Class-3], c-SC-m [Class-4], t-CS-s [Class-5], t-CS-m [Class-6], t-SC-s [Class-7] and t-SC-m [Class-8]*}; *Mammographic Mass* {*Benign [Class-1] and Malignant [Class-2]*}; *Credit Approval* {*Postitive [Class-1] and Negative [Class-2]*}; *IRIS)* {*Setosa [Class-1], Versicolar [Class-2] and Verginica [Class-3]*}; *Breast Cancer 569* {*Benign [Class-1] and Malignant [Class-2]*}; *Breast Cancer 669* {*Benign [Class-1] and Malignant [Class-2]*}, *Wine Recognition* {*Class-1, Class-2 and Class-3*}*Glass Identification* {*Window Glass [Class-1] and Non-Window Glass [Class-2]*}

FIGURE 3.11: RAN's performance with eight datasets using Precision, Recall, F1-Score and Accuracy along with ROC-AUC analysis.

every dataset, models were generated using thirty iterations in nine Research Designs (RD) (refer the Table B.1 in Section B.6). In every RD 10-Fold cross-validation was applied to determine the performance of the models. An aggregate of Precision, Recall, F1-Score, and Accuracy of all folds in all RDs was calculated for all the datasets, as shown in Figure 3.11. From the Figure 3.11a it can observed that with *Mice Protein* data RAN's scores 99.99%(ca.) for all evaluation metric, whereas for *Iris, Glass Identification, Breast Cancer, and Wine Recognitions* the observations were convincing, i.e., above 89.00% (ca.). In all the folds of nine RD, ROC curves were also plotted for each class label of the eight datasets, the mean AUC for each class of the datasets is shown in Figure 3.11b. The evaluation through Precision, Recall, F1-Score, Accuracy and ROC-AUC analysis (see Figure 3.11a and 3.11b respectively) displays the RANs capability in machine learning tasks with different kind of datasets.

The same procedure was applied to obtain average Precision, Recall, F1-Score and Accuracy for all the datasets with five other classifiers (i.e. *RBM+, KNN, LR, MLP, and SGD*). Table 3.7 shows the overall comparison (the bold fonts depicts best result and italic font shows the worst outcome). It is worth noting that being dynamic and unsupervised RANs modeling performed quite satisfactorily especially with Mice Protein data, where it outperformed SGD and RBM+, was found competent with LR, KNN and MLP classifiers.

The Table 3.7 not only compares the RAN's performance with five classifiers but also provide insight into the behavior of different models w.r.t. to different types of datasets. With Mice Protein dataset we can observe that, besides RBM+, all the methodologies performed well. One more thing can be noted that only RANs displayed a low variance whereas other algorithms observed to have relatively higher variance. In all the experiments, the variance of RANs was observed to be minimal with the highest observed *standard deviation* of *1.01%* in the experiment with Wine Recognition dataset. Experiments with the LR algorithm also the observed *standard deviation* less than *2%* in all the experiments (except the experiment with IRIS data).

The experiments with SGD also showed a consistent and minimal variation with the highest *standard deviation* of *6.4%*. Besides LR, SGD and RANs, MLP algorithm also displayed a uniform performance with the worst observed *standard deviation* of *3.34%* for F1-Score of Mice Protein dataset. The algorithm KNN performed well in all the experiments but it is worth nothing that it also displayed *standard deviation* more than *10%* in many experiments. The worst variation of KNN was observed with IRIS dataset showing *standard deviations* more than *21%*. In all the experiments the RBM+ algorithm good performance with Breast Cancer 569 and 669 datasets, but in all the experiments it displayed a very high variation with a maximum *standard deviation* of *44.07%* observed with Precision of Mice Protein dataset.

Figure 3.12 shows four graphs depicting RANs performance with different benchmark data sets. These graphs display an important aspect of RANs modeling and it's performance behavior when evaluated to different research design. The Precision, Recall, F1-Score, and Accuracy trajectories of Human Activity Recognition (HAR), Breast Cancer 669 (BC1), Toy-data (TD)

and Mice Protein (MP) Data is almost straight. The evaluation plots of Glass Identification (GI), Wine Recognition (WR), Mammographic Mass (MM), Breast cancer 569 (BC2) and Mice Protein (MP) datasets shows a minimal decline in observations w.r.t RD-1 and RD-9 Research Design. On the contrary, results from IRIS Data (ID) and Credit Approval (CA) dataset depicted a higher value while comparing the evaluation of RD-1 with RD-9 Research Designs of these data sets. Principally, the results of all four metrics of evaluation obtained similar results (with marginal variation) irrespective of the Test and Train data ratio. This is a notable observation because it shows that RANs approach obtains a satisfactory result even when trained with a small amount of data.

3.6 Conclusions

To comprehend and reasoning for emotions, ideas etc., it is evident to understand abstract concepts because they are perceived differently w.r.t concrete concepts. There have been notable efforts to study concrete concepts (features like walking or ingredients), but progress in investigating abstract concepts (generic features such as is-moving or recipe) is relatively less. This chapter proposes an unsupervised computational modeling approach, named Regulated Activation Network (RAN), that has an evolving topology and learns a representation of abstract concepts. The RANs methodology was exemplified through a 2-D artificial data, yielding a satisfactory performance evaluation of 99.12% (ca.) for Precision, Recall, F1-Score and Accuracy metrics, along with an average AUC of 99.9% (ca.) for all classes in the dataset. The RAN's comparisons with five classifiers (KNN, LR, MLP, RBM+ and SGD) displayed that RANs was equally efficient. The experiment with Toy-data1 we simulated the cognitive process of *concept creation*, *learning* and *activation propagation* and proved the hypothesis of the experiment using a classification operation.

An experiment with IRIS data showed the characteristic of RANs deep hierarchy generation and independence in choosing concept identifier. With the aid of concept hierarchy creation algorithm (proposed in Section 3.4.1), evolving nature of RANs modeling is shown using Affinity Propagation clustering algorithm (as an alternate concept identifier instead of the K-means algorithm as used in modeling with Toy-data problem). With the generated model it was shown that the model dynamically grew to a depth of six layers and performed with Precision of 94.44% (ca.), Recall of 93.33% (ca.), F1-Score of 93.26% (ca.) and Accuracy of 93.33% (ca.), along with an observed AUC of 100% (ca.), 92.00% (ca.) and 94.00% (ca.) for the three classes of data.

Modeling with UCI's Home Activity Recognition (UCIHAR) dataset exhibited the RAN's behavior of natural identification of generic concepts. The experiment demonstrated that how six labels (the activity of Walking, Walking_upstairs, Walking_downstairs, Sitting, Standing and Laying) of the dataset are automatically identified as Mobile (Walking, Walking_upstairs

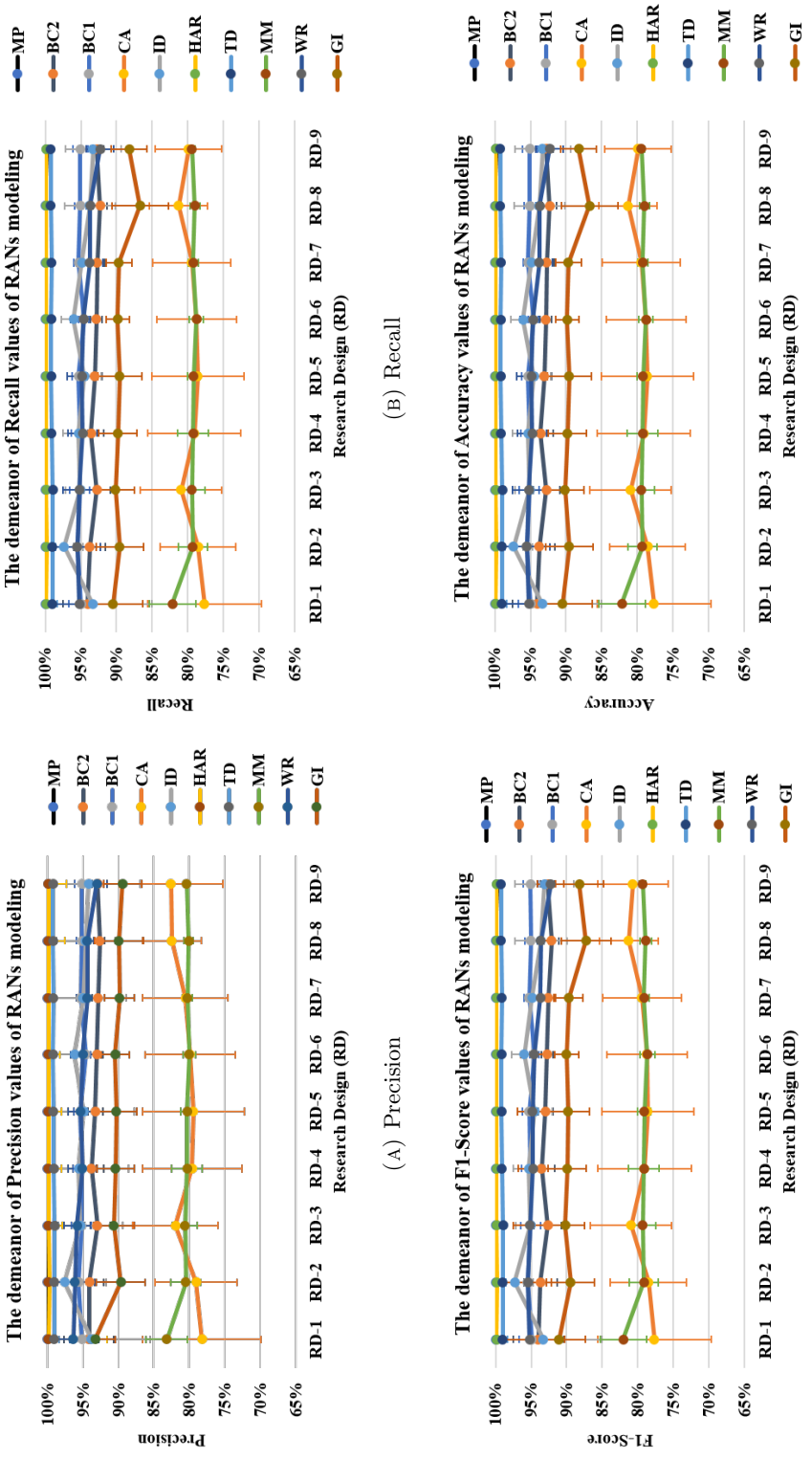
and Walking_downstairs) and Immobile (Sitting, Standing and Laying) abstract concepts. The evaluation of the model shown a performance of 99.85% (ca.) for all four metrics and AUC of 99.9% (ca.) for both abstract concepts. RANs performed equivalently when compared to all five classifiers KNN, LR, MLP, RBM+, and SGD.

The proof of concept of RANs modeling as a classifier was also provided with eight UCI benchmarks. It was identified that RANs approach performed satisfactorily displaying the best outcome of 98.9% (ca.) with *Mice Protein* dataset (for all metrics). The comparison of RANs modeling with five classifiers substantiated the effectiveness of the proposed methodology. It was also observed that the RANs performance remained similar irrespective of the size of train data. During the simulations, a non-convexity was observed in several datasets. Chapter 4 describes the second RANs modeling that can capture the non-convexity in the data.

TABLE 3.7: RANs comparison with eight datasets belonging to different domains

Data	Algo	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)	Data	Algo	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
Mice Protein	*RBM+*	*43.45 ± 44.07*	*53.50 ± 38.23*	*45.46 ± 43.36*	*53.50 ± 38.23*	**Breast Cancer 569**	*RBM+*	93.60 ± 2.69	93.51 ± 2.77	93.46 ± 2.86	93.51 ± 2.77
	KNN	98.63 ± 3.97	98.34 ± 4.84	98.07 ± 5.65	98.34 ± 4.84		*KNN*	99.80 ± 0.59	99.79 ± 0.62	99.78 ± 0.63	99.79 ± 0.62
	LR	98.99 ± 1.94	98.28 ± 3.38	98.14 ± 3.71	98.28 ± 3.38		*LR*	**99.89 ± 0.07**	**99.89 ± 0.07**	**99.89 ± 0.07**	**99.89 ± 0.07**
	MLP	98.54 ± 2.19	98.23 ± 2.71	97.83 ± 3.34	98.23 ± 2.71		*MLP*	98.67 ± 0.94	98.65 ± 0.96	98.64 ± 0.96	99.89 ± 0.07
	RAN	**99.98 ± 0.06**	**99.97 ± 0.06**	**99.89 ± 0.06**	**99.97 ± 0.06**		*RAN*	*93.17 ± 0.36*	*92.97 ± 0.36*	*92.87 ± 0.42*	*92.97 ± 0.36*
	SGD	99.11 ± 1.84	98.84 ± 2.46	98.68 ± 2.81	98.84 ± 2.46		*SGD*	99.87 ± 0.13	99.85 ± 0.18	99.83 ± 0.20	99.85 ± 0.18
Breast Cancer 669	*RBM+*	95.72 ± 3.62	95.34 ± 4.60	95.13 ± 5.16	95.34 ± 4.60	**Credit Approval**	*RBM+*	*76.44 ± 12.50*	*75.63 ± 12.98*	*74.04 ± 14.59*	*75.63 ± 12.98*
	KNN	99.46 ± 0.88	99.44 ± 0.93	99.43 ± 0.94	99.44 ± 0.93		*KNN*	95.48 ± 0.16	95.46 ± 0.17	95.46 ± 0.17	95.46 ± 0.17
	LR	99.16 ± 0.17	99.14 ± 0.17	99.15 ± 0.17	99.14 ± 0.17		*LR*	95.06 ± 0.38	95.04 ± 0.39	95.04 ± 0.39	95.04 ± 0.39
	MLP	98.96 ± 0.76	98.95 ± 0.76	98.95 ± 0.77	98.95 ± 0.76		*MLP*	98.02 ± 1.32	98.00 ± 1.34	97.99 ± 1.34	98.00 ± 1.34
	RAN	*95.18 ± 0.25*	*95.15 ± 0.24*	*95.11 ± 0.25*	*95.15 ± 0.24*		*RAN*	80.67 ± 1.37	79.58 ± 1.05	79.66 ± 1.13	79.58 ± 1.05
	SGD	**99.88 ± 0.16**	**99.88 ± 0.16**	**99.18 ± 0.16**	**99.88 ± 0.16**		*SGD*	**99.77 ± 0.39**	**99.75 ± 0.40**	**99.75 ± 0.40**	**99.75 ± 0.40**
Glass Identification	*RBM+*	*82.58 ± 10.29*	*84.19 ± 4.90*	*80.61 ± 8.42*	*84.19 ± 4.90*	**Mamographic Mass**	*RBM+*	84.85 ± 16.54	85.18 ± 14.98	82.42 ± 20.30	85.18 ± 14.98
	KNN	94.08 ± 12.12	95.97 ± 7.32	94.82 ± 10.59	95.97 ± 7.32		*KNN*	99.65 ± 0.88	99.64 ± 0.89	99.64 ± 0.89	99.64 ± 0.89
	LR	**99.52 ± 0.18**	**99.49 ± 0.18**	**99.49 ± 0.18**	**99.49 ± 0.18**		*LR*	99.41 ± 0.30	99.40 ± 0.30	99.40 ± 0.30	99.40 ± 0.30
	MLP	93.78 ± 1.40	93.28 ± 1.52	92.85 ± 1.64	93.28 ± 1.52		*MLP*	98.91 ± 2.11	98.79 ± 2.35	98.79 ± 2.35	98.79 ± 2.35
	RAN	90.07 ± 0.43	89.18 ± 1.23	89.32 ± 1.10	89.18 ± 1.23		*RAN*	*80.28 ± 0.18*	*79.20 ± 0.23*	*79.08 ± 0.24*	*79.20 ± 0.23*
	SGD	97.95 ± 0.66	97.87 ± 0.69	97.82 ± 0.70	97.87 ± 0.69		*SGD*	**99.96 ± 0.03**	**99.94 ± 0.07**	**99.93 ± 0.09**	**99.94 ± 0.07**
IRIS	*RBM+*	*79.81 ± 11.91*	*77.41 ± 11.88*	*70.66 ± 16.28*	*77.41± 11.88*	**Wine Recognition**	*RBM+*	*56.00 ± 25.66*	*67.05 ± 16.91*	*59.07 ± 21.91*	*67.05 ± 16.91*
	KNN	90.41 ± 28.77	92.80 ± 21.61	91.00 ± 27.01	92.80 ± 21.61		*KNN*	90.74 ± 26.00	92.88 ± 19.48	91.14 ± 24.70	92.88 ± 19.48
	LR	**97.38 ± 4.15**	96.64 ± 5.65	96.45 ± 6.12	96.64 ± 5.65		*LR*	94.14 ± 1.55	93.13 ± 1.82	93.00 ± 1.92	93.13 ± 1.82
	MLP	97.31 ± 0.71	**96.86 ± 1.13**	**96.81 ± 1.21**	**96.86 ± 1.13**		*MLP*	97.44 ± 0.51	97.33 ± 0.59	97.32 ± 0.59	97.33 ± 0.59
	RAN	95.43 ± 0.67	95.02 ± 0.94	94.98 ± 0.98	95.02 ± 0.94		*RAN*	94.87 ± 0.91	94.34 ± 1.00	94.29 ± 1.01	94.34 ± 1.00
	SGD	94.47 ± 6.40	94.46 ± 5.20	93.31 ± 6.78	94.46 ± 5.20		*SGD*	**98.13 ± 0.70**	**97.91 ± 0.75**	**97.91 ± 0.76**	**97.91 ± 0.75**

RBM+- Restricted Boltzmann Machine + Pipelined with Logistic Regression; KNN- K Nearest Neighbor; LR- Logistic Regression; MLP- Multi Layer Perceptron; RAN- Regulated Activation Network; SGD- Stochastic Gradient Descent



(A) Precision (B) Recall (C) F1-Score (D) Accuracy

FIGURE 3.12: RAN's evaluation metric (Precision, Recall, F1-Score and Acuracy) value behavior w.r.t. varying test and train data ratio over ten datasets [Mice Protein (MP), Breast Cancer 669 (BC1), Breast Cancer 569 (BC2), Credit Approval (CA), IRIS data (ID), Mamographic Mass (MM), Human Activity Recognition (HAR), Toy-data(TD), Wine Recognition (WR) and Glass Identification (GI)].

Chapter 4

Non-convex Abstract Concept Modeling with Regulated Activation Networks

This chapter describes the RAN's non-convex concept modeling technique, and is based upon the following National/International Conferences and posters, and part of under review Journal article. Following is the list of articles and the Technical reports:

- Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto and Amílcar F. Cardoso, *Learning Non-Convex Abstract Concepts with Regulated Activation Networks*, Journal of Annals of Mathematics and Artificial Intelligence, Springer, 2020. (Sharma et al., 2020b)

- Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto and Amílcar F. Cardoso, *Perceiving Abstract Concepts Via Evolving Computational Cognitive Modeling*, International Joint Conference on Neural Networks (IJCNN), IEEE World Congress on Computational Intelligence (IEEE WCCI), Rio de Janeiro, Brazil, 2018. (Sharma et al., 2018a)

- Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto, F. Amílcar Cardoso, Ngombo Armando,Duarte Raposo, Marcelo Fernandes, André Rodrigues , Jorge Sá Silva, Hugo Gonçalo Oliveira, Luis Macedo, Fernando Boavida, *Unveiling Markers of Stress Via Smartphone Usage* 24th Portuguese Conference on Pattern Recognition, Coimbra Portugal, RecPad-2018. (Sharma et al., 2018c)

- Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto and Amílcar F. Cardoso, *Identifying Psychological Bio-markers Computationally with Regulated Activation Networks*, in Encontro Ciência '18 (Poster), 2018.

4.1 Introduction

Investigations related to concepts is an essential domain in cognitive and psychological research. Theoretically, concepts pertain to process-oriented, symbolic or distributed, and knowledge-based conceptual representations (Kiefer and Pulvermüller, 2012). Usually, these conceptual representations are actualized via mathematical, theoretical (conceptual), or computational cognitive modeling approaches (Bechtel et al., 1998). In general, concepts are a hierarchy, where, at a higher level abstract concepts are placed, and concrete concepts are situated at, relatively, lower levels, thus, it is evident to infer the abstract concepts as the congregation of concrete concepts (Rosch et al., 1976, Tversky and Hemenway, 1984). There have been efforts to study abstract concepts mathematically (Saitta and Zucker, 1998), and theoretically (Borghi et al., 2018, 2017), however, they are seldom addressed computationally (Kiefer and Pulvermüller, 2012). This chapter proposes a computational modeling approach that builds a representation of non-convex abstract concepts via an evolving topology.

Initial computational modeling research (Rolls et al., 2008, Kyaga et al., 2013, Braver et al., 1999, O'Reilly, 2006) not only endorsed the potential of computational cognitive modeling but also helped researchers to understand the cognitive and psychological processes, validating the existing theories and encouraging to conceive or propose a new hypothesis. Computational cognitive modeling mainly comprises three types of techniques: first, are biologically inspired connectionist models Artificial Neural Networks (ANNs) (Van Gerven and Bohte, 2018), including Restricted Boltzmann Machine (RBMs) (Van Gerven and Bohte, 2018), Stacked Auto-Encoders (Vincent et al., 2010), Deep Neural Networks (Collobert and Weston, 2008) and Convolution Neural Networks (Krizhevsky et al., 2017), showing a remarkable contribution in the field like classification, and feature recognition; second, are symbolic cognitive architectures that, typically, conforms to Newell's symbol system[1] and are usually implemented as the *if-then-else* set of rules and operations. LIDA (Madl and Franklin, 2015), SOAR (Laird, 2012), ACT-R (Anderson et al., 1997) are a few symbolic cognitive architectures and contributed to almost all domains of Artificial Intelligence (AI) (such as attention, memory, language translation, decision making, etc.); the third category belongs to systems that are a blend of both symbolic and distributed systems such as CLARION (Sun and Peterson, 1996).

The computational modeling approach introduced in this chapter is a protraction of Regulated Activation Network (RAN) modeling (Pinto and Sharma, 2016) (as described in Chapter 3). RAN's model is uniquely hybrid, it is not only symbolic, and distributed, but also spatial, i.e., considers a ⟨feature-value⟩ pair of each attribute as a point in n-dimensional feature hyperspace. With RANs modeling, first convex abstract concepts are identified; further, similar convex abstract concepts are represented by a node, depicting non-convex abstract concepts (see Figure 4.1

[1] A system that contains memory (of tokens, the physical representation of a symbol), symbols (tokens that represent distal information), operations (that manipulate symbols), interpretations (of symbols to perform operations), and capacities (memory, composability, and interpretability).

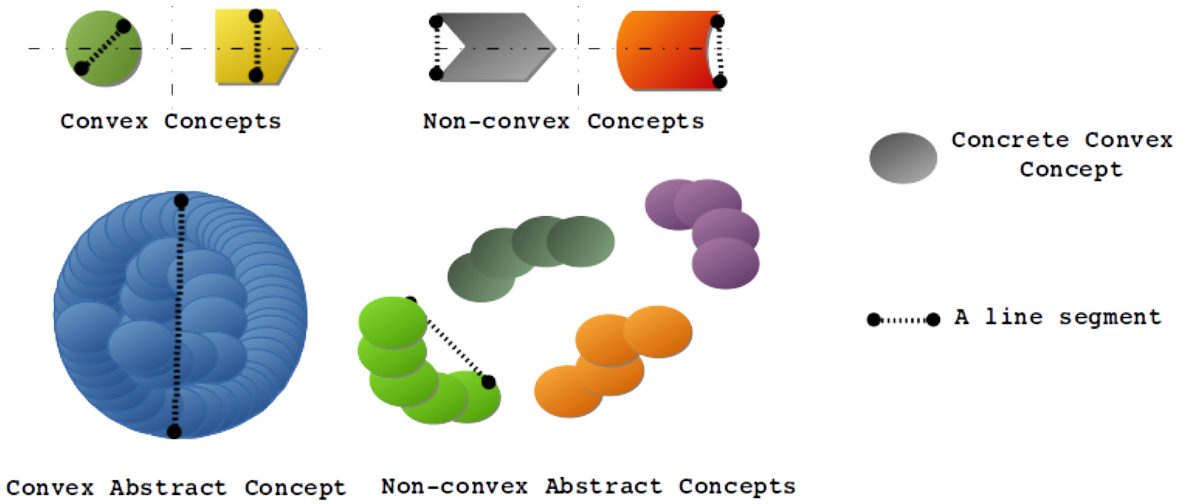


FIGURE 4.1: Elucidation of geometric understanding of convex and non-convex concepts. A geometric region is said to be convex when any line segment joining any two points lie entirely within the region, as shown by the diagram of convex concepts and convex abstract concepts. The non-convex regions are those where any line segment joining any two points falls out of the region. This Figure also illustrates the formation of convex or non-convex abstract concept via an amalgamation of two or more circular concrete convex concepts.

for the description of convex and non-convex concept geometric relevance, and how a set for concrete convex concepts tend to represent convex and non-convex abstract concepts). Besides being hybrid, RANs modeling exhibits a dynamic topology by arbitrarily growing the network upon identifying abstract concepts, both convex and non-convex, and forming their respective layers. In this chapter, RANs methodology is illustrated with an artificially generated data (i.e. Toy-data2) and demonstrate how cognitive functions of *concept creation*, *learning* and *activation propagation* are simulated. An experiment, identical to Toy-data2 was performed with data from Human Activity Recognition problem, in pursuit of determining abstract concepts depicting Mobility and Immobility. A study is also reported through this chapter, illustrating how the RAN's model can be beneficial in determining psychological and physiological markers (such as stress and illness), using a Sleep Detection data obtained from the Internet of Everything (IoE) (Joseph et al., 2017) Source, i.e., smartphone app ISABELA[2] from project SOCIALITE[3] (Armando et al., 2017). Further, RAN's proof of concept as a classifier is provided using the four benchmark datasets obtained from the UCI machine learning repository. Similar to Chapter 3, the RANs classification capability is exploited to prove the hypothesis related to non-convex abstract concept modeling.

The formulation of this chapter is as follows: In Section 4.2 the entire non-convex concept modeling with RANs is explained using Toy-data2; Experiment and outcome with Toy-data, Human Activity Recognition data, Sleep Detection data and four datasets UCI Machine Learning Repository is reported in Section 4.3; Section 4.4 puts forward the conclusions.

[2]IoT Student Advisor and BEst Lifestyle Analyzer (ISABELA)
[3]Social Oriented Internet of Things Architecture, Solutions and Environment
https://www.cisuc.uc.pt/projects/show/215

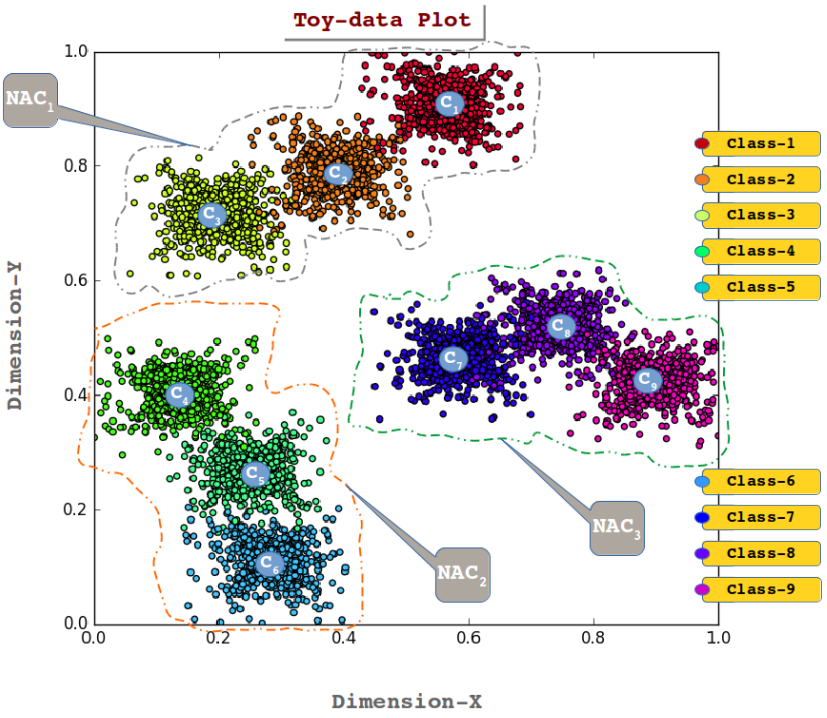


FIGURE 4.2: Plot of the artificially generated 2-D data (Toy-data2). The graph shows nine clusters depicted as Class-1,..., Class-9 along with their respective centroids (C_1,..., C_9). The Figure also marks three non-convex abstract concept (NAC) regions that are being determined by RANs methodology.

4.2 Non-convex Concept Modeling with RANs

The proposed methodology of RANs modeling incorporates nine core steps performed in three pipelined stages (the Steps 1-to-4 are already explained in Chapter 3). Besides these steps, the technique also has an optional method to associate the determined non-convex abstract concepts with the category label of input data. The section begins with the details related to notations, assumptions, boundaries, and data pre-processing methods suitable for RANs approach. Later, the RANs modeling is detailed using an artificially generated 2-D data, that has 5400 instances spread into nine categories, as shown in Figure 4.2.

RANs modeling is performed in three stages, where Stage-1 contains three steps for representing and learning convex abstract concepts, as described in Chapter 3. The Stage-2 concerns with two operations to carry out similarity relation learning while relaying the input data to newly created convex abstract concept layer. Stage-3 consists of four steps related to the creation, learning and activation propagation, to model non-convex abstract concepts. The RANs model described in this chapter simulates the cognitive functions of concept creation, learning and activation propagation. Unlike the three similar cognitive function in Chapter 3, this chapter introduces simulations related to a non-convex *concept creation* and *activation propagation*

Table 4.1: Notations

Notation	Description
W	Convex concept Inter-layer weight matrix
W'	Abstract concept Inter-layer weight matrix
w	Similarity relation weight matrix
C	Cluster center or centroids
A	Intermediate activation
A'	Output activation
a	Input activation
i, k, j	Variables to represent node index for 0^{th}, 1^{st} and 2^{nd} *layer* respectively
m, n	Arbitrary node indexes for any layer
I	I^{th} instance of input data
$f(x)$	Transfer function to obtain similarity relation
t	Variable used to depict intermediate index
n_a	Size of input Vector at Layer-0
n_A	Size of convex abstract concept vector at Layer-1
$n_{A'}$	Size of non-convex abstract concept vector at Layer-2

cognitive functions. The chapter also describes two new *learning* cognitive functions. Appendix A describes the optional method for labeling of non-convex abstract concepts.

4.2.1 Stage-1 Convex Concept Modeling

Step 1: Convex concept Identification (CCI)

The CCI operation (detailed in Section 3.3.1 of Chapter 3) is applied at the input level, i.e., at Layer-0 (see Figure 4.3). In this experiment, the K-means (Hartigan and Wong, 1979) clustering algorithm was used to identify nine groups and their centers at Layer-0 (see Step-1, in Stage-1 of Figure 4.3). The centroids recognized in the CCI process are used as learning (detailed in Section 4.2.1). Figure 4.2 displays the centroids (C_1, ..., C_9) of all the clusters, recognized as Cluster Representative Data Points (CRDP) of all nine classes.

Step 2: Convex Abstract Concept Creation (CACC)

In CACC (detailed in Section 3.3.2 of Chapter 3), we create a new layer (Layer-1) having nine nodes (see Step-2, in Stage-1 of Figure 4.3). Each node at Layer-1 acts as a convex abstract concept for clusters identified in CCI, Figure 4.2 shows all nine groups represented by the nine nodes in Layer-1, in Step-2 at Stage-1 of Figure 4.2.

Step 3: Convex Concept Inter-Layer Weight (CCILW) Assignment

In the experiment, nine cluster centers were identified in CCI process and learned as CCILW W (the CCILW assignment is explained in Section 3.3.3 of Chapter 3). The weight W_1 is

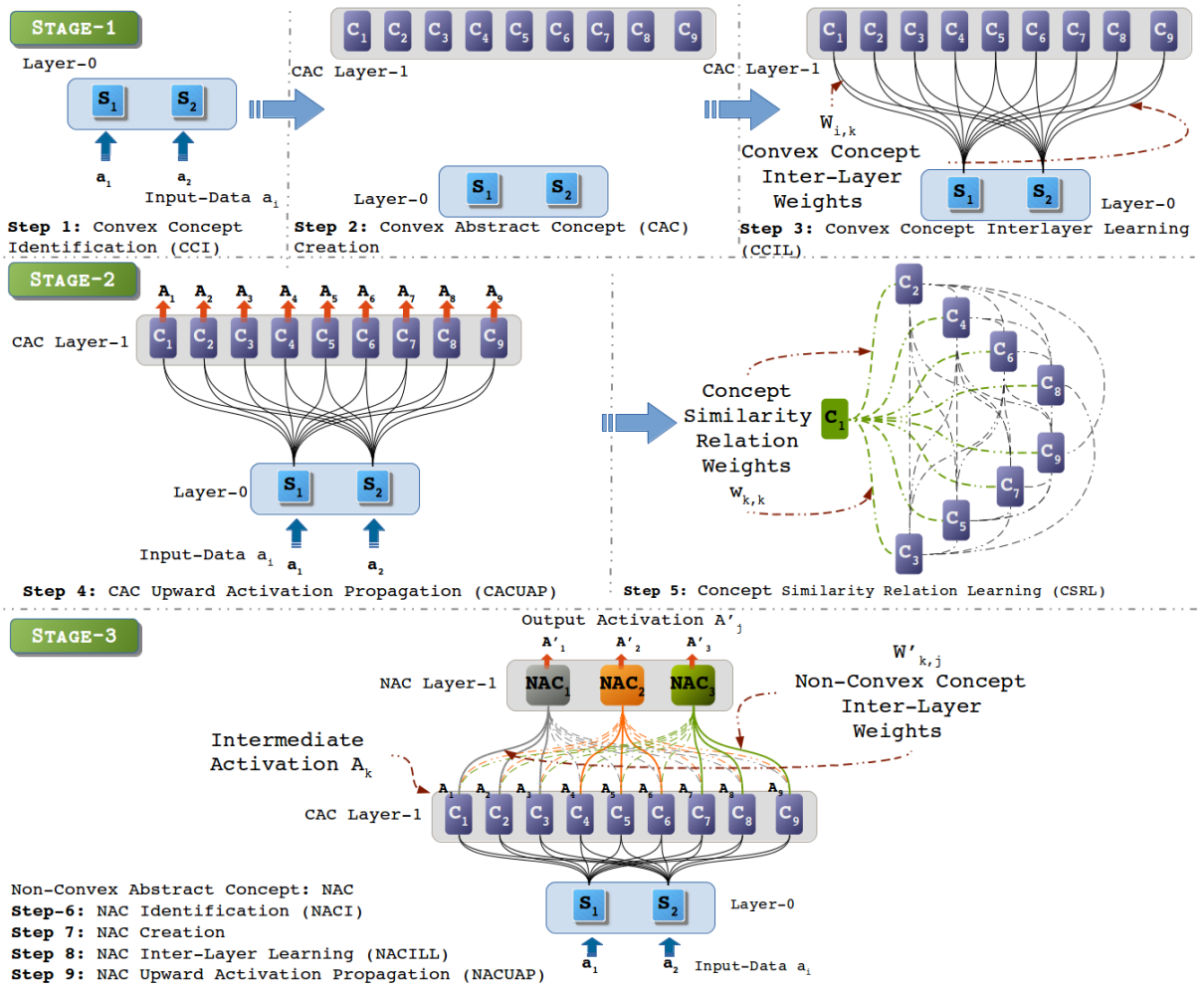


FIGURE 4.3: The 3 stages in RANs modeling technique: Stage-1 has 3 steps to model convexity, Stage-2 comprises 2-steps primarily to learn similarity relation, and Stage-3 contains the last four steps of Non-convex modeling.

$(W_{1,1}, W_{2,1})$ (the center-points of cluster C_1 see Figure 4.2 for the center of cluster C_1). Consequently, a 2×9 weight matrix W was obtained as the relation learned between nodes of Layer-0 and Layer-1 (see Step-3 in Figure 4.3).

4.2.2 Stage-2 Intra Layer Learning

Step 4: Convex Abstract Concept Upward Activation Propagation (CACUAP)

After modeling the CACs in Stage-1 of RANs methodology, Stage-2 follows with Step-4 for propagating i-dimensional input data instances a_i from the input layer (Layer-0) to k-dimensional output vector A_k at CAC layer (Layer-1). The Algorithm 1 (described in Section 3.3.4 of Chapter 3) is used to propagate the activation from input layer to CAC layer. Algorithm 1 demonstrates how the CCILW is utilized as an attractor to determine activation at nodes in the CAC layer (Layer-1) and exhibiting the DoC with which the CAC node in CAC layer represents the input data.

Step 5: Concept Similarity Relation Learning (CSRL)

The main contribution of this chapter starts from this step-5, here first learning mechanism is simulated where associations among the peer concepts are established. The prime objective of the CSRL process is achieved by determining the alikeness among the concepts at CAC layer (Layer-1) and interrelate them through a similarity function. This function also emulates the activation behavior of biological neurons, i.e., upon receiving an input stimulus several neurons that get activated concurrently are termed as affine, whereas the ones that are relatively inactive simultaneously may not be considered identical. This biological phenomenon is expressed mathematically through Equation 4.1, to calculate a pair-wise relation/weight $w_{m\to n}$ between node m, and node n at CAC layer (Layer-1). The numerator $(1 - |A_m^I - A_n^I|)$ calculates the similarity of activation[4] of node m w.r.t. node n, and the product $(1 - A_m^I) * (1 - A_n^I)$ is used to reduce the impact of similarity on weight $w_{m\to n}$ when both activations (i.e. A_m^I, and A_n^I) are very close to 0, though, similar. Consequently, we obtain a symmetric $k \times k$ matrix as, learned, concept similarity relation weights (CSRW) among the nodes within the layer.

$$w_{m\to n} = \frac{\sum_I [(1 - |A_m^I - A_n^I|) - (1 - A_m^I) * (1 - A_n^I)]}{\sum_I [1 - (1 - A_m^I) * (1 - A_n^I)]} \tag{4.1}$$

where $m \in 1, ..., k$; $n \in 1, ..., k$; and $m \neq n$

In the experiment concerned with Toy-data2 problem, the propagation of input data instances to CAC layer (Layer-1) generates a 9×9 matrix.

4.2.3 Stage-3 Non-convex Abstract Concept Modeling

Step 6: Non-convex Abstract Concept Identification (NACI)

NACI is the primary step (Step-6) in Stage-3 of RANs modeling (see Figure 4.3). The purpose of this operation is to recognize the CACs that are akin to one another, convene them, and represent them by a unique concept non-convex in nature (detailed in Section 4.2.3). To determine related CACs, the CSRW (see Section 4.2.2) is inspected, i.e., each weight $w_{k,k}$ viewed and a set of CSRWs are selected as potential Similarity-Threshold (ST)[5] values. Further, an ST is chosen heuristically. For example, in Toy-data2 use-case, at Layer-1, we compare this threshold (ST) value of 0.72 with all CSRW from node C_1 i.e. $w_{1\to 2}$, $w_{1\to 3}$, $w_{1\to k}$, $w_{1\to 9}$, and found nodes C_2, and C_3 which are closely related. The process terminates when all the nodes are traversed either in groups or individually. In our experiment, we recognized three non-convex classes as non-convex abstract concept (NAC) (NAC_1, NAC_2, and NAC_3 see Figure 4.2).

[4] A_m^I is the activation of I^{th} instance of propagated data at node C_m in Layer-1, and similarly A_n^I is the activation of I^{th} instance of propagated data at node C_n. $m \neq n$, and m, n are integer in range [1, 9]

[5] At present this value is chosen manually from SLW matrix, by conducting several trial and selection process

Step 7: Non-convex Abstract Concept Creation (NACC)

The second concept creation operation is simulated by dynamically creating a non-convex abstract concept (NAC) layer (Layer-2 in the model shown in Stage-3 of Figure 4.3), relying upon the number of related CACs determined in NACI operation. The nodes of the NAC layer represents the associated nodes of the underlying CAC layer (Layer-1). Figure 4.2 illustrates the three sets, consisting of the three CACs in each group, forming three NACs (i.e., NAC_1, NAC_2, and NAC_3 see Figure 4.2). Further, a new NAC layer (Layer-2) is created dynamically with three nodes (NAC_1, NAC_2, NAC_3 in Stage-3 of Figure 4.3), representing the three discovered convex concept groups at Layer-1.

Step 8: Non-convex Abstract Concept Inter-layer Learning (NACIL)

As aforestated, the NACs are representative of the underlying group of related CACs. The relation between NACs and CACs is termed as non-convex concept Inter-Layer Weight (NCILW) which is also determined during NACI operation. The relation between CAC layer (Layer-1) and NAC layer (Layer-2) is conceived as binary relation so that the activation of the underlying CAC nodes in Layer-1 impacts the activation of the related NAC(s) only, in layer-2. Hence, the weights are binary where the value of $W'_{k,j}$ is "1", indicating the association of CAC node k in layer-1 with node j in NAC layer-2, whereas the "0" value of $W'_{k,j}$ signifies the dissociation of node k in layer-1 with node j in NAC layer-2.

$$W' = \begin{bmatrix} W'_{1,1}, W'_{1,2}, \ldots, W'_{1,j} \\ \ldots \\ W'_{t,1}, W'_{t,2}, \ldots, W'_{t,j} \\ \ldots \\ W'_{k,1}, W'_{k,2}, \ldots, W'_{k,j} \end{bmatrix} = \begin{bmatrix} 1,1,1,0,0,0,0,0,0 \\ 0,0,0,1,1,1,0,0,0 \\ 0,0,0,0,0,0,1,1,1 \end{bmatrix} \tag{4.2}$$

Equation 4.2 shows the non-convex concept inter-layer weight (NCILW) a generic $k \times j$ matrix, and a 3×9 matrix learned from the Toy-data2 problem. The row corresponds to the number of NAC nodes in Layer-2, whereas the columns point to the number of CAC nodes at CAC layer-1. This NCILW matrix is the second *learning* simulation of this chapter.

Step 9: Non-convex Abstract Concept Upward Activation Propagation (NACUAP)

In this step the second activation propagation operation of RANs modeling which simulates the activation propagation mechanism convex concept layer to non-convex concept layer. Having generated the representation for NACs, the last step (Step-9) propagates activation of CACs of Layer-1 to their respective NACs at Layer-2. Since, each NAC represents a set of CACs in Layer-1 it is evident that NACs must exhibit high activation irrespective of which of its corresponding CAC is active (i.e., a high activation must be observed a NAC node (at Layer-2), even though

Algorithm 3: Non-convex Abstract Concept Upwards Activation Propagation algorithm

Input: Vector $[A_1, A_2, ..., A_{n_A}]$ as input at layer $l+1$
Output: New activation vector $[A'_1, A'_2, ..., A'_{n_{A'}}]$ in layer $l+2$
foreach *Node* A'_j *in layer* $l+2$ **do**
  Calculate Normalized Euclidean Distance:
  $A'_j = [1 - \prod_{k=1}^{n_A} (1 - W'_{k,j} * A_k)]$;
Where:
  k= [1, 2, ..., n_A];
  j= [1, 2, ..., $n_{A'}$];
  $W_{k,j}$ is NCILW see Equation 4.2

only one of its underlying CAC node (at Layer-1) gets high activation). For instance (see the model in Stage-3 of Figure 4.3) if at Layer-1 only node C_1 received high activation, whereas C_2 and C_3 get comparatively less activation, then also node NAC_1 should receive high activation.

$$A'_j = [1 - \prod_{k=1}^{n_A} (1 - W'_{k,j} * A_k)] \tag{4.3}$$

Equation 4.3 suffices the prerequisite as mentioned earlier of NACUAP operation; it enables us to propagate the intermediate activation A_k from Layer-1 to obtain output activation A'_j at Layer-2. Since $W'_{k,j}$ has binary values, hence only those nodes C_k that are represented by node A'_j, contribute activation (A_k) for the propagation operation. Algorithm 3 demonstrates how the NACUAP technique is applied to obtain activation at NAC nodes in Layer-2, upon receiving an intermediate activation at nodes in CAC layer-1.

4.2.4 Evaluation of RANs modeling

In order to use RANs, all three stages of the modeling procedure must be performed, by realizing all nine steps in the order as described in Section 4.2. After the ninth step in Stage-3 a three-layered model will be obtained, where the first layer is the Input-layer (Layer-0), the second layer (layer-1) is CAC layer, and the third layer (Layer-2) is NAC layer. To label the NAC nodes at NAC layer (Layer-2) non-convex abstract concept labeling operation is performed (see Appendix B.2), and consequently the input labels are attached to the NAC nodes. Further, the NACL operation generates the *True* and *Test* labels to obtain the multi-class confusion matrices. Later, the generated confusion matrix was used to calculate the evaluation metrics Precision, Recall, F1-Score and Accuracy. The Operational points were also determined using the Receiver Operation Characteristics (ROC) and Area Under Curve (AUC) analysis. Appendix B.3 describes the transformations of class-labels into binary labels and confidence-score calculations for ROC-AUC analysis. Once the transformed binary labels and confidence-scores are obtained the ROC curves are plotted to illustrate the operational points and determine the AUC of each class being represented by individual NACs at NAC layer.

The time complexity of Non-convex concept modeling is dependent upon complexity of all the nine steps. The complexity of RAN's Convex Concept Modeling is *O(f(n))* (refer Section3.3.5 in Chapter 3) where *f(n)* is the complexity of the concept identifier (or clustering algorithm). The complexity of CSRL method is given by equation 4.4, where n is number of data samples and c is the number of nodes in the layer L.

$$f(n) = O(n * c/2) \tag{4.4}$$

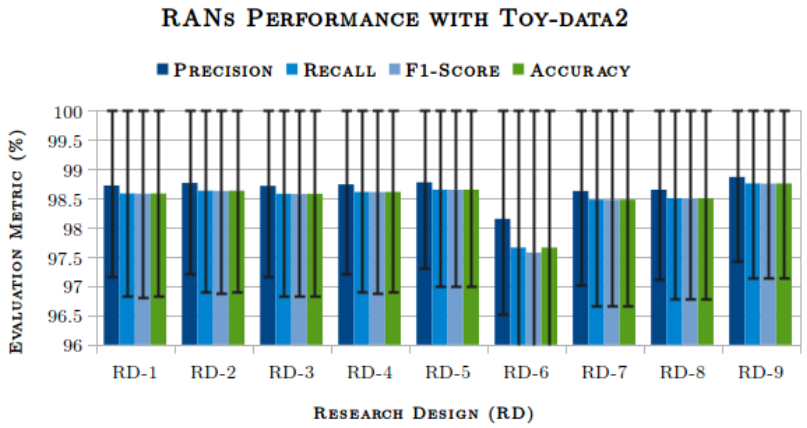
In RANs modeling, the 4 operations related to Non-convex modeling at stage-3 has different complexities: (1) the concept identification process is expressed as $O(c^2)$; (2) the concept creation has complexity of *O(c')* where *c'* is the number of groups of similar nodes; (3) the inter layer learning also has complexity of *O(c')* because it is an assignment operation and is equal to number of identified groups of similar nodes; (4) the upward activation operation has the complexity of *O(n)* when n is the number of data instances. Since c and *c'* values are significantly small therefore the overall complexity of the RANs modeling for creating a single layer is expressed by equation 4.5.

$$T(n) = O(max\{O(f(n)), O(n * c/2), O(n)\}) \tag{4.5}$$

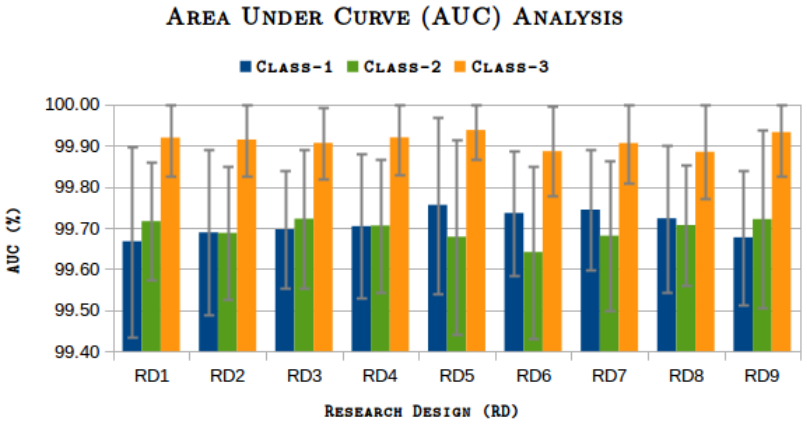
In a RAN's model, where a K-means algorithm is used, it's complexity is $T(n) = O(n^{(k+2/p)})$ ($O(n^{(k+2/p)})$ is the complexity K-means, refer Table 3.4 in Chapter 3 for complexities related to different algorithms).

4.3 Empirical Design, Results, and Inferences

This section describes the experiments performed with RANs modeling, reports and discusses results. The experiments were carried out with seven datasets; one set is an Artificially generated Toy-data2 problem (used in Section 4.2); another set is obtained from IoE source of project SOCIALITE; five other datasets are benchmarks obtained from UCI machine learning repository. With Toy-data2 30 iterations of the experiments were carried out, and with other datasets 10 times the experiments were performed for each dataset. Each iteration of the experiment is the same as used in Chapter 3 (refer Appendix B for experimental setup elaboration). RANs classification capability was compared with five different machine learning approaches, Multilayer Perceptron (MLP) (Rumelhart et al., 1986), Logistic Regression (LR) (Freedman, 2009), K Nearest Neighbors (K-NN) (Altman, 1992), Stochastic Gradient Descent (SGD) (Zhang, 2004b), and Restrict Boltzmann Machine pipelined with Logistic Regression (RBM+) (Hinton, 2012).



(A) Performance metrics



(B) ROC-AUC Analysis

FIGURE 4.4: Illustration of performance evaluation and ROC-AUC analysis of RANs modeling with Toy-data2 problem in nine research designs (RD). The ROC curve analysis plot depicts the AUC observed while validating the model.

4.3.1 Experiment with Toy-data2

The hypothesis of this experiment is to illustrate that nine convex clusters (depicted as clusters $Class1, ..., Class9$ in Figure 4.2) can be grouped and symbolically represented by three non-convex abstract regions (shown as NAC_1, NAC_2, and NAC_3 in Figure 4.2). This hypothesis can also be validated through a classification operation. To validate the assumption as mentioned earlier RANs modeling was performed with Toy-data set by traversing through nine steps as described in Section 4.2, and including the optional labeling operation, NACL (see Appendix B.2). The Stage-3 shows the pictorial view of the generated model in Figure 4.3. The nine CACs at Layer-1 corresponds to the nine convex clusters in Figure 4.2, and the three NACs at Layer-2 represents the three non-convex groups having three CACs in each (i.e. in Stage-3 of Figure 4.3, NAC_1 is mapped to the set CAC_1, CAC_2, and CAC_3, whereas NAC_2 corresponds to set CAC_4,

TABLE 4.2: RANs Comparative Study with Toy-data2

Model	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
RBM+	80.10 ± 8.45	73.26 ± 7.74	68.26 ± 11.63	73.26 ± 7.74
K-NN	99.99 ± 0.006	99.99 ± 0.006	99.99 ± 0.006	99.99 ± 0.006
LR	99.99 ± 0.002	99.99 ± 0.002	99.99 ± 0.002	99.99 ± 0.002
MLP	99.54 ± 1.36	88.48 ± 1.52	99.48 ± 1.53	99.48 ± 1.52
RANs	98.66 ± 0.23	98.48 ± 0.37	98.47 ± 0.40	98.48 ± 0.37
SGD	99.98 ± 0.02	99.98 ± 0.02	99.98 ± 0.02	99.98 ± 0.02

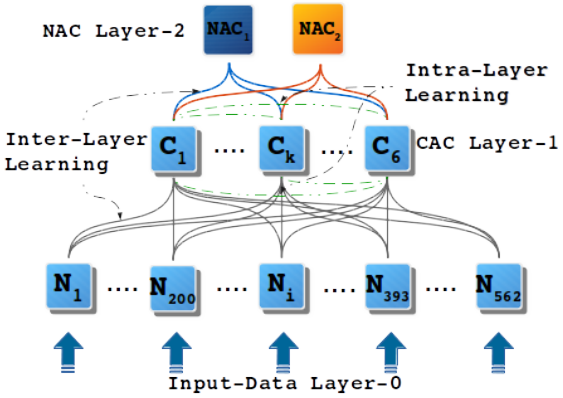
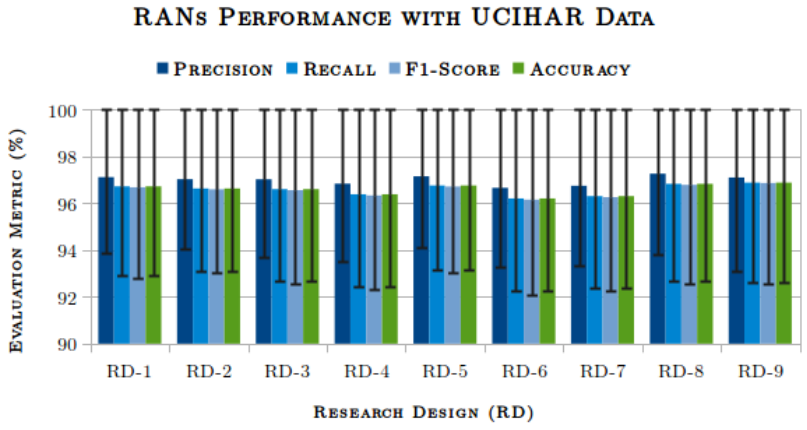


FIGURE 4.5: Model generated with the RANs approach for UCIHAR datasets. The Nodes NAC_1 and NAC_2 represents Motion and Stationary non-convex abstract concepts. The nodes C_1,.., C_6 represents the convex abstract concepts labeled as six activities Walking, Walking_upstairs, Walking_downstairs, Sitting, Standing, and Laying in the input data.

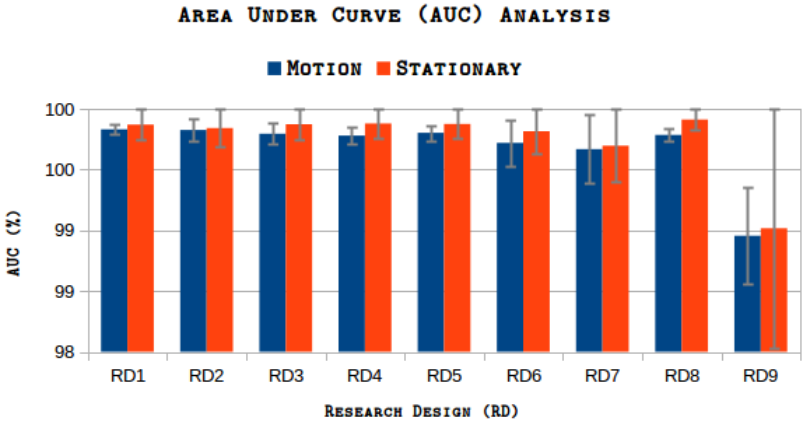
CAC_5, and CAC_6, and NAC_3 represents the set CAC_7, CAC_8, and CAC_9). The graph in Figure 4.4a shows the performance of the RAN's model for all RDs, depicting 98% (approximate) values for Precision, Recall, F1-Score, and Accuracy. The ROC-AUC analysis shown in Figure 4.4b for all RDs also exhibits the satisfactory operational characteristic of the model. The comparative study conducted with five distinct methodologies, shown in Table 4.2, portrays the RAN's fair performance, with an added advantage of being unsupervised, and having the ability to automatically map the CACs to NACs (i.e., in all five techniques the nine labels are converted in advance into three tags before evaluating, whereas this mapping is intrinsic in RANs methodology).

4.3.2 RANs Demonstration with Human Activity Recognition Problem

The goal of this experiment is to exhibit RANs methodology with benchmark data, and to show how it can represent similar categories of convex regions by the non-convex abstract concept. The use case of this illustration belonged to Human Activity Recognition (HAR) problem domain and obtained from the UCI Machine Learning Repository (its the same UCIHAR (Anguita et al., 2013) dataset is used in Section 3.4.2 of Chapter 3. The data was preprocessed to comply with the RANs requirement. The six labels of the dataset was merged to form two labels *Motion* and *Stationary* as described in Section 3.4.2 of Chapter 3.



(A) Evaluation metric



(B) ROC-AUC

FIGURE 4.6: Illustration of performance evaluation and ROC-AUC analysis in nine research designs (RD) for the RAN's model obtained with UCIHAR dataset. The ROC analysis is carried out for the two non-convex abstract concepts, i.e., Motion and Stationary, at Layer-2 of the generated model.

TABLE 4.3: RANs Comparative Study with UCIHAR dataset

Model	**Precision (%)**	**Recall (%)**	**F1-score (%)**	**Accuracy (%)**
RBM+	99.87 ± 0.05	99.87 ± 0.05	99.87 ± 0.05	99.79 ± 0.05
K-NN	99.94 ± 0.03	99.94 ± 0.03	99.94 ± 0.03	99.95 ± 0.03
LR	99.57 ± 0.22	99.57 ± 0.22	99.57 ± 0.22	99.57 ± 0.22
MLP	99.96 ± 0.02	99.96 ± 0.02	99.96 ± 0.02	99.96 ± 0.02
RANs	96.97 ± 0.22	96.57 ± 0.27	96.53 ± 0.28	96.57 ± 0.27
SGD	99.95 ± 0.03	99.95 ± 0.03	99.95 ± 0.03	99.95 ± 0.03

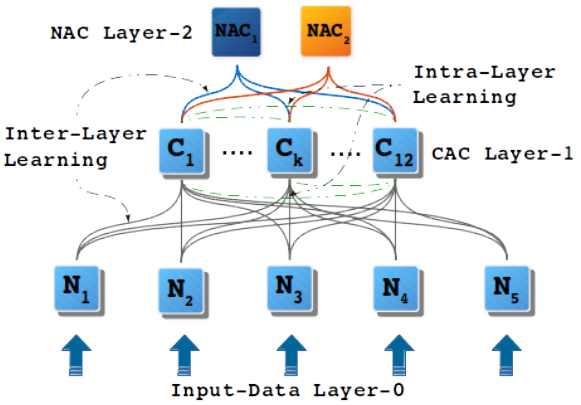


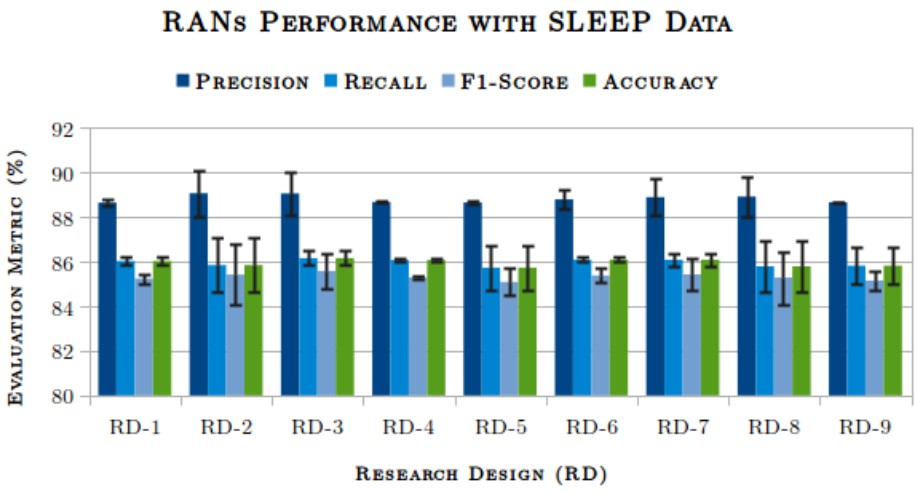
FIGURE 4.7: Model generated by RANs methodology with sleep detection data. The input layer has five nodes, Layer-1 has twelve CAC nodes, and Layer-2 depicts the two NAC nodes representing Active-Subject (NAC_1), and Inactive-Subject (NAC_2) respectively.

The modeling procedure commenced by selecting K-means as concept identifier with 'K' equal to 6, to determine six CACs as the convex abstract concept. The modeling progressed until Step-5 of Stage-2, whereby the learned CSRW was inspected to determine the Similarity-Threshold (ST) with a value of 0.75. Having obtained the ST value, Stage-3 realizes the RANs modeling, and representation is acquired as depicted in Figure 4.5. Upon performing the labeling operation NACL, it was observed that input data labels Walking, Walking_upstairs, Walking_downstairs were mapped to node NAC_1. The categories Sitting, Standing, and Laying were represented by node NAC_2. The label assignment commensurates with the hypothesis made earlier. The M-B and I-B are represented as non-convex abstract concepts obtaining an outcome of 96% (ca.) for all the parameters of validation (see Figure 4.6a). The ROC-AUC analysis also reflected a similar observation with an average area under the curve of 99% (approximate) for both NACs, i.e., Motion, and Stationary (see Figure 4.6b). To carry out comparative study the input labels were transformed into a binary label where Walking, Walking_upstairs, Walking_downstairs are grouped to form the first category (e.g., Label-1). The labels Sitting, Standing, and Laying was also combined to get the other set (e.g., Label-2). Further, these altered input labels were applied for validation. Table 4.3 logs the result of the comparisons displaying an acceptable performance, though, a bit lesser than the other techniques, with an advantage of automatically identifying, and associating input labels to NACs at Layer-2.

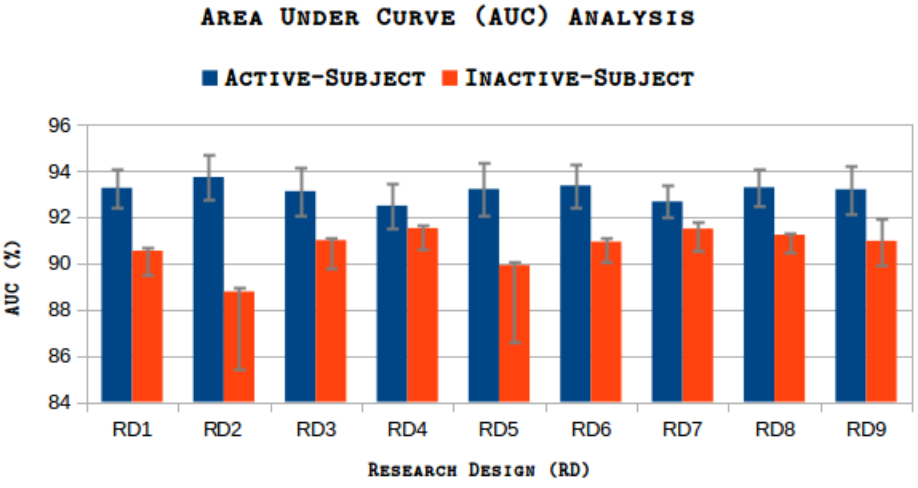
4.3.3 RANs Model Usage for Psychological and Physiological Bio-marking

The purpose of this experiment is to demonstrate the usability of RANs methodology with unlabeled data. Further exemplifying the identification of the psychological or physiological conditions of three subjects (students), based upon the statistical inferences for each student. The experiment uses sleep detection data (Sharma et al., 2017b), obtained from a smartphone app ISABELA of project SOCIALITE by capturing activities, the day of the week, light, sound,

phone utility usage, and time. For our experiment, only five attributes were utilized (*Activity, Day-of-week, Light, Sound, and Phone-Lock-State*).



(A) Evaluation metric



(B) ROC-AUC

FIGURE 4.8: Illustration of performance evaluation and ROC-AUC analysis of RANs modeling with Sleep Detection Data in nine Research Designs (RD). The ROC curve analysis plot depicts the AUC observed while validating the model.

Whether its social activity and inactivity, or physical activity and inactivity, both are responsible for adverse cognitive conditions (Cacioppo and Hawkley, 2009, Luanaigh and Lawlor, 2008, Sampson et al., 2009). This study aims to investigate students psychological and physiological conditions by modeling two abstract concepts "Active-Subject" (A-S), and "Inactive-Subject" (I-S), from the given data. The experiment begins by pre-processing the data, selecting K-mean as concept identifier with k value as 12 (value of 'k' is arbitrarily chosen to represent the number of CACs to be identified in Layer-1). Further, the modeling process is carried out until Step-5 to determine concept similarity relation weights (CSRW). Having inspected the

TABLE 4.4: Active-State (A-S) and Inactive-State (I-S) observations of the Three Subjects using model generated by RANs approach

		Student-1				**Student-2**				**Student-3**			
		Is-Inactive		**Is-Active**		**Is-Inactive**		**Is-Active**		**Is-Inactive**		**Is-Active**	
Days	**PoH**	**I-S (%)**	**DoC (%)**	**A-S (%)**	**DoC (%)**	**I-S (%)**	**DoC (%)**	**A-S (%)**	**DoC (%)**	**I-S (%)**	**DoC (%)**	**A-S (%)**	**DoC (%)**
Weekdays	*E-MH*	99.57	89.09	0.43	61.48	0.03	51.68	99.97	72.78	28.60	52.86	71.40	64.19
	M-H	89.17	91.18	10.83	72.09	22.57	69.21	77.43	85.25	8.55	63.19	91.45	87.67
	AF-H	89.61	91.35	10.39	71.29	84.65	90.07	15.35	75.67	50.00	64.68	50.00	65.17
	E-H	83.61	89.30	16.39	74.33	26.78	67.10	73.22	80.62	0.13	54.89	99.87	85.75
Weekends	*E-MH*	98.63	86.91	1.37	60.01	0.00	40.59	100.0	68.63	0.00	51.47	100.0	72.56
	M-H	93.14	93.98	6.85	70.34	29.09	69.12	70.91	82.22	5.33	57.91	94.67	83.40
	AF-H	95.03	93.51	4.97	69.70	3.06	62.33	96.94	87.29	57.58	79.35	42.42	76.47
	E-H	86.94	89.35	13.06	74.01	34.85	69.98	65.15	78.30	29.87	70.89	70.13	85.08
E-MH [Early Morning Hours 4 Am -To- 6 Am]; M-H [Morning Hours 6:01 Am -To- 11:59 Am] AF-H [Afternoon Hours 12 Pm -To- 5 Pm]; E-H [Evening Hours 5:01 Pm -To- 9:59 Pm] PoH [Period of Hours]; I-S [Inactive-State]; A-S [Active-State]; DoC [Degree of Confidence]													

CSRW, a suitable Similarity-Threshold (ST) was selected with a value of 0.70. Later, a model is built (see Figure 4.7) by performing all four steps of Stage-3. Layer-2 in Figure 4.7 depicts one non-convex abstract concept node NAC_1 as A-S, and the other node NAC_2 as I-S. Thus, we obtain two abstract nodes as expected.

For evaluating the generated model the data was also labeled for two classes, "Active-Subjects" (A-S), and "Inactive-Subject" (I-S), based upon following conditions:

- **If** Phone_is_locked and Activities are {Still, Tilting, and Unknown} and Luminescence are {'Pitch Black', 'Very Dark', 'Dark Indoors', 'Dim Indoors', 'Dim Outdoors', 'Cloudy Outdoors'} and Sound_value ⟨=12000 units **Then** Label the instance as an **Inactive-Subject**
- **Else** label the instance as an **Active-Subject**

Upon conducting the validation, the performance metrics were observed with Precision of 88.81% (ca.), Recall of 85.95% (ca.), F1-Score of 85.31% (ca.) and Accuracy of 85.95% (ca.), see Figure 4.8a for the performance of RANs for all Research Designs. The ROC analysis also displayed satisfactory results with AUC of 93.15%(ca.) for A-S, and 90.71% (ca.) for I-S, Figure 4.8b depict s the average AUC for both classes in all RDs.

The model, generated with the entire dataset, was considered as the generic Model for identifying A-S and I-S of students. The comprehensive model determining Active and Inactive states of students is used to discover percentage amount of A-S and I-S in three students for four Period-of-Hours (PoH) (see Table 4.4 for period classification), during weekdays, and weekend separately. Eight test cases, representing each student, were extracted from data altogether 24 test cases were created for three students. Each test case was passed through the generic model and percentage of Activity or Inactivity is calculated for the entire test set, for example, see

Figure 4.4 the test data of Early Morning Hour (E-MH) during weekdays for Student-1 depicted 0.42% of Active-State and 99.57% of Inactive-State. Table 4.4 displays all 24 test cases.

Generally, the following is presumably an idle behavior of students:

- On all days students are usually in Active-State during Morning Hours (M-H), Evening Hours (E-H), Afternoon Hours (AF-H).
- During the weekend at AF-H Inactive-State can be observed with the student as an exception.
- Every day at Early Morning Hours (E-MH) most students are in Inactive-State.

Based upon the afore-stated nature of student Activity interesting inference can be made, see Table 4.4. In Table 4.4 the A-S and I-S states of Student-2 and Student-3 resemble the idle state of student behavior indicating a healthy mental and physical condition of the students. It can also be observed in Table 4.4 that during weekdays, Student-2 was mostly inactive in A-FH and Student-3 was mainly active in E-MH showing variation in normal sleep behavior, which is very much related to stress. An atypical outcome was observed with Student-1's test sets. The student was always in an Inactive-State, it can be a sign of unfavorable psychological and physical conditions. The average Degree of Confidence (DoC) values, also endorses the inferences mentioned above based on their activity or inactivity (DoC explained in Section 4.2.2).

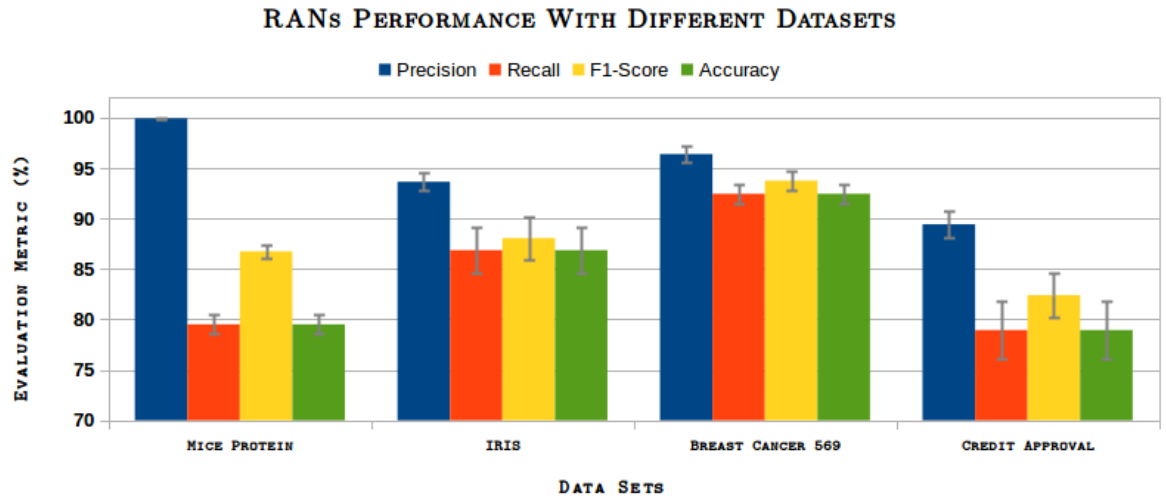
4.3.4 Substantiation of RANs modeling as Classifier

This portion of the chapter focuses on establishing RANs ability to perform classification task w.r.t distinct domains. The experiments were conducted to support the RANs usability with four benchmark data sets. The classification presented in this section identifies categories, provided with the labels of the datasets[6], as non-convex abstract concepts. A comparative study was also performed using five methodologies as mentioned earlier.

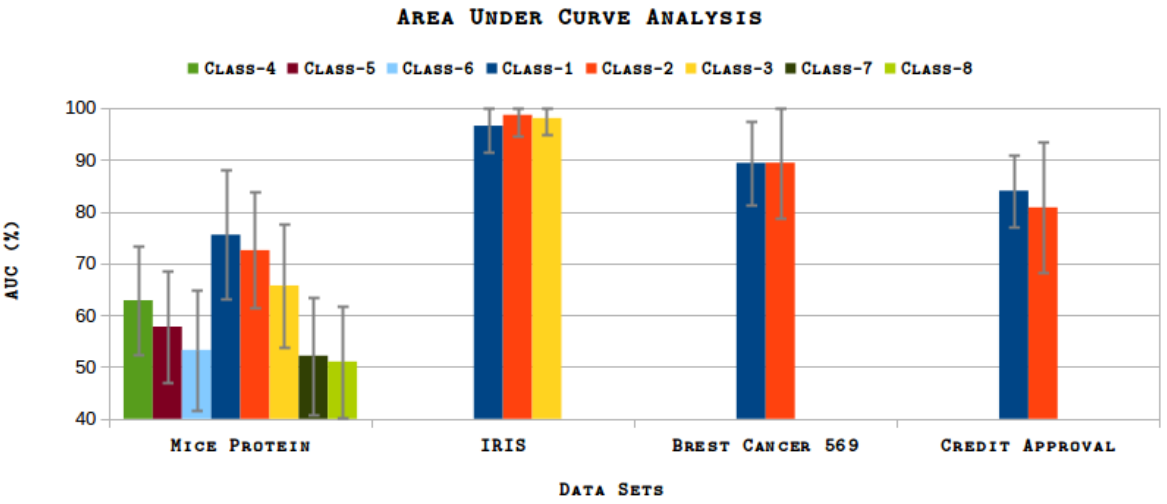
The Mice Protein (Higuera et al., 2015), Breast Cancer 569 (Street et al., 1993) datasets represent data from Medical domain, whereas Credit Approval (Quinlan, 1999) is a finance-related data, and the classical IRIS (Fisher, 1936) dataset is from botanical discipline. All investigations uses the pre-processed data, and K-mean algorithm as a concept identifier with 'K' value as 12 to identify twelve convex abstract concepts. In all use-cases, after Step-5, the CSRWs were inspected to identify appropriate Similarity-Threshold[7] values. All four studies execute the Stage-3, and models were obtained having 8, 3, 2, and 2 non-convex abstract nodes at Layer-2 for Mice protein, IRIS, Breast Cancer 569 and Credit Approval datasets, respectively.

[6]In experiments demonstrated in Section 4.3.1, and Section 4.3.2 the data labels are to be transformed based upon the hypothesis non-convex abstract concept, however experiment in Section 4.3.4 considers data's original categories to be identified as non-convex abstract concept, and therefore original labels are used in validation.

[7]Refer Table B.3 of Appendix **??** for ST values used in the experiments



(A) Performance metrics



(B) ROC-AUC

FIGURE 4.9: Illustration of performance evaluation and ROC-AUC analysis of RANs modeling with four datasets of distinct domains. The graph 4.9a shows the Precision, Recall, F1-Score and Accuracy metrics observed while evaluating the RAN's model. The graph 4.9b depicts the Area Under Curve (AUC) while performing ROC curve analysis for RAN's model generated with four datasets. The graph shows the plot of percentage AUC for classes 1 to 8. In the graph 4.9b class labels of each dataset is serially mapped as: *Mice protein* {*c-CS-s (Class-1), c-CS-m (Class-2), c-SC-s (Class-3), c-SC-m (Class-4), t-CS-s (Class-5), t-CS-m (Class-6), t-SC-s (Class-7), and t-SC-m (Class-8)*}; *IRIS)* {*Setosa (Class-1), Versicolar (Class-2), and Verginica (Class-3)*}; *Breast Cancer 569* {*Benign (Class-1), and Malignant (Class-2)*}; *Credit Approval* {*Postitive (Class-1), Negative (Class-2)*}

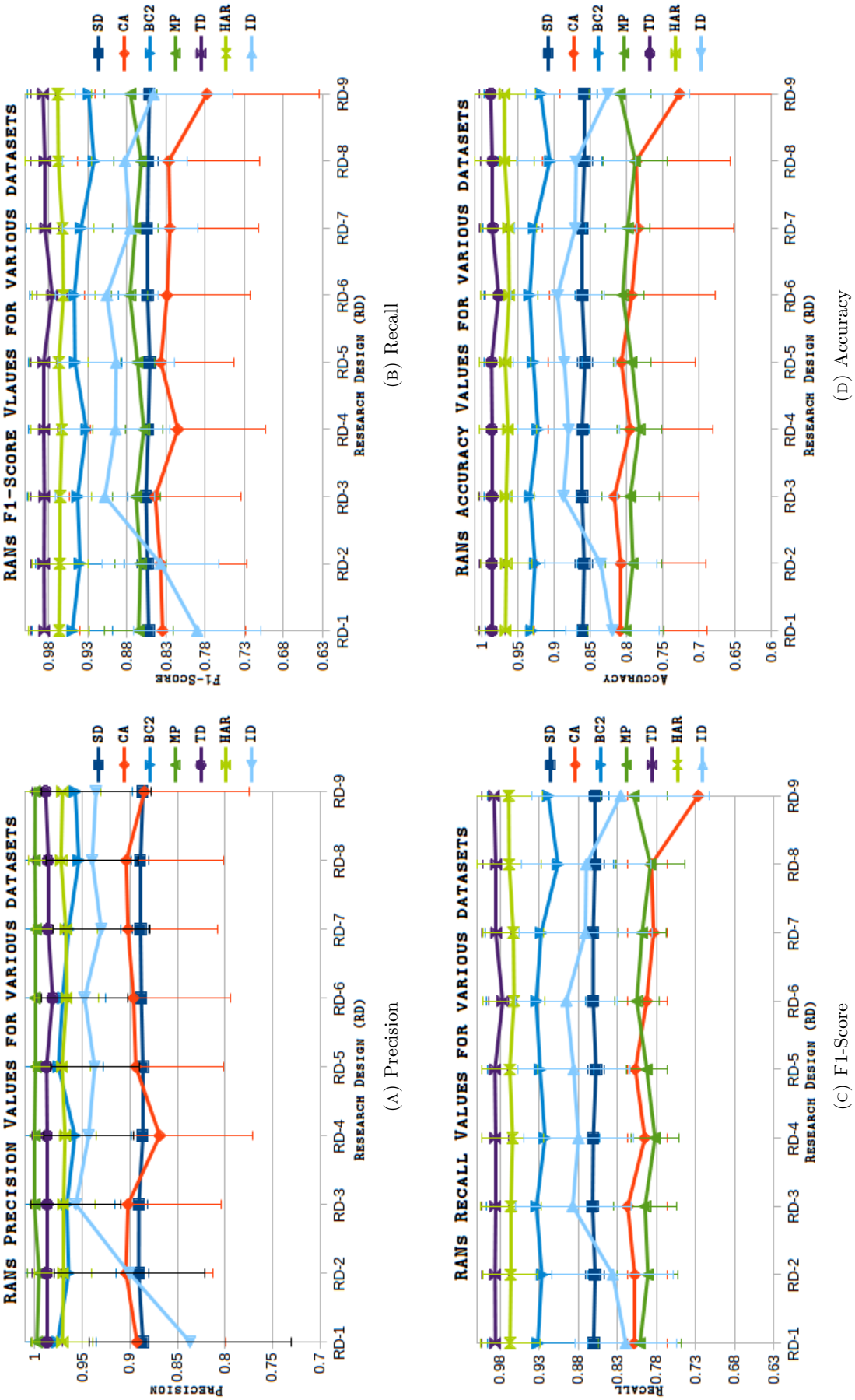
Figure 4.9a shows the evaluation graph plotted for observed Precision, Recall, F1-Score, and Accuracy metrics. The results with Mice Protein data depicts a Precision of 99.92% (ca.), Recall of 79.5% (ca.), F1-Score of 86.74% (ca.) and Accuracy of 79.5% (ca.). The observations with Breast Cancer 569 dataset obtained approximately 96.38%, 92.45%, 93.74%, and 92.45% of evaluation for the four metrics. With the IRIS dataset, the outcome was approximately 93.64%, 86.86%, 88.06%, and 86.86% for the four metrics. Furthermore, the assessment with Credit Approval data displayed the score values of 89.43%, 78.94%, 82.40%, and 78.94% (ca.) for four valuation parameters. Figure 4.9b shows the ROC curve analysis for the various classes in four datasets. In Mice Protein dataset the classes *c-CS-s, c-CS-m, c-SC-s, c-SC-m* observed to have higher representation, i.e., AUC 75.53%, 72.54%, 65.73%, 62.87% for each class. Whereas the remaining four categories *t-CS-s, t-CS-m, t-SC-s, and t-SC-m* depicted relatively lower operation characteristics with AUC of 57.79%, 53.29%, 52.16% and 51.04%, respectively. The ROC-AUC analysis of three groups *Setosa, Versicolar, and Verginica* of IRIS data obtained 96.54%, 98.63% and 98.04% of AUC for all three classes. The operational characteristics for the Breast Cancer 569 data were 89.39% and 89.41% of AUC for *Benign, and Malignant* NACs. Lastly, The AUC of *Positive, Negative* types of Credit Approval data were 84.01%, and 80.78%.

Table 4.5 logs the comparative study of RANs performance using four datasets (the best results are highlighted in bold and the worst outcome are displayed in italics). The experiment used the results of RANs methodology along with five methods (i.e., RBM+, K-NN, LR, MLP, and SGD) for comparisons. In the outcome of IRIS, Breast Cancer 569 data the RANs performance is not only satisfactory but also competent with other methodologies especially with RBM+ in both datasets. The outcome of the analysis of Mice Protein and Credit Approval datasets with RANs revealed a decent performance, with relatively better results when compared with valuation produced by RBM with both the datasets. Collectively, the assessments substantiate the RANs capability as a classifier, with the benefit of being an unsupervised modeling approach.

Figure 4.10 plots four graphs, showing Precision, Recall, F1-Score, and Accuracy values for nine Research Designs. While traversing from RD-1 to RD-9, the size of the training data varies from 90% to 10%. Despite having a variable RD, it appears that the performance of RANs remains, relatively, same for all datasets. The performance with IRIS and Credit Approval data showed the maximum variance, whereas, the efficiency with Sleep Data and Toy-data2 were almost the same in all RDs. This ability to learn and perform equivalently irrespective of the size of the training data displayed a noticeable feature of RANs modeling.

TABLE 4.5: RANs Comparison as a Classifier with four datasets

Data	Algorithm	Precision (%)	Recall (%)	F1-Score(%)	Accuracy(%)	Data	Algorithm	Precision(%)	Recall(%)	F1-Score(%)	Accuracy(%)
Mice Protein	*RBM+*	*27.21 ± 34.38*	*38.43 ± 30.47*	*28.34 ± 33.73*	*38.43 ± 30.47*	Breast Cancer 569	*RBM+*	*93.51 ± 2.75*	*93.37 ± 2.88*	*93.31 ± 2.99*	*93.37 ± 2.88*
	KNN	90.93 ± 22.95	91.24 ± 21.75	89.63 ± 25.47	91.24 ± 21.75		*KNN*	99.76 ± 0.71	99.75 ± 0.74	99.74 ± 0.74	99.75 ± 0.74
	LR	**99.97 ± 0.10**	**99.97 ± 0.10**	**99.97 ± 0.10**	**99.97 ± 0.10**		*LR*	99.81 ± 0.05	99.81 ± 0.05	99.81 ± 0.05	99.81 ± 0.05
	MLP	99.92 ± 0.16	99.91 ± 0.17	99.91 ± 0.17	99.91 ± 0.17		*MLP*	98.39 ± 0.78	98.36 ± 0.92	98.35 ± 0.92	98.36 ± 0.92
	RAN	99.92 ± 0.04	79.50 ± 0.94	86.74 ± 0.66	79.54 ± 0.94		*RAN*	96.38 ± 0.78	92.45 ± 0.98	93.74 ± 0.95	92.45 ± 0.98
	SGD	98.91 ± 1.13	98.81 ± 1.40	98.59 ± 1.62	98.81 ± 1.40		*SGD*	**99.87 ± 0.12**	**99.86 ± 0.13**	**99.86 ± 0.13**	**99.83 ± 0.13**
IRIS	*RBM+*	*70.27 ± 5.17*	*70.83 ± 4.76*	*61.72 ± 6.48*	*70.83 ± 4.76*	Credit Approval	*RBM+*	*74.39 ± 14.8*	*74.86 ± 13.57*	*72.53 ± 16.2*	*74.86 ± 13.57*
	KNN	89.31 ± 29.40	91.47 ± 21.9	89.48 ± 27.49	91.47 ± 21.9		*KNN*	95.75 ± 0.55	95.73 ± 0.53	95.73 ± 0.53	95.73 ± 0.53
	LR	85.46 ± 3.34	78.14 ± 5.97	73.95 ± 8.10	78.14 ± 5.97		*LR*	95.38 ± 0.38	95.36 ± 0.37	95.36 ± 0.37	95.36 ± 0.37
	MLP	**97.22 ± 0.49**	**96.47 ± 1.19**	**96.54 ± 1.04**	**96.47 ± 1.19**		*MLP*	98.03 ± 1.62	97.99 ± 1.66	79.99 ± 1.67	97.99 ± 7.66
	RAN	93.64 ± 0.88	86.86 ± 2.99	88.06 ± 2.14	86.86 ± 2.29		*RAN*	89.43 ± 1.26	78.94 ± 2.89	82.40 ± 2.15	78.94 ± 2.89
	SGD	84.30 ± 11.55	85.91 ± 8.61	82.33 ± 11.36	85.91 ± 8.61		*SGD*	**99.76 ± 0.38**	**99.74 ± 0.43**	**99.73 ± 0.45**	**99.74 ± 0.43**



(A) Precision (B) Recall (C) F1-Score (D) Accuracy

FIGURE 4.10: The behavior of RANs modeling for seven datasets [SD:Sleep Data, CA:Credit Approval, BC2:Breast Cancer 569, MP:Mice Protein, TD:Toy-data, HAR:UCIHAR data, and ID:IRIS data]. Each graph shows an observed evaluation metric in nine Research Designs (RD) for all the datasets.

4.4 Conclusions

Concrete concepts (such as object table, or activity like walking, speaking, sitting, etc.) are, usually, identified by their intrinsic properties, and have been represented theoretically, mathematically and computationally. The notion of abstract concepts (such as love, freedom, and Is-active) are often debated but seldom explored. In humans, abstract concepts are of great importance, because, it capacitates us with relative comprehension and enables a contextual understanding of allied concepts. In an endeavor to computationally model, and represent the contextually similar concepts as abstract concepts, an approach, named Regulated Activation Network (RAN), is proposed with this chapter. The RANs modeling is hybrid (i.e., it is symbolic, distributed, and spatial), and exhibits an evolving topology with an unsupervised learning mechanism.

The RANs modeling consists of nine steps batched into three stages (Stage-1 learns and represents convex abstract concepts, Stage-2 for concept similarity relation learning, and Stage-3 for learning and representing non-convex abstract concepts), along with an optional method to associate labels with learned abstract concepts. To demonstrate the RANs methodology a two dimensional artificially generated data (Toy-data2) was used. The generated model observed a performance with Precision of 98.66% (ca.), Recall of 98.48% (ca.), F1-Score of 98.48% (ca.) and Accuracy of 98.48% (ca.), and was discovered to be competent with five classifiers, Multilayer Perceptron (MLP), Restricted Boltzmann Machine pipelined with Logistic Regression (RBM+), Logistic Regression (LR), K Nearest Neighbor (K-NN), and Stochastic Gradient Descent (SGD). While building the RAN model in this chapter three cognitive were also simulated depicting: 2 concept creation mechanisms; 2 activation propagation mechanisms; and 3 learning mechanisms. Another experiment with RANs used a benchmark data of Human Activity Recognition obtained from the UCI machine learning repository. The objective of this investigation was to automatically identify six activities (i.e., Walking, Walking_upstairs, Walking_downstairs, Sitting, Standing, and Laying) by two non-convex abstract concepts as *Mobility* (representing Walking, Walking_upstairs, Walking_downstairs), and *Immobility* (representing Sitting, Standing, and Laying). The results reflected that the model exhibited a satisfactory outcome with Precision of 96.97% (ca.), Recall of 96.57% (ca.), F1-Score of 96.53% (ca.) and Accuracy of 96.57% (ca.).

Additionally, another experiment was performed with RANs modeling using Sleep Detection data obtained from ISABELLA smartphone app of project SOCIALITE. With this study, a generic model was generated capturing Active and Inactive states of students. Further, a probe was made on the three student subjects by using afore generated RAN's model, and it was observed that one of the three students was mostly Inactive. This observation indicated the possibility of ill physical or psychological conditions of the student. Demonstration of RANs modeling with four datasets obtained from the UCI Machine Learning Repository puts forward

the proof of concept of RAN's classification capability. The results unfolded a notable feature of RANs methodology that enables efficient modeling regardless of the size of the training data size.

The concept similarity relation learning weights (CSRLW) introduced in this Chapter are used in the following Chapter 5 for simulating the recall operations.

Chapter 5

Cognitive Behavior Modeling with Regulated Activation Networks

This chapter describes the cognitive behavior simulation of RAN's convex concept modeling technique and is based upon the following International Conferences and Technical report, under review Journal article.

- This work received recognition from *Innovation Radar of European Union as one of the 5 "excellent innovations"* that resulted from European projects in the year 2017, `https://www.innoradar.eu/innovation/20221`.
- Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto and Amílcar F. Cardoso, Emulation of Cued Recall of Abstract Concepts Via Regulated Activation Networks. IEEE Access. Submitted June 2020.
- Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto and Amílcar F. Cardoso, Reconstructing Abstract Concepts and their Blends Via Computational Cognitive Modeling. International Joint Conference of Neural Network, July 2020.
- Alexandre Miguel Pinto, and Rahul Sharma. *Concept Learning, Recall, Blending with Regulated Activation Network.* International Conference on Cognitive Modeling (ICCM) 2016. (Pinto and Sharma, 2016)
- Alexandre Miguel Pinto, Rahul Sharma, *Regulated Activation Network*, deliverable D2.4 Project Concept Creation Technology (ConCreTe), Small or Medium-Scale Research Project ICT- Future and Emerging Technology (FET) `http://conceptcreationtechnology.eu/?q=node/43`. Technical Report, 2016.

5.1 Introduction

Concepts are an important part of investigations in cognitive and psychological research. Usually, the conceptual representations may be classified into process-oriented, symbolic or distributed, and knowledge-based (Kiefer and Pulvermüller, 2012, Bechtel et al., 1998). In general, a hierarchical structure defines an organization of concepts where the concrete concepts are placed in the lower level, and the abstract concepts occupy the higher levels. Therefore, abstract concepts are also seen as the generalization of concrete concepts (Rosch et al., 1976, Tversky and Hemenway, 1984). abstract concepts are studied mathematically (Saitta and Zucker, 1998), and psychologically (Borghi et al., 2018, 2017), but computational studies are scarce (Kiefer and Pulvermüller, 2012). This chapter uses RAN's convex concept modeling (Sharma et al., 2018b, 2017b) (also described in Chapter 3), and Intra-Layer Learning (discussed in non-convex abstract concept modeling in Chaptet 4) to perform recall simulations.

The prime aspect of this chapter is to emulate the recall procedure. Context plays an eminent role in the recall of concrete concepts (such as the word *table*). This recall operation of concrete concepts is often termed as concreteness effect. The concreteness effect is expressed through the dual-coding theory (Paivio, 1990, 1971) and context availability hypothesis (Schwanenflugel et al., 1992). The dual-coding theory discusses two independent representations: the verbal system and the imaginal system, corresponding to linguistic and non-verbal information processing. According to this theory, concrete concepts (concrete words) and abstract concepts (abstract words) are both defined by the verbal system. However, the imaginal system describes only concrete concepts. With concrete words, both the systems (i.e, verbal and imaginal) gets activated, providing relevant contextual information during the recall process, whereas recall of abstract concepts involves only one system therefore lacks in providing contextual information. According to the context availability theory, the concreteness effect is determined by prior knowledge and inciting stimuli. Individually, abstract concepts are difficult to understand in context retrieval when compared with concrete concepts, which also complicates their recall procedure. An interesting work discovered that, when response pairs of abstract concepts are relevantly related to one another, they provide context for the abstract stimuli (Bransford and McCarrell, 1974). Hence, with an adequate context relation defined among the abstract concepts, their recall can be realized. This work proposes an approach that uses the learned associations (CSRL process described in Chapter 4) among the abstract concepts to simulate a regulated recall operation.

Computational models are useful in understanding the psychological and cognitive phenomena, validating the existing cognitive theories, besides helping to formulate fresh ideas related to cognition (Rolls et al., 2008, Kyaga et al., 2013, Braver et al., 1999, O'Reilly, 2006). The representations produced by computational approaches are amodal (symbolic), multimodal (distributed), or hybrid (Hayes and Kraemer, 2017). Adaptive Control of Thought-Rational

(ACT-R) (Anderson et al., 1997) is a symbolic architecture intended to model memory (Lovett et al., 2000), simulate attention (Anderson et al., 2004, 1997, 2004), decision making (Marewski and Mehlhorn, 2011), recognition (Schooler and Hertwig, 2005), and forgetting (Schooler and Hertwig, 2005). Multimodal approaches such as artificial neural networks (ANN), including Restricted-Boltzmann Machine (RBM) (Hinton, 2012), Deep Neural Networks (Collobert and Weston, 2008) stacked auto-encoders (Vincent et al., 2010), and Convolution Neural Networks (CNN) (Krizhevsky et al., 2017), have significantly contributed in feature recognition (Dara and Tumma, 2018) and distributed representation (Hinton et al., 1986). Besides, hybrid cognitive architectures like Connectionist Learning with Adaptive Rule Induction On-line (CLARION) (Sun and Peterson, 1996) simulated scenarios related to cognitive and social psychology.

This chapter utilizes the RAN's hybrid nature and modeling to build representations of convex abstract concepts and further simulate recall of abstract concepts. The model generation takes place with four basic steps of RANs approach (Sharma et al., 2018b), i.e., convex concept Identification, convex abstract concept creation, convex concept inter-layer learning, and convex abstract concept upward activation propagation (as described in Chapter 3). An Intra-layer procedure also takes place at all the layers to learn associations among the concepts at the same layer (described as CSRL in Section 4.2.2 of Chapter 4). Then, these learned associations are uniquely interpreted to determine whether the impact of their learned weights is inhibitory, excitatory, or neutral. Finally, these impacts are applied to obtain a regulatory effect on peer concepts (abstract concepts, or input layer concepts) during the recall operation. A Toy-data problem (Toy-data3) is used for modeling with RANs and demonstrating the novel Geometric Back-Propagation Algorithm for simulation cued recall operation. A benchmark dataset of image domain, MNIST, is also used to demonstrate the cued recall experiment. These experiments also show how blends of abstract concepts can be recalled.

The remaining of this chapter is organized in the following way: Section 5.2 puts forward the state of the art related to recall operations; the RANs modeling, the Intra-Layer Regulation algorithm, and Geometric Back-propagation algorithm are detailed in Section 5.3 using a Toy-data problem; the cued recall demonstration with MNIST dataset is reported in Section 5.8; Section 5.9 concludes the chapter.

5.2 Advances in Recall Research

Recall or retrieval is a cognitive process (Buzsáki, 2010) of remembering a thing or an event. While recalling, the brain activates a neural assembly that was created when the original event occurred (Buzsáki, 2010). In psychology, there are a plethora of articles studying the recall process. Psychologists used free-recall, cued-recall, and serial-recall as tools to investigate memory processes (Bower, 2000). The recall has been used to study the effect of cognitive strategies

such as chunking, in the use of mnemonics for memorization of things (such as large numbers) (Bermingham et al., 2013). A research reported the benefits of subsequent recall operations where memories are related or competing (Rafidi et al., 2018). The proverb "Practice makes a man perfect" relates to the fortification of memory, and an investigation has shown how retrieval plays an important role in this process (Antony et al., 2017). Technologies such as fMRI [1], MRI [2], PET [3], and EEG [4] were used in validating several recall related hypotheses (Polyn et al., 2005, Hulbert and Norman, 2014, Kapur et al., 1995, Rafidi et al., 2018).

Notable contributions to model the memory recall procedures are observed in literature. Based upon the Temporal Context Model (Sederberg et al., 2008, Howard and Kahana, 2002) of human behavior, human memory performance was modeled using a probabilistic approach during free-recall experiments (Socher et al., 2009). A computational model of interaction of Prefrontal Cortex and medial temporal lobe in memory usage was designed to study the Prefrontal Control in a recall process (Becker and Lim, 2003). The model was a simple neural network with quick and flexible reinforcement learning exhibiting strategic recall. Another computational model differentiates the recall from the recognition process depending upon the number of cues involved in the retrieval procedure (Srivastava and Vul, 2017). For encoding, the model used an inference-based model of memory (Gillund and Shiffrin, 1984), and retrieval was carried out using a Bayesian observer model (Wei and Stocker, 2015). A large number of computational psychology contributions studying the recall process along with the recognition, using the neural networks, is available (Ruppin and Yeshurun, 1990, Biggs and Nuttall, 2015, Ruppin and Yeshurun, 1991, Recanatesi et al., 2015)

An interesting work simulated the free-recall process using the ACT-R architecture (Taatgen, 1996), showing that the classical effect of primacy and recency can be recreated through rehearsal theory based upon ACT-R and Baddeley's phonological Loop (Baddeley, 1992). ACT-R was also used to propose a new theory of memory retrieval to predict intricate serial and free recall operations (Thomson et al., 2015). This research also focused on the prospects of associative learning by introducing a strengthening and decaying mechanism depending upon the similarity of the input stimulus. The serial recall has been modeled in a scientific contribution using the same architecture to explain the processes involved while recalling a list of words (Anderson and Matessa, 1997). The traditional ACT-R recall operations had a limitation: here the memory access depends upon limiting the capacity of the activation process, consequently inducing errors in the contents being recalled. This theory overcomes such limitations by predicting the latency and errors in a serial recall process.

The free recall process was also modeled using the CLARION architecture to determine the role of distractions in an incubation task (Hélie et al., 2008). This study made a striking observation

[1] Functional Magnetic Resonance Imaging
[2] Magnetic Resonance Imaging
[3] Positron-emission tomography
[4] Electroencephalography

that rehearsals play an important role in memory consolidation during free recall procedure, and distractions can hinder the free recall and eventually effect memory strengthening. CLARION was also used to emulate, acquire, and expound human-centric data relevant to incubation and insight through free recall, lexical decision, and problem-solving tasks (Hélie and Sun, 2010).

In our work, a Geometric Back-propagation mechanism is added to the RANs methodology, to simulate the recall of learned Abstract Concepts. The Abstract Concepts are of great value in our lives, as they help us in attaining abilities such as relative comprehension, effective decision-making etc. Most of the investigations related to Abstract Concept came from psychological research, such as a study related to the role of emotional contents in processing and representing Abstract Concepts (Kousta et al., 2011). While studying languages, the cognitive linguists and psychologists made a considerable amount of effort to investigate about the representations and modeling of Abstract Concepts, like the internal representation of Abstract Concepts via amodal symbols (Barsalou and Wiemer-Hastings, 2005). Abstract words were also studied for their association and context in comparison with Concrete words (Barsalou and Wiemer-Hastings, 2005). An interesting work showed that the metaphors are mostly represented as Abstract Concepts (Gibbs Jr, 1996). Besides psychological research, biomedical research has also contributed to studies related to Abstract Concepts, such as identifying brain regions involved in the processing of Abstract Concept with the help of MRI scans (Binder et al., 2005).

As mentioned earlier, Abstract Concepts are complex[5] hence are difficult to understand when compared with Concrete Concepts. Therefore, in computational modeling realm, Concrete Concepts are mainly addressed by feature recognition techniques (Ji et al., 2013, Ripley, 2007, LeCun et al., 2015, Anderson, 1996). However, there are a few computational approaches to the modeling of Abstract Concepts, besides our own. Natural Language Processing research reports some work in Abstract Concept representation, as a text-based multimodal architecture to study the Abstract and Concrete representations of the language used in daily life (Hill and Korhonen, 2014). Semantic similarity among the Abstract and Concrete nouns (in Greek, and English) were also studied using semantic network-based Distributed Semantic Model (Iosif et al., 2013, Iosif, 2013a). In the image processing domain, an interesting work contributed with a technique to build an Abstract feature of face using deep sparse auto-encoder (Le, 2013).

The method we propose uses the RANs modeling approach to build representations of Abstract Concepts. We also describe how the learned associations among the Abstract Concepts deduce context, which is very useful in the recall process. As stated before, to simulate the recall procedure we propose an algorithm that propagated activation from higher layer Abstract nodes to input layer nodes. While simulating the recall operation, we also demonstrate how learned association among the Abstract Concepts helps in excitation, inhibition, and regulation of activation at nodes.

[5] According to Dual coding theory (mentioned in Section 5.1) Abstract Concepts are only represented by verbal systems and makes it difficult deduce contextual information for this reason they are complex.

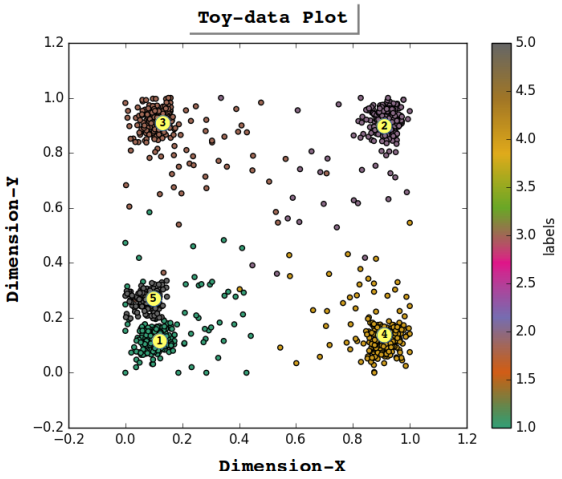


FIGURE 5.1: Graph of 2-D Toy-data3 with the five cluster, along with their respective cluster centers [1, ..., 5].

5.3 RANs Methodology to Simulate Recall Operation

Here we describe the emulation of the recall process using RANs modeling as described in Chapter 3. At first, for background understanding, the RAN's convex abstract concept modeling is described along with two learning mechanisms, i.e., inter-layer and intra-layer learning. Having explained the RANs modeling, the two contributions of this chapter are described: first, we present and discuss the Regulation mechanism; secondly, a novel Geometric Back-propagation Algorithm is proposed that propagates activations from abstract level to input level. The RANs methodology and the proposed algorithm are illustrated using a Toy-data3 dataset. At the end of the section, the experiments of RANs modeling with the Toy-data are also reported, demonstrating the recall operation.

5.4 Modeling with RANs

This section is dedicated to describing convex abstract concept modeling with RANs. For demonstrating the RANs methodology, a Toy-data problem (Toy-data3) is used (see Figure 5.1). The Toy-data consists of 1800 data instances randomly distributed into five categories. Out of the five clusters, three are far apart from one another; the other two clusters are very close to each other. This arrangement of clusters was introduced into the Toy-data problem to demonstrate the Excitatory and Inhibitory impact of concepts, representing each cluster at an abstract level. The modeling is performed using the four basic steps (as explained in Chapter 3), where the Step-1b and Step-4b are similarity relation learning (explained in Chapter 4) operations at different layers.

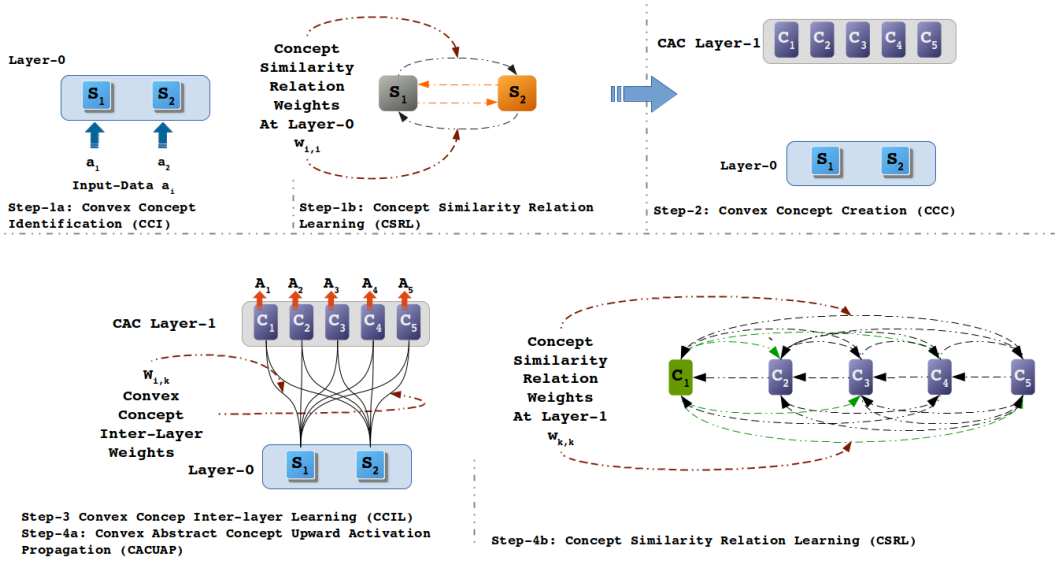


FIGURE 5.2: RANs convex abstract concept modeling process. The procedure displays the four steps in RANs modeling. This figure shows the three learning procedures, i.e., two Similarity Relation Learning at two layers, and one inter-layer learning between the Layer-1 and Layer-0. In Step 1 the Similarity Relation Learning (Step 1-a) is performed along with concept identification Process (Step 1-b). Similarly, in Step-4 the Similarity Relating Learning (Step 4-b) is performed together with the Upward Activation Propagation method (Step 4-a).

5.4.1 Step-1a: Convex Concept Identification (CCI) process

In this operation, the cluster centers are determined for the inter-layer learning operation (see Section 5.4.4). Step-1a in Figure 5.2 shows the input Layer-0, where nodes S_1 and S_2 correspond to the dimensions of the input Toy-data3. To identify the five Convex groups the K-mean (Hartigan and Wong, 1979) clustering algorithm is chosen as concept identifier (CI), and the value of K is set to 5 to determine five clusters. Five cluster centers are also identified in this process, as shown in Figure 5.1, as Cluster Representative Data Points (CRPD).

5.4.2 Step-1b and Step-4b Concept Similarity Relation Learning (CSRL)

The CSRL is an intra-layer operation as described in Section 4.2.2 of Chapter 4. This learning mechanism is performed two times while modeling with Toy-data3, First, in Step-1b at the input Layer-0, (see Figure 5.2 Step-1b). Second, when the input data is propagated upward to convex concept Layer-1. The learning at Layer-0 was of size 2×2 as the input layer has two nodes, whereas, the learning at Layer-1 was of size 5×5 (refer Figure 5.4 for the CSRL weights).

5.4.3 Step-2: Convex Abstract Concept Creation (CACC) operation

The convex abstract concept creation operation is the method described in Section 5.4.1 of Chapter 3 to create a new layer dynamically. The Step-2 in Figure 5.2 depicts the creation of a new convex abstract concept (CAC) Layer-1 having five nodes (C_1, ..., C_5). These Five nodes (C_1, C_2, C_3, C_4 and C_5) represent Clusters 3, 1, 4, 2 and 5 of Figure 5.1, respectively. The count of the CAC layer nodes depends upon the number of clusters identified in CCI operation at the input layer, i.e., if k clusters were determined in CCI mechanism then in Step-2 a new layer is created consisting of k nodes.

5.4.4 Step-3: Convex Concept Inter-Layer Weight (CCILW) Assignment

The inter-layer learning is the second learning mechanism in RANs modeling to learn association among the nodes at CAC layer and input layer as described in Chapter 3 . In the experiment with Toy-data3, a 5×2 weight matrix was learned between the five nodes at CAC layer-1 and two nodes of input layer-0, as shown in Figure 5.2 Step-3. Having completed the Step-3 a basic RAN's model is obtained consisting of input layer-0, the CAC layer-1, learning between two layers and learning among the nodes at input layer-0.

5.4.5 Step-4a: Convex Abstract Concept Upward Activation Propagation (CACUAP)

This step is used to propagate i-dimensional (i.e. 2-D) input data vector a to CAC layer and obtaining a k-dimensional (5-D) data vector A. In order to build more than one layer, all the steps are to be repeated iteratively using Algorithm 2 described in Chapter 3.

Upon propagating all input values to the CAC layer, the observed outputs A are used to perform the concept similarity relation learning (CSRL) process (described in Section 4.2.2 of Chapter 4), as shown in Figure 5.2 at Step-4b. After completing the Step-4b, the RANs modeling procedure terminates, and a model is obtained as shown in Step-3 of Figure 5.2.

5.5 Regulation Mechanism

The Regulation operation in RANs modeling is performed in three steps. First, an Impact Factor of CSRL matrix is computed. Second, we determine the intra-layer (IL) contribution of activation at a node by another node in the same layer. Third, we obtain activation at a node by a function of self-activation and intra-layer activation induced by other nodes on the latter.

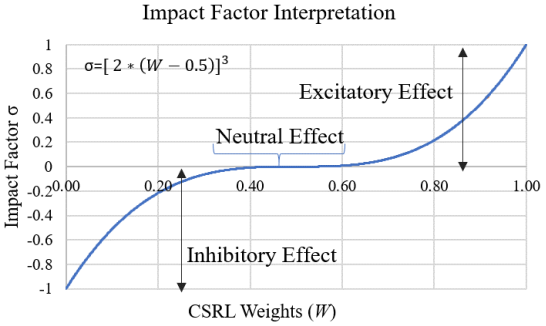


FIGURE 5.3: Excitatory, Inhibitory and Neutral effect of CSRL weights (W), when transformed using the Pinto's Effect (or Impact-Factor) σ.

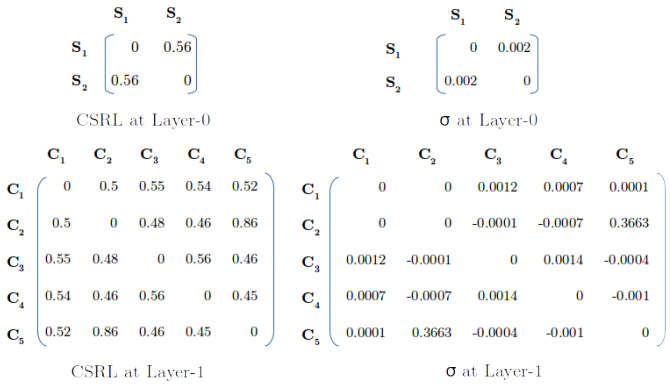


FIGURE 5.4: The CSRL weight matrices learned with Toy-data and their corresponding Impact Factor (σ) at Layer-0 and Layer-1. The σ is calculated using Equation 5.1.

5.5.1 Impact Factor (σ) Construction and Interpretation

This Impact Factor (or *Pinto's Effect*) is a function that interprets the CSRL weight values (in the range [0, 1]) as Excitatory, Inhibitory or Neutral weights. The purpose of CSRL weights was to determine how two nodes (for eg. S_1 and S_2) were concurrently active. If the CSRL weight is intermediate, i.e. 0.5, it signifies that the two nodes were 50% concurrently active (depicting a state of confusion). Therefore, these nodes do not have an impact over each other in the same layer. If the CSRL weights of the two nodes were '0', it signifies the two nodes were never simultaneously active. This also indicates that the two nodes were Inhibitors of one another. At last, if the CSRL weights of the two nodes is '1', this means that the nodes are always conjointly active, showing that they also Excite each others activation.

$$\sigma_{m \to n} = [2 * (W_{m \to n} - 0.5)]^3 \tag{5.1}$$

The aforementioned comprehension of CSRL weights W is exhibited by a mathematical Equation 5.1, where $\sigma_{m \to n}$ is the impact of node m over node n. Figure 5.3 shows the graphical view of the Impact-Factor σ (Equation 5.1), depicting the Excitatory, Inhibitory or Neutral interpretations of CSRL weights. Figure 5.4 shows the CSRL and their respective σ weights

for both the layers. At Layer-0 the nodes S_1 and S_2 have a very minimal Excitatory impact on each other (in Figure 5.1 every node at Layer-1 can be related to clusters C_1, .., C_5 serially). However, at Layer-1 node C_1 has no impact over node C_2 (and vice versa). There are many negative weights in the σ matrix of Layer-1 indicating that these nodes inhibit each other. In Figure 5.1 we see that the clusters C_2 and C_5 are very close, and the activations observed at both the nodes must be very similar. Hence, high CSRL weight is learned between node C_2 and C_5. It is also interesting to observe that both exhibit good excitatory behavior over one another.

5.5.2 Intra-Layer Activation

The objective of calculating intra-layer (IL) Activation is to determine the amount of activation a node n receives from all the other m nodes of the same layer. To obtain the intra-layer activation at node n the approach must address three prospects. First, the intra-layer activation must consider the impact (σ) of Excitatory, Inhibitory or Neutral effects of all m nodes over node n. Second, the current activation of m nodes and their CSRL weight (W$m \rightarrow n$) to node n should be considered in calculating the activation of node n. Third, the intra-layer activation computed for node n must be in the range [0, 1]. Equation 5.4 conform to all the three requirements.

$$\chi_m = (a_m * W_{m \rightarrow n}) \tag{5.2}$$

$$\neg\chi_m = (1 - a_m) * (1 - W_{m \rightarrow n}) \tag{5.3}$$

$$IL(a_n) = \frac{\sum\limits_{m} \sigma_{m \rightarrow n}(\chi_m + \neg\chi_m)}{\sum\limits_{m} \sigma_{m \rightarrow n}} \tag{5.4}$$

5.5.3 Intra-Layer Regulation

To find the Actual Activation (AA) (a_n) at node n, we use a Regulation Factor (ρ) to decide the share of self-contribution of activation by node n and intra-layer Activation at node n, i.e. IL (a_n). Equation 5.5 shows the mathematical function for the intra-layer regulation operation. From Equation 5.5 we can observe that if ρ is '0', i.e. there is no Regulation, only the activation of node n contributes to the Actual Activation.

$$AA(a_n) = (1 - \rho) * a_n + \rho * IL(a_n) \tag{5.5}$$

Algorithm 4: Intra-Layer Regulation

Input: Current Activation a_n at node n at layer L.
Input: CSRL W at layer L
Input: Impact Matrix σ at layer L
Initialization: The Regulation Factor ρ, between [0, 1]
foreach a_n *in* L **do**
    Calculate IL(a_n), using Equation 5.4
    Calculate Actual Activation AA(a_n), using Equation 5.5
return AA(a_n)

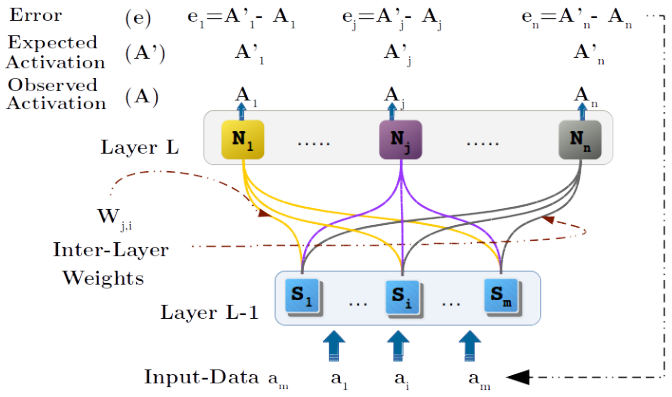


FIGURE 5.5: Single window operation in Geometric Back-propagation operation. Figure also shows the Error calculation and propagation.

Algorithm 4 presents the Intra-Layer Regulation operation in an algorithmic form. This Regulation operation has its importance when propagating the activation from an abstract concept layer to the input layer, as described in Section 5.6.

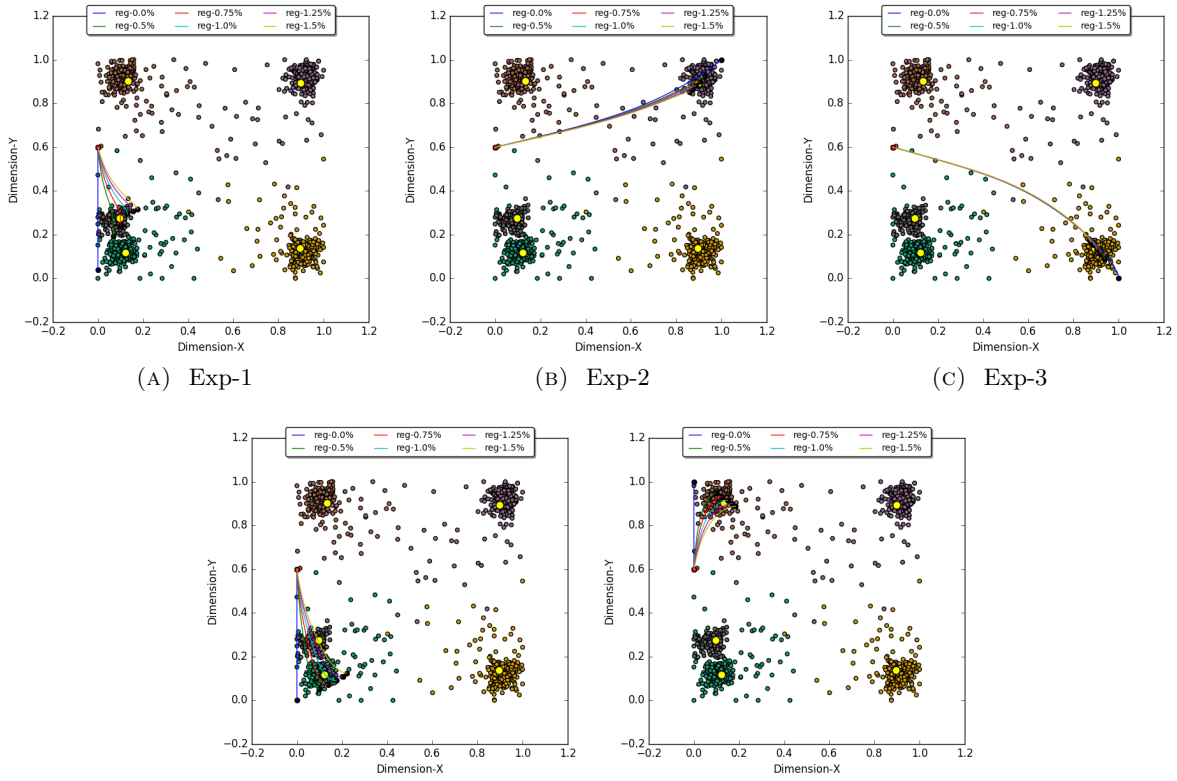
5.6 Geometric Back-propagation (GBP) Operation

The Geometric Back Propagation (GBP) is a downward propagation mechanism in RANs modeling. This method enable us to determine an activation vector a_m ($< a_1, .., a_i, .., a_m >$) at layer *L-1*, for an Expected Activation (E-A) vector A'_n ($< A'_1, .., A'_j, .., A'_n >$) at layer L. This is a window operation that takes place between two adjacent layers, i.e. layer L and layer *L-1*. For instance, if the RAN's model has three layers L_0 (Input layer), L_1 and L_2 (Output layer), then two GBP operations take place, first between L_2 and L_1, then between L_1 and L_0. Figure 5.5 shows the single window operation between two layers.

The GBP mechanism commences with an Expected Activation (A'_n) vector at layer L. Next, a starting input vector a_m ($< a_1, .., a_i, .., a_m >$) is injected in layer *L-1*. Now we enter into a cycle where we propagate the activation of the nodes in layer *L-1* upwards to layer L and determine the Observed Activation (O-A) vector A_n ($< A_1, .., A_j, .., A_n >$) at layer L. Afterwards, an error vector e is calculated using A'_n and A_n (Expected and Observed Activation vector) through Equation 5.6. The error vector e is used to determine an accumulated delta value $\triangle_{a_i}$ (see

Algorithm 5: Geometric Back-Propagation Operation

Input: A *ExpectActivation* activation A'_n ($< A'_1, .., A'_j, .., A'_n >$) at layer L, with n nodes
Input: Desired Maximum Iteration *maxIter*
Output: An activation pattern a_m ($< a_1, .., a_i, .., a_m >$) at layer *L-1*, with m nodes
Set Regulation Factor ρ between [0, 1]
Set *currentActivation* = *Starting-Point* (a vector of activation $< a_1, \ldots, a_m >$)
Set *previousActivation* = *currentActivation*
Set *PropagateActivation*= CCUAP of *currentActivaiton* to layer L (see Section 5.4.5)
Set *ObservedlActivation* (A)= Regulate *PropagateActivation* via Algorithm 4
Calculate error vector (e) at layer L using Equation 5.6
Set *iter* = 0
repeat
  Set *iter* = *iter* + 1
  foreach a_i *in previousActivation* **do**
    Calculate the delta ($\triangle_{a_i,A_j}$) using Equation 5.7
    Calculate the sum of delta for a_i, i.e. $\triangle_{a_i}$, using Equation 5.8
    if $\triangle_{a_i} > 0$ **then**
      $a_{temp} = a_i + \triangle_{a_i} * (1 - a_i)$
      if $a_{temp} > 1$ **then**
        Assign $a_i = 1$
      else if $a_{temp} < 0$ **then**
        Assign $a_i^{new} = 0$
      else
        Assign $a_i^{new} = a_{temp}$
    else
      $a_{temp} = a_i + \triangle_{a_i} * (a_i)$
      if $a_{temp} > 1$ **then**
        Assign $a_i^{new} = 1$
      else if $a_{temp} < 0$ **then**
        Assign $a_i^{new} = 0$
      else
        Assign $a_i^{new} = a_{temp}$
  Set *currentActivation* = $< a_1^{new}, .., a_i^{new}, .., a_m^{new} >$ (new activation vector at layer *L-1*)
  Set *previousActivation* = *currentActivation*
  Set *PropagateActivation*= CCUAP of *currentActivaiton* to layer L (see Section 5.4.5)
  Set *ObservedlActivation* (A)= Regulate *PropagateActivation* via Algorithm 4
  Calculate error vector (e) at layer L using Equation 5.6
until *iter* = *maxIter*;
return *currentActivation*



(A) Exp-1 (B) Exp-2 (C) Exp-3

(D) Exp-4 (E) Exp-5

FIGURE 5.6: The trajectories of activation observed at input Layer-0 while carrying out a thousand iterations of the GBP algorithm. The Red circle is the starting point of trajectory and the Black circle is the activation value after the thousandth iteration. The graphs also depict the trajectories observed at input Layer-0 with six regulation factors ρ (0%, 0.5%, 0.75%, 1%, 1.25% and 1.5%). Each graph visualizes the recalled activation for five SCR experiments.

Equation 5.8) based upon the function expressed by Equation 5.7. This $\triangle_{a_i}$ value is then added to the activation vector at layer *L-1* (See Equation 5.9) to obtain a new input vector a_m^{new}. The cycle is repeated with the new a_m^{new} input at layer *L-1* until the error is minimized, or the cycle equals the user-defined maximum iteration threshold. Algorithm 5 provides the detailed Geometric Back-propagation Algorithm. As mentioned earlier, the GBP operation takes place between two consecutive layers. However, if the hierarchy has more than two layers, then with the window operation it is possible to propagate down the injected E-As at the nodes of the top-most layer-L, to the Input layer-0.

$$e_j = A'_j - A_j \tag{5.6}$$

$$\triangle_{a_i,A_j} = (W_{j,i} - a_i) * (e_j) \tag{5.7}$$

$$\triangle_{a_i} = \sum_{j=1,..,i} \triangle_{a_i,A_j} \tag{5.8}$$

$$a_i^{new} = a_i + \triangle_{a_i} \tag{5.9}$$

5.7 Recall Demonstration with Toy-data

There are two types of experiments performed in this section. First, the Single Cue Recall (SCR) operation, where the recall is performed based upon the Expected Activation by one node in an abstract concept layer. Second, Multiple Cue Recall (MCR) mechanism; here, the recall procedure is carried out for the Expected Activation at all the nodes in the abstract concept layer. The experiments demonstrated in this section use the two-layered model generated with RANs methodology (refer the two-layered model obtained at Step-3 in Figure 5.2 in Section 5.4). In the model, the five abstract concept nodes (C_1, C_2, C_3, C_4 and C_5) correspond to Clusters 3, 1, 4, 2 and 5 respectively (see Figure 5.1 for the clusters). In both SCR and MCR operation, the five abstract concept nodes at Layer-1 will be injected with an Expected Activation set. Further, the Geometric Back-propagation Algorithm (Algorithm 5) shall perform a thousand iterations of downward propagation of activation, to obtain appropriate values at input Layer-0 as recalled activation. In all the recall simulations, the GBP operation is initialized with values [0, 0.60] as *Starting-Point* and *maxIter* is set for 1000 iterations. The Expected Activation varies with the experiment, and two sets of the regulation factor were determined empirically to demonstrate both recall procedures.

5.7.1 Single Cue Recall (SCR) Experiment

In SCR experiments, the objective was to determine the recalled activation at the input Layer-0, by injecting binary activation values as Expected Activation in abstract concept Layer-1. The Expected Activation vector contains value 1 for only one abstract concept node and for the remaining nodes 0 is assigned. In all SCR experiments, six regulation factors ρ (0%, 0.5%, 0.75%, 1%, 1.25% and 1.5%) were used. Table 5.1 logs the E-A and the thousandth iteration value of O-A for the five SCR experiments. We describe next the five experiments to show the SCR operation along with the observations:

- **Exp-1:** This is the first experiment where we injected an E-A vector *[0, 0, 0, 0, 1]* of activation at abstract concept Layer-1. The objective is to recall activations at input Layer-0 for which a very high activation is observed at node C_5 at Layer-1 and comparatively lower activation for other four nodes in Layer-1. The GBP algorithm is executed six times with an E-A of *[0, 0, 0, 0, 1]* for the six different regulation factors (ρ). The observation for Exp-1 (see Table 5.1) shows that with a ρ of 0.75% the maximum activation of 0.85 was observed at node C_5. It is also observed that node C_2 observed a good activation which was expected as nodes C_2 and C_5 represents the cluster 1 and 5 (see Figure 5.1) which are close to one another. Figure 5.6a shows the six trajectories for six regulation factors, each trajectory is formed by one thousand iterations. In Figure 5.6a the yellow marker shows the CRDP of the cluster C_5, and the trajectory with ρ of 0.75% converge closest to

TABLE 5.1: Observations of Activations at Abstract Concept Layer-1 for SCR Experiments

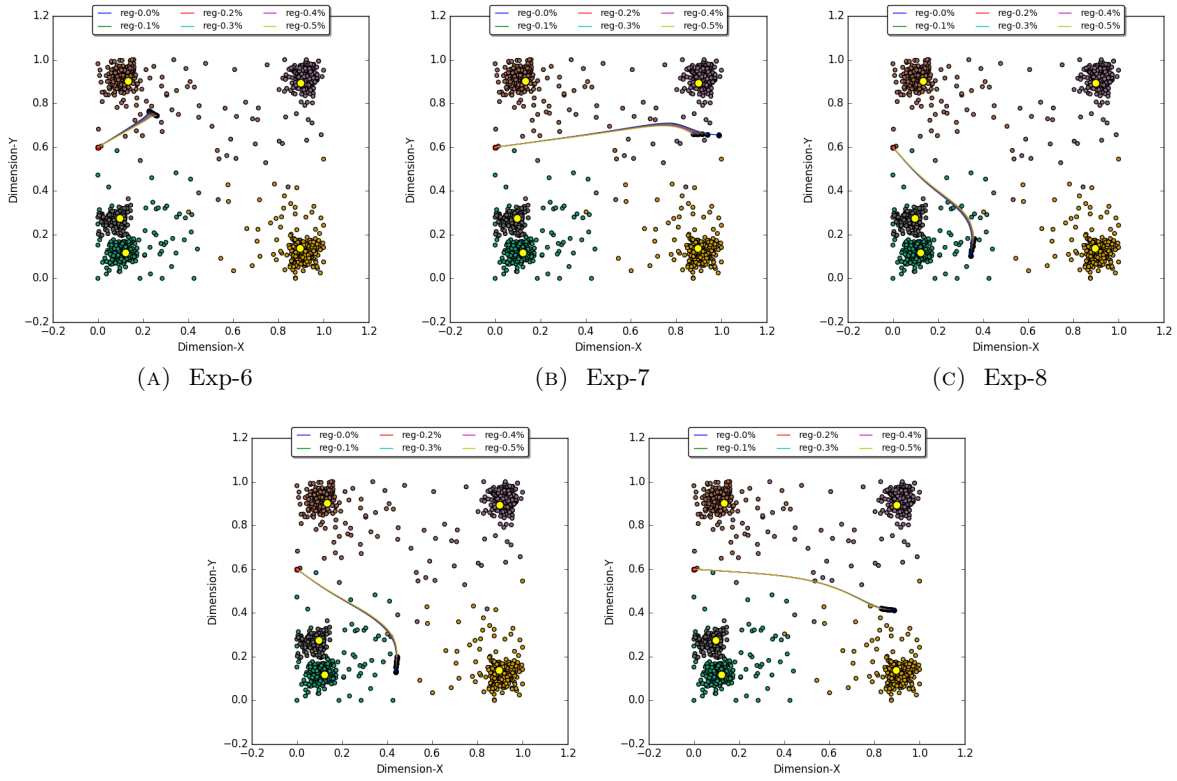
Regulation	**Experiment**	**E-A at Layer-1**					**O-A at Layer-1**				
%	————	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5
0	Exp-1	0	0	0	0	1	0.08	0.61	0.07	0.01	0.47
0.5	Exp-1	0	0	0	0	1	0.14	0.70	0.09	0.03	0.72
0.75	**Exp-1**	0	0	0	0	**1**	0.16	**0.64**	0.10	0.04	**0.85**
1	Exp-1	0	0	0	0	1	0.19	0.56	0.11	0.05	0.82
1.25	Exp-1	0	0	0	0	1	0.19	0.54	0.12	0.06	0.75
1.5	Exp-1	0	0	0	0	1	0.20	0.52	0.12	0.06	0.71
0	Exp-2	0	0	0	1	0	0.08	0.01	0.08	0.61	0.02
0.5	Exp-2	0	0	0	1	0	0.10	0.02	0.11	0.83	0.03
0.75	**Exp-2**	0	0	0	**1**	0	0.11	0.02	0.11	**0.93**	0.04
1	Exp-2	0	0	0	1	0	0.12	0.03	0.12	0.90	0.04
1.25	Exp-2	0	0	0	1	0	0.12	0.03	0.13	0.83	0.05
1.5	Exp-2	0	0	0	1	0	0.13	0.03	0.13	0.78	0.05
0	Exp-3	0	0	1	0	0	0.01	0.07	0.57	0.07	0.06
0.5	Exp-3	0	0	1	0	0	0.02	0.09	0.76	0.10	0.08
0.75	**Exp-3**	0	0	**1**	0	0	0.02	0.10	**0.84**	0.11	0.09
1	Exp-3	0	0	1	0	0	0.03	0.11	0.93	0.11	0.09
1.25	Exp-3	0	0	1	0	0	0.03	0.11	0.91	0.12	0.10
1.5	Exp-3	0	0	1	0	0	0.03	0.12	0.84	0.13	0.10
0	Exp-4	0	1	0	0	0	0.06	0.57	0.07	0.01	0.42
0.5	Exp-4	0	1	0	0	0	0.08	0.75	0.10	0.02	0.50
0.75	**Exp-4**	0	**1**	0	0	0	0.09	**0.80**	0.12	0.02	**0.52**
1	Exp-4	0	1	0	0	0	0.10	0.77	0.13	0.03	0.53
1.25	Exp-4	0	1	0	0	0	0.10	0.73	0.14	0.03	0.54
1.5	Exp-4	0	1	0	0	0	0.11	0.69	0.15	0.04	0.54
0	Exp-5	1	0	0	0	0	0.58	0.07	0.01	0.07	0.13
0.5	Exp-5	1	0	0	0	0	0.78	0.09	0.02	0.10	0.15
0.75	Exp-5	1	0	0	0	0	0.85	0.10	0.02	0.11	0.16
1	**Exp-5**	**1**	0	0	0	0	**0.88**	0.10	0.03	0.12	0.17
1.25	Exp-5	1	0	0	0	0	0.84	0.11	0.03	0.13	0.17
1.5	Exp-5	1	0	0	0	0	0.79	0.11	0.04	0.13	0.18

E-A [Expected Activation], O-A [Observed Activation]

this CRPD. Thus, an activation vector *[0.1, 0.24]* is recalled at input nodes [S_1, S_2] for the given E-A vector *[0, 0, 0, 0, 1]*.

- **Exp-2:** In this experiment, the E-A provided to the GBP algorithm was *[0, 0, 0, 1, 0]* to recall activation at Layer-0 that is strongly represented by node C_4. For each regulation factor, the GBP algorithm was run, the O-A obtained at Layer-1 are listed in Table 5.1 and the corresponding recalled activation at Layer-0 is shown in Figure 5.6b. From the observations, it can be deduced that the experiment with ρ of 0.75% produced the best outcome and recalling activation *[0.9, 0.9]* for the input Layer-0.

- **Exp-3:** In this experiment, the GBP algorithm was supplied with an E-A vector of *[0, 0, 1, 0, 0]* to recall input Layer-0 vector that is represented by node C_3 at Layer-1. The



(A) Exp-6 (B) Exp-7 (C) Exp-8

(D) Exp-9 (E) Exp-10

FIGURE 5.7: The trajectories of activation observed at input Layer-0 while carrying out a thousand iterations of the GBP algorithm. The Red circle is the starting point of trajectory and the Black circle is the activation value after the thousandth iteration. The graphs also depict the trajectories observed at input Layer-0 with six regulation factors ρ (0%, 0.1%, 0.2%, 0.3%, 0.4% and 0.5%). Each graph visualizes the recalled activation for five MCR experiments.

Figure 5.6c and Table 5.1 shows the observation of the recall trajectories at Layer-0 and O-A vector at Layer-1, respectively. The experiment with a regulation of 0.75% displayed the best representation. A vector *[0.92, 0.11]* was recalled at Layer-0 for the injected E-A vector.

- **Exp-4:** This experiment aims to recall an input vector that closely represents the abstract concept node C_2 by feeding the GBP algorithm with an E-A vector of *[0, 1, 0, 0, 0]*. After applying the six regulation factors to each GBP operation, it was observed that the experiment with ρ of 0.75% displayed the best result. Table 5.1 the O-A for the E-A. Figure 5.6d shows the trajectories of the recalled values and the best outcome with ρ of 0.75% that converges to an activation vector *[0.14, 0.07]*.

- **Exp-5:** This experiment shows the recall vector obtained by initializing an E-A vector *[1, 0, 0, 0, 0]* with all GBP experiment with six regulation values. Unlike the previous experiments, the best O-A was obtained with a regulation factor of 1%. Figure 5.6d shows the recall outcome for all six regulation factors. At input Layer-0 the recall operation with ρ of 1% results into an activation vector *[0.15, 0.91]* for the given E-A.

5.7.2 Multiple Cue Recall (MCR) Experiment

The MCR experiments were carried out to determine the recall vector at input Layer-0 for an E-A vector at Layer-1. The constituents of the E-A vector are Degree of Confidence (DoC) values that define the expected representation of each abstract concept node at Layer-1. To demonstrate MCR, five experiments were performed, and in every experiment six regulation factors ρ (0%, 0.1%, 0.2%, 0.3%, 0.4% and 0.5%) were used to make inferences. Table 5.2 lists the observations of O-A for their respective E-A in each MCR experiment. Figure 5.7 displays the trajectories of recalled activation in Layer-0 for six regulation factors with respect to each experiment. The E-A vectors used in the MCR experiments are vectors obtained by propagating an input activation from Layer-0 to Layer-1. Hence, in MCR simulation we also have an Expected Recall (E-R) vector to perform the evaluation. Next we present the MCR experiment descriptions along with their observations:

- **Exp-6:** In this experiment an E-A vector of *[0.57, 0.16, 0.06, 0.15, 0.25]* is provided to the GBP algorithm. With this E-A we want to recall activation at input Layer-0 that is 57%, 16%, 06%, 15% and 25% represented by nodes C_1, C_2, C_3, C_4 and C_5, respectively. Table 5.2 lists the O-A observed for the six regulation factors, the results with ρ of 0% and 0.1% shows the outcome which is almost identical to E-A. The E-R vector of this experiment was *[0.2256, 0.7610]* and with ρ of 0% and 0.1% the observed recall was *[0.2260781, 0.7647118]* and *[0.2343844, 0.7602842]*, respectively, which are also similar to the Expected Recall vector. Figure 5.7a shows the trajectories of all the recalled activation vectors at Layer-0 for the E-A vector w.r.t. their six regulation factors.

- **Exp-7:** For this experiment the GBP algorithm was injected with an E-A vector of *[0.07, 0.04, 0.23, 0.46, 0.05]* for recalling an activation vector at the input Layer-0. The six O-A vectors obtained for the six regulation factors can be referred for Table 5.2. The O-A vector for ρ of 0% and 0.1% was almost the same as E-A vector. The recalled activations regulation factor 0% was *[0.9875402, 0.6551013]* which is almost similar to the E-R vector *[0.9896, 0.6568]*. Figure 5.7b show all the recalled trajectories for this experiment.

- **Exp-8:** In this experiment the GBP algorithm is initialized with the E-A vector of *[0.09, 0.5, 0.22, 0.05, 0.52]*. The E-R vector for this experiment was *[0.3458, 0.1157]*, and two similar vectors ,*[0.3444873, 0.1032499]* and *[0.3489568, 0.1284956]*, were recalled in this experiment using regulation of 0% and 0.1% respectively, Figure 5.7c shows all recalled trajectories. The O-A vectors obtained with the regulation of 0% and 0.1% were also identical to E-A vector of this experiment, refer Table 5.2.

- **Exp-9:** The recall simulation in this experiment was instantiated with an E-A vector of *[0.09, 0.40, 0,28, 0.07, 0.35]*, and a recall vector of *[0.4410, 0.1341]* is expected at the input Layer-0. Upon using the GBP algorithm with six regulation factors the recall operation without regulation, i.e. ρ of 0%, produced the most similar recall vector *[0.4370989,*

Table 5.2: Observations of Activations at Abstract Concept Layer-1 for MCR Experiments

Regulation	**Experiment**	**E-A at Layer-1**					**O-A at Layer-1**				
%	————	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5
0	**Exp-6**	0.57	0.16	0.06	0.15	0.25	0.579	0.162	0.063	0.148	0.246
0.1	**Exp-6**	0.57	0.16	0.06	0.15	0.25	0.567	0.163	0.066	0.151	0.247
0.2	Exp-6	0.57	0.16	0.06	0.15	0.25	0.556	0.164	0.068	0.154	0.249
0.3	Exp-6	0.57	0.16	0.06	0.15	0.25	0.546	0.166	0.070	0.156	0.250
0.4	Exp-6	0.57	0.16	0.06	0.15	0.25	0.537	0.167	0.072	0.159	0.251
0.5	Exp-6	0.57	0.16	0.06	0.15	0.25	0.529	0.168	0.074	0.161	0.252
0	**Exp-7**	0.07	0.04	0.23	0.46	0.05	0.072	0.039	0.233	0.465	0.051
0.1	**Exp-7**	0.07	0.04	0.23	0.46	0.05	0.086	0.048	0.235	0.482	0.061
0.2	Exp-7	0.07	0.04	0.23	0.46	0.05	0.093	0.052	0.235	0.486	0.067
0.3	Exp-7	0.07	0.04	0.23	0.46	0.05	0.099	0.055	0.236	0.487	0.071
0.4	Exp-7	0.07	0.04	0.23	0.46	0.05	0.103	0.058	0.236	0.487	0.074
0.5	Exp-7	0.07	0.04	0.23	0.46	0.05	0.107	0.061	0.236	0.485	0.077
0	**Exp-8**	0.09	0.50	0.22	0.05	0.42	0.091	0.502	0.218	0.051	0.414
0.1	**Exp-8**	0.09	0.50	0.22	0.05	0.42	0.099	0.497	0.221	0.056	0.424
0.2	Exp-8	0.09	0.50	0.22	0.05	0.42	0.105	0.492	0.223	0.061	0.430
0.3	Exp-8	0.09	0.50	0.22	0.05	0.42	0.110	0.486	0.224	0.064	0.434
0.4	Exp-8	0.09	0.50	0.22	0.05	0.42	0.114	0.481	0.225	0.067	0.437
0.5	Exp-8	0.09	0.50	0.22	0.05	0.42	0.117	0.476	0.226	0.070	0.439
0	**Exp-9**	0.09	0.40	0.28	0.07	0.35	0.090	0.401	0.280	0.070	0.350
0.1	**Exp-9**	0.09	0.40	0.28	0.07	0.35	0.096	0.397	0.281	0.076	0.355
0.2	Exp-9	0.09	0.40	0.28	0.07	0.35	0.101	0.394	0.282	0.080	0.358
0.3	Exp-9	0.09	0.40	0.28	0.07	0.35	0.105	0.391	0.282	0.084	0.360
0.4	Exp-9	0.09	0.40	0.28	0.07	0.35	0.109	0.388	0.282	0.087	0.362
0.5	Exp-9	0.09	0.40	0.28	0.07	0.35	0.112	0.385	0.281	0.090	0.363
0	**Exp-10**	0.07	0.09	0.44	0.26	0.10	0.068	0.093	0.440	0.263	0.099
0.1	**Exp-10**	0.07	0.09	0.44	0.26	0.10	0.073	0.098	0.437	0.264	0.105
0.2	Exp-10	0.07	0.09	0.44	0.26	0.10	0.076	0.103	0.434	0.264	0.110
0.3	Exp-10	0.07	0.09	0.44	0.26	0.10	0.079	0.106	0.431	0.264	0.114
0.4	Exp-10	0.07	0.09	0.44	0.26	0.10	0.082	0.109	0.428	0.265	0.117
0.5	Exp-10	0.07	0.09	0.44	0.26	0.10	0.084	0.112	0.425	0.265	0.120

E-A [Expected Activation], O-A [Observed Activation];

0.1308456], and the corresponding O-A vector at Layer-1. However, the outcome with 0.1% regulations was also similar to a recalled vector *[0.4392012, 0.1518065]* at input Layer-0, refer Figure 5.7d.

- **Exp-10:** This experiment uses an E-A vector *[0.07, 0.09, 0.44, 0.26, 0.10]* in order to obtain an E-R vector *[0.8813, 0.4145]* at input Layer-0. The six simulations were carried out with different regulation factors, and it was observed that the results with ρ of 0% and 0.1% produced results very near to the E-R, i.e. *[0.8873921, 0.4137484]* and *[0.8702277, 0.4153301]* respectively, see Figure 5.7e for all trajectories. The same observations were made at O-A vectors for ρ of 0% and 0.1% at Layer-1, refer Table 5.2.

5.7.3 Discussion

The experiments in Sections 5.7.1 and 5.7.2 demonstrate a notable behavior of Regulated Activation Networks by simulating the cued recall operation through a Toy-data3 problem. The Intra-Layer Learning (i.e. CSRL) is uniquely used by RANs modeling to interpret the association among the concepts as Inhibitory, Excitatory or Neutral. Further, the Intra-Layer Regulation (Algorithm 4) uses the Intra-Layer Learning (CSRL) and its interpretations to produce a regulatory effect over the activation of the concepts (at same layer). The Geometric Back-propagation operation (Algorithm 5) is a method analogous to remembering something learned in an abstract form and recalling its concrete features. For example, while remembering the abstract concepts "Home" we recall concrete features linked to our homes like "Mother", "Father", "Wife", "Pets" etc.

In the graphs of Figures 5.6 and 5.7 we can see that all the trajectories commence from a starting point (red dot) and converge to a point after one thousand iterations. Each point in a trajectory represents a temporal mental state while recalling a concrete concept. Every time a concrete concept (activation vector in Layer-0) is recalled, the corresponding abstract concept (at Layer-1) is compared with the Expected abstract concepts. The difference between expected and observed activation is propagated back as an error to the previously recalled activations at Layer-0. In the next time instance, the corrected recalled activation at Layer-0 repeats the process until the one thousand iterations.

It was observed that, without regulation, i.e. 0% ρ, the trajectory converges to a point, but with a little bit of regulation, the result improves. For instance, in the graphs of Figure 5.6 only one abstract concept was being recalled and the results improved when the regulation was introduced. In the two experiments (SCR & MCR) we can see the two different sets of regulation factors are considered. These sets were obtained empirically, but we can see that the set of the regulation factor for SCR experiment has a higher value. This is because the GBP algorithm strives to minimizes the error at each abstract concept node at Layer-1, and in geometrical context, similarity cannot be the same for more than one abstract concept. Thus, the trajectory converges to a point, but the result improves when a little regulation is induced. With MCR experiment the best outcome is observed with little or no regulation because the expected similarity (DoC, E-A) is a non-zero value. The other reason is that these are possible expected similarity vector unlike the ones in SCR experiments.

5.8 Cued Recall Demonstration with MNIST Data

The MNIST (LeCun et al., 1999) dataset is a collection of handwritten images of digits [0, 1, 2, 3, 4, 5, 6, 7, 8 and 9], where each image is black & white in color and 28×28 pixels in size. This dataset of image domain is used to demonstrate the Cued recall operation of learned

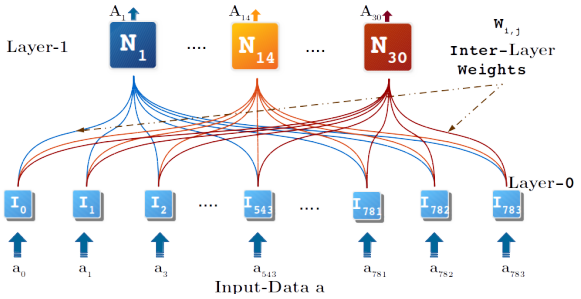


FIGURE 5.8: RANs model generated with MNIST dataset.

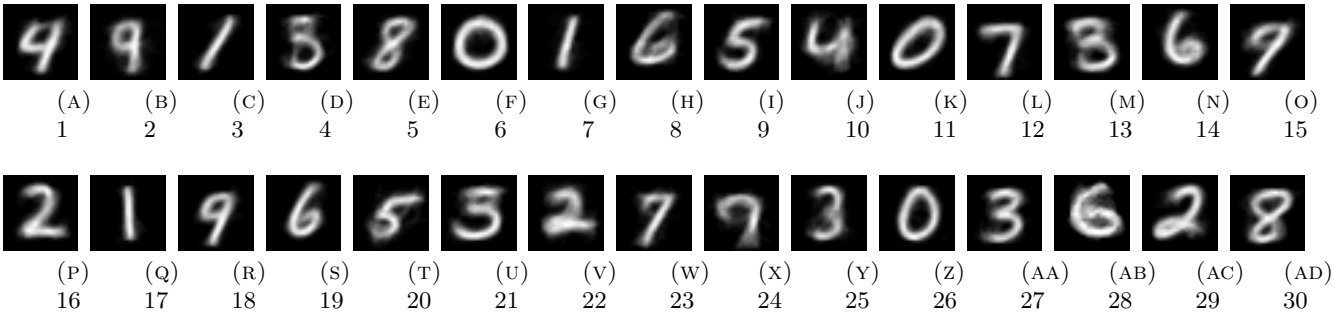


FIGURE 5.9: The thirty CRDPs (Cluster Centers), each node in Layer-1 of Figure 5.8 acts as the Abstract represenatative of each CRDP.

abstract concepts representing different digits. Two types of investigations were conducted with this dataset: first, is Multiple Binary Valued Cue Recall (MBVCR), where the E-A vector is a binary value ([0, 1]) vector; second, is the Multiple Cue Recall (MCR). For the experiment, one thousand images were selected randomly from the MNIST dataset. The 28×28 image was transformed in a single vector of 784 attributes, where each attribute corresponds to a pixel of the image. Further, the attribute values of the data were normalized between [0, 1] using Min-Max normalization (black pixel is Min, i.e. 0, and a white pixel is Max,i.e. 255). Having preprocessed the data, the RANs modeling procedure is instantiated by selecting the K-means clustering algorithm as concept identifier. The K is initialized with 30 to determine thirty categories in the input space. The model was configured to grow one level deep and build a convex abstract concept (CAC) Layer-1. After carrying out all four steps of RANs modeling (see Section 5.4) a model is obtained, see Figure 5.8.

In Figure 5.8, the Layer-0 has 784 nodes representing each pixel, CAC Layer-1 has 30 nodes representing the thirty categories identified at the CCI process in RANs modeling. The Inter-Layer Weights (ILWs) are the cluster centers (CRDPs) of the thirty clusters. Figure 5.9 shows the ILWs reconstructed in the image form of 28×28 pixels. In RANs modeling a CRDP is the optimum representative of an input level category at CAC Layer-1. Therefore, it images in Figure 5.9 are the best represented by CAC node N_1, ..., N_{30}, respectively. In the same figure it's noticeable that each digit is represented by at least two CAC node of Layer-1. The digit 9 is represented by the largest number of nodes, i.e. N_2, N_{15}, N_{18} and N_{24}, whereas digit 4 was

TABLE 5.3: Expected Activation injected at thirty CAC nodes in Layer-1 of RANs model for MNIST data

Digit	Exp	Expected Activation (E-A)
0	MBVCR	[0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0]
1	MBVCR	[0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0]
2	MBVCR	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0]
3	MBVCR	[0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0]
4	MBVCR	[1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
5	MBVCR	[0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0]
6	MBVCR	[0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0]
7	MBVCR	[0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0]
8	MBVCR	[0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1]
9	MBVCR	[0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0]
2& 5	MBVCR	[0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,1,1,0,0,0,0,0,0,1,0]
3 & 5	MBVCR	[0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,1,0,1,0,0,0]
0 & 1	MBVCR	[0,0,1,0,0,1,1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0]
0	MCR	[0.26,0.25,0.22,0.30,0.27,1.00,0.23,0.30,0.28,0.30,0.35,0.26,0.32,0.26,0.25,0.28,0.22,0.23,0.23,0.25,0.33,0.28,0.25,0.29,0.28,0.39,0.28,0.31,0.27,0.30]
1	MCR	[0.29,0.37,1.00,0.35,0.43,0.22,0.49,0.33,0.34,0.32,0.26,0.31,0.31,0.28,0.34,0.39,0.38,0.35,0.34,0.39,0.31,0.35,0.43,0.31,0.42,0.27,0.36,0.24,0.31,0.33]
2	MCR	[0.28,0.35,0.39,0.38,0.40,0.28,0.38,0.36,0.33,0.31,0.26,0.29,0.36,0.35,0.29,1.00,0.34,0.31,0.37,0.34,0.36,0.43,0.34,0.29,0.47,0.32,0.37,0.33,0.37,0.34]
3	MCR	[0.31,0.38,0.31,0.46,0.35,0.32,0.32,0.31,0.35,0.33,0.26,0.32,1.00,0.33,0.31,0.36,0.31,0.33,0.31,0.30,0.40,0.34,0.33,0.35,0.38,0.34,0.46,0.35,0.27,0.37]
4	MCR	[1.00,0.46,0.29,0.32,0.38,0.26,0.30,0.38,0.37,0.44,0.31,0.42,0.31,0.39,0.44,0.28,0.31,0.48,0.36,0.37,0.29,0.34,0.37,0.42,0.34,0.34,0.34,0.31,0.32,0.40]
5	MCR	[0.37,0.42,0.34,0.44,0.47,0.28,0.36,0.41,1.00,0.29,0.39,0.36,0.35,0.34,0.37,0.33,0.36,0.42,0.43,0.40,0.42,0.33,0.38,0.35,0.43,0.37,0.45,0.29,0.33,0.41]
6	MCR	[0.39,0.40,0.28,0.38,0.33,0.26,0.31,0.45,0.34,0.38,0.28,0.31,0.33,1.00,0.34,0.35,0.32,0.36,0.42,0.32,0.32,0.40,0.33,0.35,0.35,0.36,0.34,0.42,0.33,0.33]
7	MCR	[0.42,0.45,0.31,0.34,0.35,0.26,0.34,0.32,0.36,0.34,0.28,1.00,0.32,0.31,0.41,0.29,0.35,0.47,0.33,0.35,0.29,0.32,0.43,0.47,0.38,0.30,0.34,0.25,0.28,0.34]
8	MCR	[0.38,0.41,0.43,0.39,1.00,0.27,0.40,0.38,0.47,0.32,0.33,0.35,0.35,0.33,0.40,0.40,0.37,0.44,0.42,0.44,0.38,0.39,0.45,0.32,0.44,0.34,0.44,0.29,0.39,0.48]
9	MCR	[0.46,1.00,0.37,0.40,0.41,0.25,0.39,0.38,0.42,0.38,0.29,0.45,0.38,0.40,0.40,0.35,0.39,0.51,0.40,0.39,0.32,0.40,0.43,0.44,0.43,0.32,0.44,0.32,0.30,0.40]

represented by two nodes N_1 and N_{10}. The Figures 5.9d, 5.9m, 5.9u and 5.9x depict that the CAC nodes N_4, N_{13}, N_{21} and N_{24} do not represent individual digit. Node N_4 and N_{13} jointly represent digits 3 and 8, node N_{21} looks like two digits 3 and 5, and N_{24} depicts digits 7 and 9.

For simplicity, Figure 5.8 shows only Inter-Layer Learning, but the Intra-Layer Learning (CSRL weights) was also performed on both, input Layer-0 and CAC Layer-1. The CSRL weights at input Layer-0 was a 784×784 matrix, and at CAC Layer-1 a 30×30 matrix was learned. These two Intra-Layer Learning were utilized by GBP algorithm to simulate the recall operations. In all the experiments the GBP algorithm is configured to iterate five hundred times. The GBP algorithm was initialized with a vector with activation 1 for all 784 nodes of input Layer-0. The image at Iter-0 (see Table 5.4 & 5.6) is white because activation 1 corresponds to pixel value 255 depicting white color.

In each experiment, the two cued recall demonstrations, MBVCR and MCR, use the Expected Activating (E-A) vector as listed in Table 5.3. The experiments of single digits and combined digits for MBVCR operation used an E-A vector of binary values, where binary 1 at a node N is assigned w.r.t the digit(s) being recalled. For instance, the E-A vector of digit 2 is formed by initializing the E-A vector with binary 1 for nodes N_{16}, N_{22} and N_{29} and binary 0 for remaining 27 CAC nodes (see Table 5.3). The E-A vectors of MCR experiments are the actual activation values obtained by propagating upward the Inter-Layer Weights (the CRDPs see Figure 5.9) as input. The weights represented by Figures 5.9f, 5.9c, 5.9p, 5.9m, 5.9a, 5.9i, 5.9n, 5.9l, 5.9e and 5.9b were provided as input to CCUAP operation to observe their respective activation at CAC Layer-1. These observed activation vectors were used as E-A for each digit recall operation (see last ten MCR E-As in Table 5.3).

TABLE 5.4: Intuitive MBVCR observations with RANs model of MNIST data

Digit	ρ	Iter⇒	0	3	5	8	11	19	25	35	41	71	81	91	101	151	201	251	301	351	401	451	501
0	0%																						
0	0.009%																						
1	0%																						
1	0.009%																						
2	0%																						
2	0.009%																						
3	0%																						
3	0.009%																						
4	0%																						
4	0.009%																						
5	0%																						
5	0.009%																						
6	0%																						
6	0.009%																						
7	0%																						
7	0.009%																						
8	0%																						
8	0.009%																						
9	0%																						
9	0.009%																						

5.8.1 Multiple Binary Valued Cue Recall (MBVCR) Operation

For the MBVCR operation, the RANs model generated with MNIST data (see Figure 5.8) was used, in order to obtain the recalled activation at input Layer-0 for a given Expected Activation vector at CAC Layer-1. As described earlier, the E-A vector for MBVCR is a vector of binary values, which is provided as input to the GBP algorithm to perform the recall operation. The experiments themselves are divided into two categories, i.e. Intuitive and Non-Intuitive recall.

Intuitive MBVCR experiment

In this experiment, by intuition we hypothesize, if all CAC nodes (representing a digit) are activated with value 1 then its recall at Layer-0 must depict that digit. For example, if the CAC nodes N_6, N_{11} and N_{26} (refer Figure 5.9) are activated with value 1 (and 0 for others) then we should get an image depicting a blend of digit zero after recall operation. We performed this Intuitive recall experiment for all ten digits. The binary E-A vector of all ten digits for Intuitive MBVCR operation is listed in Table 5.3. Table 5.4 displays the recalled images of all twenty experiments. For every digit two investigations were made: first, without regulation, i.e. ρ=0; second, with a regulation of 0.009%.

TABLE 5.5: Non-intuitive MBVCR observations with RANs model of MNIST data

Digit	ρ	Iter⇒	0	3	5	8	11	19	25	35	41	71	81	91	101	151	201	251	301	351	401	451	501
2 & 5	0%																						
2 & 5	0.009%																						
3 & 5	0%																						
3 & 5	0.009%																						
0 & 1	0%																						
0 & 1	0.009%																						

The first observation is that there is a very insignificant difference between the images recalled with and without regulation. After the second iteration, the digit being recalled begins to appear. Beyond 80^{th} iteration no significant change is observed in the recalled images. The recalled images of digits 0, 1, 2, 3, 7 and 8 are recognizable after 500^{th} iteration. However, the digits 4, 5 and 9 is not very discernible in their last iteration, this is because these digits are cross-represented by CAC nodes (see Figure 5.9). All the images recalled in this experiment contains noise (i.e. the gray shades) because E-A vector has two values either 0 or 1 and a node can be 100% similar to only one node. Therefore, the GBP algorithm adjusts the activation at CAC node such that the best representation of the E-A is achieved.

Non-intuitive MBVCR experiment

In these experiments, an E-A vector contains activation value 1 for CAC nodes representing two different digits. The objective of the experiment was to determine what is recalled at the input Layer-0 when the CAC nodes, representing two different digits, expect high activation. The three E-As used in this experiment are a combination of activation 2^s-with-5^s, 3^s-with-5^s and 0^s-with-1^s (refer Table 5.3 for E-A vectors for the coupled digits). The observation without regulation and with regulation are similar, see Table 5.5. The blend of 2^s-with-5^s recalls an image that looks like letter 'x'. The fusion of 3^s-with-5^s recall an image similar to digit '3'. The combination of 0^s-with-1^s, in the beginning looked like a symbol 'Φ' which later gets distorted. We can also observe that the images obtained after all the iteration had lesser noise when compared to the Intuitive MBVCR experiments. This is probably because the number of CAC nodes were expecting activations, i.e. more cues were provided.

5.8.2 Multiple Cue Recall (MCR) Experiment

This experiment is the same as the one discussed in Section 5.7.2. The E-A vectors are the activation values observed at CAC nodes by propagating the Inter-Layer weights using CCUAP operation of RANs modeling. Figure 5.9 shows the images, re-constructed for each Inter-Layer weight. The E-As corresponding to Figures 5.9f, 5.9c, 5.9p, 5.9m, 5.9a, 5.9i, 5.9n, 5.9l, 5.9e and 5.9b are listed in Table 5.3, and are used in MCR demonstrations of this section.

TABLE 5.6: The observations of Multiple Cue Recall operation with the RANs model of MNIST data

Digit	ρ	Iter⇒	0	3	5	8	11	19	25	35	41	71	81	91	101	151	201	251	301	351	401	451	501
0	0%																						
0	0.009%																						
1	0%																						
1	0.009%																						
2	0%																						
2	0.009%																						
3	0%																						
3	0.009%																						
4	0%																						
4	0.009%																						
5	0%																						
5	0.009%																						
6	0%																						
6	0.009%																						
7	0%																						
7	0.009%																						
8	0%																						
8	0.009%																						
9	0%																						
9	0.009%																						

The objective of this experiment is the same as that of MBVCR experiments, i.e., obtaining an activation vector at input Layer-0 that corresponds to an E-A vector. However, in this experiment, an Expected Recall (E-R) is already known. Therefore, the E-As of ten digits (see MCR E-As in Table 5.3) are expected to recall the images in Figures 5.9f, 5.9c, 5.9p, 5.9m, 5.9a, 5.9i, 5.9n, 5.9l, 5.9e and 5.9b.

In this experiment also, the observations with and without regulation are identical. It is also worth noting that after 500^{th} iteration the recalled images of all ten digits are similar to the E-R images of each digit.

5.8.3 Discussion

There are a few things worth noticing in the recall demonstrations of RANs modeling with MNIST dataset. First, we can reconstruct cognizable images of a digit by activating the CAC nodes representing that digit. Second, it possible to recall both Intuitive and Non-intuitive blend of learned abstract concepts (in these experiments the abstract concepts are a generic

representation of digits). Third, the recalled activations, with and without regulation, are similar for a complex dataset like MNIST. At last, the more cues we provide in the E-A vector the accurate the recall operation becomes.

5.9 Conclusion

Concepts are normally perceived in a hierarchical form, and the concrete concepts occupy the lower level, whereas the abstract concepts take up the relatively higher level in the hierarchy. According to the context availability theory, the context among the concrete concepts are easily discernible when compared to abstract concepts, hence, the abstract concepts are difficult to understand. This abstruseness with abstract concepts adds to the complexity of their recall process. In this work, abstract concept models generated through the RANs approach are utilized to demonstrate the cued recall of learned abstract concepts.

To demonstrate the effect of regulation over the recall process, a Toy-data problem (Toy-data3) was considered. At first, we modeled with Toy-data to identify five abstract concepts. The proposed regulation algorithm used the learned Intra-Layer weight to determine excitatory, neutral and inhibitory impact induced by peer nodes over one another. Two types of cued recall experiments were performed using the unique Geometric Back-propagation algorithm: first, the Single Cue Recall (SCR) simulation where the recall was simulated by activating only one abstract concept; second, the Multiple Cue Recall (MCR) operation to retrieve the activation vector at the input level by injecting multiple cues at the abstract Nodes. In SCR experiments the regulation induced by peer nodes improved the recalled values. However, the observations with MCR operations were promising because it retrieved identical activation as expected.

The benchmark MNIST dataset was used to exhibit cued recall as blends of learned abstract concepts. A two-layered model was generated with RANs to obtain thirty abstract concepts generically representing digits. In Multiple Binary Valued Cue Recall (MBVCR) experiment, multiple abstract nodes were injected with high activation to recall as blends of digits. Interestingly, it was observed in all the experiments that the blend of abstract nodes recalled an image of the digit that they represent at the abstract Level. The blend of different digits also produced some intriguing outcome such as a blend of 2 and 5 recalled x, and blend of 0 with 1 looked like a Φ. The MCR operations were interesting, upon injecting the multiple cues the recalled image was very similar to the expected recalled image.

Both the experiments displayed how concepts can be contextually associated and impact each other's activation through regulation. Further, with cue recall operations it can be concluded that the more cues are injected to an abstract concept the better recall results are obtained.

Chapter 6

Conclusion

Concepts can be represented by *symbolic*, *distributed* or *spatial* conceptual representations. Individually, all three representations have proven their potential in the field of machine learning and computational cognitive modeling. For contextual processing brain requires both symbolic and distributed representation (Roy, 2011), whereas the spatial representations, such as conceptual spaces (Gärdenfors, 2004), are considered to occupy the space between the symbolic and distributed representations. The aim of this research work was to develop a computational approach that not only unifies the virtues of the three conceptual representation but also emulate the regulatory mechanism of Axoaxonic synapses (Garrett, 2014) and simulate cognitive processes such as concept creation, concept recall and learning. This Chapter addresses the two research objectives (see Section 1.2.2 of Chapter 1) of the work by answering the three research questions individually. Further, the applications of the RANs modeling are briefed along with future work directions.

6.1 Solutions to Research Problems

In this research two problems were focused upon: first related unified view of concept representation; and second regarding emulation of regulation operation of Axoaxonic synapses (see Section 1.2.1 of Chapter 1). To solve these two problems two hypothesis were made:

- A computational network where every node symbolically represents a concept (abstract of concrete) and geometry among the concepts is considered in modeling, such a network can be considered a hybrid of symbolic, distributed and spatial representation. A hierarchical network of concepts (nodes) can address the representation of abstract and concrete concepts computationally. If the hierarchy generation is dynamic, the cognitive processes such as concept creation, learning and activation propagation can be simulated.

- The regulation behavior of Axoaxonic synapses can be simulated if the concepts at the same level are able to impact each other's activation. In the process of regulating activation of other concepts, a dynamic state of concepts can be captured similar to dynamic state when recalling a concept.

The two hypothesis were used to formulate three research questions which helped in carrying out the research work. In this section we show how we tried to address issues related to the the research questions

Research Question 1: *Which technique can be useful in viewing concepts as data points in n-dimensional feature hyperspace? How to build a hierarchy of concepts where higher level concepts abstractly represent concepts at lower levels? How to learn the relation between concepts at the same level, and different levels in the hierarchy?*

Answer: The n-dimensional view of concepts in RANs modeling is inspired by the theory of conceptual spaces (Gärdenfors, 2004) (see Appendix A.3), where the $\langle$ *Feature, Value* $\rangle$ pairs of the data instance enable us to view the concepts in multi-dimensional hyperspace. In RANs modeling clustering algorithms were considered because the majority of them view data as points and further groups these points based upon some distance measures. According to the prototype theory of categorization (Rosch, 1975, Mervis and Rosch, 1981, Rosch, 1983) the most generic point in the cluster acts as its abstract representative. Inspired by this prototype theory, the convex abstract concept modeling (described in Chapter 3) utilizes the clusters counts to dynamically create a new layer of abstract concepts and uses the cluster centers as learning (see CCILL in Section 3.3.3) between the layers. The concept hierarchy creations algorithm (Algorithm 2 in Chapter 3) shows how deep hierarchy convex abstract concepts can be created dynamically and without supervision. Chapter 4 describes another kind of hierarchy creation which is an extension of RANs modeling on Chapter 3. The non-convex abstract concept layer is created using an intra-layer learning mechanism (see CSRL method in Section 4.2.2) that determines the similarity relation among the nodes in a layer.

Research Question 2: *How to propagate activation (signals) from the input-to-abstract level using both types of learning? How to strew activation from abstract-to-input level using the learning? How use learning to have a regulatory effect of activation of one concept over other?*

Answer: In RANs modeling, every node in the network *symbolically* represents a concept. The lower level nodes are termed as concrete concepts and the, relatively, higher level nodes represent the abstract concepts. The *activation* value of every node depicts the degree with which the node represents the concept symbolically: for example if a node gets an activation of 0.78 it means the node identifies the concept with confidence of 78%. The CACUAP (see Section 3.3.4) and NACUAP (see Section 4.2.3) are two algorithms to propagate activation from input to output levels. These propagation methods transfer the activation confirming to the assumption of percentage representation of the nodes (refer Section 3.1 of Chapter 3). RANs

modeling has another activation propagation method, GBP algorithm (Algorithm 5 Chapter 5) which propagates activation from an abstract-to-concrete level (or from Output-to-Input layer). The CSRL method introduced in Section 4.2.2 of Chapter 4 is uniquely interpreted to obtain the *excitatory*, *inhibitory* and *neutral* effect of one concept over other. Further, these interpretations are used to calculate the *regulation* effect of one concept on other concepts.

Research Question 3: *what use cases are suitable for RANs modeling? Which methods are suitable to validate the machine learning capability of the methodology? How to simulate and verify the cognitive behaviors exhibited by the approach and substantiate the processes like recall and blend retrieval of concepts?*

Answer: The datasets used in the research work were mostly benchmarks obtained from sources like the UCI machine learning repository. Three datasets were artificially generated to describe the RANs methodology. Besides these datasets, one benchmark data from the image domain and one dataset from the IoT domain were also used in this research work. RAN can perform classification tasks as described in Chapters 3 and 4 and standard metrics *Precision*, *Recall*, *F1-Score*, *Accuracy* and *ROC-AUC* curve analysis were chosen to validate the RANs classification performance. The classification operation was also helpful in proving the hypothesis of RAN's convex and non-convex abstract concept modeling (as described in Chapters 3 and 4 respectively). In Chapters 3 and 4 the simulation of cognitive functions of *concept creation*, *learning* and *activation* propagation are demonstrated using a Toy-data problem and validated through analogy and experimental results. Another Toy-data problem was used in Chapter 5 to simulate the *recall* cognitive function. The GBP operation (Algorithm 5 in Chapter 5) enables the RANs modeling to perform *cued recall* simulations (see Sections 5.7 & 5.8 of Chapter 5). RANs GBP method also enables the *intuitive* and *non-intuitive* blend retrievals of abstract concepts (refer Section 5.8). The RANs behavior modeling was validated by comparing the empirical outcome of the simulations with the expectations of the experiments.

6.2 Research Contribution

Regulated Activation Network is a computational model that learns representations of concepts (or concrete & abstract concepts) without supervision. RANs modeling is hybrid in nature, i.e., it consists of properties of *symbolic*, *distributed* and *spatial* conceptual representations. Topologically RANs modeling is distributed (i.e., connectionist) where learning is performed by viewing concepts as points in n-dimension feature hyperspace. Moreover, every node of the network *symbolically* identifies a concept. Two variations of RANs modeling were developed during this research work: firstly, modeling of convex abstract concepts (see Chapter 3) was introduced; and secondly model of non-convex abstract concepts (see Chapter 4). Both versions displayed the novelty of RANs modeling where the network evolves by creating a new layer of

concepts dynamically. In the convex and non-convex abstract concept modeling with RANs 2 concept creation, 3 learning and 2 activation propagation cognitive function were also simulated.

The RANs classification capability was tested with eleven benchmarks of UCI machine learning repository, three artificial datasets, one dataset from the image domain and one dataset from IoT sources. In the experiments, the performance of both versions of RANs modeling was quite satisfactory. It was also observed that the RAN's model outperformed the classifier RBM+ in a large number of cases. In several experiments the RANs produced similar outcomes as of MLP, LR, SGD and K-NN, and in some experiments, RANs even outperformed one of the classifiers. The RANs modeling also produced very good results when trained with a limited amount of data. Besides satisfactory performance, the RANs modeling automatically identifies the similar categories-labels in data as one abstract concept (see the experiments with UCIHAR data in Sections 3.4.2 & 4.3.2). The IoT dataset obtained from project SOCIALITE was used to model the *Active* and *Inactive* state of students through their smartphone usage. The generated RANs model was used and the outcome was statistically analyzed, and based upon the assumptions of the experiment the results indicated towards the ill psychological or physiological of one student participant.

The third activation propagation operation of this research was introduced in Chapter 5 as the GBP algorithm which propagates activation from the output-to-input layer. In behavior simulation experiments of Chapter 5 the GBP algorithm was used to simulate the cognitive recall operations. The *Single Cue Recall* experiments in Section 5.7.1 depicted the recall of an abstract concept when cues are limited. The *Multiple Cue Recall* experiments (see Sections 5.7.2 & 5.8.2) were able to retrieve almost exact activation values as expected at the input layer nodes. In *Single Cue Recall* experiments with Toy-data we also show how *excitatory*, *inhibitory* and *neutral* effect regulates the activation of a concept to obtain better recall values. The *Multiple Binary Valued Cue Recall* displayed one more feature of RANs modeling, i.e., the blend retrieval of abstract concepts (see Section 5.8.1). All *intuitive* blends were obtained as per the expectations, whereas, the *non-intuitive* blends not only produced an intriguing combination of recalled abstract concepts but also with lesser noise when compared with *intuitive* blends.

Apart from classification and cognitive function simulations, the RANs modeling is useful in performing the data analysis tasks. RANs modeling performs both *dimension expansion* and *dimension reduction* operation in the data. By alternately reducing and expanding the dimension of the data and building a multi-layered RANs model, the data gets transformed and becomes more comprehensible (refer to Appendix E).

6.3 Prospective Research & Development Work

I the past 15 years the research in *distributed* computational model, from Recurrent to Deep Neural Networks, experienced an extreme amount of success in industrial application. All these models some way or the other addresses the learning process in the human brain. Besides, the *symbolic* (ACT-R) and *hybrid (symbolic + distributed)* (CLARION) approaches have been useful in simulating and studying the human cognitive processes. The RANs model developed as part of this research is a computational cognitive model that unifies the virtues of *symbolic*, *distributed*, and *spatial* representations. The RANs modeling simulates the cognitive processes, such as *concept creation*, *learning*, and *recall* operations as a pattern of static events. Moreover, it will be interesting to study temporal aspects of activities as it related to the sequence of patterns. In the future, I would like to simulate and model sequence of events, the temporal aspect, that induces thought in both *learning* and *forgetting* processes.

Perceiving, learning, abstracting and reasoning are four important factors recognized by DARPA for an effective AI to process information (Launchbury, 2017). According to DARPA: the first wave of AI-enabled perceiving and reasoning over a narrowly defined problem, where models had no learning capabilities and poor handling of uncertainties; the second wave of AI the methodologies are good in learning and perceiving with distinction in classification and prediction capabilities, but have no contextual capabilities and minimal reasoning (Launchbury, 2017). The RANs model is capable of learning context among the concepts at the same level in the models which helps in associating alike concepts together. In the future, I would like to study and develop a methodology to learn association among multiple models and contribute to the *contextual learning* research.

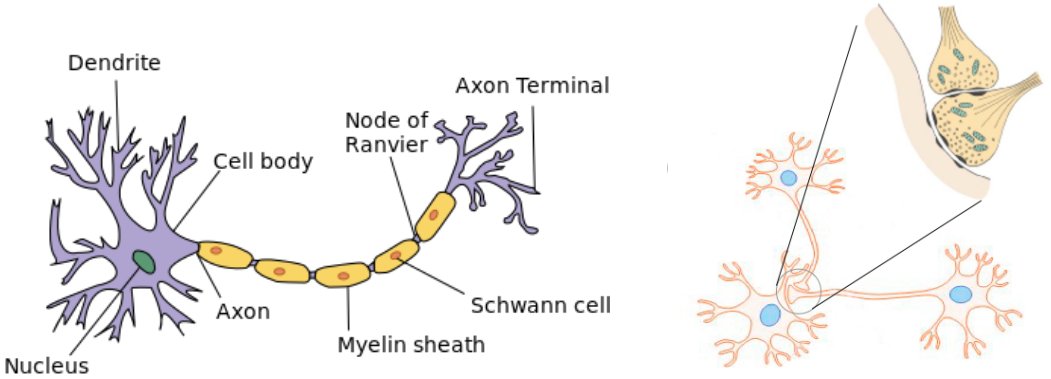
Appendix A

Theories and Principles Used in Regulated Activation Networks Modeling

This section provides information about the biological inspiration for regulation operation proposed in this article. It also highlights the principles behind RANs modeling.

A.1 Axoaxonic Synapses

The nerve cell (neuron) consists of several main components: the Dendrites, the Cell body, and the Axon, as shown in Figure A.1a. The Dendrites are the tree-like receptive network made of nerve fiber that carries electrical signals into the cell body. The cell body performs



(A) Diagram of Biological Human Neuron (Wikipedia, 2018)

(B) Axoaxonic Synapses

FIGURE A.1: The two views of human neuron. Figure A.1a details an individual neuron cell in human. Figure A.1b shows the Axoaxonic connection at a neuron from another neuron

the integration of these signals and, if the resulting electrical potential exceeds some threshold, the neuron "fires" propagating a similar signal to other neurons along its Axon. The Axon is a single long fiber carrying electrical signals from the cell body to others. The connection point between the Axon of a cell and a Dendrite of another is called a Synapse. When an electric signal traverses the whole Axon and reaches one of its terminations it releases chemicals called Neurotransmitters, which diffuse across the Synaptic gap, and are absorbed by the receptive neuron's Dendrite. Depending on the neurotransmitter, this absorption can either enhance or inhibit the receptive neuron's activation. The Synapse's effectiveness can be adjusted by the signals passing through it so that the Synapses' can learn from the activities in which they participate, and this, in part, is responsible for human memory.

Other kinds of synapses occur in biological brains, such as axoaxonic synapses, as shown in figure A.1b. These synapses occur when the axon of a neuron connects to the axon of another neuron instead of to its dendrites. Such configuration usually plays a *regulatory* role by mediating presynaptic inhibition and presynaptic facilitation (Garrett, 2014). By virtue of artificial axoaxonic synapses, this contribution realizes the regulatory phenomenon in RANs modeling, that induces tuning effect to the activation at nodes during recall operation.

A.2 Principles of Regulated Activation Networks

The principles of RANs (Pinto and Barroso, 2014) modeling conceptualize a technique: which is connectionist in topology; capable of representing and simulating the varying cognitive phenomena of an agent, its learning and recall processes, the association of ideas, and the creation of new concepts. To emulate the dynamic cognitive state such as the Priming (Jacoby, 1983) and False Memory (Roediger and Blaxton, 1987, Roediger and McDermott, 1995) two things were identified as essential: first, temporal activation state of a concept in agent's brain; and second, the association establishment between two concepts with respect to their activation at a given instance of time. To address the habituation and sensitization phenomena two more things were stated by the author: first, the model should have a dynamic behavior; and second, the model must have a leaning mechanism create new concepts abstract in nature. The first version of RANs modeling (Pinto and Barroso, 2014) demonstrated a single-layered learning and reasoning mechanism, with a simulation of depicting a cognitive process of spreading activation (Collins and Loftus, 1975).

The second version of RANs (Sharma et al., 2018b) modeling introduced the evolving connectionist topology, and abstract representation of concepts (described in Chapter 3). The extension of the second version of RANs modeling with a unique translation of the RAN's Intra-Layer learning mechanism (Pinto and Barroso, 2014, Sharma et al., 2018a) used to model non-convex concepts as described in Chapter 4. Chapter 5 also extends the second version of RANs to simulate recall behavior where lateral associations among concepts are interpreted are

inhibitors, inert and exciters, and later they are used in regulating the activation of concept nodes during recall operation.

A.3 Spreading Activation

Spreading Activation is a theory of memory (Anderson, 1983b) based on Collins and Quillian's computer model (Collins and Quillian, 1969) which has been widely used for the cognitive modeling of human associative memory and in other domains such as information retrieval (Crestani, 1997). It intends to capture both the way information is represented and how it is processed. According to the theory, long-term memory is represented by nodes and associative links between them, forming a semantic network of concepts. The links are characterized by a weight denoting the associative or semantic relation between the concepts. The model assumes activating one concept implies the spreading of activation to related nodes, making those memory areas more available for further cognitive processing. This activation decays over time, and the further it spreads, which can occur through multiple levels (McNamara and Altarriba, 1988), the weaker it is. That is usually modeled using a decaying factor for activation. The method of spreading activation has been central in many cognitive models due to its tractability and resemblance of interrelated groups of neurons in the human brain (Roediger et al., 2001).

A.4 Theory of Conceptual Spaces

Feature(or Value) set representation can alternatively be seen as spatial representations, where entity(or feature or concept) is represented by one or more number of entity(or feature or concept) and their activation(or value) at particular instance of time and thus representing entity(or feature or concept) into a multi-dimension space.

The theory of Conceptual Spaces proposed by *Gärdenfors'*(Gärdenfors, 2004) promoted the idea of seeing concepts into multi-dimension spaces, over here the representations are inspired by their geometrical aspects of concepts rather than symbolism or connectionism. The *similarity* among the concepts can be identified based upon the geometrical distance between the objects. The conceptual spaces, thus, serves as a natural way or tool to representing similarity relation among concepts (entities or objects). Since the conceptual spaces consider the geometrical aspects and easily address the notion of *similarity* relation, thus helps in understanding several cognitive phenomena like recognition. Formally we can say, a conceptual space S is formed by a set/class of *quality dimensions* $D_1, ..., D_n$. A point in S is represented by a vector $v = \langle d_1, ..., d_n \rangle$, where $\{1,....n\}$ represent index of dimensions.To form spatial structures on concepts topological characteristics of quality dimension can be utilized thus obtaining a *natural concept*, a *convex*

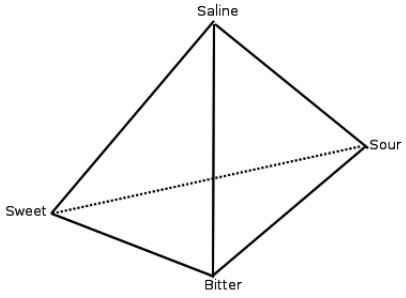


FIGURE A.2: Henning's taste tetrahedron(Gärdenfors, 2004)

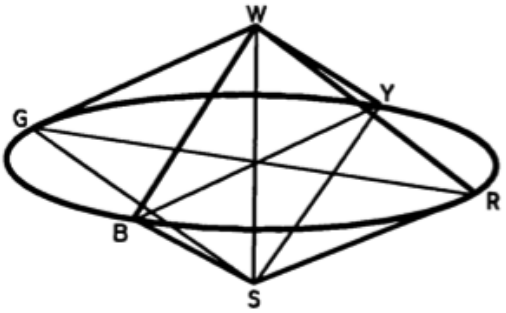


FIGURE A.3: The color space (Sivik and Taft, 1994)

region of conceptual space. Any convex region C that falls between to points $v_1 \& v_2 \epsilon S$, then any point x between $v_1 \& v_2$ also belong to the same convex region C.

The quality dimension is the basic requirement for conceptual spaces (Gärdenfors, 2004). The work of addressing the human taste perception based upon the different type of receptors: *sweet, sour, saline and bitter*, is used as an example to demonstrate quality dimension for sensory space representation (Gärdenfors, 2004). Here the quality space is represented by a tetrahedron (1 dimension for each taste) as shown in Figure A.2. Any point on each plane of the tetrahedron (Figure A.2) represents a taste, however, no taste is mapped as point lying interior to the tetrahedron.

Another example is a color space with the dimensions hue, saturation (or chromaticism), and brightness. Each quality dimension has a particular geometrical structure. For example, hue is circular, whereas brightness and saturation correspond with finite linear scales (Figure A.3). It is important to note that the values on a dimension need not be numbers. Quality dimensions may be grouped into domains. A domain is a set of integral (as opposed to separable) dimensions, meaning that no dimension can take a value without every other dimension in the domain also taking a value. Therefore, hue, saturation, and brightness in the above color model form a single domain. Conceptual space is simply "a collection of one or more domains" (Gärdenfors, 2004). For example, a conceptual space of elementary colored shapes could be defined as a space comprising the above domain of color and a domain representing the perceptually salient features of a given set of shapes.

A property corresponds to a region of a domain in a conceptual space. A concept, represented in terms of its properties, corresponds to a region in space, normally including multiple domains. Property is a special case of concept. For instance, the concept "red' is a region in the color space. It is also a property of anything which is red. An object is a point in space. The spatial location of an object in a conceptual space allows the calculation of the distance between objects, which gives rise to a natural way of representing similarities. The distance measure may be a true metric, or non-metric, such as a measure based on an ordinal relationship or the length

of a path between vertices in a graph. When calculating distance, the salience (weight) of the dimensions is varied. It is the context where a concept is used that determines which dimensions are the most prominent, and hence, have bigger weights.

The theory of conceptual spaces also address the [1] *prototype theory* of categorization (Rosch, 1975, 1983, Mervis and Rosch, 1981), if concepts are described as convex regions of conceptual space, prototype effects are indeed to be expected. In a convex region, one can describe positions as being more or less central. For example, if color concepts are identified with convex subsets of the color space, the central points of these regions would be the most prototypical examples of the color.

Such spatial representations naturally afford to reason in terms of spatial regions. Boundaries between regions are fluid, an aspect of the representation that may be usefully exploited by creative systems searching for new interpretations of familiar concepts. Besides, conceptual spaces are particularly powerful in dealing with concept learning and concept combination. However, the conceptual spaces theory imposes some constraints on what kinds of subspaces can be considered concepts, i.e., requiring them to be convex, which may compromise its applicability in general.

[1] The main idea of prototype theory is that within a category of objects, like those instantiating a concept, certain members are judged to be more representative of the category than others. For example, robins are judged to be more representative of the category "bird" than are ravens, penguins and emus

Appendix B

Support Utilities & Configurations for Modeling with Regulated Activation Networks

The appendix describes several methods that are useful in the evaluation process of RANs modeling.

B.1 Abstract Concept Labeling (ACL)

This method is optional and useful when the input data is labeled. With this mechanism, we associate an identifier to every abstract concept node N_j. Having generated the RAN's model with CCI, then trough CACC, CCILL, input data is sorted label-wise, and perform CACUAP operation. The propagated Data is inspected class-wise, and label node N_j with a class-name for which it got the maximum count of highest activation. For example, suppose input data for class-X has 100 instances, after inspecting the propagated data, it is observed that node N_1 received highest activation 74-times, whereas, with remaining 26 cases other nodes experienced maximum activation, therefore, we recognize node N_1 as representative of class-X. *True-Labels* are identified by directly mapping class of each input instance to its respective node representative. *Observed-Labels* are obtained by propagating every test-instance through CACUAP operation, inspecting which abstract node received the highest activation for that data-unit, and label it with the class represented by that node. True-Labels and Observed-Labels are used to validate the model's performance.

B.2 Non-convex Abstract Concept Labeling (NACL)

NACL is a non-obligatory method in RANs modeling. It is applied to symbolize NAC nodes at Layer-2, by associating them to input label incorporated with the data instances. Having generated the RAN's model with all nine steps, input train data is sorted label-wise, and each input instance was propagated upward using both upward activation operations (i.e., CACUAP, and NACUAP) serially. The class-wise inspection of Activation of node NAC_j associate classes to node NAC_j as labels. For example, suppose the Layer-2 of the model has two nodes NAC_1 and NAC_2, and input data for class-X has 100 instances. The inspection of the activation of all 100 instances observed that node NAC_1 received highest activation 74-times, whereas, with remaining 26 instances node NAC_2 experienced maximum activation, therefore, we recognize node NAC_1 as representative of class-X. *True-Labels* are identified by directly mapping class of each input instance to its respective NAC node representative. *Observed-Labels* are obtained by propagating every test-instance through, both, upward activation operations and inspecting which abstract node received the highest activation for that data-unit, and label it with the class represented by that node. True-Labels and Observed-Labels are used to validate the model's performance.

B.3 ROC curve analysis of the model generated with RANs

This study is carried out by two processes, first the input True-labels are transformed into a separate vector of binary labels, individually for all abstract nodes (i.e. 1 for class c1, 0 for all other classes), second, calculating the confidence score for each instance of the input data (or test-data). Both processes are described as follows:

1 **Node-wise binary transformation of input true-labels**: For example, suppose there are three classes (c1, c2, c3) represented by three abstract nodes (n1, n2, and n3) in RAN's model at Layer-2, and let True-label be [c1, c2, c2, c1, c2, c3, c3] for seven test instances, then for node n1 label will be [1, 0, 0, 1, 0, 0, 0] where 1 represents class c1, and 0 depicts others (i.e. c2, and c3).

2 **Node-wise confidence-score calculation**: This is calculated by averaging activation-value and confidence-indicator of activation for an input instance at an abstract node. Activation-value is an individual activation of an activation vector obtained by propagating up the data using the upward activation propagation (UAP i.e., CACUAP and NACUAP) mechanism of RANs whereas, confidence-indicator is calculated by min-max normalization operation of activation vector. For example, after UAP operation each node (n1, n2, and n3) receives activation [0.89, 0.34, 0.11] (a vector of activation), and confidence-indicator

TABLE B.1: Ratio distribution of Train and Test data in nine Research Designs

RD-1		**RD-2**		**RD-3**		**RD-4**		**RD-5**	
Train	*Test*	*Train*	*Test*	*Train*	*Test*	*Train*	*Test*	*Train*	*Test*
90%	10%	80%	20%	70%	30%	60%	40%	50%	50%
RD-1		**RD-7**		**RD-8**		**RD-9**		———	
Train	*Test*	*Train*	*Test*	*Train*	*Test*	*Train*	*Test*	———-	———
40%	60%	30%	70%	20%	80%	10%	90%	———-	———
Research Design: RD, Training Data Size: Train, Test Data size: Test									

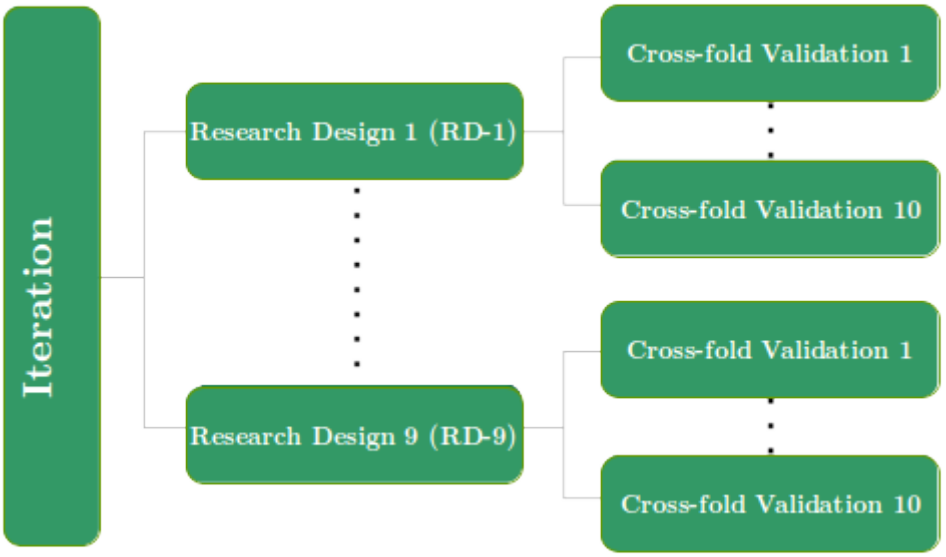


FIGURE B.1: One iteration in the Experimental Setup.

is min-max ([0.89, 0.34, 0.11]) = [1.0, 0.29, 0.0]. and the confidence-score for nodes n1= (0.89 + 1.0)/2.0 = 0.95, n2= (0.34 + 0.29)/2.0 = 0.32, and n3= (0.11 + 0.11)/2.0 = 0.05.

B.4 Research Design

Table B.1 lists the nine Research Designs (RD) used in the experiments of this article. In every RD the ratio of the Train and Test data is varied to capture the ability of the classifier being inspected.

B.5 Experimental Setup

To validate the classifiers the experimental setups were created with many iterations. Figure B.1 shows one iteration of an experimental setup, it consists of nine Research Designs as described by Table B.1. Each Research Design carry out 10-fold cross validation.

Table B.2: Dataset specific configuration details of models in simulations of chapter 3

Data	Algo	Configurations	Data	Algo	Configurations
Toy-data	*RBM +*	Lr=0.000001, iter=500, comp=20 max_iter=30, C=70	**UCIHAR**	*RBM +*	Lr=0.06, iter=500, comp=10 max_iter=10, C=1
	K-NN	n_neighbors=30 max_iter=10, C=1		*K-NN*	n_neighbors= 15 max_iter=30, C=1
	MLP	Rs=1, hls=10, iter=250		*MLP*	Rs=1, hls=10, iter=400
	RAN	CLS=5, Desired_depth=1		*RAN*	CLS=2, Desired_depth=1
	SGD	alpha=0.0001, n_iter=5, epsilon=0.25		*SGD*	alpha=0.1, n_iter=10, epsilon=0.25
Mice Protein	*RBM +*	Lr=0.1, iter=500, comp=20 max_iter=30, C=30	**Breast Cancer 569**	*RBM +*	Lr=0.006, iter=100, comp=10 max_iter=30, C=1
	K-NN	n_neighbors=15 max_iter=4, C=0.00001		*K-NN*	n_neighbors=30 max_iter=10, C=0.001
	MLP	Rs=1, hls=10, iter=300		*MLP*	Rs=1, hls=10, iter=200
	RAN	CLS=8, Desired_depth=1		*RAN*	CLS=2, Desired_depth=1
	SGD	alpha=0.1, n_iter=10, epsilon=0.25		*SGD*	alpha=,0.0001 n_iter=5, epsilon=0.25
Breast Cancer 669	*RBM +*	Lr=0.001, iter=100, comp=10 max_iter=30, C=1	**Credit Approval**	*RBM +*	Lr=0.006, iter=100, comp=10 max_iter=30, C=1
	K-NN	n_neighbors=10 max_iter=10, C=0.001		*K-NN*	n_neighbors=30 max_iter=10, C=0.001
	MLP	Rs=1, hls=10, iter=200		*MLP*	Rs=1, hls=10, iter=200
	RAN	CLS=2, Desired_depth=1		*RAN*	CLS=2, Desired_depth=1
	SGD	alpha=0.0001, n_iter=5, epsilon=0.25		*SGD*	alpha=0.0001, n_iter=5, epsilon=0.25
Glass Identification	*RBM +*	Lr=0.001, iter=400, comp=10 max_iter=30, C=5	**Mamographic Mass**	*RBM +*	Lr=0.01, iter=500, comp=20 max_iter=30, C=5
	K-NN	n_neighbors=15 max_iter=5, C=0.00001		*K-NN*	n_neighbors=30 max_iter=5, C=1
	MLP	Rs=1, hls=10, iter=200		*MLP*	Rs=1, hls=10, iter=250
	RAN	CLS=2, Desired_depth=1		*RAN*	CLS=2, Desired_depth=1
	SGD	alpha=0.01, n_iter=10, epsilon=0.25		*SGD*	alpha=0.0001, n_iter=5, epsilon=0.25
IRIS	*RBM +*	Lr=0.01, iter=1000, comp=20 max_iter=30, C=5	**Wine Recognition**	*RBM +*	Lr=0.01, iter=500, comp=20 max_iter=30, C=50
	K-NN	n_neighbors=15 max_iter=10, C=1		*K-NN*	n_neighbors=15 max_iter=10, C=0.01
	MLP	Rs=1, hls=10, iter=400		*MLP*	Rs=1, hls=10, iter=300
	RAN	CLS=3, Desired_depth=1		*RAN*	CLS=3, Desired_depth=1
	SGD	alpha=0.01, n_iter=10, epsilon=0.25		*SGD*	alpha=0.01, n_iter=10, epsilon=0.25
Lr-Learning Rate; iter-Iterations; comp-Number of Hidden Components of RBM; RS-Random State hls=Hidden Layer Sizes; CLS-Number of clusters at the input layer of RAN					

B.6 Model Configurations and Research Design

Various experiments, reported in this thesis, were conducted with several datasets, using six modeling techniques including the proposed methodology i.e. RANs modeling. Table B.2 shows the configurations of all the models for all the experiments in Chapter 3. Table B.3 lists the configuration of methodologies used in Chapter 4. The experiments were carried out using python programing language, and implementations of Restricted Boltzmann Machine pipelined with Logistic Regression (RBM+), Logistic Regression (LR), K-Nearest Neighbor (K-NN), Multilayer Perceptron (MLP), and Stochastic Gradient Descent (SGD) models of Scikit-learn library (Pedregosa et al., 2011). It is to be noted that experiments with RBM were carried out, pipelined with the LR algorithm using the default configuration of its implementation in scikit-learn library.

Table B.3: Dataset specific configuration details of models in simulations of chapter 4

Data	Algo	Configurations	Dataset	Model	Configuration
Toy-data	*RBM+*	Lr=0.001, iter=200, comp=10; C=1	**UCIHAR**	*RBM+*	Lr=0.001, iter=200, comp=10; C=1
	KNN	n_neighbors=30		*KNN*	n_neighbors=15
	LR	max_iter=5,C=0.00001		*LR*	max_iter=5,C=0.00001
	MLP	Rs=1, hls=5, iter=200		*MLP*	Rs=1, hls=10, iter=200
	RAN	CLS=9, ST=0.72		*RAN*	CLS=6, ST=0.75
	SGD	alpha=0.0001,n_iter=5, epsilon=0.25		*SGD*	alpha=0.0001,n_iter=5, epsilon=0.25
Mice Protein	*RBM+*	Lr=0.006, iter=500, comp=10; C=1	**Breast Cancer 569**	*RBM+*	Lr=0.001, iter=500, comp=20;C=1
	KNN	n_neighbors=30		*KNN*	n_neighbors=30
	LR	max_iter=5,C=0.001		*LR*	max_iter=5,C=0.00001
	MLP	Rs=1, hls=10, iter=400		*MLP*	Rs=1, hls=10, iter=200
	RAN	CLS=12, RT=0.97		*RAN*	CLS=12, ST=0.80
	SGD	alpha=0.0001,n_iter=5, epsilon=0.25		*SGD*	alpha=0.0001,n_iter=5, epsilon=0.25
IRIS Data	*RBM+*	Lr=0.001, iter=500, comp=20; C=1	**Credit Approval**	*RBM+*	Lr=0.001, iter=500, comp=20; C=1
	KNN	n_neighbors=15		*KNN*	n_neighbors=30
	LR	max_iter=5,C=0.01		*LR*	max_iter=5,C=0.00001
	MLP	Rs=1, hls=10, iter=400		*MLP*	Rs=1, hls=10, iter=200
	RAN	CLS=12, ST=0.82		*RAN*	CLS=12, ST=0.84
	SGD	alpha=0.001,n_iter=5, epsilon=0.25		*SGD*	alpha=0.0001,n_iter=5, epsilon=0.25

Lr-Learning Rate; iter-Iteration; comp-number of hidden components of RBM; Rs-random state; hls-Hidden Layer Sizes; ST-Similarity Threshold; CLS- number of clusters 'k' being identified by concept identifier K-mean;

Appendix C

Observations with Convex Abstract Concept Modeling

This Appendix provides the observations of experiments of Chapter 3 showing the classification performance of RANs modeling along with five classifiers (Restricted Boltzmann Machine pipelined with Logistic regression, Logistic Regression, Multi Layer Perceptron, K Nearest Neighbor and Stochastic Gradient Descent). The investigations were carried out with one artificially generated Toy-data problem and nine benchmark datasets obtained from the UCI machine learning repository.

TABLE C.1: Observations with Toy-data for Convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	100.000	100.000	100.000	100.000
K-NN	RD-2	100.000	100.000	100.000	100.000
K-NN	RD-3	100.000	100.000	100.000	100.000
K-NN	RD-4	100.000	100.000	100.000	100.000
K-NN	RD-5	99.987	99.987	99.987	99.987
K-NN	RD-6	100.000	100.000	100.000	100.000
K-NN	RD-7	99.934	99.933	99.933	99.933
K-NN	RD-8	99.925	99.925	99.925	99.925
K-NN	RD-9	99.772	99.675	99.644	99.675
LR	RD-1	99.621	99.600	99.600	99.600
LR	RD-2	99.483	99.467	99.467	99.467
LR	RD-3	99.653	99.644	99.644	99.644
LR	RD-4	99.689	99.683	99.683	99.683
LR	RD-5	99.672	99.667	99.667	99.667
LR	RD-6	99.649	99.644	99.644	99.644
LR	RD-7	99.698	99.695	99.695	99.695
LR	RD-8	99.652	99.650	99.650	99.650
LR	RD-9	99.721	99.719	99.719	99.719
MLP	RD-1	99.556	99.533	99.533	99.533
MLP	RD-2	99.414	99.400	99.400	99.400
MLP	RD-3	99.561	99.556	99.556	99.556
MLP	RD-4	99.571	99.567	99.567	99.567
MLP	RD-5	99.415	99.413	99.413	99.413
MLP	RD-6	99.255	99.244	99.243	99.244
MLP	RD-7	99.273	99.257	99.256	99.257
MLP	RD-8	98.724	98.692	98.686	98.692
MLP	RD-9	65.811	76.681	69.504	76.681
RBM+	RD-1	92.635	89.109	87.852	89.109
RBM+	RD-2	92.064	87.703	85.939	87.703
RBM+	RD-3	91.940	87.389	85.542	87.389
RBM+	RD-4	91.405	86.359	84.102	86.359
RBM+	RD-5	90.791	84.966	82.036	84.966
RBM+	RD-6	90.449	84.474	81.280	84.474
RBM+	RD-7	90.044	83.464	79.752	83.500
RBM+	RD-8	89.722	82.685	78.457	82.685
RBM+	RD-9	88.750	81.104	76.112	81.104
SGD	RD-1	97.960	97.595	97.483	97.600
SGD	RD-2	97.716	97.349	97.252	97.349
SGD	RD-3	97.561	97.102	96.986	97.102
SGD	RD-4	97.592	97.112	96.981	97.112
SGD	RD-5	97.049	96.404	96.108	96.404
SGD	RD-6	96.673	95.651	95.333	95.651
SGD	RD-7	96.044	94.587	93.822	94.587
SGD	RD-8	94.226	92.347	90.962	92.347
SGD	RD-9	89.172	89.077	86.240	89.077
RAN	RD-1	99.036	99.000	99.002	99.000
RAN	RD-2	99.024	99.000	99.001	99.000
RAN	RD-3	98.966	98.956	98.956	98.956
RAN	RD-4	99.108	99.100	99.100	99.100
RAN	RD-5	99.167	99.160	99.160	99.160
RAN	RD-6	99.184	99.178	99.178	99.178
RAN	RD-7	99.118	99.114	99.114	99.114
RAN	RD-8	99.213	99.208	99.208	99.208
RAN	RD-9	99.211	99.207	99.207	99.207

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE C.2: Observations with Mice Protein data for Convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	100.000	100.000	100.000	100.000
K-NN	RD-2	100.000	100.000	100.000	100.000
K-NN	RD-3	100.000	100.000	100.000	100.000
K-NN	RD-4	100.000	100.000	100.000	100.000
K-NN	RD-5	100.000	100.000	100.000	100.000
K-NN	RD-6	100.000	100.000	100.000	100.000
K-NN	RD-7	100.000	100.000	100.000	100.000
K-NN	RD-8	99.617	99.593	99.587	99.593
K-NN	RD-9	88.053	85.429	83.001	85.429
LR	RD-1	100.000	100.000	100.000	100.000
LR	RD-2	100.000	100.000	100.000	100.000
LR	RD-3	100.000	100.000	100.000	100.000
LR	RD-4	100.000	100.000	100.000	100.000
LR	RD-5	100.000	100.000	100.000	100.000
LR	RD-6	99.558	99.488	99.484	99.488
LR	RD-7	94.709	90.749	89.765	90.749
LR	RD-8	100.000	100.000	100.000	100.000
LR	RD-9	96.654	94.326	93.978	94.326
MLP	RD-1	100.000	100.000	100.000	100.000
MLP	RD-2	100.000	100.000	100.000	100.000
MLP	RD-3	100.000	100.000	100.000	100.000
MLP	RD-4	100.000	100.000	100.000	100.000
MLP	RD-5	100.000	100.000	100.000	100.000
MLP	RD-6	100.000	100.000	100.000	100.000
MLP	RD-7	95.973	94.832	93.817	94.800
MLP	RD-8	95.270	93.575	91.872	93.575
MLP	RD-9	95.652	95.654	94.786	95.654
RBM+	RD-1	89.874	93.333	91.058	93.333
RBM+	RD-2	2.630	16.216	4.525	16.216
RBM+	RD-3	59.688	73.028	64.029	73.028
RBM+	RD-4	2.654	16.290	4.564	16.290
RBM+	RD-5	29.103	47.568	33.870	47.568
RBM+	RD-6	99.995	99.995	99.995	99.995
RBM+	RD-7	4.429	18.780	6.566	18.780
RBM+	RD-8	99.993	99.992	99.992	99.992
RBM+	RD-9	2.656	16.298	4.568	16.298
SGD	RD-1	100.000	100.000	100.000	100.000
SGD	RD-2	99.994	99.994	99.994	100.000
SGD	RD-3	99.986	99.984	99.983	99.984
SGD	RD-4	99.987	99.986	99.986	99.986
SGD	RD-5	99.910	99.894	99.887	99.894
SGD	RD-6	99.714	99.650	99.612	99.650
SGD	RD-7	99.582	99.487	99.434	99.487
SGD	RD-8	98.440	98.107	97.820	98.107
SGD	RD-9	94.387	92.482	91.426	92.482
RAN	RD-1	100.000	100.000	100.000	100.000
RAN	RD-2	100.000	100.000	100.000	100.000
RAN	RD-3	100.000	100.000	100.000	100.000
RAN	RD-4	100.000	100.000	100.000	100.000
RAN	RD-5	100.000	100.000	100.000	100.000
RAN	RD-6	100.000	100.000	100.000	100.000
RAN	RD-7	99.995	99.994	99.994	99.994
RAN	RD-8	99.978	99.977	99.978	99.977
RAN	RD-9	99.850	99.826	99.829	99.826

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

Table C.3: Observations with Breast Cancer 569 for Convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	100.000	100.000	100.000	100.000
K-NN	RD-2	100.000	100.000	100.000	100.000
K-NN	RD-3	100.000	100.000	100.000	100.000
K-NN	RD-4	100.000	100.000	100.000	100.000
K-NN	RD-5	100.000	100.000	100.000	100.000
K-NN	RD-6	100.000	100.000	100.000	100.000
K-NN	RD-7	100.000	100.000	100.000	100.000
K-NN	RD-8	99.935	99.934	99.934	99.934
K-NN	RD-9	98.226	98.131	98.109	98.131
LR	RD-1	100.000	100.000	100.000	100.000
LR	RD-2	99.913	99.912	99.912	99.912
LR	RD-3	99.768	99.766	99.766	99.766
LR	RD-4	99.826	99.825	99.824	99.825
LR	RD-5	99.965	99.965	99.965	99.965
LR	RD-6	99.884	99.883	99.883	99.883
LR	RD-7	99.850	99.850	99.849	99.850
LR	RD-8	99.869	99.868	99.868	99.868
LR	RD-9	99.903	99.903	99.902	99.903
MLP	RD-1	99.829	99.825	99.824	99.825
MLP	RD-2	99.827	99.825	99.824	99.825
MLP	RD-3	98.970	98.947	98.944	98.900
MLP	RD-4	98.787	98.772	98.768	98.772
MLP	RD-5	98.990	98.982	98.981	98.982
MLP	RD-6	99.015	99.006	99.004	99.006
MLP	RD-7	97.272	97.218	97.202	97.218
MLP	RD-8	97.531	97.478	97.463	97.478
MLP	RD-9	97.796	97.758	97.746	97.758
RBM+	RD-1	97.376	97.269	97.277	97.269
RBM+	RD-2	96.259	96.193	96.197	96.193
RBM+	RD-3	94.419	94.390	94.391	94.390
RBM+	RD-4	95.232	95.140	95.155	95.140
RBM+	RD-5	94.420	94.389	94.387	94.389
RBM+	RD-6	92.795	92.778	92.760	92.778
RBM+	RD-7	90.950	90.926	90.867	90.926
RBM+	RD-8	92.194	92.151	92.093	92.151
RBM+	RD-9	88.794	88.338	88.021	88.338
SGD	RD-1	100.000	100.000	100.000	100.000
SGD	RD-2	99.983	99.982	99.982	99.982
SGD	RD-3	99.973	99.973	99.973	99.973
SGD	RD-4	99.951	99.950	99.950	99.950
SGD	RD-5	99.941	99.939	99.939	99.939
SGD	RD-6	99.864	99.798	99.794	99.798
SGD	RD-7	99.802	99.791	99.791	99.791
SGD	RD-8	99.689	99.460	99.403	99.460
SGD	RD-9	99.647	99.721	99.673	99.721
RAN	RD-1	94.139	94.035	93.960	94.035
RAN	RD-2	94.040	93.772	93.667	93.772
RAN	RD-3	92.937	92.752	92.650	92.752
RAN	RD-4	93.748	93.596	93.516	93.596
RAN	RD-5	93.249	93.078	92.986	93.078
RAN	RD-6	93.034	92.830	92.733	92.830
RAN	RD-7	92.834	92.675	92.588	92.675
RAN	RD-8	92.609	92.281	92.146	92.281
RAN	RD-9	92.921	92.738	92.648	92.738

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE C.4: Observations with Breast Cancer 669 for Convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	99.863	99.857	99.858	99.857
K-NN	RD-2	99.860	99.857	99.857	99.857
K-NN	RD-3	99.765	99.762	99.762	99.762
K-NN	RD-4	99.859	99.857	99.857	99.857
K-NN	RD-5	99.802	99.800	99.800	99.800
K-NN	RD-6	99.695	99.690	99.691	99.690
K-NN	RD-7	99.616	99.612	99.613	99.600
K-NN	RD-8	99.520	99.518	99.518	99.518
K-NN	RD-9	97.121	96.978	96.932	96.978
LR	RD-1	99.599	99.571	99.575	99.571
LR	RD-2	99.168	99.143	99.146	99.143
LR	RD-3	99.159	99.143	99.145	99.143
LR	RD-4	99.124	99.107	99.109	99.107
LR	RD-5	99.125	99.114	99.116	99.114
LR	RD-6	99.064	99.048	99.050	99.048
LR	RD-7	99.036	99.020	99.023	99.020
LR	RD-8	99.152	99.143	99.144	99.143
LR	RD-9	99.007	99.000	99.001	99.000
MLP	RD-1	99.863	99.857	99.858	99.857
MLP	RD-2	99.793	99.786	99.787	99.786
MLP	RD-3	99.436	99.429	99.429	99.429
MLP	RD-4	99.260	99.250	99.251	99.250
MLP	RD-5	99.089	99.086	99.086	99.086
MLP	RD-6	98.897	98.881	98.883	98.881
MLP	RD-7	98.698	98.673	98.677	98.673
MLP	RD-8	97.835	97.821	97.812	97.821
MLP	RD-9	97.766	97.746	97.735	97.746
RBM+	RD-1	97.780	97.714	97.698	97.714
RBM+	RD-2	98.027	98.000	97.998	98.000
RBM+	RD-3	97.679	97.665	97.664	97.665
RBM+	RD-4	97.307	97.296	97.290	97.296
RBM+	RD-5	97.169	97.150	97.139	97.150
RBM+	RD-6	96.375	96.371	96.356	96.371
RBM+	RD-7	96.147	96.130	96.108	96.130
RBM+	RD-8	94.465	94.249	94.145	94.249
RBM+	RD-9	86.523	83.456	81.729	83.456
SGD	RD-1	100.000	100.000	100.000	100.000
SGD	RD-2	99.967	99.967	99.967	99.967
SGD	RD-3	99.992	99.992	99.992	99.992
SGD	RD-4	99.988	99.988	99.988	99.988
SGD	RD-5	99.971	99.970	99.970	99.970
SGD	RD-6	99.872	99.869	99.869	99.869
SGD	RD-7	99.713	99.707	99.707	99.707
SGD	RD-8	99.868	99.866	99.866	99.866
SGD	RD-9	99.534	99.520	99.519	99.520
RAN	RD-1	95.121	95.000	94.936	95.000
RAN	RD-2	95.590	95.510	95.471	95.510
RAN	RD-3	95.201	95.143	95.100	95.143
RAN	RD-4	95.194	95.163	95.127	95.163
RAN	RD-5	95.344	95.308	95.273	95.308
RAN	RD-6	94.600	94.587	94.546	94.587
RAN	RD-7	95.321	95.315	95.288	95.315
RAN	RD-8	95.095	95.077	95.043	95.077
RAN	RD-9	95.132	95.101	95.063	95.101

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE C.5: Observations with Credit Approval data for Convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	95.404	95.373	95.374	95.373
K-NN	RD-2	95.477	95.448	95.443	95.448
K-NN	RD-3	95.313	95.300	95.301	95.300
K-NN	RD-4	95.336	95.318	95.314	95.318
K-NN	RD-5	95.805	95.796	95.795	95.796
K-NN	RD-6	95.366	95.350	95.350	95.350
K-NN	RD-7	95.625	95.610	95.611	95.610
K-NN	RD-8	95.397	95.385	95.385	95.385
K-NN	RD-9	95.589	95.575	95.574	95.575
LR	RD-1	94.680	94.627	94.628	94.627
LR	RD-2	94.506	94.478	94.473	94.478
LR	RD-3	94.717	94.700	94.699	94.700
LR	RD-4	94.808	94.794	94.790	94.794
LR	RD-5	95.477	95.465	95.466	95.465
LR	RD-6	95.232	95.225	95.225	95.225
LR	RD-7	95.468	95.460	95.460	95.460
LR	RD-8	95.261	95.253	95.254	95.253
LR	RD-9	95.392	95.383	95.384	95.383
MLP	RD-1	99.564	99.552	99.551	99.552
MLP	RD-2	99.197	99.179	99.178	99.179
MLP	RD-3	99.355	99.350	99.350	99.350
MLP	RD-4	98.122	98.090	98.087	98.090
MLP	RD-5	98.299	98.288	98.287	98.288
MLP	RD-6	98.324	98.300	98.298	98.300
MLP	RD-7	97.198	97.152	97.146	97.152
MLP	RD-8	96.178	96.098	96.088	96.098
MLP	RD-9	95.983	95.967	95.964	95.967
RBM+	RD-1	93.956	93.886	93.873	93.886
RBM+	RD-2	90.706	90.597	90.565	90.597
RBM+	RD-3	88.568	88.305	88.205	88.305
RBM+	RD-4	80.638	79.536	78.959	79.536
RBM+	RD-5	74.321	73.827	73.258	73.827
RBM+	RD-6	69.972	68.428	66.685	68.428
RBM+	RD-7	65.278	63.642	60.243	63.642
RBM+	RD-8	62.939	62.002	58.557	62.002
RBM+	RD-9	61.541	60.468	55.982	60.468
SGD	RD-1	99.995	99.995	99.995	99.995
SGD	RD-2	99.976	99.975	99.975	99.975
SGD	RD-3	99.993	99.993	99.993	99.993
SGD	RD-4	99.966	99.965	99.965	99.965
SGD	RD-5	99.902	99.880	99.874	99.880
SGD	RD-6	99.853	99.813	99.797	99.813
SGD	RD-7	99.850	99.842	99.841	99.842
SGD	RD-8	99.594	99.580	99.579	99.580
SGD	RD-9	98.789	98.744	98.744	98.744
RAN	RD-1	78.216	77.677	77.651	77.677
RAN	RD-2	79.039	78.562	78.523	78.562
RAN	RD-3	81.933	80.967	80.946	80.967
RAN	RD-4	79.599	79.090	79.053	79.090
RAN	RD-5	79.400	78.619	78.590	78.619
RAN	RD-6	79.876	78.791	78.739	78.791
RAN	RD-7	80.538	79.435	79.404	79.435
RAN	RD-8	82.433	81.303	81.305	81.303
RAN	RD-9	82.565	79.913	80.686	79.913

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE C.6: Observations with IRIS data for Convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	100.000	100.000	100.000	100.000
K-NN	RD-2	100.000	100.000	100.000	100.000
K-NN	RD-3	100.000	100.000	100.000	100.000
K-NN	RD-4	100.000	100.000	100.000	100.000
K-NN	RD-5	100.000	100.000	100.000	100.000
K-NN	RD-6	100.000	100.000	100.000	100.000
K-NN	RD-7	100.000	100.000	100.000	100.000
K-NN	RD-8	100.000	100.000	100.000	100.000
K-NN	RD-9	13.678	35.172	18.975	35.172
LR	RD-1	100.000	100.000	100.000	100.000
LR	RD-2	100.000	100.000	100.000	100.000
LR	RD-3	100.000	100.000	100.000	100.000
LR	RD-4	100.000	100.000	100.000	100.000
LR	RD-5	99.625	99.600	99.600	99.600
LR	RD-6	98.660	98.556	98.553	98.556
LR	RD-7	96.240	95.714	95.691	95.714
LR	RD-8	94.095	92.667	92.499	92.667
LR	RD-9	87.816	83.185	81.675	83.185
MLP	RD-1	95.575	94.000	93.720	94.000
MLP	RD-2	97.677	97.333	97.317	97.333
MLP	RD-3	97.206	96.889	96.878	96.889
MLP	RD-4	97.904	97.667	97.658	97.667
MLP	RD-5	97.829	97.600	97.592	97.600
MLP	RD-6	97.220	96.778	96.751	96.778
MLP	RD-7	97.302	97.048	97.040	97.048
MLP	RD-8	97.279	96.833	96.801	96.833
MLP	RD-9	97.818	97.556	97.543	97.556
RBM+	RD-1	97.746	97.356	97.009	97.356
RBM+	RD-2	97.245	96.633	96.558	96.633
RBM+	RD-3	82.388	79.341	74.181	79.341
RBM+	RD-4	85.148	77.950	74.091	77.950
RBM+	RD-5	71.755	70.627	61.090	70.627
RBM+	RD-6	77.097	69.485	60.012	69.485
RBM+	RD-7	71.091	67.635	56.447	67.635
RBM+	RD-8	72.441	70.506	61.139	70.506
RBM+	RD-9	63.353	67.133	55.413	67.133
SGD	RD-1	99.109	98.874	98.832	98.874
SGD	RD-2	98.542	98.356	98.239	98.356
SGD	RD-3	98.358	98.007	97.790	98.007
SGD	RD-4	97.814	97.194	96.934	97.194
SGD	RD-5	96.785	96.244	95.741	96.244
SGD	RD-6	96.287	95.589	94.932	95.589
SGD	RD-7	94.238	93.346	92.066	93.346
SGD	RD-8	89.889	89.631	87.141	89.631
SGD	RD-9	79.179	82.933	78.091	82.933
RAN	RD-1	93.778	93.333	93.306	93.333
RAN	RD-2	97.566	97.333	97.323	97.333
RAN	RD-3	95.364	95.111	95.099	95.111
RAN	RD-4	95.593	95.333	95.316	95.333
RAN	RD-5	94.905	94.533	94.500	94.533
RAN	RD-6	96.191	96.000	95.991	96.000
RAN	RD-7	94.961	94.857	94.852	94.857
RAN	RD-8	94.575	93.736	93.639	93.736
RAN	RD-9	94.190	93.259	93.138	93.259

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE C.7: Observations with UCIHAR data for Convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	99.971	99.971	99.971	99.971
K-NN	RD-2	99.971	99.971	99.971	99.971
K-NN	RD-3	99.974	99.974	99.974	99.974
K-NN	RD-4	99.981	99.981	99.981	99.981
K-NN	RD-5	99.967	99.967	99.967	99.967
K-NN	RD-6	99.966	99.966	99.966	99.966
K-NN	RD-7	99.956	99.956	99.956	99.956
K-NN	RD-8	99.937	99.937	99.937	99.937
K-NN	RD-9	99.918	99.918	99.918	99.918
LR	RD-1	99.971	99.971	99.971	99.971
LR	RD-2	99.985	99.985	99.985	99.985
LR	RD-3	99.984	99.984	99.984	99.984
LR	RD-4	99.985	99.985	99.985	99.985
LR	RD-5	99.979	99.979	99.979	99.979
LR	RD-6	99.969	99.969	99.969	99.969
LR	RD-7	99.972	99.972	99.972	99.972
LR	RD-8	99.949	99.949	99.949	99.949
LR	RD-9	99.940	99.940	99.940	99.940
MLP	RD-1	99.961	99.961	99.961	99.961
MLP	RD-2	99.981	99.981	99.981	99.981
MLP	RD-3	99.968	99.968	99.968	99.968
MLP	RD-4	99.976	99.976	99.976	99.976
MLP	RD-5	99.963	99.963	99.963	99.963
MLP	RD-6	99.963	99.963	99.963	99.963
MLP	RD-7	99.964	99.964	99.964	99.964
MLP	RD-8	99.936	99.936	99.936	99.936
MLP	RD-9	99.924	99.923	99.923	99.923
RBM+	RD-1	99.831	99.829	99.829	99.829
RBM+	RD-2	99.767	99.766	99.766	99.766
RBM+	RD-3	99.743	99.741	99.741	99.741
RBM+	RD-4	99.749	99.747	99.747	99.747
RBM+	RD-5	99.720	99.717	99.718	99.717
RBM+	RD-6	99.731	99.728	99.728	99.728
RBM+	RD-7	99.664	99.659	99.659	99.659
RBM+	RD-8	99.575	99.566	99.566	99.566
RBM+	RD-9	99.363	99.345	99.346	99.345
SGD	RD-1	99.981	99.981	99.981	99.981
SGD	RD-2	99.995	99.995	99.995	99.995
SGD	RD-3	99.985	99.985	99.985	99.985
SGD	RD-4	99.987	99.987	99.987	99.987
SGD	RD-5	99.986	99.986	99.986	99.986
SGD	RD-6	99.980	99.980	99.980	99.980
SGD	RD-7	99.977	99.977	99.977	99.977
SGD	RD-8	99.956	99.956	99.956	99.956
SGD	RD-9	99.951	99.951	99.951	99.951
RAN	RD-1	99.874	99.874	99.874	99.874
RAN	RD-2	99.841	99.840	99.840	99.840
RAN	RD-3	99.855	99.854	99.854	99.854
RAN	RD-4	99.867	99.867	99.867	99.867
RAN	RD-5	99.861	99.860	99.860	99.860
RAN	RD-6	99.853	99.853	99.853	99.853
RAN	RD-7	99.856	99.856	99.856	99.856
RAN	RD-8	99.845	99.845	99.845	99.845
RAN	RD-9	99.851	99.850	99.850	99.850

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE C.8: Observations with Mamographic Mass data for Convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	100.000	100.000	100.000	100.000
K-NN	RD-2	100.000	100.000	100.000	100.000
K-NN	RD-3	100.000	100.000	100.000	100.000
K-NN	RD-4	100.000	100.000	100.000	100.000
K-NN	RD-5	100.000	100.000	100.000	100.000
K-NN	RD-6	100.000	100.000	100.000	100.000
K-NN	RD-7	99.966	99.966	99.966	99.966
K-NN	RD-8	99.515	99.504	99.503	99.500
K-NN	RD-9	97.349	97.296	97.295	97.296
LR	RD-1	99.540	99.524	99.523	99.524
LR	RD-2	99.647	99.641	99.641	99.641
LR	RD-3	99.566	99.560	99.560	99.560
LR	RD-4	99.319	99.309	99.309	99.309
LR	RD-5	99.458	99.447	99.447	99.447
LR	RD-6	99.150	99.138	99.138	99.138
LR	RD-7	99.458	99.450	99.450	99.450
LR	RD-8	98.787	98.767	98.766	98.767
LR	RD-9	99.803	99.799	99.799	99.800
MLP	RD-1	100.000	100.000	100.000	100.000
MLP	RD-2	100.000	100.000	100.000	100.000
MLP	RD-3	100.000	100.000	100.000	100.000
MLP	RD-4	100.000	100.000	100.000	100.000
MLP	RD-5	100.000	100.000	100.000	100.000
MLP	RD-6	99.920	99.920	99.920	99.920
MLP	RD-7	99.914	99.914	99.914	99.914
MLP	RD-8	95.148	94.602	94.595	94.602
MLP	RD-9	95.220	94.693	94.686	94.693
RBM+	RD-1	93.588	93.504	93.498	93.504
RBM+	RD-2	93.151	93.086	93.080	93.086
RBM+	RD-3	92.793	92.752	92.751	92.752
RBM+	RD-4	92.682	92.296	92.264	92.296
RBM+	RD-5	93.771	92.748	92.609	92.748
RBM+	RD-6	93.611	93.410	93.400	93.410
RBM+	RD-7	54.903	59.941	48.258	59.941
RBM+	RD-8	56.489	57.673	45.082	57.673
RBM+	RD-9	92.668	91.180	90.806	91.180
SGD	RD-1	99.996	99.996	99.996	99.996
SGD	RD-2	99.922	99.848	99.812	99.800
SGD	RD-3	99.989	99.989	99.989	99.989
SGD	RD-4	99.939	99.915	99.911	99.900
SGD	RD-5	99.994	99.994	99.994	99.994
SGD	RD-6	99.908	99.820	99.764	99.820
SGD	RD-7	99.984	99.983	99.983	99.983
SGD	RD-8	99.977	99.976	99.976	100.000
SGD	RD-9	99.958	99.957	99.957	100.000
RAN	RD-1	83.158	82.143	82.035	82.143
RAN	RD-2	80.461	79.281	79.154	79.300
RAN	RD-3	80.560	79.440	79.327	79.440
RAN	RD-4	80.287	79.249	79.148	79.249
RAN	RD-5	80.308	79.183	79.060	79.183
RAN	RD-6	79.994	78.792	78.656	78.800
RAN	RD-7	80.242	79.226	79.124	79.200
RAN	RD-8	80.082	78.992	78.880	78.992
RAN	RD-9	80.335	79.415	79.322	79.415

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE C.9: Observations with Wine Recognition data for Convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	100.000	100.000	100.000	100.000
K-NN	RD-2	100.000	100.000	100.000	100.000
K-NN	RD-3	99.649	99.630	99.630	99.630
K-NN	RD-4	99.736	99.722	99.723	99.722
K-NN	RD-5	99.349	99.326	99.325	99.326
K-NN	RD-6	99.457	99.439	99.439	99.439
K-NN	RD-7	98.922	98.880	98.880	98.880
K-NN	RD-8	98.102	97.972	97.973	97.972
K-NN	RD-9	21.417	40.977	25.289	40.977
LR	RD-1	96.190	95.000	94.998	95.000
LR	RD-2	95.458	94.722	94.700	94.722
LR	RD-3	95.391	94.630	94.580	94.630
LR	RD-4	95.168	94.583	94.516	94.583
LR	RD-5	94.453	93.708	93.577	93.708
LR	RD-6	93.240	92.243	92.062	92.243
LR	RD-7	93.160	92.080	91.914	92.100
LR	RD-8	92.634	91.329	91.106	91.329
LR	RD-9	91.606	89.876	89.552	89.876
MLP	RD-1	96.574	96.111	96.088	96.111
MLP	RD-2	98.180	98.056	98.053	98.056
MLP	RD-3	97.899	97.778	97.776	97.778
MLP	RD-4	97.288	97.222	97.215	97.222
MLP	RD-5	97.837	97.753	97.749	97.753
MLP	RD-6	97.505	97.477	97.474	97.477
MLP	RD-7	97.655	97.600	97.599	97.600
MLP	RD-8	97.030	96.993	96.995	96.993
MLP	RD-9	97.003	96.957	96.956	96.957
RBM+	RD-1	99.984	99.981	99.981	99.981
RBM+	RD-2	45.821	60.389	51.077	60.389
RBM+	RD-3	49.296	65.247	55.565	65.247
RBM+	RD-4	73.371	73.005	67.026	73.005
RBM+	RD-5	85.943	83.322	82.103	83.322
RBM+	RD-6	16.150	40.187	23.040	40.187
RBM+	RD-7	44.238	60.235	50.783	60.235
RBM+	RD-8	45.219	61.800	52.071	61.800
RBM+	RD-9	43.966	59.327	50.005	59.327
SGD	RD-1	98.821	98.448	98.452	98.448
SGD	RD-2	98.493	98.296	98.303	98.296
SGD	RD-3	98.417	98.247	98.248	98.247
SGD	RD-4	98.737	98.625	98.628	98.625
SGD	RD-5	98.679	98.566	98.563	98.566
SGD	RD-6	98.103	97.960	97.956	98.000
SGD	RD-7	97.641	97.421	97.413	97.421
SGD	RD-8	97.521	97.298	97.271	97.298
SGD	RD-9	96.720	96.369	96.336	96.369
RAN	RD-1	96.324	95.185	95.181	95.185
RAN	RD-2	96.094	95.509	95.457	95.509
RAN	RD-3	95.795	95.204	95.168	95.204
RAN	RD-4	95.071	94.759	94.721	94.759
RAN	RD-5	95.317	94.861	94.805	94.900
RAN	RD-6	94.997	94.645	94.593	94.645
RAN	RD-7	94.343	93.763	93.701	93.763
RAN	RD-8	94.350	93.704	93.630	93.704
RAN	RD-9	92.993	92.304	92.234	92.304

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE C.10: Observations with Glass Identification data for Convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	99.571	99.545	99.527	99.545
K-NN	RD-2	99.549	99.535	99.527	99.535
K-NN	RD-3	99.698	99.692	99.689	99.692
K-NN	RD-4	98.740	98.721	98.710	98.721
K-NN	RD-5	98.721	98.692	98.677	98.692
K-NN	RD-6	98.344	98.295	98.265	98.295
K-NN	RD-7	97.561	97.467	97.407	97.467
K-NN	RD-8	95.287	94.942	94.688	94.942
K-NN	RD-9	59.288	76.852	66.878	76.852
LR	RD-1	99.621	99.545	99.559	99.545
LR	RD-2	99.787	99.767	99.771	99.767
LR	RD-3	99.421	99.385	99.391	99.385
LR	RD-4	99.445	99.419	99.423	99.419
LR	RD-5	99.461	99.439	99.443	99.439
LR	RD-6	99.625	99.612	99.614	99.612
LR	RD-7	99.546	99.533	99.536	99.533
LR	RD-8	99.598	99.593	99.593	99.593
LR	RD-9	99.152	99.119	99.123	99.119
MLP	RD-1	97.040	96.818	96.636	96.818
MLP	RD-2	92.349	91.395	90.663	91.395
MLP	RD-3	93.948	93.385	92.973	93.385
MLP	RD-4	93.474	92.791	92.248	92.791
MLP	RD-5	94.003	93.551	93.169	93.551
MLP	RD-6	93.968	93.488	93.064	93.488
MLP	RD-7	93.453	93.000	92.567	93.000
MLP	RD-8	93.658	93.256	92.832	93.256
MLP	RD-9	92.159	91.865	91.502	91.865
RBM+	RD-1	90.698	90.455	89.895	90.455
RBM+	RD-2	88.029	86.860	85.372	86.860
RBM+	RD-3	87.924	87.390	86.116	87.390
RBM+	RD-4	86.824	86.167	84.464	86.167
RBM+	RD-5	87.238	86.903	85.526	86.903
RBM+	RD-6	86.359	85.160	82.786	85.160
RBM+	RD-7	83.608	81.651	77.312	81.651
RBM+	RD-8	74.515	76.967	68.130	76.967
RBM+	RD-9	58.012	76.166	65.861	76.166
SGD	RD-1	99.467	99.436	99.412	99.436
SGD	RD-2	98.060	97.961	97.903	97.961
SGD	RD-3	98.386	98.328	98.294	98.328
SGD	RD-4	97.455	97.345	97.276	97.345
SGD	RD-5	98.006	97.935	97.889	97.935
SGD	RD-6	97.804	97.724	97.672	97.724
SGD	RD-7	97.543	97.453	97.395	97.453
SGD	RD-8	97.483	97.380	97.312	97.380
SGD	RD-9	97.362	97.250	97.182	97.250
RAN	RD-1	93.249	90.515	91.055	90.515
RAN	RD-2	89.642	89.535	89.459	89.535
RAN	RD-3	90.613	90.128	90.211	90.128
RAN	RD-4	90.400	89.783	89.929	89.783
RAN	RD-5	90.300	89.551	89.765	89.551
RAN	RD-6	90.466	89.837	90.008	89.837
RAN	RD-7	89.812	89.667	89.685	89.667
RAN	RD-8	89.913	86.711	87.289	86.711
RAN	RD-9	89.417	88.214	88.210	88.214

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

Appendix D

Observations with Non-convex Abstract Concept Modeling

This Appendix provides the observations of experiments of Chapter 4 showing the classification performance of RANs modeling along with five classifiers (Restricted Boltzmann Machine pipelined with Logistic regression, Logistic Regression, Multi Layer Perceptron, K Nearest Neighbor and Stochastic Gradient Descent). The investigations were carried out with one artificially generated Toy-data problem and six benchmark datasets obtained from the UCI machine learning repository.

TABLE D.1: Observations with Toy-data for Non-convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	100.000	100.000	100.000	100.000
K-NN	RD-2	100.000	100.000	100.000	100.000
K-NN	RD-3	100.000	100.000	100.000	100.000
K-NN	RD-4	100.000	100.000	100.000	100.000
K-NN	RD-5	100.000	100.000	100.000	100.000
K-NN	RD-6	100.000	100.000	100.000	100.000
K-NN	RD-7	100.000	100.000	100.000	100.000
K-NN	RD-8	100.000	100.000	100.000	100.000
K-NN	RD-9	99.983	99.983	99.983	99.983
LR	RD-1	100.000	100.000	100.000	100.000
LR	RD-2	100.000	100.000	100.000	100.000
LR	RD-3	100.000	100.000	100.000	100.000
LR	RD-4	100.000	100.000	100.000	100.000
LR	RD-5	100.000	100.000	100.000	100.000
LR	RD-6	100.000	100.000	100.000	100.000
LR	RD-7	99.997	99.997	99.997	99.997
LR	RD-8	99.998	99.998	99.998	99.998
LR	RD-9	99.994	99.994	99.994	99.994
MLP	RD-1	100.000	100.000	100.000	100.000
MLP	RD-2	100.000	100.000	100.000	100.000
MLP	RD-3	100.000	100.000	100.000	100.000
MLP	RD-4	100.000	100.000	100.000	100.000
MLP	RD-5	100.000	100.000	100.000	100.000
MLP	RD-6	100.000	100.000	100.000	100.000
MLP	RD-7	99.931	99.924	99.924	99.924
MLP	RD-8	99.979	99.979	99.979	99.979
MLP	RD-9	95.913	95.430	95.388	95.430
RBM+	RD-1	66.187	65.398	55.320	65.398
RBM+	RD-2	70.698	65.937	56.436	65.937
RBM+	RD-3	75.065	66.635	58.277	66.635
RBM+	RD-4	76.478	68.040	60.868	68.040
RBM+	RD-5	81.654	71.262	66.310	71.262
RBM+	RD-6	84.641	73.989	70.593	73.989
RBM+	RD-7	86.929	79.259	78.055	79.259
RBM+	RD-8	89.063	83.101	82.677	83.101
RBM+	RD-9	90.158	85.742	85.779	85.742
SGD	RD-1	100.000	100.000	100.000	100.000
SGD	RD-2	99.998	99.998	99.998	99.998
SGD	RD-3	99.995	99.995	99.995	99.995
SGD	RD-4	99.992	99.992	99.992	99.992
SGD	RD-5	99.993	99.993	99.993	99.993
SGD	RD-6	99.990	99.990	99.990	99.990
SGD	RD-7	99.981	99.981	99.981	99.981
SGD	RD-8	99.967	99.967	99.967	99.967
SGD	RD-9	99.940	99.940	99.940	99.940
RAN	RD-1	98.718	98.581	98.578	98.581
RAN	RD-2	98.761	98.629	98.626	98.629
RAN	RD-3	98.712	98.577	98.574	98.577
RAN	RD-4	98.737	98.608	98.606	98.608
RAN	RD-5	98.773	98.647	98.644	98.647
RAN	RD-6	98.148	97.658	97.572	97.658
RAN	RD-7	98.625	98.475	98.472	98.475
RAN	RD-8	98.646	98.502	98.499	98.502
RAN	RD-9	98.865	98.753	98.750	98.753

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE D.2: Observations with UCIHAR data for Non-convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	99.981	99.981	99.981	99.981
K-NN	RD-2	99.952	99.951	99.951	99.951
K-NN	RD-3	99.964	99.964	99.964	99.964
K-NN	RD-4	99.949	99.949	99.949	99.949
K-NN	RD-5	99.944	99.944	99.944	99.944
K-NN	RD-6	99.947	99.947	99.947	99.947
K-NN	RD-7	99.927	99.926	99.926	99.926
K-NN	RD-8	99.926	99.926	99.926	99.926
K-NN	RD-9	99.874	99.874	99.874	99.874
LR	RD-1	99.578	99.573	99.573	99.573
LR	RD-2	99.392	99.383	99.384	99.383
LR	RD-3	99.309	99.298	99.298	99.298
LR	RD-4	99.394	99.386	99.386	99.386
LR	RD-5	99.735	99.732	99.732	99.732
LR	RD-6	99.850	99.850	99.850	99.850
LR	RD-7	99.686	99.684	99.684	99.684
LR	RD-8	99.353	99.343	99.344	99.343
LR	RD-9	99.853	99.852	99.852	99.852
MLP	RD-1	99.990	99.990	99.990	99.990
MLP	RD-2	99.961	99.961	99.961	99.961
MLP	RD-3	99.981	99.981	99.981	99.981
MLP	RD-4	99.964	99.964	99.964	99.964
MLP	RD-5	99.969	99.969	99.969	99.969
MLP	RD-6	99.973	99.972	99.972	99.972
MLP	RD-7	99.943	99.943	99.943	99.943
MLP	RD-8	99.949	99.949	99.949	99.949
MLP	RD-9	99.915	99.915	99.915	99.915
RBM+	RD-1	99.932	99.932	99.932	99.932
RBM+	RD-2	99.919	99.919	99.919	99.919
RBM+	RD-3	99.902	99.902	99.902	99.902
RBM+	RD-4	99.871	99.871	99.871	99.871
RBM+	RD-5	99.890	99.890	99.890	99.890
RBM+	RD-6	99.868	99.867	99.867	99.867
RBM+	RD-7	99.851	99.851	99.851	99.851
RBM+	RD-8	99.862	99.861	99.861	99.861
RBM+	RD-9	99.747	99.745	99.745	99.745
SGD	RD-1	99.989	99.989	99.989	99.989
SGD	RD-2	99.951	99.951	99.951	99.951
SGD	RD-3	99.957	99.957	99.957	99.957
SGD	RD-4	99.971	99.971	99.971	99.971
SGD	RD-5	99.971	99.971	99.971	99.971
SGD	RD-6	99.959	99.959	99.959	99.959
SGD	RD-7	99.938	99.938	99.938	99.938
SGD	RD-8	99.937	99.937	99.937	99.937
SGD	RD-9	99.882	99.882	99.882	99.882
RAN	RD-1	97.111	96.718	96.677	96.718
RAN	RD-2	97.023	96.628	96.588	96.628
RAN	RD-3	97.020	96.599	96.554	96.599
RAN	RD-4	96.833	96.378	96.328	96.378
RAN	RD-5	97.145	96.753	96.711	96.753
RAN	RD-6	96.655	96.199	96.146	96.199
RAN	RD-7	96.741	96.298	96.258	96.298
RAN	RD-8	97.255	96.823	96.780	96.823
RAN	RD-9	97.096	96.876	96.861	96.876

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE D.3: Observations with Breast Cancer 669 for Non-convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	99.188	99.143	99.148	99.143
K-NN	RD-2	99.586	99.571	99.573	99.571
K-NN	RD-3	99.487	99.476	99.477	99.476
K-NN	RD-4	99.719	99.714	99.715	99.714
K-NN	RD-5	99.747	99.743	99.743	99.743
K-NN	RD-6	99.719	99.714	99.715	99.714
K-NN	RD-7	99.636	99.633	99.633	99.633
K-NN	RD-8	99.218	99.214	99.213	99.214
K-NN	RD-9	96.902	96.778	96.736	96.778
LR	RD-1	98.638	98.571	98.579	98.571
LR	RD-2	99.031	99.000	99.004	99.000
LR	RD-3	99.020	99.000	99.003	99.000
LR	RD-4	99.124	99.107	99.109	99.107
LR	RD-5	98.960	98.943	98.945	98.943
LR	RD-6	99.157	99.143	99.145	99.143
LR	RD-7	99.138	99.122	99.125	99.122
LR	RD-8	99.170	99.161	99.162	99.161
LR	RD-9	98.993	98.984	98.986	98.984
MLP	RD-1	99.325	99.286	99.290	99.286
MLP	RD-2	99.586	99.571	99.573	99.571
MLP	RD-3	99.675	99.667	99.668	99.667
MLP	RD-4	99.401	99.393	99.394	99.393
MLP	RD-5	98.898	98.886	98.888	98.886
MLP	RD-6	98.920	98.905	98.907	98.905
MLP	RD-7	98.717	98.694	98.698	98.694
MLP	RD-8	97.808	97.804	97.797	97.804
MLP	RD-9	97.514	97.508	97.499	97.508
RBM+	RD-1	97.994	97.857	97.873	97.857
RBM+	RD-2	98.199	98.136	98.143	98.136
RBM+	RD-3	97.973	97.919	97.927	97.919
RBM+	RD-4	97.906	97.871	97.877	97.871
RBM+	RD-5	97.799	97.771	97.777	97.771
RBM+	RD-6	97.707	97.695	97.698	97.695
RBM+	RD-7	97.444	97.437	97.438	97.437
RBM+	RD-8	96.883	96.880	96.872	96.880
RBM+	RD-9	95.256	95.189	95.138	95.189
SGD	RD-1	99.986	99.986	99.986	99.986
SGD	RD-2	99.986	99.986	99.986	99.986
SGD	RD-3	99.996	99.996	99.996	99.996
SGD	RD-4	99.975	99.974	99.974	99.974
SGD	RD-5	99.975	99.974	99.974	99.974
SGD	RD-6	99.918	99.917	99.917	99.917
SGD	RD-7	99.737	99.727	99.728	99.727
SGD	RD-8	99.785	99.780	99.780	99.780
SGD	RD-9	99.234	99.200	99.203	99.200
RAN	RD-1	93.273	85.700	88.162	85.700
RAN	RD-2	92.514	86.857	88.525	86.857
RAN	RD-3	93.201	88.181	89.612	88.181
RAN	RD-4	92.334	84.621	87.191	84.621
RAN	RD-5	93.308	86.483	88.726	86.483
RAN	RD-6	91.495	80.405	83.017	80.405
RAN	RD-7	89.391	81.467	83.314	81.467
RAN	RD-8	88.491	76.866	77.695	76.866
RAN	RD-9	81.202	64.435	64.360	64.435

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE D.4: Observations with Breast Cancer 569 data for Non-convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	100.000	100.000	100.000	100.000
K-NN	RD-2	100.000	100.000	100.000	100.000
K-NN	RD-3	100.000	100.000	100.000	100.000
K-NN	RD-4	100.000	100.000	100.000	100.000
K-NN	RD-5	100.000	100.000	100.000	100.000
K-NN	RD-6	100.000	100.000	100.000	100.000
K-NN	RD-7	100.000	100.000	100.000	100.000
K-NN	RD-8	99.935	99.934	99.934	99.934
K-NN	RD-9	97.869	97.778	97.759	97.778
LR	RD-1	99.829	99.825	99.824	99.825
LR	RD-2	99.740	99.737	99.736	99.737
LR	RD-3	99.710	99.708	99.707	99.708
LR	RD-4	99.826	99.825	99.824	99.825
LR	RD-5	99.826	99.825	99.824	99.825
LR	RD-6	99.855	99.854	99.854	99.854
LR	RD-7	99.875	99.875	99.875	99.875
LR	RD-8	99.825	99.825	99.824	99.825
LR	RD-9	99.786	99.786	99.785	99.786
MLP	RD-1	99.497	99.474	99.469	99.474
MLP	RD-2	99.399	99.386	99.383	99.386
MLP	RD-3	98.456	98.421	98.413	98.421
MLP	RD-4	98.835	98.816	98.812	98.816
MLP	RD-5	98.647	98.632	98.628	98.632
MLP	RD-6	98.763	98.743	98.738	98.743
MLP	RD-7	97.178	97.093	97.070	97.093
MLP	RD-8	97.761	97.741	97.732	97.741
MLP	RD-9	97.005	96.940	96.919	96.940
RBM+	RD-1	97.216	97.018	97.035	97.018
RBM+	RD-2	96.094	96.044	96.039	96.044
RBM+	RD-3	95.054	95.023	95.008	95.023
RBM+	RD-4	94.985	94.917	94.921	94.917
RBM+	RD-5	94.103	94.063	94.054	94.063
RBM+	RD-6	92.594	92.558	92.517	92.558
RBM+	RD-7	90.866	90.764	90.649	90.764
RBM+	RD-8	92.236	92.213	92.162	92.213
RBM+	RD-9	88.438	87.766	87.364	87.766
SGD	RD-1	100.000	100.000	100.000	100.000
SGD	RD-2	99.983	99.982	99.982	99.982
SGD	RD-3	99.807	99.759	99.737	99.759
SGD	RD-4	99.914	99.912	99.912	99.912
SGD	RD-5	99.914	99.912	99.912	99.912
SGD	RD-6	99.936	99.936	99.936	99.936
SGD	RD-7	99.801	99.787	99.788	99.787
SGD	RD-8	99.863	99.860	99.860	99.860
SGD	RD-9	99.625	99.598	99.591	99.598
RAN	RD-1	97.617	93.158	94.995	93.158
RAN	RD-2	96.451	92.643	94.012	92.643
RAN	RD-3	96.629	93.334	94.314	93.334
RAN	RD-4	95.791	92.241	93.209	92.241
RAN	RD-5	97.526	92.867	94.654	92.867
RAN	RD-6	97.078	93.363	94.745	93.363
RAN	RD-7	96.442	92.732	93.888	92.732
RAN	RD-8	95.380	90.588	92.226	90.588
RAN	RD-9	95.753	91.823	92.902	91.823

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE D.5: Observations with IRIS for Non-convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	98.963	98.533	98.373	98.533
K-NN	RD-2	93.690	91.333	90.242	91.333
K-NN	RD-3	100.000	100.000	100.000	100.000
K-NN	RD-4	100.000	100.000	100.000	100.000
K-NN	RD-5	100.000	100.000	100.000	100.000
K-NN	RD-6	100.000	100.000	100.000	100.000
K-NN	RD-7	100.000	100.000	100.000	100.000
K-NN	RD-8	100.000	100.000	100.000	100.000
K-NN	RD-9	11.111	33.333	16.667	33.333
LR	RD-1	91.069	87.467	86.171	87.467
LR	RD-2	90.893	87.700	86.271	87.700
LR	RD-3	85.710	79.778	76.831	79.778
LR	RD-4	85.171	78.167	74.824	78.167
LR	RD-5	84.072	76.267	71.842	76.267
LR	RD-6	84.165	75.889	71.518	75.889
LR	RD-7	83.390	74.381	69.019	74.381
LR	RD-8	82.747	72.333	65.615	72.333
LR	RD-9	81.904	71.259	63.424	71.259
MLP	RD-1	96.621	94.267	94.724	94.267
MLP	RD-2	96.779	94.867	95.139	94.867
MLP	RD-3	97.341	96.889	96.852	96.889
MLP	RD-4	98.297	98.167	98.164	98.167
MLP	RD-5	97.105	96.667	96.638	96.667
MLP	RD-6	97.093	96.667	96.643	96.667
MLP	RD-7	97.294	96.952	96.935	96.952
MLP	RD-8	96.920	96.500	96.480	96.500
MLP	RD-9	97.541	97.259	97.246	97.259
RBM+	RD-1	79.825	82.533	77.465	82.533
RBM+	RD-2	72.794	73.333	65.060	73.333
RBM+	RD-3	75.070	70.289	61.460	70.289
RBM+	RD-4	73.232	70.133	60.840	70.133
RBM+	RD-5	67.174	68.013	57.993	68.013
RBM+	RD-6	67.483	69.778	60.267	69.778
RBM+	RD-7	65.257	67.324	56.726	67.324
RBM+	RD-8	64.905	68.383	59.347	68.383
RBM+	RD-9	66.670	67.674	56.289	67.674
SGD	RD-1	98.526	98.333	98.144	98.333
SGD	RD-2	94.282	93.700	92.452	93.700
SGD	RD-3	92.504	92.367	90.699	92.367
SGD	RD-4	87.852	87.333	84.695	87.333
SGD	RD-5	87.852	87.333	84.695	87.333
SGD	RD-6	84.447	84.733	81.278	84.733
SGD	RD-7	80.889	82.829	78.456	82.829
SGD	RD-8	67.189	74.008	66.155	74.008
SGD	RD-9	65.127	72.519	64.407	72.519
RAN	RD-1	83.663	81.933	79.035	81.933
RAN	RD-2	90.066	83.567	83.739	83.567
RAN	RD-3	95.692	88.689	90.829	88.689
RAN	RD-4	94.338	87.967	89.483	87.967
RAN	RD-5	93.710	88.573	89.385	88.573
RAN	RD-6	94.772	89.506	90.588	89.506
RAN	RD-7	93.010	87.076	87.578	87.076
RAN	RD-8	93.968	86.958	88.260	86.958
RAN	RD-9	93.562	82.556	84.594	82.556

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE D.6: Observations with Mice Protein Data for Non-convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	100.000	100.000	100.000	100.000
K-NN	RD-2	100.000	100.000	100.000	100.000
K-NN	RD-3	100.000	100.000	100.000	100.000
K-NN	RD-4	100.000	100.000	100.000	100.000
K-NN	RD-5	100.000	100.000	100.000	100.000
K-NN	RD-6	100.000	100.000	100.000	100.000
K-NN	RD-7	99.267	99.225	99.205	99.225
K-NN	RD-8	88.555	87.670	84.398	87.670
K-NN	RD-9	30.560	34.245	23.097	34.245
LR	RD-1	100.000	100.000	100.000	100.000
LR	RD-2	100.000	100.000	100.000	100.000
LR	RD-3	100.000	100.000	100.000	100.000
LR	RD-4	100.000	100.000	100.000	100.000
LR	RD-5	100.000	100.000	100.000	100.000
LR	RD-6	100.000	100.000	100.000	100.000
LR	RD-7	100.000	100.000	100.000	100.000
LR	RD-8	100.000	100.000	100.000	100.000
LR	RD-9	99.708	99.658	99.658	99.658
MLP	RD-1	100.000	100.000	100.000	100.000
MLP	RD-2	100.000	100.000	100.000	100.000
MLP	RD-3	100.000	100.000	100.000	100.000
MLP	RD-4	100.000	100.000	100.000	100.000
MLP	RD-5	100.000	100.000	100.000	100.000
MLP	RD-6	100.000	100.000	100.000	100.000
MLP	RD-7	99.950	99.948	99.948	99.948
MLP	RD-8	99.709	99.683	99.674	99.683
MLP	RD-9	99.582	99.557	99.551	99.557
RBM+	RD-1	91.695	93.143	91.252	93.143
RBM+	RD-2	2.630	16.216	4.525	16.216
RBM+	RD-3	76.394	81.096	75.682	81.096
RBM+	RD-4	2.654	16.290	4.564	16.290
RBM+	RD-5	38.115	54.808	42.136	54.808
RBM+	RD-6	16.122	31.572	17.822	31.572
RBM+	RD-7	11.442	20.106	9.758	20.106
RBM+	RD-8	2.654	16.290	4.564	16.290
RBM+	RD-9	3.145	16.378	4.715	16.378
SGD	RD-1	99.348	99.482	99.347	99.482
SGD	RD-2	99.629	99.757	99.673	99.757
SGD	RD-3	98.743	98.977	98.758	98.977
SGD	RD-4	99.585	99.605	99.504	99.605
SGD	RD-5	99.585	99.605	99.504	99.605
SGD	RD-6	99.555	99.699	99.606	99.699
SGD	RD-7	97.294	97.605	97.219	97.605
SGD	RD-8	99.014	99.063	98.912	99.063
SGD	RD-9	96.552	95.537	94.770	95.537
RAN	RD-1	99.786	80.089	86.509	80.089
RAN	RD-2	99.572	79.099	86.295	79.099
RAN	RD-3	100.000	79.464	86.847	79.464
RAN	RD-4	100.000	78.195	85.861	78.195
RAN	RD-5	99.904	79.236	86.655	79.236
RAN	RD-6	99.981	80.502	87.639	80.502
RAN	RD-7	99.907	79.829	86.925	79.829
RAN	RD-8	100.000	78.767	86.160	78.767
RAN	RD-9	99.974	80.891	87.575	80.891

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

TABLE D.7: Observations with Credit Approval data for Non-convex Concept Modeling

Algo	**RD**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Accuracy (%)**
K-NN	RD-1	96.754	96.716	96.709	96.716
K-NN	RD-2	96.539	96.493	96.496	96.493
K-NN	RD-3	95.863	95.850	95.850	95.850
K-NN	RD-4	95.746	95.730	95.730	95.730
K-NN	RD-5	95.353	95.345	95.343	95.345
K-NN	RD-6	95.488	95.475	95.472	95.475
K-NN	RD-7	95.392	95.375	95.375	95.375
K-NN	RD-8	95.247	95.235	95.236	95.235
K-NN	RD-9	95.374	95.367	95.368	95.367
LR	RD-1	95.865	95.821	95.813	95.821
LR	RD-2	96.109	96.045	96.048	96.045
LR	RD-3	95.413	95.400	95.401	95.400
LR	RD-4	95.222	95.206	95.208	95.206
LR	RD-5	94.903	94.895	94.893	94.895
LR	RD-6	95.188	95.175	95.174	95.175
LR	RD-7	95.157	95.139	95.139	95.139
LR	RD-8	95.411	95.403	95.403	95.403
LR	RD-9	95.127	95.117	95.117	95.117
MLP	RD-1	100.000	100.000	100.000	100.000
MLP	RD-2	99.853	99.851	99.851	99.851
MLP	RD-3	99.751	99.750	99.750	99.750
MLP	RD-4	98.069	98.052	98.051	98.052
MLP	RD-5	97.829	97.808	97.805	97.808
MLP	RD-6	97.750	97.725	97.722	97.725
MLP	RD-7	97.404	97.345	97.338	97.345
MLP	RD-8	96.210	96.116	96.104	96.116
MLP	RD-9	95.403	95.267	95.248	95.267
RBM+	RD-1	94.523	94.433	94.408	94.433
RBM+	RD-2	90.656	90.560	90.531	90.560
RBM+	RD-3	86.251	86.025	85.891	86.025
RBM+	RD-4	80.798	80.232	79.864	80.232
RBM+	RD-5	74.713	74.105	73.445	74.105
RBM+	RD-6	67.774	66.393	64.346	66.393
RBM+	RD-7	64.511	62.931	59.222	62.931
RBM+	RD-8	58.409	61.098	55.588	61.098
RBM+	RD-9	51.916	57.952	49.480	57.952
SGD	RD-1	99.985	99.985	99.985	99.985
SGD	RD-2	99.985	99.985	99.985	99.985
SGD	RD-3	99.959	99.959	99.959	99.959
SGD	RD-4	99.897	99.889	99.889	99.889
SGD	RD-5	99.897	99.889	99.889	99.889
SGD	RD-6	99.878	99.875	99.875	99.875
SGD	RD-7	99.752	99.745	99.745	99.745
SGD	RD-8	99.707	99.702	99.702	99.702
SGD	RD-9	98.792	98.622	98.550	98.622
RAN	RD-1	89.244	80.837	83.459	80.837
RAN	RD-2	90.406	80.746	83.720	80.746
RAN	RD-3	90.199	81.660	84.394	81.660
RAN	RD-4	86.849	79.479	81.508	79.479
RAN	RD-5	89.375	80.688	83.681	80.688
RAN	RD-6	89.606	79.265	82.888	79.265
RAN	RD-7	90.198	78.376	82.509	78.376
RAN	RD-8	90.377	78.653	82.676	78.653
RAN	RD-9	88.462	72.665	77.795	72.665

Algo [Algorithm]; RD [Research Design]; MLP [Multi Layer Perceptron]
LR [Logistic Regression]; SGD [Stochastic Gradient Descent];
RBM+ [Restricted Boltzmann Machine pipelined with Logistic Regression]
RAN [Regulated Activation Network]; K-NN [K Nearest Neighbor]

Appendix E

Data Manipulation Via Multiple Dimension Transformations

This appendix describes two important application of RANs modeling: first, dimension reduction and expansion of data; second, visualization of manipulated data. Convex abstract concept modeling is used in the demonstration as described in Section 3.3 of Chapter 3 and K-means clustering algorithm is used as a concept identifier. In this experiment yamanishi (Gutteridge et al., 2008) data is used. The RANs model was initialized with a the desired depth of *6* to grow up five layers above the input layer. The K value of K-means algorithm is set to *2, 30, 2, 40, 2* and *8* in order to create layers *1, 2, 3, 4,* and *5* with *2, 30, 2, 40, 2* and *8* nodes in each layer respectively. The input Layer-0 has 326 nodes equal to the number of attributes of the input data, and the data was pre-processed according to the RANs requirements (see Section 3.1 of Chapter 3). The model is generated by applying the CHC algorithm (see Algorithm 2 in Section 3.4.1 of Chapter 3). Figure E.1 shows the generated model where Layer-0 is the input layer, dimension reduction takes place at Layer-1, Layer-3 and Layer-5, whereas the dimension expansion happens in Layer-2, Layer-4 and Layer-6.

Having obtained the model, the data is as again propagated from input Layer-0 to Layer-5. At layers *1, 3,* and *5* the transformed 2-dimensional data is visualized and shown in Figure E.1. The effect of alternating the dimension expansion & reduction it very interesting. The graph to visualize the first dimension reduction at Layer-1 depicts that the data consists of two categories. After first dimension expansion at Layer-2, the second dimension reduction is obtained at Layer-3. The graph at Layer-3 depicts two classes where one class is convex and the other is transformed in a non-convex cluster. The last dimension reduction at Layer-5 we can observe a drastic transformation in the data which looks like a curve depicting a single non-convex cluster.

Figure E.2 shows an analysis of the graph generated from the transformed 2-D data obtained at Layer-5. The graph displays six regions showing the representation of Nodes 1 and 2 of Layer-5 in Figure E.1. In region **A** the activation of the representative Node is between 0.9 to

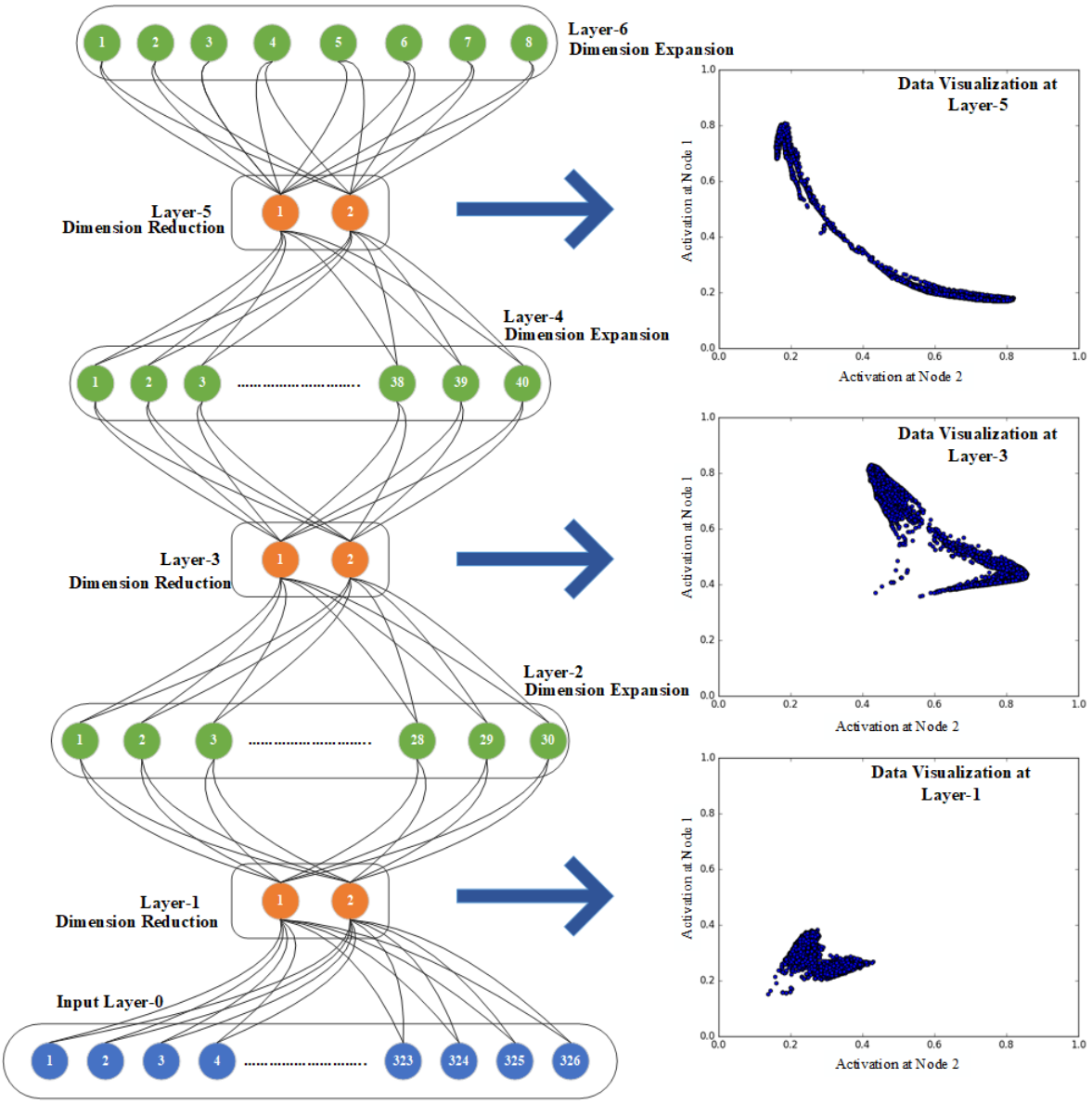


FIGURE E.1: Visualization of 2-D data transformed through RANs Model at three different levels.

1.0 therefore it strongly represents the Node. Similarly, the regions **B, C, D,** & **E** represented by the activation in the range 0.50 to 0.90, and not necessarily the best representation of the Node. The region **F** cover the area where the Node is getting activation less than 0.50. An activation value 0.50 can also be stated in terms of Degree of Confidence (see Section 3.3.2 of Chapter 3), i.e., with a confidence of 50% the Node represents the data. Alternately it can be stated the Node does not represent the data with a confidence of 50%, indicating a state of confusion.

By viewing the plotted data points in Figure E.1 for Layer-5 we can say that the data points neither strongly nor reasonably represented the two nodes of Layer-5. Therefore, there must exist more than two classes in the data. The eight red ovals in Figure E.2 shows how we can

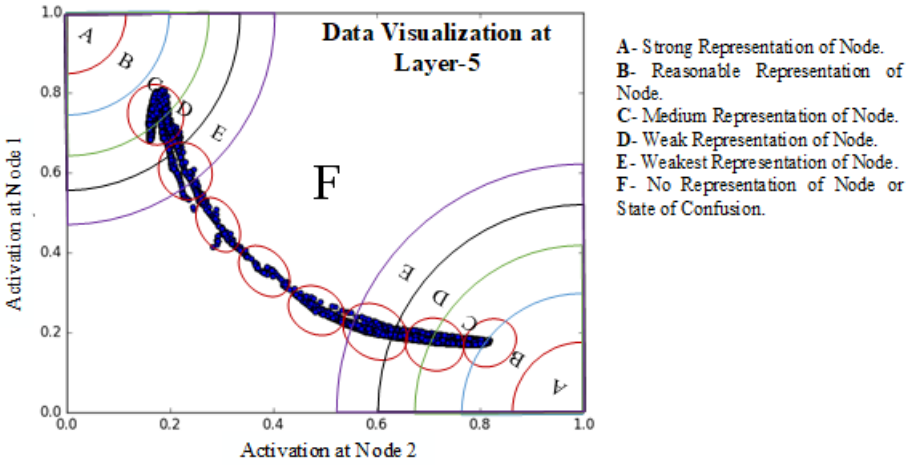


FIGURE E.2: Analysis of the 2-D data graph obtained at Layer-5.

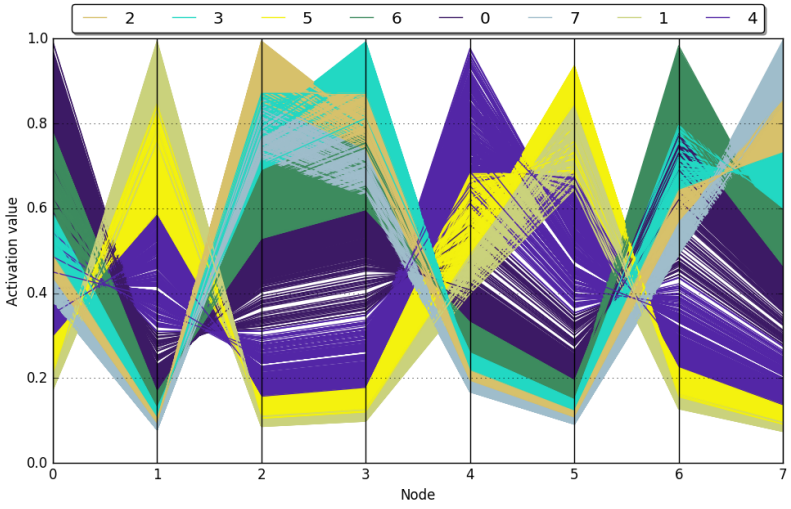


FIGURE E.3: Activation observed at eight Nodes of Layer-6.

infer the number of classes present in the data. Eight categories are identified in this example because with eight Nodes in Layer-6 we observe strong, reasonable or medium activation at each node. Figure E.3 shows the parallel coordinate plot of activation observed at all eight Nodes of Layer-5 of Figure E.1. If we want to have only strong representation by each node the number of Nodes at Layer-6 will increase. The choice of the range of activation values for Node representation varies with the data understanding and choice of the data analyst.

Index

.

Bibliography

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for Boltzmann Machines. *Cognitive science*, 9(1):147–169, 1985.

Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

Shunichi Amari. A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, (3):299–307, 1967.

Robert A Amsler. A taxonomy for English nouns and verbs. In *Proceedings of the 19th annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1981.

John R Anderson. *The Architecture of Cognition*. Harvard University Press, Cambridge, MA, USA, 1983a. ISBN 0674044258.

John R Anderson. ACT: A simple theory of complex cognition. *American Psychologist*, 51(4): 355, 1996.

John R Anderson and Scott Douglass. Tower of Hanoi: Evidence for the cost of goal retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 27(6):1331, 2001.

John R Anderson and Christian Lebiere. The newell test for a theory of cognition. *Behavioral and brain Sciences*, 26(5):587–601, 2003.

John R Anderson and Michael Matessa. A production system theory of serial memory. *Psychological Review*, 104(4):728, 1997.

John R Anderson, Michael Matessa, and Christian Lebiere. ACT-R: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction*, 12(4):439–462, 1997.

John R Anderson, Dan Bothell, Christian Lebiere, and Michael Matessa. An integrated theory of list memory. *Journal of Memory and Language*, 38(4):341–380, 1998.

John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036, 2004.

J.R. Anderson. A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, 22(3):261–295, 1983b.

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition using Smartphones. In *Empirical Methods in Natural Language Processing*, 2013.

James W Antony, Catarina S Ferreira, Kenneth A Norman, and Maria Wimber. Retrieval as a fast route to memory consolidation. *Trends in Cognitive Sciences*, 21(8):573–576, 2017.

Ngombo Armando, Duarte Raposo, Marcelo Fernandes, André Rodrigues, Jorge Sá Silva, and Fernando Boavida. WSNs in FIWARE–Towards the Development of People-Centric Applications. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 445–456. Springer, 2017.

G P Babu and M N Murty. A near-optimal initial seed value selection in k- means algorithm using a genetic algorithm. *Pattern Recognition*, 14(10):763–169, 1993.

Alan Baddeley. Working memory. *Science*, 255(5044):556–559, 1992.

Shumeet Baluja, Dean Pomerleau, and Todd Jochem. Towards automated artificial evolution for computer-generated images. *Connection Science*, 6(2-3):325–354, 1994.

Tudor Barbu. Unsupervised SIFT-based Face Recognition Using an Automatic Hierarchical Agglomerative Clustering Solution. *Procedia Computer Science*, 22:385 – 394, 2013.

P Bardley and Fayyad U. Refining initial points for k-means clustering. *In Proceedings of the 15th ICML*, pages 91–99, 1998.

Lawrence W Barsalou and Katja Wiemer-Hastings. Situating abstract concepts. *Grounding cognition: The role of perception and action in memory, language, and thought*, pages 129–163, 2005.

Lawrence W Barsalou, Ava Santos, W Kyle Simmons, and Christine D Wilson. *Language and simulation in conceptual processing*, pages 245–284. Oxford University Press, 2008.

Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.

William Bechtel, George Graham, and David A Balota. *A companion to cognitive science*. Wiley-Blackwell, 1998.

Suzanna Becker and Jean Lim. A computational model of prefrontal control in free recall: strategic memory use in the California Verbal Learning Task. *Journal of Cognitive Neuroscience*, 15(6):821–832, 2003.

David Beckett, Tim Berners-Lee, Eric Prud'hommeaux, and Gavin Carothers. Terse RDF Triple Language W3C Recommendation, 2014. URL `https://www.w3.org/TR/turtle/`. Accessed online: 20^{th} July, 2015.

R Bellman, R Kalaba, and L A Zadeh. Abstraction and pattern classification. *Mathematics Analysis and Application*, 2:581–586, 1966.

Y Bengio, A Courville, and P Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Yoshua Bengio and etal. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

Kristin P Bennett and Olvi L Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization methods and software*, 1(1):23–34, 1992.

Douglas Bermingham, Robert D Hill, Dan Woltz, and Michael K Gardner. Cognitive strategy use and measured numeric ability in immediate-and long-term recall of everyday numeric information. *PloS one*, 8(3):e57999, 2013.

Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284 (5):34–43, 2001.

Michael R Berthold. Towards bisociative knowledge discovery. In *Bisociative Knowledge Discovery*, pages 1–10. Springer, 2012.

James C Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.

James C Bezdek, L O Hall, and L P Clarke. Review of mr image segmentation techniques using pattern recognition. *Medical physics*, 20(4):1033–1048, 1992.

Greg Bickerman, Sam Bosley, Peter Swire, and Robert M Keller. Learning to Create Jazz Melodies Using Deep Belief Nets. In *Proceedings of the First International Conference on Computational Creativity*, pages 228–237, 2010.

Chris Biemann. Ontology learning from text: A survey of methods. In *LDV forum*, volume 20, pages 75–93, 2005.

David Biggs and Andrew Nuttall. Neural Memory Networks. Technical report, Stanford University, 2015.

Jeffrey R Binder. In defense of abstract conceptual representations. *Psychonomic Bulletin & Review*, 23(4):1096–1108, 2016.

Jeffrey R Binder, Chris F Westbury, Kristen A McKiernan, Edward T Possing, and David A Medler. Distinct brain systems for processing concrete and abstract concepts. *Journal of Cognitive Neuroscience*, 17(6):905–917, 2005.

Margaret A Boden. *The Creative Mind: Myths and Mechanisms*. George Weidenfeld and Nicolson Ltd, New York, NY, 1990.

Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings, 2014. arXiv preprint arXiv:1406.3676v3.

Ann AM Borghi and Ferdin Binkofski. *The WAT Proposal and the Role of Language*, pages 19–37. Springer, 2014.

Anna M Borghi, Ferdinand Binkofski, Cristiano Castelfranchi, Felice Cimatti, Claudia Scorolli, and Luca Tummolini. The challenge of abstract concepts. *Psychological Bulletin*, 143(3): 263–292, 2017.

Anna M Borghi, Laura Barca, Ferdinand Binkofski, and Luca Tummolini. Varieties of abstract concepts: development, use and representation in the brain. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 373(1752):20170121, 2018.

Jelmer P Borst and John R Anderson. Using the ACT-R Cognitive Architecture in combination with fMRI data. In *An introduction to model-based cognitive neuroscience*, pages 339–352. Springer, 2015.

Maroua Bouzid, Carlo Combi, Michael Fisher, and Gérard Ligozat. Guest editorial: temporal representation and reasoning. *Annals of Mathematics and Artificial Intelligence*, 46(3):231–234, 2006.

Gordon H Bower. *A brief history of memory research*, pages 3–32. Oxford University Press, 2000.

Oliver Bown and Sebastian Lexer. Continuous-time recurrent neural networks for generative and interactive musical performance. In *Workshops on Applications of Evolutionary Computation*, pages 652–663. Springer, 2006.

John D Bransford and Nancy S McCarrell. *A sketch of a cognitive approach to comprehension: Some thoughts about understanding what it means to comprehend*, pages 191–121. Cambrige University Press, 1974.

T S Braver, D M Barch, and J D Cohen. Cognition and control in schizophrenia: A computational model of dopamine and prefrontal function. *Biological Psychiatry*, 46(3):312–328, 1999.

Dan Brickley. OWL Web Ontology Language, 2004. URL `http://www.w3.org/TR/rdf-schema/`. Accessed online: 20^{th} July, 2015.

György Buzsáki. Neural syntax: cell assemblies, synapsembles, and readers. *Neuron*, 68(3): 362–385, 2010.

John T Cacioppo and Louise C Hawkley. Perceived social isolation and cognition. *Trends in Cognitive Sciences*, 13(10):447–454, 2009.

William H Calvin and Derek Bickerton. *Lingua ex machina: Reconciling Darwin and Chomsky with the human brain*. MIT press, 2001.

Nicoletta Calzolari, Laura Pecchia, and Antonio Zampolli. Working on the italian machine dictionary: a semantic approach. In *Proceedings of the 5th conference on Computational linguistics*, pages 49–52. Association for Computational Linguistics, 1973.

Nicholas L Cassimatis, J Gregory Trafton, Magdalena D Bugajska, and Alan C Schultz. Integrating cognition, perception and action through mental simulation in robots. *Robotics and Autonomous Systems*, 49(1-2):13–23, 2004.

Nicholas Louis Cassimatis. *Polyscheme: a cognitive architecture for intergrating multiple representation and inference schemes*. PhD thesis, Massachusetts Institute of Technology, 2001.

Daniel Castro, Paulo Félix, and Jesús Presedo. A method for context-based adaptive QRS clustering in real time. *Journal of Biomedical and Health Informatics*, 19(5):1660–1671, 2015.

Anna R Chambers and Simon Rumpel. A stable brain from unstable components: emerging concepts and implications for neural computation. *Neuroscience*, 357:172–184, 2017.

Samuel D Chapman, Christoph Adami, Claus O Wilke, and B K C Dukka. The evolution of logic circuits for the purpose of protein contact map prediction. *PeerJ*, 5:e3139, 2017.

Hao Chen and Burt M Sharp. Content-rich biological network constructed by mining PubMed abstracts. *BMC Bioinformatics*, 5(1):147, 2004.

T Fang Chiu, D J Chen, and Y Wang. A robust and scalable clustering algorithm for mixed type attributes in large database environments. *In Proceedings of the 7th ACM SIGKDD, San Francisco, CA*, pages 263–268, 2001.

W Chumsamrong, P Thitimajshima, and Y Rangsanseri. Synthetic aperture radar (sar) image segmentation using a new modified fuzzy c-means algorithm. In *Geoscience and Remote Sensing Symposium, 2000. Proceedings. IGARSS 2000. IEEE 2000 International*, volume 2, pages 624–626, 2000.

Dan CireşAn, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural networks*, 32:333–338, 2012.

Adam Coates and Andrew Y Ng. Learning feature representations with k-means. *Neural Networks: Tricks of the Trade. Springer*, 2012.

Allan M Collins and Elizabeth F Loftus. A spreading-activation theory of semantic processing. *Psychological Review*, 82(6):407, 1975.

A.M. Collins and M.R. Quillian. Retrieval time from semantic memory. *Jpurnal of Verbal Learning and Verbal Behavior*, 8(2):240–247, 1969.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167. ACM, 2008.

Ornella Cominetti, Anastasios Matzavinos, Sandhya Samarasinghe, Don Kulasiri, Sijia Liu, Philip Maini, and Radek Erban. DifFUZZY: a fuzzy clustering algorithm for complex datasets. *International Journal of Computational Intelligence in Bioinformatics and Systems Biology*, 1(4):402–417, 2010.

Albert T Corbett and Akshat Bhatnagar. Student modeling in the ACT programming tutor: Adjusting a procedural learning model with declarative knowledge. In *User modeling*, pages 243–254. Springer, 1997.

Mark Craven and Johan Kumlien. Constructing Biological Knowledge Bases by Extracting Information from Text Sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 77–86. AAAI Press, 1999.

F Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11:453–482, 1997.

Holk Cruse. *Neural networks as cybernetic systems*. Thieme Stuttgart, 1996.

S Dara and P Tumma. Feature extraction by using deep learning: A survey. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1795–1801. IEEE, 2018.

Celso M de Melo and Jonathan Gratch. Evolving expression of emotions through color in virtual humans using genetic algorithms. In *Proceedings of the Fourth International Conference on Computational Creativity*, pages 248–257. Citeseer, 2010.

Dictionary.com. Dictionay meaning of word "association", 2015a. URL `https://www.dictionary.com/browse/association`. Accessed online: 8^{th} June, 2015.

Dictionary.com. Dictionay meaning of word "concept", 2015b. URL `https://www.dictionary.com/browse/concept?s=t`. Accessed online: 8^{th} June, 2015.

Dictionary.com. Dictionay meaning of word "ontology", 2015c. URL `https://www.dictionary.com/browse/concept?s=t`. Accessed online: 12^{th} June, 2015.

Guy Dove. Beyond perceptual symbols: A call for representational pluralism. *Cognition*, 110 (3):412–431, 2009.

Guy Dove. Thinking in words: language as an embodied medium of thought. *Topics in cognitive science*, 6(3):371–389, 2014.

Werner Dubitzky, Tobias Kötter, Oliver Schmidt, and Michael R Berthold. Towards creative information exploration based on Koestler's concept of bisociation. In *Bisociative Knowledge Discovery*, pages 11–32. Springer, 2012.

Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybneticics*, 3(3):32–57, 1973.

Douglas Eck and Jürgen Schmidhuber. Learning the long-term structure of the blues. In *International Conference on Artificial Neural Networks*, pages 284–289. Springer, 2002.

Jeffrey A Edlund, Nicolas Chaumont, Arend Hintze, Christof Koch, Giulio Tononi, and Christoph Adami. Integrated information increases with fitness in the evolution of animats. *PLoS computational biology*, 7(10):1–13, 2011.

David Eigen, Jason Rolfe, Rob Fergus, and Yann LeCun. Understanding Deep Architectures using a Recursive Convolutional Network. In *International Conference on Learning Representations*. CBLS, April 2014.

M Elter, R Schulz-Wendtland, and T Wittenberg. The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process. *Medical Physics*, 34(11):4164–4172, 2007.

I W Evett and E J Spiehler. Rule induction in forensic science. Technical report, Central Research Establishment, Home Office Forensic Science Service, 1987.

Scott E Fahlman and Christian Lebiere. The cascade-correlation learning architecture. In *Advances in neural information processing systems*, pages 524–532, 1990.

B Fasel. An introduction to bio-inspired artificial neural network architectures. *Acta neurologica belgica*, 103(1):6–12, 2003.

Paulo Fazendeiro and Jose Valente de Oliveira. Observer-biased fuzzy clustering. *Transactions on Fuzzy Systems*, 23(1):85–97, 2015.

Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7(2):179–188, 1936.

E Forgy. Cluster analysis of multivariate data: Efficiency versus interpretability of classification. *Biometrics*, 1965.

Michele Forina, Riccardo Leardi, C Armanino, Silvia Lanteri, P Conti, and P Princi. PARVUS: An extendable package of programs for data exploration, classification and correlation. *Journal of Chemometrics*, 4(2):191–193, 1988.

Stan Franklin and Lee McCauley. Interacting with IDA. In *Agent autonomy*, pages 159–183. Springer, 2003.

Stan Franklin and FG Patterson Jr. The LIDA architecture: Adding new modes of learning to an intelligent, autonomous, software agent. In *Procedings of Integrated Design and Process Technologe*, pages 764–1004. Society of Design Process Science, 2006.

Stan Franklin, Arpad Kelemen, and Lee McCauley. IDA: A cognitive agent architecture. In *International Conference on Systems, Man, and Cybernetics*, volume 3, pages 2646–2651. IEEE, 1998.

Stan Franklin, Uma Ramamurthy, Sidney K D'Mello, Lee McCauley, Aregahegn Negatu, Rodrigo L Silva, and Vivek Datla. Lida: A computational model of global workspace theory and developmental learning. In *Fall Symposium: AI and Consciousness*, pages 61–66. AAAI, 2007.

Stan Franklin, Tamas Madl, Sidney D'Mello, and Javier Snaider. LIDA: A systems-level architecture for cognition, emotion, and learning. *Transactions on Autonomous Mental Development*, 6(1):19–41, 2014.

David A Freedman. *Statistical models: theory and practice*. Cambridge University Press, 2009.

Brendan J Frey and Delbert Dueck. Clustering by Passing Messages Between Data Points. *Science*, 315(5814):972–976, 2007.

Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening ontologies with DOLCE. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 166–181. Springer, 2002.

Christophe Garcia and Manolis Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 26(11):1408–1423, 2004.

Peter Gärdenfors. *Conceptual spaces: The geometry of thought*. MIT press, 2004.

Peter Gärdenfors. Conceptual spaces as a framework for knowledge representation. *Mind and Matter*, 2(2):9–27, 2004.

Bob Garrett. *Study Guide to Accompany Bob Garrett's Brain & Behavior: An Introduction to Biological Psychology*. Sage Publications, 2014.

Matt Gedigian, John Bryant, Srini Narayanan, and Branimir Ciric. Catching Metaphors. In *Proceedings of the Third Workshop on Scalable Natural Language Understanding*, pages 41–48. Association for Computational Linguistics, 2006.

Dileep George. *How the brain might work: A hierarchical and temporal model for learning and recognition*. PhD thesis, Stanford University, 2008.

Raymond W Gibbs Jr. Why many concepts are metaphorical. *Cognition*, 61(3):309–319, 1996.

Gary Gillund and Richard M Shiffrin. A retrieval model for both recognition and recall. *Psychological Review*, 91(1):1, 1984.

Roxana Girju, Adriana Badulescu, and Dan Moldovan. Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135, 2006.

Ben Goertzel, Cassio Pennachin, Nil Geissweiller, Moshe Looks, Andre Senna, Welter Silva, Ari Heljakka, and Carlos Lopes. An Integrative Methodology for Teaching Embodied Non-Linguistic Agents, Applied to Virtual Animals in Second Life. In *Proceedings of the 2008 Conference on Artificial General Intelligence 2008: Proceedings of the First AGI Conference*, pages 161–175. IOS Press, 2008. ISBN 978-1-58603-833-5.

Robert L Goldstone. The role of similarity in categorization: Providing a groundwork. *Cognition*, 52(2):125–157, 1994.

Robert L Goldstone, Alan Kerstan, and Paulo F Carvalho. *Concepts and Categorization*, pages 607–630. John Wiley & Sons. Inc., 2013.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Stephen Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cognitive science*, 11(1):23–63, 1987.

Frédéric Gruau. Automatic Definition of Modular Neural Networks. *Adaptive Behavior*, 3(2): 151–183, 1994a.

Frédéric Gruau. Automatic definition of modular neural networks. *Adaptive behavior*, 3(2): 151–183, 1994b.

Frédéric Gruau, Darrell Whitley, and Larry Pyeatt. A Comparison Between Cellular Encoding and Direct Encoding for Genetic Neural Networks. In *Proceedings of the 1st Annual Conference on Genetic Programming*, pages 81–89. MIT Press, 1996.

Tom Gruber. Ontology. Entry in the Encyclopedia of Database Systems, Ling Liu and M. Tamer Özsu, 2008.

Saikat Guha, Rajeev Rastogi, and Kyuseok Shim. ROCK: A robust clustering algorithm for categorical attributes. In *Proceedings of 15th International Conference on Data Engineering*, pages 512–521. IEEE, 1999.

Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: an efficient clustering algorithm for large databases. In *ACM SIGMOD Record*, volume 27, pages 73–84, 1998.

Kevin Gurney. *An introduction to neural networks*. CRC press, 2014.

Claudio Gutierrez, Carlos A Hurtado, Alberto O Mendelzon, and Jorge Pérez. Foundations of semantic web databases. *Journal of Computer and System Sciences*, 77(3):520–541, 2011.

Alex Gutteridge, Michihiro Araki, Minoru Kanehisa, Wataru Honda, and Yoshihiro Yamanishi. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24(13):i232–i240, 2008.

Mark Andrew Hall. *Correlation-based feature selection for machine learning*. PhD dissertation, University of Waikato Hamilton, 1999.

John A Hartigan and Manchek A Wong. Algorithm AS 136: A K-means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 415. Association for Computational Linguistics, 2004.

Justin C Hayes and David JM Kraemer. Grounded understanding of abstract concepts: The case of STEM learning. *Cognitive Research: Principles and Implications*, 2(1):7, 2017.

Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall, 1994.

Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545. Association for Computational Linguistics, 1992.

Donald O Hebb. *The organization of Bhavior. A Neuropsychological Theory*. John Wiley, 1949.

John Hebeler, Matthew Fisher, Ryan Blace, and Andrew Perez-Lopez. *Semantic web programming*. John Wiley & Sons, 2011.

Sebastien Hélie and Ron Sun. Incubation, insight, and creative problem solving: a unified theory and a connectionist model. *Psychological Review*, 117(3):994–1024, 2010.

Sébastien Hélie and Ron Sun. An integrative account of memory and reasoning phenomena. *New Ideas in Psychology*, 35:36–52, 2014.

Sébastien Hélie, Ron Sun, and Liling Xiong. Mixed effects of distractor tasks on incubation. In *Proceedings of the 30th Annual Meeting of the Cognitive Science Society*, pages 1251–1256. Austin: Cognitive Science Society, 2008.

Jim Hendler and Frank van Harmelen. The semantic web: webizing knowledge representation. *Foundations of Artificial Intelligence*, 3:821–839, 2008.

HierarchicalClusteringWiki. Hierarchical clustering wiki. `http://en.wikipedia.org/wiki/Hierarchical_clustering`, 2015. [Online; accessed June].

Clara Higuera, Katheleen J Gardiner, and Krzysztof J Cios. Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PLOS ONE*, 10(6): e0129126, 2015.

Felix Hill and Anna Korhonen. Learning Abstract Concept Embeddings from Multi-Modal Data: Since You Probably Can't See What I Mean. In *Empirical Methods in Natural Language Processing*, pages 255–265. Association for Computational Linguistics, 2014.

G E Hinton. Boltzmann machine. *Scholarpedia*, 2(5):1668, 2007. doi: 10.4249/scholarpedia.1668. revision #91076.

G E Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009. doi: 10.4249/scholarpedia.5947. revision #91189.

G E Hinton, J L McClelland, and D E Rumelhart. Distributed representations. In David E. Rumelhart, James L. McClelland, and CORPORATE PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, pages 77–109. MIT Press, Cambridge, MA, USA, 1986. ISBN 0-262-68053-X. URL `http://dl.acm.org/citation.cfm?id=104279.104287`.

Geoffrey Hinton. *A practical guide to training restricted Boltzmann machines*, pages 599–619. Springer Berlin Heidelberg, 2012.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Geoffrey E Hinton, Terrence J Sejnowski, and David H Ackley. *Boltzmann Machines: Constraint satisfaction networks that learn.* Carnegie-Mellon University, Department of Computer Science Pittsburgh, PA, 1984.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, 2006.

Arend Hintze, Jeffrey A Edlund, Randal S Olson, David B Knoester, Jory Schossau, Larissa Albantakis, Ali Tehrani-Saleh, Peter Kvam, Leigh Sheneman, Heather Goldsby, et al. Markov brains: A technical introduction, 2017. arXiv preprint arXiv:1709.05601.

HMMWiki. Hidden morkov models wiki. `http://en.wikipedia.org/wiki/Hidden_Markov_model`, 2015. [Online; accessed Jan].

Thomas Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57. ACM, 1999.

Douglas R Hofstadter. The Copycat Project: An Experiment in Nondeterminism and Creative Analogies. Technical report, MIT Cambridge AI Labs, 1984.

Anthony J G D Holtmaat, Joshua T Trachtenberg, Linda Wilbrecht, Gordon M Shepherd, Xiaoqun Zhang, Graham W Knott, and Karel Svoboda. Transient and persistent dendritic spines in the neocortex in vivo. *Neuron*, 45(2):279–291, 2005.

Ping Hou, Marc Denecker, et al. Fo (fd): Extending classical logic with rule-based fixpoint definitions. *Theory and Practice of Logic Programming*, 10(4-6):581–596, 2010.

Marc W Howard and Michael J Kahana. A distributed representation of temporal context. *Journal of Mathematical Psychology*, 46(3):269–299, 2002.

Fu Jie Huang and Yann LeCun. THE NORB DATASET, 2004. URL `https://cs.nyu.edu/~ylclab/data/norb-v1.0/`. Courant Institute, New York University.

Z Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3):283–304, 1998a.

Zhexue Huang. A fast clustering algorithm to cluster very large categorical data sets in data mining. In *DMKD*, page 0. Citeseer, 1997.

Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998b.

J C Hulbert and K A Norman. Neural differentiation tracks improved recall of competing memories following interleaved study and retrieval practice. *Cerebral Cortex*, 25(10):3994–4008, 2014.

Michael Iannacone, Shawn Bohn, Grant Nakamura, John Gerth, Kelly Huffer, Robert Bridges, Erik Ferragut, and John Goodall. Developing an ontology for cyber security knowledge graphs. In *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, page 12. ACM, 2015.

Elias Iosif. *Network-based distributional semantic models*. PhD thesis, Technical University of Crete, 2013a.

Elias Iosif. *Network-based distributional semantic models*. PhD thesis, Technical University of Crete, Chania, Greece, 2013b.

Elias Iosif, Alexandros Potamianos, Maria Giannoudaki, and KA Zervanou. Semantic similarity computation for abstract and concrete nouns using network-based distributional semantic models. In *Proceedings of the 10th International Conference on Computational Semantics*, pages 328–334. Potsdam, Germany:[sn], 2013.

Larry L Jacoby. Perceptual enhancement: Persistent effects of an experience. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 9(1):21, 1983.

Anil K Jain and Richard C Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988. ISBN 0-13-022278-X.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation, 2014. arXiv preprint arXiv:1412.2007v2.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1): 221–231, 2013.

Bradley Joseph, Loucks Jeff, Macaulay James, and Noronha Andy. White paper on Internet of Everything (IoE) Value Index How Much Value Are Private-Sector Firms Capturing from IoE in 2013? http://internetofeverything.cisco.com/sites/default/files/docs/en/ioe-value-index_Whitepaper.pdf, 2017. [Online; accessed 6-Dec-2017].

Matjaž Juršič, Borut Sluban, Bojan Cestnik, Miha Grčar, and Nada Lavrač. Bridging concept identification for constructing information networks from text documents. In *Bisociative Knowledge Discovery*, pages 66–90. Springer, 2012.

Matjaž Juršič, Borut Sluban, Bojan Cestnik, Miha Grčar, and Nada Lavrač. Bisociative knowledge discovery. chapter Bridging Concept Identification for Constructing Information Networks from Text Documents, pages 66–90. Springer-Verlag, 2012.

Yoshitaka Kameya and Taisuke Sato. Computation of probabilistic relationship between concepts and their attributes using a statistical analysis of Japanese corpora. In *Proceedings of Symposium on Large-scale Knowledge Resources*, pages 65–68. Citeseer, 2005.

Kyoko Kanzaki, Eiko Yamamoto, Hitoshi Isahara, and Qing Ma. Construction of an Objective Hierarchy of Abstract Concepts via Directional Similarity. In *Proceedings of the 20th International Conference on Computational Linguistics*, page 1147. Association for Computational Linguistics, 2004.

David Kappel, Stefan Habenschuss, Robert Legenstein, and Wolfgang Maass. Network plasticity as bayesian inference. *PLoS computational biology*, 11(11):e1004485, 2015.

Shitij Kapur, Fergus I M Craik, Corey Jones, Gregory M Brown, Sylvain Houle, and Endel Tulving. Functional role of the prefrontal cortex in retrieval of memories: a PET study. *Neuroreport*, 6(14):1880–1884, 1995.

G Karypis, EH Han, and V Kumar. Multilevel refinement for hierarchical clustering. technical report tr-99-020. *Department of Computer Science, University of Minnesota, Minneapolis*, 1999a.

George Karypis, Eui-Hong (Sam) Han, and Vipin Kumar. Chameleon: Hierarchical Clustering Using Dynamic Modeling. *Computer*, 32(8), August 1999b.

Samuel Kaski. Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *The International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence*, volume 1, pages 413–418. IEEE, 1998.

L Kaufman and P Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, New York, NY, 1990.

Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann LeCun. Learning Convolutional Feature Hierachies for Visual Recognition. In *Advances in Neural Information Processing Systems*, volume 23, 2010.

Markus Kiefer and Friedemann Pulvermüller. Conceptual representations in mind and brain: theoretical developments, current evidence and future directions. *Cortex*, 48(7):805–825, 2012.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. arXiv preprint arXiv:1312.6114v10.

Walter Kintsch. Metaphor comprehension: A computational theory. *Psychonomic bulletin & review*, 7(2):257–266, 2000.

KmeanClusterImage. Kmean clustering mimul's. `http://www.mimul.com/pebble/default/2012/04/08/1333871113118.html`, 2015. [Online; accessed July].

Arthur Koestler. *The Act of Creation Macmillan*. Oxford, England:Macmillan, 1964.

Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.

Boicho N Kokinov. The DUAL Cognitive Architecture: A Hybrid Multi-Agent Approach. In *11^{th} European Conference on Artificial Intelligence*, pages 203–207. John Wiley & Sons, 1994.

Daphne Koller, Nir Friedman, and Francis Bach. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Iuliia Kotseruba and John K Tsotsos. 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, pages 1–78, 2018.

Stavroula-Thaleia Kousta, Gabriella Vigliocco, David P Vinson, Mark Andrews, and Elena Del Campo. The representation of abstract words: Why emotion matters. *Journal of Experimental Psychology: General*, 140(1):14, 2011.

Martin Krallinger, Carlos Rodriguez-Penagos, Ashish Tendulkar, and Alfonso Valencia. PLAN2L: a web tool for integrated text mining and literature-based bioentity relation extraction. *Nucleic Acids Research*, 37(suppl_2):W160–W165, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, 60(6):84–90, 2017.

Peter Kvam, Joseph Cesario, Jory Schossau, Heather Eisthen, and Arend Hintze. Computational evolution of decision-making strategies, 2015. arXiv preprint arXiv:1509.05646.

Peter D Kvam and Arend Hintze. Rewards, risks, and reaching the right strategy: Evolutionary paths from heuristics to optimal decisions. *Evolutionary Behavioral Sciences*, 12(3):177, 2018.

S Kyaga, M Landén, M Boman, C M Hultman, N Långström, and P Lichtenstein. Mental illness, suicide and creativity: 40-Year prospective total population study. *Journal of Psychiatric Research*, 47(1):83–90, 2013.

John E Laird. *The Soar cognitive architecture*. MIT press, 2012.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284, 1998.

Ora Lassila and Ralph R Swick. Resource Description Framework (RDF) Model and Syntax Specification. Recommendation 22 february 1999, W3C, Cambridge, MA, 1999. URL `http://www.w3.org/TR/REC-rdf-syntax/`.

John Launchbury. A darpa Perspective on Artificial Intelligence, 2017. URL `https://www.youtube.com/watch?v=-O01G3tSYpU`. YouTube video. Accessed online: March, 2017.

Quoc V Le. Building high-level features using large scale unsupervised learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8595–8598. IEEE, 2013.

Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*, pages 396–404, 1990.

Yann LeCun, Corinna Cortes, and Christopher J C Burges. The MNIST DATABASE of handwritten digits, 1999. URL `http://yann.lecun.com/exdb/mnist/`. Accessed: 2015-07-10.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

Douglas B Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

Alessandro Lenci, Gianluca E Lebani, and Lucia C Passaro. The Emotions of Abstract Words: A Distributional Semantic Analysis. *Topics in Cognitive Science*, 10(3):550–572, 2018.

Michael KK Leung, Hui Yuan Xiong, Leo J Lee, and Brendan J Frey. Deep learning of the tissue-regulated splicing code. *Bioinformatics*, 30(12):i121–i129, 2014.

Richard L Lewis and Shravan Vasishth. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive science*, 29(3):375–419, 2005.

M. Lichman. UCI Machine Learning Repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Dekang Lin and Patrick Pantel. DIRT @SBT@Discovery of Inference Rules from Text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 323–328. ACM, 2001.

Dekang Lin and Xiaoyun Wu. Phrase clustering for discriminative learning. *Annual Meeting of the ACL and IJCNLP*, pages 1030–1038, 2009.

Cheng-Yuan Liou, Jau-Chi Huang, and Wen-Chie Yang. Modeling word perception using the Elman network. *Neurocomputing*, 71(16-18):3150–3157, 2008.

Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. Autoencoder for words. *Neurocomputing*, 139:84–96, 2014.

Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004.

Mengchen Liu, Jiaxin Shi, Zhen Li, Chongxuan Li, Jun Zhu, and Shixia Liu. Towards better analysis of deep convolutional neural networks. *IEEE transactions on visualization and computer graphics*, 23(1):91–100, 2017.

David Livingstone. *Unsupervised Learning*, pages 119–144. John Wiley & Sons, Ltd, 2009.

Stuart P Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

Marsha C Lovett, Larry Z Daily, and Lynne M Reder. A source activation theory of working memory: Cross-task prediction of performance in ACT-R. *Cognitive Systems Research*, 1(2): 99–118, 2000.

Conor Ó Luanaigh and Brian A Lawlor. Loneliness and the health of older people. *International Journal of Geriatric Psychiatry: A journal of the psychiatry of late life and allied sciences*, 23(12):1213–1221, 2008.

H P Luhn. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

Junshui Ma, Robert P Sheridan, Andy Liaw, George E Dahl, and Vladimir Svetnik. Deep neural nets as a method for quantitative structure–activity relationships. *Journal of chemical information and modeling*, 55(2):263–274, 2015.

Penousal Machado, Juan Romero, and Bill Manaris. Experiments in computational aesthetics. In *The art of artificial evolution*, pages 381–415. Springer, 2008.

Tamas Madl and Stan Franklin. Constrained incrementalist moral decision making for a biologically inspired cognitive architecture. In *A Construction Manual for Robots' Ethical Systems*, pages 137–153. Springer, 2015.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196. IEEE, 2015.

Julian N Marewski and Katja Mehlhorn. Using the ACT-R architecture to specify 39 quantitative process models of decision making. *Judgment and Decision Making*, 6(6):439–519, 2011.

David Marr and Herbert Keith Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 200(1140):269–294, 1978.

James B. Marshall. A self-watching model of analogy-making and perception. *Journal of Experimental & Theoretical Artificial Intelligence*, 18(3):267–307, 2006.

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

Deborah L McGuinness and Frank van Harmelen. RDF vocabulary description language 1.0: RDF schema, 2004. URL `https://www.w3.org/TR/owl-features/`. Accessed online: 20^{th} July, 2015.

T.P. McNamara and J. Altarriba. Depth of spreading activation revisited: Semantic mediated priming occurs in lexical decisions. *Journal of Memory and Language*, 27(5):545–559, 1988.

Sarnoff Mednick. The associative basis of the creative process. *Psychological Review*, 69(3):220, 1962.

Marco Mernberger, Gerhard Klebe, and Eyke Hüllermeier. SEGA: Semiglobal Graph Alignment for Structure-Based Protein Comparison. *IEEE/ACM Transaction on Computational Biology Bioinformatics*, 8(5):1330–1343, 2011.

Carolyn B Mervis and Eleanor Rosch. Categorization of natural objects. *Annual Review of Psychology*, 32(1):89–115, 1981.

Dragana Miljkovic, Tjaša Stare, Igor Mozetič, Vid Podpečan, Marko Petek, Kamil Witek, Marina Dermastia, Nada Lavrač, and Kristina Gruden. Signalling network construction for modelling plant defence response. *PLoS One*, 7(12):e51822, 2012.

George Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

Takashi Mitsuishi, Wasaki Katsumi, and Shidama Yasunari. The concept of fuzzy set and membership function and basic properties of fuzzy set operation. *Formalized Mathematics*, 2:351–356, 2001.

Kristine Monteith, Tony R Martinez, and Dan Ventura. Automatic Generation of Music for Inducing Emotive Response. In *Proceedings of the First International Conference on Computational Creativity*, pages 140–149, 2010.

F Murtagh. A review of fast techniques for nearest neighbour searching. In *Compstat 1984*, pages 143–147. Springer, 1984.

Allen Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA, USA, 1990. ISBN 0-674-92099-6.

Hai Thanh Nguyen, Katrin Franke, and Slobodan Petrovic. Towards a generic feature-selection measure for intrusion detection. In *20th International Conference on Pattern Recognition*, pages 1529–1532. IEEE, 2010.

Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67. IJCAI, 1999.

S Nolfi, O Miglino, and D Parisi. Phenotypic plasticity in evolving neural networks. In *Proceedings of: From Perception to Action*, pages 146–157. IEEE, 1994.

David Norton, Derrall Heath, and Dan Ventura. Establishing Appreciation in a Creative System. In *Proceedings of the First International Conference on Computational Creativity*, pages 26–35, 2010.

Steven J Nowlan and John C Platt. A Convolutional Neural Network Hand Tracker. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, pages 901–903. MIT Press, 1994.

Enkhbold Nyamsuren and Niels A Taatgen. Human reasoning module. *Biologically Inspired Cognitive Architectures*, 8:1–18, 2014.

Gustavo Olague, Eddie Clemente, León Dozal, and Daniel E. Hernández. Evolving an Artificial Visual Cortex for Object Recognition with Brain Programming. In Oliver Schuetze, Carlos A Coello Coello, Alexandru-Adrian Tantar, Emilia Tantar, Pascal Bouvry, Pierre Del Moral, and Pierrick Legrand, editors, *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III*, pages 97–119. Springer International Publishing, 2014.

Clark F Olson. Parallel algorithms for hierarchical clustering. *Parallel computing*, 21(8):1313–1325, 1995.

Randal S Olson, Arend Hintze, Fred C Dyer, David B Knoester, and Christoph Adami. Predator confusion is sufficient to evolve swarming behaviour. *Journal of The Royal Society Interface*, 10(85):20130305, 2013.

Toshihide Ono, Haretsugu Hishigaki, Akira Tanigami, and Toshihisa Takagi. Automated extraction of information on protein–protein interactions from the biological literature. *Bioinformatics*, 17(2):155–161, 2001.

Carlos Ordonez. Clustering binary data streams with k-means. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 12–19. ACM, 2003.

Randall C O'Reilly. Biologically based computational models of high-level cognition. *Science*, 314(5796):91–94, 2006.

Margarita Osadchy, Yann Le Cun, and Matthew L Miller. Synergistic Face Detection and Pose Estimation with Energy-Based Models. *The Journal of Machine Learning Research*, 8: 1197–1215, 2007.

Allan Paivio. *Imagery and verbal processes*. Holt, Rinehart & Winston, 1971.

Allan Paivio. *Mental representations: A dual coding approach*. Oxford University Press, 1990.

Lance Parsons. Evaluating subspace clustering algorithms. In *in Workshop on Clustering High Dimensional Data and its Applications, SIAM International Conference on Data Mining*, pages 48–56, 2004.

Adam Pease and Ian Niles. Ieee standard upper ontology: a progress report. *The Knowledge Engineering Review*, 17(1):65–70, 2002.

F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Witold Pedrycz. Collaborative fuzzy clustering. *Pattern Recognition Letters*, 23(14):1675–1686, 2002.

Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

Dzung Pham, Jerry L Prince, Chenyang Xu, and Azar P Dagher. An automated technique for statistical characterization of brain tissues in magnetic resonance imaging. *International journal of pattern recognition and artificial intelligence*, 11(08):1189–1211, 1997.

Steven Pinker. How the mind works. *Annals of the New York Academy of Sciences*, 882(1): 119–127, 1999.

Steven Pinker. *The language instinct: How the mind creates language*. Penguin UK, 2003.

Alexandre Miguel Pinto and Leandro Barroso. Principles of Regulated Activation Networks. In *Graph-Based Representation and Reasoning*, pages 231–244. Springer, 2014.

Alexandre Miguel Pinto and Rahul Sharma. Concept Learning, Recall, and Blending with Regulated Activation Networks (poster). In *Proceedings of the 14th International Conference on Cognitive Modeling*, pages 282–284. Penn State, 2016.

Roberto Poli. Ontology for knowledge organization. *Advances in Knowledge Organization*, 5: 313–319, 1996.

Sean M Polyn, Vaidehi S Natu, Jonathan D Cohen, and Kenneth A Norman. Category-specific cortical activity precedes retrieval during memory search. *Science*, 310(5756):1963–1966, 2005.

Friedemann Pulvermüller, Markus Härle, and Friedhelm Hummel. Walking or talking?: Behavioral and neurophysiological correlates of action verb processing. *Brain and language*, 78(2): 143–168, 2001.

J Ross Quinlan. Simplifying decision trees. *International Journal of Human-Computer Studies*, 51(2):497–510, 1999.

Rodrigo Quian Quiroga. Concept cells: the building blocks of declarative memory functions. *Nature Reviews Neuroscience*, 13(8):587, 2012.

Nicholas J Radcliffe. Genetic set recombination and its application to neural network topology optimisation. *Neural Computing & Applications*, 1(1):67–90, 1993.

Nicole S Rafidi, Justin C Hulbert, Paula P Brooks, and Kenneth A Norman. Reductions in Retrieval Competition Predict the Benefit of Repeated Testing. *Scientific Reports*, 8(1):11714, 2018.

Vijay V Raghavan and S K M Wong. A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science*, 37(5):279–287, 1986.

H Ralambondrainy. A conceptual version of the k-means algorithm. *Pattern Recognition Letters*, 16:1147–1157, 1995.

Vilayanur S Ramachandran, Sandra Blakeslee, and Neil Shah. *Phantoms in the brain: Probing the mysteries of the human mind*. William Morrow New York, 1998.

RbloggersKmeansplots. R-bloggers k-means plots. `http://www.r-bloggers.com/bot-botany-k-means-and-ggplot2/`, 2015. [Online; accessed June].

Stefano Recanatesi, Mikhail Katkov, Sandro Romani, and Misha Tsodyks. Neural network model of memory retrieval. *Frontiers in Computational Neuroscience*, 9:149, 2015.

Joseph Reisinger, Erkin Bahceci, Igor Karpov, and Risto Miikkulainen. Coevolving strategies for general game playing. In *Symposium on Computational Intelligence and Games*, pages 320–327. IEEE, 2007.

Eric Rignot, Rama Chellappa, and Pascale Dubois. Unsupervised segmentation of polarimetric sar data using the covariance matrix. *Geoscience and Remote Sensing, IEEE Transactions on*, 30(4):697–705, 1992.

C J Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. ISBN 0408709294.

John Ringdahl. Validation based cascade-correlation training of artificial neural networks. Online, 2020. URL `http://lup.lub.lu.se/student-papers/record/9007642`. Bachelor Thesis. Accessed online: June, 2020.

Brian D Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 2007.

H L Roediger, D A Balota, and J M Watson. Spreading activation and arousal of false memories. *The nature of remembering: Essays in honor of Robert G Crowder*, pages 95–115, 2001.

Henry L Roediger and Teresa A Blaxton. Effects of varying modality, surface features, and retention interval on priming in word-fragment completion. *Memory & cognition*, 15(5):379–388, 1987.

Henry L Roediger and Kathleen B McDermott. Creating false memories: Remembering words not presented in lists. *Journal of experimental psychology: Learning, Memory, and Cognition*, 21(4):803, 1995.

E Rolls, M Loh, G Deco, and G Winterer. Computational models of schizophrenia and dopamine modulation in the prefrontal cortex. *Nature Reviews Neuroscience*, 9:696–709, 2008.

Eleanor Rosch. Cognitive representations of semantic categories. *Journal of Experimental Psychology: General*, 104(3):192, 1975.

Eleanor Rosch. Prototype classification and logical classification: The two systems. *New trends in conceptual representation: Challenges to Piaget's theory*, pages 73–86, 1983.

Eleanor Rosch, Carolyn B Mervis, Wayne D Gray, David M Johnson, and Penny Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8(3):382–439, 1976.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1961.

Frank Rosenblatt. A comparison of several perceptron models. *Self-Organizing Systems*, pages 463–484, 1962.

Asim Roy. A theory of the brain - the brain uses both distributed and localist (symbolic) representation. In *International Joint Conference on Neural Networks, IJCNN*, pages 215–221, 2011.

D E Rumelhart, G E Hinton, and R J Williams. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, 1986.

E Ruppin and Y Yeshurun. Recall and recognition in an attractor neural network model of memory retrieval. *Connection Science*, 3(4):381–400, 1991.

Eytan Ruppin and Yehezkel Yeshurun. An attractor neural network model of recall and recognition. In *Advances in Neural Information Processing Systems*, pages 642–648. Morgan Kaufmann Publishers Inc., 1990.

Enrique H Ruspini. A new approach to clustering. *Information and control*, 15(1):22–32, 1969.

Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson Education Limited Malaysia, 2016.

Andrey Rzhetsky, Ivan Iossifov, Tomohiro Koike, Michael Krauthammer, Pauline Kra, Mitzi Morris, Hong Yu, Pablo Ariel Duboué, Wubin Weng, W John Wilbur, et al. GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. *Journal of Biomedical Informatics*, 37(1):43–53, 2004.

Lorenza Saitta and Jean-Daniel Zucker. Semantic abstraction for concept representation and learning. In *Proceedings of the Symposium on Abstraction, Reformulation and Approximation*, pages 103–120. AAAi, 1998.

G Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

G Salton, A Wong, and C S Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620, 1975.

Gerald Salton, editor. *Automatic Text Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1988. ISBN 0-2:1-1227-8.

Gerard Salton and Christopher Buckley. Term-weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5), 1988.

Gerard Salton and Michael J McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. ISBN 0070544840.

Elizabeth L Sampson, Christopher J Bulpitt, and Astrid E Fletcher. Survival of community-dwelling older people: the effect of cognitive impairment and social engagement. *Journal of the American Geriatrics Society*, 57(6):985–991, 2009.

Scott Sanner, John R Anderson, Christian Lebiere, and Marsha Lovett. Achieving efficient and cognitively plausible learning in backgammon. In *International Cconference on Machine Learning*, pages 823–830. Morgan Kaufmann Publishers Inc., 2000.

Vivian Santos, Manuela Hürliman, Brian Davis, Siegfried Handschuh, and André Freitas. Semantic Relation Classification: Task Formalisation and Refinement. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V)*, pages 30–39, 2016.

J D Schaffer, D Whitley, and L J Eshelman. Combinations of genetic algorithms and neural networks: a survey of the state of the art. In *Proceedings in International Workshop on Combinations of Genetic Algorithms and Neural Networks*, pages 1–37. IEEE, 1992.

Lael J Schooler and Ralph Hertwig. How forgetting aids heuristic inference. *Psychological Review*, 112(3):610, 2005.

Paula J Schwanenflugel. *Why are abstract concepts hard to understand?*, pages 223–250. Lawrence Erlbaum Associates, Inc, 1991.

Paula J Schwanenflugel, Carolyn Akin, and Wei-Ming Luh. Context availability and the recall of abstract and concrete words. *Memory & Cognition*, 20(1):96–104, 1992.

Claudia Scorolli, Ferdinand Binkofski, Giovanni Buccino, Roberto Nicoletti, Lucia Riggio, and Anna Maria Borghi. Abstract and concrete sentences, embodiment, and languages. *Frontiers in Psychology*, 2:227, 2011.

Per B Sederberg, Marc W Howard, and Michael J Kahana. A context-based theory of recency and contiguity in free recall. *Psychological Review*, 115(4):893, 2008.

O G Selfridge. Pandemonium: A paradigm for learning. In *Proceedings of the symposium on the mechanisation of thought processes*, pages 513–534, 1985.

Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3626–3633. IEEE, 2013.

Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In *International Conference on Learning Representations*. CBLS, 2014.

Thomas Serre, Gabriel Kreiman, Minjoon Kouh, Charles Cadieu, Ulf Knoblich, and Tomaso Poggio. A quantitative theory of immediate visual recognition. In Paul Cisek, Trevor Drew,

and John F Kalaska, editors, *Computational Neuroscience: Theoretical Insights into Brain Function*, volume 165 of *Progress in Brain Research*, pages 33–56. Elsevier, 2007.

Rahul Sharma, Alexandre Miguel Pinto, and Vivek Kumar Singh. Energy and resource usage-aware buildings via cognitive internet of things agents. pages 1–10, 2017a. URL `http://www.efs2017.uc.pt/projectos/efs2017/atas/pdfs/Id93.pdf`.

Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto, Amílcar F Cardoso, Duarte Raposo, Marcelo, André Rodrigues, Jorge Sá Silva, and Fernando Boavida. Computational Concept Modeling for Student Centric Lifestyle Analysis: A Technical Report on SOCIALITE Case Study. Technical report, Center of Information Science University of Coimbra, Portugal, 2017b.

Rahul Sharma, Bernadete Ribeiro, A M Pinto, and F A Cardoso. Perceiving Abstract Concepts Via Evolving Computational Cognitive Modeling. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018a.

Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto, and F Amílcar Cardoso. Modeling Abstract Concepts For Internet of Everything: A Cognitive Artificial System. In *13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO)*, pages 340–345. IEEE, 2018b.

Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto, F Amílcar Cardoso, Ngombo Armando, Duarte Raposo, Marcelo Fernandes, André Rodrigues, Jorge Sá Silva, Hugo Gonçalo Oliveira, et al. Unveiling Markers of Stress Via Smartphone Usage. In *Proceedings of the 24th Portuguese Conference of Pattern Recognition*, pages 130–131. Univeristy of Coimbra, 2018c.

Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto, and F. Amílcar Cardoso. Exploring geometric feature hyper-space in data to learn representations of abstract concepts. *Applied Sciences*, 10(6), 2020a.

Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto, and F Amílcar Cardoso. Learning non-convex abstract concepts with regulated activation networks. *Annals of Mathematics and Artificial Intelligence*, pages 1–29, 2020b.

Ekaterina Shutova. Models of Metaphor in NLP. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 688–697. Association for Computational Linguistics, 2010.

Lars Sivik and Charles Taft. Color naming: A mapping in the IMCS of common color terms. *Scandinavian Journal of Psychology*, 35(2):144–164, 1994.

Thabet Slimani. A study investigating typical concepts and guidelines for ontology building, 2015. arXiv preprint arXiv:1509.05434.

Benjamin D Smith and Guy E Garnett. Improvising musical structure with hierarchical neural nets. In *Proceedings of the AIIDE 2012 Workshop on Musical Metacreation*, pages 63–67, 2012.

Richard Socher, Samuel Gershman, Per Sederberg, Kenneth Norman, Adler J Perotte, and David M Blei. A Bayesian analysis of dynamics in free recall. In *Advances in Neural Information Processing Systems*, pages 1714–1722. Curran Associates Inc., 2009.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

John F Sowa. Ontology, 2015. URL `http://www.jfsowa.com/ontology/`. Accessed online: 19^{th} June, 2015.

K G Srinivasa, Amrinder Singh, A O Thomas, K R Venugopal, and L M Patnaik. Generic feature extraction for classification using fuzzy c-means clustering. In *Third International Conference on Intelligent Sensing and Information Processing*, pages 33–38. IEEE, 2005.

Nisheeth Srivastava and Edward Vul. A simple model of recognition and recall memory. In *Advances in Neural Information Processing Systems*, pages 293–301. Curran Associates, Inc., 2017.

K O Stanley, B D Bryant, and R Miikkulainen. Real-time neuroevolution in the NERO video game. *Transactions on Evolutionary Computation*, 9(6):653–668, 2005a.

Kenneth Stanley, Nate Kohl, Rini Sherony, and Risto Miikkulainen. Neuroevolution of an automobile crash warning system. In *Proceedings of the 7th annual conference on genetic and evolutionary computation*, pages 1977–1984. ACM, 2005b.

Kenneth O Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.

Kenneth O Stanley and Risto Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of artificial intelligence research*, 21:63–100, 2004.

Carlo Strapparava, Alessandro Valitutti, and Oliviero Stock. Automatizing two creative functions for advertising. In *Proceedings of the 4th International Joint Workshop on Computational Creativity*, pages 99–108, 2007.

W Nick Street, William H Wolberg, and Olvi L Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *Biomedical Image Processing and Biomedical Visualization*, volume 1905, pages 861–871. International Society for Optics and Photonics, 1993.

Ron Sun. The CLARION cognitive architecture: Extending cognitive modeling to social simulation. *Cognition and multi-agent interaction*, pages 79–99, 2006.

Ron Sun. *The Cambridge handbook of computational psychology*. Cambridge University Press, 2008.

Ron Sun and Frederic Alexandre. *Connectionist-symbolic integration: From unified to hybrid approaches*. Psychology Press, 2013.

Ron Sun and Todd Peterson. Learning in reactive sequential decision tasks: The CLARION model. In *International Conference on Neural Networks*, volume 2, pages 1073–1078. IEEE, 1996.

Ron Sun and Xi Zhang. Top-down versus bottom-up learning in cognitive skill acquisition. *Cognitive Systems Research*, 5(1):63–89, 2004.

Ron Sun, Todd Peterson, and Edward Merrill. A hybrid architecture for situated learning of reactive sequential decision making. *Applied Intelligence*, 11(1):109–127, 1999.

Ron Sun, Edward Merrill, and Todd Peterson. From implicit skills to explicit knowledge: a bottom-up model of skill learning. *Cognitive science*, 25(2):203–244, 2001.

Ron Sun, Nick Wilson, and Robert Mathews. Accounting for certain mental disorders within a comprehensive cognitive architecture. *Cognitive Computation*, 3(2):341–359, 2011.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence earning with Neural Networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Don R Swanson. Medical literature as a potential source of new knowledge. *Bulletin of the Medical Library Association*, 78(1):29, 1990.

Niels Taatgen. A model of free-recall using the ACT-R architecture and the phonological loop. In *In HJvd Herik & T. Weijters (Eds.), Proceedings of Benelearn-96*. Citeseer, 1996.

T Teijeiro, P Félix, J Presedo, and D Castro. Heartbeat Classification Using Abstract Features From the Abductive Interpretation of the ECG. *Journal of Biomedical and Health Informatics*, 22(2):409–420, 2018.

Asuka Terai and Masanori Nakagawa. A Neural Network Model of Metaphor Generation with Dynamic Interaction. In *International Conference on Artificial Neural Networks*, pages 779–788. Springer Berlin Heidelberg, 2009a.

Asuka Terai and Masanori Nakagawa. A neural network model of metaphor generation with dynamic interaction. In *International Conference on Artificial Neural Networks*, pages 779–788. Springer, 2009b.

Robert Thomson, Aryn Pyke, Laura M Hiatt, and J Greg Trafton. An Account of Associative Learning in Memory Recall. In *37th Annual Conference of the Cognitive Science Society*, pages 2386–2391, Pasadena, CA, 2015.

Miles Thorogood and Philippe Pasquier. Computationally created soundscapes with audio metaphor. In *Proceedings of the Fourth International Conference on Computational Creativity*, pages 1–7, 2013.

Peter M Todd. A connectionist system for exploring melody space. In *Proceedings of the International Computer Music Conference*, pages 65–68, 1992.

Toi.expert. BISOCIATION Tool, 2019. URL `https://toi.expert/tool/bisociation/?lang=en`. Accessed online: 2^{nd} January.

Nao Tokui, Hitoshi Iba, and etal. Music composition with interactive evolutionary computation. In *Proceedings of the 3rd international conference on generative art*, volume 17, pages 215–226, 2000.

Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656. IEEE, 2015.

J G Trafton, N L Cassimatis, M D Bugajska, D P Brock, F E Mintz, and A CS Schultz. Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 35(4):460–470, 2005.

Endel Tulving. Similarity relations in recognition. *Journal of Verbal Learning and Verbal Behavior*, 20(5):479–496, 1981.

Peter D Turney and Michael L Littman. Measuring Praise and Criticism: Inference of Semantic Orientation from Association. *ACM Transactions on Information Systems*, 21(4):315–346, 2003.

Amos Tversky. Features of similarity. *Psychological review*, 84(4):327, 1977.

Barbara Tversky and Kathleen Hemenway. Objects, parts, and categories. *Journal of Experimental Psychology: General*, 113(2):169–193, 1984.

Régis Vaillant, Christophe Monrocq, and Yann Le Cun. Original approach for the localisation of objects in images. *IEE Proceedings-Vision, Image and Signal Processing*, 141(4):245–250, 1994.

Frank van der Velde. Communication, Concepts and Grounding. *Neural Networks*, 62(C): 112–117, 2015.

Frank Van der Velde and Marc De Kamps. Neural blackboard architectures of combinatorial structures in cognition. *Behavioral and Brain Sciences*, 29(1):37–70, 2006.

Frank Van Der Velde and Marc De Kamps. Learning of control in a neural architecture of grounded language processing. *Cognitive Systems Research*, 11(1):93–107, 2010.

Arie Van Deursen and Tobias Kuipers. Identifying Objects Using Cluster and Concept Analysis. In *Proceedings of the 21st International Conference on Software Engineering*, pages 246–255. ACM, 1999.

Marcel Van Gerven and Sander Bohte. *Artificial neural networks as models of neural information processing*. Frontiers Media SA, 2018.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI*, volume 2, page 5. Phoenix, AZ, 2016.

Tony Veale. From Conceptual Mash-ups to Bad-ass Blends: A Robust Computational Model of Conceptual Blending. In *Proceedings of the Third International Conference on Computational Creativity*, pages 1–8, 2012.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, 2010.

Xue-Xin Wei and Alan A Stocker. A bayesian observer model constrained by efficient coding can explain'anti-bayesian'percepts. *Nature Neuroscience*, 18(10):1509, 2015.

Bernard Widrow and Marcian E Hoff. Adaptive switching circuits. Technical report, Electronics Labs Stanford University, 1960.

Wikipedia. Neuron. `https://simple.wikipedia.org/wiki/Neuron`, 2018. Online; accessed 15 August 2018.

S K M Wong, Wojciech Ziarko, and Patrick C N Wong. Generalized Vector Spaces Model in Information Retrieval. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 18–25. ACM, 1985.

S K M Wong, W Ziarko, V V Raghavan, and P C N Wong. On Modeling of Information Retrieval Concepts in Vector Spaces. *ACM Transactions on Database Systems*, 12(2):299–321, 1987.

Ping Xiao and Josep Blat. Generating apt metaphor ideas for pictorial advertisements. In *Proceedings of the Fourth International Conference on Computational Creativity*, pages 8–15, 2013.

Ping Xiao, Khalid Alnajjar, Mark Granroth-Wilding, Kat Agres, and Hannu Toivonen. Meta4meaning: Automatic metaphor interpretation using corpus-derived word associations. In *Proceedings of the 7th International Conference on Computational Creativity*, 2016.

Ping Xiao, Hannu Toivonen, Oskar Gross, Amílcar Cardoso, João Correia, Penousal Machado, Pedro Martins, Hugo Goncalo Oliveira, Rahul Sharma, Alexandre Miguel Pinto, Alberto Díaz, Virginia Francisco, Pablo Gervás, Raquel Hervás, Carlos León, Jamie Forth, Matthew Purver, Geraint A. Wiggins, Dragana Miljković, Vid Podpečan, Senja Pollak, Jan Kralj, Martin Žnidaršič, Marko Bohanec, Nada Lavrač, Tanja Urbančič, Frank Van Der Velde, and Stuart Battersby. Conceptual Representations for Computational Concept Creation. *ACM Computing Survey*, 52(1):1–33, February 2019. ISSN 0360-0300.

M S Yang. A survey of fuzzy clustering. *Mathematical and Computer modelling*, 18(11):1–16, 1993.

Jong Yoon, Vijay Raghavan, and Venu Chakilam. BitCube: clustering and statistical analysis for XML documents. *Journal of Intelligent Information Systems*, 17(2-3):241–254, 2001.

Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization, 2015. arXiv preprint arXiv:1506.06579.

Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, pages 116–124. ACM, 2004a.

Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116. ACM, 2004b.