中国科学技术大学

University of Science and Technology of China

# 博士学位论文

论文题目 A Study on Angular Softmax

作者姓名 Jamshaid UL Rahman

学科专业 计算数学

导师姓名 陈卿教授 & 杨周旺教授

完成时间 二〇二〇年五月

# 中国科学技术大学

# 博士学位论文

# A Study on Angular Softmax

作者姓名：　Jamshaid Ul Rahman

学科专业：　Computational Mathematics

导师姓名：　Prof. Qing Chen & Prof. Zhouwang Yang

完成时间：　二〇二〇年五月

University of Science and Technology of China

A dissertation for doctor's degree

# A Study on Angular Softmax

Author: Jamshaid Ul Rahman

Speciality: Computational Mathematics

Supervisor: Prof. Qing Chen & Prof. Zhouwang Yang

Finished time: May, 2020

# 中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文，是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外，论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名：＿＿＿＿＿＿　　　　签字日期：2020-05-22

# 中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一，学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权，即：学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文编入《中国学位论文全文数据库》等有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

保密的学位论文在解密后也遵守此规定。

☑公开　□保密（＿＿年）

作者签名：＿＿＿＿＿＿　　　　导师签名：＿＿＿＿＿＿

签字日期：2020-06-02　　　　签字日期：2020-06-02

I would like to dedicate this thesis to my beloved father, whose unyielding love, support, and encouragement have enriched my soul and inspired me to pursue and complete this work.

# Declaration

I am hereby declaring that the except where specific references are made from the other literature, the contents of the present study are original and not been submitted in the whole or in a part for consideration of any other degree or qualification before, or any other university or degree awarding organization. The present dissertation is my own research work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the acknowledgments.

Jamshaid Ul Rahman
May 2020

# Acknowledgements

I humbly thanks to ALLAH Almighty, the Merciful and the Beneficent, who gave me health, thoughts and co-operative people to enable me to achieve this goal.

Foremost, I have to thanks my research supervisors, **Prof. Qing Chen** and **Prof. Zhouwang Yang**. Without their assistance and dedicated involvement in every step throughout my research, this work would have never been accomplished. I would like to thank you very much for your support and understanding over these past three years.

I would also like to show gratitude to my thesis reviewers and defense committee, including Prof. Shen Liyong, Prof. He Jinsong, Prof. Chen Renjie, Prof. Zhang Juyong, Prof. Wu Chunlin and Prof. Li Xin.

During the past three years, I have many people to thanks for listening to me, tolerate me, and on their time and support. I cannot begin to express appreciation and my gratitude for their sincere friendship. Dr. Masood Ur Rehman, Muhammad Ajmal, Dr. Tanveer Hussain and Dr. Q Iqbal have been unwavering in their professional and personal support during the time I spent at the University. I would also like to thank all my labmats for many enjoyable hours during working together in the Laboratory of Data Sciences and Optimization. For many memorable evenings out and in, I must thank everyone above as well as Mansoor Iqbal and Muhammad Hassan. I cannot forget to express appreciation too few of my friends and teachers in the globe; Dr. M. Suleman Butt, Dr. A K Mangal, Dr. N Anjum Dr. Tasneem M.Shah, Dr. G. Murtza and Dr. Akhter ALi on the stimulating and encouraging discussions.

Most importantly, none of this could possible without my family. My father, who offered his most sincere encouragement via phone calls and supporting me throughout my life. Without their encouragement it would have been impossible for me to accomplish this work. My special gratitude is due to my brother Zulqarnain and sisters for their love and support. I would never be able to pay back the love and affection showered upon by my father, sisters and brother. I owe the deepest gratitude to my wife and beloved daughters for their continued and unfailing love, support, understanding, care and dedicated efforts which contributed a lot for completion of my work. I consider to myself as luckiest one in the world to have such a caring and lovely family, standing beside me with their unconditional support and love.

# Abstract

After the development of Deepface and DeepID methods in 2014, deep learning methods for image recognition has dramatically improved the state-of-the-art performance on Deep Convolutional Neural Networks (DCNNs) and reshaped the research landscape of image processing and data analysis. In spite of rapid improvement in deep learning algorithms, it still has various challenges like adjustment of appropriate loss function and optimization strategy to handle large scale problems in many computer vision applications including Face Recognition (FR) and Handwritten Digit Recognition (HDR). This thesis focus on these challenges and their better solution.

For both computer vision tasks, there are some advanced approaches based on the Convolutional Neural Network (CNN) that are able to learn image features via the softmax loss, but softmax can only learns those separable features that are considered not discriminative enough. A number of modifications set to boost up the discriminative power of the softmax and normally used for encouraging the separability of image features, but these modifications are not suitable for the intra-class variation. On this concern multiplicative angular margin, additive angular margin and additive cosine margin has been introduced to restrict the boundary closer to the weight vector of each class but the random selection of marginal values is again a big issue in multiplicative angular margin and additive angular margin. As a solution on this issue, we present a novel approach to handle these problems via presenting an additive parameter relative to multiplicative angular margin for DCNNs and reformulate the softmax loss through combined angular margin and additive margin. Moreover, an automatically fine-tuning is offered to adjust the additive parameter as a seedling element growing in the result of marginal seed. Experimental results on additive parameter demonstrate that our approach is better than numerous current state-of-the-art approaches using the similar network architecture and benchmarks.

On the other hand, there is no doubt Stochastic Gradient Descent (SGD) avoid spurious local minima and touch those that generalize well, but it decelerates the convergence of regular gradient descent. For linear convergence in strongly convex functions, numerous variance reduction algorithms have been intended, but only few of them are suitable to train DCNNs. A simple modification offered by recently deigned Laplacian Smoothing Gradient Descent (2018) dramatically reduces the optimality gap in SGD and applicable to train DCNNs. Motivated by Laplacian Smoothing Stochastic Gradient Descent (LS-SGD) and inspired from the additive parameter, we adopt this simple modification of gradient descent and stochastic gradient descent to design a novel strategy assembled with a modified form of softmax and LS-SGD. Our approach expresses a flexible learning job with adjustable additive margin and is flexible to amend with SGD and LS-SGD.

## Keywords:

Additive Parameter, Angular Margin, Deep Convolutional Neural Networks, Image Recognition, Softmax Loss.

# Table of contents

# List of figures

# List of tables

# Chapter 1

# Introduction

Deep learning or deep structured learning is a part of a very larger family of machine learning methods constructed via Artificial Neural Networks (ANNs). Deep learning is an ambiguous word, as it has gone over and done with several altered meanings during the years. A comprehensive, modern definition can be found in Goodfellow's book *Deep Learning* [23] and Skansi's book *Introdution to Deep Learning* [73]. There, it is defined as the decomposition of complex concepts into simple ones and the recombination into new complex concepts. A procedure therefore has to found a hierarchy of concepts. A visualization of hierarchy would be a multi-layered graph and can be called "Deep" in a graph theory context [10, 11].

The image recognition system is the labeling process practical to a segmented object of a scene, that system commonly requires basic feature extraction. In computer vision, the deep learning methods attain significant importance in advanced research on image recognition. In recent times, deep learning-based pattern recognition techniques have established extremely promising results in various computer vision applications. In deep learning, every individual level learns to transform the input data into some extent further abstract and compound representation. The raw input data may be a matrix of pixels in image recognition task, the layer one abstract the pixels and applicable to encode the edges, the next layer or the layer two may compose and capable to encode arrangements of the edges, the layer three may encode eyes and nose; and the layer four may be referring to recognize that the image contains that face. The recognition task initiate with the dataset into training stage, that containing random projection and extraction of the features with different architectures, optimization strategies, activation functions and loss functions used to extract discriminative deep feature. The important role in deep feature learning is to design a suitable loss function and gradient decent strategy to attain state-of-the-art performance. However, there are many types of loss function exists for DCNNs, but the softmax and the modified form of softmax considered more appropriate to handle image recognition tasks. Specially for face recognition and handwritten digit recognition, the performance of angular form of softmax and marginal angular form of softmax are better than the other forms of softmax.

In this study, we discussed the marginal based angular forms of softmax and the present study offer a supervision signal for discriminative image features through a modification in angular softmax to boost up the power of loss function. The present work includes two main applications of the image recognition via DCNNs. The overall focus is to model appropriate marginal angular softmax, as well as to adjust with the network and make it stable for the gradient decent and implement the constructed model on face recognition system and also make it applicable for the handwritten digit recognition.

This chapter is on brief introduction of the structure of artificial neural networks and image recognition system. In this chapter, we discuses about the artificial neural network and its applications in section 1.1 and a brief study on introduction to image recognition system is included in section 1.2, the subsections 1.2.1 and 1.2.2 contains basic introduction of both applications, Face Recognition (FR) and Handwritten Digit Recognition (HDR) systems respectively. The section 1.3 consists on the motivation and objectives of the main problem and a brief introduction of our proposed strategy to handle the challenges in recognition system is included in the section 1.4. The section 1.5 consists on the outline of this thesis. In next section, we have to look on artificial neural networks.

## 1.1 Artificial Neural Networks

Artificial neural networks (ANNs) are considered as computational models encouraged by biological neural networks, that applicable to approximate commonly unknown functions. Habitually, they are motivated by the actions of neurons and the signals that they transfer between input, processing, and output from the brain [100]. However, the goal of neural networks is not to model the human brain, the neural network research is fundamentally based on and driven through engineering and mathematical disciplines rather than biological brain function. To build a machine learning or deep learning algorithm, we have to define a model, a cost-function and a procedure to optimize, that befit the assembly of the data. A learning algorithm is built to approximate the given function $f$, for classification, $f$ maps an input $x$ to a class $y$.

The learning algorithm determined $\hat{f}$ with weight $w$ such that

$$y = \hat{f}(x;w). \tag{1.1}$$

Where, $\hat{f}$ be a composite function of any number of functions $\hat{f}_i$, with $i = 1,2,3,\ldots,n$ such that

$$\hat{f} = \hat{f}_n \circ \hat{f}_{n-1} \circ \cdots \circ \hat{f}_3 \circ \hat{f}_2 \circ \hat{f}_1. \tag{1.2}$$

This representation of the structure is known as network, where $\hat{f}_1$ represents the first layer or input layer, $\hat{f}_2$ denotes the second layer, and so on. The final layer $\hat{f}_n$ is called the output layer. There are $n-1$ hidden layers are between input and output layers, these layers are called hidden layers, as their performance is only implicitly constrained through the training data, such as the data do not show the desired output for each of those layers. Where $n$ be the number of layers, that defines the depth of the network. This modest chain structure of the complex structure is only one potential mode to build a network and was first introduced in 1958 by Frank Rosenblatt [66]. Presently, there are several kinds of ANNs (brief study is included in the section 2.1), the most famous are Feedforward, Regulatory feedback, Radial basis function and Recurrent neural network. In which the feedforward is the simplest type of ANNs, having different sub types, the convolutional neural network (CNN) or ConvNet is the largely used network composed from feedforward, constructed of one or more convolutional layers. There are many units in each layer that act parallel and each unit denotes a vector-to-scalar function. Maximum units described as accepting a vector of inputs $x$, and $W$ be the mapping from $x$ to $z$ with bias $b$, to transformation, that can be model as

$$z = W^T x + b. \tag{1.3}$$

The activation function of a node describes output of that node given by input. An activation function $g(z)$ is applied element-wise on $z$. The unit design is an active area of research and is not yet held by final guiding theoretical principles.

$$g(z) = activation(0,z). \tag{1.4}$$

.

There are many types of activation functions, Rectified linear unit (ReLU) [53], Leaky ReLU [51], and Parametric rectified linear unit (PReLU) [28] are favorable for image recognition task [95]. The information in a NN propagates in a forward direction via input nodes, by hidden-layers into the output node, that is called forward propagation. For training task, forward propagation can continue until it yields a cost in an output node. The back-propagation system permits the information from the cost to flow backward by the network in order to calculate the gradient. The back-propagation system generates the gradient of scalar $z$ with respect to one of $x$ in the graph, similarly next gradient with esteem to the next parent of $z$ in the graph. This process repeated backward over the graph until $x$ is reached.

A learning procedure needs a metric with which to evaluate the distance between two function, that metric is called loss function in neural networks, and the important problem in the neural network architectures is to explore appropriate loss functions. A number of techniques has been proposed by modifying the loss function to improve the performance of DCNNs, where the Euclidean loss and softmax loss gain a significant importance in development of deep feature learning and tendency towards learning with robust features is to emphasize DCNNs with extra discriminative information. A brief overview of loss functions is addressed in next chapter, that also addressing some special cases of multiplicative angular margin via adjusting the margin to integer value and with decimal values of margin, that could be adjustable with the convolutional layers to train the model.

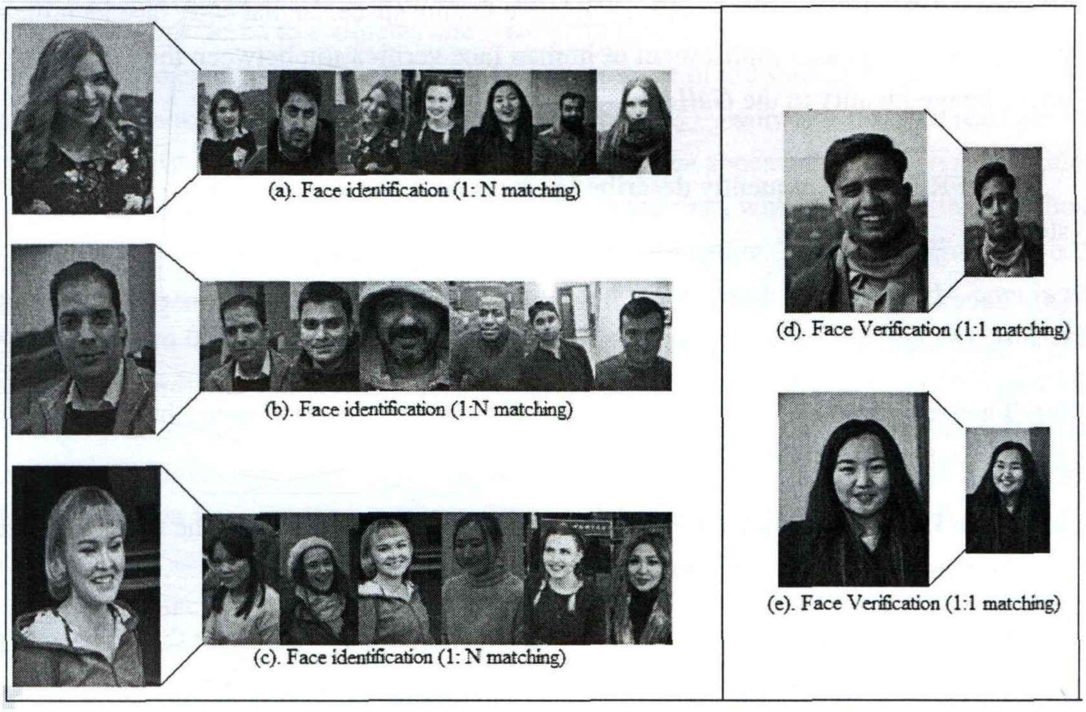
## 1.2 Image Recognition

Image Recognition is the automatic and programmed recognition of regularities and patterns in data. Image recognition is mainly related to AI [9], with modern world applications such as knowledge discovery databases and data mining. In current era of computer vision, image recognition using deep learning methods has achieved top scores on many tasks including FR and HDR. However, the deep learning methods have enabled rapid progress in many other applications, but for performance evaluation of the deep process, researchers commonly adopt these two main applications to evaluate their results. As our main focus is to build an appropriate deep learning approach for image recognition, therefor we also use these two main applications for better and fair comparison with already existing techniques.

In this section, we briefly introduce these two applications. The next subsection 1.2.1 is a brief overview of FR system using deep CNN.

### 1.2.1 Face Recognition

The FR system consists on multiple phases and considered as multipart problem of identifying and verifying people in an image or video by face. The task is trivially performed through the humans, even under the varying light and when faces changed via means of age or obstructed with accessories or facial hair. Yet, FR is remained a multifaceted and challenging computer vision problem for decades. Biometric approaches are castoff for either one of two purposes, identification or verification. In identification, the main objective is to identify the individual human face based on comparison of the face features collected against a data-set of previously collected samples of the human faces. In other words, a system that is designed for the purpose of identification will answer the question: "*Who is the subject*?".



(a). Face identification (1: N matching)

(b). Face identification (1:N matching)

(c). Face identification (1: N matching)

(d). Face Verification (1:1 matching)

(e). Face Verification (1:1 matching)

Fig. 1.1 Face Identification vs. Face Verification

On the other hand, in the verification applications, we wish to verify about the human face whether the human face is the person that they claim to be, that is done through the validating the collected human face features against a previously collected human face feature sample for the individual face. Biometric approach designed for verification could answer the question: "*Is the subject who he says he is*"?. The Fig. 1.1, explaining the Identification vs. Verification task, (a) represents the identification $(1 : N\ matching)$, a matching of one face of a girl from the data set of boys and girls, while (b) and (c) displays the one boy and one girl from the many boys and many girls. The (d) and (c), represents verification $(1 : 1\ matching)$, a boy from his own faces and a girl from her own faces respectively.

In the terms of testing procedure, FR system can be estimated under open-set or closed-set protocols. In closedset FR protocol, the testing images are predefined in the available training data set. It is normal to classify testing human face images to the given face identities. In this situation, face verification is equal to execution of the identification for a pair of human faces respectively. Thus, closedset FR can be well addressed as a classification problem and features are predictable to be separable. In open-set FR protocol, the testing images are generally disjoint from the training date set, that makes FR system more challenging task. Meanwhile it observed impossible to categorize faces to known identities in training data set, which need to face mapping to a discriminative feature space. In that case, face identification may be noticed as accomplishment of human face verification between the *probe* face and every image identity in the *Gallery*.

The FR system frequently described as a process that initially involves following four steps;

(a). **Face Detection:** Take the input image, locate the face area in the image and cropped or mark with the dots or bounding box.

(b). **Face Alignment:** Take the cropped face, normalize that cropped human face to be consistent with the available dataset.

(c). **Face Feature Extraction:** Take the normalized face and extract the features from the human face that can be further used for the face recognition task.

(d). **Face Recognition:** On the bases of feature extraction, FR execute the process of face matching counter to the different known faces already existing in the prepared dataset.

That system might have a distinct program or module for each step or could combine some or all steps together into a single procedure. Fig. 1.2 representing the overall structure of FR system.

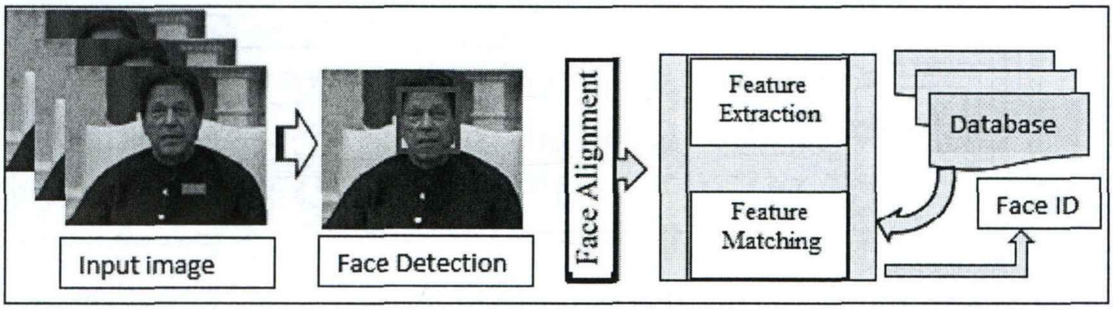


Fig. 1.2 General representation of initial process consisting on face detection to locate the face area in the image, face alignment to normalize that cropped human face, feature extraction to extract the features from the human face and face recognition to execute the process of face matching.

In present work, we focus on the deep FR module as an application to evaluate the performance of loss function. The deep FR is categorized as human face identification and human face verification. In both scenario, a set of the known subjects is primarily join up in the system *Gallery*, and during the the time of testing, a new subject *Probe* is obtained. The face verification considered to estimates $one - to - one$ similarity concerned between the *Probe* and *Gallery* to express whether the both face images are of the similar subject. However in face identification, it is applicable to computes $one - to - many$ similarity to regulate the specific identity related to the *Probe*. In case, when *Probe* appears in the *Gallery* identities, this is referred to as closed-set identification and in the case, when the *Probe* include those who are not in the *Gallery*, that is called open-set identification. The FR system consists of human face processing, deep feature extraction and facial matching, that can be seen in the Fig. 1.3, where faces in training data is localize via face detector and aligned to normalized canonical coordinates, then forward to CNN for feature extraction. There are different loss function with gradient descent are use to extract discriminative deep feature during training process and human face matching methods are used to face feature classification task when the deep feature from the human faces of testing data are extracted.

The loss function and gradient descent are used with the deep CNN to judge the performance of any neural network and play an important role in CNN training. DCNN produces a huge loss, if the neural network does not perform sound using the present parameter setting. Practically, FR methods achieve high performance, the significant part in deep feature learning is to design a suitable loss function to attain state-of-the-art performance. Intended loss functions are described on the bases of Euclidean distance, angular margin and softmax loss. The softmax loss function is normally used for encouraging the separability of facial

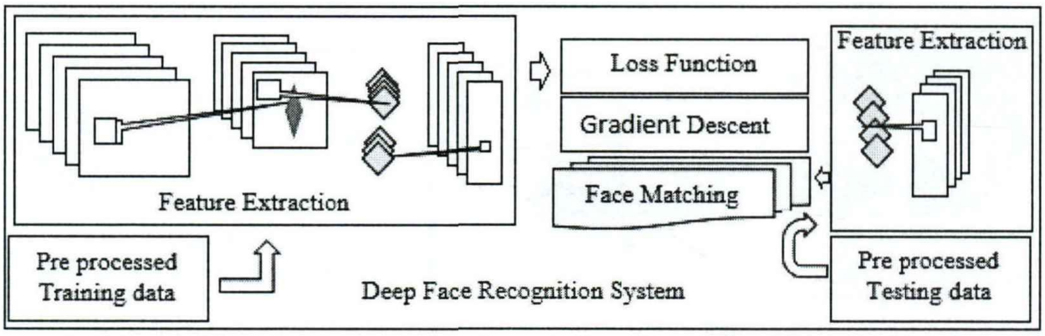

Feature Extraction
Loss Function
Gradient Descent
Face Matching
Feature Extraction
Pre processed Training data
Deep Face Recognition System
Pre processed Testing data

Fig. 1.3 Data after pre-processing through face detector and alignment used as training data pass in deep FR system. Initially FR handle recognition difficulty before starting the training and testing tasks. There are different CNN architectures, loss functions and gradient decent approaches used to extract deep feature during training process. In testing phase, the face matching approaches are used in the feature classification task.

features. For outstanding performance on FR system, the modern research is to train DCNNs based on softmax approaches and triplet loss approaches. A brief literature review of image recognition with different, stable and adjustable loss function is included in next chapter. An introduction to another application is given in next subsection.

### 1.2.2 Handwritten Digit Recognition

The HDR System includes interpretation and reception of digits written by human via a machine. Due to the dissimilarity, variation and difference in pattern and shape of handwritten digits [52], it is very difficult for an automatic machine to understand the handwritten digits. Handwritten digit Recognition has an extensive area of advanced research due to its massive applications like automatic cheques processing in bank, billing and for automatic postal service. The human writing styles are naturally differing in pattern and shape from person to person that makes HDR a challenging task. The Fig. 1.4 representing some examples of the handwriting digits and the recognized digits as results.

The HDR system frequently described as a process that involves following main steps;

(a). **Pre-processing:** Pre-processing contains five steps, size normalization and centering, interpolating missing points, smoothing, angle alteration and resampling of points.

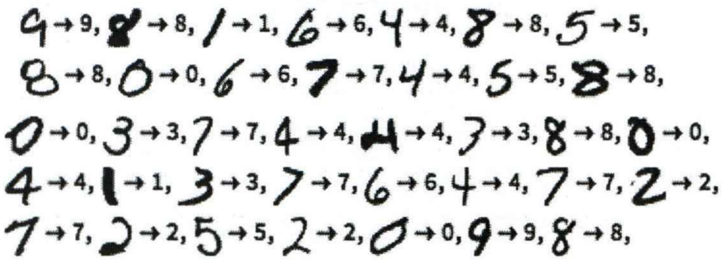


Fig. 1.4 Examples of digits with different pattern, shape & style and the corresponding recognized digit.

(b). **Segmentation:** In segmentation phase, the individual characters of an image are separated or we can say, this module applicable to segments the image into isolated characters.

(c). **Digit's Feature Extraction:** The key purpose of feature extraction stage is to extract that pattern which is most pertinent for classification.

(d). **Classification and Recognition:** On the bases of feature extraction, artificial neural network based classifiers compare the input feature with existing pattern in the prepared dataset and find out the appropriate matching.

DCNNs architecture is renowned for the extraction of complex features and the important problem in the neural network architectures is to explore appropriate loss functions. A loss function is incredibly simple method of evaluating how the algorithm models the dataset. Most deep learning processes use some sort of loss function in the process of optimization, or to search the appropriate weights for given data. Several techniques with motivated concepts and structure are presented during the last two years by introducing angular margin margin for the recognition tasks.

In the both above mentioned applications, the selection of loss function play a vital role to handle the difficulty and to improve the performance of complete architecture. Keeping this point, we focus to model a appropriate, suitable and adjustable loss function for image recognition tasks.

## 1.3 Motivation and Objectives

In resent time, the image recognition received a blooming interest and attention from the scientists and industrial researchers as well as from the modern student's community. The interest from students and general public is frequently due to the modern development in mobile technology, internet shopping, robotics and security issues, that increased the demand for useful applications as well as for advanced security systems. To construct numerous artificial systems, including above mentioned applications, accurate, perfect and robust automated image recognition systems, algorithms, methods and techniques are required. The main aim of this research is to design a better and reliable technique for image recognition system bases on the adjustment, reforming and amendment in different steps of CNNs including adjustment of appropriate loss function and optimization strategy to handle large scale problems.

The solution proposed in this study is helpful to solve the difficult task of robust face recognition, object recognition, handwritten digit recognition and action recognition tasks and is applicable to improve the performance of structure of CNN for other many computer vision tasks, such an impact would be of countless scientific importance in modern deep learning approaches and would be valuable for researchers and developers from academic and industrial community.

The main objectives of this work listed below:

(a). To summarize the process of image recognition and model a proper strategy to improve the performance of deep convolutional neural network for image recognition.

(b). To minimize the loss or error to train large scale data set via adjusting a proper loss function, parameters and gradient descent strategy.

## 1.4 Main Contribution

The original involvement of this contribution is to investigate the proper and suitable parameters for the marginal softmax loss function and modification based on the anguler and additive margin in DCNNs to enhance the performance of the recognition system. The softmax loss function is normally used for encouraging the separability of image features. For outstanding performance of image recognition system, the modern research is to train DCNNs based on angular margin, additive angular margin and additive cosine margin. We present a novel approach to handle the random selection of marginal values via presenting an additive parameter relative to multiplicative angular margin for DCNNs and reformulate the softmax loss through combined angular margin and additive margin. Moreover, an automatically fine-tuning is offered to adjust the additive parameter as a seedling element growing in the result of marginal seed. Furthermore, inspired by Laplacian Smoothing Stochastic Gradient Descent and encouraged from our proposed additive parameter, we adopt that simple modification of gradient descent and stochastic gradient descent to design a novel system assembled with a modified softmax and LS-SGD.

This study focus on the following two main challenges and better solutions on these challenges.

#### Challenge I

For effective and accurate discriminating performance, a number of modifications set to boost up the discriminative power of the softmax loss but it is not good for the intra-class variation. On this concern researcher proposed a scale parameter generating gradients to the separated samples for intra-class variance. All these approaches are based on the similar strategy for maximizing inter-class variance and minimizing intra-class variance. Introducing different margin to restrict the boundary closer to the weight vector of each class, we can adopt multiplicative angular margin, additive angular margin and additive cosine margin. In the case of marginal loss function, the problem is to adjust the additive marginal values appropriately by the use of integer values and decimal margin values.

**Strategy to handle:**

We present a novel approach to handle the first issue described above via presenting an additive parameter relative to multiplicative angular margin for DCNNs and reformulate the softmax loss through combined angular margin and additive margin. Moreover, an automatically fine-tuning is offered to adjust the additive parameter as a seedling element growing in the result of marginal seed.

#### Challenge II

On the other hand, there is no doubt SGD avoid spurious local minima and touch those that generalize well, but it decelerates the convergence of regular gradient descent. The performance of SGD to improve the power of algorithm remains of interest, particularly to optimize nonconvex function.

**Strategy to handle:**

For the second task, we adopt a bit similar modification offered by Laplacian Smoothing Gradient Descent (LS-SGD) to design a novel strategy assembled with a modified angular softmax and LS-SGD that dramatically reduces the optimality gap in SGD.

To evaluate the performance of our methodology, we experiments on two major image recognition applications and concluded on the basis of experiments that our approach is better than numerous current state-of-the-art approaches. To evaluate for Face Recognition, we train the model on publically available dataset CASIA-WebFace and our experiments on famous benchmarks YouTube Faces (YTF) and Labeled Face in the Wild (LFW) achieve better performance. Also the performance of the model is evaluated for handwritten digit recognition, experimental results demonstrate a state-of-the-art performance on famous database of handwritten digits the Modified National Institute of Standards and Technology (MNIST) database.

The conclusion highlights of this work are summarized below.

(a). This study presents a novel mode towards additive and angular margin softmax loss by offering an additive parameter controlled over multiplicative angular margin, that is more effective for DCNNs to learn highly discriminative features.

(b). The proposed model is trainable, extremely easy to implement and is appropriate for the larger amount of margin. Experimental results on additive parameter demonstrate that our approach is better than numerous current state-of-the-art approaches.

(c). Our approach expresses a flexible learning job with adjustable additive margin and is flexible to amend with SGD and LS-SGD.

## 1.5 Road Map

The rest of the dissertation is consisting on literature review and related work, methodology and its applications, concluding remarks and references.

The brief literature review and related work is presented in Chapter 2 of the thesis. This chapter is divided into two sections. In first section, we discussed about the neural architectures, methods and algorithm, that are modeled during the last few years. A summery on neural networks and methods based on the loss functions is included in first part of the chapter. The second part of the chapter is based on the most relevant related research and strategy behind latest models.

An efficient technique for image recognition is proposed in Chapter 3 of this study. This chapter containing the modified form of loss function. We explain the modeling of our proposed approach based on the modification on the loss function, the angular form is discussed with the margin values and margin strategy is presented. An automatic adjustment of the seedling element as the result of angular marginal seed is offered. An analysis on mathematical formulation, compression with margin base methods and significant impact is included in the same chapter.

The Chapter 4 contain two famous computer vision application. We implement our proposed methodology on Face Recognition (FR) and Handwritten Digit Recognition (HDR) systems. The complete process for FR and HDR is described in this chapter with implementation details, experiments, there results and analysis. For FR, we train our model on publicly available face dataset CASIA WebFace and evaluate the performance of proposed approach on famous benchmarks YouTube Faces (YTF) and Labeled Face in the Wild (LFW). For HDR, the performance is evaluated on famous database of handwritten digits the Modified National Institute of Standards and Technology (MNIST) database.

The Chapter 5 of this study contain conclusion from the proposed work and some future directions.

# Chapter 2

# Literature Review and Related Work

The present chapter is organized for the brief literature review and related work. This chapter is divided into two sections, in first section 2.1, we discuss about the neural architectures and methods, that are modeled during the last few years. A summery on neural networks and methods based on the loss functions is included in first part of this chapter. The second section 2.2 of the chapter is based on the most relevant related research and strategy behind latest methods.

## 2.1 Brief Literature Review

In the present era of Artificial Intelligent (AI), Deep Convolutional Neural Networks (DCNNs) have remarkable improvement in the state-of-the-art performance in numerous computer vision tasks [1, 5, 8, 37, 47, 48, 62, 69, 84, 98] including object recognition, face recognition, speech recognition and hand-written digit recognition. Modern digital world is full of shapes, design and patterns, moreover these patterns can be physically viewed, recognized or derived using mathematical tools with algorithms.

Recognition system is being conventionally associated to the field of AI in which computational mathematics plays a major role in the development and enhancement of methods and algorithms. An appropriate ideal and better recognition approach would contemplate classification matters as well as representation and demonstration. Day by day AI applications facing challenges owing to large amount of complex data structures, on the other hand neural network refining deeper and complex architecture. The famous networks are shown in Fig. 2.1. A perceptron is a very simple model of the neuron that take the inputs from the data, take some operation, applies through activation and forward to the output layer.

All the nodes are fully connected in the Feed forward neural network (NN), the activation streams from the input layer to the output layer, without back loops, there is one hidden layer between input and output. The Residual basis feed forward (RBF) are feed forward neural network and it use a radial-basis mode as activation for the substitute of the logistic function. the Deep feed forward (DFF) neural networks open a new direction towards deep learning, there are just FFNNs, but with more than one hidden layers. The Recurrent Neural Networks (RNN) present many kinds of recurrent-cells. The first NN of this category is called $Jordannetwork$ [43], where the every hidden cell received it's own output after fixed interval. Long or short term memory LSTM [6], presents a $memory - cell$, that can process data with the time gaps. The RNNs may process the texts by "keeping in mind" the ten previous words, while LSTM networks referred to process video frame "keeping in mind" the something that occurred before many frames. The network LSTM are widely applicable for the handwritten digits and speech recognition.

Gated Recurrent Units (GRUs) are similar to LSTMs with different gating units and are less resource consumer as compaied to LSTMs [16]. The Auto-Encoders (AE) [50] represented for classification, clustering and feature compression. The AE is capable to train data without supervision and known to compress features, while Vibrational Auto-Encoders (VAEs) [21] compress probabilities instead of features. Although AEs are cool, but sometimes, instead to judge the most robust features, they just adapt to input data that is an example of overfitting, and Denoising Auto-Encoders (DAEs) [83] add a bit of noise on the input cells and randomly switch bits in input, and Sparse Auto-Encoders (SAE) reveal some hidden grouping patters [64]. Like the above, the Markov Chains (MC) [3], Hopfield Networks (HN)[7], Boltzmann Machines (BM), Restricted Boltzmann Machines (RBM) and Deep Belief Network (DBNs) are also deals traditionally. The most commonly applied to analyzing visual imagery, the Deep Convolutional Network (DCN) or deep convolutional neural network (DCNN) currently are stars of ANN, most commonly applied to analyzing visual imagery the pooling layers or feature convolution cells and kernels are serving different jobs [69], convolution kernels process input data and pooling layers simplify that data and reducing unnecessary features [71]. Naturally castoff for the image recognition and operate on the small subset of image (roughly about $20x20pixels$ ). On the image, there is input sliding window slide on the image, pixel by pixel. That prepossessed data is forward to convolution layers, first layer referred to detects gradients, second layer for lines, third layer for shapes, and so on to the scale of specific objects. Deconvolutional Networks (DNs) are

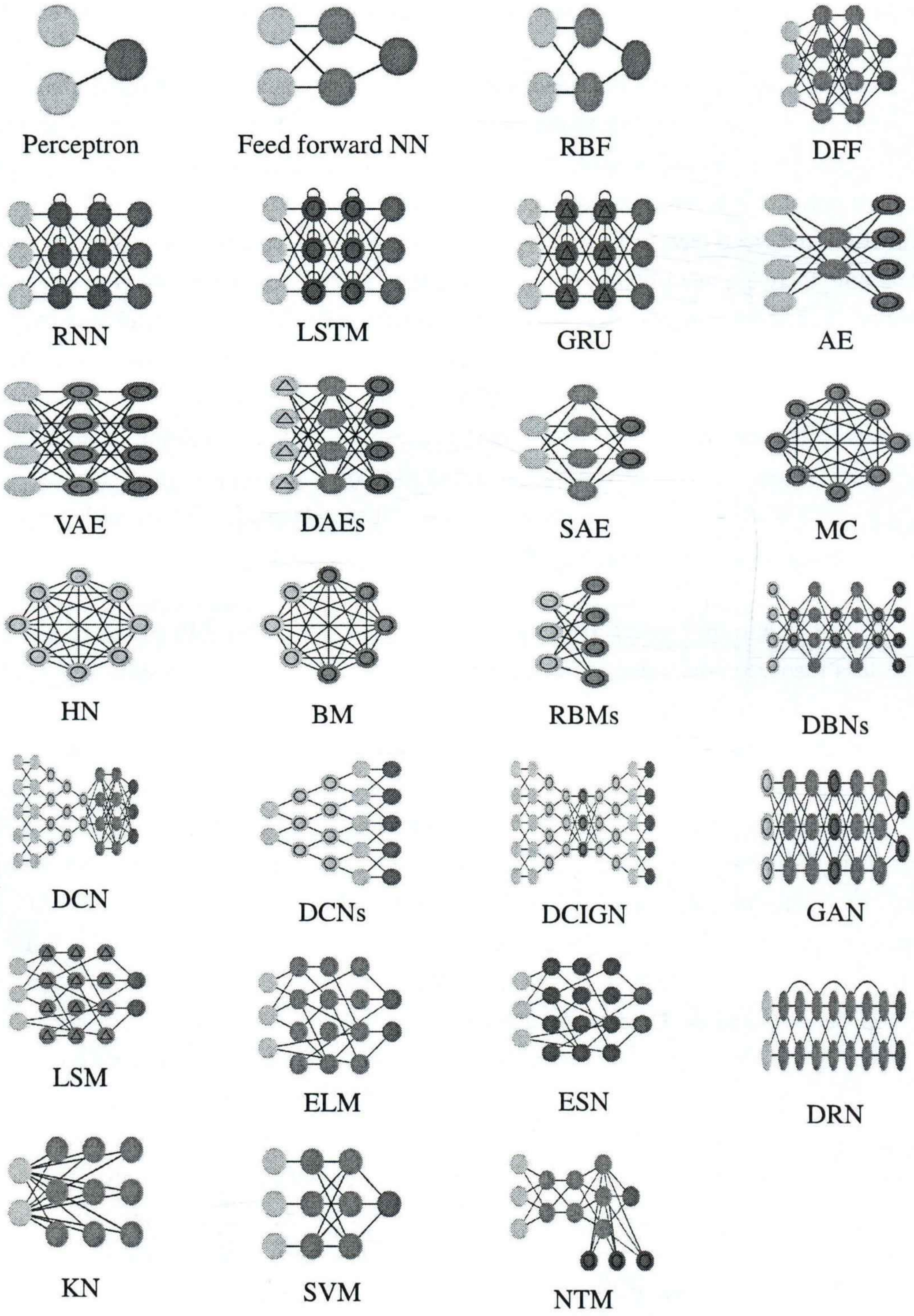


Fig. 2.1 Andrew's explanation [81] on neural networks topological cheatsheet.

DCNs overturned network, for object recognition task, DN takes cat image, and yields a vector such as $[cat : 1,\ dog : 0,\ horse : 0]$. A model, Deep Convolution Inverse Graphics Network (DCIGN) [35] aims to learn an interpretable representation of images, disentangled with respect to 3-dimensional structure and viewing transformations such as depth rotations and lighting variations. Generative Adversarial Network (GAN), a network represents a large family of double networks [24], that are composed from discriminator and generator. The activation functions are replaced by threshold levels in Liquid-state machine (LSM), while Extreme Learning Machine (ELM) [93] is an attempt to reduce complexity behind FF networks and Echo-state Network (ESN) [63] is a subtype of recurrent networks with a special training approach. Deep residual network (DRN) is a kind of RNN, while kohonen network (KN) and Support Vector Machine (SVMs) [61] are working as traditional way and not always considered to be a neural network. Neural Turing Machines (NTMs)[26], a new class of recurrent neural networks which decouple computation from memory by introducing an external memory unit, get results, enhance them but the actual decision path is typically hidden.

For learning predictive simulations used for complex tasks frequently needs very large amounts of annotated data. For instance, the Convolutional Neural Networks (CNNs) are large in dimension and naturally contain more parameters than the training samples, that raises a number of challenges. Due to the nonintrusive and ordinary characteristics, the image recognition has become a prominent biometric modern technique for identity authentication and has been extensively used in many modern areas of technology, such as military, finance, language translation and security [89]. In image recognition tasks, Face Recognition (FR) and Handwritten Digit Recognition (HDR) are the main applications discussed in recent years for performance evaluation of DCNNs.

In the early 1990s, succeeding the primer of the modern Eigenface method used for FR and detection via determining the variance of image in a collection of face images, the study of image recognition became popular [82]. In early 2000s, FR gave rise to local-feature-based image recognition. Gabor feature based classification[44] and Face description with local binary patterns [2], as well as their high-dimensional extensions [101], succeeded robust performance over some invariant possessions of the local filtering. Unluckily, handcrafted image features suffered from a lack of particularity and solidity. In early 2010s, the learning-based local descriptors were presented for the image recognition community [14, 40] , where the local filters was learned for well particularity, and the encoding codebook is learned for better compactness. Although promoting from the robust learning ability, CNNs also have to face the crucial problems of overfilling. Significant effort such as large-scale training data [68], ImageNet classification with deep CNNs [34], data augmentation [34, 78], regularization [25, 30, 74, 85] and stochastic pooling [96] has been put to address on this issue. In 2014, DeepFace [80] and DeepID [55] attained the significant improvement and achieved the state-of-the-art accuracy on famous benchmark *Labeled Face in the Wild* (LFW) [31], approaching performance on the unconstrained condition for the first time ( Human: 97.53% vs. DeepFace: 97.35%). Since then, research concentration has moved to deep-learning-based approaches, and the accuracy was histrionically boosted to above 99.80% in just three years.

Before 2017, the Euclidean-distance-based loss function played an important role, but after that the angular-margin-based loss functions as well as feature and weight normalization considered favorable[89]. It is observed that, although a number of loss share similar basic idea, the new one is frequently premeditated to assist the training procedure by means of easier parameter or sample selection. A modern trend towards feature learning with even stronger image features is to reinforce CNNs with more discriminative information. Naturally, the learned features of image are good if intra-class compactness and inter-class separability are instantaneously maximized [18, 47, 48, 88]. While this may not be easy due to the inherent large intra-class variations in many responsibilities [48], the solid representation ability of CNNs sorts it possible to learn invariant features on the road to this direction. A number of techniques has been proposed by modifying the loss function to improve the performance of DCNNs [4, 45, 47, 60, 70, 75], where the Euclidean loss and softmax loss gain a significant importance in development of deep feature learning and tendency towards learning with robust features is to emphasize DCNNs with extra discriminative information.

In present era, the Deep CNNs architecture is renowned for the extraction of complex features and the important problem in the neural network architectures is to explore appropriate loss functions. A number of CNN architectures are proposed in the last few years, including, GoogLeNet [78], VGGNet [72], AlexNet [34] and ResNet [29]. The performance of FR system is important in terms of high specification devices and mobile devices, normally researcher evaluate the performance of loss functions on ResNet [4] and MobileNet [31] architectures. The ResNet represents a standard DCNN architecture with deep structure and extremely used in research and demanding computer vision tasks. The ResNet architecture is made with building blocks of residual units, that are demonstrated in Fig. 2.2. The ResNet unit learns a mapping between inputs and outputs using residual connections.

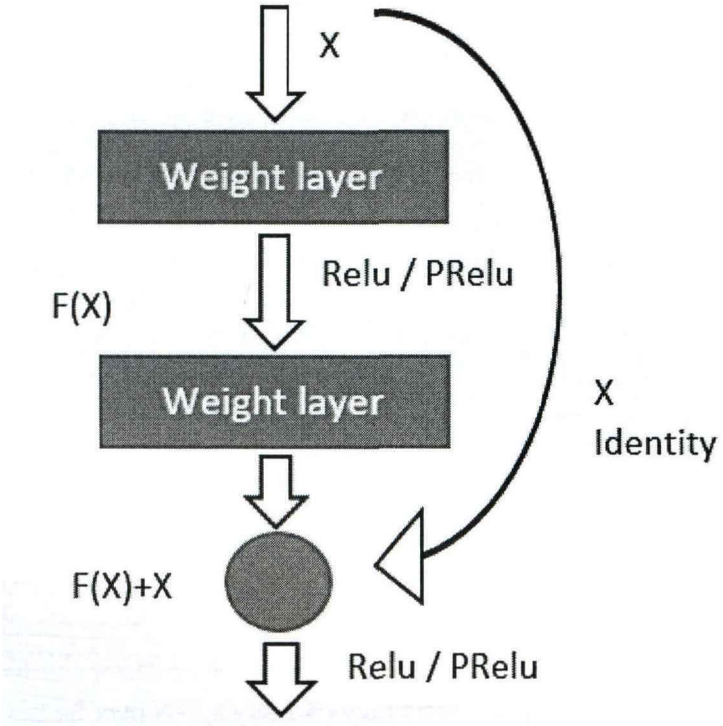


Fig. 2.2 The Basic Residual block used in ResNet [29]. A residual block is considered as a function of $X$ where $X$ is the input and $F(X)$ is the function on $X$ and $X$ is added to the output of $F(X)$, given that output $F(X)$ has the same dimension to that of $X$.

The development in the performance in image recognition was pragmatic along with the line of growing depth of the CNN architectures such as GoogLeNet [79] and ResNet [29]. Though, it is initiate that after certain depth, the performance of CNN tends to saturate on the way to mean accuracy, i.e., more depth has practically no effect over performance [29]. At the same time, applications in the large scale image recognition would be prohibitive due to the need of high computational resources for deep architectures.

In recent days, scientists are also working over the additional aspects of the CNN model like loss functions and optimizers. One of the main mechanism done in this field contains the progress of suitable loss functions, specifically designed for image recognition. Early job towards loss functions contain Center-Loss [91] and Triplet-Loss [70] that motivated on reducing the distance between the positive sample and current sample and rise the distance for the negative ones, thus closely linking to image recognition. Present modern well-known

| Year | Method | Loss Function | Architecture | Training Set | Acc (%) |
|---|---|---|---|---|---|
| 2014 | DeepFace [80] | Softmax | Alexnet | Facebook | 97.35 |
| 2014 | DeepID2 [75] | Contrastive | Alexnet | CelebFaces+ | 99.15 |
| 2015 | FaceNet[70] | Triplet | GoogleNet24 | Google | 99.63 |
| 2015 | DeepID3[76] | Contrastive | VGGNet10 | CelebFaces+ | 99.53 |
| 2015 | Baidu [46] | Triplet | CNN-9 | Baidu | 99.77 |
| 2015 | VGGface[56] | Triplet | VGGNet-16 | VGGface | 98.95 |
| 2015 | Light-CNN [94] | Triplet | light CNN | MS-Celeb-1M | 98.8 |
| 2016 | Center Loss [91] | Center | Lenet | CASIA-WebFace | 99.28 |
| 2016 | L-Softmax [48] | L-Softmax | VGGNet-18 | CASIA-WebFace | 98.71 |
| 2017 | Range Loss [102] | Range | VGGNet-16 | CASIA-WebFace | 99.52 |
| 2017 | NormFace [87] | Contrastive | ResNet-28 | CASIA-WebFace | 99.19 |
| 2017 | vMF loss [27] | vMF | ResNet-27 | MS-Celeb-1M | 99.58 |
| 2017 | Marginal Loss [20] | Marginal | ResNet-27 | MS-Celeb-1M | 99.48 |
| 2017 | SphereFace [47] | A-Softmax | ResNet-64 | CASIA-WebFace | 99.42 |
| 2018 | CCL [57] | CenterInvariant | ResNet-27 | CASIA-WebFace | 99.12 |
| 2018 | AMS [86] | AMS | ResNet-20 | CASIA-WebFace | 99.12 |
| 2018 | Ring loss [103] | Ring | ResNet-64 | MS-Celeb-1M | 99.50 |
| 2018 | CosFace [88] | CosFace | ResNet-64 | CASIA-WebFace | 99.33 |
| 2019 | L2-Softmax [59] | L2-Softmax | ResNet-101 | MS-Celeb-1M | 99.78 |
| 2019 | ArcFace [19] | ArcFace | ResNet-100 | MS-Celeb-1M | 99.83 |
| 2019 | Our approach [58] | Modified Softmax | ResNet-36 | CASIA-WebFace | 99.58 |

Table 2.1 Performance summery (accuracy (%)) of DCNN over LFW dataset using different methods, architectures and loss function.

loss functions like Soft-Margin-Softmax [42], Range-Loss [102], Congenerous-Cosine [49], Minimum-Margin Loss for deep FR [90], L2-Softmax Loss [60], Large-Margin Softmax Loss [48], CosFace [88], ArcFace [18] and A-Softmax Loss [47] have exposed capable performance over lighter CNN models and some over and above results over large CNN models.

A theoretically attractive angular margin used in A-Softmax [47] have amazing performance on face recognition and is introduced to push the classification boundary closer to weight of each class to encourage the discriminability of features. That famous contribution offer a new direction for the researchers to focus on different margin to restrict the boundary closer to the weight vector of each class. Several techniques with motivated concepts and structure are presented during the last two years by introducing multiplicative angular margin, additive angular margin and additive cosine margin [48, 88, 18].

For optimization task to train the network with loss function, typically used Stochastic Gradient Descent [12] is a gradient based optimization processes circumvent spurious local minima, but it slows down the convergence of regular gradient descent [69, 54]. Numerous stimulating variance reduction techniques [17, 33] have been proposed for strongly convex functions to recover the linear convergence rate, but not appropriate to train DCNNs. Laplacian Smoothing Stochastic Gradient Descent [54] is beneficial to reduce noise in SGD and appropriate to use for training of deep neural network. In spite of rapid improvement in deep learning algorithms, it still has various challenges including adjustment of appropriate loss function and optimization strategy to handle large scale problems.

## 2.2 Related Work

The rise in the performance of algorithms for image recognition is apparent along the line of rising depth of the network architectures such as ResNet [29] and GoogLeNet [79]. It is observed in average modern network architectures, that after definite and certain depth, the network's performance has a tendency to saturate towards mean accuracy, i.e., more and more depth has practically no effect over performance [29]. Parallel to this, a large scale application of image recognition system would be expensive owing to the need of high computational resources for deep architectures. Thus, in recent time, to better the performance, the computer vision community are also working on the other phases of the CNN model like loss functions and optimizers, etc. The primary and foremost works done in this area consist on the development of suitable and appropriate loss functions. Initial works towards loss functions include Triplet Loss [70] and Center Loss [91] which concentrated on decreasing the distance between the positive sample and the current sample and rise the distance for the negative ones, thus closely linking to human recognition.

Modern research in deep learning approaches on loss function and optimization strategy [5, 8, 37, 47, 60, 54, 84, 98] play a important role in deep recognition system to improve the performance of network. Naturally, softmax loss is considered good to optimize inter-class difference and effective to stabilize un-normalized vector to a probability distribution, that's why classification loss functions for DCNNs are habitually constructed by softmax loss. Mostly, loss function is modified using the original softmax loss [47, 48, 60, 70, 75, 98] and the optimizer is adjusted on the base of gradient based optimization techniques [54, 67, 99]. If $x_i$ and $y_i$ be the input feature and its labels respectively then the simple form of original softmax loss $L_S$ for $N$ number of training samples is given as

$$L_S = \frac{1}{N}\Sigma_i(-log\frac{e^{f_{y_i}}}{\Sigma_j e^{f_j}}) \tag{2.1}$$

The notation $f_j$ is the *j-th* element corresponding to the class vector $f$. The softmax loss attained good results on its modified form, the angular-margin and cosine-margin loss [18, 47, 88] has been designed to learned features theoretically separable with angular distance. That work on sofmax loss with adjustment on angular discriminative strategy is considered beneficial for classification task to improve the performance of DCNNs for recognition system and has been discussed in next subsections.

### 2.2.1 Deep Hypersphere Embedding

To model a proper expression designed for the modified softmax loss, we initially define input feature $x_i$ and the label $y_i$, and recall the original softmax loss, where $f_j$ represent the *j-th* element (and $j \in [1;k]$, for $k$ class) of the class score vector $f$, for $N$ number of training samples. In DCNNs, $f$ referred to the output of a fully connected layer, therefore $f_j = {W_j}^T x_i + b_j$ and $f_{y_i} = {W_{y_i}}^T x_i + b_j$, where $x_i$ be the training sample, $W_j$ and $W_{y_i}$ be the *j-th* and $y_i$*-th* column of $W$ respectively. This modification used in softmax, we obtain

$$L_i = -log\frac{e^{{W_{y_i}}^T x_i + b_{y_i}}}{\Sigma_j e^{{W_j}^T x_i + b_j}}. \tag{2.2}$$

If $\theta_i$ be the angle between $W_i$ and $x$, then we can write the angular form of the softmax as

$$L_i = -log\frac{e^{||{W_{y_i}}^T||||x_i||cos(\theta_{y_i},i)+b_{y_i}}}{\Sigma_j e^{||{W_j}^T||||x_i||cos(\theta_j,i)+b_j}}, \quad \forall\ 0 \leq \theta_j, i \leq \pi. \tag{2.3}$$

As stated above, via normalizing $||W_j|| = 1$, $\forall\ j$ in every iteration and zero the biases. Then we have the modified softmax loss, that is

$$L_{modified-Softmax} = \frac{1}{N}\Sigma_i\left(-log\frac{e^{||x_i||cos(\theta_{y_i},i)+0}}{\Sigma_j e^{||x_i||cos(\theta_j,i)+0}}\right). \tag{2.4}$$

Though by using modified softmax loss, we can learn features by means of angular boundary, these features are not still necessarily discriminative. Meanwhile angles can be used as the distance metric, that is normal to join angular margin to learned features to boost

up the discrimination power.
Instead of model a new kind of loss function and assembling a weighted mixture with softmax loss, SphereFace [47] introduced a more natural technique to learn the angular margin. As the above analysis shows that the decision boundaries greatly effect on the feature distribution, thus the SphereFace use this basic concept to operate decision boundaries to launch angular margin.

Let's assume the learned feature $x$, from the class 1, is given and the angle between $x$ and $W_i$ is $\theta_i$, this is known that for the modified softmax, $cos(\theta_1) > cos(\theta_2)$ is required to correctly classify $x$. But what will be happen if we have $cos(m\theta_1) > cos(\theta_2)$, for $m = 2$ adopted to properly classify $x$? In actual fact, it is making the decision supplementary stringent than previous and for class 1 the decision boundary is $cos(m\theta_1) = cos(\theta_2)$, in the same way, if require $cos(m\theta_2) > cos(\theta_1)$ to proper classification of features from the class 2 and the decision boundary for this class is $cos(m\theta_2) = cos(\theta_1)$, as discussed in SphereFace [47]. From angular margin point of view, appropriate classification of $x$ from identity 1 needs $\theta_1 < \frac{\theta_2}{m}$, whereas properly classifying $x$ from identity 2 needs $\theta_2 < \frac{\theta_1}{m}$. Thus, by directly reformulating the equation (2.4) based on this idea, the modified softmax loss can be written as

$$L_{angular} = \frac{1}{N}\Sigma_i\left(-log\frac{e^{||x_i||cos(m\theta_{y_i},i)}}{e^{||x_i||cos(m\theta_{y_i},i)} + \Sigma_j e^{||x_i||cos(\theta_j,i)}}\right). \quad (2.5)$$

In order to get a rid of its restriction and make optimizable for DCNNs, SphereFace, introduced monotonically decreasing angle function. A-Softmax has robust requirements for a precise classification with $m \geq 2$, that makes an angular classification margin between learned features for the different classes, that loss not only to imposes discriminative power of the learned features through angular margin, but also reduces a novel hypersphere interpretation.

The assessment of the decision boundaries is presented in Table 2.2, This is from optimizing the inner product to optimizing the angles from the original softmax to modified softmax, and from modified softmax loss to SphereFace or A-Softmax, that makes the decision boundary more stringent and more separated. The angular margin rises through the larger $m$ and be zero for $m = 1$.

The SGD used to optimize the model, initially with a very large value of $(\lambda)$ that is equal to the strategy adopted for optimizing the original softmax. Then steadily reduce the values

| Loss | Decision Boundary |
|---|---|
| Softmax | $(W_1 - W_2)x = b_2 - b_1$ |
| Modified Softmax | $\|\|x\|\|cos(\theta_1 - cos(\theta_2) = 0$ |
| A-Softmax | For class 1, $\|\|x\|\|cos(m\theta_1 - cos(\theta_2) = 0$<br>For class 2, $\|\|x\|\|cos(\theta_1 - cos(m\theta_2) = 0$ |

Table 2.2 The decision boundaries in binary classes.

during training. A 64-layers ResNet architecture used in SphereFace that offered the angular softmax loss to learn discriminative human face features with angular margin with different integer values of $m > 0$ and achieved the State-of-the-art performance 99.42% on LFW and 95.0% on YTF dataset at $m = 4$. It indicates that A-Softmax is more appropriate for open-set FR. Moreover, the challenging learning job defined in SphereFace make a full use of the superior learning competence of deeper architectures.

### 2.2.2 Large Margin Cosine Loss

The Large Margin Cosine Loss [88] start by reconsidering the original softmax from a cosine viewpoint. The softmax separates the features from different classes via maximizing posterior probability of ground-truth class. Assumed an input feature vector $x_i$ with its label $y_i$, the original softmax loss can be reformulated formulated as

$$L_s = \frac{1}{N}\sum_{i=1}^{N} -log(p_i) \tag{2.6}$$

$$L_s = \frac{1}{N}\sum_{i=1}^{N} -log\frac{e^{f_{y_i}}}{\sum_{j=1}^{C} e^{f}_{j}} \tag{2.7}$$

Where $p_i$ be the posterior probability for $x_i$ being correctly classified. $N$ denotes the number of training samples and $C$ denotes the number of classes. The $f_j$ commonly designated as activation of the fully-connected layer with weight vector $W_j$ and bias $b_j$. If $b_j = 0$, then $f_j$ for the angle $\theta_j$ between $W_j$ and $x$, is given by

$$f_j = {W_j}^T x = ||{W_j}^T||||x||cos(\theta_j) \tag{2.8}$$

For the effective feature learning, norm of $W$ must be necessarily invariable. On this end, CosFace strategy fixed $||W_j|| = 1$ through $L2$ normalization. For testing, normally the FR score of a testing face pair is calculated according to the cosine similarity between the two feature vectors, that suggests the norm of feature vector $x$ is not contributing in scoring function. Therefore, in the training stage, it could be fixed, $||x|| = s$, that modification can be formulated as

$$L_s = \frac{1}{N}\sum_i -log\frac{e^{s\ cos(\theta_{y_i},i)}}{\sum_j e^{s\ cos(\theta_j,i)}} \tag{2.9}$$

That resulting model learns the features that are separable in angular space, this is refer in CosFace as the Normalized Softmax Loss (NSL). Though, the features learned via NSL are not sufficiently discriminative, because NSL just emphasizes accurate classification. To handle this issue, CosFace also present the cosine margin for the classification boundary, that is obviously incorporated into cosine formulation of Softmax. Assuming $\theta_i$ be the angle between learned feature vector and weight vector of class 1 and class 2, the NSL forces $cos(\theta_1) > cos(\theta_2)$ for class 1 and similarly for class 2, so the features from the different classes are correctly classified. For the large margin classifier, it further require $cos(\theta_1) - m > cos(\theta_2)$ and $cos(\theta_2) - m > cos(\theta_1)$ for $m \geq 0$ to control the magnitude of the cosine margin and introduce Large Margin Cosine Loss (LMCL) as

$$L_{LMC} = \frac{1}{N}\sum_i -log\frac{e^{(s\ cos(\theta_{y_i},i)-m)}}{e^{(s\ cos(\theta_{y_i},i)-m)} + \sum_{j\neq y_i} e^{s\ cos(\theta_j,i)}} \tag{2.10}$$

Where $N$ be the training samples, $x_i$ be the $i$ - $th$ feature vector corresponding to ground-truth class of $y_i$, and $W_j$ be the weight of the $j$ -$th$ class, and $\theta_j$ be the angle between weight $W_j$ and samples $x_i$. The architecture used in CoseFace [88] is similar to SphereFace [47], that has 64 convolutional layers and is based on residual units. The scaling parameter $s$ in LMCL is set to 64 empirically, The DCNN models trained with SGD optimization technique, with the batch size of 64 on 8 GPUs, that achieved the performance 99.33% on LFW and 96.1% on YTF dataset. It is clear that the model without margin at $m = 0$ leads to the poorest performance. As $m$ being increased, the performance is improved consistently on LFW and YTF datasets, and get saturated at $m = 0.35$. This demonstrates that by increasing the margin $m$, the discriminative power of the learned features can be significantly improved.

### 2.2.3 Additive Angular Margin Loss

The center loss penalizes distance between deep features and corresponding class center in Euclidean space to achieve intra-class compactness. A sofmax loss in SphereFace assumed the linear transformation in last fully connected layer and can be used as representation of the class centers in an angular space and penalizes the angles between the deep features and their corresponding weights in a multiplicative way. A popular line of on this research is to incorporate the margins in well-established loss functions to maximize face class separability. The Additive Angular Margin Loss [19] is applicable to obtain highly discriminative features for the face recognition. Recall the most widely used classification loss, softmax loss, that is presented as

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N} log\left(\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}\right) \tag{2.11}$$

Where $x_i$, denotes the deep feature of the $i\ th$ sample, belonging to the $y_i\ th$ class. $W_j$ denotes the $j\ th$ column of the weight $W$ and $b_j$ is the bias term. ArcFace use the similar strategy like spherface and CosFace, fix the bias $b_j = 0$ and transform the logit as $W_j^T x_i = ||W_j||||x_i||cos(\theta_j)$, where $\theta_j$ is the angle between the weight $W_j$ and the feature the $x_i$, and fix the individual weight $||W_j|| = 1$ via $L2$ normalization and re-scale the mbedding feature $||x_i||$ to $s$.

$$L_2 = \frac{1}{N}\sum_i -log\left(\frac{e^{(s\ cos(\theta_{y_i}))}}{e^{(s\ cos(\theta_{y_i}))} + \sum_{j\neq y_i} e^{s\ cos(\theta_j)}}\right). \tag{2.12}$$

As embedding features are distributed around each feature center on the hypersphere, ArcFace add additive angular margin $m$ between $x_i$ and $W_{y_i}$.

$$L_3 = \frac{1}{N}\sum_i -log\left(\frac{e^{(s\ cos(\theta_{y_i}+m))}}{e^{(s\ cos(\theta_{y_i}+m))} + \sum_{j\neq y_i} e^{s\ cos(\theta_j)}}\right). \tag{2.13}$$

The combination of all of the margin penalties, ArcFace implement SphereFace, CosFace and ArcFace in a united framework with $m_1$, $m_2$ and $m_3$ as the hyper-parameters via adjusting the decimal random values.

$$L_4 = \frac{1}{N}\sum_i -log\left(\frac{e^{(s\ cos(m_1\theta_{y_i}+m_2)-m_3)}}{e^{(s\ cos(m_1\theta_{y_i}+m_2)-m_3)}+\sum_{j\neq y_i} e^{s\ cos(\theta_j)}}\right). \tag{2.14}$$

A geometrical comparison on the decision boundaries in the binary classification case is illustrated in Fig.2.3. The additive angular margin considered as a better geometric attribute as it is clear that angular margin demonstrations the exact correspondence towards the geodesic distance, that is a representation of the decision boundaries in the case of binary classification. The ArcFace contend to have a constant linear angular margin value throughout the entire interval and CosFace and SphereFace have a nonlinear angular margin. The Additive Angular Margin Loss in case of ArcFace, effectively enhance the discriminative power of the feature embeddings learned through DCNNs for FR task, its comprehensive experiments demonstrate that the method consistently outperforms.

These three approaches based on the modification of softmax play a vital role in the field of computer vision. Considering the remarkable results and achievement of these methods,

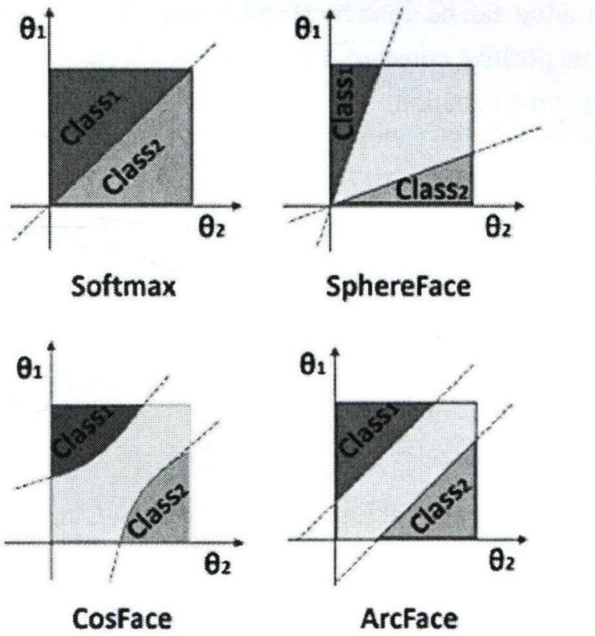


Fig. 2.3 The decision margins of the different types of sofmax available in related work for binary classification case. The the grey areas are the decision margins and dashed line denotes the decision boundary.

we adopt a bit similar strategy to improve the performance of DCNNs, and described in the next chapter.

### 2.2.4 Laplacian Smoothing Gradient Descent

As compared to the data extents, the processors speed is slower in all computer vision applications based on DCNNs. The SDG is a workhorse for optimizing a large scale learning of linear classifiers under convex loss functions [12, 13] and executes a parameter update to each training sample $x_i$ with label $y_i$ and permitted the processors to access shared memory. No doubt, SGD is beneficial for gradient-based optimization procedures circumvent spurious local minima but it decelerates the convergence of conventional gradient descent. To find the better minima, the Laplacian Smoothing Gradient Descent [54] proposed on the basis of theoretical explanation of Hamilton-Jacobi partial differential equations, via pre-multiply the gradient by the inverse of the tri-diagonal circular convolution matrix and is applicable to reduces the optimality gap in SGD. A carefully modeled positive definite matrix introduced in LS-SGD and used to smooth the gradient to reduce noise in SGD, applicable to improve the training of DNNs. The methodology involves multiplication of normal gradient through the inverse form of a matrix generating from discrete Laplacian or high order generalizations of Laplacian. In the result of solving a tri-diagonal linear system with the original gradient, the gradient smoothing can be done by simply pre-multiply the gradient to the inverse of a matrix $A_\sigma$ for some positive constant $\sigma \geq 0$, that tri-diagonal circular convolution matrix is defined in the following equation.

$$A_\sigma = \begin{pmatrix} 2\sigma+1 & -\sigma & 0 & \cdots & 0 & -\sigma \\ -\sigma & 2\sigma+1 & -\sigma & \cdots & 0 & 0 \\ 0 & -\sigma & 2\sigma+1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\sigma & 0 & 0 & \cdots & -\sigma & 2\sigma+1 \end{pmatrix} \tag{2.15}$$

If the discrete one-dimensional Laplacian denoted by $L$ and $I$ be the identity matrix, then $I - \sigma L = A_\sigma$ and a forward finite difference $D_+$ can be written as as

$$D_+ = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \cdots & 0 & -1 \end{pmatrix} \tag{2.16}$$

The backward finite difference $D_- = -D_+^T$ and $D_- D_+ = L$, or we can write

$$A_\sigma = I - \sigma D_- D_+ \tag{2.17}$$

For better convergence in the existence of a very noisy stochastic gradient, the operator $A_\sigma^{-1}$ acts like denoiser. If $f_i(w) = f(w, x_i, y_i)$ be the loss of the given model on training data $\{x_i, y_i\}$, then LS-SGD can be defined as

$$w^{k+1} = w^k - \gamma A_\sigma^{-1} \nabla f_{i_k}(w^k). \tag{2.18}$$

Where $\gamma$ is the step size and $i_k$ is the random sample. The LS-SGD used explicit to relax the implicit strategy to perform layer-wise gradient smoothing that assistances to sidestep sharp minima to reach the global minima. That theatrical modification in SGD affectedly decrease the optimality gap and comfortable to find better minima.

# Chapter 3

# Methodology

Recent studies on loss functions clearly describing that better normalization is helpful for improving the performance of image recognition system. Several methods based on different loss functions have been proposed for this task to obtain discriminative features. An efficient technique for IR is proposed in Chapter 3 of this thesis. This chapter containing the modified form of angular softmax loss function. We explain the modeling of angular sofmax based on the modification on the softmax loss function in section 3.1, the subsection 2.1 containing the modified form of angular softmax and the subsection 3.1.2 offering a additive parameter and a proper adjustment of the margin for the loss function, an automatic adjustment of the seedling element as the result of angular marginal seed is offered in a particular way for the angular softmax to learn angularly discriminative features. Significant impact of our proposed method is discussed in subsection 3.1.3.

## 3.1 Angular Softmax

As concluded in the previous chapter, that the angular form of the softmax loss function is more suitable to use for recognition system. The literature and the remarks on the literature presented in section 2.2 has proven that the margin for the angular softmax is flexible for image recognation tasks. A-Softmax [47] known as SphereFace, is used to compress the intra-class angular distribution and to increase the inter-class margin. For effective and accurate discriminating performance, a number of modifications set to boost up the discriminative power of the softmax loss but it is not good for the intra-class variation. On this concern researcher proposed a scale parameter [19, 47, 88] generating gradients to the separated samples for intra-class variance. All these approaches are based on the similar strategy for maximizing inter-class variance and minimizing intra-class variance. For the marginal loss function, the concern is to adjust the additive marginal values appropriately by the use of integer values and decimal margin values. Proposing a better solution on this matter, we offer an additive parameter relative to multiplicative angular margin for deep image recognition. The proposed additive parameter is adjusted in softmax as a seedling element, as it is just growing in the result of margin seed.

### 3.1.1 Modification in Sofmax

To increase the learn capability of DCNNs through the angular margin approaches for softmax loss, A-Softmax [47] achieved remarkable progress and has strong geometric explanation towards the discriminative angular distance metric and can be taken as compiler for discriminative feature on a hypersphere manifold that essentially matches the prior that images lie on a manifold. In the case of binary-class, the probabilities obtained through the softmax loss for learned feature vector $x$, weights $w_i$ and base $b_i$ of the last fully connected layer corresponding to class $i$, are

$$Prob_1 = \frac{e^{{W_1}^T x + b_1}}{e^{{W_1}^T x + b_1} + e^{{W_2}^T x + b_2}} \tag{3.1}$$

and

$$Prob_2 = \frac{e^{{W_2}^T x + b_2}}{e^{{W_1}^T x + b_1} + e^{{W_2}^T x + b_2}} \tag{3.2}$$

The predicted label will be allocated to class 1 if $Prob_1 > Prob_2$ and class 2 if $Prob_1 < Prob_2$. This is clear that ${W_1}^T x + b_1$ and ${W_2}^T x + b_2$ determine the classification result via comparing $Prob_1$ and $Prob_2$, where $(W_1 - W_2)x + b_1 - b_2 = 0$ is the decision boundary. If $\theta_i$ be the angle between $W_i$ and $x$, then we can rewrite ${W_i}^T x + b_i$ as $||{W_i}^T||\ ||x||cos(\theta_i) + b_i$. If normalize the weights $W_i = 1$ and base zero, that is $b_i = 0$, then the final result depends only on the angles $\theta_1$ and $\theta_2$. Even though this analysis is assembled on binary-calss case, that is trivial to generalize the analysis for multi-class case. For the period of training, the softmax loos after modification in weight and base, encourages the features from *i-th* class to have minor angle $\theta_i$ as compared to others.

The softmax take an un-normalized vector function, and stabilizes it to a probability distribution. The softmax loss is naturally good in optimization of the inter-class difference, that's why classification loss functions for deep structures are typically constructed by using softmax loss. As discussed in the previous chapter, that the original softmax loss can be written in the shape of angular form, that can be reformulate in cosine form as

$$L_{S_\theta} = -\frac{1}{N}\sum_i log\left(\frac{e^{\left(||W_{y_i}{}^T||||x_i||cos(\theta_{y_i},i)+b_{y_i}\right)}}{\Sigma_j e^{\left(||W_j{}^T||||x_i||cos(\theta_j,i)+b_j\right)}}\right), \quad \forall\ 0 \leq \theta_j, i \leq \pi. \tag{3.3}$$

As per the spherface strategy, By fixing the bias $b_j = 0$ and normalize the weight $W_j = 1$, the softmax can be modified as

$$L_{S_\theta} = -\frac{1}{N}\sum_i log\left(\frac{e^{\left(||x_i||cos(\theta_{y_i},i)\right)}}{\Sigma_j e^{\left(||x_i||cos(\theta_j,i)\right)}}\right), \quad \forall\ 0 \leq \theta_j, i \leq \pi. \tag{3.4}$$

That modification based on normalization of features and weights makes the predictions only be subject to the angle between the feature vector and the weight. One can consider angle as a distance metric, that is accepted to integrate angular margin for learned features to enhance the discrimination power.

Angular Margin Softmax Loss is a hypersphere embedding method introduced for deep structure to learn discriminative face features with angular margin, where the non-monotonicity of the cosine function has been adjusted by a piece-wise function. By A-Softmax loss, the learned features generate a discriminative angular distance metric that can be defined equitant to geodesic-distance on a hypersphere manifold and introduce a monotonically decreasing function (depending on an angle between weight and features) $\psi_{1(\theta_{y_i})}$ for the modified sofmax loss and has been presented as

$$L_{\psi_{1(\theta_{y_i})}} = -\frac{1}{N}\sum_i log\left(\frac{e^{\left(||x_i||\psi_1(\theta_{y_i})\right)}}{e^{\left(||x_i||\psi_1(\theta_{y_i})\right)}+\Sigma_{j\neq y_i} e^{\left(||x_i||\psi_1(\theta_j)\right)}}\right) \tag{3.5}$$

The piecewise monotonically decreasing function $\psi_{1(\theta_{y_i})} = (-1)^k cos(m\theta) - 2k$ for $\frac{k\pi}{m} \leq \theta_{y_i} \leq \frac{(k+1)\pi}{m}$ and the integer value $m \geq 1$ used for the size of angular margin. The angular margin becomes larger for larger value of $m$ and the constrained region of the manifold come to be lesser. For the marginal loss function, the concern is to adjust the additive marginal values appropriately by the use of integer values and decimal margin values. Proposing a better solution on this matter, we offer an additive parameter relative to multiplicative angular margin for deep face recognition. The proposed additive parameter is adjusted in softmax as a seedling element, as it is just growing in the result of margin seed.

### 3.1.2 Additive Parameter

In this section, we are presenting a simple approach for deep feature learning, based on additive and multiplicative angular margin. Recalling the angular softmax loss offered by the SphereFace, we reformulate the original softmax loss by considering large-margin softmax as

$$L_{S_{m\theta}} = \frac{1}{N}\sum_{i} -log\left(\frac{e^{(||W||||x_i||cos(m\theta)+b)}}{e^{(||W||||x_i||cos(m\theta)+b)} + \sum_j e^{(||W||||x_i||cos(\theta_j)+b)}}\right). \tag{3.6}$$

The integer $m$ represents the angular margin, the $\theta$ is the angle between the weight of last fully connected-layer $W$ and deep features vector $x$. A piecewise monotonically decreasing function$\psi_2$ defined to remove nonmonotonicity of the function, that can be written as

$$\psi_2(\theta) = \frac{1}{1+\phi}\Big((-1)^k cos(m\theta) - 2k + \phi cos(\theta)\Big) \tag{3.7}$$

The $\theta \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}]$ and $\phi$ is dynamic hyper-parameter used to control weight that further normalized. To build a cosine layer, we adopt the same implementation used by spherface for both feature and weight normalization. Then the simple loss can be formulating as

$$L_{\psi_2(\theta)} = \frac{1}{N}\sum_{i} -log\left(\frac{e^{(||x_i||\psi_2(\theta)}}{e^{(||x_i||\psi_2(\theta))} + \sum_j e^{(||x_i||cos(\theta_j))}}\right). \tag{3.8}$$

The parameter $m$ can be used for learning an angular-margin between different identities. We offered an additive parameter $\varepsilon$, an automatic adjustment of the seedling element as the result of angular marginal seed is offered for the angular softmax to learn angularly discriminative features, that additive parameter in the shape of seedling element $\varepsilon$ is defined as

$$\varepsilon = \frac{m}{2(m+1)}, m \geq 1. \tag{3.9}$$

It is observed that for all positive integer $m$, the additive parameter $\varepsilon \in [0.25, 0.5)$, as illustrated in Table 3.1.

| Margin ($m$) | Additive parameter ($\varepsilon$) |
|---|---|
| 1 | 0.25 |
| 2 | 0.33 |
| 3 | 0.375 |
| 4 | 0.40 |
| 5 | 0.416 |

Table 3.1 The values of additive parameter $\varepsilon$ corresponding to the positive integer $m$.

To develop additive parameter $\varepsilon$ as a seedling element rising in the result of margin seed $m$, we reformulate the monotonically decreasing function $\psi_2(\theta)$ to make more separated decision boundary, a monotonically decreasing function $\psi_3$ is defined as

$$\psi_3(\theta) = \psi_2(\theta) - \varepsilon \tag{3.10}$$

That modification is more simple to take gradient during the forward and backward propagation and trivial to optimize the modified softmax loss using stochastic gradient descent. Finally, a softmax loss based on a seedling parameter $\varepsilon$ growing in the result of multiplicative angular marginal seed $m$ is defined as

$$L_{\psi_3(\theta)} = \frac{1}{N}\sum_i -log\left(\frac{e^{(||x_i||\psi_3(\theta)}}{e^{(||x_i||\psi_3(\theta))} + \sum_j e^{(||x_i||cos(\theta_j))}}\right). \tag{3.11}$$

To implement our proposed methodology, we can use any case of the additive parameter to modify the loss function, For example, a case of modified softmax is defined by setting angular margin $m = 3$, that generate a automatic adjusted additive parameter corresponding to the angular margin. So, if angular margin is $m = 3$, it produce the fixed value of additive margin $\varepsilon = 0.375$ to generate modified monotonically decreasing parameter $\psi_{(\varepsilon=0.375)} = [(-1)^{\lfloor k \rfloor} cos(\theta) - 2k] - 0.375$. That loss function can be written as

$$L_{\psi_\varepsilon(\theta,3)} = -\frac{1}{N}\Sigma_i \log \frac{e^{||x_i||\psi_\varepsilon(\theta)}}{e^{||x_i||\psi_\varepsilon(\theta)} + \Sigma_j e^{||x_i||cos\theta_j}} \tag{3.12}$$

The proposed approach is theoretically slight similar to the original sofmax loss and computationally it is more appealing and beneficial because of its combined effect of additive and angular margin and is easily adjustable with both optimization techniques SGD and

LS-SGD.

If we compare the decision margin of our approach to other margin based methods, it can be analysis that our methodology is unique. For example, suppose $w_1$ and $w_2$ are the weights for the binary-class $d_1$ and $d_2$ respectively and $m_1$ is considered as multiplicative angular margin and $m_2$ as additive angular margin. The decision boundary in case of softmax loss is $||w_2||cos(\theta_2) = ||w_1||cos(\theta_1)$, the boundary depends on the magnitudes of cosine of angles and weights. A-softmax offer a multiplicative angular margin $m_1$ and decision boundary $d_1$ in this case is $cos(m_1\theta_1) \geq cos(\theta_2)$ and for $d_2$ is $cos(m_1\theta_2) \geq cos(\theta_1)$. In Large margin cosine loss the decision boundary is define with the cosine additive margin $m_2$, for $d_1$ is $cos(\theta_1) - m_2 \geq cos(\theta_2)$ and for $d_2$ is $cos(\theta_2) - m_2 \geq cos(\theta_1)$. Our proposed approach boosts up the power of angular margin and cosine margin via combined effect multiplicative angular margin $m_1$ and cosine additive margin $m_2$. Such that the decision boundary is $cos(m_1\theta_1) - m_2 \geq cos(\theta_2)$ for $d_1$ and $cos(m_1\theta_2) - m_2 \geq cos(\theta_1)$. Moreover, our case to adjust the additive margin value $m_2$ is based on the selection of multiplicative angular margin $m_1$, we offer a seedling element as an additive parameter to control the additive margin.

| Methods | Loss | Margin |
|---|---|---|
| SphereFace | $\frac{1}{N}\Sigma_i\left(-log\frac{e^{\|\|x_i\|\|cos(m\theta_{y_i},i)}}{e^{\|\|x_i\|\|cos(m\theta_{y_i},i)}+\Sigma_j e^{\|\|x_i\|\|cos(\theta_j,i)}}\right)$ | $m \in \{1,2,3,4\}$ |
| CosFace | $\frac{1}{N}\Sigma_i\left(-log\frac{e^{s\left(cos(\theta_{y_i},i)-m\right)}}{e^{s\left(cos(\theta_{y_i},i)-m\right)}+\Sigma_j e^{scos(\theta_j,i)}}\right)$ | $0 \leq m \leq 0.45$ |
| ArcFace | $\frac{1}{N}\Sigma_i\left(-log\frac{e^{s\left(cos(m_1\theta_{y_i}+m_2,i)-m_3\right)}}{e^{s\left(cos(m_1\theta_{y_i}-m_2,i)-m_3\right)}+\Sigma_j e^{scos(\theta_j,i)}}\right)$ | $0.9 \leq m_1 \leq 1.35$,<br>$0 \leq m_2 \leq 0.5$,<br>$0 \leq m_3 \leq 0.35$ |
| Proposed Approach | $\frac{1}{N}\Sigma_i\left(-log\frac{e^{\|\|x_i\|\|cos(m\theta_{y_i},i)-\varepsilon}}{e^{\|\|x_i\|\|cos(m\theta_{y_i},i)-\varepsilon}+\Sigma_j e^{\|\|x_i\|\|cos(\theta_j,i)}}\right)$ | $m \in \{1,2,3,4,5\}$<br>$\varepsilon = \frac{m}{2(m+1)}$,<br>$0.25 \leq \varepsilon < 0.5$ |

Table 3.2 Comparison of the angular softmax loss with the corresponding margin values. It is clear that, in our proposed approach, there is no need to select the random values of additive margin as used in ArcFace and CosFace.

The Table 3.2 represents a simple comparison of the angular softmax loss with the corresponding margin values of the different methods adopted for the FR. Mathematical representation of additive margin shows that the additive margin in the shape of additive parameter $\varepsilon$ is dependent on margin $m$.

### 3.1.3 Significant Impact

The development of loss function based on margin is categorized into three types; the first one is multiplicative angular margin used in SphereFace [47], where the margin is merged into the loss in a multiplicative way. Second is additive angular margin used in ArcFace [18] incorporated to loss in an additive way, and the third is additive cosine margin CosFace [88] that is introduced to maximize the margin in angular space.

In additive and angular margin strategy SphereFace use integer values as margin element, CosFace and ArcFace use decimal numbers as margin. In A-softmax, the margin value $m$ is multiplied to angle $\theta$, the angular margin value is combined into the loss via multiplicative way. Our approach to manage the margin is significantly different than the A-softmax strategy, because the margin is enforced by subtracting margin value in an additive way and the additive margin is adjusted automatically. For the marginal loss function, the issue is to adjust the additive marginal values appropriately by the use of integer values and decimal margin values. Our proposed parameter is offering an appropriate adjustment of decimal values in additive margin, that refinement is proportional to the integer value of angular margin. A seedling element as an additive parameter is growing automatically as a response of every altering integer value of angular margin. For example, if $m = 3$ is an angular margin seed then the automatically generating seedling element is $\varepsilon = 0.375$, substitute to improve the performance of piecewise decreasing effect of modified softmax loss for optimizing the similarity. In mathematical model of modified softmax loss represented, the capacity of the additive parameter is automatically controlled by angular margin, where the angular margin forced the additive margin to remain in the interval $[0.25, 0.49]$.

Our proposed approach is significantly different then SphereFace [47], ArcFace [18] and CosFace [88] as the loss is based on additive margin and angular margin, that is employed by subtracting the parameter $\varepsilon$ from the cos function. A better action is prerequisite after the inner product of weight vector $W$ and feature vectors $x_i$, which is computationally expensive. The generating *cos* function as a result of inner product of weight and feature vector is failed to learn the similarity feature because of its sinusoidal appearance with two maximum values over the interval 0 to $\pi$. Conceptually the cosine margin is less good then angular margin, but computationally it is more appealing to achieve results using the monotonically decreasing parameter to maximize the value of similarity for same classes and minimize the value for different classes. The additive parameter is significantly effective because of its piece-wise monotonically decreasing nature over the interval.

The adjustment of additive parameter is simple to take gradient during the forward and backward propagation and is trivial to optimize the modified softmax loss using Stochastic Gradient Descent, as the SGD optimization strategy is a pillar for solving a large scale learning of linear classifiers and perfectly adjusted with the marginal softmax, However, when training data is too large the convergence of large margin is more difficult than softmax with small margin. The additive parameter with SGD has the capacity to reduce the resistance during training a big image data by offering the small additive value $\varepsilon$ growing in the result of margin seed$m$. Because of combined effect of additive and angular margin, our approach is significantly unique, more efficient and stable to work for deep neural network.

We adopt two different computer vision application to evaluate the proposed strategy and experimentally prove that the additive parameter approach for angular softmax is more suitable for recognition systems. In the next chapter, we briefly discussed about the both applications and analysis.

# Chapter 4

# Applications

In this chapter, we discussed the implementation of our proposed method on two main computer vision application. We expose the experimental details and results on both applications also, a analysis of our results with the other modern techniques used for both tasks are included in this chapter. We implement our proposed methodology on FR and HDR systems, the complete process for FR is described in section 4.1 with implementation details, experiments, results and comparison of our proposed approach with the other methods for face recognition. The second application is addressed in section 4.2. We adopt HDR system to evaluate the performance of our method, this section offering implementation details for HDR, containing results and analysis.

## 4.1 Face Recognition

As because of its natural characteristics, facial recognition has become a prominent biometric method used for authentication of identity and has been extensively cast-off in numerous technological zones. We use this application to evaluate the performance of proposed modified angular softmax loss waith the additive parameter.

For a fair comparison of our results with the modern existing methods we use the similar publicly available web-collected dataset CASIA-WebFace for training and popular public available face datasets LFW [31] and YTF [92] for testing purpose, the details about these benchmarks are summarized in the Table 4.1. The dataset CASIA-Webface [41] is used to train the framework over convolutional layers supervised by the additive parameter enclosed in softmax loss. The trained weights generated after training process are evaluated over

| Datasets | Number of Subjects | Number of Images |
|---|---|---|
| CASIA [41] | 10,575 | 494,414 |
| LFW [31] | 5,749 | 13,233 |
| YTF [92] | 1,595 | 3,425 videos |

Table 4.1 A summary of publicly available training dataset CASIA-WebFace used for training and LFW and YTF for testing.

the test set containing face pairs of LFW and YTF. The complete process of training and testing framework for performance evaluation of softmax loss functions using DCNNs is summarized in the Fig.4.1.

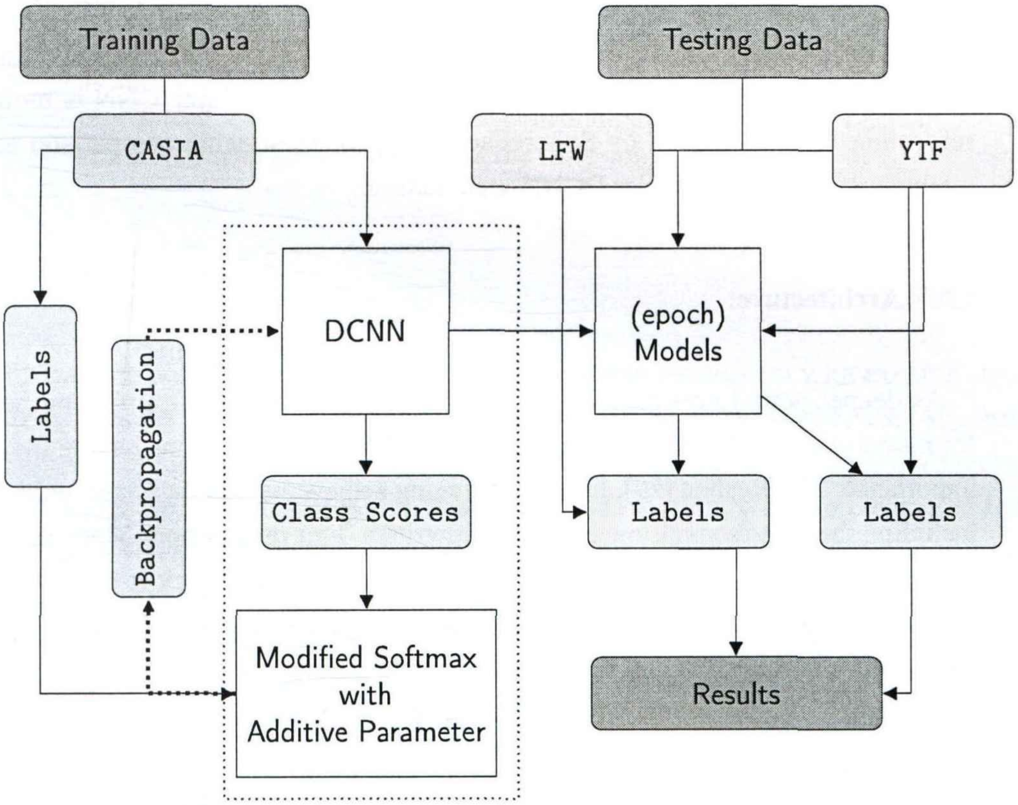


Fig. 4.1 The DCNNs framework for performance evaluation of modified softmax loss with the additive parameter functions, the epoch represents the transfer of the trained model molded from the CASIA for testing over testing faces from LFW and YTF.

### 4.1.1 Implementation Details

This section is dedicated to designate the experimental setup used for the performance evaluation of softmax with additive parameter functions for FR system.

The implementation particulars are discussed in the following four steps.

(a). **Experimental Settings:**

A normally MTCNN [97] is used for face detection and alignment in unconstrained environment, we espoused the same strategy for training data set as well as for testing images and aligned the faces permitting the detected landmarks. To make similarity transformation, we take both eyes, nose and both corners of mouth as reference points and then each RGB images is normalized by subtracting 128 and then dividing with 128 by adopting aligned images of size $112 \times 96$. A $36-layer$ CNN is molded by reforming ResNet offered by SphereFace, used for reasonable comparison as other methods practice a bit similar DCNN architecture.

(b). **CNN Architecture:**

As deeper neural networks are considered more difficult for training purpose and for numerous visual recognition tasks, the depth of representations having the central importance. The ResNet [29], have encouraging achievement in many recognition tasks including the face recognition and the handwritten digit recognition. Motivated by this offered strategy, we adopt a bit similar residual learning framework to ease the training of neural networks via modifying the different layers in the architecture. We adjust the network architecture for FR system by setting the convolutional units containing multiple convolution layers with output channels via fine-tuned, stable and learnable size of kernel, the detailed structure is exposed in the Table 4.2. The first columns denote the layer name (blocks of residual units) and the second columns represent the output size and filter size of the model. The block in the second column shows a series of convolutions with the filter size of the convolution and the number of filters used. The convolution layers are symbolized by a multiplier in the second column and the layer's entries in the column are arranged in order.

| Layer | CNN |
|---|---|
| Unit1 | [64 filters of size (3 × 3)] × 1, with stride 2<br>$\left[\begin{pmatrix}3\times 3, & 64\\ 3\times 3, & 64\end{pmatrix}\times 2\right]$ |
| Unit2 | [128 filters of size (3 × 3)] × 1, with stride 2<br>$\left[\begin{pmatrix}3\times 3, & 128\\ 3\times 3, & 128\end{pmatrix}\times 4\right]$ |
| Unit3 | [256 filters of size (3 × 3)] × 1, with stride 2<br>$\left[\begin{pmatrix}3\times 3, & 256\\ 3\times 3, & 256\end{pmatrix}\times 8\right]$ |
| Unit4 | [512 filters of size (3 × 3)] × 1, with stride 2<br>$\left[\begin{pmatrix}3\times 3, & 512\\ 3\times 3, & 512\end{pmatrix}\times 2\right]$ |
| FC | 512 |

Table 4.2 CNN architectures with different convolutional layers. Each convolution unit contain multiple convolution layers and residual units as summarized in second column. where the brackets, for example $[3\times 3, 128]\times 4$ denotes 4 cascaded convolution layers with 128 filters of size 3 × 3, with stride 2 and FC is the fully connected layer.

(c). **Training:**

In training portion, to attain results for a reasonable assessment with existing state-of-the-art results, we use publicly available training dataset CASIA-WebFace [41], that has 494,414 images for human faces of 10,575 identities available with noisy labels. As the margin $m$ is the hyper-parameter in our loss function, we train our model for the different values of hyper-parameter $m$. The additive parameter with different values of $\varepsilon$ is adjusted in function $\psi_3(\theta, m)$ to update the loss function for every training, modified softmax loss and fully connect layer (FC512) is used at the end of DCNN architecture. The FC512 is transforms the input to a vector of dimension 512. For forward and backward propagation, we adopt A-softmax strategy by substituting multi-angle formula instead of direct implementation on marginal cosine expressions. The CNN model is trained with SGD, the weight parameter is fixed to $5e^{-4}$ for 256 batch size and 0.1 is fixed as initial learning rate and is divided by 10 at the 10K, 18K and 22K iterations and run the training to produce 24 epochs. We repeat the training five times taking margin m=1,2,3,4,5 respectively.

(d). **Testing:**

For testing purpose, we use two famous datasets contains faces with large differences in pose, expression and illuminations. LFW dataset [31] consist on 13,233 images of 5,749 identities and YTF dataset [92] is a data of 1,595 different identities from 3,424 videos. The 24 epoch devolved during every training experiment are tested on LFW and YTF using 36 layer CNN for different values of margin, we achieve the stat-of-the-art performance on both benchmarks. In the testing phase, the testing data is fed into additive parameter softmax to extract deep face features that will use to compute the similarity to complete face recognition. The face image is obtained by concatenating its horizontally flipped features with original face features and the thresholder and classifier are hired for respective goal.

### 4.1.2 Results and Analysis

After running the model five times for training on CASIA-WebFace, we get 24 epochs in each trail at different margin values, where $m \in \{1,2,3,4,5\}$ using the monotonically decreasing function $\psi_3$ with the additive parameter $\varepsilon \in \{0.250, 0.333, 0.375, 0.400, 0.416\}$. A successful training on CASIA-WebFace exposing that the fine-tuning on softmax with the additive parameter is significantly attractive, as in all the trails the training accuracy is better with the minimum data loss. We perform the testing on datasets LFW and YTF to perceive the effect of different values of margin $m$. The accuracy on LFW is plotted in Fig. 4.2 and summarized in Table 4.3, while the testing performance of face recognition on YTF is presented in Fig. 4.3 and summarized in Table 4.4. A bar-chart for better understanding is portrayed in Fig. 4.4.

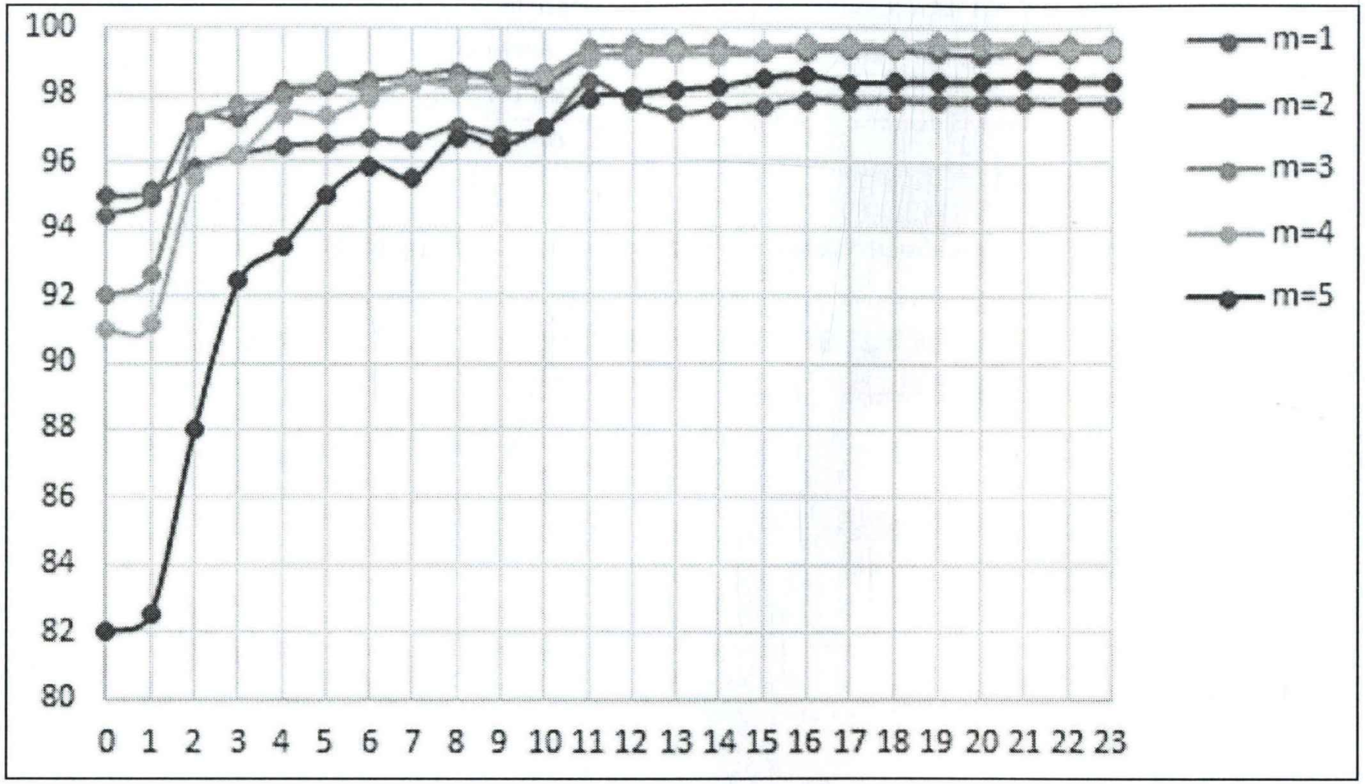


Fig. 4.2 Evaluation results (on *y axis*) on LFW dataset corresponding to the 24 developed models (on *x axis*).

| Margin (*m*) | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Additive Parameter ($\varepsilon$) | 0.25 | 0.33 | 0.375 | 0.40 | 0.416 |
| Accuracy | 97.83% | 99.35% | 99.58% | 99.50% | 98.57% |

Table 4.3 The accuracy on LFW using $\varepsilon$ with the different margin values.

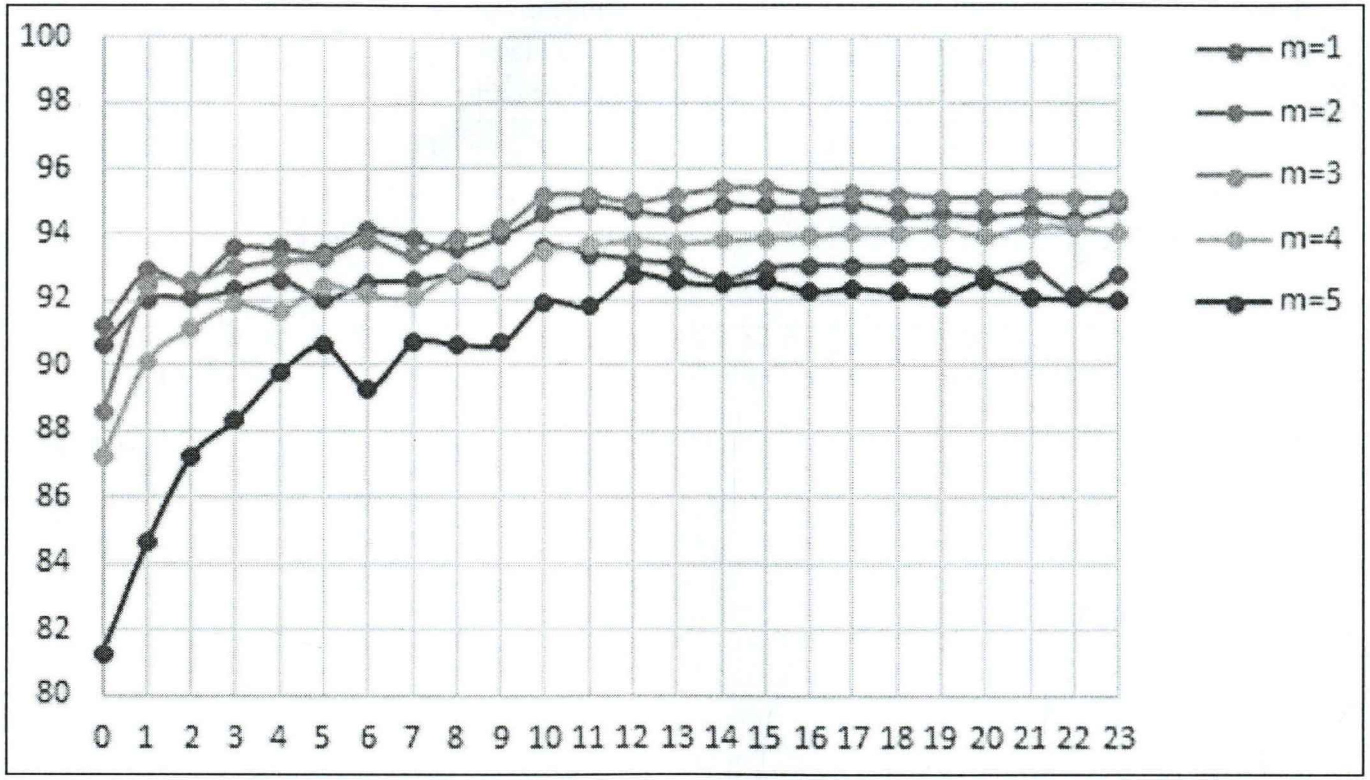


Fig. 4.3 Evaluation results (on $y$ *axis*) of FR on YTF dataset corresponding to the 24 developed models (on $x$ *axis*).

| Margin ($m$) | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Additive Parameter ($\varepsilon$) | 0.25 | 0.33 | 0.375 | 0.40 | 0.416 |
| Accuracy | 93.58% | 94.90% | 95.38% | 94.22% | 92.62% |

Table 4.4 The accuracy on datasets YTF using $\varepsilon$ with the different margin values.

The bar chart in Fig. 4.4 represents the accuracy of the proposed approach at the different margin values with the additive parameter. The FR performance on LFW is significantly improve from 97.83% to 99.58%and 92.62% to 95.38% on YTF. On the view of these results, it is the matter-of-fact that the proposed approach towards deep face features learning has state of the arts accuracy on margin $m = 3$ with additive parameter $\varepsilon = 0.375$.

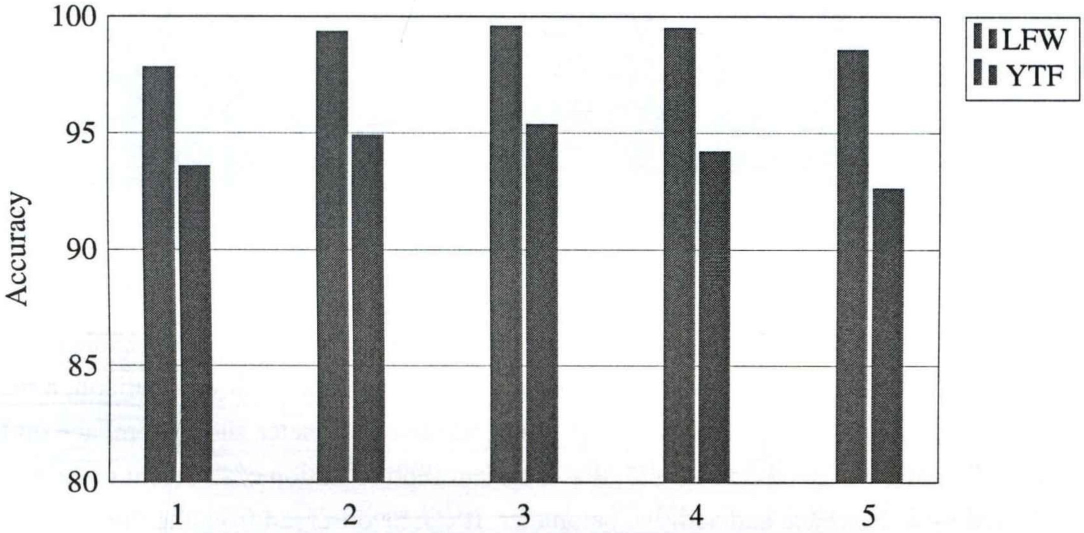


Fig. 4.4 Accuracy on LFW and YTF datasets by means of additive parameter with different margin values $m \in \{1,2,3,4,5\}$.

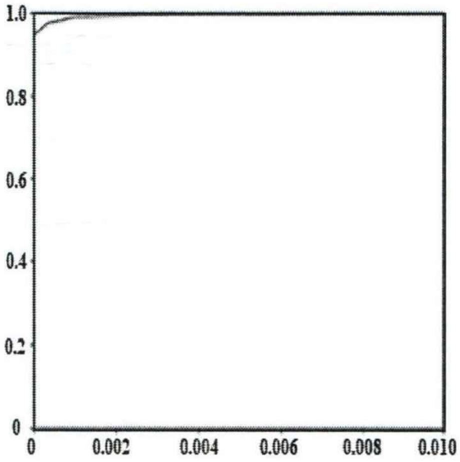


Fig. 4.5 ROC plot of the FPR on $x-axis$ versus the TPR on $y-axis$.

The probability of a binary outcome is portrayed in Fig. 4.5 by receiver operating characteristic (ROC) curve through the plot of the false positive rate (FPR) on $x - axis$ versus the true positive rate (TPR) on $y - axis$ for a number of different candidate threshold values from 0.0 to 1.0. The above analysis on the bases of our results indicating that while $m$ becomes larger, the accuracy also becomes better, which shows that stronger discrimination power get from increasing angular margin, but the accuracy is declined at $m = 5$ showing that the largest margin is damaging the stability of model. It is observed that the performance accuracy of FR on LFW is 99.58% and on YTF is 95.38% corresponding to the margin $m = 3$ and additive parameter $\varepsilon = 0.375$. That is the best evaluation performance using additive parameter trained on CASIAWebFace and is significantly better than several other techniques trained on the same dataset.

The most widely used benchmarks are LFW and used for unconstrained face verification on images and videos. From the last few years a number of loss functions professionally used on these datasets for evaluating the performance. SpherFace have remarkable results on LFW and YTF using different margin values from 1 to 4. For a fair comparison, a margin wise comparison of our proposed sotmax using additive parameter and SphereFace on LFW and YTF dataset is given in Fig. 4.6, that is a clear representation of different margin values adopted by SphereFace and additive parameter. It can be observed from the Fig. 4.6, that the performance on both LFW and YTF is better at $m = 3$ using the additive parameter. The best results on LFW is 99.58% with the additive parameter at $m = 3$ and the 99.42% at $m = 4$ by SpherFace.

In our model trained on CASIA webFace, we use the same benchmarks LFW and YTF for reasonable comparison on performance evaluation. As reported in Table 4.5, softmax based on additive parameter consistently performs better than the current state-of-the-art approaches on the same datasets. Which shows that an additive parameter along with the angular margin can particularly improve the discriminative power of deeply learned features, indicating the effectiveness of approach. The ResNet-64 architecture is used in CosFace and A-softmax to train CASIAWebFace dataset and for testing of the models both methods are evaluated on LFW and YTF datasets. As our network architecture is similar to the architecture of CosFace and A-softmax and also we use the same benchmarks, that's why we compare our results with the results of CosFace and A-softmax.

The Fig. 4.7 is presented for better understanding of the testing results on LFW and YTF via additive parameter, A-softmax and CosFace. Compared with models trained on big

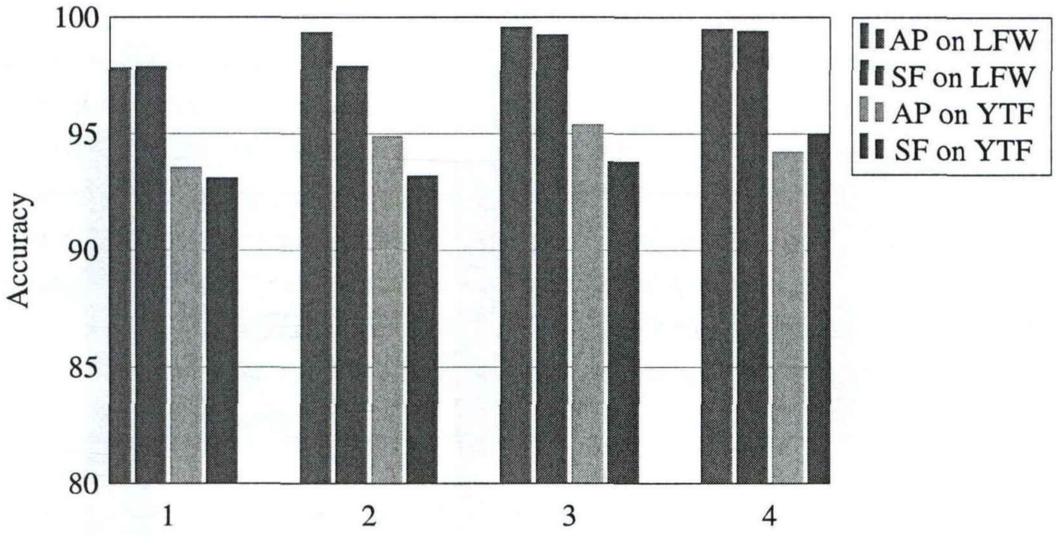


Fig. 4.6 Margin wise comparison of Additive Parameter (AP) and SphereFace (SF) on LFW and YTF dataset. The angular margin values are taken from the positive integers $m \in \{1, 2, 3, 4\}$, used in both methods to improve the performance of DCNN.

| Methods | LFW (%) | YTF (%) |
|---|---|---|
| DeepFace [80] | 97.35 | 91.40 |
| Deep FR [56] | 98.95 | 97.30 |
| FaceNet [70] | 99.65 | 95.10 |
| DeepID2+ [77] | 99.47 | 93.20 |
| Softmax + Center Face [91] | 99.05 | 94.40 |
| Triplet Loss [70] | 98.70 | 93.40 |
| CosFace [88] | 99.33 | 96.10 |
| SphereFace [47] | 99.42 | 95.00 |
| Proposed approach [58] | 99.58 | 95.38 |

Table 4.5 A summary of testing accuracy (%) on LFW and YTF dataset with the latest state-of the-art techniques.

datasets, our approach is competitive and outstripping enactment in the most of the existing results as listed in Table 4.5.

Its observed that the softmax with additive parameter on LFW record the accuracy 99.58% at $m = 3$ that is better than 99.42% at $m = 4$ via A-sotmax and 99.33% at $m = 0.35$ via Coseface. Testing accuracy on LFW by our proposed model is better than the testing accuracy of A-sotmax and CosFace. The recorded values on YTF via additive parameter

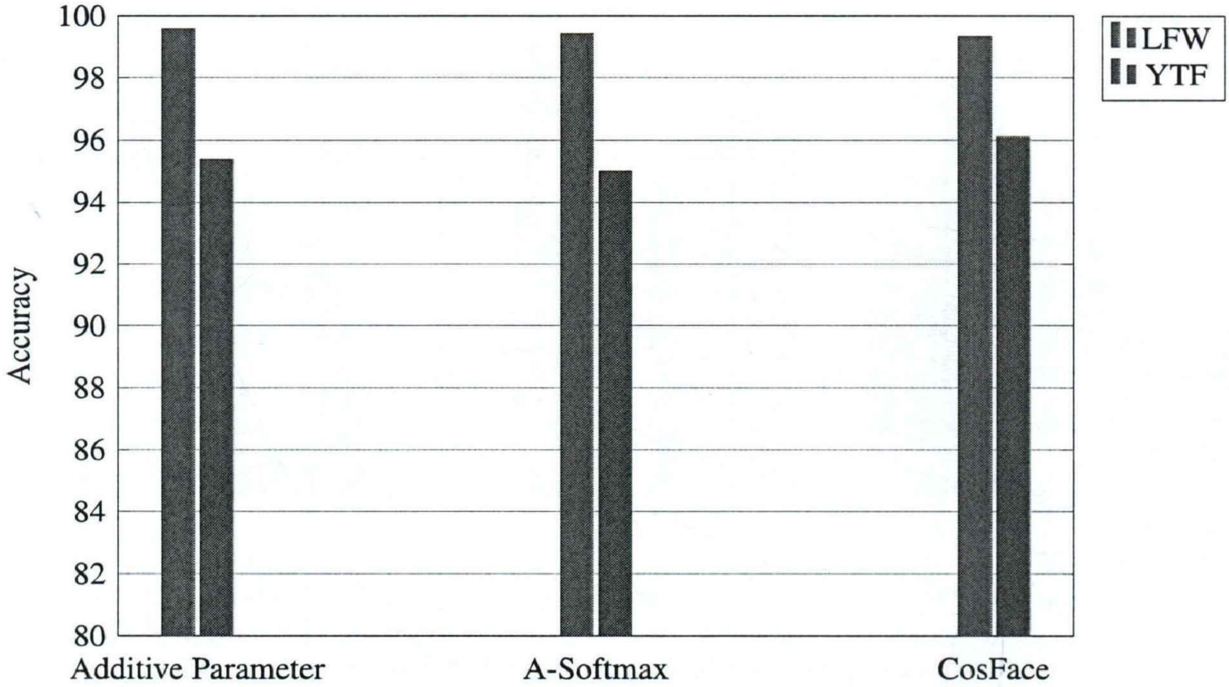


Fig. 4.7 The testing results on LFW and YTF using additive parameter, A-softmax and CosFace.

at $m = 3$ is 95.38%, which is also better than 95.0% at $m = 4$ via SphereFace but smaller than the 95.61% at $m = 0.35$ via CosFace. The average existing FR systems achieve high performance on the big training datasets. While focusing on the stability of the architecture and concerning to improve the margin strategy, the SpherFace, CosFace and our approach have the significant enhancement on the trained models on publicly available dataset CAISA-WebFace, having noisy labels.

## 4.2 Handwritten Digit Recognition

Handwritten Digit Recognition (HDR), is the ability of a machine to interpret and receive intelligible handwritten digits input from other sources such as photographs, paper documents, touch-screens and other modern devices. A HDR system handles configuring, formatting, executes correct segmentation of digits, and finds the most appropriate digit. The digit recognition task initiate with the dataset into training stage, that containing random projection and extraction of the features with different architectures and loss functions used to extract discriminative deep feature. For the handwritten digit recognition, we fixed the angular margin to introduce a unit margin softmax loss. The improved alternative form of softmax is trainable, easy to optimize and stable for usage along with Stochastic Gradient Descent and Laplacian Smoothing Stochastic Gradient Descent. For evaluation of the modified loss, we use a vision application for classification task to recognize hand-written digits (from 0 to 9) using famous benchmark MNIST [38], containing 60,000 training images and 10,000 test images and is well-known for pattern recognition and learning approaches. Few samples from the dataset MNIST are shown in Fig. 4.8 with different patterns. For the handwritten digit recognition process, the Fig. 4.9 represents a deep recognition structure for performance evaluation of loss function.

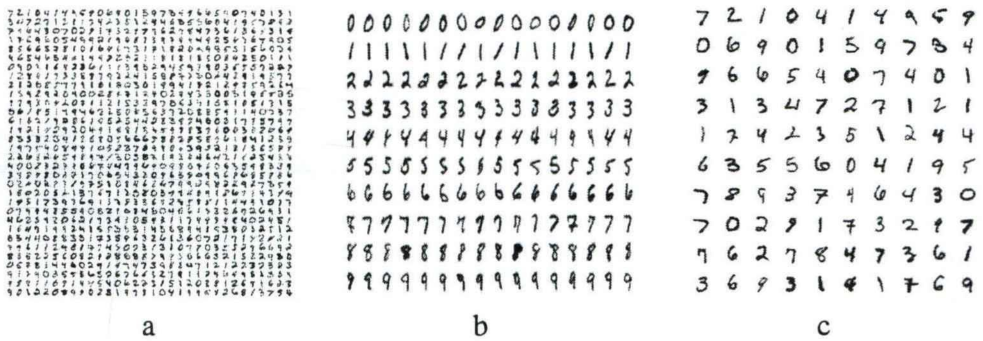


Fig. 4.8 Samples of handwritten digits from (training and testing patterns) the MNIST Dataset.

### 4.2.1 Implementation Details

In recognition task initiate with the dataset into training stage, that containing random projection and extraction of the features with different architectures and loss functions used to extract discriminative deep feature. For optimization task to train the network with loss function, typically used Stochastic Gradient Descent (SGD) [12] a gradient based optimization processes circumvent spurious local minima, but it slows down the convergence of regular gradient descent [69, 54]. Numerous stimulating variance reduction techniques [17, 33] have been proposed for strongly convex functions to recover the linear convergence rate, but not appropriate to train DCNNs. Laplacian Smoothing Stochastic Gradient Descent (LS-SGD) [54] is beneficial to reduce noise in SGD and appropriate to use for training of deep neural network.

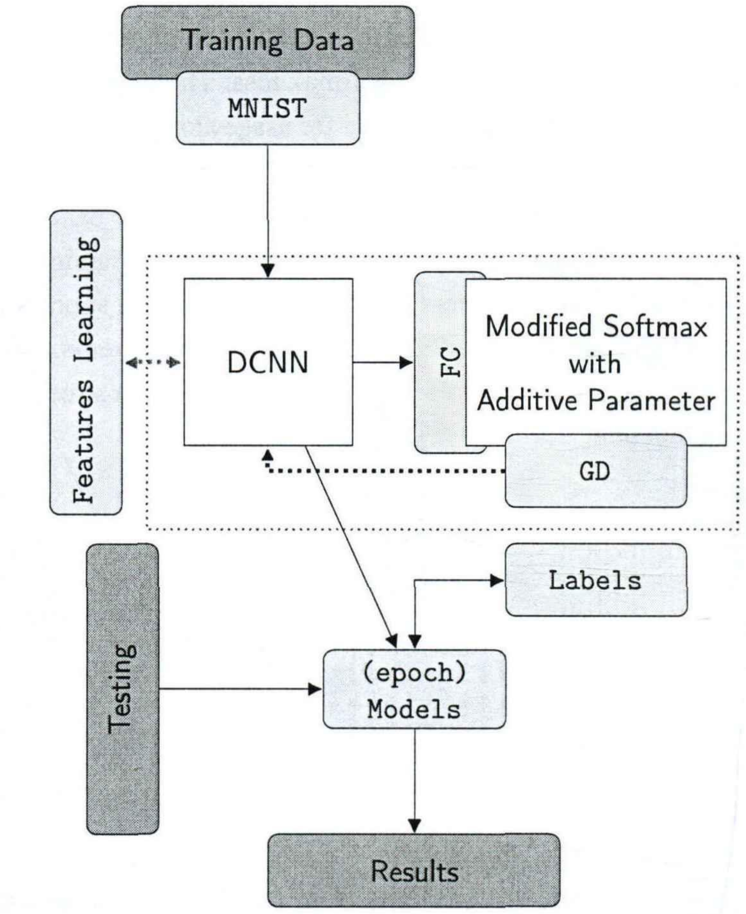


Fig. 4.9 A flow chart represents a process of HDR using deep convolutional neural networks for performance evaluation supervised by the modified softmax loss.

To enhance the discrimination power of softmax loss function, we propose a special case of multiplicative angular margin entitled a unit softmax by fixing the margin to integer value $m = 1$ with a additive parameter $\varepsilon = 0.25$ and setting the convolutional layers to train the model on MNIST dataset. We build a deep model on MNIST dataset using the proposed approach, during training phase, A unit softmax monitors the ConvNet to learn deep features and in the testing phase, the digit features are extracted from the ConvNet to perform characters recognition. Moreover, we experiments the modified softmax loss with the SGD and LS-SGD, both optimizer achieved state-of-the-art performance with consistent improvements.

The modified form of the softmax for handwritten digit recognition is defined by setting angular margin $m = 1$ and fixing the value of additive margin $\varepsilon = 0.25$ to generate modified monotonically decreasing parameter $\psi_{\varepsilon=0.25} = [(-1)^{\lfloor k \rfloor} cos(\theta_{min}) - 2k] - 0.25$ , where $\theta_{min} \in [k\pi, (k+1)\pi]$. Upgrade the equation by adjusting the proposed parameters, a unit softmax is given as

$$L_{\psi_\varepsilon(\theta_{min},1)} = -\frac{1}{N}\Sigma_i \log \frac{e^{||x_i||\psi_\varepsilon(\theta_{min})}}{e^{||x_i||\psi_\varepsilon(\theta_{min})} + \Sigma_j e^{||x_i||cos\theta_j}} \tag{4.1}$$

After the inner product of weight and feature vectors in angular margin approach, a better action is essential that is computationally expensive. The unit softmax approach is theoretically slight similar to the original sofmax loss and computationally it is more appealing and bifacial because of its combined effect of additive and angular margin and it is proficient to adjust with both optimization techniques SGD and LS-SGD. The adjustment of a unit softmax is simple to take gradient during the forward and backward propagation and is trivial to optimize the modified softmax loss using SGD and LS-SGD.

We adjust the network architecture by setting the convolutional units containing multiple convolution layers with output channels on the input by adjusting stable and learnable size of kernel. Motivated by the ResNet [29], we readjust the network architecture by setting the convolutional units containing multiple convolution layers with output channels on the input by adjusting stable and learnable size of kernel, the detailed structure is exposed in the Table 4.6 is applicable to use for handwritten digit recognition.

| **Layer** | **CNN Structure** |
|---|---|
| Conv 1.x | $\left[3\times3,\ 64\right]\times 1$ |
| Conv 2.x | $\left[3\times3,\ 64\right]\times 3$ |
| Conv 3.x | $\left[3\times3,\ 64\right]\times 3$ |
| Conv 4.x | $\left[3\times3,\ 64\right]\times 3$ |
| *FC*1 | 256 |

Table 4.6 Convolutional layers structure for neural network. The notation Conv1.x, Conv2.x, Conv3.x and Conv4.x represents the convolutional units containing multiple convolution layers and FC1 is the fully connected layer.

For experiment of a unit softmax with both SGD and LS-SGD optimization strategy, we begin with a learning rate of 0.1 and divide it by 10 at 10k and 12k iterations to generate 50 epochs and other default parameters are set for PyTorch implementation on GPU. Begin with the dataset into training stage that containing random projection and extraction of the features. In this phase to visualize all digits clearly, each letter in dataset is attentive to displayed in haphazard style and then respective characters are extracted individually to examine its assembly and forward to the learning process for training. This process is iterated until all the objects in the dataset get trained completely.

### 4.2.2 Results and Analysis

The Fig.4.10 represents the training on the Dataset MNIST using SGD and LS-SGD, accuracy (is taken on vertical axis) corresponding to 50 epochs (on horizontal axis) and the Fig. 4.11 represents the loss (on vertical axis) corresponding to 50 epoch (horizontal axis). Its observed from the Fig. 4.10, that a unit softmax is efficiently work with both the optimization schemes and the performance of LS-SGD to train data is significantly improved after the epoch 10 and smoothly exhibits the better heightening as compared to SGD. The Fig.4.11 exposed that, a unit softmax along with LS-SGD attain the notable results on the loss during the training instead of SGD. The minimum loss through SGD is 0.434 and as a result of LS-SGD is 0.384.

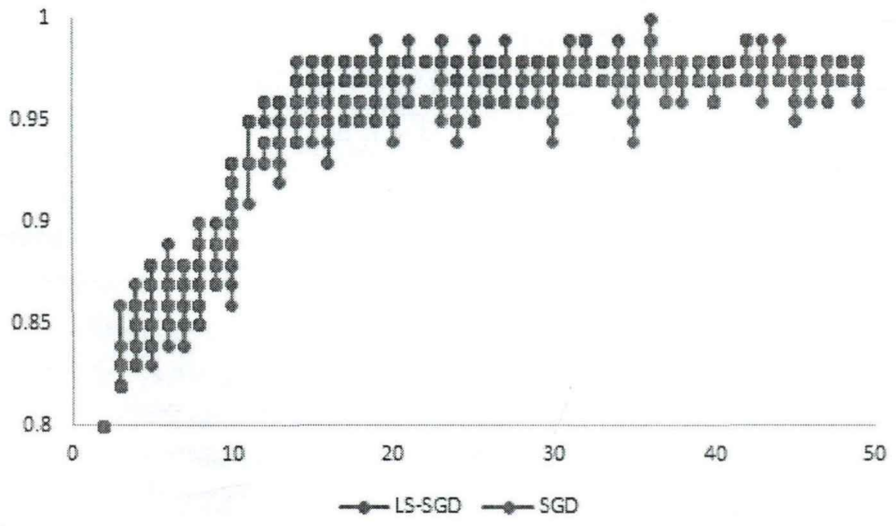


Fig. 4.10 Training using SGD and LS-SGD. Accuracy (%) is taken on vertical axis corresponding to 50 epochs on horizontal axis.

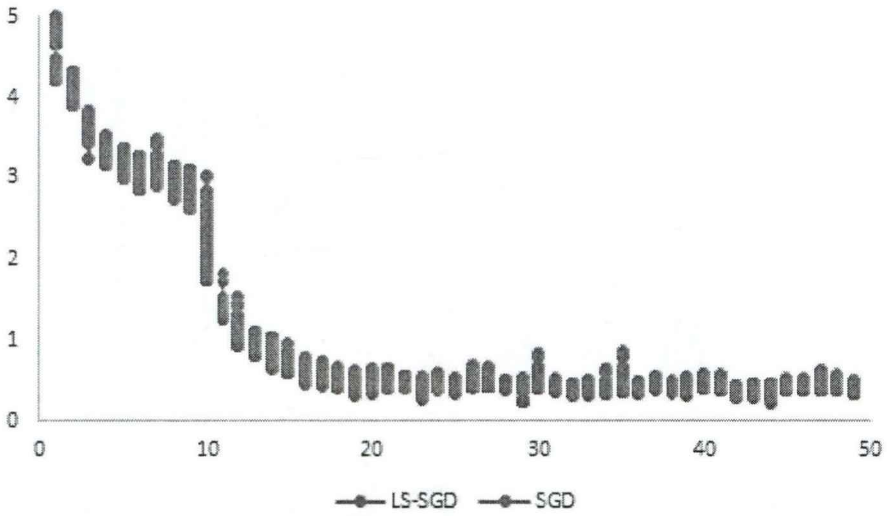


Fig. 4.11 Loss using SGD and LS-SGD. Loss is taken on vertical axis corresponding to 50 epochs on horizontal axis.

For evaluation, we simply construct the final ensemble classifier and use it with the discriminative feature to predict labels. To generate some clusters, the handwritten digits with similar structure and vector-value are assembled together under a single label and every digit feature is analyzed to find the pattern by proposed algorithm.

On both SGD and LS-SGD schemes, the testing accuracy corresponding to randomly selected models $r1$, $r2$, $r3$, $\ldots$, $r10$ is shown in Fig.4.12. Testing accuracy is taken on vertical axis and randomly selected models $r1$, $r2$, $r3$, $\ldots$, $r10$ on horizontal axis. It can be analyzed that the accuracy on the models generated via LS-SGD is better then the generated models via SGD.

The training and testing accuracy represents the proficiency of a unit softmax loss to improve the power of DCNNs along with the both optimization techniques SGD and LS-SGD. However, the results on training and testing demonstrate the advantage of the smoothed gradient as compared to stochastic gradient, but there is a minor difference in results of the modified softmax with SGD and LS-SGD, both optimization schemes are adjustable and stable for a unit softmax loss and helps to find better minima. However, we discuss only one case of additive parameter for the handwritten digits but the other cases also effectively working for digits recognition. The overall focus in this application is to evaluate the re-

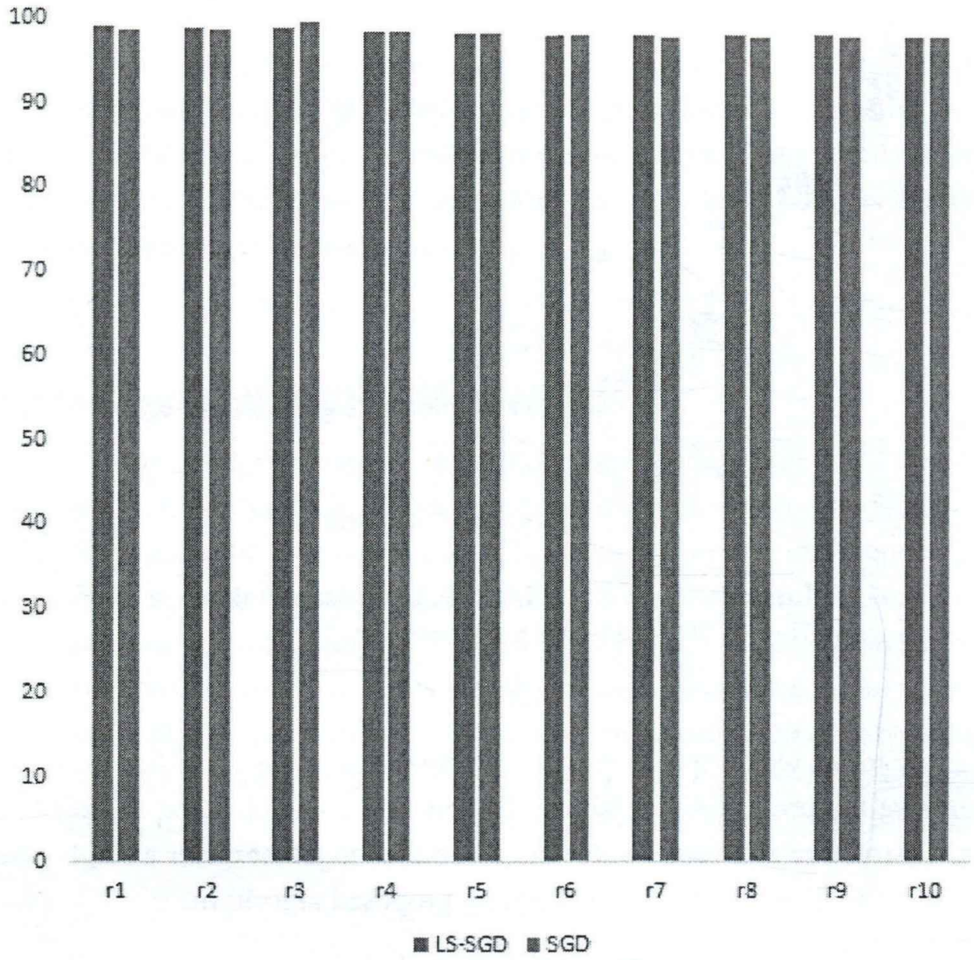


Fig. 4.12 Testing accuracy on randomly selected models $r1$, $r2$, $r3$, ..., $r10$.

sults for SGD and LS-SGD, that's why we discussed only a unit softmax for digit recognition.

The average advanced methods used for handwritten digits resignation are modeled on the base of softmax loss function and its modified forms. For a fair comparison with the latest techniques we compare the alternative form of softmax with those methods having the results on MNIST dataset. However, the profomance of L-Softmax [48] on MNIST dataset is better then the many approaches, but it use maximum margin values to achieve the rate 0.31. Our results on handwritten digits recognition are better then many other state-of-the-art techniques.

| Method | Error Rate |
|---|---|
| DropConnect [85] | 0.57 |
| CNN [32] | 0.53 |
| FitNet [65] | 0.51 |
| L-Softmax [48] | 0.31 to 0.40 |
| Maxout [25] | 0.45 |
| GenPool [39] | 0.31 |
| Unit Softmax with SGD | 0.434 |
| Unit Softmax with LS-SGD | 0.384 |

Table 4.7 Recognition loss (%) on MNIST dataset using a unit softmax loss.

Experimental study summarized in Table 4.7, shows that the loss of a unit softmax is 0.384 with LS-SGD and 0.434 with SGD. It is notable that the results of a unit softmax with both optimization techniques attains state-of-the-art performance on MNIST.

# Chapter 5

# Conclusion and Future Work

In this chapter, we include the conclusion of the proposed work and some future directions.

## 5.1 Conclusion

This presented novel approach towards additive and angular margin softmax loss by offering an additive parameter controlled over multiplicative angular margin, that is more effective for DCNNs to learn highly discriminative face features. An automatic adjustment of the seedling element as the result of angular marginal seed is offered in a particular way for the angular softmax to learn angularly discriminative features. The adjustment of additive parameter is simple to take gradient during the forward and backward propagation and is trivial to optimize the modified softmax loss. Because of combined effect of additive and angular margin, our approach is significantly unique, more efficient, trainable, easy to optimize and stable for usage for image recognition tasks.

We train the model on publically available dataset CASIA-WebFace and our experiments on famous benchmarks YouTube Faces and Labeled Face in the Wild achieve better performance than the various state-of-the-art approaches. It is observed that the performance accuracy of FR on LFW is 99.58% and on YTF is 95.38% corresponding to the margin $m = 3$ and additive parameter $\varepsilon = 0.375$. That is the best evaluation performance using additive parameter trained on CASIAWebFace and is significantly better than several other techniques trained on the same dataset. A comprehensive description of experiments is included in the literature to demonstrate that our technique is better than various current state-of-the-art methods. However, our improved alternative form of softmax is trainable, extremely easy to implement and is appropriate for the larger amount of margin but still need exactly accurate correspondence for largest size of margin.

Moreover, the proposed modification in softmax expresses a flexible learning job with adjustable additive margin and is flexible to amend with stochastic gradient descent and laplacian smoothing gradient descent. To enhance the discrimination power of softmax, a special case of multiplicative angular margin is offered by fixing the margin and setting the convolutional layers to train the model on famous dataset. Experimental results demonstrate a state-of-the-art performance on famous database of handwritten digits the Modified National Institute of Standards and Technology (MNIST) database. The experimental results on MNIST dataset demonstrated the advantages of our modified softmax loss over the state-of-the-art alternatives. It can be observed that the loss of the proposed strategy is 0.384 with LSSGD and 0.434 with SGD, its notable that the results of a unit softmax with both optimizations techniques achieve the state-of-the-art performance compared to the other deep CNN architectures.

For the both applications discussed in Chapter 4, the proposed model is trainable, extremely easy to implement and is appropriate for the larger amount of margin. Experimental results demonstrate that our approach is better than numerous current state-of-the-art approaches and expresses a flexible learning job.

## 5.2 Future Work

As it is concluded that, our approach via improved alternative form of softmax is trainable, extremely easy to implement and is appropriate for the larger amount of margin but still need exactly accurate correspondence for largest size of margin. To further improve the discriminative power of the face recognition system and to stabilize the training process, the exactly accurate correspondence for largest size of margin is required, that will be formed by constructing more appropriate mathematical formula like our proposed method, that could handle the computational complexity during training.

In order to stabilize training, we adopted combined margin strategies for modify the angular softmax, that will suitable for other computer vision tasks, like object recognitions and human action recognitions. On this concern, a novel approach for object recognitions and human action recognitions tasks will generate a promising results via implementation of our proposed method.

Other than the main focus on computer vision tasks, the similar modification based on loss function and gradient descent will be applicable to construct new techniques for speech recognition and signal processing jobs as well as to build new deep learning algorithm for solving complex mathematical models based on partial differential equations.

# References

[1] Agarwal, S., Terrail, J. O. D., and Jurie, F. (2018). Recent advances in object detection in the age of deep convolutional neural networks. *arXiv preprint arXiv:1809.03193*.

[2] Ahonen, T., Hadid, A., and Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (12):2037–2041.

[3] Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43.

[4] Ashiquzzaman, A. and Tushar, A. K. (2017). Handwritten arabic numeral recognition using deep learning neural networks. In *2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pages 1–4. IEEE.

[5] Ba, J., Mnih, V., and Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*.

[6] Baddeley, A. D. and Warrington, E. K. (2017). Amnesia and the distinction between long-and short-term memory 1. In *Exploring working memory*, pages 18–38. Routledge.

[7] Barra, A., Bernacchia, A., Santucci, E., and Contucci, P. (2012). On the equivalence of hopfield networks and boltzmann machines. *Neural Networks*, 34:1–9.

[8] Bhatia, E. N. (2014). Optical character recognition techniques: a review. *International journal of advanced research in computer science and software engineering*, 4(5).

[9] Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

[10] Bodlaender, H. L., Gilbert, J. R., Hafsteinsson, H., and Kloks, T. (1995). Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *Journal of Algorithms*, 18(2):238–255.

[11] Borovkov, A. (2017). *Image Classification with Deep Learning*. PhD thesis, Universität Hamburg.

[12] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer.

[13] Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311.

[14] Cao, Z., Yin, Q., Tang, X., and Sun, J. (2010). Face recognition with learning-based descriptor. In *2010 IEEE Computer society conference on computer vision and pattern recognition*, pages 2707–2714. IEEE.

[15] Chaudhari, P., Oberman, A., Osher, S., Soatto, S., and Carlier, G. (2018). Deep relaxation: partial differential equations for optimizing deep neural networks. *Research in the Mathematical Sciences*, 5(3):30.

[16] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555.*

[17] Defazio, A., Bach, F., and Lacoste-Julien, S. (2014). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654.

[18] Deng, J., Guo, J., Xue, N., and Zafeiriou, S. (2018). Arcface: Additive angular margin loss for deep face recognition. *arXiv preprint arXiv:1801.07698.*

[19] Deng, J., Guo, J., Xue, N., and Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699.

[20] Deng, J., Zhou, Y., and Zafeiriou, S. (2017). Marginal loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 60–68.

[21] Doersch, C. (2016). Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908.*

[22] Evans, L. C. (2010). Partial differential equations. second. vol. 19. *Graduate Studies in Mathematics. American Mathematical Society, Providence, RI.*

[23] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning.* MIT press.

[24] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[25] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013). Maxout networks. *arXiv preprint arXiv:1302.4389.*

[26] Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401.*

[27] Hasnat, M., Bohné, J., Milgram, J., Gentric, S., Chen, L., et al. (2017). von mises-fisher mixture model-based deep learning: Application to face verification. *arXiv preprint arXiv:1706.04264.*

[28] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

[29] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[30] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580.*

[31] Huang, G. B., Mattar, M., Berg, T., and Learned-Miller, E. (2008). Labeled faces in the wild: A database forstudying face recognition in unconstrained environments.

[32] Jarrett, K., Kavukcuoglu, K., LeCun, Y., et al. (2009). What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th international conference on computer vision*, pages 2146–2153. IEEE.

[33] Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323.

[34] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[35] Kulkarni, T. D., Whitney, W. F., Kohli, P., and Tenenbaum, J. (2015). Deep convolutional inverse graphics network. In *Advances in neural information processing systems*, pages 2539–2547.

[36] Laval, J. A. and Leclercq, L. (2013). The hamilton–jacobi partial differential equation and the three representations of traffic flow. *Transportation Research Part B: Methodological*, 52:17–30.

[37] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

[38] LeCun, Y., Cortes, C., and Burges, C. J. (1998). The mnist database of handwritten digits, 1998. *URL http://yann. lecun. com/exdb/mnist*, 10:34.

[39] Lee, C.-Y., Gallagher, P. W., and Tu, Z. (2016). Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In *Artificial Intelligence and Statistics*, pages 464–472.

[40] Lei, Z., Pietikäinen, M., and Li, S. Z. (2013). Learning discriminant face descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):289–302.

[41] Li, S., Yi, D., Lei, Z., and Liao, S. (2013). The casia nir-vis 2.0 face database. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 348–353.

[42] Liang, X., Wang, X., Lei, Z., Liao, S., and Li, S. Z. (2017). Soft-margin softmax for deep classification. In *International Conference on Neural Information Processing*, pages 413–421. Springer.

[43] Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019.*

[44] Liu, C. and Wechsler, H. (2002). Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image processing*, 11(4):467–476.

[45] Liu, C.-L., Sako, H., and Fujisawa, H. (2004). Discriminative learning quadratic discriminant function for handwriting recognition. *IEEE Transactions on Neural Networks*, 15(2):430–444.

[46] Liu, J., Deng, Y., Bai, T., Wei, Z., and Huang, C. (2015). Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv preprint arXiv:1506.07310.*

[47] Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. (2017a). Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220.

[48] Liu, W., Wen, Y., Yu, Z., and Yang, M. (2016). Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, page 7.

[49] Liu, Y., Li, H., and Wang, X. (2017b). Learning deep features via congenerous cosine loss for person recognition. *arXiv preprint arXiv:1702.06890.*

[50] Lu, X., Tsao, Y., Matsuda, S., and Hori, C. (2013). Speech enhancement based on deep denoising autoencoder. In *Interspeech*, pages 436–440.

[51] Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3.

[52] Makkar, T., Kumar, Y., Dubey, A. K., Rocha, Á., and Goyal, A. (2017). Analogizing time complexity of knn and cnn in recognizing handwritten digits. In *2017 Fourth International Conference on Image Information Processing (ICIIP)*, pages 1–6. IEEE.

[53] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

[54] Osher, S., Wang, B., Yin, P., Luo, X., Pham, M., and Lin, A. (2018). Laplacian smoothing gradient descent. *arXiv preprint arXiv:1806.06317.*

[55] Ouyang, W., Luo, P., Zeng, X., Qiu, S., Tian, Y., Li, H., Yang, S., Wang, Z., Xiong, Y., Qian, C., et al. (2014). Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection. *arXiv preprint arXiv:1409.3505.*

[56] Parkhi, O. M., Vedaldi, A., Zisserman, A., et al. (2015). Deep face recognition. In *bmvc*, volume 1, page 6.

[57] Qi, X. and Zhang, L. (2018). Face recognition via centralized coordinate learning. *arXiv preprint arXiv:1801.05678.*

[58] Rahman, J. U., Chen, Q., and Yang, Z. (2019). Additive parameter for deep face recognition. *Communications in Mathematics and Statistics*, pages 1–15.

[59] Ranjan, R., Castillo, C., and Chellappa, R. (2019). L2 constrained softmax loss for discriminative face verification. US Patent App. 15/938,898.

[60] Ranjan, R., Castillo, C. D., and Chellappa, R. (2017). L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507.*

[61] Read, J. (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL student research workshop*, pages 43–48. Association for Computational Linguistics.

[62] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

[63] Rodan, A. and Tino, P. (2010). Minimum complexity echo state network. *IEEE transactions on neural networks*, 22(1):131–144.

[64] Rolfe, J. T. and LeCun, Y. (2013). Discriminative recurrent sparse auto-encoders. *arXiv preprint arXiv:1301.3775.*

[65] Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. (2014). Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550.*

[66] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[67] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747.*

[68] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

[69] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

[70] Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

[71] Sewak, M., Karim, M. R., and Pujari, P. (2018). *Practical Convolutional Neural Networks: Implement Advanced Deep Learning Models Using Python.* Packt Publishing Ltd.

[72] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556.*

[73] Skansi, S. (2018). *Introduction to Deep Learning: from logical calculus to artificial intelligence*. Springer.

[74] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

[75] Sun, Y., Chen, Y., Wang, X., and Tang, X. (2014). Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996.

[76] Sun, Y., Liang, D., Wang, X., and Tang, X. (2015a). Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*.

[77] Sun, Y., Wang, X., and Tang, X. (2015b). Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2892–2900.

[78] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

[79] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

[80] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708.

[81] Tch, A. (2017). The mostly complete chart of neural networks, explained. *Retrieved Aug 4, 2017, from https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464.*

[82] Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86.

[83] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.

[84] Voulodimos, A., Doulamis, N., Doulamis, A., and Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018.

[85] Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R. (2013). Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066.

[86] Wang, F., Cheng, J., Liu, W., and Liu, H. (2018a). Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930.

[87] Wang, F., Xiang, X., Cheng, J., and Yuille, A. L. (2017). Normface: l 2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049. ACM.

[88] Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., and Liu, W. (2018b). Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274.

[89] Wang, M. and Deng, W. (2018). Deep face recognition: A survey. *arXiv preprint arXiv:1804.06655.*

[90] Wei, X., Wang, H., Scotney, B., and Wan, H. (2020). Minimum margin loss for deep face recognition. *Pattern Recognition*, 97:107012.

[91] Wen, Y., Zhang, K., Li, Z., and Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer.

[92] Wolf, L., Hassner, T., and Maoz, I. (2011). *Face recognition in unconstrained videos with matched background similarity*. IEEE.

[93] Wu, S., Wang, Y., and Cheng, S. (2013). Extreme learning machine based wind speed estimation and sensorless control for wind turbine power generation system. *Neurocomputing*, 102:163–175.

[94] Wu, X., He, R., Sun, Z., and Tan, T. (2018). A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896.

[95] Xu, B., Wang, N., Chen, T., and Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853.*

[96] Zeiler, M. D. and Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557.*

[97] Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503.

[98] Zhang, Q., Yang, L. T., Chen, Z., and Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, 42:146–157.

[99] Zhang, S., Choromanska, A. E., and LeCun, Y. (2015). Deep learning with elastic averaging sgd. In *Advances in Neural Information Processing Systems*, pages 685–693.

[100] Zhang, W., Itoh, K., Tanida, J., and Ichioka, Y. (1990). Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Applied optics*, 29(32):4790–4797.

[101] Zhang, W., Shan, S., Gao, W., Chen, X., and Zhang, H. (2005). Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 786–791. IEEE.

[102] Zhang, X., Fang, Z., Wen, Y., Li, Z., and Qiao, Y. (2017). Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5409–5418.

[103] Zheng, Y., Pal, D. K., and Savvides, M. (2018). Ring loss: Convex feature normalization for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5089–5097.

# Research work conducted during PhD

1. **Jamshaid ul Rahman**, CHEN Qing, Zhouwang Yang, "Additive Parameter for Deep Face Recognition", *Communications in Mathematics and Statistics*, (2019).

2. **Jamshaid ul Rahman**, Akhter Ali, Masood Ur Rehman and Rafaqat Kazmi "A Unit Softmax with Laplacian Smoothing Stochastic Gradient Descent for Deep Convolutional Neural Networks" $2^{nd}$ *International Conference on Intelligent Technologies and Applications, (INTAP 2019).*

3. **Jamshaid ul Rahman**, Muhamad Usman, Masood ur rehman, "The Laplacian energy of diameter 4 trees" *Journal of Discrete Mathematical Sciences & Cryptography, Taylor & Francis*, (2019).

4. **Jamshaid ul Rahman**, Muhammad Suleman, DianChen Lu, Ji-Huan He, and Muhammad Ramzan. "He-elzaki method for spatial diffusion of biological population". *Fractals*, (2019).

5. **Jamshaid ul Rahman**, Umair Khan, Shafiq Ahmad, Muhammad Ramzan, Muhammad Suleman, Dianchen Lu, Saba Inam "Numerical simulation of Darcy-Forchheimer 3D unsteady nanofluid flow comprising carbon nanotubes with Cattaneo-Christov heat flux and velocity and thermal slip conditions" *Processes*, (2019).

6. Muhammad Suleman, Dianchen Lu, Ji-Huan He, UMER FAROOQ,YE SHENG HUI, **Jamshaid ul Rahman**, "Numerical Investigation of Fractional HIV model using Elzaki Projected Differential Transform Method" *Fractals*, (2018)

7. Muhammad Suleman, Dianchen Lu, Chen Yue, **Jamshaid Ul Rahman** and Naveed Anjum, "He–Laplace method for general nonlinear periodic solitary solution of vibration equations" *Journal of Low Frequency Noise, Vibration and Active Control*, (2018).

8. Muhammad Suleman, Dianchen Lu, **Jamshaid Ul Rahman**, Naveed Anjum "Analytical Solution of Linear Fractionally Damped Oscillator by Elzaki Transformed Method" *DJ Journal of Engineering and Applied Mathematics*, (2018).