IMPROVING CNN ACCURACY BY TRAINING ON AUXILIARY DATA SOURCE

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science and Engineering

Submitted By
Harshdeep Singh
Roll. No. 801732018

Under the supervision of:
Dr. R. K. Sharma
Professor

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA – 147004

July 2019

CERTIFICATE

==========

I hereby certify that the work which is being presented in the thesis entitled, "*Improving CNN Accuracy by Training on Auxiliary Data Source* ", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, is an authentic record of work carried out under the supervision of *Dr. R. K. Sharma* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Signature:

(Harshdeep Singh)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. R.K. Sharma)
Professor,
Computer Science and
Engineering Department
TIET, Patiala.
31.07.2019

i

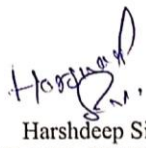
ACKNOWLEDGEMENT

I express my sincerest regards and gratitude to my supervisor Dr. R. K. Sharma, Professor, Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala for his valuable guidance and suggestions. Without his encouragement and guidance, thesis would not have been materialized. I feel privileged to offer my sincere thanks and owe an enormous deal of gratitude to my supervisor for his guidance and gave me full time to understand the minute details of each and every step, for successful completion of thesis.

I would like to express my gratitude to Dr Ashutosh Mishra, Assistant Professor Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala for his kind cooperation and encouragement which helped in the completion of this work.

The generous support of all the staff members of Computer Science Department is greatly appreciated. I would like to express my heartiest thanks to my parents and friends for their help and wishes for the successful completion of this work.

Above all, I express my indebtedness to the "**ALMIGHTY**" for all his blessings and kindness.



Harshdeep Singh
(Roll No 801732018)

ii

ABSTRACT

The standard model of supervised learning assumes that training and test data are drawn from the same underlying distribution. This thesis explores an area in which a second, (auxiliary), source of data is available and is drawn from a different distribution. This auxiliary data might be plentiful, but of significantly lower quality, than the training and test data. In the CNN framework, the Softmax function gives the probabilities that are further used in classification. This thesis considers using the auxiliary data in either of these roles. This auxiliary data framework is applied to a problem of classifying images of MNIST Handwritten dataset and also on Handwritten Gurmukhi Script dataset. Experiments show that even when the training dataset is small, training with auxiliary data can produces improvements in accuracy. When the dataset is large, the improvements in accuracy are even higher.

TABLE OF CONTENTS

==========

LIST OF FIGURES

==========

LIST OF TABLES

==========

CHAPTER-1

==========

INTRODUCTION

==========

1.1 Classification Problem

Classification is a form of data analysis that utilizes models to describe important classes of information (data). Categorical class (i.e. discrete or unordered descriptions) is predicted by such models, called classifiers. For example, a classification model can be developed to categorize bank loan applications as secure or dangerous. Such assessment can assist us to better understand the information as a whole. Researchers have suggested many techniques of classification in machine learning involving, statistics, pattern recognition, and medical science. The majority of algorithms reside in memory and usually assume a small dataset size. Recent data mining and deep learning study has constructed on these projects, develops scalable classification and forecast methods that can handle big quantities of information (data) residing in a disk. There are many applications of classification, including target marketing, fraud detection, manufacturing, predictive efficiency, and medical diagnostics. In Section 1.1.1 we present the classification idea. In section 1.1.2, the general classification method is described as a two-stage method.

1.1.1 Defining the Classification

To know which credit candidates is "secure" or "dangerous" for the bank, a bank loan officer requires an assessment of their information. A sales director at an electronics shop requires information assessment to imagine if a client is going to purchase a fresh laptop with a specific profile. A medical researcher intends to evaluate information on breast-cancer to predict, which one of three particular medicines a person should take. For each one of them the job of the data analysis is to classify where the classifier model is created to estimate class (categorical) labelling, such as for credit request information "secure" or "dangerous," for advertising information "yes" or "no," or for the medical information "therapy A," "therapy B," or "therapy C." These categories are represented by discrete values that don't have any significance to order between values. For

instance, values 1, 2, and 3 may serve as the basis for A, B and C therapies, where no implicit order exists between these treatment regimes. Suppose the marketing manager wants to estimate how long a specified customer spends on electronics shop during a sale. An instance of a numerical forecast is this data analysis job where the built model predicts a continually assessed feature or a structured value rather than a class label. Analysis of regression is the most common statistical method used for the numerical prediction, therefore the two terms have been used synonymously, although there are other numerical prediction methods. Classification and numerical prediction are the main types of prediction problems. Classification is the focus of this section.

1.1.2 General Classification Approach

Classification of data is the two steps process: the learning process (where the classification model is constructed) and the classification phase (in which class labels are predicted for data provided). For the application data for loans in Fig. 1.1 the process is shown. (In fact, we may expect many more features to be taken into account. (The information is streamlined for illustrative reasons). The first step is to build a classifier that describes a default data class or concept. This is the learning stage (or training phase), where a classification algorithm builds an analysis or "learning from" classification through a course consisting of database tuples and associated class labels. A tuple, X, is depicted by the vector of a n-dimensional variable, X= (x1, x2, …, xn), showing n measures from n of the database characteristics, A1, A2, ..., An. Each tuple, X, is believed to belong to a predefined class, as defined in an attribute called the label class attribute. The attribute for the class label is unordered and discrete. It is categorical (or nominal) in that every value is used as a class or category. The individual tuples that make up the training collection are termed tuples and are analyzed randomly from the list. Data tuples can be referred to as tests, examples, situations, information points or items in the framework of classification. As every training tuple's classifier is provided, this step is also known as supervised learning (i.e. the classifier's learning is "controlled" because it says which class each tuple belongs to).

This first step in the classification process can also be seen as a mapping or function learning, y = f (X) which can predict the class Y associated with a given tuple X. In this regard, the mapping or feature between the information groups is to be learned. This mapping usually takes the shape of the guidelines for classification, decision-making

bodies or mathematical formulas. The mapping in our instance is described as classification laws which define credit requests as secure or dangerous (Fig. 1.1 (a)). These guidelines may be used for the classification of potential information multiples and for a closer look at the information contents. They provide an overview of compressed data.

The model will be used for classification in the second phase (Fig. 1.1 (b)). Firstly, the classifier's predictive accuracy is tested. If we use the training set to assess the accuracy of the classifier, this prediction would probably be hopeful, as the classifier continues to overfit the information (i.e. during training it may integrate some specific anomalies of the training records that are not present in the overall data set). A classifier's accuracy on a specified test set is the proportion of test set tuples that the classifier properly classifies. The classifier can be used to classify successive information tuples for which the class label is not recognized if the classifier's accuracy is deemed sufficient. For example, the classification rules learned in Fig. 1.1 (a) may be used to approve or reject new or future loan applicants by analyzing data from previous loan applications.

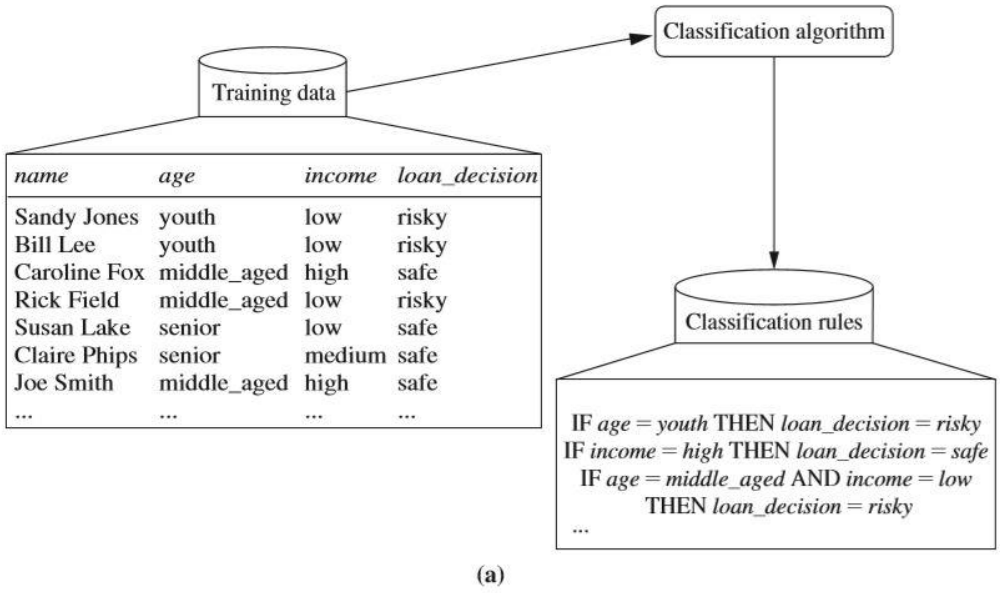


Fig. 1.1: Classification process. (a) Learning phase of algorithm. (Ref: http://shareengineer.blogspot.com/2012/09/classification-and-clustering.html)

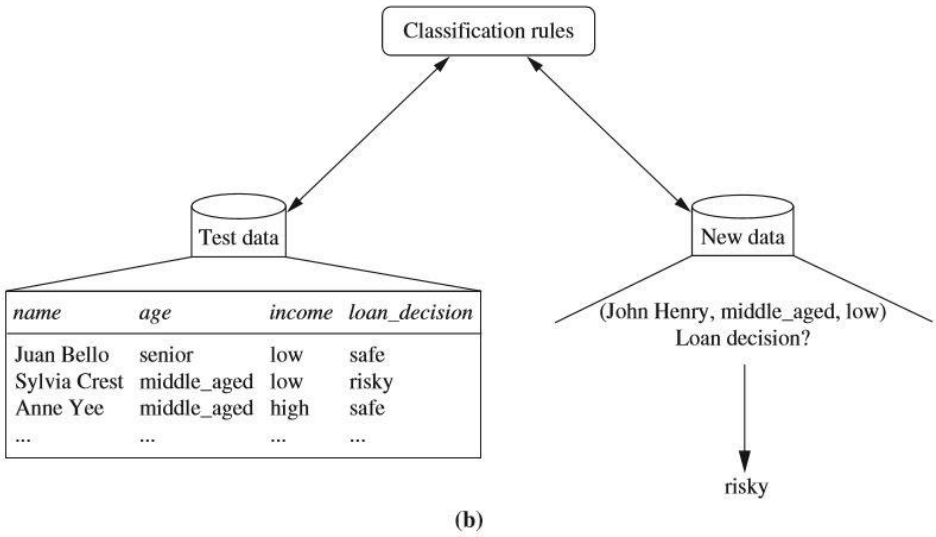


Fig. 1.1: Classification process. (b) Classification

(Ref.: http://shareengineer.blogspot.com/2012/09/classification-and-clustering.html)

1.2 Techniques of Classification

Fig. 1.2 contains a few classification models. In this section, these models are described, in brief.

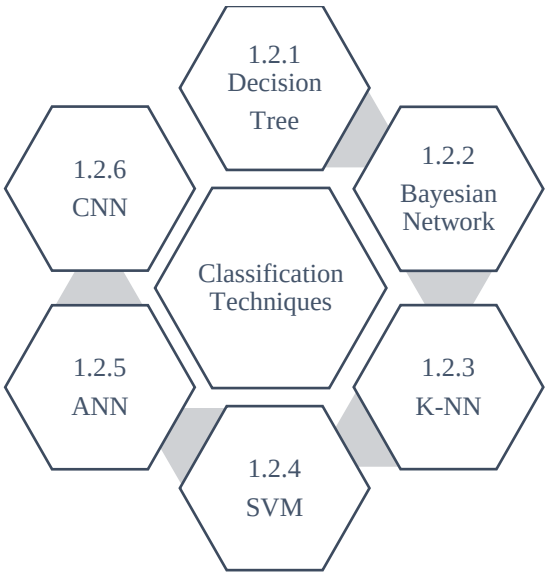


Fig. 1.2: Classification techniques

1.2.1 Decision Tree Induction

Decision tree algorithms are most widely used in classification algorithms (Twa *et al.*, 2005). Decision tree offers an effectively understandable method of modeling and also

simplifies the method of classification (Broadley and Utgoff, 1992). The decision tree is a transparent system that allows consumers to readily follow a tree structure to see how they make the decision (Jang *et al.*, 1993).

Decision tree's key goal is to create a model based on countless input factors that calculates the importance of a necessary variable (Kesavaraj *et al.*, 2013). Normally all decision tree algorithms are built into two phases (i) the growth of tree; training is reciprocally divided on the basis of local optimal criteria until the majority of records belonging to the score have the same label (Rutkowski *et al.*, 2012) (ii) a size reduced to simplify the understanding of the tree (Patil *et al.*, 2010).

In 1986, the choice algorithm for ID3 (Iterative Dichotomiser 3) was launched (Quilan *et al.*, 1986 and Quilan *et al.*, 1987). Due to its efficiency and ease, it is one of the most commonly used algorithms in data mining and machine learning (Quilan *et al.*, 1986). The ID3 is the information gain algorithm. Some of ID3 Decision Tree's strengths and weaknesses are described in (Sharma *et al.*, 2013). The Strengths are as follow:

i) Algorithm understanding is easy and
ii) The whole training data is considered in the final decision of the network.

The weaknesses are as follow:

i) Back-tracking is not used for search,
ii) Handling of missing values are not there and
iii) Global optimization is not used.

C4.5 is a well-known decision trees algorithm. The ID3 algorithm is expanded and its disadvantages induced by ID3 are minimized. In the C4.5 cutting process, the discomfort strands are eliminated by exchanging them from the tree with leaf nodes once it was created (Bhukya *et al.*, 2010).

The strengths of C4.5 are given below:

i) Training data deals with missing feature values,
ii) It deals with both discrete and continuous features and it provide the facility of pruning in both pre and post methods (Sharma *et al.*, 2013, Adhatrao *et al.*, 2013).

Its weaknesses are given below:

i) For the small data set, it is not suitable (Sharma *et al.*, 2013) and

ii) As comparing with other decision trees, the processing time is high.

1.2.2 Bayesian Networks

The graphical system for likelihood connections is used by the Bayesian Network (BN) between a collection of variables (Phyu *et al.*, 2009). BN structure S consists of a direct acyclic graph (DAG) and S nodes communicate one - to-one with X. The arcs show unexpected effects between the nodes, while the shortage in S of feasible arcs encodes conditional freedoms (Soofi *et al.*, 2017). Tasks can usually be divided into 2 subtasks that are; (a) DAG structure learning in the network, and (b) parameter setting.

One of the issues with the classification of Bayesian networks is that constant attributes are generally discretized. The method of converting the permanent attribute into a separate one has brought problems with classification (Yang *et al.*, 2009, Friedman *et al.*, 1996). These problems could involve noise, lack of data and the awareness of shift to class variable attributes (Ren *et al.*, 2008). The other technique Bayesian used in an information set provided at an UCI machine learning library shows that constant characteristics offer a stronger rating accuracy than other methods with the Bayesian network classification system using Gaussian kernel function.

Bayesian network's advantages as presented in Cover *et al.* (1967) are:

i) The properties are smooth; in this model, Minor modifications do not affect the operation of the scheme.

ii) Applicability of this network is flexible; model that are identical can be used for resolving both classification and regression issues.

iii) Missing data can be handled.

1.2.3 K- Nearest Neighbor

K nearest neighbor is a classification technique where the data is classified based on its nearest neighbor values. It can either be structure based or be structure less. The structure-based technique is free of the association between training data samples, rather it has a basic structure of its own (Wu *et al.*, 2008). In structure less technique entire data is treated as training data and the distance is calculated between all training

points. The smallest distance is known as nearest neighbor (Bhatia, 2010). It is effectively applicable to large datasets and is robust when applied to noisy datasets. It is simple to implement and understand. However, the space requirement and time taken to classify are two major constraints to its widespread application. Also, it has a tendency to bin continuous data.

1.2.4 Support Vector Machines

Support Vector Machine is one of the most widely used techniques in data classification. Support vectors are the points that lie the closest to the decision surface (Berwick, 2011). In this a high dimensional space is classified using a hyper-plane (Ahmad *et al.*, 2010). Binary classification is easiest achieved by using the maximal margin classifier (Wu *et al.*, 2008). SVM can not only classify non- linearly separable problems but also the ones with high dimensionality. However, to attain good results from an SVM, a lot of parameters need to be correctly set which is not easily achieved.

1.2.5 ANN

Artificial Neural Networks are inspired by biological neural networks are a good choice for when the dataset has a large number of inputs and the relation between the attributes is unknown. The interconnected neuron architecture computes the output from input values adaptively. Connections between many neurons are given weights and an output is produced based on the input received. These neurons are organized into layers such that the input layer receives input and the output layer produces output. If the output is correct the weights on the connections are reinforced and if the output is incorrect then the weights are recalculated. Given their complex composition, the neural networks when implemented on one system can be slow. However, their architecture is such that it is suitable for parallel implementation. This reduced the developmental speed and cost. This is a widely popular technique being used in a variety of real-world applications such as visual pattern recognition, hand writing analysis and speech recognition etc.

1.2.6 CNN

The class of neuro-artificial networks, which have taken a lead in multiple computer vision assignments, Convolution Neural Network (CNN) attracts concern across several fields, including classification. The CNN approach is designed to learn the

spatial hierarchy of features automatically and adaptively by backpropagation of various building blocks, essentially convolutional layers, pooling and fully-connected layers. This section gives a view of the fundamental ideas of CNN and its implementation to multiple classification functions and describes the difficulties it poses and potential guidelines.

1.2.6.1 Defining CNN (Convolutional Neural Network):

CNN is a profound study model for information handling that has a grid pattern such as pictures which inspire to discover spatial hierarchies from low-to high-level models in an automatic and adaptive manner. It consists of three kinds of layers. The first two are convolution and pool layers. The 3rd one, a fully connected layer, maps the characteristics that have been obtained into final production, such as ranking.

A convolution layer performs an important part in CNN, consisting of a set of math's, such as a specialty linear type, convolution. Images store pixel values in a 2-D grid, i.e. a number array, and at every image position a small parameter grid called the kernel, an optimizable feature extractor, is used, making Convolutional NNs highly effective for the work of image-processing because a feature can appear anywhere within the image. When one layer enters the next layer in its production, extracted functions can become more complicated hierarchically and slowly. The method of optimizing kernels is called learning to minimize the distinction between inputs and input realities through a back propagation and gradient descent optimization algorithm among other things. The kernels are used to optimize parameters.

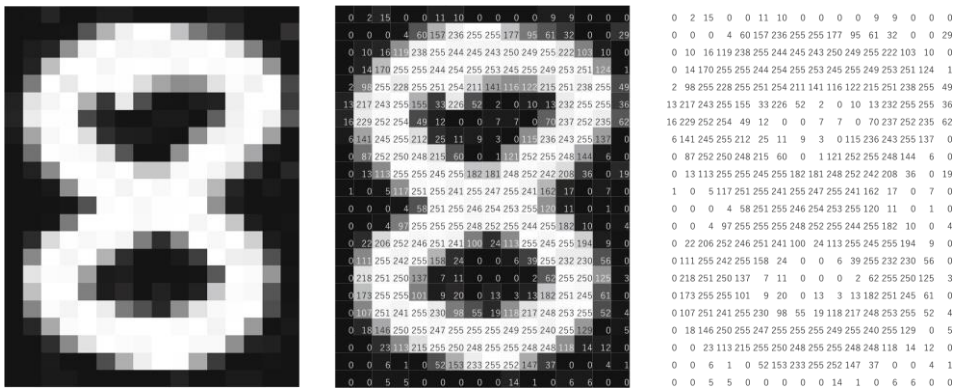


Fig. 1.3: A picture is seen by a machine as a number set.

The left matrix includes digits from 0 to 255, which are each pixel luminous on the left-hand picture. In the center picture both are overlaid.

(Source: http://yann.lecun.com/exdb/mnist)

1.2.6.2 The Architecture of CNN

The planning of the Convolutional Neural Network comprises several blocks, for example, Convolution layers (includes convolution and Non-linear functions), Pooling-layers (includes filter size, strides, padding, and the method of Pooling) and Fully Connected layers (includes no. of weights and the activation). The architecture of CNN consists of a multi-layer convolution layers stack and then a pooling layer that followed by single or more than one fully connected layers. This phase is called a forward Propagation (Fig. 1.4). Input information is converted to output through these layers.

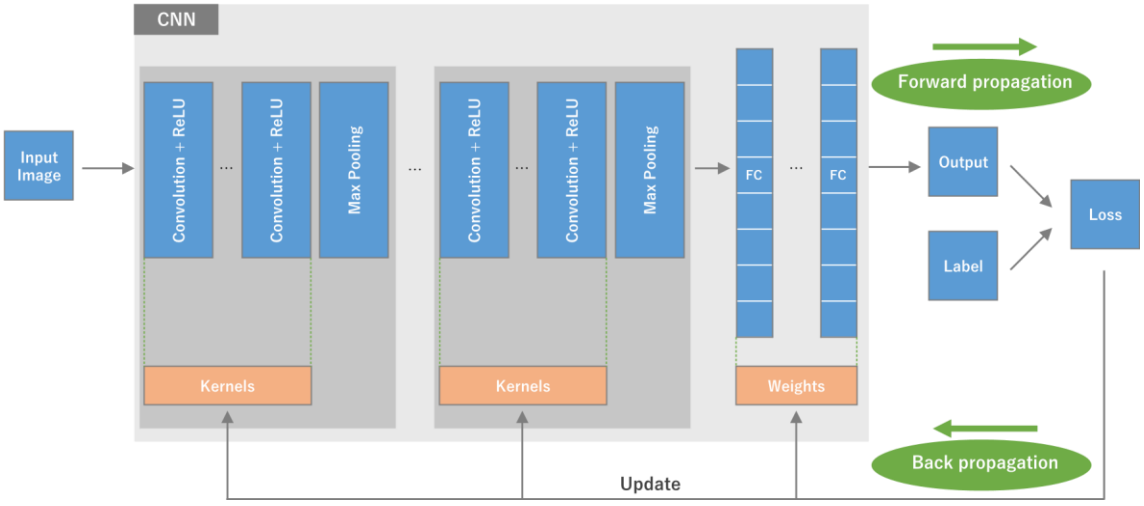


Fig. 1.4: Architecture of CNN (Source: Yamashita, 2018).

i) Convolution Layer

This layer is a core part of the Convolutional Neural Network, which usually be composed of the combo of linear and non-linear operations, that is, the procedure of convolution and the non-linear activation.

Convolution

Convolution is a sequential operation specific for the extraction of features of the input, in which a small or a tiny number array, called kernel, is applied over the input that means a number array called a tensor. Element wise products are calculated between each kernel aspect and the tensor input at every tensor place and summed up in order to achieve the yield value in the appropriate output matrix place, called a feature map shown in Fig. 1.5 (a–c). This approach is recursive by applying a no. of kernels to the

feature-maps representing the different features of the input tensors, so that various kernels can be considered as various function extractors shown in Fig. 1.5 (d).

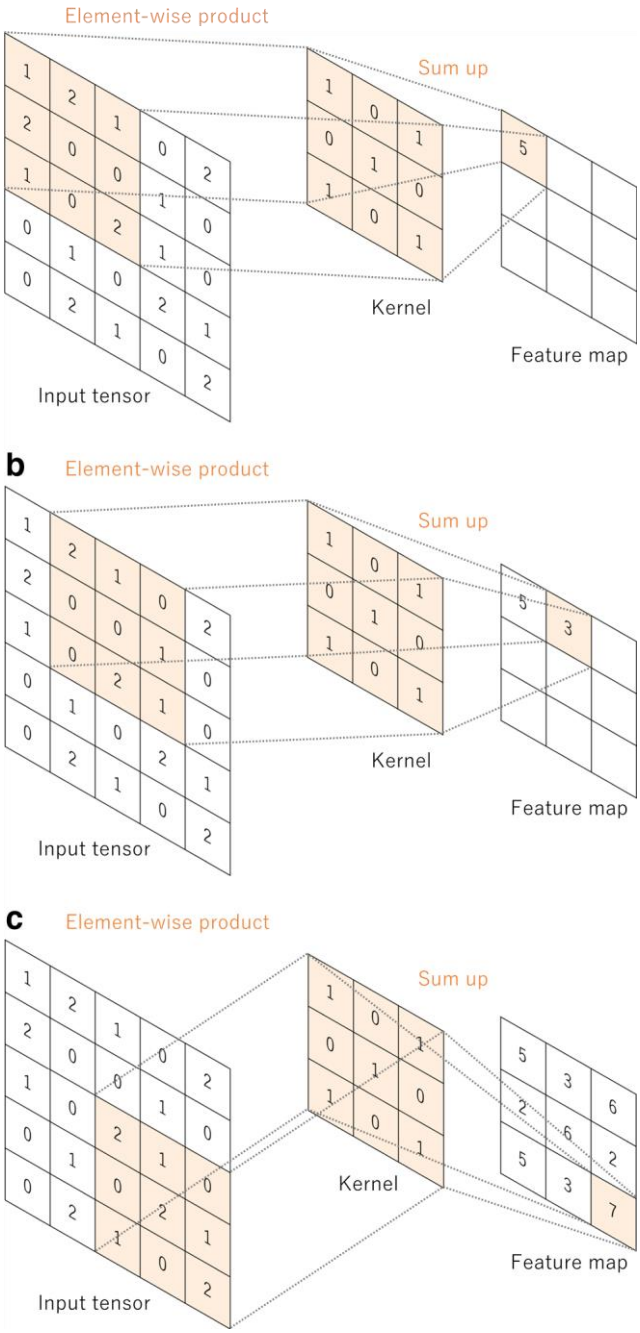


Fig. 1.5 (a – c): An instance of a kernel volume of 3X3, no padding, and 1 step convolution operation (Source: Yamashita, 2018).

The input tensor is applied by a kernel and an elementary product is computed at every point from each of the element of the kernel to the input tensor, summing the value of output to the appropriate location of the output tensor, which is referred to as the feature

map. Examples are shown how kernels derive characteristics from an input tensor in convolution layers. Multiple kernels operate like a horizontal edge sensor i.e. at the top, vertical edge sensor shown in middle, and sketch detector shown at the bottom as distinct functionality extractors.

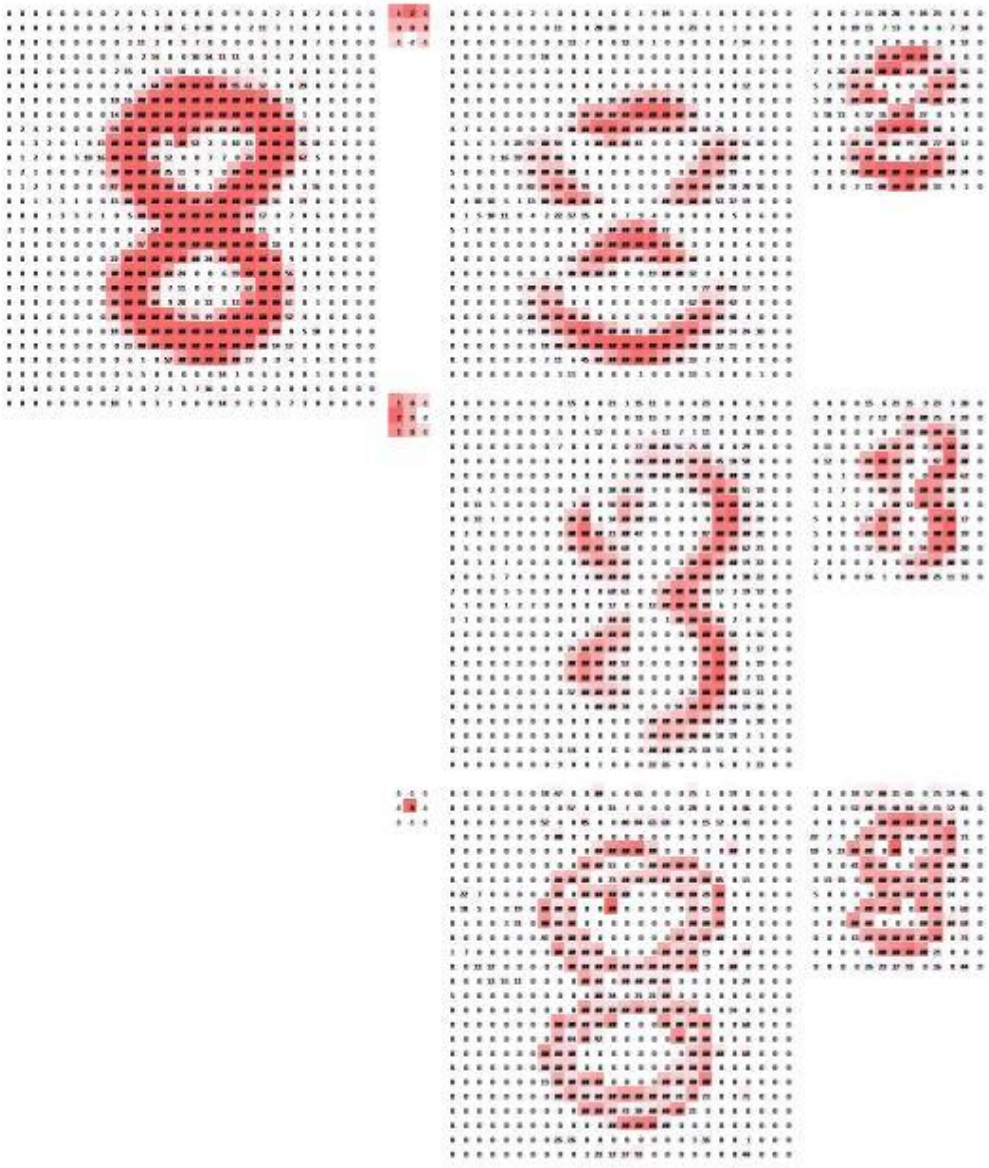


Fig. 1.5 (d): An instance of a kernel volume of 3X3, no padding, and 1 step convolution operation (Source: Yamashita, 2018).

Kernel amount and kernel size are two main hyper-parameters for the convolution method. The first is generally 3 × 3 but sometimes 5 × 5 or 7 × 7. The latter is arbitrary and describes the range of manufacturing features. The overlapping operation of every kernel core does not allow the external element of the input tensor to be overlapped, reducing the height of the output function map and also for the width, relative to the tensors. A technique to address this issue is padding, typically zero padding, where

zeros rows and columns are attached on both sides of a tensor to adjust the middle of a kernel to the external part, maintain the identical in plan dimension to the other side of the tensor process.

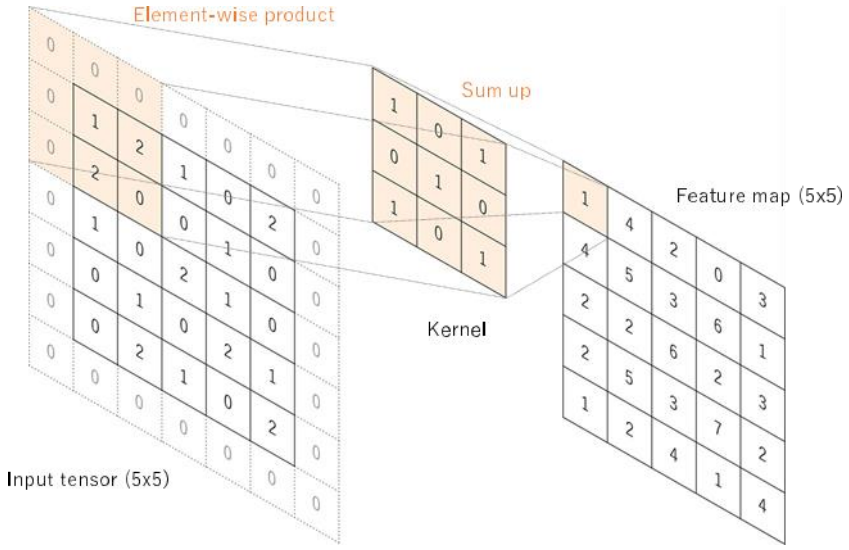


Fig. 1.6: The image shows zero padding in convolution operation for retaining the in-plane dimensions (Source: Yamashita, 2018).

Typically, current Convolutional NN architectures use zero padding to maintain flat dimensions, to use more layers. Following the convolution operation, each map would become smaller without zero padding. The distance between 2 consecutive kernel places is also a phase that defines the conversion process. The frequent choice of a step is 1; although, to accomplish down-sampling of the function maps, a process bigger than step 1 is sometimes used. An alternative down-sampling method, as outlined below, is a pooling procedure. Weight sharing is the main characteristic of a convolution procedure: kernels are distributed across all picture locations. The following features of convolution activities are created by weight sharing:

i) Letting kernel translation local feature patterns invariant as kernels proceed beyond all image positions and detecting learned local patterns,

ii) Learning the function models of the spatial hierarchies by down-sampling together with a pooling procedure, leading in a progressively wider field of perspective being captured.

iii) Improve model effectiveness by decreasing the amount of learning parameters compared to fully linked neural networks.

As described later, with regard to the convolution layer, the process of training a CNN model is to recognize the kernels that work supreme for a particular task based on a prescribed training dataset. Kernels are the exclusive parameters that are automatically learned in the convolution layer during the training process; in contrast, the size of kernel, number of kernels, padding, and stride are hyper-parameters that are best needed before the training process begins.

Table 1.1: A list of parameters and hyper-parameters in a CNN

CNN Layers	Hyper-parameters	Parameters
Convolution layer	Activation function, Kernel size, stride, padding, number of kernels.	Kernels
Pooling layer	Pooling method, filter size, stride, padding	None
Fully connected layer	Number of weights, activation function	Weights
Others	Model-architecture, loss-function, weight initialization, mini-batch size, regularization, optimizer, learning rate, epochs, splitting of dataset	

Nonlinear Activation Function

A nonlinear activation function transfers the inputs of a linear operation, such as convolution. Although nonlinear features are linear, such as hyperbolic or sigmoid. Tangent (tanh) feature, earlier used as mathematical depictions of biological neuronal behavior, the most prevalent non-linear activation function currently used is the linear rectified unit (ReLU), which merely calculates the feature: $f(x)=max(0, x)$ (Fig. 1.7)

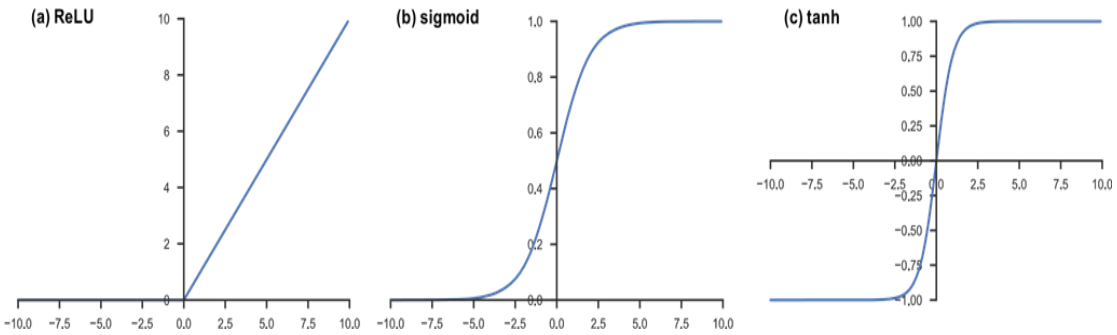


Fig. 1.7: Commonly applied activation functions to neural networks (Source: Yamashita, 2018)

ii) Pooling layer

A pooling layer offers a typical sampling procedure, which decreases the dimensionality of in-plan model maps so that tiny changes and distortions can occur as an invariance and the amount of following learning parameters can be decreased. It is worth noting that in any grouping layer, no learning parameter is available, while in pooling activities, filter volume, step and padding are like turbocharged activities.

A) Max Pooling

The most common way of pooling is by max pooling. Patches are extracted from input character maps; the maximum value is output in every patch and other values are discarded (Fig. 1.8).

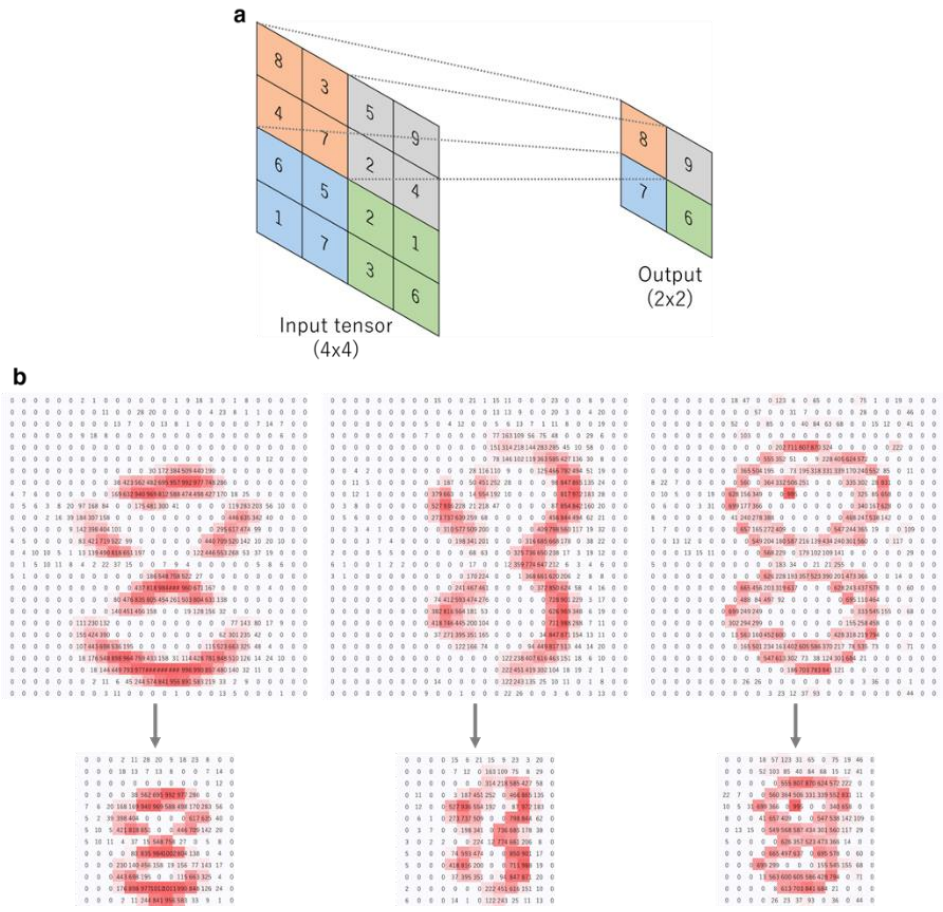


Fig.1.8: a. Max Pooling procedure (Source: Yamashita, 2018).

Most frequently in exercise is a max pool with a filter volume 2x 2. This shows the in-plan aspect of the characteristic fields by a factor of 2. The depth dimension of function maps is unchanged in contrast to height and breadth.

B) Global average pooling

Another noteworthy pooling activity is a median worldwide pooling (Lin *et al.*, 2013). An extreme down-sampling type is done through a global average pooling where a characteristic map with a maximum height of 1 to 1 is simply sampled using the average of all items in each characteristic, while the depth of the characteristic map is maintained. Typically, this function is only used once in the fully connected layers. The benefits of worldwide median pooling include: (1) decreases the number of parameters to be studied and (2) allows CNN to handle varying size inputs.

iii) Fully connected layer

The yield Feature Maps of the ultimate convolution or pooling layer are usually blurred, that is, they are converted to a one-dimensional range of figures or vectors, and linked to one or more fully connected layers, also recognized as thick layers. Once the features are drawn by the convolution layers and taken from the bundling layers, they are mapped by a subset of fully connected layers to the final outputs of the network, such as the probabilities for each class in classification tasks. Typically, the ultimate completely linked layer has the same amount of input nodes as the class amount. A nonlinear function, like a ReLU, as outlined above follows each fully connected layer.

iv) Final layer activation function

In the last Fully Connected layer, the activation function generally differs from the others. Each assignment requires to be chosen with a suitable activation function. A multi-class classification task activating function is a Softmax function that normalizes real output values from the last layer fully connected to the target class probabilities where each value is ranging from 0 to 1 with a sum of all the values 1. Last layer Activation Function typical decisions various types of tasks that Summarized in the below table.

Table 1.2: A list of commonly applied last layer activation functions for various tasks

Activation functions and its Tasks	
Sigmoid	Binary-classification
Softmax	Multiclass-single-class classification
Sigmoid	Multiclass multi-class-classification
Identity	Regression to continuous values

1.3 Training of CNN

Network learning is a method to find kernels in convolution layers and weights in completely linked layers, so as to minimize variations in performance projections and marks on a training dataset. The backpropagation algorithm is the technique widely

used in neural network learning, with loss feature and gradient estimation algorithm. A model of results under specific kernel sand weights is determined by a loss function by propagation in advance on the trained dataset, and by the optimization algorithm known as backpropagation, gradient descent, among others, learning parameters, namely kernels and weights.

1.3.1 Loss function

A loss function, also called a cost function, measures compatibility between the network output predictions by forward propagation and labelling of ground truth. Cross-entropy is the common multi-class loss function, while medium-squared-errors are normally added to constant scores for regression. A sort of loss function has to be determined by the specified assignments as part of one hyper-parameter.

1.3.2 Gradient descent

Gradient descent is frequently used as an optimization algorithm which changes the network's learning parameters, i.e. kernels and weights, to minimize losses iteratively. The loss function gradient gives us the order in which the feature is steepest and each training parameter in its adverse path is modified to an independent stage magnitude depending on a hyper-parameter called the learning frequency (Fig. 1.9).

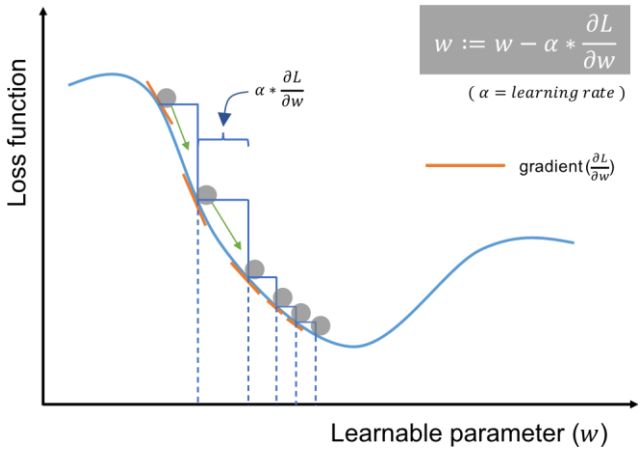


Fig. 1.9: An optimization algorithm called the gradient descent (Source: Yamashita, 2018).

Mathematically, the gradient is a partial loss derivative with regard to each learning parameter and a single parameter adjustment is made as follows:

$$w := w - \alpha * \partial L / \partial w$$

Where w means every parameter to be learned, α is the frequency to be learned, and L is the loss feature. In reality, it is essential to remember that a teaching speed is one of the most significant hyper-parameters before starting the training. Practically, for reasons like memory limits, a subset of the training data set called the mini-batch is used to calculate the loss function gradients for parameters and to update parameters. This technique is known as a mini-batch-gradient descent (SGD), and a mini-batch-size is also called a hyper-parameter. In fact, there are several suggestions and commonly used improvements on the gradient-descending algorithm, such as SGD with impulses, the RMS prop and Adam etc. (Yamashita *et al.*, 2018; Goodfellow, 2014; Su *et al.*, 2019).

1.3.3 Data and ground truth labels

Data and label of the ground truth are the main research components of deep learning or other methods of machine learning. The cautious compilation of information and floor reality labeling for the training and testing of a template is obligatory for a good profound training venture but the acquisition of high-quality marked information can be expensive and tedious. In such cases, special attention should be paid to the quality of the ground truth labels while multiple medical image datasets may be opened to the public.

Whole Dataset are typically divided into 3 sets: Training, Validation, and Test set (Fig. 1.10), although certain variants are available such as Cross-Validation. A teaching set is used to form a network that calculates loss values by Forward-propagation and updates learning parameters by reverse propagation. A validation system is used to assess the model, adjust hyper-parameters and carry out model selection during the training phase.

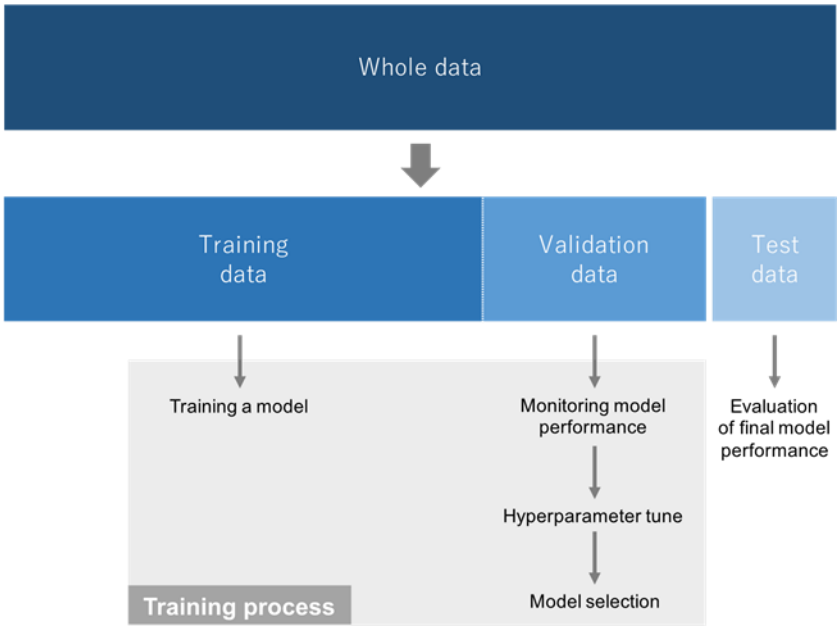


Fig. 1.10: Available information is divided into three groups: a training set, a test set, and a validation set (Source: Yamashita, 2018).

A set of trainings is used for the training of a network in which loss values are calculated via forward propagation. A validation set is used to track model efficiency, fine-tune hyper-parameters and execute model selection during the training phase. At the end of the project, a test set is ideally used for the evaluation of the final model which is finished and selected during training with a training and validation set

Ideally, only once will a sample set be used at the end of the project to assess the efficiency of the final model, which has been finalized and chosen during training processes with practice and validation sets.

1.3.4 Overfitting

Further overfitting refers to the case in which a model acquires a specific statistic of the training set, i.e. memorizes the irrelevant noise rather than acquires the signal and thus performs on a further dataset less effectively. This is one of the biggest challenges in machine learning because an overworked model can't be generalized to unseen information. In this context, as mentioned in the earlier section, a sample set performs a key position in the correct performance evaluation of machine learning designs. The loss and precision of training sets are monitored to recognize overfitting of training data (Fig. 1.11). In comparison with the validation set, the model will probably have been overstated to the training data if it performs well on the training set. Methods to

minimize overfitting have are More Training data, Data Augmentation, Regularization, Batch Normalization, and Reduce Architecture Complexity. More coaching information is the greatest way to reduce overfitting. A model educated on a bigger data set generalizes better, but in medical imaging that is not always possible. Other alternatives include regularization with decomposition or weight loss, normalization of the batch, increase in information and reduction of architectural complexity. Dropout is a new regulating method where random activations are set to 0. This makes the model less susceptible to certain network weights. Dropout is an activating method that has lately been implemented. Decay in weight, also called L2, decreases overfit by penalizing the weights of the model so that weights are reduced to just tiny amounts. Normalizing batch is a sort of additional layer that adaptively normalizes the entry values in the following layer, decreasing the danger of overfitting and enhancing the flow through the networks of gradients, providing greater yields for teaching and decreasing the dependency on initialization. Despite these attempts, the problem remains that the validations are overpowered rather than the practice set due to the data leak during the fine-tuning and model choice hyper-parameter phase. For this reason, it is important to monitor the model generalization by recording the final model output on a distinct (invisible) test collection and, ideally, internal validation datasets as appropriate.

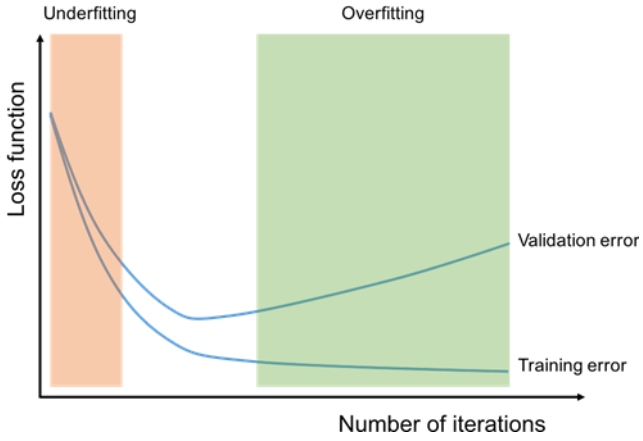


Fig. 1.11: Example of an overfitting and underfitting (Source: Yamashita, 2018).

The model has been overfitted with the training information when the model works well with the workstation in comparison to the validation set. The template is not compatible with information if both instructions and test sets are malfunctioning. The longer a

skilled network works, the quicker it can operate, the more the network adapts to the training data and loses its widely available ability.

1.3 Organization of Thesis

This thesis is organized in six chapters as given below.

In Chapter1, we describe about the classification problem and its architecture. Recently proposed some models/algorithms of classification and from these our focused model CNN is explained briefly and its training methodology is explained in this section that is further improved in chapter4 (proposed work).

In Chapter2, we focused on the review of the literature related with classification, convolutional neural network, character recognition and auxiliary information approach. This illustrates the current and interrelated work of text/image classification systems.

In Chapter3, we describe the datasets used for the implementation. Both the datasets used in this thesis are focused on the handwritten data. The most reliable resource for scientists and learner's alike dataset (MNIST) and the most focused Gurmukhi script dataset.

In Chapter4, we describe our proposed methodology of improving the training part of CNN. Auxiliary information approach is defined in this section that demonstrates improvements.

In Chapter5, the implementing results are described that shows the proposed method consistently better than the convolutional neural network.

In Chapter6, the whole thesis is summarized in a brief manner that describes our motivation of work and achievements in the work.

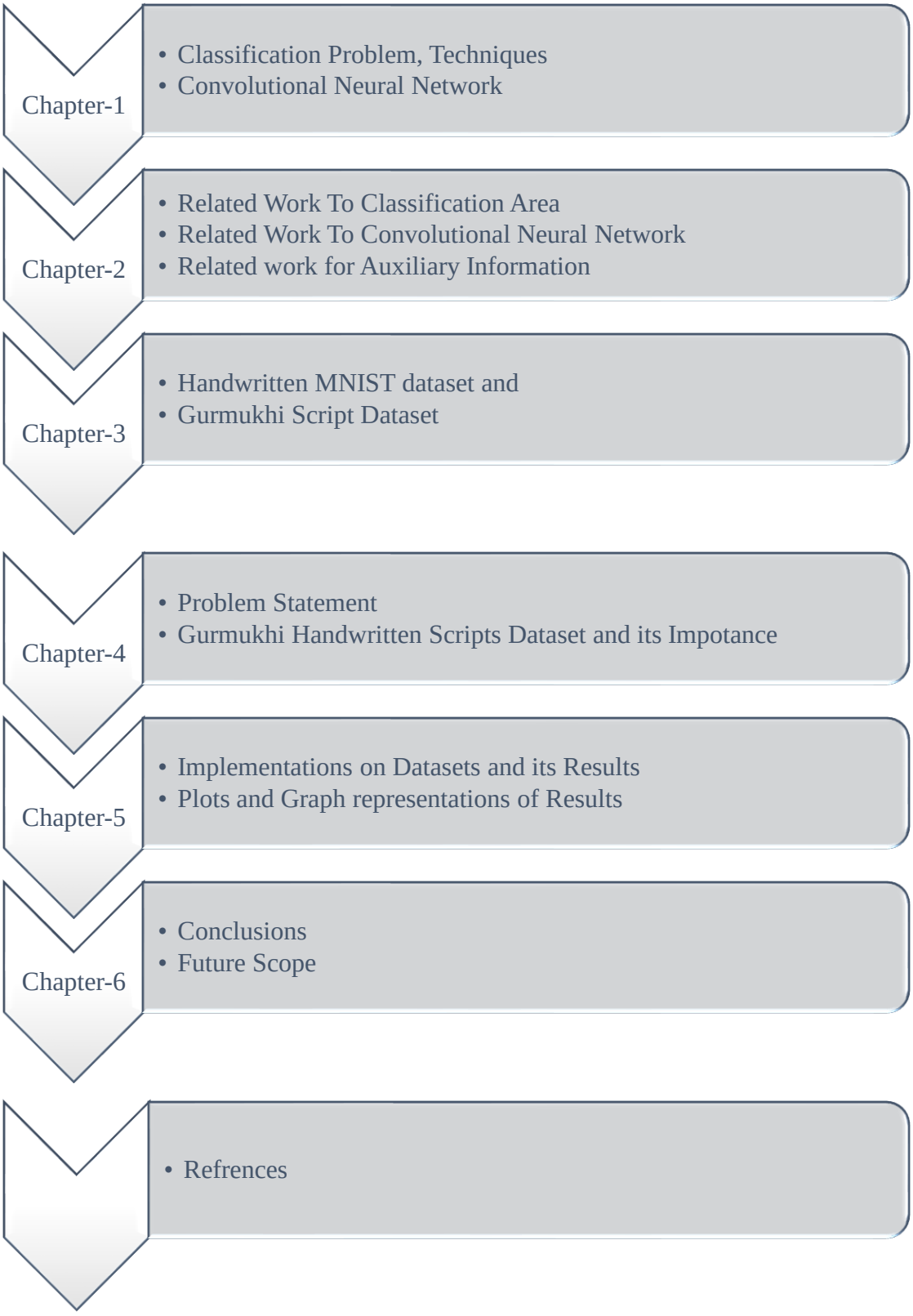


Fig. 1.12: Workflow of thesis

CHAPTER-2

==========

LITERATURE SURVEY

==========

This chapter illustrates the state-of-art of the current and interrelated work of text/image classification systems. This highlights the survey, mechanism, working, benefits and limitations of the related work in the field of text/image classification using their feature analysis from the text/image dataset.

2.1 Work Related with Object-Classification

Hinton and Lennard (1968) discussed about the intestinal transit rate of patients complaining of constipation may be measured simply using radio-opaque markers. Those patients with a normal transit rate do not need chemical or osmotic laxatives, though a bulk laxative may be helpful. Patients with a slow transit rate appear to need regular laxatives. A few patients are seen who conceal their bowel actions. Some young patients have a normal transit rate round the colon but stools accumulate in a large and insensitive rectum. These patients do not require laxatives so much as training and local treatment to help the rectum to empty.

Tan *et al.* (1982) this work provides the revised update of the classification of SLE (systemic lupus erythematosus). This work updates new immunologic knowledge for batter classification of disease. Due to low specificity and sensitivity, earlier work not include Raynaud phenomenon and alopecia. This proposed work criteria were 96% specific and 96% sensitive when tested with systemic lupus erythematosus and control data gathered from 18 clinics. This work gains good sensitive and specificity from earlier developed work.

Japkowicz *et al.* (1995) discussed about the techniques for detecting novelty in the classification that uses a redundancy compression and non-redundancy compression. The approach basically based on training a auto encoder for reconstructing the positive inputs.

Pang *et al.* (2002) discussed the sentimental classification by using machine learning algorithm. For experiments they used sentimental problem like predict weather condition is positive or negative and prediction of market price etc. Already developed machine learning algorithm like Naïve Bayes, entropy classification and support vector machines are not performed on sentimental data or problem. On sentimental data machine learning algorithm gain 50% accuracy but, in this work, gain 58% to 64% accuracy on data statistics. This work performs well from machine learning algorithm when data are human generated.

Yin *et al.* (2003) discussed the classification based on predictive association rules. Recently two classification approach used to know as associative classification and traditional classification. But these two classifications suffer from two problem: one is generating a large number of association rules and another one is may appear over fitting due to rules evaluation. More number of associations may lead overhead. These two approaches are faster but accuracy is not so good. In this work proposed a new method for classification, classification based on predicative association rules *(CPAR).* CPAR combine the benefits of both associative and traditional approach. CPAR used greedy algorithm to generate rule instead of generating large number of rules. CPAR used more rule for test data then traditional association. CPAR use expected accuracy to examine rule and use best k rule for prediction.

Verma and Zisserman (2005) investigates classification of texture from single images obtained under unknown illustration and viewpoint. A texture image is primarily a function of the variables like texture surface, the camera, texture surface albedo, the illumination and the camera viewing position. A statistical learning approach joint distribution of filter responses is developed for the problem of textures. The novelty here is that filters of rationally invariant are used.

Louis *et al.* (2007) work describe about the classification of tumorous of central nervous system. WHO held survey on classification of the central nervous system and listed some entities like angiocentric, papillary glioneuronal tumors *etc.* prediction of the medical related things is predicted through WHO grade? It is component which is used to the response of therapy and outcomes.

Cheng *et al.* (2008) discussed the effective classification for direct dis-criminative patterns. For pattern classification two step approach used, one is mining process and

another one is frequent patterns. Both the process processed by feature selection. But these two processes may be expensive when the problem is large scale with less support. In this work proposed a discriminative approach using the concept of mining approach known as DDPMine. It can handle the efficiency issue which arising from two approach. DDPMine cannot generate complete pattern set. It performs branch and bound searching algorithm on mining discriminative pattern. A feature centered mining concept used to generate pattern. It shows excellent result with DDPMine and get speedup without any type of downgrade of accuracy.

Perronnin *et al.* (2010) use the framework of the Fisher kernel. This study is divided into two parts. In first, several modifications are done over the original framework to boost the accuracy of the FK. In another part, the ImageNet and the Flicker groups are compared and analysed using the classification. The proposed study resulted with a precision rate of 47.9 % to 58.3%.

Deng *et al.* (2010) proposed a study on classifications using 10,000 classes of 9 million images. A category distance based on the WordNet hierarchy is measured with the use of different datasets. The cost functions of classifications produce more informative results.

Sanchez and Perronnin (2011) proposed a compression technique for the classification based on the large-scale images. This study uses the dataset of the ILSVR2010 and Flicker with 1 million images. The results of this study depict that the larger the training set has a high impact on the dimensions of the accuracy.

Ciresan *et al.* (2011) proposed a GPU model using the on-line gradient descent-based CNN. In this study, three datasets are used such as an MNIST dataset of handwritten material, NORB containing dataset of 3d objects and CIFAR dataset based on the natural colour of 32 X 32 pixels. Is used in the model building. The results of the proposed study have an error rate of 0.35% for MNIST, 2.53% for NORB and 19.51% for CIFAR10.

Ciresan *et al.* (2012) use the deep learning neural network based on the classification of the images. It has a dataset of MNIST handwritten images with 35 different columns for the entire MCDNN (Multicolumn Deep Natural Networks). The results of this study are computed using the normalizes the dataset into the 5 different columns. This study

also implemented on the other deep learning model such as CNN, MLP, and MCDNN where each has a different error rate of 0.40%, 0.35%, and 0.23%.

Sermanet *et al.* (2012) worked on classification problem of real-world house number digits using CNN (Convolutional Neural Network or ConvNets). CNN networks structure is inspired biologically. In this the concept of augmentation applied on traditional architecture by learning multi-stage features, by using Lp pooling. They also analyze the benefits of different pooling methods and multi stage features in ConvNets. By establishing a new state of art, achieves 94.85% accuracy on the SVHN dataset with 45.2% error improvement on this.

Krizhevsky *et al.* (2012) trained a deep network for classification of a large-scale dataset that consists 1.2 million high-resolution in the contest of ImageNet LSVRC-2010 over thousand different classes. The neural network (Deep Convolutional neural network), has 60 million parameters and 650,000 neurons that achieved considerably better top-1 and top-5 error rates of 37.5% and 17.0% receptively. The network consists 5 Conv layers some of them are followed by max-polling layers, and 3 fully connected layers with 1000-way Softmax. For reducing "Overfitting", they use data augmentation to artificially enlarge the dataset and a regularization method called Dropout that helps for reducing the test-errors. There models are learned using stochastic gradient descent with batch size of 128, momentum set on 0.9, and weight decay of 0.0005 and Dropout set to zero the output of each hidden neuron with the probability of 0.5. They also attain the second-best entry in ILSVRC-2012 competition by winning top-5 test error rate of 15.3% as compared to 26.2% and shows that a large, deep convolutional neural network results with record-breaking achievement on highly challenging dataset with purely supervised learning.

2.2 Work Related with CNN

LeCun *et al.* (1989) this work discussed about the back-propagation network via the network architecture. Learning network ability can be increased by this work is basically applied on to recognize handwritten code with in Zip file. Up to final classification, start from normalized image a single network used to learn the recognition. In their work all the operation were performed using the back-propagation simulator.

LeCun *et al.* (1990) discussed the back-propagation technique for the hand written digit recognition. Design a highly constrained network architecture for the minimal processing of the data. Inputs are isolated digit or normalized images. For the Zip code digit this work produces the 1% error rate and 9% reject rate. Zip codes are provided by the U.S. postal services. This paper shows the big back propagation network which apply on image recognition problems instead of large, hard pre-processing stage. After performing training, the error rate on 7291 handwritten and 2549 printed digits was 1.1% and MSE was 0.017.

LeCun and Bengio (1995) discussed the convolution network used for speech, images and for time series. In this designed model of recognition, the hand modify feature collect the needed information from input and remove irrelevant information. After that the collected feature are divided into classes. In this work a fully-oriented network model used for classifier.

Glorot and Bengio (2010) discussed a new scheme which describe the performance of random initialization with respect to deep neural network. Design a new algorithm which tells why standard gradient perform decent from random initialization with respect to neural network. They first study the impact of the nonlinear activation function to design new algorithm. After study they observe logistic activation is not suited with random initialization for deep network due to its mean. Mean value derives from hidden layer into saturation. During the training of the neural network these saturated value units move out from saturation. To reduce this drawback, design a new algorithm which study how gradients and activation increase across layer. Doing this a new initialization is convergence.

LeCun *et al.* (2010) this work discussed about convolutional network and application of its in-vision domain. To understand the task such as audio perception, language understanding and visual input require a good representation of perception. To learn these, feature convolutional networks (ConvNets) used which is a biologically trainable architecture. In this network different layer are work like filter bank, non-linearity's and feature layers. Respect to multiple stage, ConvNets learns hierarchies of feature. They design a new algorithm which require a smaller number of labelled samples to train ConvNets.

Howard (2013) discussed a new technique which add improvement on current existing convolutional neural network which is based on image classification. In this technique, they work on more images transformations to train the data and to generate prediction at test time data set use more image transformation. They applied algorithm on high visual or resolution images. This achieve the error rate 13.55% on top 5 classification. This method improves the error rate of 11.7%.

Goodfellow *et al.* (2014) design a new concept which address a sub problem for discriminate multi digit number. Approach to solve this problem is completely different from segmentation localization and recognition steps. In this work they use an approach which is integrates these three steps using the convolutional network that work or operates on pixel of the images. They use the implementation concept from (Dean *et al.* 2012) to train the neural network on high visual quality images. With the depth of convolutional network, performance of this work is increased. They use SVHN dataset to recognize digit number and gain 96% accuracy. And they perform experiments on eleven datasets created from street imagery contain millions of data and gain 90% accuracy. This scheme is also implemented on CAPTCHA service and gain a superior 99.8% accuracy. After summarize the results, this scheme best to apply on CAPTCHA or street number recognition.

Hsu Chih-Wei *et al.* (2014) discussed about the classification of support vector and also discussed about those that are not familiar with SVM. In this paper, they outline an approach called "cookbook" that give ordinarily the reasonable results. They purpose a recipe that is given to SVM novices for obtaining the results that are acceptable. The model they choose is RBF Kernel. It concludes that if thousands of attributes are present then subset is chosen before giving any data to SVM.

Oquab *et al.* (2014) this work discussed about the convolutional neural network model of learning mid-level image representation. In this work design a model which reuse layers which trained on ImageNet dataset to calculate the mid-level representation of image in PASCAL VOC dataset. When using the only 12% of data it's gives a good result on the PASCAL VOC dataset. It is the improvements of the all type of target classes.

Zeiler and Fergus (2014) introduced a new visualization technique that gives vision to intermediate layers and operation of the classifier. They also do analysis on different

model to search the performance. They work on ImageNet data set and generalizes that it's a good dataset than others. This work outperforms the Krizhevske *et al* (2012) model. In this work they obtain 14.8 % test error when combine multiple number models and get improvements of 1.6 %. For the conclusion this model is less good to the PASCAL dataset.

Kim (2014) discussed the new model of convolutional neural network for trained vectors. Previously developed CNN model trained vectors for sentence level classification. In this paper develop a CNN model with less hyper parameter for vectors which gain batter result on multiple benchmarks. It's a simple architecture which allow both task specific and vectors. This model gives excellent output for five tasks out of seven tasks including of question classification and sentiment analysis. Results of this method are good against other existing method.

Lin *et al.* (2014) this work discussed a new deep network name as Network in Network to increase the model inequality. To examine the input, used linear filter followed by non-linear filter in the convolutional model. In this work a micro neural network model developed with multi-linear input. Staking is used for implement the NIN model. This work can improve the performance of CIFAR-10 and CIFAR-100, and can also enhanced performance on SVHN and MNIST datasets.

Simonyan and Zisserman (2015) this work determines the convolutional network depth effect on the accuracy of large-scale image recognition. They increase the depth using 3 × 3 very small convolution filters. They made a best ConvNets performing model that will be used in future to improve the deep visual in computer vision. Claim that this model gives well result then other model in convolutional neural network. For test use the seven model which give 7.30% test error. After submission, decreased the error 6.8% using ensemble of 2 model.

Zhang *et al.* (2015) discussed the convolutional networks of character level for text-classification. In this work different size of dataset of character used. In this way it gains good result. Comparison are showing between many models like traditional model such as bag of words, n-grams and its variants deep learning model like ConvNets and neural network. In this scheme experiments done on both of model one is traditional model and learning model.

2.3 Work Related with Character Recognition

Brown *et al.* (1983) utilized component vectors and an evaluation of the word length to represent to word attributes. The deference is taking into account the extricated component vectors utilizing the k-closest neighbors system. The recognizer was prepared on information of one client and tried on information of another two. Acknowledgment rate ran from 64.5 % to 81.3%.

Kurtzberg (1987) developed a system to perceive unconstrained handwritten discrete images in view of elastic coordinating against an arrangement of models produced by individual authors. He likewise presented component investigation with versatile matching to wipe out far-fetched prototypes.

Noubound and Plamondon (1991) utilized basic way to deal with perceive online handwritten characters. They introduced an ongoing requirement free hand printed character recognition framework in light of a basic methodologies. A chain code is extricated, after the preprocessing operation to speak to the characters. The order is in light of the utilization of processor devoted to examine the string.

Veltman and Prasad (1994) utilized HMM to segregated on-line transcribed character and accomplished a normal error rate of 7.8 % over completely un-constrained letter set comprising of the Lowercase English letter set.

Powalka *et al.* (1994) added to a word-based recognizer which utilized an exceptionally constrained arrangement of components comprising of a grouping of ascenders and descenders and a guess of the length of the word. A fuzzy logic-based coordinating method is utilized. Acknowledgment rates got for a 250 dictionary word data was 60.6 %. Script writing of 18 clients was assessed, every written work 200 words.

Bontempi and Marcelli (1994) introduced a strategy taking into account a genetic method algorithm utilizes the engine of a learning framework to create model of the string characters match to execution the order. Learning procedure, gave by a genetic based algorithm, permitted the framework to have both an author autonomous center and an adjustment plan to nicely tune the recognizer to the user's style. The framework was performed more than 15 subjects, distinctive from the ones to get the preparation set. An acknowledgment rate of 84.4 % was gotten, with a rejection rate of 15.7 % and

a mistake rate of 0.9% after adjusting author, the framework had the capacity display a recognition rate of 95.8% on that author.

Dunea and Dorizzi (1994) introduced a framework devoted to the acknowledgment of English current hand writing words drawn on a digitizing tablet. This framework utilizes an analytical perspective as a part of the sense that its tries to limit the word letters to be perceived. The framework in a mono-scripter connection has acknowledgment rates min-normal-max of 80.8 % - 94.2 % - 96.6 % individually, also, is the situation of multi scripter connection rate was 93.4 % for 15432 models and 91.4 % for 5625 models.

Li and Yeung (1997) introduced a way to deal with online manually written alphanumeric character acknowledgment in light of successive penmanship signals. By applying this scheme, an online manually written character of word is characterized by a succession of predominant focuses in strokes and a grouping of composing bearings between back to back predominant focused.

Cho *et al.* (1997) displayed the three advanced neural network classifiers (NNC) to understand complicated pattern recognition issues that incorporate different multi-layer classifier, HMMs perceptron cross breed classifier and framework versatile self-sorting out guide classifier. This proposed work was identified with unconstrained written by hand numerals.

Kimura *et al.* (1997) introduced a two-stage various leveled framework comprising of a statistical pattern acknowledgment module and simulated neural network to perceive a substantial number of classifications including comparable classification sets. This work was identified with hand written Kanji characters. The right acknowledgment rates of the samples are 98.09 % and 97.59 % for the initial information and separately the test information's.

Wakahara and Odaka (1997) introduced a far-off tolerant stroke coordinating strategy that employments stroke-based relative change. They introduced exploratory results for kanji characters and acquired acknowledgment rate of 97.4 % in the event of information openly written in square style and 96 % for test information written in quick and handwritten penmanship style.

Chan and Yeung (1999) introduced a basic methodology for perceiving online penmanship. Their methodology accomplished sensible rate, genuinely high precision and adequate resistance to varieties. The acknowledgment rates are 97.60 % for the digits, 98.90 % for uppercase letters, 97.34 % for lowercase letters, and 98.40 % for the consolidated sets. At the point when the rejected cases are rejected from the computation, the rates can be expanded to 99.90 %, 99.53 %, 98.50 % and 98.27 %, separately.

Spitz (1999) dealt with shape-based word acknowledgment. The procedure depends on the change of content pictures into shape codes of character, and on exceptional lexical that contains data on the state of words. Ambiguity is diminished by layout coordinating utilizing models got from surrounding text, exploiting the nearby consistency of textual style, face and size and in addition picture quality. The work tells the impacts of lexical substance, structure and preparing on the execution of a word engine.

Zhou *et al.* (1999) portrayed another sort of neural network quantum neural system further more displayed its application to the acknowledgment of written by hand numerals. Quantum neural system consolidated the upside of neural demonstrating and fuzzy theoretic standard. It proposes an effective fusion system and achieves a high quality of 99.20 percent.

Hu *et al.* (2000) utilized HMM for essayist independent online penmanship acknowledgment framework utilizing blend of point situated and stroke oriented components. Mistake rates for the character acknowledgment test outcomes are precise to inside under 1 % with 94% certainty.

Brakensiek *et al.* (2002) portray an essayist autonomous online handwriting recognition system which is looking at the viability of a few confidence measures. Their acknowledgment frame work for separate German words is in view of HMMs utilizing a word reference. They analyze the proportion of rejected words to misrecognized words utilizing four distinctive certainty measure. They utilize a largest online handwriting database of a few authors comprises of cursive script tests of 156 distinct authors. The preparation of the essayist independent framework is performed utilizing 3440 words of 140 writers. Testing is done with 2081 expressions of 32 diverse scholars. The acknowledgment results are resolved utilizing an expanding edge τ and utilizing the gauge framework without dismissal a word acknowledgment rate of 86.0%

(1901 words are perceived effectively) is accomplished trying the whole test-arrangement of 2171 words. The displayed results allude to a solitary word acknowledgment rate utilizing a lexicon of 2200 words.

Sharma *et al.* (2008) talked about a procedure that perceives characters in two stages. Initially arrange perceives the strokes, in 2nd stage, character is assessed on the basis of perceived strokes. For 60 essayists and an arrangement of 41 Gurmukhi characters, they have achieved acknowledgment as 90.08 % accuracy.

Sharma *et al.* (2009) introduce another stride as adjustment of perceived strokes in online handwriting recognition system. The improvement of perceived strokes incorporates: strokes recognizable proof as needy and significant ward strokes; the adjustment of strokes as for their positions; the blend of strokes to perceive character. They have accomplished a general acknowledgment rate as 81.02 % in online manually written cursive penmanship for an arrangement of 2576 Gurmukhi word reference words.

Singh *et al.* (2012) discussed Gurmukhi character recognition for separated characters is proposed. Gabor Filter-based extraction method is used. The database consists of 200 patterns of every of 35 characters of the Punjabi script taken from distinct writer. These samples are first preprocessed and normalized to 30×30 sizes. The best accuracy gained is 95.29 % as five-fold cross validation of entire database with SVM (support vector machine) classifier having RBF kernel.

2.4 Work Related with Auxiliary Information Approach

Srivastava (1967) introduces the sample survey uses for making the use of an auxiliary variable that increases the accuracy of the estimators. In this author defines three estimators, *i. e.*, ratio estimator, product estimator and mean per unit. Also define the significant use of the product estimator, that when correlation coefficient is negative between the two characters. In this suggested estimator from the three, are the ratio estimator.

Srivastava and Jhajj (1981) defined a class of estimator, as ratio function of sample mean to the mean of population and the ratio of sample variance to auxiliary variable for the population variance. For the mean squared error and for bias, asymptotic

expressions are obtained. In this, they extend the class of estimators to ones depend also upon the sample variance ratio to population variance of the auxiliary variable. They show the results could be extended when information on more than one auxiliary variable is available.

Kadilar and Cingi (2006) proposed an estimator using an Auxiliary variable for the population variance in simple random sampling. They show that estimator proposed by them are more efficient than regression estimators and the traditional ratio and they obtain the mean square error equation of their estimator. They additionally support the results with numeric illustration. They work with the suggestion of Isaki that presents ratio estimator using Auxiliary information for the population variance.

CHAPTER-3

==========

DATASET DESCRIPTION

==========

In this Chapter, we explain the description of both the datasets (MNIST and Gurmukhi script dataset). In section 3.1, we discussed briefly about handwritten digits (from 0 to 9), ten classes and each class contain some data from the whole dataset used that is of 42,000. This dataset is divided for training purpose as discussed in Chapter 1. Also, in section 3.2, we discussed brief introductory of handwritten Gurmukhi Script that contains 79 classes and in this each class have 100 images.

3.1 MNIST dataset

MNIST is the de facto "Hello World" computer vision data set ("Modified National Institute of Standards and Technology"). This classic handwritten image dataset has served as the basis for benchmarking classification algorithms since its launch in 1999. MNIST continues a reliable resource for scientists and learners alike as fresh machine learning methods emerge.

Table 3.1: Handwritten digits of 10 Classes (MNIST)

				
Zero	One	Two	Three	Four
				
Five	Six	Seven	Eight	Nine

The MNIST database was built from the original NIST database; hence, modified NIST or MNIST. In our implementation we use 42,000 images (some of these are used for

training purpose images and some of these can also be used for cross-validation purposes) both the datasets are drawn for same distribution. All of these black and white handwritten digits are normalized in size, and also centered in a fixed-size image, where its intensity lies at the center of the given image with the dimensions of 28 × 28 pixels. Thus, at each image dimensionality, sample vector is of 28 * 28 = 784, whereas each element of this is binary. This is quite simple database from others for the user who want to experiment on pattern-recognition methods and also for the machine-learning techniques on data of real-world while efforts spending on preprocessing are minimal and also for formatting spend minimal. By the use of the references that are provided on the official Web site, innovative students and research educators of machine learning can have benefit from a set of machine learning literature.

3.2 Gurmukhi Handwritten Script

This script was formulated during the 16th century by Guru Nanak Dev Ji, the first Sikh Guru, and was promoted by Guru Angad Dev Ji, the second Sikh Guru. Gurmukhi is the most well-known script utilized for composing the Punjabi dialect in India. The name Gurmukhi is gotten from the Old Punjabi term "Gurumukhi", signifying "from the mouth of the Guru". There are 40 letters in Gurmukhi. A large portion of the characters are with an even line above them. A vertical line is utilized to demonstrate the end of a sentence. Two vertical bars demonstrate a longer delay between sentences or sections. The Gurmukhi script contains 35 letters. Out of these three letters are different in Gurmukhi script as they build the basis of vowel and are not consonants. Table 3.2 shows these Gurmukhi characters and for our implementations we use the 79 classes as we augment these letters.

Table 3.2: Gurmukhi Handwritten Scripts using 79 Classes

(Source: Kumar, Munish & K. Sharma, R & Jindal, M. 2013)

							
141	**142**	**143**	**144**	**145**	**146**	**147**	**148**

							
149	**150**	**151**	**152**	**153**	**154**	**155**	**156**
							
157	**158**	**159**	**160**	**161**	**162**	**164**	**165**
							
166	**167**	**168**	**169**	**170**	**171**	**172**	**173**
							
174	**175**	**176**	**177**	**179**	**180**	**181**	**182**
							
183	**184**	**185**	**186**	**187**	**188**	**189**	**190**
191	**192**	**193**	**194**	**195**	**196**	**197**	**198**
200	**201**	**202**	**203**	**204**	**205**	**206**	**207**
208	**209**	**210**	**211**	**212**	**214**	**215**	**216**

							
217	**218**	**222**	**223**	**224**	**225**	**351**	

In this dataset we collect 100 different user handwritten scripts of 79 different classes.

For the training purpose we divide the whole datasets into two different partitions that is Training dataset and Validation dataset. In Chapter 5, for implementing the training process we use different dataset size like for MNIST dataset we use 6K images, 8K images, 10K images for different experiments. The Gurmukhi script dataset also partitioned in 15*79, 21*79, 27*79 images for implementing both, CNN and Proposed CNN. In the next chapter we discussed about the improvements in the training methodology of CNN by using Auxiliary Information approach.

CHAPTER-4

==========

IMPROVING THE TRAINING METHODOLOGY OF CNOVOLUTIONAL NEURAL NETWORK

==========

In this chapter, we have worked on the improvement in the training methodology of CNN classifier by redefining the Softmax function used in network for turns the logits into the probabilities with the sum of one. As we know in recent the researchers are very keen to develop the classification models that are 100% accurate on the unseen dataset. We discussed about the Auxiliary Information approach that directs our results into the improvement phase in the accuracy on the unseen data (Proposed Model).

4.1 Problem Statement

In machine learning applications, auxiliary information is often accessible. For instance, information may have been collected in distinct nations in medical apps or with somewhat distinct class label definitions. Data may have been collected in financial analysis in previous years or with mildly distinct definitions of characteristics (e.g. changes over moment in the meanings of "efficiency" and "customer cost index"). A challenge for machine learning is to discover methods to use this information to enhance the target classification job efficiency.

In this research, we suggest and research a profound education structure where the issue of classification learning is based both on the label and on a probability evaluation of the trust or faith in this label. We first discover how to alter one of the fundamental algorithms for learning (the Convolutional Neural Network) readily in order to acknowledge probabilistic evaluations and discover a performance classification with fewer instances.

A bias / variance analysis allows users to understand the utility of auxiliary data. Since the actual learning figures are sparse there is elevated variance and therefore heavy mistake among a trained classifier. The addition of additional information may decrease this variance, but may boost the partition because the additional data are taken from a

distinct range than the actual information. This assessment indicates that the use of supplementary information should be reduced as the quantity of actual training data rises.

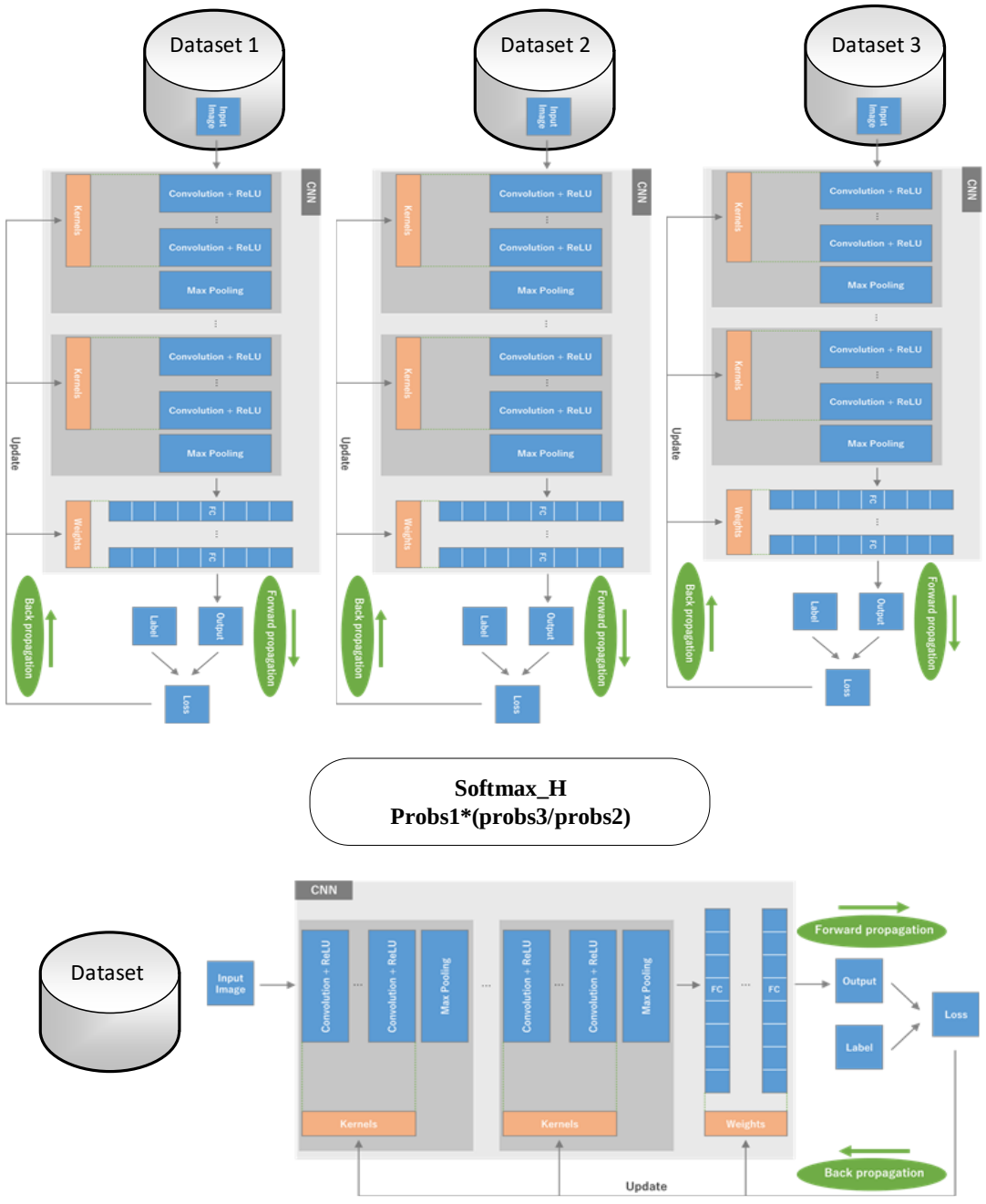


Fig. 4.1: Workflow of proposed technique

Because the fresh system relies heavily on the exactness of the subjective probabilistic assessments, the measuring inconsistencies and noise may become vulnerable. To tackle the issue, we are proposing a new approach centered on the CNN and that is on the Softmax function, which is redefined in our proposed method to enhance the

classification with lower instances. In addition to incorporating learned probabilities transmitted to fresh classifier, we show the value of our technique in a variety of Handwritten Data Sets (such as MNIST, Gurmukhi Script).

4.2 Methodology

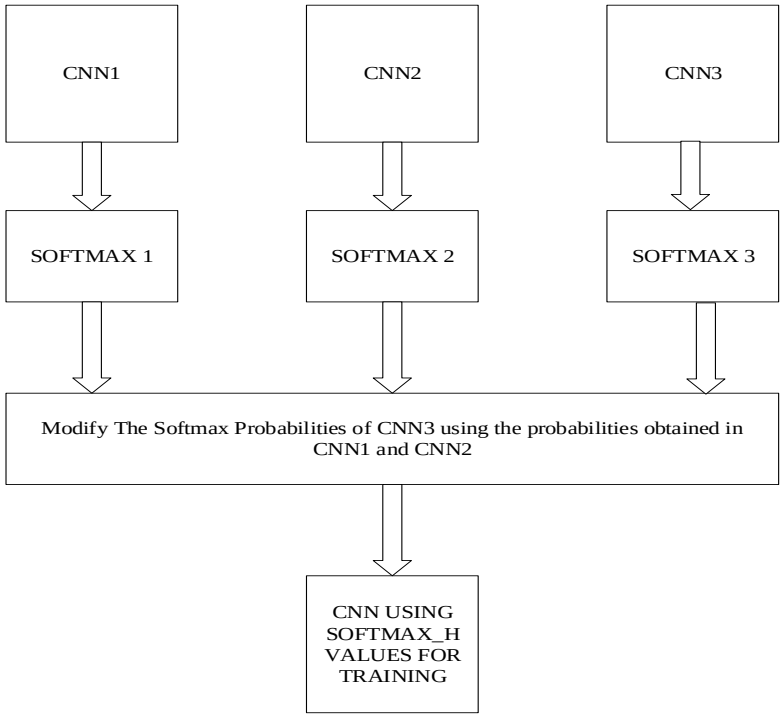


Fig. 4.2: CNN-based classifier using Auxiliary Information Approach

1. Collect Input Data for all Convolutional Neural Networks

2. Train the dataset using our approach in which CNN train for each and every image in the dataset, with the sequence in which firstly from CNN1 Softmax values are calculated for a particular image then from CNN2 and then from CNN3. After that we transfer its probabilities and with the use the auxiliary information approach i.e. improves the learning process of Softmax values, computes by CNN1 will be multiplied with computed values of Softmax from CNN3 and their results are divided by computed Softmax values of CNN2, we proposed as function renamed as Softmax_H.

3. After this stored results in proposed Softmax_H function transfers to the main CNN which we want to improve.

4. This function calls recursively in training process of CNN that enhances the results.

4.3 Experimental Setup

In the experiments, we use two datasets i.e. Handwritten MNIST dataset and Gurmukhi Scripts datasets. We divide the whole dataset in three parts. Each division consists of different images and with different number of images for describing the value of data and its evaluation both.

In these Experiments, we describe two main notations called CNN and CNNA, where CNN shows the Convolution Neural Network without Auxiliary information and CNNA shows CNN with Auxiliary Information Approach for transferring the Softmax values of pretrained Models to train the model.

We experiment on the basis of data size of the dataset, Batch size of the model, and on different models i.e. proposed and simple.

We use Python3 platform to do these experiments on CPU, with some packages like NumPy, pickle, tqdm, Matplotlib etc.

CHAPTER-5

==========

IMPLEMENTATION AND RESULTS

==========

In this Chapter, the results obtained by using the classification models extracted from data as their input are presented. We use the proposed method of training to improve the results of the CNN by using the auxiliary information approach as explained in Chapter 4. In this chapter, we present and discuss both the results attained by using CNN approach in which we use the four-layer model explained in Chapter 1 and the CNNA (CNN with Auxiliary Information) approach. There are various parameters used for comparing the results attained by our approach such as dataset size, batch size and on terminating conditions.

The datasets used for attaining the results are explained in Chapter 3. For the broader view, the plots are also shown in this chapter, for each and every experiment we show the plot of reducing loss at every iteration and also show the accuracy of each class in the dataset.

TABLE 5.1: Accuracy on Handwritten MNIST dataset using Conventional (CNN) and Proposed CNN (CNNA)

METHOD	TRAINING DATA	VALIDATION DATA	TERMINATE CONDITION		NO. OF LAYERS	ACCURACY		
			Epoch	Loss<=0.001		CNN1	CNN2	CNN3
BATCH SIZE = 32, LEARNING RATE = 0.01								
CNN	6K	4K	2	-	4	96.38	95.78	94.77
CNNA	6K	4K	2	-	16	96.60	96.45	96.25
CNN	8K	4K	2	-	4	97.08	95.78	94.23
CNNA	8K	4K	2	-	16	97.90	97.10	97.15
CNN	10k	4K	2	-	4	96.30	96.97	97.40
CNNA	10K	4K	2	-	16	97.17	97.32	97.50

BATCH SIZE = 64, LEARNING RATE = 0.01								
CNN	6k	4K	50	0.0001	4	97.10	97.25	97.42
CNNA	6K	4K	50	0.0001	16	97.50	97.97	97.60
CNN	8K	4K	50	0.0001	4	98.22	98.65	98.10
CNNA	8K	4K	50	0.0001	16	98.32	98.88	98.42
CNN	10K	4K	50	0.0001	4	97.15	97.25	98.10
CNNA	10K	4K	50	0.0001	16	97.90	97.85	98.54

TABLE 5.2: Accuracy on Handwritten Gurmukhi Script dataset using Conventional (CNN) and Proposed CNN (CNNA)

METHOD	TRAINING DATA	VALIDATION DATA	TERMINATE CONDITION		NO. OF LAYERS	ACCURACY		
			Epoch	Loss < = 0.0001		CNN1	CNN2	CNN3
BATCH SIZE = 128, LEARNING RATE = 0.01								
CNN	15*79	19*79	50	0.0001	4	68.49	68.42	68.35
CNNA	15*79	19*79	50	0.0001	16	70.82	70.09	70.75
CNN	21*79	19*79	50	0.0001	4	78.28	76.35	75.82
CNNA	21*79	19*79	50	0.0001	16	79.28	79.75	76.22
CNN	27*79	19*79	50	0.0001	4	78.15	78.28	83.94
CNNA	27*79	19*79	50	0.0001	16	79.08	78.75	84.01

Fig. 5.1: Graphical representations of CNN results for 6k (MNIST)

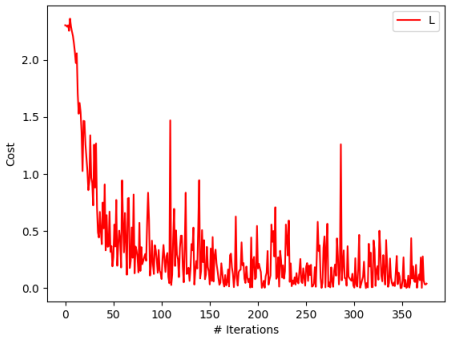
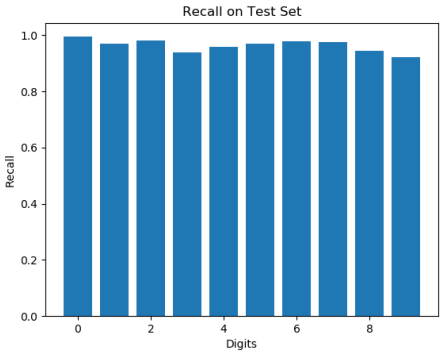
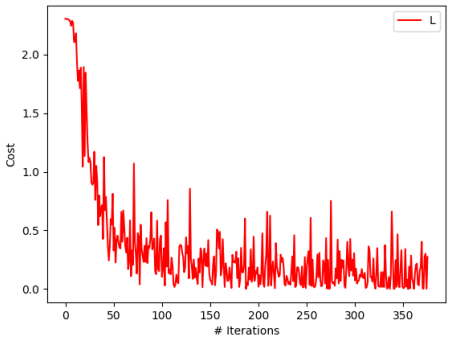
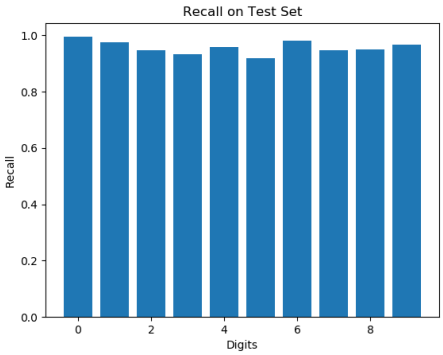
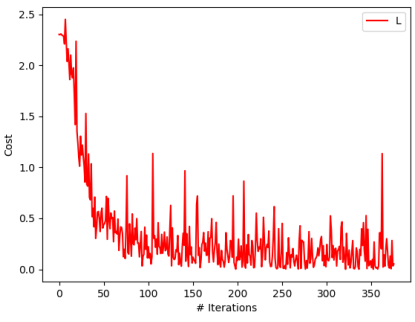
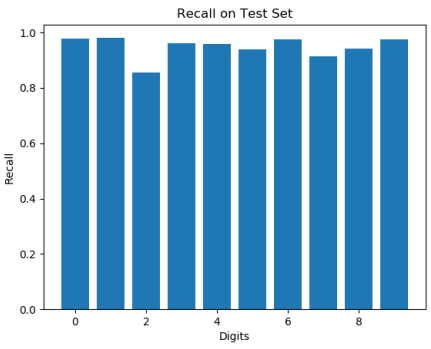
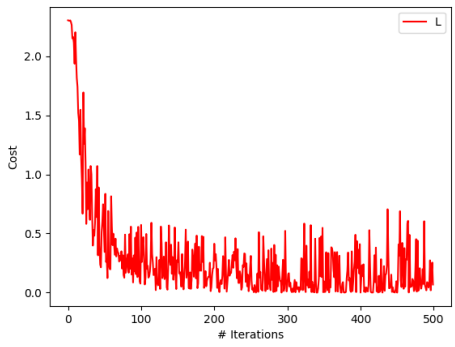
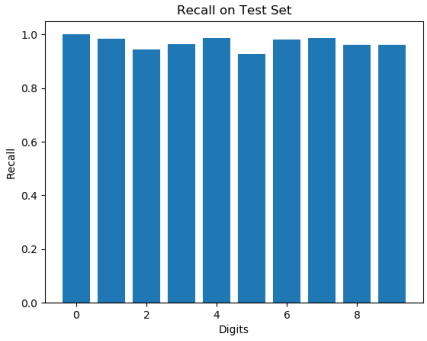
	
CNN1 Cost	**CNN1 Training Accuracy**
	
CNN2 Cost	**CNN2 Training Accuracy**
	
CNN3 Cost	**CNN3 Training Accuracy**

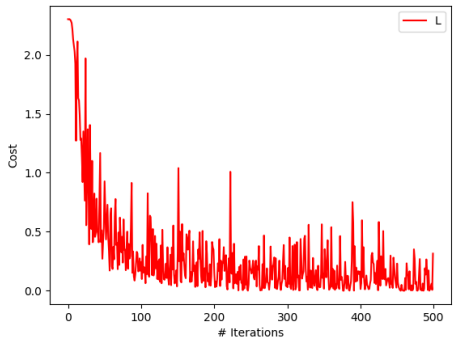
Fig. 5.2: Graphical representations of CNNA results for 6k (MNIST)



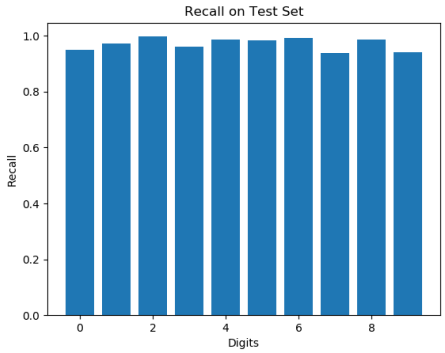
CNN1 Cost



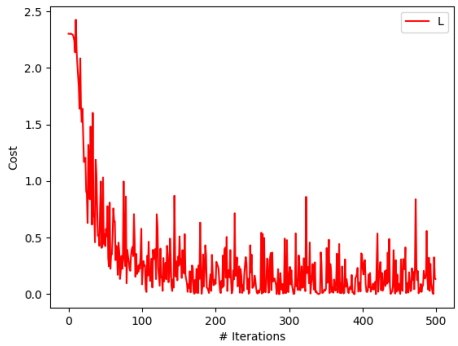
CNN1 Training Accuracy



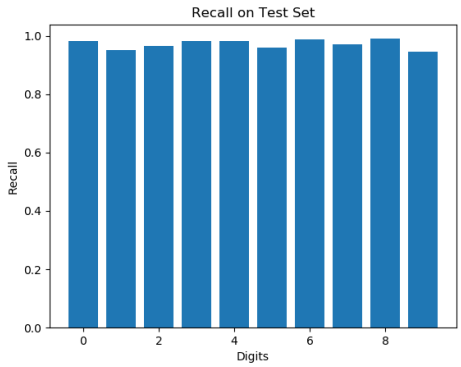
CNN2 Cost



CNN2 Training Accuracy

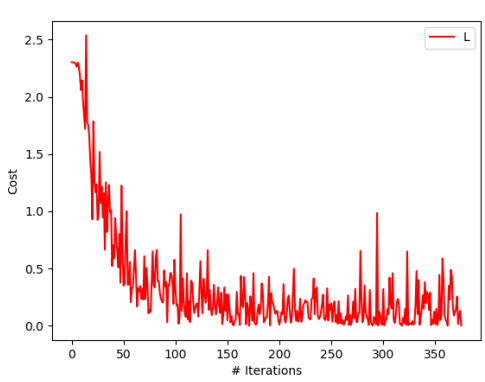


CNN3 Cost

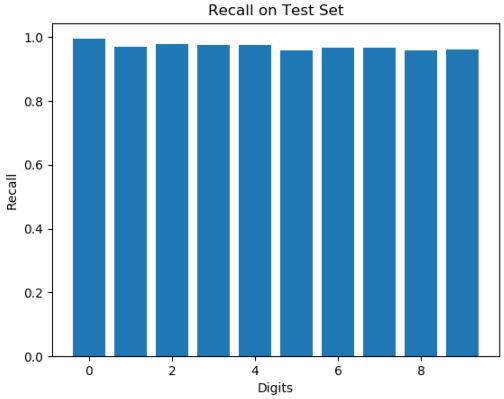


CNN3 Training Accuracy

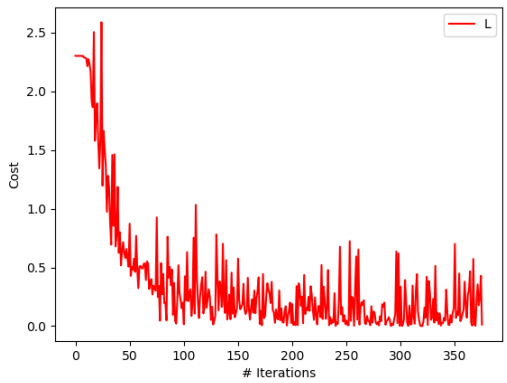
Fig. 5.3: Graphical representations of CNN results for 8k (MNIST)



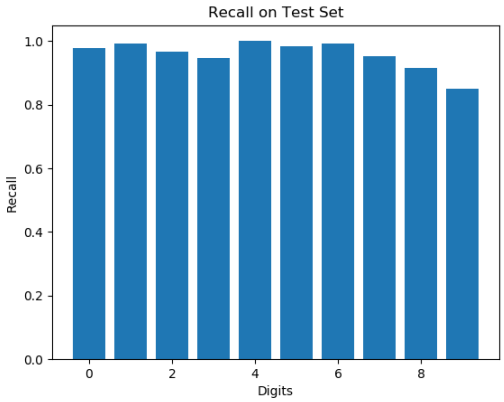
CNN1 Cost



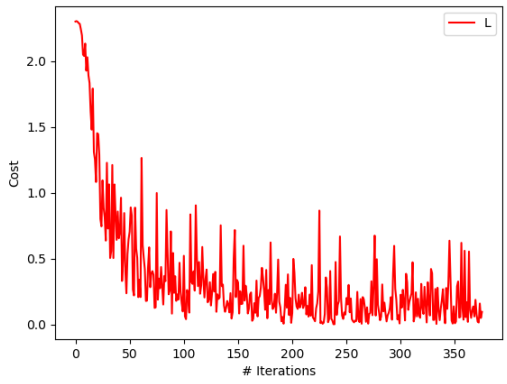
CNN1 Training Accuracy



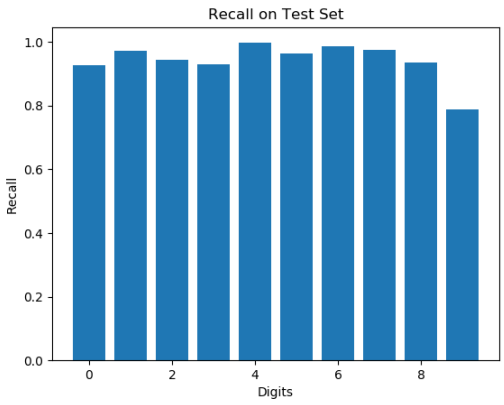
CNN2 Cost



CNN2 Training Accuracy

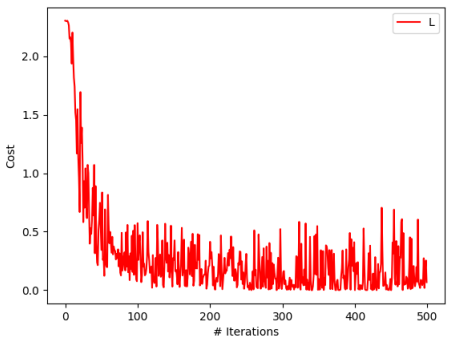


CNN3 Cost

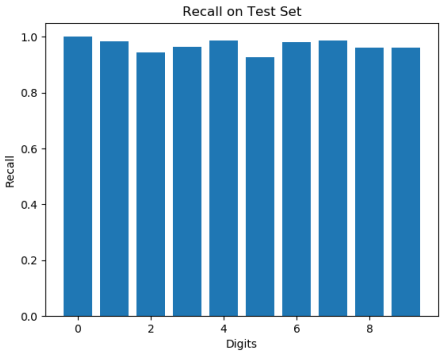


CNN3 Training Accuracy

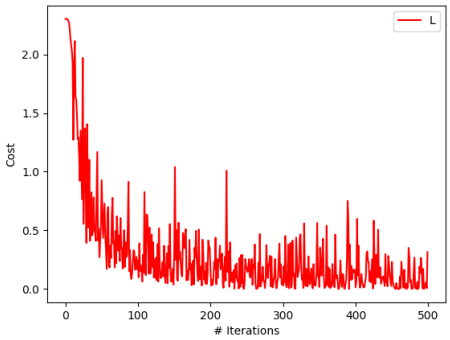
Fig. 5.4: Graphical representations of CNNA results for 8k (MNIST)



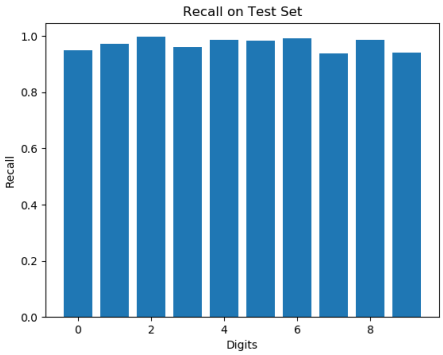
CNN1 Cost



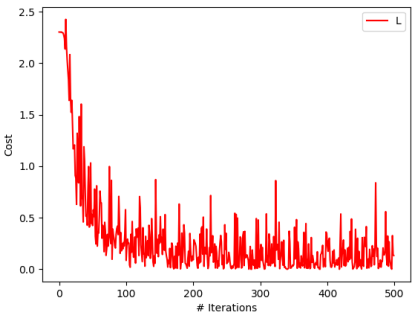
CNN1 Training Accuracy



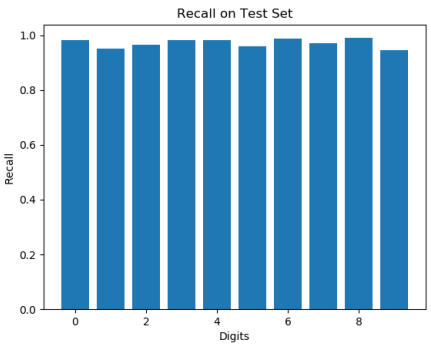
CNN2 Cost



CNN2 Training Accuracy

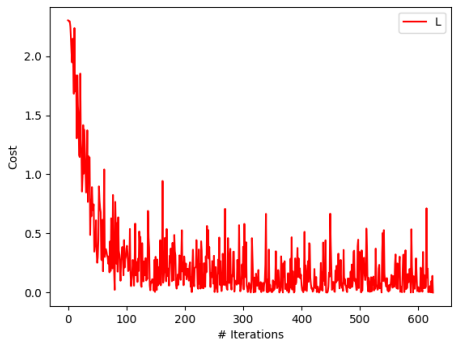


CNN3 Cost

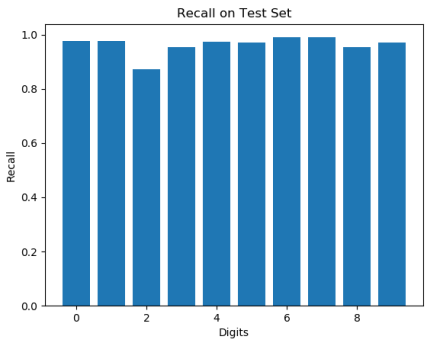


CNN3 Training Accuracy

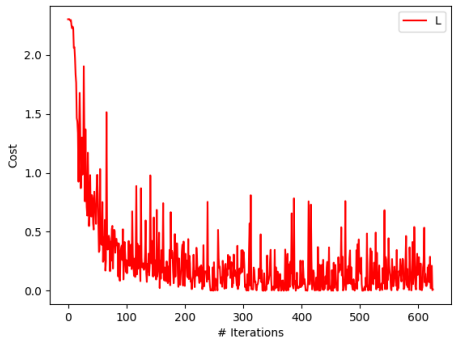
Fig. 5.5: Graphical representations of CNN results for 10k (MNIST)



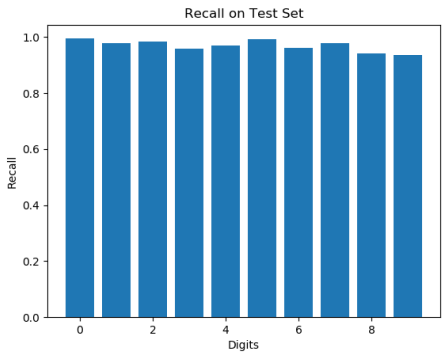
CNN1 Cost



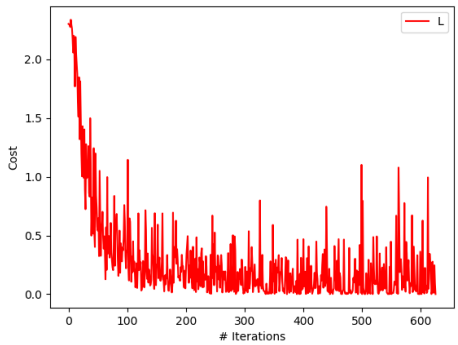
CNN1 Training Accuracy



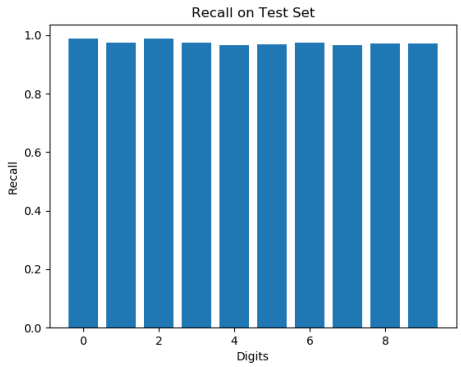
CNN2 Cost



CNN2 Training Accuracy

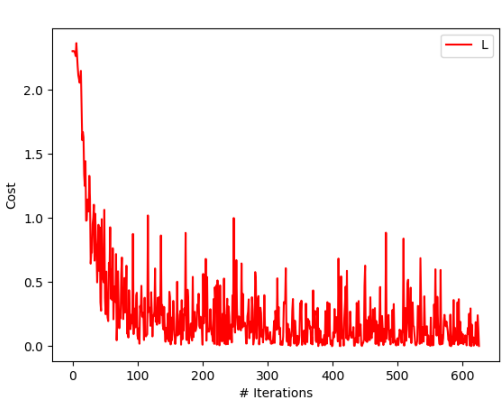


CNN3 Cost



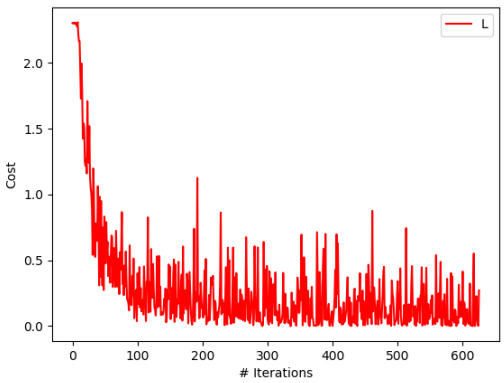
CNN3 Training Accuracy

Fig. 5.6: Graphical representations of CNNA results for 10k (MNIST)

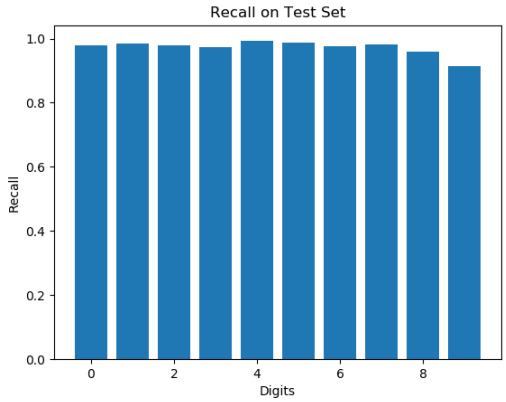


CNN1 Cost

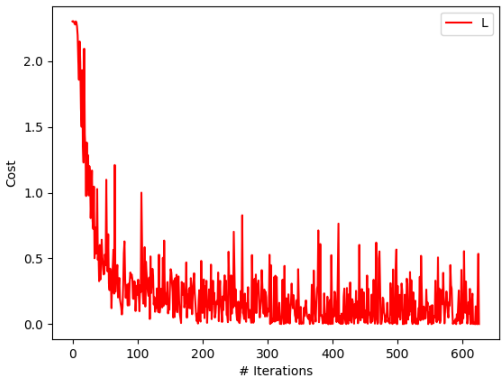
CNN1 Training Accuracy



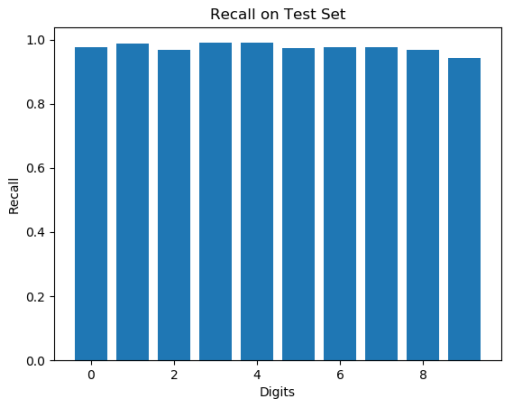
CNN2 Cost



CNN2 Training Accuracy

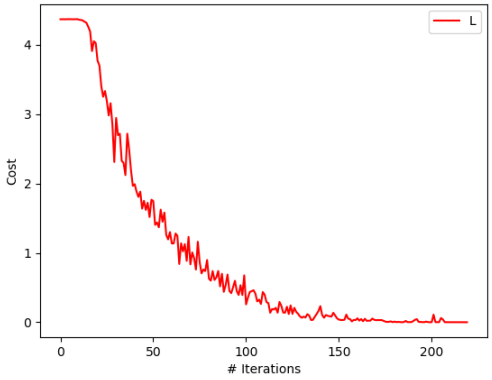


CNN3 Cost



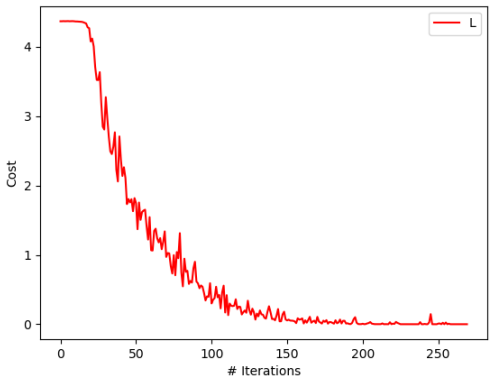
CNN3 Training Accuracy

Fig. 5.7: Graphical representations of CNN results for 15*79 (Gurmukhi script dataset)

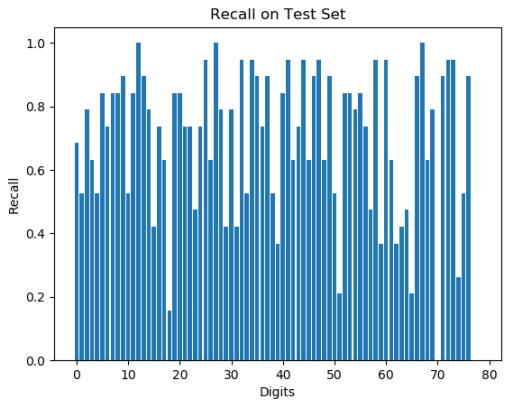


CNN1 Cost

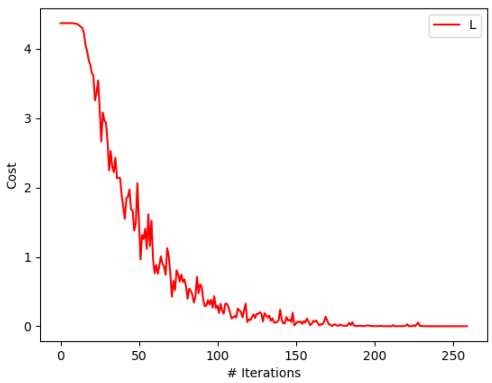
CNN1 Training Accuracy



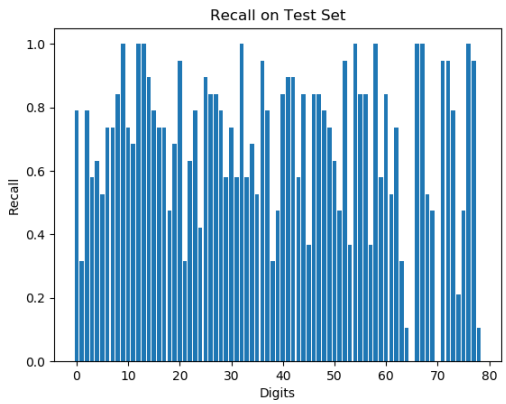
CNN2 Cost



CNN2 Training Accuracy

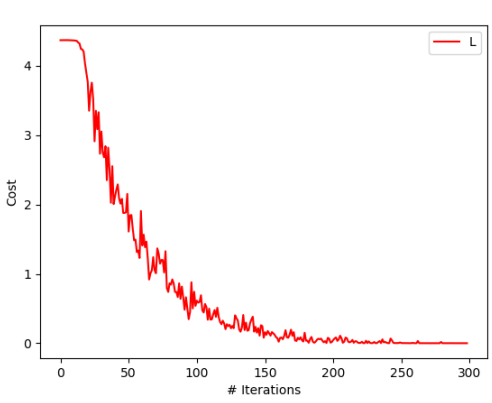


CNN3 Cost

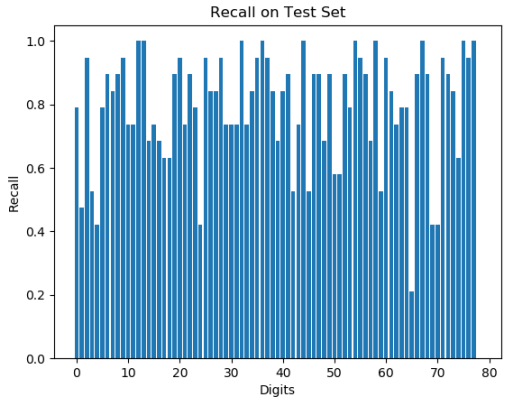


CNN3 Training Accuracy

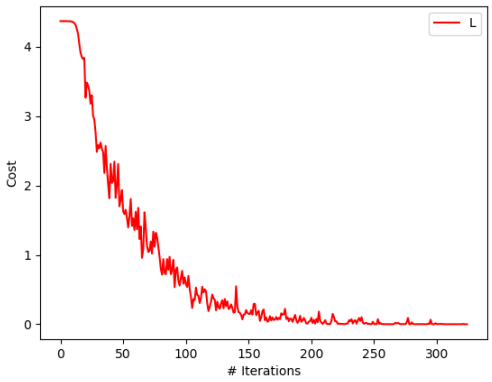
Fig. 5.8: Graphical representations of CNN results for 21*79 (Gurmukhi script dataset)



CNN1 Cost

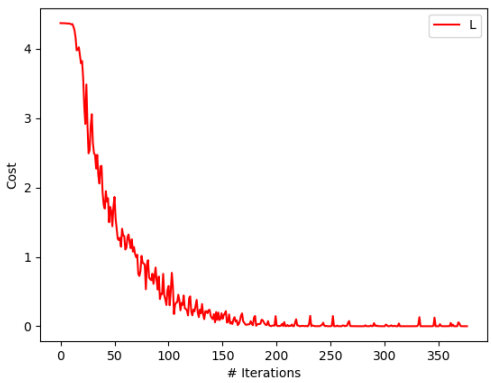


CNN1 Training Accuracy

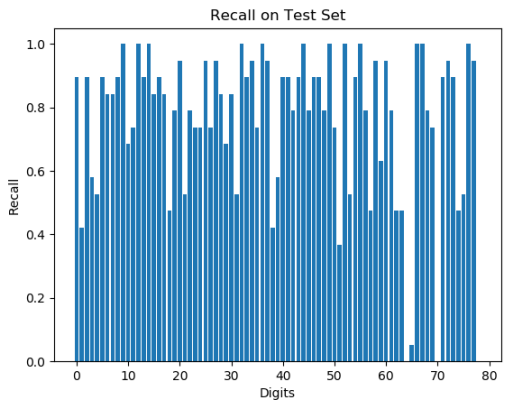


CNN2 Cost

CNN2 Training Accuracy

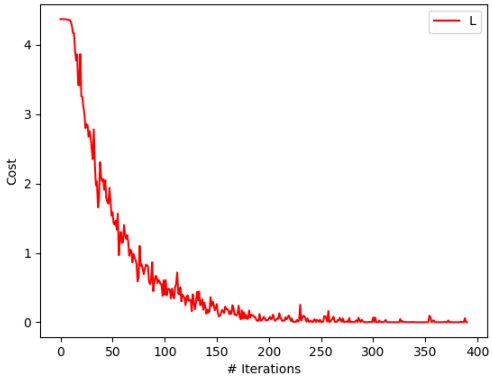


CNN3 Cost

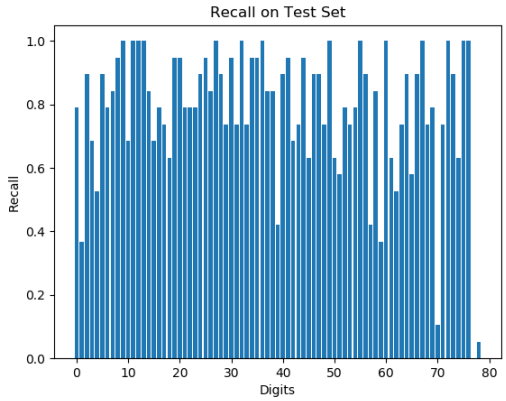


CNN3 Training Accuracy

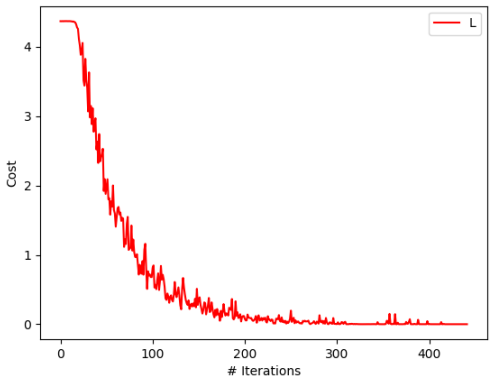
Fig. 5.9: Graphical representations of CNN results for 27*79 (Gurmukhi script dataset)



CNN1 Cost

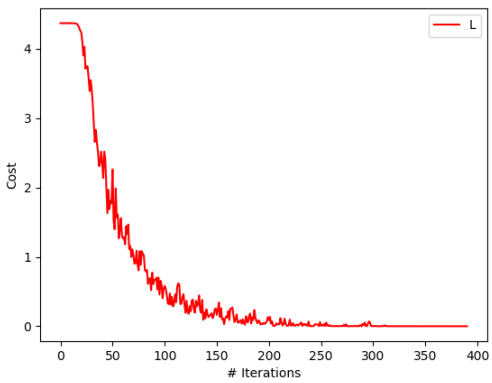


CNN1 Training Accuracy



CNN2 Cost

CNN2 Training Accuracy



CNN3 Cost

CNN3 Training Accuracy

From above results, we conclude that our proposed model performs consistently better than a convolutional model. With the change in batch size and terminating condition the models also improves the results. We conclude all the results by working on different size of dataset with separate images, with the different datasets, batch size and with all these experiments we conclude that proposed model increases the final results with the improvement of 1 – 3 % accuracy.

CHAPTER-6

==========

CONCLUSIONS AND FUTURE SCOPE

==========

In our work, a modified version of pre-existing Softmax function has been proposed which has considerably improved the recognition accuracy during the training stage. In this approach, an auxiliary information is defined and used on the MNIST and Gurmukhi dataset. On comparing the accuracy of the conventional approach to the modified approach which used the auxiliary information, an improvement of approx. (1 - 2 %) of accuracy was achieved. As a part of further exploration, this modified Softmax function can be put to use in other domains and suitability can be accessed.

==========

REFERENCES

==========

[1] Adhatrao, Kalpesh, Aditya Gaykar, Amiraj Dhawan, Rohit Jha, and Vipul Honrao. "Predicting students' performance using ID3 and C4. 5 classification algorithms." *arXiv preprint arXiv:1310.2071* (2013).

[2] Ahmad, Iftikhar, Azween B. Abdulah, and Abdullah S. Alghamdi. "Towards the designing of a robust intrusion detection system through an optimized advancement of neural networks." In *Advances in Computer Science and Information Technology*, pp. 597-602. Springer, Berlin, Heidelberg, 2010.

[3] Berwick, Robert. "An Idiot's guide to Support vector machines (SVMs)." *Retrieved on October* 21 (2003): 2011.

[4] Bhatia, Nitin. "Survey of nearest neighbor techniques." *arXiv preprint arXiv:1007.0085* (2010).

[5] Bhukya, Devi Prasad, and S. Ramachandram. "Decision tree induction: an approach for data classification using AVL-tree." *International Journal of Computer and Electrical Engineering* 2, no. 4 (2010): 660.

[6] Brodley CE, Utgoff PE. Multivariate versus univariate decision trees: Citeseer, 1992. Brodley, Carla E., and Paul E. Utgoff. *Multivariate versus univariate decision trees*. Amherst, MA: University of Massachusetts, Department of Computer and Information Science, 1992.

[7] Brown, Tom B., Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. "Adversarial patch." *arXiv preprint arXiv:1712.09665* (2017).

[8] Cheng, Hong, Xifeng Yan, Jiawei Han, and S. Yu Philip. "Direct discriminative pattern mining for effective classification." In *2008 IEEE 24th International Conference on Data Engineering*, pp. 169-178. IEEE, 2008.

[9] Chen, Yahui. "Convolutional neural network for sentence classification." Master's thesis, University of Waterloo, 2015.

[10] Cireşan, Dan, Ueli Meier, and Jürgen Schmidhuber. "Multi-column deep neural networks for image classification." *arXiv preprint arXiv:1202.2745* (2012).

[11] Cireşan, Dan C., Ueli Meier, Jonathan Masci, Luca M. Gambardella, and Jürgen Schmidhuber. "High-performance neural networks for visual object classification." *arXiv preprint arXiv:1102.0183* (2011).

[12] Cover, Thomas M., and Peter Hart. "Nearest neighbor pattern classification." *IEEE transactions on information theory* 13, no. 1 (1967): 21-27.

[13] Deng, Jia, Alexander C. Berg, Kai Li, and Li Fei-Fei. "What does classifying more than 10,000 image categories tell us?." In *European conference on computer vision*, pp. 71-84. Springer, Berlin, Heidelberg, 2010.

[14] Friedman, Nir, and Moises Goldszmidt. "Discretizing continuous attributes while learning Bayesian networks." In *ICML*, pp. 157-165. 1996.

[15] Glorot, Xavier, and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249-256. 2010.

[16] Goodfellow, Ian J., Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. "Multi-digit number recognition from street view imagery using deep convolutional neural networks." *arXiv preprint arXiv:1312.6082* (2013).

[17] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).

[18] Jang, J-SR. "ANFIS: adaptive-network-based fuzzy inference system." *IEEE transactions on systems, man, and cybernetics* 23, no. 3 (1993): 665-685.

[19] Japkowicz, Nathalie, Catherine Myers, and Mark Gluck. "A novelty detection approach to classification." In *IJCAI*, vol. 1, pp. 518-523. 1995.

[20] Kesavaraj, Gopalan, and Sreekumar Sukumaran. "A study on classification techniques in data mining." In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pp. 1-7. IEEE, 2013.

[21] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In *Advances in neural information processing systems*, pp. 1097-1105. 2012.

[22] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In *Advances in neural information processing systems*, pp. 1097-1105. 2012.

[23] LeCun, Yann, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. "Backpropagation applied to handwritten zip code recognition." *Neural computation* 1, no. 4 (1989): 541-551.

[24] LeCun, Yann, Koray Kavukcuoglu, and Clément Farabet. "Convolutional networks and applications in vision." In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 253-256. IEEE, 2010.

[25] LeCun, Yann, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. "Handwritten digit recognition with a back-propagation network." In *Advances in neural information processing systems*, pp. 396-404. 1990.

[26] LeCun, Yann, and Yoshua Bengio. "Convolutional networks for images, speech, and time series." *The handbook of brain theory and neural networks* 3361, no. 10 (1995): 1995.

[27] Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." *arXiv preprint arXiv:1312.4400* (2013).

[28] Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." *arXiv preprint arXiv:1312.4400* (2013).

[29] Louis, David N., Hiroko Ohgaki, Otmar D. Wiestler, Webster K. Cavenee, Peter C. Burger, Anne Jouvet, Bernd W. Scheithauer, and Paul Kleihues. "The 2007 WHO classification of tumours of the central nervous system." *Acta neuropathologica* 114, no. 2 (2007): 97-109.

[30] Oquab, Maxime, Leon Bottou, Ivan Laptev, and Josef Sivic. "Learning and transferring mid-level image representations using convolutional neural networks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1717-1724. 2014.

[31] Patil, Dipti D., V. M. Wadhai, and J. A. Gokhale. "Evaluation of decision tree pruning algorithms for complexity and classification accuracy." *International Journal of Computer Applications* 11, no. 2 (2010): 23-30.

[32] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79-86. Association for Computational Linguistics, 2002.

[33] Perronnin, Florent, Jorge Sánchez, and Thomas Mensink. "Improving the fisher kernel for large-scale image classification." In *European conference on computer vision*, pp. 143-156. Springer, Berlin, Heidelberg, 2010.

[34] Phyu, Thair Nu. "Survey of classification techniques in data mining." In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, pp. 18-20. 2009.

[35] Quinlan, J. Ross. "Induction of decision trees." *Machine learning* 1, no. 1 (1986): 81-106.

[36] Quinlan, J. Ross. "Simplifying decision trees." *International journal of man-machine studies* 27, no. 3 (1987): 221-234.

[37] Ren, Jianhua, I. Jenkinson, Jiangping Wang, D. L. Xu, and J. B. Yang. "A methodology to model causal relationships on offshore safety assessment focusing on human and organizational factors." *Journal of Safety Research* 39, no. 1 (2008): 87-100.

[38] Rutkowski, Leszek, Lena Pietruczuk, Piotr Duda, and Maciej Jaworski. "Decision trees for mining data streams based on the McDiarmid's bound." *IEEE Transactions on Knowledge and Data Engineering* 25, no. 6 (2012): 1272-1279.

[39] Sánchez, Jorge, and Florent Perronnin. "High-dimensional signature compression for large-scale image classification." In *CVPR 2011*, pp. 1665-1672. IEEE, 2011.

[40] Sharma, Seema, Jitendra Agrawal, Shikha Agarwal, and Sanjeev Sharma. "Machine learning techniques for data mining: A survey." In *2013 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1-6. IEEE, 2013.

[41] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[42] Su, Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One pixel attack for fooling deep neural networks." *IEEE Transactions on Evolutionary Computation* (2019).

[43] Soofi, Aized Amin, and Arshad Awan. "Classification techniques in machine learning: applications and issues." *Journal of Basic and Applied Sciences* 13 (2017): 459-465.

[44] Soofi, Aized Amin, and Arshad Awan. "Classification techniques in machine learning: applications and issues." *Journal of Basic and Applied Sciences* 13 (2017): 459-465.

[45] Tan, Eng M., Alan S. Cohen, James F. Fries, Alfonse T. Masi, Dennis J. Mcshane, Naomi F. Rothfield, Jane Green Schaller, Norman Talal, and Robert J. Winchester. "The 1982 revised criteria for the classification of systemic lupus erythematosus." *Arthritis & Rheumatism: Official Journal of the American College of Rheumatology* 25, no. 11 (1982): 1271-1277.

[46] Twa, Michael D., Srinivasan Parthasarathy, Cynthia Roberts, Ashraf M. Mahmoud, Thomas W. Raasch, and Mark A. Bullimore. "Automated decision tree classification of corneal shape." *Optometry and vision science: official publication of the American Academy of Optometry* 82, no. 12 (2005): 1038.

[47] Varma, Manik, and Andrew Zisserman. "A statistical approach to texture classification from single images." *International journal of computer vision* 62, no. 1-2 (2005): 61-81.

[48] Varma, Manik, and Andrew Zisserman. "A statistical approach to texture classification from single images." *International journal of computer vision* 62, no. 1-2 (2005): 61-81.

[49] Wang, Shuang-cheng, Rui Gao, and Li-min Wang. "Bayesian network classifiers based on Gaussian kernel density." *Expert Systems with Applications* 51 (2016): 207-217.

[50] Wu, Xindong, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan *et al.* "Top 10 algorithms in data mining." *Knowledge and information systems* 14, no. 1 (2008): 1-37.

[51] Yang, Ying, and Geoffrey I. Webb. "Discretization for naive-Bayes learning: managing discretization bias and variance." *Machine learning* 74, no. 1 (2009): 39-74.

[52] Yang, Chenghai, Gary N. Odvody, Carlos J. Fernandez, Juan A. Landivar, Richard R. Minzenmayer, and Robert L. Nichols. "Evaluating unsupervised and supervised image classification methods for mapping cotton root rot." *Precision Agriculture* 16, no. 2 (2015): 201-215.

[53] Yamashita, Rikiya, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. "Convolutional neural networks: an overview and application in radiology." *Insights into imaging* 9, no. 4 (2018): 611-629.

[54] Yin, Xiaoxin, and Jiawei Han. "CPAR: Classification based on predictive association rules." In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pp. 331-335. Society for Industrial and Applied Mathematics, 2003.

[55] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." In *European conference on computer vision*, pp. 818-833. Springer, Cham, 2014.

[56] Zhang, Xiang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." In *Advances in neural information processing systems*, pp. 649-657. 2015.