**ENGLISH HANDWRITTEN DIGIT RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS**



*A project submitted to the Islamic University in partial fulfillment of the requirements for the Degree of Hon’s of Science in Information and Communication Technology*

**Submitted By:**

**Md Sumon Mia**  
Roll No: 1418013 Reg. No: 1328  
Session: 2015-2016  
Department of Information and   
Communication Technology  
Islamic University  
Kushtia – 7003, Bangladesh.

**Supervised By:**

**Dr. Paresh Chandra Barman**

Professor  
Department of Information and   
Communication Technology  
Islamic University  
Kushtia – 7003, Bangladesh.

DEPT. OF INFORMATION AND COMMUNICATION TECHNOLOGY  
ISLAMIC UNIVERSITY, BANGLADESH  
December, 2019.

Dedicated

To

My Beloved

Parents and Teachers

**CERTIFICATE**

I am pleased to certify that **Md Sumon Mia**, examination Roll No: **1418013**, has performed a project work titled, “**ENGLISH HANDWRITTEN DIGIT RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS**”, under my supervision in academic year **2015-2016**, for the partial fulfillment of the requirements for the degree of B.Sc.(Hon’s) of Science. So far as I concern this is an original project work that she carried out for B.Sc.(Hon’s) in the Department of Information and Communication Technology, Islamic University, Bangladesh.

I strongly declare that this dissertation has not been copied from any other project or submitted to elsewhere prior submission to this department.

………………………..

**Dr. Paresh Chandra Barman**

Supervisor

Professor  
Department of Information and Communication Technology  
Islamic University, Bangladesh.

**Acknowledgement**

First of all, I convey my deep respect to Allah, the creator and sustainer who has given me strength and opportunity to complete the Project “**ENGLISH HANDWRITTEN DIGIT RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS**”.

I express my sincere respect, thanks and gratitude to my project supervisor  **Dr. Paresh Chandra Barman,** Professor, Department of Information and Communication Technology, Islamic University, Bangladesh, for his suggestion to select the topic; continuous supervision, advice, encouragement, constant guidance and other fruitful helps throughout the duration of my project. His dedication, collaboration, interaction, indefatigable care and keen interest at every step of work were the key factors in the success of my project. Really his expanded helping hands enable me to complete the project. I am grateful to him.

I would like to express thanks and gratitude to my all respectable teachers, of my Dept. of Information and Communication Technology. They all speak out their delegations about my project work.

I would like to express my intellectual debts of gratitude to the numerous authors, to all my classmates, well-wishers who give me confidence.

Finally, I would like to convey my respect to my parents whose continued sacrifice and inspiration over the years were inevitable for the higher education of their children.

**Author**

Md Sumon Mia  
December, 2019.

**Abstract**

In recent time Convolutional Neural Networks, Deep Learning has brought a dramatic twist in the field of Artificial Intelligence. Handwritten Digit recognition is gaining huge demand in the field of Deep Learning. Our challenge is to improve the accuracy of handwritten English Digit 0 to 9 recognition. Generally, the handwritten Digit recognition process consists of four steps: data preprocessing, segmentation, the feature extraction and selection, application of supervised learning algorithms .In this paper we take MNIST dataset for implement Handwritten Digit Recognition using Convolutional Neural Network (CNN).We have total 70000 images for training and testing. Each Digit represent as 28 by 28 gray scale pixel for better result. The Digit are passed into input layer and then pass hidden layer that consist of activation and polling layers and finally it is mapped into fully connected layer and given a softmax classifier to classify the digits. The network is trained using gradient descent algorithm and back propagation algorithm. It has been broadly used in pattern recognition, speech recognition, face recognition, text categorization and document analysis. The accuracy of the project is more than 98 percent.

Table of Contents

[Chapter 1 7](#_Toc46413126)

[INTRODUCTION 7](#_Toc46413127)

[1.1 Introduction 8](#_Toc46413128)

[1.2 Background Study 9](#_Toc46413129)

[1.3 Aims and Objectives 9](#_Toc46413130)

[Chapter 2 10](#_Toc46413131)

[Literature Review and Related Work 10](#_Toc46413132)

[2.1 Introduction 11](#_Toc46413133)

[2.2 Fully Connected Multi-layer Neural Network 12](#_Toc46413134)

[2.3 K-Nearest Neighbour Classifier 13](#_Toc46413135)

[2.4 Support Vector Machine (SVM) 13](#_Toc46413136)

[Chapter 3 15](#_Toc46413137)

[Design and Methodology 15](#_Toc46413138)

[3.1 Working method 16](#_Toc46413139)

[3.2 Environment setup 19](#_Toc46413140)

[3.3 Implementation of LeNet Architecture 20](#_Toc46413141)

[Chapter 4 22](#_Toc46413142)

[Result and Discussion 22](#_Toc46413143)

[4.1 Screenshot 23](#_Toc46413144)

[4.2 Discussion of the Obtained Simulated Results 25](#_Toc46413145)

[Chapter 5 30](#_Toc46413146)

[Conclusion 30](#_Toc46413147)

[5.1 Introduction 31](#_Toc46413148)

[5.2 Future work 31](#_Toc46413149)

[5.3 Refferences 33](#_Toc46413150)

[3.1 Working Principle 34](#_Toc46413151)

# 

# Chapter 1

# INTRODUCTION

## 1.1 Introduction

Handwritings are the most common standard and regular medium that are used in Communication. In Handwritten Digit recognition system is a mechanism that used to recognize human handwritten Digit in English language either from scanned handwritten image. In the deep learning Convolutional Neural Networks is being used for visual image analyze. Object detection, face recognition, robotics, video analysis, segmentation, pattern recognition, natural language processing, spam detection, topic categorization, regression analysis, speech recognition, image classification are some of the examples that can be done using Convolutional Neural Networking. The accuracy in these fields including Handwritten English Digit Recognition using Convolutional Neural Network has been reached human level perfection. In 1998 the framework of Convolutional Neural Network is designed by LeCun et al. It was adapt in Handwritten Digit classification direct from pixel value of an image. Gradient descent and back propagation are used to training these models. Handwritten Digit is given an input and systems recognize the digit. The Neuron’s give an output of 0 or 1 if the weighted sum is below or above some threshold value. Various decision making models are formed by different weights and threshold values.

A simple Artificial Neural Network has an input layer, output layer and hidden layer between input and output layer. In the network the first layer of perceptron’s that makes very simple decisions, by multiplying the weights with the inputs. In this way a perceptron in the second layer can make even more complex decision than a perceptron in the first layer. The layers away from the first layer make progressively more complex decisions compared to the first layer. For learning purpose we should continuously change the weights so that the network finds out the aggregate and compares it with a threshold value of bias. If a small change in the weights modifies the output in the direction we want to proceed then we can use small weights or we can take large weights for training, this method is like hit and trial which we use in solving higher degree polynomials. The goal of this article is to observe the influence of hidden layers of a CNN for Handwritten Digits. We have used different types of Convolutional Neural networks on MNIST dataset that’s written in python language. We have 60000 images for training dataset and 10000 images for testing dataset. Each digit is representing 28 by 28 gray scale pixels for better result. The main purpose of this paper is to analyze the variation of outcome results for using a different combination of hidden layers of Convolutional Neural Network. We use gradient descent and back propagation algorithm for training dataset and forward propagation for testing dataset.

## 1.2 Background Study

There are several research works based on English handwritten digit recognition using deep learning. The convolutional neural networks was introduced for better supervised learning and accuracy. Recently some researchers have shown a better accuracy more than 99.50% using convolutional neural networks. So recent most of the researchers of English handwritten digit recognition are using deep CNN architecture. There are some other classifier like support vector machine(SVM), Neural Networks(NN) etc. for handwritten digit recognition. But the performance of deep CNN is better than other classifier. As a result CNN has become most popular recent trend for English handwritten digit recognition. Modified CNN architecture by adding more layers or more nodes, changing optimization method and activation function has become a way to break the state of the accuracy. This strategy is followed by the some researcher’s work. Most of the system were shallow learning methods like feature extraction and multilayer perception technique.

## 1.3 Aims and Objectives

Our destination is to classify and recognize English handwritten digit from sample image of handwritten isolated words for which we have decided to classify 10 distinct digits. The large number of potential application of English handwritten digit recognition such as traffic number plat recognition, automatic ID card reading, reading bank cheque and digitization of documents etc.

# Chapter 2

# Literature Review and Related Work

## 2.1 Introduction

An early notable attempt in the area of character recognition research is by Grimsdale in 1959. The origin of a great deal of research work in the early sixties was based on an approach known as analysis-by-synthesis method suggested by Eden in 1968. The great importance of Eden's work was that he formally proved that all handwritten characters are formed by a finite number of schematic features, a point that was implicitly included in previous works. This notion was later used in all methods in syntactic (structural) approaches of character recognition.

CNN is playing an important role in many sectors like image processing. It has a powerful impact on many fields. Even, in Nano-technologies like manufacturing semiconductors, CNN is used for fault detection and classification. Handwritten digit recognition has become an issue of interest among researchers. There are a large number of papers and articles are being published these days about this topic. In research it is shown that Deep Learning algorithm like multilayer CNN using Keras with Theano and Tensorflow gives the highest accuracy in comparison with the most widely used machine learning algorithms like SVM, KNN & RFC. Because of its highest accuracy, Convolutional Neural Network (CNN) is being used in a large scale in image classification, video analysis etc. Many researchers are trying to make sentiment recognition in a sentence. CNN is being used in natural language processing and sentiment recognition by varying different parameters. Many researchers are trying to increase the accuracy with less error in CNN. In another research, they have shown that deep nets perform better when they are trained by simple back-propagation. Their architecture results in the lowest error rate on MNIST compare to NORB and CIFAR10. Researchers are working on this issue to reduce the error rate as much as possible in handwriting recognition. In one research, an error rate of 1.19% is achieved using 3-NN trained and tested on MNIST. Deep CNN can be adjustable with the input image noise. Some researchers are trying to come up with new techniques to avoid drawbacks of traditional convolutional layer's. Ncfm (No combination of feature maps) is a technique which can be applied for better performance using MNIST datasets. Its accuracy is 99.81% and it can be applied for large-scale data. New applications of CNN are developing day by day with many kinds of research. Researchers are trying hard to minimize the error rates. Using MNIST datasets and CIFAR, error rates are being observed.

In Germany, a traffic sign recognition model of CNN is suggested. It proposed a faster performance with 98.30% accuracy. Loss function was designed, which is applicable for light-weighted 1D and 2D CNN. In this case the accuracies were 93% and 91% respectively.

## 2.2 Fully Connected Multi-layer Neural Network

Deep Learning deals with training multi-layer artificial neural networks, also called Deep Neural Networks. After Rosenblatt perceptron was developed in the 1950s, there was a lack of interest in neural networks until 1986, when Dr. Hinton and his colleagues developed the back propagation algorithm to train a multilayer neural network. This network extracts features based on the entire spatial domain of images hence the number of parameters required is very high. The problem with these networks is they tend to be over parameterized, in order of 100,000’s which is unwanted when working with complex classification problems with complex data sets. The number of layers and the number of neurons are referred to as hyper parameters of a neural network, and these need tuning. Cross-validation techniques must be used to find ideal values for these.

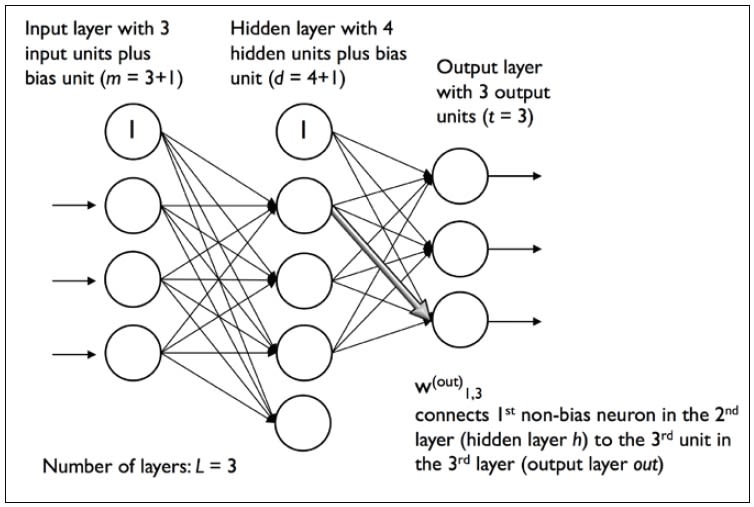


Fig 2.1: Multi-layer Artificial Neural Network

## 2.3 K-Nearest Neighbor Classifier

In pattern recognition, the **k**-**nearest** neighbor’s algorithm (**k**-**NN**) is a non-parametric method used for classification and regression. In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression. A KNN classifier with a distance measure like Euclidean distance between the data sets input images is also capable of classification of digits but at higher error rate than a fully connected ML neural network. The key features of this classifier is that it requires no training time and no input from the programmer in terms of knowledge for designing the system. The big over head of this classifier is memory requirement and the classification or recognition time. We take into consideration that this nearest-neighbor system works on raw pixels instead of feature vectors.

## 2.4 Working process of KNN Algorithm

**Step 1** − For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

**Step 2** − Next, we need to choose the value of K the nearest data points. K can be any integer.

**Step 3** – For each point in the test data do the following −

* **3.1** − Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
* **3.2** − Now, based on the distance value, sort them in ascending order.
* **3.3** − Next, it will choose the top K rows from the sorted array.
* **3.4** − Now, it will assign a class to the test point based on most frequent class of these rows.

**Step 4** – End

## 2.4 Support Vector Machine (SVM)

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).

# Chapter 3

# Design and Methodology

## 3.1 Working method

A Convolutional Neural Network (CNN) is a type of feed-forward Artificial Neural Network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex.

Convolutional Neural Networks consist of neurons that have learnable weights and biases. Each neuron receives some input, performs a dot product and optionally follows it with a non-linearity.



Fig 3.1: Convolutional Neural Network Basic Layout.

The whole Convolutional Neural Network expresses a differentiable score function that is further followed by a Softmax function. The data input into the Convolutional Neural Network is arranges in the form of its width, height and depth as shown in figure below.



Fig 3.2: Arrangement of Neurons in CNN.

Layers of Convolutional Neural Network

A CNN consists of a lot of layers. These layers when used repeatedly lead to a formation of a Deep Neural Network. Three main types of layers used to build a CNN are:

1. **Input:** This layer holds the raw pixel values of image.

2. **Convolutional Layer:** This layer gets the results of the neuron layer that is connected to the input regions. We define the number of filters to be used in this layer. Each filter may be a 5x5 window that slider over the input data and gets the pixel with the maximum intensity as the output.

3. **Rectified Linear Unit [ReLU] Layer**: This layer applies an element wise activation function on the image data. We know that a CNN uses back propagation. So in order to retain the same values of the pixels and not being changed by the back propagation, we apply the ReLU function.

4. **Pooling Layer:** This layer perform a down-sampling operation along the spatial dimensions (width, height), resulting in volume.

5. **Fully Connected Layer:** This layers is used to compute the score classes which class has the maximum score corresponding to the input digits

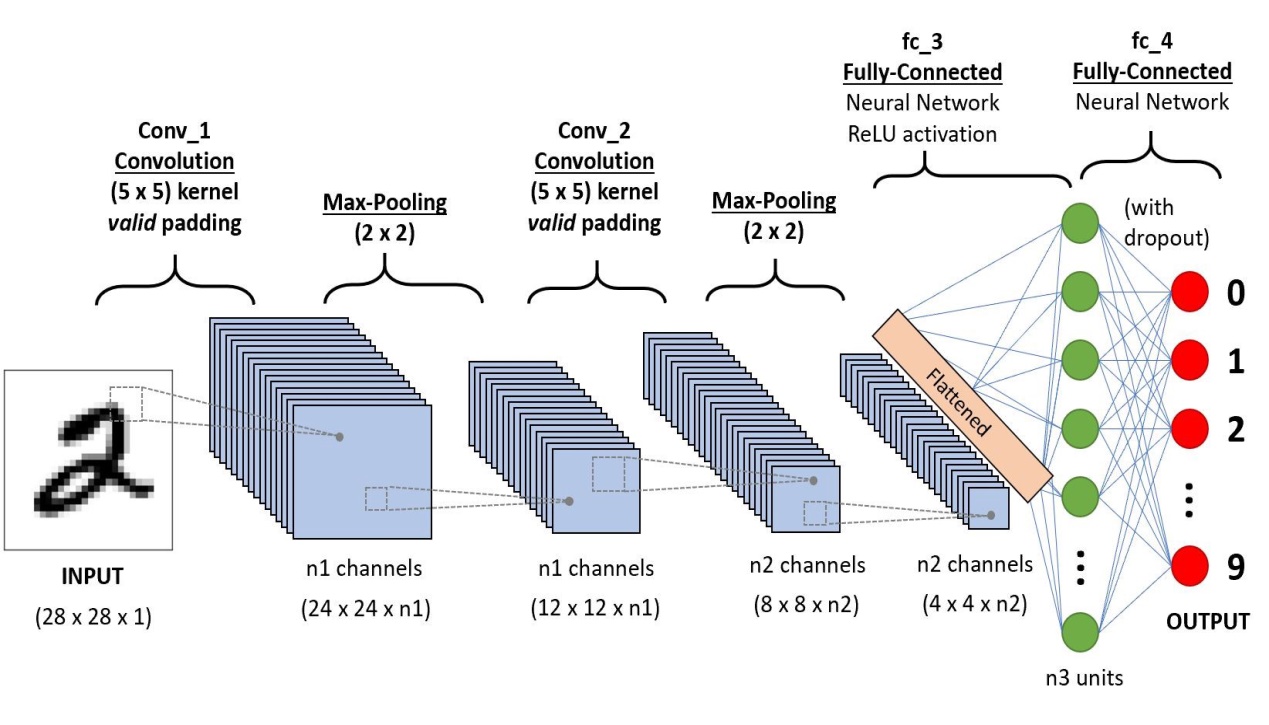


Fig 3.3: Convolutional Neural Network Model

CNN for Handwritten Digit Recognition

The CNN for Handwritten Digit Recognition works in three main phases.

**Phase1 - Input MNIST Data1:** The first phase is to input the MNIST data. The MNIST data is provided as 784-d array of pixels. So firstly we convert it to gray scale images using 28x28 matrix of pixels.

**Phase2 – Building Network Architecture:** In the second phase, we define the models to be used to build a convolutional neural network. Here, we use the Sequential class from Keras to build the network. In this network, we have three layer sets of layers “CONV =>ReLU=> POOL”.

a) **First Convolution Layer:** In the first layer, we take 20 convolutional filters that go as a sliding window of size 5x5 over all the images of 28x28 matrix size and try to get the pixels with most intensity value.

b) **ReLU Function:** We know that convolution is a method that uses Back Propagation. So using the ReLU function as the activation function just after the convolutional layer reduces the likelihood of the vanishing gradient and avoids sparsity. This way we don’t lose the important data and even get rid of redundant data like a lot of 0’s in the pixels.

c) **Pooling Layer:** The pooling layer gets the data from the ReLU function and down-samples the steps in the 3D tensor. In short it pools all the pixels obtained from previous layers and again forms a new image matrix of a smaller size. These images are again input into the second set of layers. “CONV =>ReLU=> POOL” and this process goes on till we get to a smallest set of pixels from which we can classify the digit.

**Phase 3 –Fully Connected Layer:** The fully connected layer is used to connect each of the previous layers to the next layers. This layer consists of 500 neurons. Finally, we apply a Softmax Classifier that returns a list of probabilities for each of the 10 class labels. The class label with the largest probability is chosen as the final classification from the network and shown in the output. This output received is used to make the confusion matrix for the model. In this we can add more number of layers but adding more layers might affect the accuracy of the system. Since, it uses multiple layers, so it is called a Deep Learning system.

## 3.2 Environment setup

Here we define our list of libraries we need to install for keras library to work for our networks. The most important library is the NUMPY is a library that provides support for large, multi-dimensional arrays where we can store our input pixel matrix of size 28 by 28, using numpy we can express images as multi-dimensional arrays of pixel intensity values. We can also rely on the NumPy’s built-in highly advanced mathematical functions and we can apply logistic regression on the image. The next library which is to be installed is the Python SCIPY library. It adds futher help for scientific and technical computing of our functions. The important sub package of SciPy is the package that has a huge amount of distance functions which are implemented using trees. Normally after extracting features the image is represented as a list of numbers, in order to compare these two images we need distance computation methods, such as Euclidean distance. Next up is PILLOW library useful for manipulations on image such as resizing, rotation. Then we come to OPENCV library and the main goal of this library is real-time image processing. Next we can install SCIKIT-LEARN library which is by the way not a image vision library but a machine learning library. This library helps us with advanced computer vision whether it may be in clustering, quantization, classification models. The library next to be installed is h5py to store large numerical datasets, it also provides support for NumPy arrays it has efficient and long term storage of NumPy arrays.

## 3.3 Implementation of LeNet Architecture

After installing these libraries in python we can use our keras deep learning library to implement our network and create python files for network creation and instantiation. In our implementation we have trained the network in such a way that it learns many filters of size 5 by 5 and then pass it through a ReLU activation function followed by 2 by 2 max pooling in both dimensions. We then take the output of Max-pooling layers to apply it to fully connected layers. Our fully connected layers contain around 500 units which we will pass through another ReLU activation that enables us to combine them into classes , which are useful for identifying our image , we have 10 classes one for each digit to classify. Finally we apply a softmax classifier that will give us the list of probabilities, one for each of the 10 classes created. The class label with largest probability will be taken as the final classification of our network. In our driver program datasets can be downloaded from mldata submodule and load it to our network, then train the network using Stochastic Gradient Descent with learning rate and number of iteration. Now we are running keras on the top of TENSOR FLOW as backend to train the network. In our implementation we have given 20 epochs for better accuracy

****

Fig 3.4: The architecture of LeNet5

# Chapter 4

# Result and Discussion

## 4.1 Screenshot

In this model we used MNIST dataset. There are 60000 train data and 10000 test data in this dataset. We take an image as input and passes through this model and model predict this image as digit.

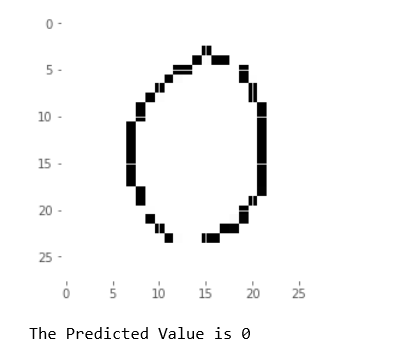


Fig4.1: Original image Fig 4.2:Predicted image

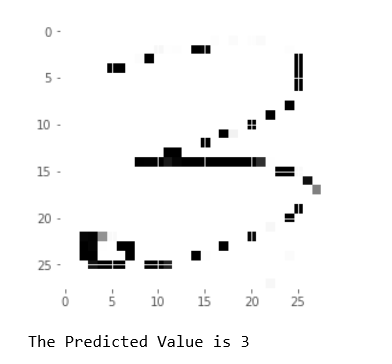


Fig 4.3: Original image Fig 4.4: Predicted image

In figure 4.1 we take an original image input zero and it pass through the model model predicted it zero that is shown in figure 4.2.in figure 4.3 we take an original image input three and it pass through the model model predicted it three that is shown in figure 4.

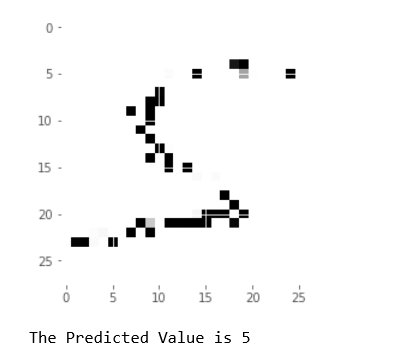


 Fig 4.5: Original image Fig 4.6: Predicted image

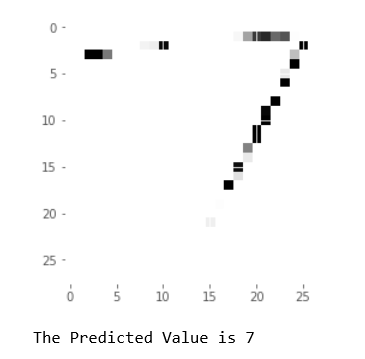


Fig 4.7: Original image Fig 4.8:Predicted image

In figure 4.5 we take an original image input five and it pass through the model model predicted it five that is shown in figure 4.6.in figure 4.7 we take an original image input seven and it pass through the model model predicted it seven that is shown in figure 4.8

## 4.2 Discussion of the Obtained Simulated Results

Our network has been trained with 60000 data and tested with 10000 data. We obtain the accuracy up to>98 percent which is good enough for our testing classification. We have given learning rate 0.01 to our algorithm and obtained good classification results, every time after training we are taking random inputs from our testing dataset and calculating the efficiency each time it is executed. The interesting properties of this algorithm is that the training error keep decreasing over time but the test error goes through a minimum and starts increasing after a certain number of iterations , this is possibly because of the higher learning rate and by decreasing it we can get our results , if not reduced the learning rate the Stochastic gradient descent may get stuck in local minimum and finds it difficult to predict the optimized weights , which affects the prediction and accuracy of our network.

In this section, CNN has been applied on the MNIST dataset in order to observe the variation of accuracies for handwritten digits. The accuracies are obtained using Tensorflow in python. Table 1 shows the minimum and maximum training and validation accuracies of CNN found after the simulation for the six different cases by varying number of hidden layers for the recognition of handwritten digits.

Table 1: Performance of CNN for the five different cases for various Hidden Layers, Batch Size and Epochs

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Case** | **Number of Hidden Layers** | **Batch Size** | **Minimum**  **Training**  **Accuracy** | | **Minimum**  **Validation**  **Accuracy** | | **Maximum**  **Training Accuracy** | | **Maximum**  **Validation**  **Accuracy** | | **Overall**  **Performance**  **Validation**  **Accuracy**  (%) |
| **Epoch** | **Accuracy**  **(%)** | **Epoch** | **Accuracy**  **(%)** | **Epoch** | **Accuracy**  **(%)** | **Epoch** | **Accuracy**  **(%)** |
| **1** | **4** | **100** | **1** | **90.98** | **1** | **97.45** | **13** | **98.50** | **14** | **99.13** | **98.29** |
| **2** | **4** | **128** | **1** | **91.45** | **1** | **98.25** | **14** | **99.25** | **15** | **99.80** | **99.02** |
| **3** | **3** | **100** | **1** | **93.23** | **3** | **97.35** | **14** | **100** | **15** | **99.50** | **98.42** |
| **4** | **3** | **128** | **1** | **96.50** | **1** | **98.50** | **15** | **98.25** | **14** | **99.60** | **99.05** |
| **5** | **4** | **100** | **1** | **92.68** | **1** | **98.23** | **13** | **99.85** | **12** | **99.26** | **98.74** |
|  | | | | | | | | | | | |

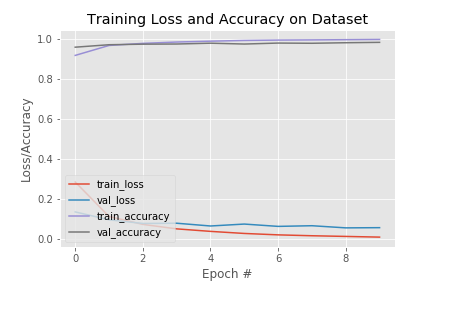


Fig 4.9: Observe loss and accuracy for case 1

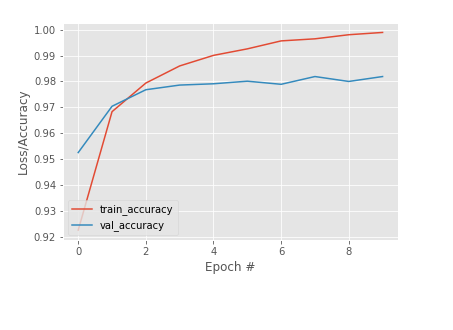


Fig 4.10: Observe loss and accuracy for case 2

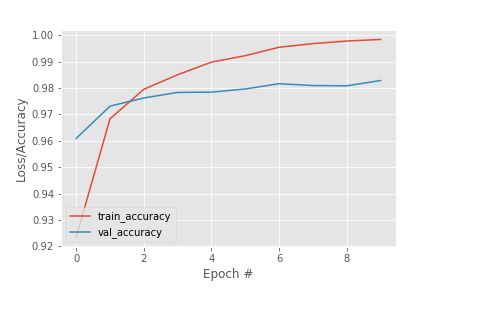


Fig 4.11: Observe loss and accuracy for case 3

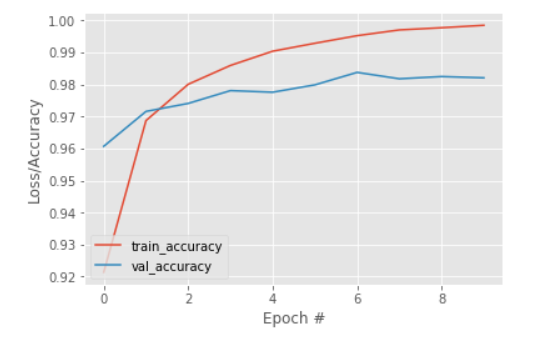


Fig 4.12: Observe loss and accuracy for case 4

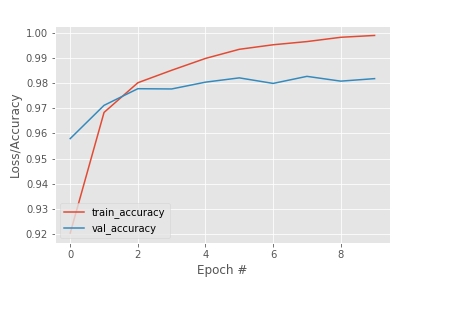


Fig 4.13: Observe loss and accuracy for case 5

# Chapter 5

# Conclusion

## 5.1 Introduction

Performance of a network depends on many factors like low memory requirements, low run time and better accuracy, although in this paper it is primarily focused on getting better accuracy rate for classification. Before Artificial neurons had better accuracy but now the branch of computer vision mainly depends on deep learning features like convolutional neural networks. Research is still going on in this field and researches have developed many forms of LeNet architecture like LeNet-1,LeNet-4, Boosted LeNet-4 and also combination of many methods like LeNet-4 with KNN’s , but for a quite long time our LeNet architecture was considered as state of the art. Many other methods like Tangent Distance Classifier were developed using LeNet architecture. The main aim of this paper deals with one of the method in which it can be implemented , there are several methods in which they can be done and using different frameworks like matlab, octave. The branch of computer vision in artificial intelligence primary motive is to develop a network which is better to every performance measure and provide results for all kinds of datasets which can be trained and trained and recognized. In this paper, the variations of accuracies for handwritten digit were observed for 10 epochs by varying the hidden layers. The accuracy curves were generated for the six cases for the different parameter using CNN MNIST digit dataset. The maximum and minimum accuracies were observed for different hidden layers variation with a batch size of 1000. Among all the observation, the maximum accuracy in the performance was found 98.27% for 10 epochs in case 3. In digit recognition, this type of higher accuracy will cooperate to speed up the performance of the machine more adequately. However, the minimum accuracy among all observation in the performance was found 98.07% in case 4.

## 5.2 Future work

Fixed size Convolutional Neural Networks has been applied to many applications like handwritten digit recognition , machine printed character recognition and on-line handwriting recognition, they can also be useful for signature verification .The more the training examples the more is the accuracy of the networks .Unsupervised machine learning was made easier using Convolutional Neural networks , some of the future works possible to implement by CNN’s are compressing or obtaining same results from smaller networks by optimization tricks , more invariant feature learning such that the input images doesn’t gets distorted. The major 3D vision networks is a scope for researches to develop using LeNet architecture and more biologically concordant methods , a hope for future is that Unsupervised CNN’s. In the future, our plan is to observe the variation in the overall classification accuracy by varying the number of hidden layers and batch size.

## 5.3 Refferences

[1] Yann LeCun, Leon Bottou, Yoshua Bengio and Patrick Haffner, paper on “Gradient Based Learning Applied to Document Recognition”, Proc of the IEEE, NOVEMBER 1998.

[2]Y.LeCun,L.Jackel,L.bottom,A.brunot,C.Cortes,J.Denker,H.Drucker,I.guyon,U.muller paper on “Comparision of Learning Algorithms for handwritten digit recognition” .

[3] Y.LeCun, B.Baser, J.S.Denker, D.Henderson, R.E.Howard, R.Hubbard, and L.D.Jackel , Handwritten digit recognition with a back- prorogation network in D.Tourezky, Advances in Neural Information Processing Systems 2, Morgan Kaufman(1990)

[4] Corima Cartes and Vladimir Vapnik the Soft Margin Classifier, Machine Learning to appear(1995)

[5] Haider A.Alwzwazy, Hayder M. Albehadili, Younes S.Alwan ,Naz E Islam paper on “Handwritten Digit Recognition Using Convolutional Neural Networks” Vol 4 ,Issue 2,February 2016

[6] Xuan Yang, Jing Pu paper on “Mdig: Multi-digit Recognition using Convolutional Neural Network on Mobile.

[7] Saeed Al-Mansoori paper on “Intelligent Digit Recognition using Artificial Neural Networks Vol 5, Issue 5, (Part-3) May 2015, pp 46-51

[8] Y. LeCun, "The MNIST database of handwritten digits," *http://yann. Lecun.com/exdb/mnist/,*1998.

## 3.1 Working Principle

A Convolutional Neural Network (CNN) is a type of feed-forward Artificial Neural Network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex.

Convolutional Neural Networks consist of neurons that have learnable weights and biases. Each neuron receives some input, performs a dot product and optionally follows it with a non-linearity.



Fig 3.1: Convolutional Neural Network Basic Layout.

The whole Convolutional Neural Network expresses a differentiable score function that is further followed by a Softmax function. The data input into the Convolutional Neural Network is arranges in the form of its width, height and depth as shown in figure below.



Fig 3.2: Arrangement of Neurons in CNN.

Layers of Convolutional Neural Network

A CNN consists of a lot of layers. These layers when used repeatedly, lead to a formation of a Deep Neural Network. Three main types of layers used to build a CNN are:

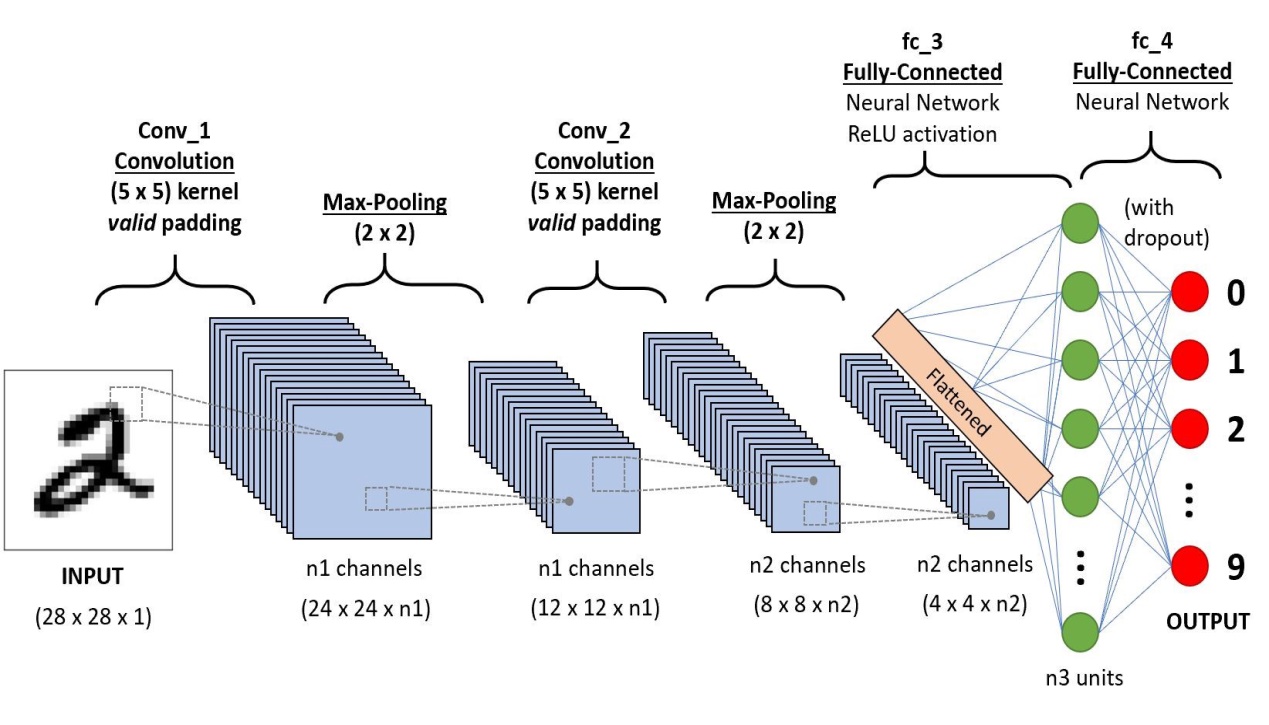
1. **Input:** This layer holds the raw pixel values of image.

2. **Convolutional Layer:** This layer gets the results of the neuron layer that is connected to the input regions. We define the number of filters to be used in this layer. Each filter may be a 5x5 window that slider over the input data and gets the pixel with the maximum intensity as the output.

3. **Rectified Linear Unit [ReLU] Layer**: This layer applies an element wise activation function on the image data. We know that a CNN uses back propagation. So in order to retain the same values of the pixels and not being changed by the back propagation, we apply the ReLU function.

4. **Pooling Layer:** This layer perform a down-sampling operation along the spatial dimensions (width, height), resulting in volume.

5. **Fully Connected Layer:** This layers is used to compute the score classes i.e which class has the maximum score corresponding to the input digits



CNN for Handwritten Digit Recognition

The CNN for Handwritten Digit Recognition works in three main phases.

1. **Phase1 - Input MNIST Data1:** The first phase is to input the MNIST data. The MNIST data is provided as 784-d array of pixels. So firstly we convert it to gray scale images using 28x28 matrix of pixels.

2. **Phase2 – Building Network Architecture:** In the second phase, we define the models to be used to build a convolutional neural network. Here, we use the Sequential class from Keras to build the network. In this network, we have three layer sets of layers “CONV =>ReLU=> POOL”.

a) **First Convolution Layer:** In the first layer, we take 20 convolutional filters that go as a sliding window of size 5x5 over all the images of 28x28 matrix size and try to get the pixels with most intensity value. b) **ReLU Function:** We know that convolution is a method that uses Back Propagation. So using the ReLU function as the activation function just after the convolutional layer reduces the likelihood of the vanishing gradient and avoids sparsity. This way we don‟t lose the important data and even get rid of redundant data like a lot of 0‟s in the pixels. c) **Pooling Layer:** The pooling layer gets the data from the ReLU function and down-samples the steps in the 3D tensor. In short it pools all the pixels obtained from previous layers and again forms a new image matrix of a smaller size. These images are again input into the second set of layers i.e. “CONV =>ReLU=> POOL” and this process goes on till we get to a smallest set of pixels from which we can classify the digit.

**3. Phase 3 –Fully Connected Layer:** The fully connected layer is used to connect each of the previous layers to the next layers. This layer consists of 500 neurons. Finally, we apply a Softmax Classifierthat returns a list of probabilities for each of the 10 class labels. The class label with the largest probability is chosen as the final classification from the network and shown in the output.

This output received is used to make the confusion matrix for the model. In this we can add more number of layers but adding more layers might affect the accuracy of the system. ince, it uses multiple layers, so it‟s called a Deep Learning system.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

**Batch Size Minimum**

**Training**

**Accuracy**

**Minimum**

**Validation**

**Accuracy**

**Maximum**

**Training**

**Accuracy**

**Maximum**

**Validation**

**Accuracy**

**Overall**

**Performance**

**Validation**

**Accuracy**

**(%)**

***Epoch Accuracy***

***(%)***

***Epoch Accuracy***

***(%)***

***Epoch Accuracy***

***(%)***

***Epoch Accuracy***

***(%)***

1 3 100 1 91.94 1 97.73 13 98.99 14 99.16 99.11

2 4 100 1 90.11 1 97.74 14 98.94 14 99.24 99.21

3 3 100 1 94.35 3 98.33 15 100 15 99.06 99.06

4 4 100 1 92.94 1 97.79 15 99.92 13 99.92 99.20

5 3 100 1 91.80 1 98.16 13 99.09 12 99.12 99.09

6 4 100 1 90.50 1 97.13 15 99.24 13 99.26 99.07

In the first case shown in figure 4, the first hidden layer

is the convolutional layer 1 which is used for the feature

extraction. It consists of 32 filters with the kernel size of 3×3

pixels and the rectified linear units (ReLU) is used as an

activation function to enhance the performance. The next

hidden layer is the convolutional layer 2 consists of 64 filters

with a kernel size of 3×3 pixels and ReLU. Next, a pooling

layer 1 is defined where max pooling is used with a pool size

of 2×2 pixels to minimize the spatial size of the output of a

convolution layer. A regularization layer dropout is used

next to the pooling layer 1 where it randomly eliminates

25% of the neurons in the layer to reduce overfitting. A

flatten layer is used after the dropout which converts the 2D

filter matrix into 1D feature vector before entering into the

fully connected layers. The next hidden layer used after the

flatten layer is the fully connected layer 1 consists of 128

neurons and ReLU. A dropout with a probability of 50% is

used after the fully connected layer 1. Finally, the output

layer which is used here as fully connected layer 2 contains

10 neurons for 10 classes and determines the digits

numbered from 0 to 9.

A softmax activation function is incorporated with the

output layer to output digit from 0 through 9. The CNN is fit

over 15 epochs with a batch size of 100. The overall

validation accuracy in the performance is found at 99.11%.

At epoch 1 the minimum training accuracy of 91.94% is

found and 97.73% of validation accuracy is found. At epoch

13, the maximum training accuracy is found 98.99% and at

epoch 14, the maximum validation accuracy is found

99.16%. The total test loss for this case is found

approximately 0.037045.

Fig. 4.