# 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 [3]. 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 cheques and digitization of documents etc.

# Chapter 2

# Literature review

# 2.1 Introduction

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 [18]. 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 hyperparameters of a neural network, and these need tuning. Cross-validation techniques must be used to find ideal values for these.

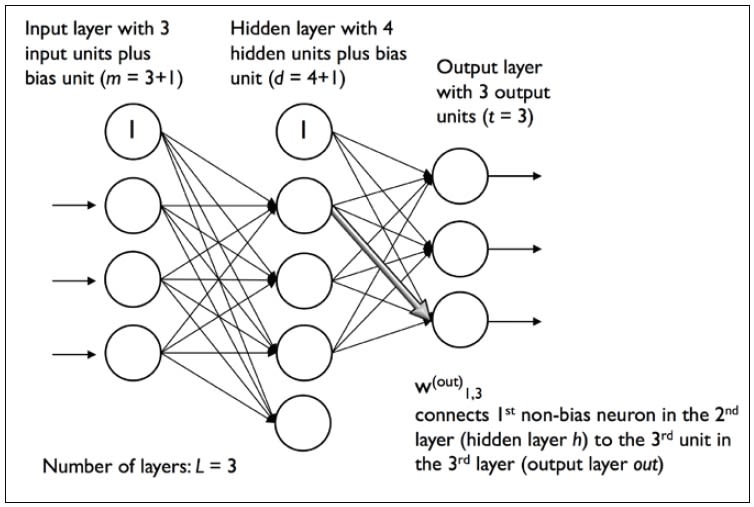


Fig 2.1: Multi-layer Artificial Neural Network

## 2.3 K-Nearest Neighbour 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 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).

The followings are important concepts in SVM −

* **Support Vectors** − Datapoints that are closest to the hyperplane is called support vectors. Separating line will be defined with the help of these data points.
* **Hyperplane** − As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.
* **Margin** − It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

# Chapter 3

# Methodology

## 3.1 Working methology

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. 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

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.

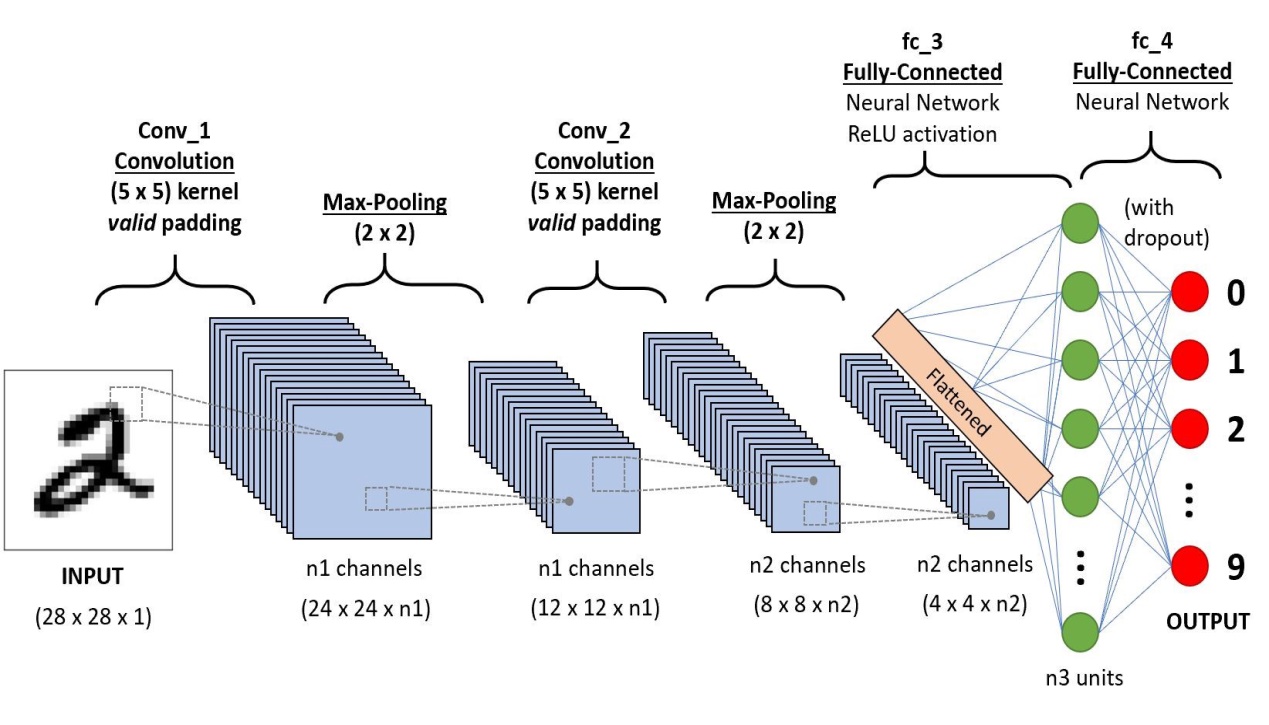
2. **Convolutional Layer:** This layer gets the results of the neuron layer that is connected to the input regions. We define the number of filters to be used in this layer. Each filter may be a 5x5 window that slider over the input data and gets the pixel with the maximum intensity as the output.

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





*C. 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 grayscale 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.

# 

# 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.

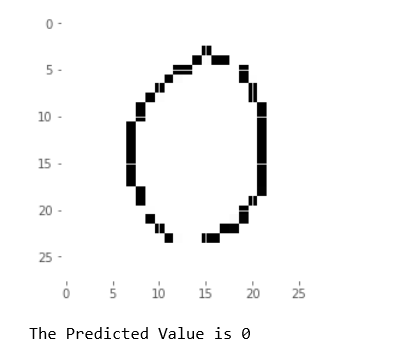


Fig4.1: Original image Fig 4.2:Predicted image

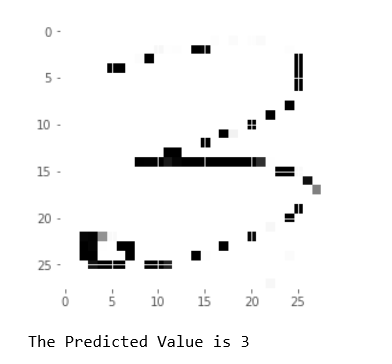


Fig 4.3:Original image Fig 4.4: Predicted image

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

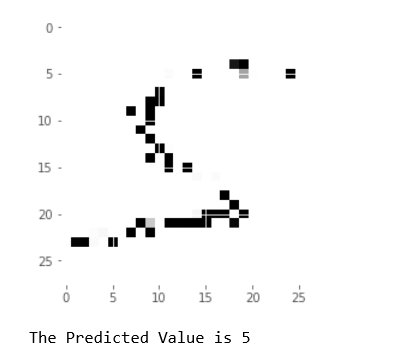


 Fig 4.5: Original image Fig 4.6: Predicted image

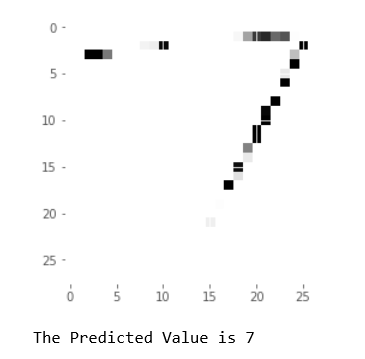


Fig 4.7: Original image Fig 4.8:Predicted image

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

## 4.2 Test Result

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.

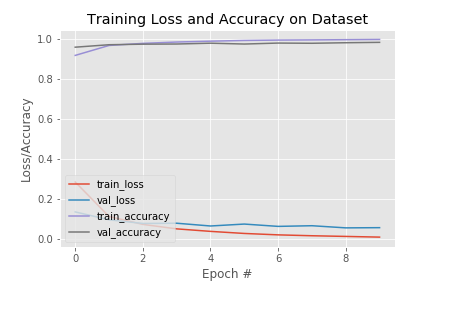


Fig 4.9: Observe loss and accuracy for case 1

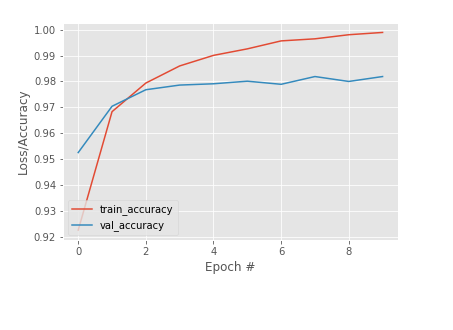


Fig 4.10: Observe loss and accuracy for case 2

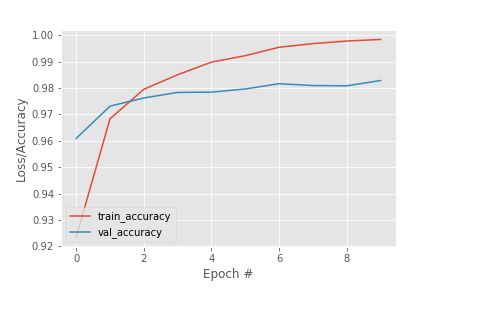


Fig 4.11: Observe loss and accuracy for case 3

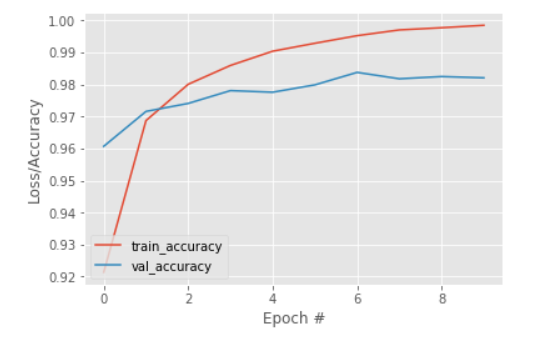


Fig 4.12: Observe accuracy for case 4

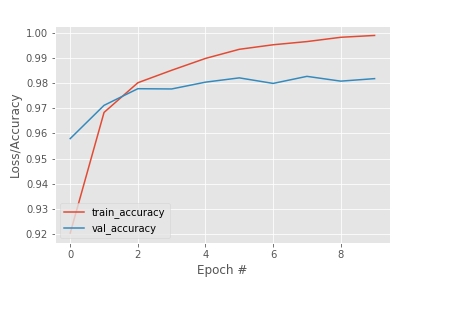


Fig 4.13: Observe accuracy for case 5

For case 1 minimum training and validation accuracy is found in epoch 1. The maximum training accuracy is 99.34 and validation accuracy is 98.34.The maximum testing accuaracy is 98.12 For case 2 minimum training and validation accuracy is found in epoch 1. The maximum training accuracy is 99.44 and validation accuracy is 98.20. The maximum testing accuaracy is 98.13 .For case 3 minimum training and validation accuracy is found in epoch 1. The maximum training accuracy is 99.90 and validation accuracy is 98.34. The maximum testing accuaracy is 98.27. For case 4 minimum training and validation accuracy is found in epoch 1. The maximum training accuracy is 99.54 and validation accuracy is 98.18. The maximum testing accuaracy is 98.07. For case 5 minimum training and validation accuracy is found in epoch 1.the maximum training accuracy is 99.88 and validation accuracy is 98.26. The maximum testing accuaracy is 98.17.

## 3.2 Libraries required to install

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 subpackage of SciPy is the package that has a huge amount of distance funtions 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.





We can implement Convolutional Neural Network using python or matlab. We use python to implement this project because we have our keras deep learning library built in python. By using keras model we can implement our network and create driver program to call the network to take input from the dataset. The driver program has learning algorithm, training and testing dataset. We use MNIST dataset to implement this network. MNIST datasets are the best and well known, and easily understood dataset in the computer vision branch and machine learning to use it as first dataset which we can use in our journey of deep learning. After implementing our network we can find that our network can classify the Handwritten English Digit up to >98 percent accuracy with less training time. This implementation can be done in both CPU or GPU enabled system, but CPU takes more training time than GPU. We use 60000 data for training the network and use 10000 data for testing the network. Each digit is taken as 28 by 28 grey scale image which are available from MNIST dataset and can be directly downloaded. These grey scale pixel intensities fall in the range of 0 to 255. All digits are presented on black background color with a light foreground color being white, the digit itself and include various shades of grey. To recognize the Handwritten English Digit a seven layer Convolutional Neural Network with one input layer, five hidden layers and one output layer. The input layer consists of 28 by 28 pixel image which mean that the networks contain 784 neuron as input data. The input pixels are gray scale with a value 0 for a white pixel and 1 for a black pixel. Here, this model of CNN has five hidden layers. The first hidden layer is Convolutional layer which is responsible for feature extraction for input data. The layer performs convolutional operation to small localize area by convolving a filter with the previous layer.

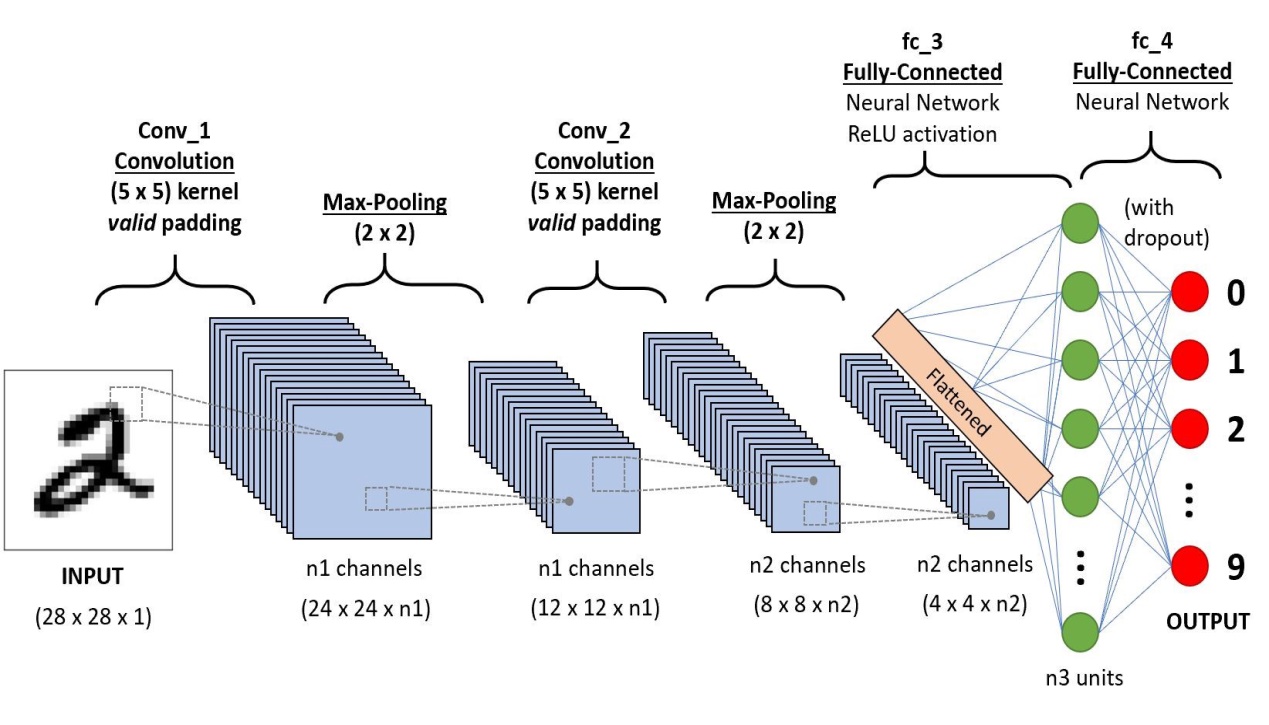


Fig 3.1: Convolutional Neural Network Model

In addition, it consists of multiple feature maps with learnable kernels and rectified linear units. The kernel size determines the locality of the filters. Rectified linear units are used as an activation function at the end of each convolution layer as well as a fully connected layer to enhance the performance of the model. The next hidden layer is the pooling layer 1. It reduces the output information from the convolution layer and reduces the number of parameters and computational complexity of the model. The different types of pooling are max pooling, min pooling, average pooling, and L2 pooling. Here, max pooling is used to subsample the dimension of each feature map. Convolution layer 2 and pooling layer 2 which has the same function as convolution layer 1 and pooling layer 1 and operates in the same way except for their feature maps and kernel size varies. A Flatten layer is used after the pooling layer which converts the 2D featured map matrix to a 1D feature vector and allows the output to get handled by the fully connected layers. A fully connected layer is another hidden layer also known as the dense layer. It is similar to the hidden layer of Artificial Neural Networks (ANNs) but here it is fully connected and connects every neuron from the previous layer to the next layer. In order to reduce over fitting, dropout regularization method is used at fully connected layer 1. It randomly switches off some neurons during training to improve the performance of the network by making it more robust. This causes the network to become capable of better generalization and less compelling to over fit the training data. The output layer of the network consists of ten neurons and determines the digits numbered from 0 to 9. Since the output layer uses an activation function such as softmax, which is used to enhance the performance of the model, classifies the output digit from 0 through 9 which has the highest activation value. The images that are used for training and testing the network all are the gray scale image with a size of 28×28 pixels. Character x is used to represent a training input where x is a 784-dimensional vector as the input of x is regarded as 28×28 pixels. The equivalent desired output is expressed by y(x), where y is a 10-dimensional vector. The aim of the network is to find the convenient wights and biases so that the output of the network approximates y(x) for all training inputs x as it completely depends on weight values and bias values. To compute the network performances, a cost function is defined, expressed by equation 1.

(1)

Where w is the cumulation of weights in the network, b is all the biases, n is the total number of training inputs and a is the actual output. The actual output a depends on x, w, and b. C(w, b) is non-negative as all the terms in the sum is non-negative. Moreover, C(w, b) =0, precisely when desired output y(x) is comparatively equal to the actual output, a, for all training inputs, n. To reduce the cost C(w,b) to a smaller degree as a function of weight and biases, the training algorithm has to find a set of weight and biases which cause the cost to become as small as possible. This is done using an algorithm known as gradient descent. In other words, gradient descent is an optimization algorithm that twists its parameters iteratively to minimize a cost function to its local minimum. The gradient descent algorithm deploys the following equations [25] to set the weight and biases.

(2)

(3)

However, the gradient descent algorithm may be unusable when the training data size is very large. Therefore, to enhance the performance of the network, a stochastic version of the algorithm is used. In Stochastic Gradient Descent (SDG) a small number of iteration will find effective solutions for the optimization problems. Moreover, in SDG, a small number of iteration will lead to a suitable solution. The Stochastic Gradient Descent algorithm utilizes the following equations.

(4)

(5)

To find the amount of weight that contributes to the total error of the network Backpropagation method is used. The backpropagation of the network is illustrated by the following equations.

(6)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Case** | **Number of Hidden Layers** | **Batch Size** | **Minimum**  **Training**  **Accuracy** | | **Minimum**  **Validation**  **Accuracy** | | **Maximum**  **Training Accuracy** | | **Maximum**  **Validation**  **Accuracy** | | **Overall**  **Performance**  **Validation**  **Accuracy** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | |
|  | | | | | | | | | | | |

# 

# 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.