

# Effect of Learning Rate on Neural Network and Convolutional Neural Network

Pankaj Singh Rathore<sup>1</sup>

<sup>1</sup>Department of Electronics and Communication,  
Vivekananda Institute of Technology,  
Jaipur (India)

Naveen Dadich<sup>2</sup>

<sup>2</sup>Department of Electronics and Communication,  
Vivekananda Institute of Technology,  
Jaipur (India)

Ankit Jha<sup>3</sup>

<sup>3</sup>Department of Applied Mathematics,  
Defence Institute of Advanced Technology,  
Pune (India)

Debasish Pradhan<sup>4</sup>

<sup>4</sup>Department of Applied Mathematics,  
Defence Institute of Advanced Technology,  
Pune (India)

**Abstract:-** The modern era is very much concentrated on the applications of the Machine learning mainly on neural network. Modelling and simulation of human intelligence processes by machine are the Artificial Intelligence (AI). Machine Learning is a technique of AI that gives the computational system the ability to learn with information without being explicitly programmed. Based on the applications, there are various types of neural network such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) etc. There are some important parameters which should be decided before designing a neural network such as the number of layers, the number of neurons per layer, epochs, learning rate, optimizers, etc. This paper examines the response of learning rate in training and testing the neural network model. The same can be seen verified with learning rate and loss in the model. This also implies that by choosing appropriate optimizer could provide the best possible result and a comparison made between the learning rate of Neural Network and Convolutional Neural Network.

**Keywords:** Loss, Learning Rate, Machine Learning, Neural Network, TensorFlow.

## I. INTRODUCTION

The neural networks are a set of algorithms which help in recognizing patterns. According to one perspective the pattern can be recognized viz. pattern associated with the change in temperature of the environment, cipher text in cryptology, inventory, spam detection, handwritten character recognition, etc. The Figure1 shows the typical block diagram of Neural Network's image recognition. Multilayer feedforward artificial neural network (ANN) is used in recognition of images. The removal of noise, cropping, resizing, thinning, normalization, etc. are the various image processing techniques. Sometimes binarization [1] of image has been used to get an effective model. There are many websites from where we can download the standard datasets and work on them such as ALEXNET, MNIST, ImageNet, CIFAR10, etc.

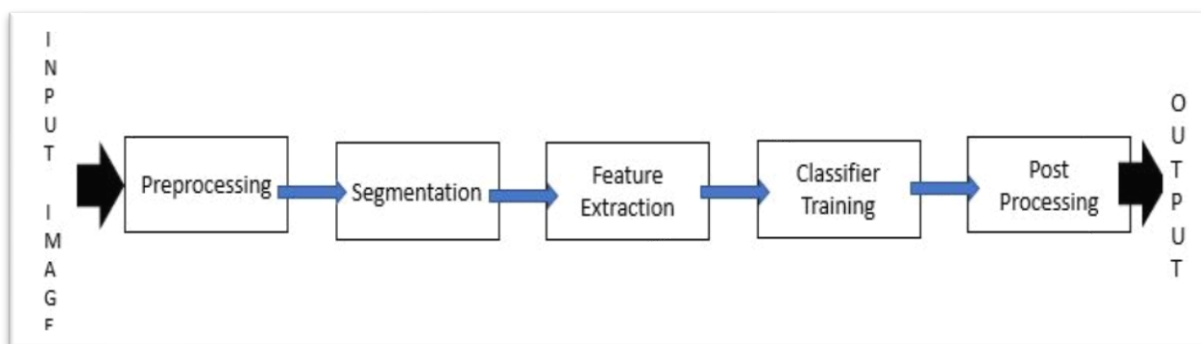


Figure1. The typical block diagram Image Recognition.

Earlier a few software was available for NN implementation. But the scenario is completely different, there are various software is available such as Python, WEKA Tool, MATLAB, etc. to handle the NN. Along with this TensorFlow, Keras, Theano, OpenCV, etc. open sourced libraries are available which are very much helpful in NN implementation. This paper work is carried out on the MNIST handwritten digits dataset, that contain 60K training examples and 10K testing examples with fixed image size of 28 X 28 i.e. 784 pixels and implemented on Python using libraries of TensorFlow.

The extension of the Neural Network in the field of image classification/recognition can be seen by the evolution of Convolutional Neural Network (CNN) [3]. It helped in training and test the model with great accuracy. The

various blocks of CNN model which can be seen in Figure2.

A. NEURAL NETWORK

The artificial neurons are interconnected in a system to share the messages. This system is known as the Neural Network (NN). The basis of a NN mainly consists of three layers: an input layer, an output layer and the most important hidden layer as shown in Figure 2. Here  $X_1, X_2$  and  $X_3$  are the inputs to the input layer while  $Y_1$  and  $Y_2$  are the outputs of the output layer obtained using the hidden layer unit consisting of  $h_1, h_2, h_3$  and  $h_4$ . The hidden layers may consist of several layers depend upon the available computational power and accuracy user wants. The interconnected system consists of weights  $W_i$  (i.e.  $W_0, W_1, W_2, W_3 \dots$ ) and biases  $b_i$  (i.e.  $b_1, b_2, b_3 \dots$ ). The bias unit adjusts the weights of the input layer.

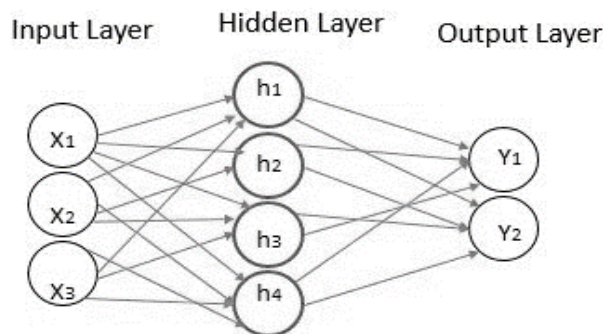


Figure 2: The Artificial Neural Network.

Mathematically, the Neural Network model is represented by equation (1). It consists of input unit, output unit and the activation function. The use of a NN is just like using a black box in which the system will model itself according to the input and the output fed to the system. The model will be tested on the new input fed to the system and

subsequently the accuracy is measured. The output of the NN completely depends upon the threshold function used. There are different types of activation or threshold functions such as sigmoid, unit step, hyperbolic tangent (Tanh) functions, etc.

$$y = \sum x_i w_i + b \quad \text{Equation (1)}$$

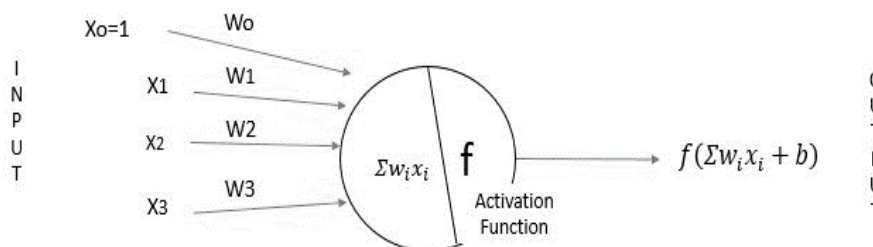


Figure 3: The Mathematical Model of Neural Network.

CONVOLUTION NEURAL NETWORK

Convolutional Neural Network is a type of Neural Network which consists of one or more convolutional layers with

subsampling layer and followed by one or more fully connected layers. When the input layer convolves with different types of filters such as noise removal filter, image resizing, color conversion etc. produces a layer of

convolutions, formally known as convolutional layers. Feature extraction consists of Rectified Linear Unit (ReLU), Max Pooling, Convolutional layers, etc. Classifier

consists of Flattening, Fully Connected and SoftMax layers.

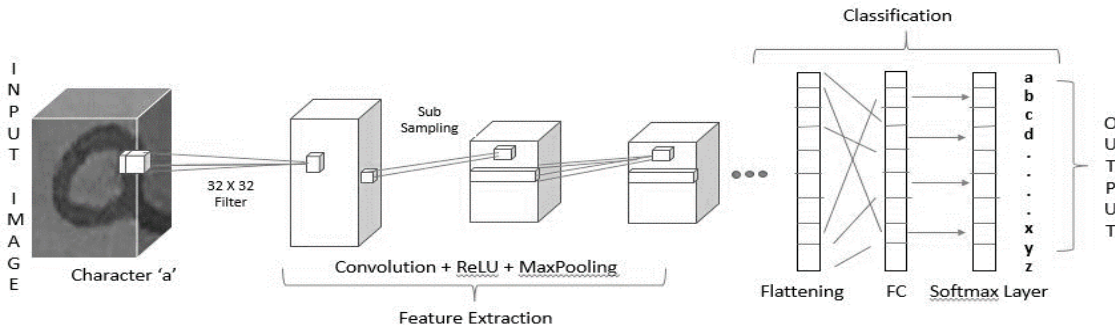


Figure 4: A Simple CNN Block Diagram

The mathematical functions used as the activation functions in the neural network are:

▪ *Sigmoid Function*

For logistic regression and neural network implementations, the sigmoid function has been widely used.

$$f(x) = \frac{1}{1+e^{-x}} \quad \text{Equation (2)}$$

Where x is the input from the input layer.

▪ *Rectified Linear Unit (ReLU)*

Some of the deep learning networks use rectified linear units (ReLU) for hidden layers. It has output 0 if the input is less than 0, and raw output otherwise. The ReLU are the simplest non-linear activation functions.

$$f(x) = \max(x, 0) \quad \text{Equation (3)}$$

Where x is the input from the hidden layer.

▪ *Softmax*

To deal with the classification problems, the sigmoid and the ReLU activations cannot help in an effective way. So SoftMax function can resolve the issue to a very extent. The output of the SoftMax function is equivalent to a categorical probability distribution, it tells us the probability that any of the classes are true.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^n e^{z_k}} \quad \text{Equation (4)}$$

where z is the value associated with the output of the hidden layer to get the output in the form of probabilities i.e. it will lie in between 0 and 1.

*PRE-PROCESSING*

Pre-processing removes the variability present in the images for image recognition/classification, data missing. There are various data missing pre-processing techniques available such as grayscale conversion, morphological operations (dilation and filling), binarization, image resizing and cropping, etc.

Grayscale conversion and binarization are the common but the most important image processing steps. But for data missing, mean and median of the sample comes into

picture. Generally, the missing data are replaced by mean of the sample data (training data/ testing data).

*FEATURE EXTRACTION*

It is a very important section of the image classification/recognition, text recognition, etc. It involves reducing the number of resources required to describe a large set of data. This decreases the computational power and time of the system. The features can be extracted in such a way that the feature data/vector can effectively able to define the real data. There are various methods of feature extraction some of them are as follows: boundary extraction, thresholding, blob detection, scale-invariant feature transform (SIFT), the histogram of oriented gradients (HOG), local binary patterns, speeded-up robust features (SURF), etc. After extracting the features and flattening of the, the featured vector will be directly fed to the neural network to train the network.

II. TYPES OF LEARNING EXAMPLES

The network learns by adjusting the weights and biases connections between network's neurons. The Neural Networks can be trained using different learning algorithms. Among such three major learning algorithms are:

▪ *Supervised Learning*

The Neural Network is provided with both input as well as the output. An error is being calculated based on the difference of target output and the actual output. The network updates its weights and biases according to the calculated error.

▪ *Unsupervised Learning*

It is mainly used in data mining. The network is provided with the sets input data only. Now it is the responsibility of the network to find the pattern associated with the input data.

▪ *Reinforcement Learning*

It is similar to supervised learning with the difference of guerdon that is given in feedback in place of target output

based on system performance. This strongly relates how the learning works in the environment. The main aim of reinforcement learning is to maximize which the guerdon the system receives via the trial-and-error method.

In brief, for different problems, the different type of algorithm is used. The Artificial Neural Network(ANN) learning algorithm is mathematically dedicated algorithm which modifies the weights and biases of the neuron at each iteration. In supervised or unsupervised learning, the learning algorithms and learning rule can be used. While building a predictive model it is required to have a well-defined training and validation customs to follow so that it can produce an accurate and robust results. Problems like learning rate and computation volumes exceed arises due to the large size of samples and the numbers of layers and neurons in the network architecture. Along with these the configuration of the system also depends. This work has been completed on a CPU based system with the configuration of Intel(R) Core(TM) i-3 2310M CPU @ 2.10Ghz processor. The strength and excellence of the neural network depends on the factors stated above.

### III INTENTION

As previously stated about the NN and CNN, they are the tools to handle the non-linear data which is now implemented in python with the libraries of TensorFlow. In this paper, there is a discussion of choosing learning rate for NN and CNN and shows the difference in the testing accuracy at same learning rate to both neural network and convolutional neural network. The following steps were

taken to develop the network: Data collection, Data extraction and Pattern Classification. The paper [2] by Igiri et.al shows the effective prediction of 80% at the learning rate of 0.1 and 90% prediction at the learning rate of 0.8.

The data has been imported from the MNIST database of handwritten digits. They are further extracted and feed into the graph input of TensorFlow. The placeholder (a variable used to assign the data at the later stage) of TensorFlow is used for both input data(28X28) and labelled data (10 classes). The weights and biases are set using the tensors known as Variables. While constructing a model using the TensorFlow, all the operations (shortly ops) are being encapsulated into scopes by calling name\_scope(). The ops are the nodes in the graph to perform computation and produce the Tensors.

The SoftMax, *Error (using cross entropy), Gradient Descent Optimizer, Accuracy functions are always defined under name\_scope(). To start the training in TensorFlow, Sessions are created. Backpropagation and cost functions are defined for the optimization and loss operations respectively under the sessions. The last step of this NN is to evaluate of the model (how accurate the model is?). There is a unique feature of TensorFlow, its visualization using TensorBoard. But to use the TensorBoard, log directory should be created and the operations(op) to write the log in it with the help of Summary.FileWriter(). This section shows an example of TensorFlow visualization. The Nodes of the Graph can be easily seen in Figure5, whereas Figure5 represents the accuracy and loss of the network with the number of iterations.*

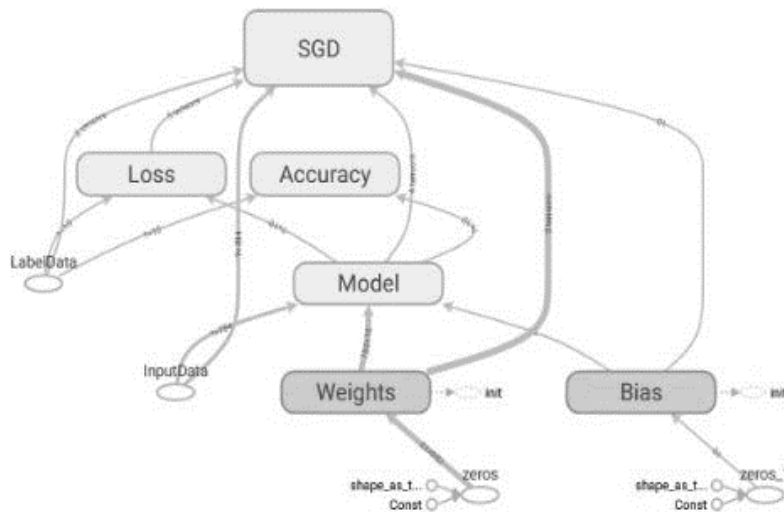


Figure5 Main Graph of the Neural Network Model.

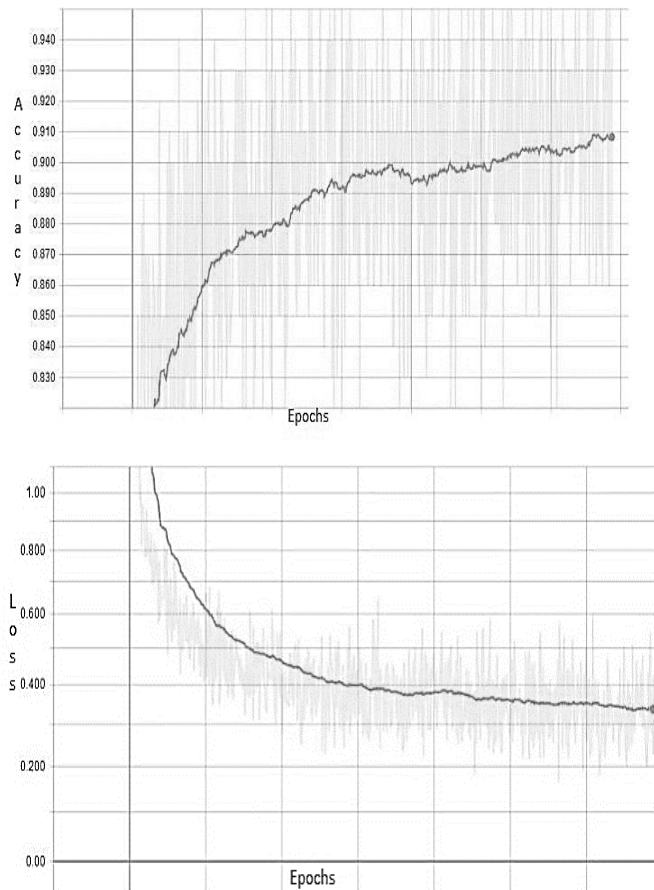


Figure6 Graph of Accuracy vs Epochs and Loss vs Epochs.

Now, the task is to study the effect of learning rate on accuracy and loss of the model. This is done by changing the learning rate values, the accuracy and loss were noted down.

IV RESULT AND CONCLUSION

Under this section, calculation for accuracy and loss done for the respective learning rate. While performing the task on MNIST dataset over the simple Neural Network some parameters kept fixed and are listed below in Table1. Similarly, the same task performed using the Convolutional Neural Network.

S. No.	Parameters	Values
1.	Number of Hidden Layers	2
2.	Number of Neurons in Hidden Layer 1	500
3.	Number of Neurons in Hidden Layer 2	500
4.	Batch Size	128

Table1 Fixed Neural Network Parameters.

Firstly, a Table2 drawn to test the Neural Network accuracy and loss, whereas Table3 drawn to test the performance of data over the Convolutional Neural Network. From the Table2 and Figure7, it can be clearly seen that if the model

is trained with a very high or very low learning rate values, the model responses with very low accuracy and huge loss. At the same time, the model performed effectively for the learning rate values between 0.001 to 0.2.

Learning Rate	Testing Accuracy	Loss
0.00001	0.16	2.36
0.00009	0.43	1.89
0.0001	0.44	1.87
0.0002	0.63	1.58
0.0005	0.77	1.14
0.0009	0.81	0.86
0.001	0.83	0.79
0.005	0.8922	0.41
0.007	0.8951	0.37
0.01	0.9	0.34
0.05	0.9182	0.29
0.09	0.9165	0.28
0.1	0.9165	0.3
0.2	0.9136	0.33
0.25	0.906	0.36
0.27	0.83	0.8
0.28	0.8	1.01
0.29	0.78	1.35
0.3	0.76	1.7
0.4	0.65	2.3

Table2 Learning Rates in Neural Network

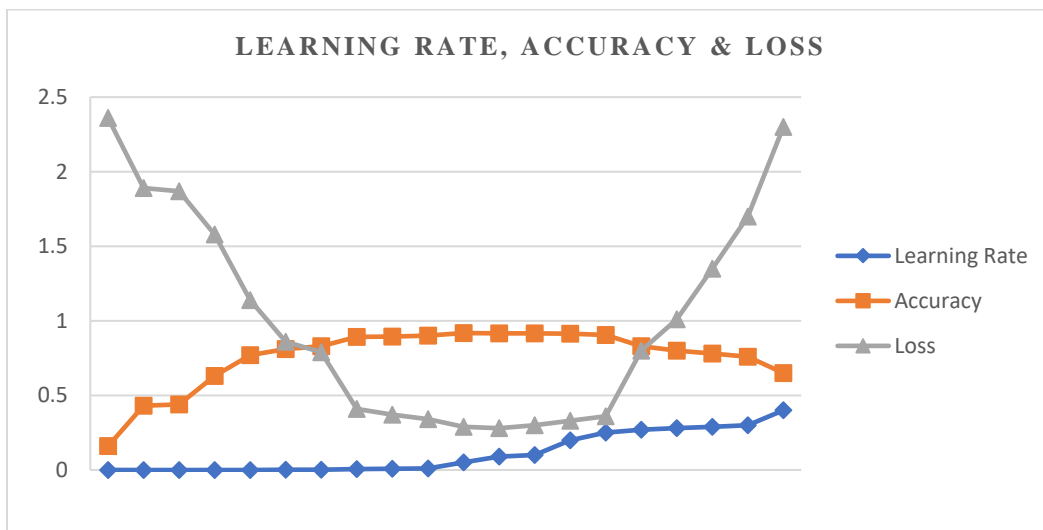


Figure7 Relationship between Learning Rate, Accuracy and Loss of the Neural Network

S. No.	Parameters	Values
1.	Number of Convolutional Layers	2
2.	Number of MaxPooling Layer	2
3.	Number of Convolutional Filters with Kernel Size	32, 5
4.	Flattening with Fully Connected Layer	1, 1
3.	Batch Size	128
4.	Number of Epochs	2000
5.	Dropout	0.75



Table2 Fixed Convolutional Neural Network Parameters.

Learning Rate	Test Accuracy	Loss
0.00001	0.9838	0.02
0.00009	0.9925	0.035
0.0001	0.9865	0.06
0.0002	0.9872	0.05
0.0005	0.9869	0.044
0.0009	0.9914	0.04
0.001	0.9912	0.035
0.002	0.9903	0.043
0.003	0.9849	0.071
0.004	0.9832	0.09
0.005	0.9762	0.112
0.006	0.9835	0.09
0.007	0.9563	0.25
0.008	0.9672	0.2
0.05	0.1135	2.44
0.07	0.1032	2.5
0.09	0.1028	2.45
0.1	0.1	2.63
0.2	0.101	4.79

Table2 Learning Rates in Convolutional Neural Network

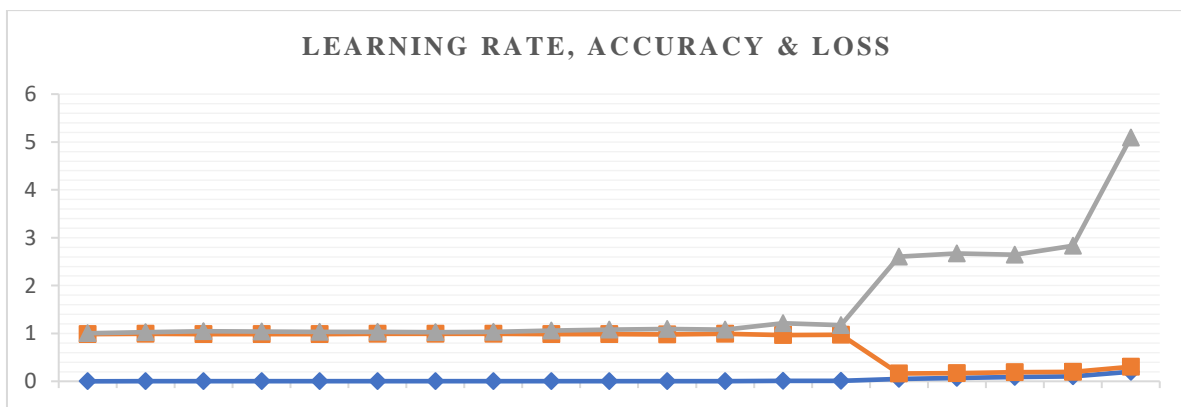


Figure8 Relationship between Learning Rate, Accuracy and Loss of the Convolutional Neural Network

The model shows very high accuracy at lower learning rates and shows poor responses at high learning rates. The dependency of network performance on learning rate can be clearly seen from the Figure7 and Figure8. A plot (Figure9) has been created that shows the difference in the accuracies of simple Neural Network and Convolutional Neural Network model. Some conclusions can be drawn

from the above figures and tables: lower learning rates are preferred when the number of layers are more. Because lower learning rate overcomes the overfitting problem of the network which has more numbers of layers (CNN). The Figure9 also shows that the accuracy of the Convolutional Neural Network model is much higher than the simple Neural Network model.

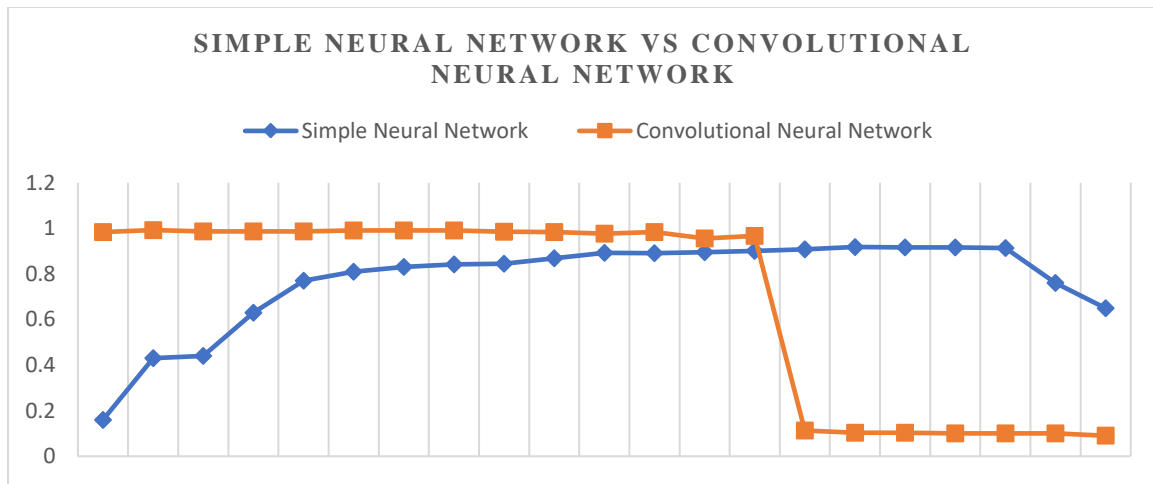


Figure9 Accuracy Between Simple Neural Network and Convolutional Neural Network

## V. REFERENCES

- [1]. Amit Choudhary, Rahul Rishi, Savita Ahlawat, *Off-line Handwritten Character Recognition using Features Extracted from Binarization Technique*, AASRI Conference on Intelligent Systems and Control 2013.
- [2]. Igiri Chinwe Peace, Anyama Oscar Uzoma, Silas Abasiama Ita, *Effect of Learning Rate on Artificial Neural Network in Machine Learning*, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181, IJERTV4IS020460, Vol. 4 Issue 02, February-2015.
- [3]. Xin Jia, *Image Recognition Method Based on Deep Learning*, CCDC 2017.
- [4]. Rejean Plamondon, Sargur N. Srihari, *On-Line and Off-Line Handwriting Recognition: A Comprehensive Review*, IEEE Transaction on Pattern Analysis and Machine Intelligence January 2000.
- [5]. Ankit Sharma, Dipti R Chaudhary, *Character Recognition Using Neural Network*, IJETT 2013.