

PROJECT REPORT

on

Handwritten Digit

Classification

(CSE 6th Semester Project)

2021-2022



Name: Ruchi Sharma

Section: E

University Roll No: 2014447

Class Roll No: 33

Session: 2021-2022

Evaluator:

Ms. Parul Madan

Mr. Pankaj Kumar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA DEEMED TO BE UNIVERSITY,
DEHRADUN

INTRODUCTION:

- ▶ Scientists believe that the most intelligent device in a person is the human brain.
- ▶ There is no computer which can beat the level of efficiency of the human brain.
- ▶ These inefficiencies of the computer has led to the evolution of the “Artificial Neural Network”.
- ▶ They differ from conventional systems in the sense that rather than being programmed these system learn to recognize the pattern.

MAIN GOALS and APPLICATIONS:

- ▶ Handwritten Digit Classification is used to recognize the hand drawn digits given as input by the user.
- ▶ A handwritten digit classification system is used to visualize artificial neural networks.
- ▶ It is already widely used in the automatic processing of bank cheques, postal addresses, in mobile phones etc.

MOTIVATIONS:

Machine Learning is a part of Data Science and being interested in everything having a relation with the Machine Learning, the independent project was a great occasion to give me the time to learn

and confirm my interest for this field. The fact that we can make perform classification and predictions give the ability for machines to learn by themselves is both powerful and limitless in term of application possibilities that's why I decided to conduct my project around Machine Learning.

ABOUT PROJECT:

I wanted to make my project as much informative as possible by approaching every different steps of the machine learning process and trying to understand them deeply. I chose to take the Neural Network approach for my Handwritten Digit Classification model. This model is trained on a dataset with a large variety of handwritten digits and can recognize the hand-drawn digits from the user given as input.

SOFTWARE REQUIREMENTS:

- Jupyter Notebook
- Libraries (NumPy, Matplotlib, Tensorflow, Keras, os, cv2)

HARDWARE REQUIREMENTS:

- 2 GHz Intel or high processor • Minimum of 180 GB HDD • At least should have 2 GB RAM

LANGUAGE USED:

- Python

LIBRARIES USED:.

Numpy –

NumPy is a Python Programming Language library that is used to provide us a simple yet powerful data structure and is also used to perform a number of mathematical operations on arrays.

Matplotlib. pyplot –

It is a plotting library used in Python Programming Language and it is used to provide an object oriented API for displaying bar plots.

Tensorflow--

Tensorflow is an open-source library for numerical computation and large-scale machine learning that ease Google Brain TensorFlow, the process of acquiring data, training models, serving predictions, and refining future results. Tensorflow bundles together Machine Learning and Deep Learning models and algorithms.

Open CV--

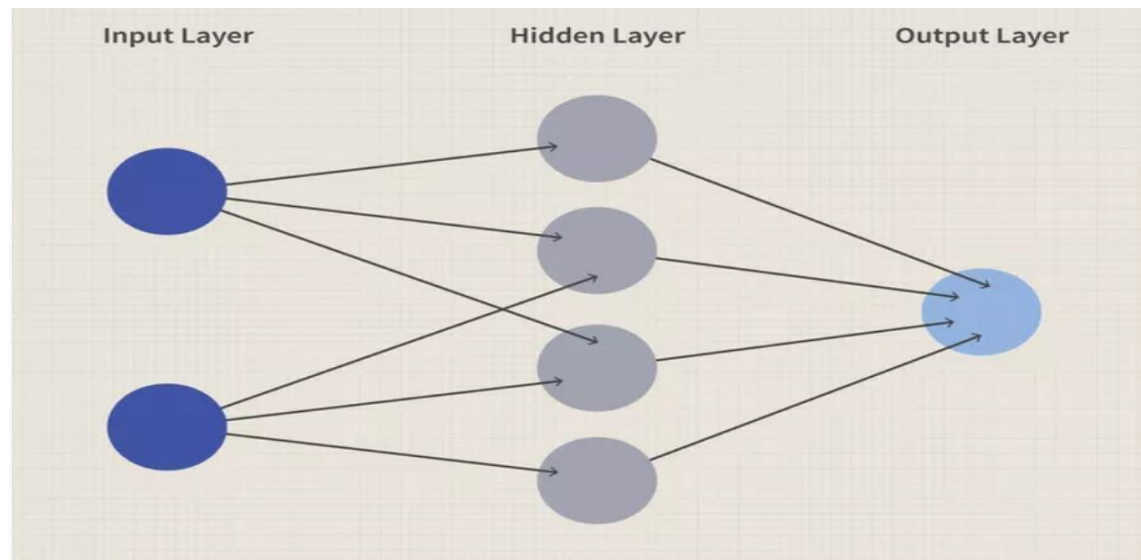
OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human.

Neural Network:

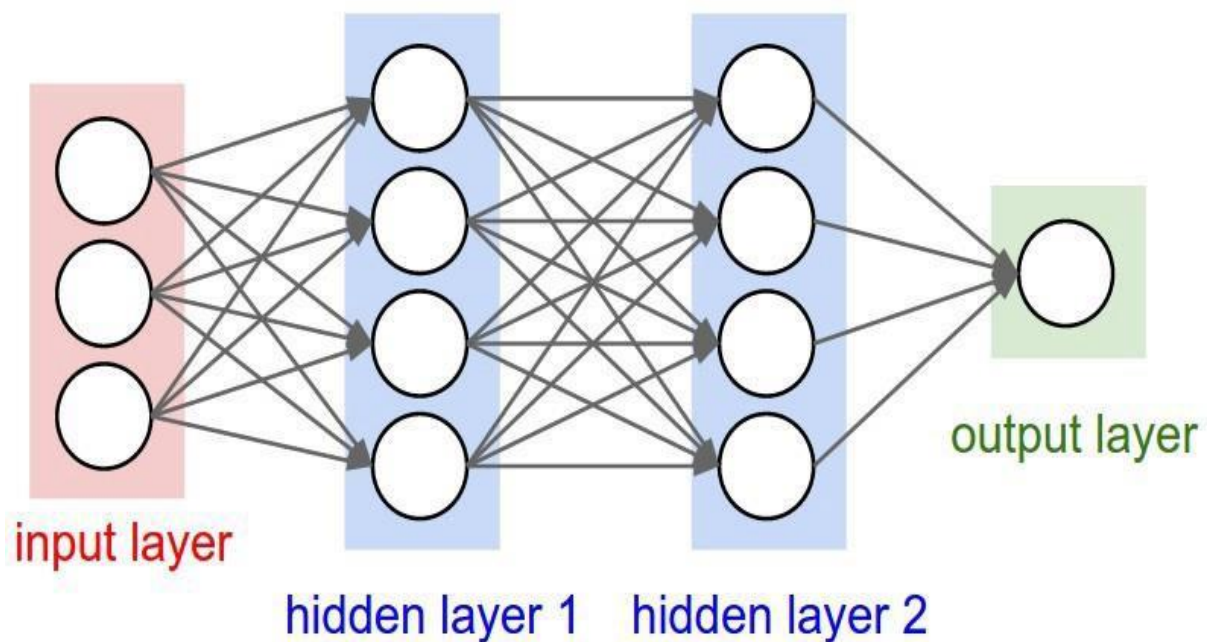
A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature.

A neural network contains layers of interconnected nodes. Each node is known as perceptron and is similar to a multiple linear regression. The

perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear.



(Diagrammatic Representation of a Neural Network with one hidden layer)



(Diagrammatic Representation of a Neural Network with multiple hidden layers)

LAYERS IN NEURAL NETWORK:

The layer is a group, where number of neurons together and the layer is used for the holding a collection of neurons. Simply we can say that the layer is a container of neurons. In these layers there will always be an input and output layers and we have zero or more number of hidden layers. The entire learning process of neural network is done with layers. In this the neurons are placed within the layer and that each layer has its purpose and each neuron perform the same function.

Different layers in Neural Networks:

Input Layer:

The input layer is the most responsible layer for receiving the inputs and these inputs are loaded from some external sources like csv file or web service etc. In neural networks we must maintain one input layer to takes the inputs and perform some calculations through its neurons and then the output is transmitted to the next layers.

Output Layer:

The output layer is mostly responsible for producing the final output results. There must be always an output layer in the neural networks. The output layer takes the inputs which are passed in from the layers before it, and performs the calculations through its neurons and then the output is computed.

Hidden Layer:

In neural networks, a hidden layer is located between the input and output of the algorithm, in which the function applies weights to the inputs and directs them through an activation function as the output.

To improve the efficiency of our model, we can increase the number of hidden layers, where the output of one hidden layer serves as the input of next hidden layer, with the increase in number of hidden layers the complexity of our model also increases thereby it takes more time in the training process (also depends on the number of epochs taken).

Using Deep Learning (Neural Network) for Classification:

In this machine learning project, the dataset that is used for the purpose of this project has been taken from Kaggle for my handwritten digit classification model using Neural Network.

The Dataset:

- ▶ Training of this model is done by a dataset named MNIST in CSV which is taken from Kaggle.
- ▶ MNIST dataset has a training set of 60,000 examples, and a test set of 10,000 examples.
- ▶ All the images in the dataset are of 28x28 pixels.

Training Model:

- ▶ Train the model on the training data set(xtri and y_trn). I will iterate 10 times over the entire dataset to train on and after this will store the data.

```
Epoch 1/10
1875/1875 [=====] - 58s 31ms/step - loss: 2.0615 - accuracy: 0.2059
Epoch 2/10
1875/1875 [=====] - 60s 32ms/step - loss: 1.3954 - accuracy: 0.4712
Epoch 3/10
1875/1875 [=====] - 57s 31ms/step - loss: 0.5319 - accuracy: 0.8300
Epoch 4/10
1875/1875 [=====] - 58s 31ms/step - loss: 0.1989 - accuracy: 0.9441
Epoch 5/10
1875/1875 [=====] - 58s 31ms/step - loss: 0.1270 - accuracy: 0.9633
Epoch 6/10
1875/1875 [=====] - 59s 32ms/step - loss: 0.0947 - accuracy: 0.97220s
Epoch 7/10
1875/1875 [=====] - 58s 31ms/step - loss: 0.0759 - accuracy: 0.9777
Epoch 8/10
1875/1875 [=====] - 66s 35ms/step - loss: 0.0598 - accuracy: 0.9823
Epoch 9/10
1875/1875 [=====] - 62s 33ms/step - loss: 0.0525 - accuracy: 0.9840
Epoch 10/10
1875/1875 [=====] - 67s 36ms/step - loss: 0.0453 - accuracy: 0.9865
```

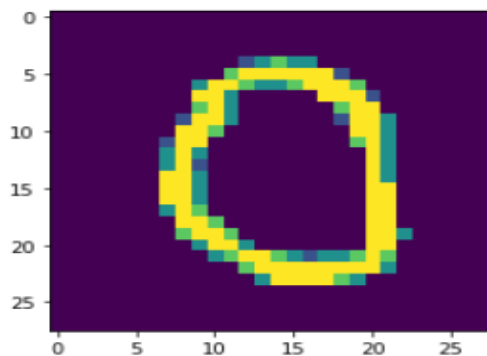
Testing Model:

- Then we will predict any random number to check whether the required result is coming or not. Like in the below photo we're predicting the 459th value i.e 0 and we can see that accuracy in the 0th position is maximum so the number is 0.

```
prd=model.predict(np.array([xtsi[459]]))
```

```
plt.imshow(xtsi[459])
```

```
<matplotlib.image.AxesImage at 0x2518b0d71c8>
```



```
prd
```

```
array([[9.9959356e-01, 8.9965104e-14, 2.1719182e-13, 1.4187754e-15,  
        5.1909355e-10, 8.7063859e-07, 4.0558237e-04, 5.3898355e-11,  
        9.0830132e-10, 6.0522916e-09]], dtype=float32)
```

Conclusion:

This model presented the usage of Deep Learning (Neural Networks) for classifying a given handwritten digit. Handwritten digit recognition is the first step to the vast field of Artificial Intelligence and Computer Vision. As seen in the model, CNN proves to be very effective in classification problems.

The results can be made more accurate with more convolution layers and more number of hidden neuron.