CSE 574 MACHINE LEARNING

Project 1

Handwritten Digits Classification using Neural Networks

Archita Pathak (50207253)
Venkata Gayatri Pratyusha Gundugola (50204847)
. Saani Ausaf (50207373)
Group No. 39

OBJECTIVE

- ✓ Implement a Multilayer Perceptron Neural Network and evaluate its performance in classifying handwritten digits.
- ✓ Use the same network to analyze a more challenging face dataset.
- ✓ Compare the performance of the neural network against a deep neural network using the TensorFlow library.

DESCRIPTION

Neural network can be graphically represented as in Figure 1. As depicted below, there are totally 3 layers in the neural network:

• The first layer comprises of (d + 1) units, each represents a feature of image (there is one extra unit representing the bias).

• The second layer in neural network is called the hidden units. In this document, we denote m + 1 as the number of hidden units in hidden layer. There is an additional bias node at the hidden layer as well. Hidden units can be considered as the learned features extracted from the original data set. Since number of hidden units will represent the dimension of learned features in neural network, it's our choice to choose an appropriate number of hidden units. Too many hidden units may lead to the slow training phase while too few hidden units may cause the under-fitting problem.

• The third layer is also called the output layer. The value of lth unit in the output layer represents the probability of a certain hand-written image belongs to digit l. Since we have 10 possible digits, there are 10 units in the output layer. In this document, we denote k as the number of output units in output layer.

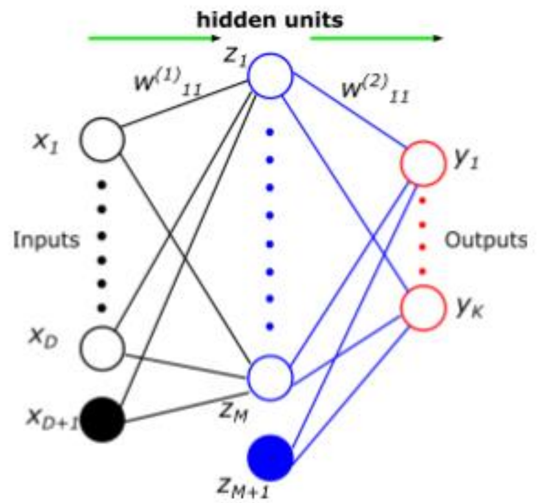


Figure 1: Neural network

SIMULATION & RESULTS

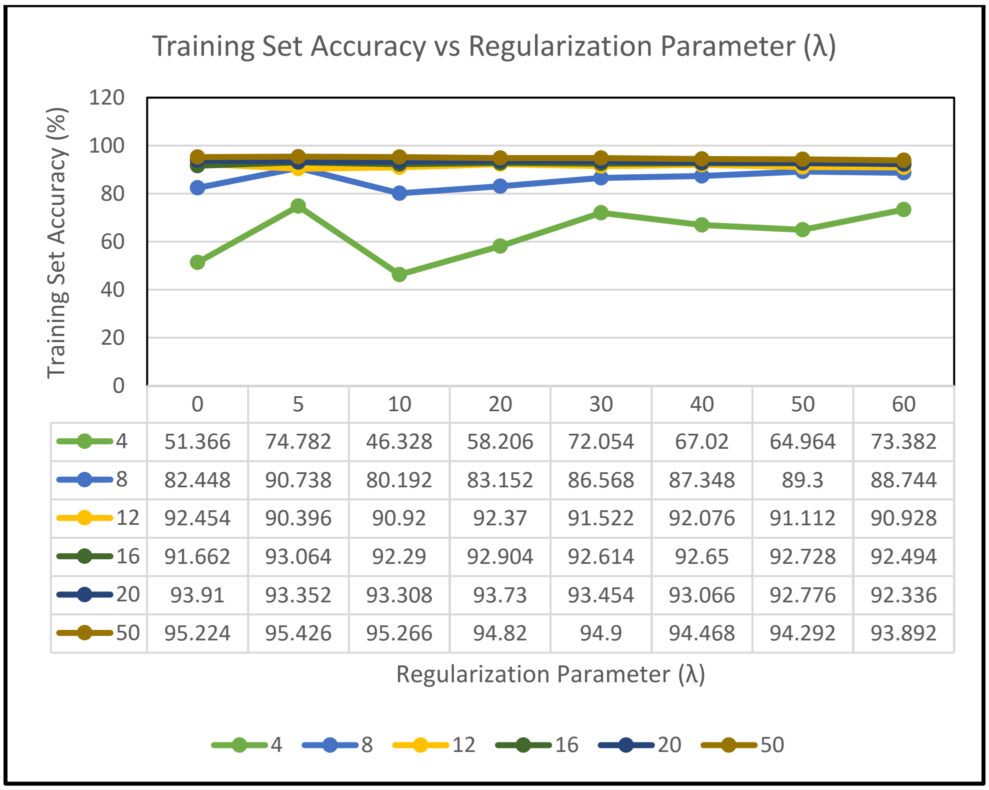
TASK I – Handwritten Digits Classification using Neural Networks

Preprocessing of data –

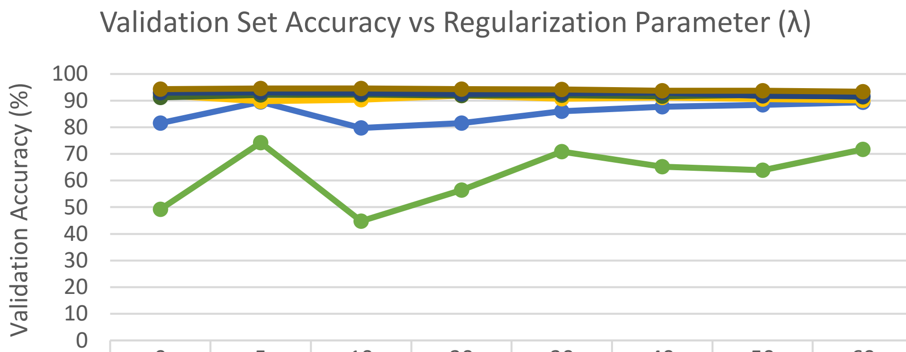
During the preprocessing process, the training data set was divided into two subsets: real training data set and validation data set. The real training data set and validation data set have around 50,000 (randomly sampled examples) and 10,000 (validation set to estimate hyper-parameters of the network) samples respectively with their truth values.

Feature extraction –
Feature selection is meant to be an important step in the classification process. Its aim is to reduce the number of features and at the same time to try to maintain or even improve the performance of the program. In this assignment, a set of 719 features is obtained after the feature extraction.

Determining the best-case hyper-parameters –
The neural network was experimented for different values of hyper-parameters and hidden units in order to determine the influence of these parameters on the network performance. The evaluation has been initially performed using 4 hidden units and varied lambda value (regularization parameter) from 0 to 60, in the intervals [0,5,10,20,30,40,50,60]. Further, the number of hidden units were increased gradually to the values [8, 12, 16, 20, 50], while continuing to vary lambda in same pattern. The training set accuracy, validation set accuracy, test set accuracy and network training time have been recorded simultaneously for all the above cases and plotted against different hyper-parameters and hidden units. Based on the results, the parameters producing the best results have been determined. The recorded results have been depicted below.



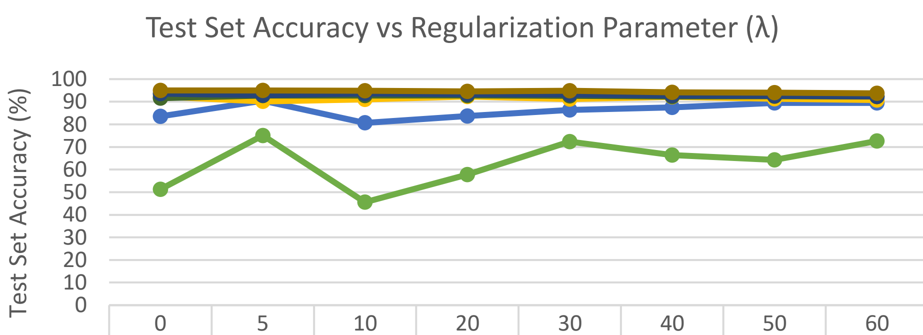
	0	5	10	20	30	40	50	60
4	51.366	74.782	46.328	58.206	72.054	67.02	64.964	73.382
8	82.448	90.738	80.192	83.152	86.568	87.348	89.3	88.744
12	92.454	90.396	90.92	92.37	91.522	92.076	91.112	90.928
16	91.662	93.064	92.29	92.904	92.614	92.65	92.728	92.494
20	93.91	93.352	93.308	93.73	93.454	93.066	92.776	92.336
50	95.224	95.426	95.266	94.82	94.9	94.468	94.292	93.892



	0	5	10	20	30	40	50	60
4	49.27	74.24	44.77	56.4	70.91	65.19	63.92	71.68
8	81.54	89.55	79.74	81.61	85.96	87.72	88.45	89.43
12	92.02	89.8	90.36	91.89	90.66	91.15	90.63	90.1
16	91.24	92.14	92.29	91.89	91.99	91.68	92.11	91.46
20	92.99	93.11	92.92	92.54	92.73	92.95	91.78	91.64
50	94.29	94.53	94.58	94.27	94.25	93.68	93.74	93.35

Regularization Parameter (λ)

4 8 12 16 20 50



	0	5	10	20	30	40	50	60
4	51.26	75.08	45.55	57.82	72.29	66.36	64.26	72.61
8	83.57	90.46	80.63	83.67	86.35	87.45	89.44	89.43
12	92.32	90.12	91.03	92.24	91.15	92.13	91.33	90.71
16	91.56	92.68	92.71	92.79	92.52	92.34	92.44	92.35
20	93.54	92.94	93.6	93.34	93.11	92.69	92.57	92.33
50	94.96	94.99	94.85	94.55	94.79	94.19	94.01	93.71

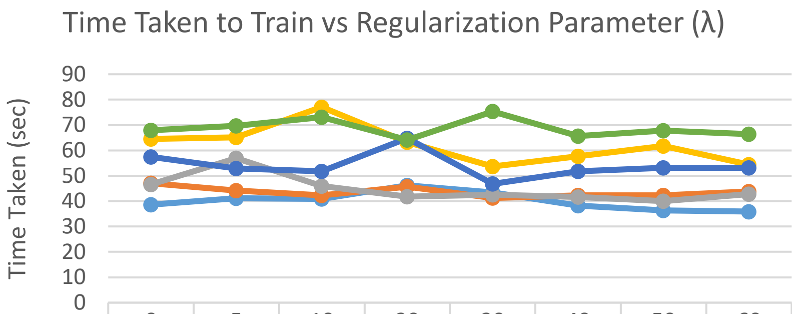
Regularization Parameter (λ)

4 8 12 16 20 50

From the aforementioned results, we can deduce that the highest accuracy is obtained with regularization parameter, $\lambda = 5$ and number of hidden units = 50. The accuracies obtained with these values are reported below –

Training Set Accuracy – *95.426%*, Validation Set Accuracy – *94.53%*, Test Set Accuracy – *94.99%*.

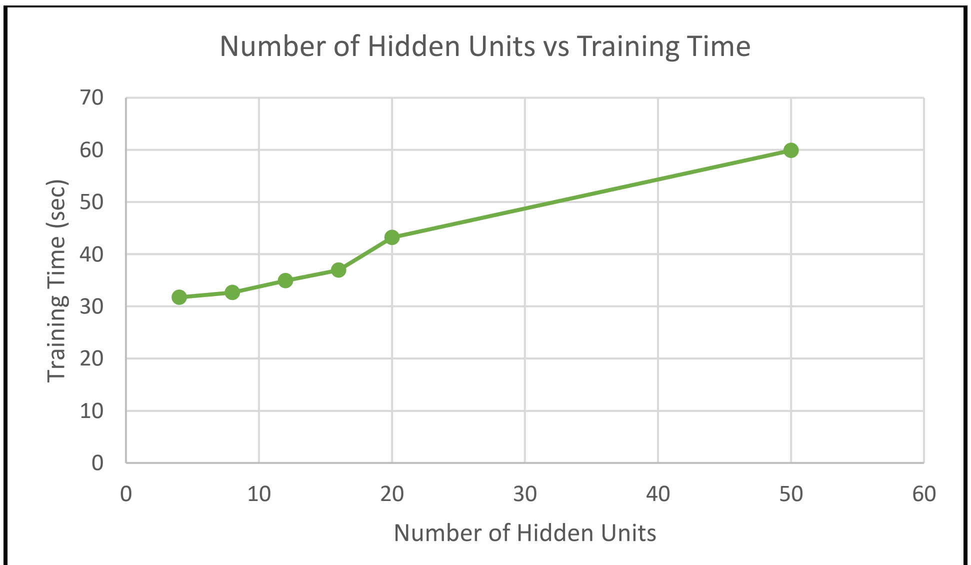
Evaluating the relationship between the time taken to train and regularization parameter –

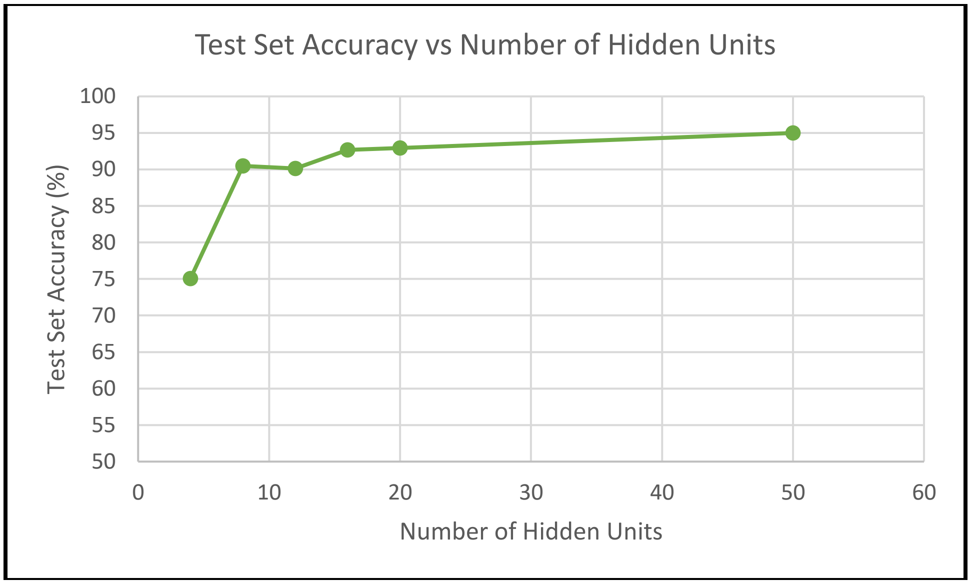


	0	5	10	20	30	40	50	60
4	38.569	41.139	40.956	46.216	43.44	38.232	36.406	35.902
8	47.05	44.19	42.31	45.633	41.237	42.23	42.32	43.797
12	46.6	56.91	45.908	41.744	42.5	41.63	40.01	42.83
16	64.496	65.15	77.061	63.27	53.66	57.74	61.76	54.35
20	57.498	52.898	51.79	64.78	46.81	51.72	53.194	53.09
50	67.917	69.677	73.13	64.11	75.31	65.67	67.76	66.41

The above figure displays the relationship between the time taken to train the data against the regularization parameter (λ) with increasing number of hidden units. It can be deduced that, as the number of hidden units increases, the time taken to evaluate also increases. This is because, as the number of hidden units increases, the gradients and weights computations also increase which directly enhance the computational complexity of the network. Hence, a relatively longer time is required to converge to an optimal solution.

Considering Regularization Parameter (λ) = 5, the following observations were made.





Hence, it can be concluded that with Regularization Parameter (λ) = 5, best accuracy is achieved when the number of hidden units were 50. Also, as mentioned earlier, as the number of hidden units increases, the time taken to train also increases. Hence, training time is maximum when the program was evaluated with 50 hidden units.

<u>TASK II – Analyzing the CelebA Data Set using Neural Network</u>

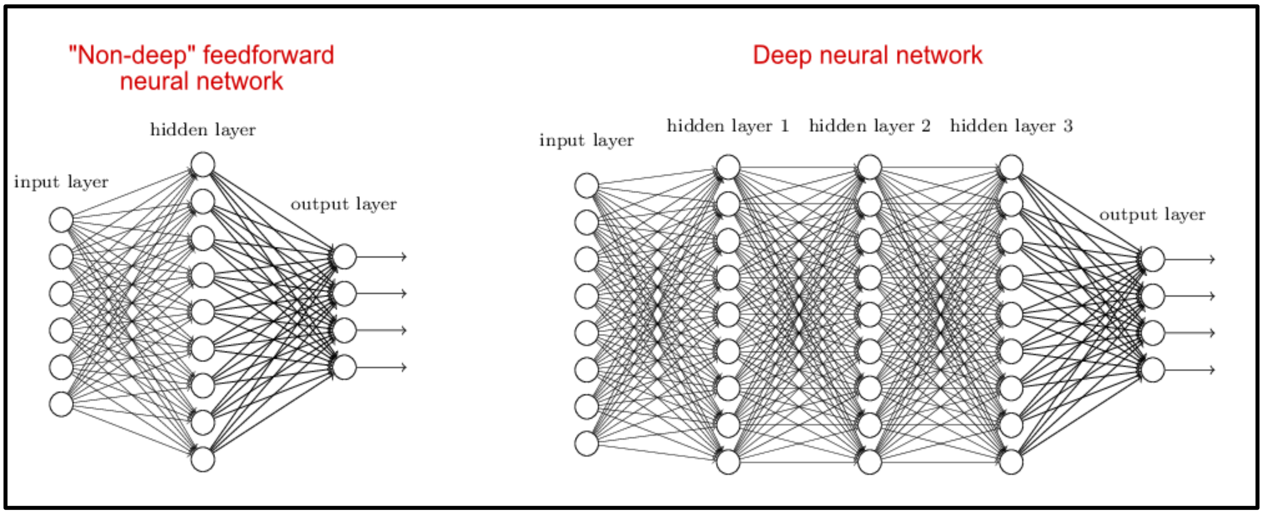
CelebFaces Attributes Dataset is a large-scale face attributes dataset with more than 200K celebrity images. CelebA has large diversities, large quantities, and rich annotations, including:
- 10,177 number of identities,
- 202,599 number of face images, and
- 5 landmark locations, 40 binary attributes annotations per image.

For this programming assignment, we are provided a subset of the images. The subset consists of data for 26407 face images, split into two classes. One class contains images in which the individual is wearing glasses and the other class contains images in which the individual is not wearing glasses. Each image is a 54×44 matrix, flattened into a vector of length 2376.
On data analysis using the "CelebA Data (pickle)" file and keeping Regularization Parameter (λ) = 10, the following output is obtained.

```
Time taken to train: 153.29042768478394

 Training set Accuracy:84.8720379147%

 Validation set Accuracy:83.8273921201%

 Test set Accuracy:84.9735049205%
```

<u>TASK III- Performance of Single Hidden Layer vs Deep Neural Networks</u>

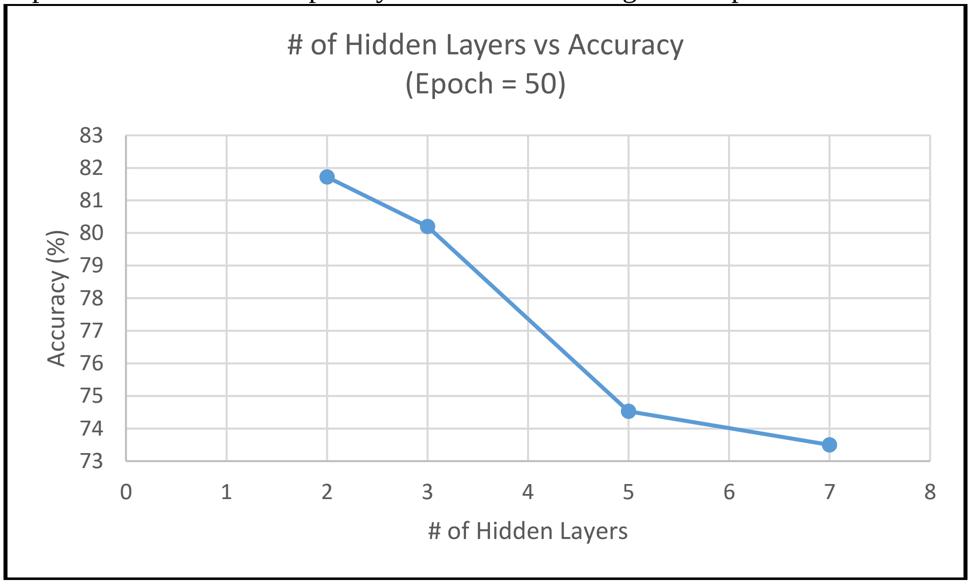
Deep-learning networks are distinguished from the more commonplace single-hidden-layer neural networks by their depth; that is, the number of node layers through which data passes in a multistep process of pattern recognition. Traditional machine learning relies on shallow nets, composed of one input and one output layer, and at most one hidden layer in between. More than three layers (including input and output) qualifies as “deep” learning. So deep is a strictly defined, technical term that means more than one hidden layer. In deep-learning networks, each layer of nodes trains on a distinct set of features based on the previous layer’s output. The further you advance into the neural net, the more complex the features your nodes can recognize, since they aggregate and recombine features from the previous layer.



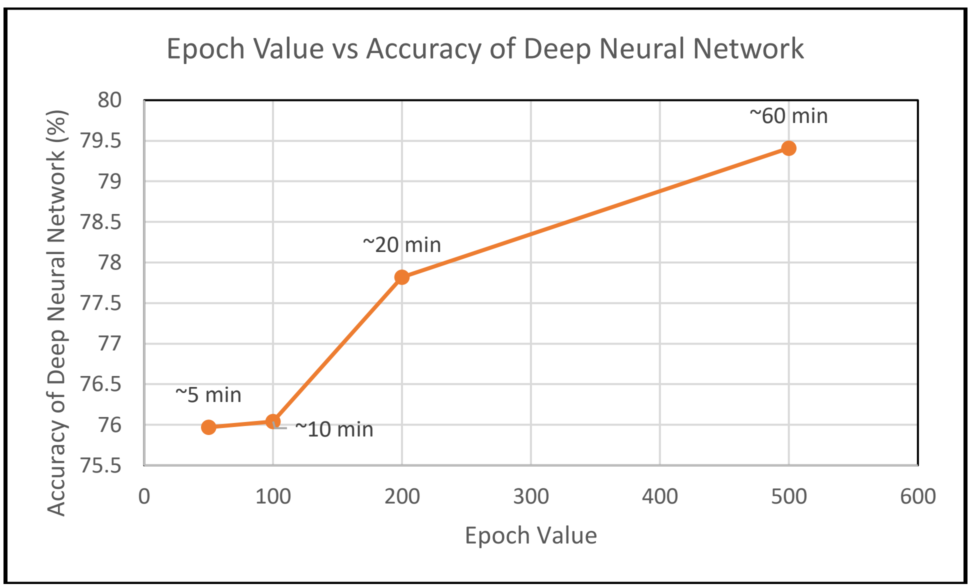
In this assignment, to experiment with Neural Networks with multiple layers, Google’s TensorFlow library has been used.
On evaluating the accuracy of deep neural networks on CelebA Dataset the following results have been obtained. In order to acquire an optimal accuracy, the networks have been evaluated on personal servers and the university server (sprinsteen.cse.buffalo.edu). Both the evaluations have been reported below.

```
Testing CelebA Dataset with deep Neural Network (with multiple hidden layers)

        Accuracies -
        On personal machine -
        2 hidden layers: 81.75%
        3 hidden layers: 78.87%
        5 hidden layers: 74.94%
        7 hidden layers: 73.50%

        On springsteen@cse.buffalo.edu -
        2 hidden layers: 81.72%
        3 hidden layers: 80.20%
        5 hidden layers: 74.53%
        7 hidden layers: 75.97%

        Time Taken-
        For 2,3 & 5 hidden layers ~ 2 minutes.
        For 7 hidden layers ~ 5 minutes
```

From the above results, it can be observed that, on testing the accuracy of the CelebA Dataset on single hidden layer neural networks was ~ 85%. On testing the accuracy of deep neural networks (epoch = 50) on the same dataset, a comparatively lesser accuracy rate was recorded. Also, as the number of hidden layers increased, the accuracy rate decreased and the time taken for evaluation of the network increased. This might be possible due to the complexity in the network design of deep neural networks.



Considering the worst case scenario, on evaluating the performance of deep neural network with seven hidden layers on the CelebA Dataset, an accuracy of 75.97 % has been reported for 50 iterations (epoch = 50). On increasing the epoch value, it is observed that the accuracy rate also increases to a certain value. Also, it can be observed that since the number of iterations increases, the time taken to evaluate the network also increases significantly. Below are the results produced when the deep neural network with 7 hidden layers was evaluated with the epoch values [50, 100, 200, 500].



REFERENCES –

[1] https://deeplearning4j.org/neuralnet-overview

[2] https://www.tensorflow.org/install/install_windows

[3] http://nbviewer.jupyter.org/github/ubdsgroup/ubmlcourse/blob/master/notebooks/ProgrammingAssignment1.ipynb

[4] http://docs.scipy.org/doc/scipy-0. 14.0/reference/generated/scipy.optimize.minimize.html

[5] LeCun, Yann; Corinna Cortes, Christopher J.C. Burges. "MNIST handwritten digit database"