

Learning Activations in Neural Networks

Summary:

The selection of best suitable Activation function is a challenging factor when it comes to the performance of the artificial neural network. This document illustrates end to end ANN development where best parameters of the activation function are learned by the network by exploiting a flexible functional form, $k_0 + k_1x$ with parameters k_0 ; k_1 being learned from multiple runs.

1: Introduction

Developed a feed-forward neural network consisting of an input layer L0, Which consisting of m nodes for m features and two hidden layers L1 and L2 consisting of n and p nodes respectively.

The flow of algorithm Starts by splitting the dataset into two parts, Train & test, proceeded with the initial settings such as sampling the parameters k_0 & k_1 from some distribution, then updating the parameters on epochs and final parameter values reflected at the end of training, Plotting train vs test loss, train and test accuracy, F1-Score and various other metrics for visualization purpose.

2: Methodology

2.1. Network Architecture:

MNIST Dataset, Image size is of $28*28*1$ dimension.

Input layer dim : 784 nodes
Hidden layer 1 dim: 256 nodes
Hidden layer 2 dim: 128 nodes
Output layer dim: 10 nodes

2.2. Weight Initialization:

i) He Weight Initialization:

Weight initialization technique is used as He Weight Initialization. The calculation for the weights is given by the below formula:

$$W^{[l]} = np.random.randn(size_l, size_l-1) * np.sqrt(2 / (size_l-1 + size_l))$$

3. Results:

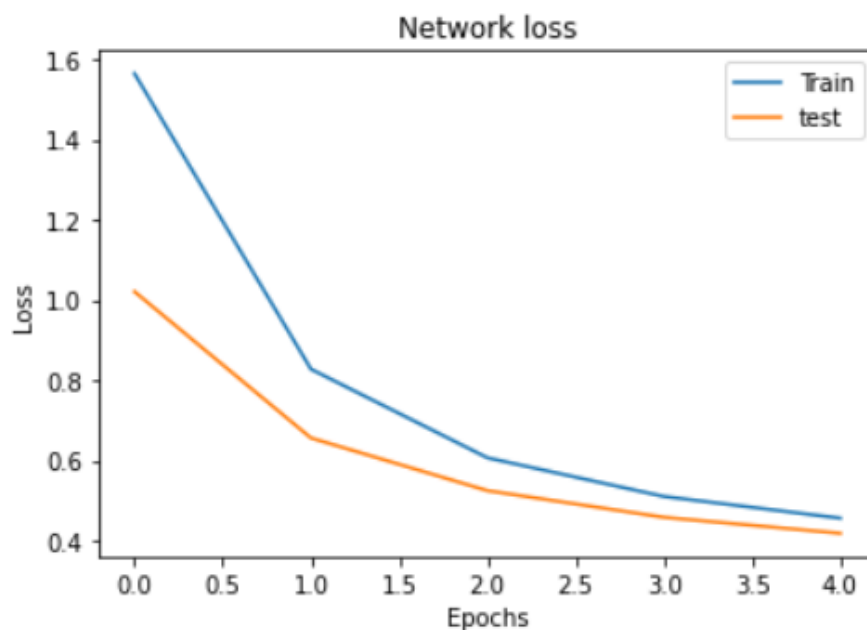
Trained the complete network by 10 epochs. The accuracy and time taken for every epoch is as follows.

```
In [141]: 1 NN = NeuralNetwork(network=[784, 256, 128, 10],epochs=10)
          2 NN.trainNetwork(x_train, y_train, test_x, test_y)
```

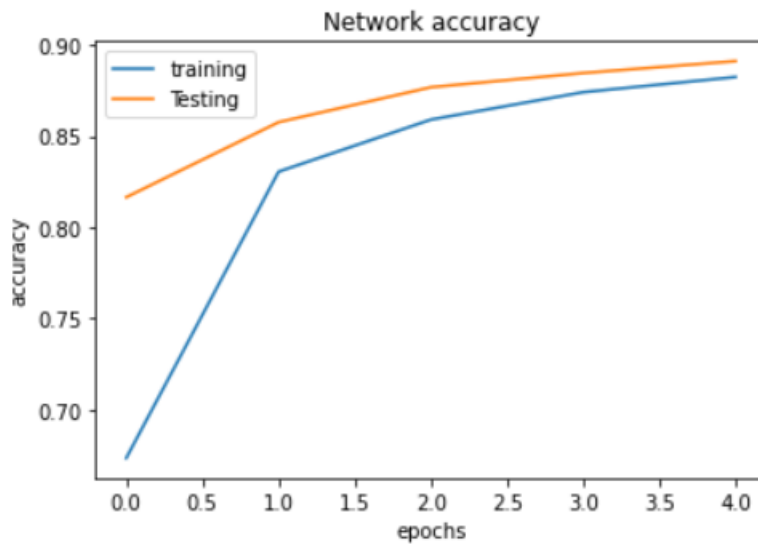
Epoch: 1, Time Taken: 99.46s, Accuracy: 78.238%
 Epoch: 2, Time Taken: 98.52s, Accuracy: 80.467%
 Epoch: 3, Time Taken: 98.57s, Accuracy: 81.686%
 Epoch: 4, Time Taken: 98.84s, Accuracy: 82.295%
 Epoch: 5, Time Taken: 98.84s, Accuracy: 88.133%
 Epoch: 6, Time Taken: 98.53s, Accuracy: 89.143%
 Epoch: 7, Time Taken: 99.15s, Accuracy: 89.695%
 Epoch: 8, Time Taken: 99.56s, Accuracy: 90.133%
 Epoch: 9, Time Taken: 99.16s, Accuracy: 90.486%
 Epoch: 10, Time Taken: 98.41s, Accuracy: 90.733%
 Total Time Taken to Train the Network for 10 Epochs: 989.04s

4. Visualizations:

Train Vs. Test Loss:



Train Vs. Test Accuracy:



F1 Score of the Network:

```
1 prediction = model.predict(train_x)
2
3 final_pred = np.argmax(prediction, axis=1)
4 test_y_final = np.argmax(train_y, axis=1)
5
6 print("F1 Score for the network is :",f1_score(test_y_final, final_pred, average='micro'))
```

F1 Score for the network is : 0.8879428571428571

Confusion Matrix:

