E1-313:TIPR-Assignment1 Report **Kishalay Das**
TA: Anirban Biswas Student Id-15938
Instructor: M NARASIMHA MURTY Submission Date: 04/03/2019

Python version 3 is used to solve all problems.

1 Part-1:MNIST DataSet

Configuration Used:

- Learning Rate: 0.004
- Batch-Size: 500
- Epoc: 50
- Output Layer Node: 10

These are used for all the experiment done as following.

1.1 Task-I/II: Results of Different No of Layers and Different Numbers of Neurons at each layer.

As mentioned in the task,I have built the Multi-Layer Neural Network from scratch and run it on MNIST Data set, varying different hidden layers and different number of neurons in each of it.Plots of all the results are as follows:

Number Of Layer:1

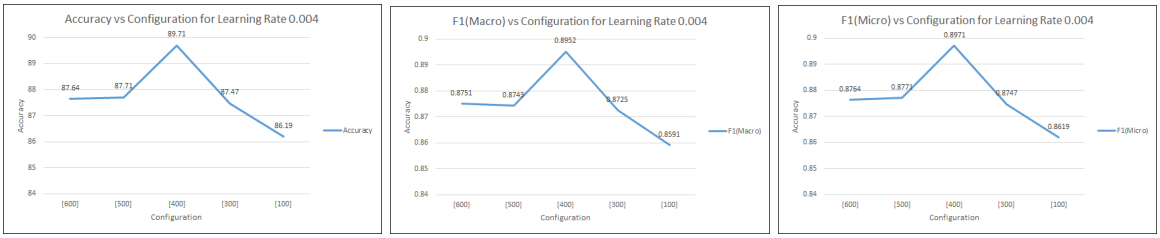

Fig 1:-Accuracy,F1-Score(Micro/Macro)in on mnist Dataset for one hidden layer of different sizes.

Number Of Layer:2

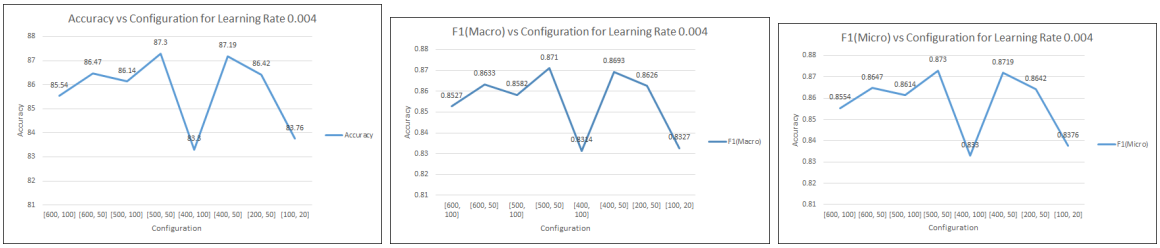

Fig 2:-Accuracy,F1-Score(Micro/Macro)in on mnist Dataset for two hidden layers of different sizes.

Number Of Layer:3

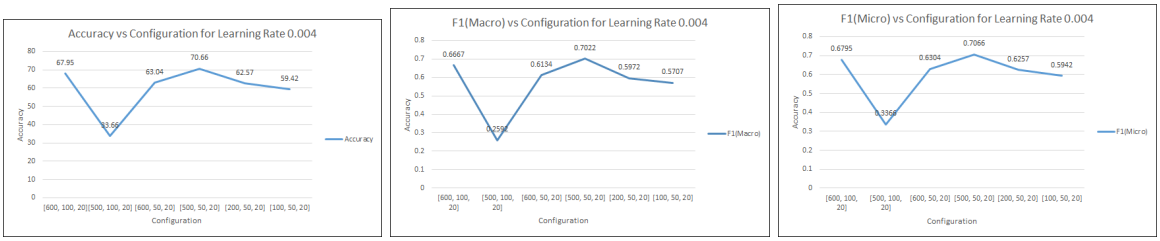

Fig 3:-Accuracy,F1-Score(Micro/Macro)in on mnist Dataset for three hidden layers of different sizes.

So from obtained results,finally I am fixing hidden layer configuration as [400]

1.2 Task-III: Different Activation Functions

I have used tanh and relu activation function and done the experiment.Following is the results compared to Sigmoid:

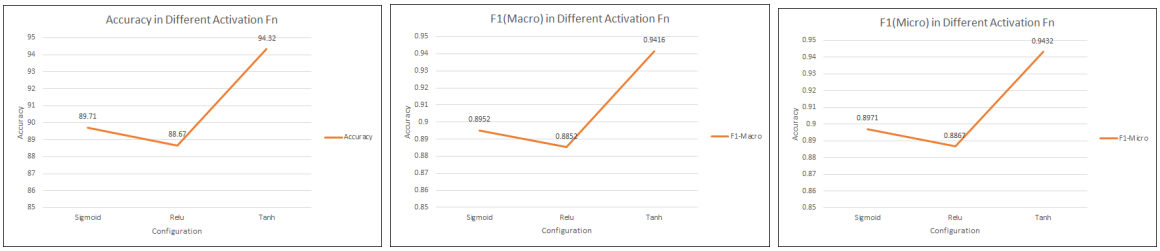

Fig 4:-Accuracy,F1-Score(Micro/Macro)in on mnist Dataset for different activation functions.

1.3 Task-IV:Different Initialization Techniques

To initialize weight vectors,I have used 3 techniques: Random,Normal and Zero functions.Following are the results obtained:

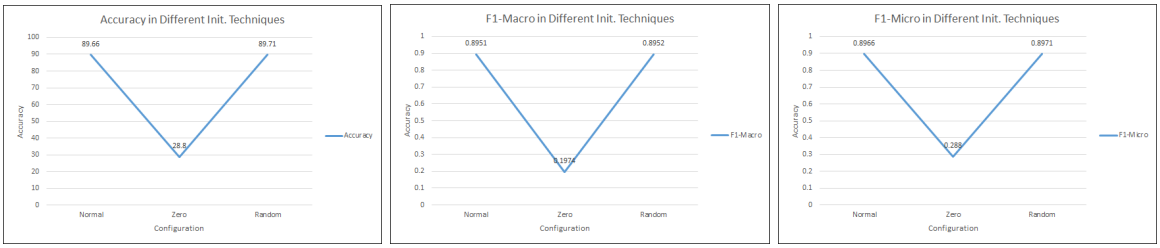

Fig 5:-Accuracy,F1-Score(Micro/Macro)in on mnist Dataset for different Initialization techniques.

1.4 Task-V:Keras Results

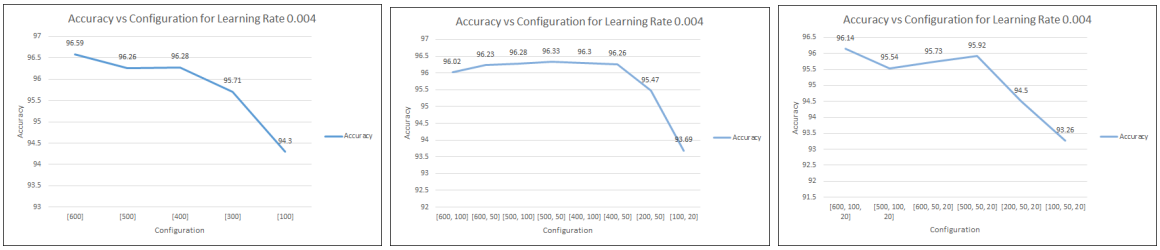

Fig 6:-Accuracy Result By keras on Mnist Dataset for different configuration.

2 Part-2:Cat-Dog Dataset

Configuration Used:

- Learning Rate: 0.003
- Batch-Size: 500
- Epoc: 50
- Output Layer Node: 2
- Input Dimension:500(After Reduction by PCA)

These are used for all the experiment done as following.

2.1 Task-I/II: Results of Different No of Layers and Different Numbers of Neurons at each layer.

In this part i run MLNN on CatDog Dataset, varying different hidden layers and different number of neurons in each of it.Plots of all the results are as follows:

Number Of Layer:1

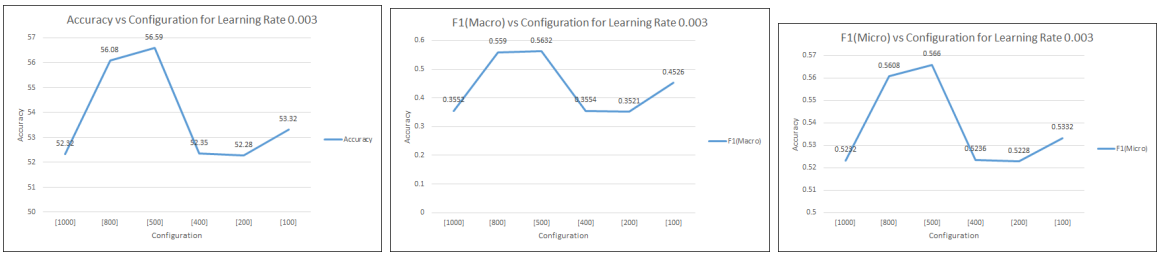

Fig 7:-Accuracy,F1-Score(Micro/Macro)in on CatDog Dataset for one hidden layer of different sizes.

Number Of Layer:2

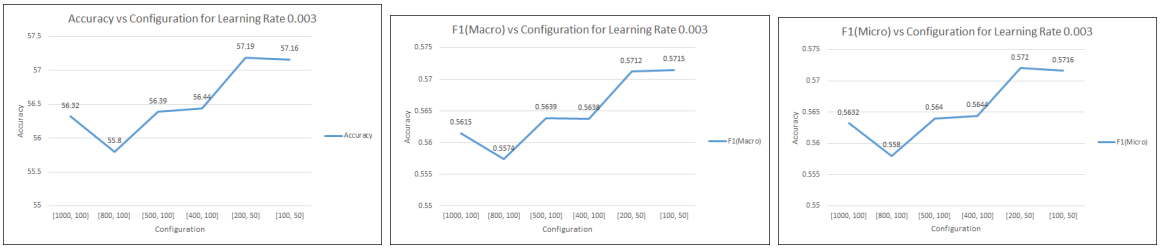

Fig 8:-Accuracy,F1-Score(Micro/Macro)in on CatDog Dataset for two hidden layers of different sizes.

Number Of Layer:3

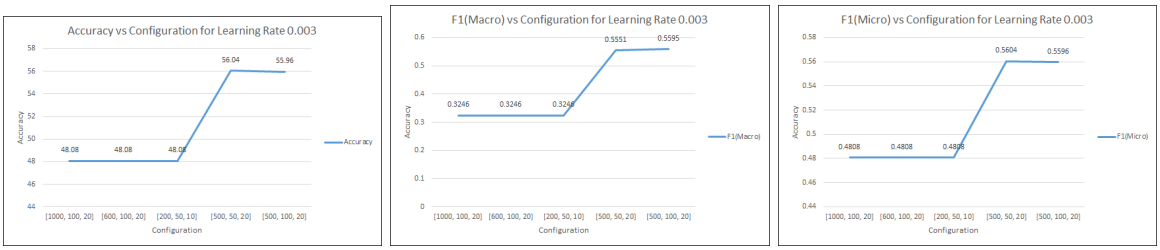

Fig 9:-Accuracy,F1-Score(Micro/Macro) on CatDog Dataset for three hidden layers of different sizes.

So Finally I am fixing hidden layer configuration as [100,50]

2.2 Task-III: Different Activation Functions

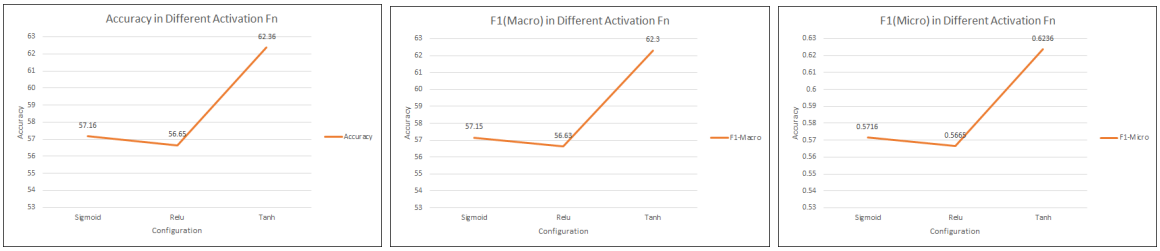

Fig 10:-Accuracy,F1-Score(Micro/Macro) on CatDog Dataset for different activation functions.

2.3 Task-IV:Different Initialization Techniques

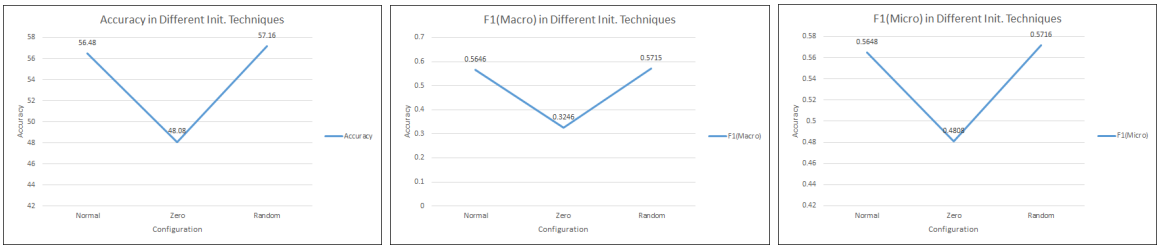

Fig 11:-Accuracy,F1-Score(Micro/Macro) on CatDog Dataset for different Initialization techniques.

2.4 Task-V:Keras Results

I have run CatDog DataSet with same configuration mentioned above with Keras Library.The Results as follows:

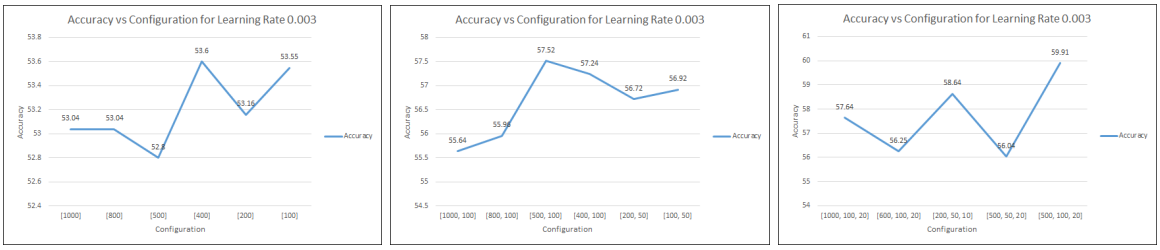

Fig 12:-Accuracy Result By keras on CatDog Dataset for different configurations.

3 Part-3:Dolphin and Pubmed Dataset

3.1 Dolphin Dataset

I have run my designed MLNN on Dolphin Dataset with following configurations.

Learning Rate: 0.003

Batch-Size: 2

Epoc: 50

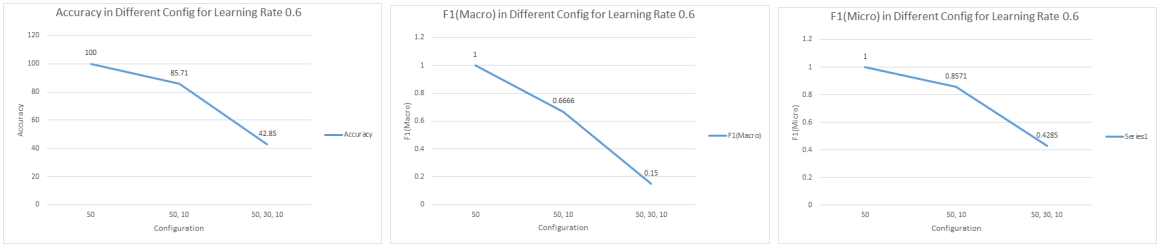

Fig 13:-Accuracy,F1-Score(Micro/Macro)in on Dolphin Dataset for hidden layers of different sizes.

For single and two layers it is giving better results that Bayes and almost similar like KNN.But with 3 layers it drops.My remarks on this is due to low epoc size it is happening such.It needs more number of epoc iterartions to learns 3-Layer Network(As number of data sets in 62 only).

3.2 Pubmed Dataset

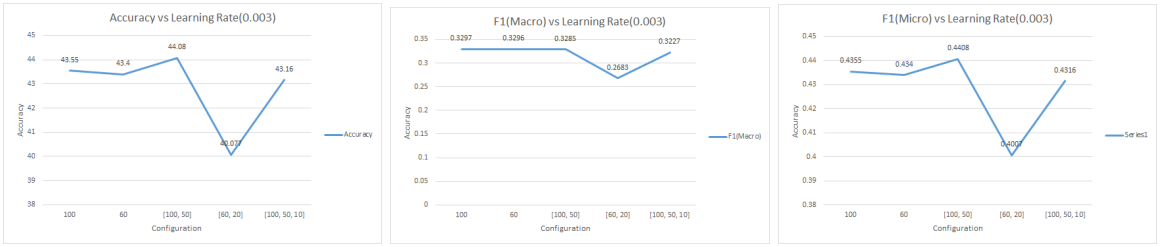

Fig 14:-Accuracy,F1-Score(Micro/Macro)in on Pubmed Dataset for hidden layers of different sizes.

Similarly,for pubmed also,it is giving slighly better results that Bayes and KNN.Here 3 layer NN also works fine as we have sufficient data to train the model.