

**CZ4042 – Neural Networks Project 1 Report**

**Prepared by:**

Jordan Lam Yu Cheng [JLAM014@e.ntu.edu.sg](mailto:JLAM014@e.ntu.edu.sg)

Nathanael Raj nathanae001@e.ntu.edu.sg

*Note: Reports for Part A and Part B have been organised separately.*

Neural Networks Project 1 Part A

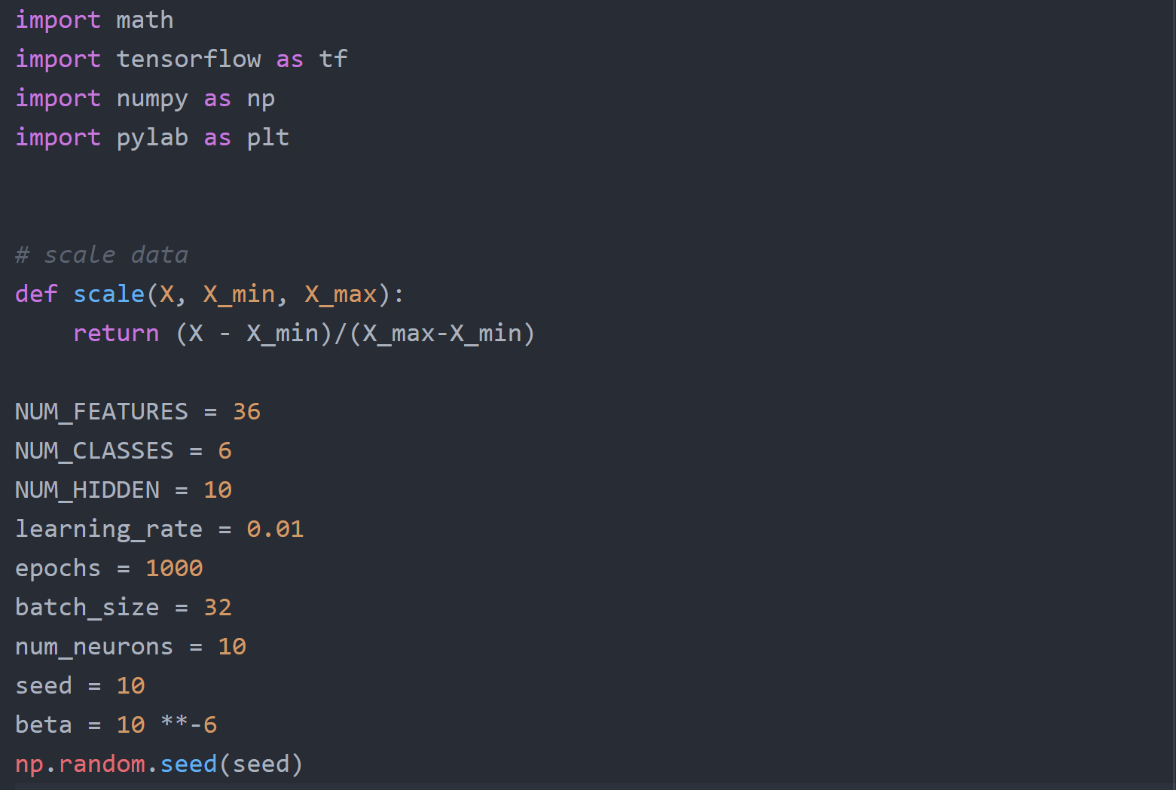
# Introduction

In this project, we aim to classify the Landsat satellite dataset. The dataset consists of multi-spectral values of pixels in 3x3 neighbourhoods in a satellite image and the classification associated with the central pixel in each neighbourhood. Each data point has 36 features, and a corresponding true class. We performed Machine Learning using three and four layer neural networks and achieved an overall test accuracy of 87% after optimising several parameters.

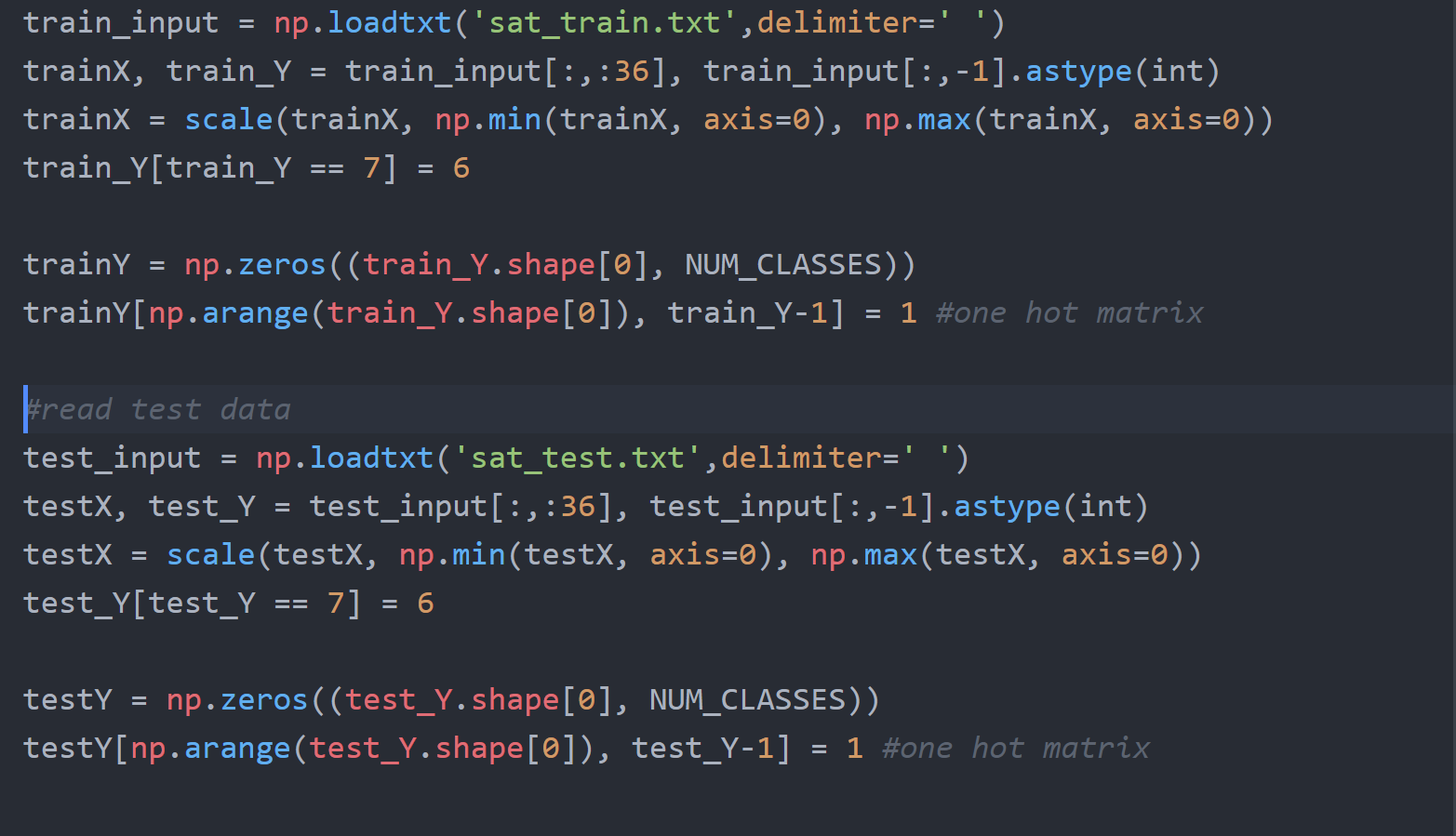
Methods

We implemented the neural network in tensorflow with one hidden layer initially and tuned several parameters using a test set to improve accuracy. Subsequently, we added an additional hidden layer to compare the performance between a one and two hidden layer neural network. Details on how the neural network was implemented are contained in the next section.

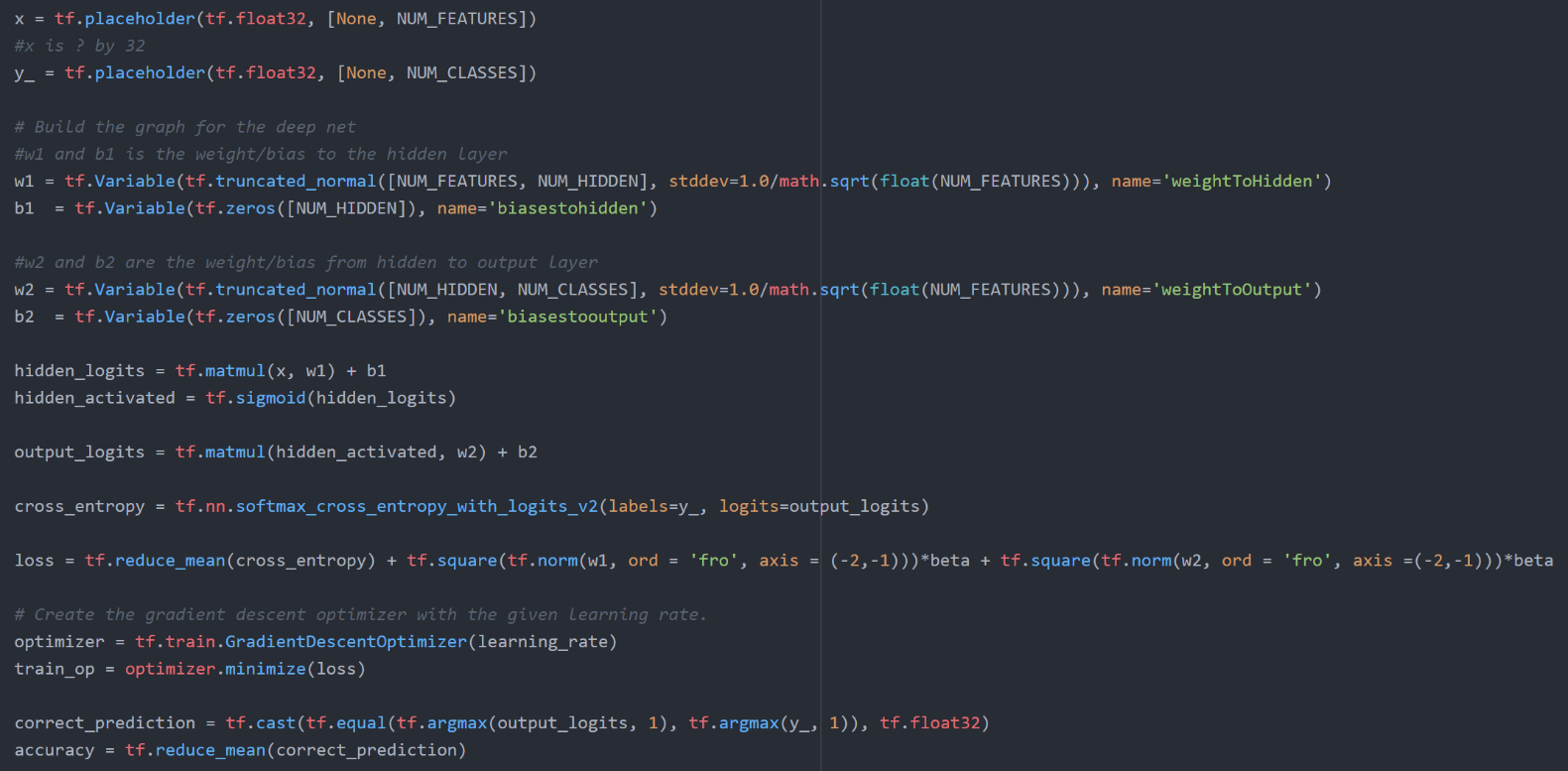
Experiment

****

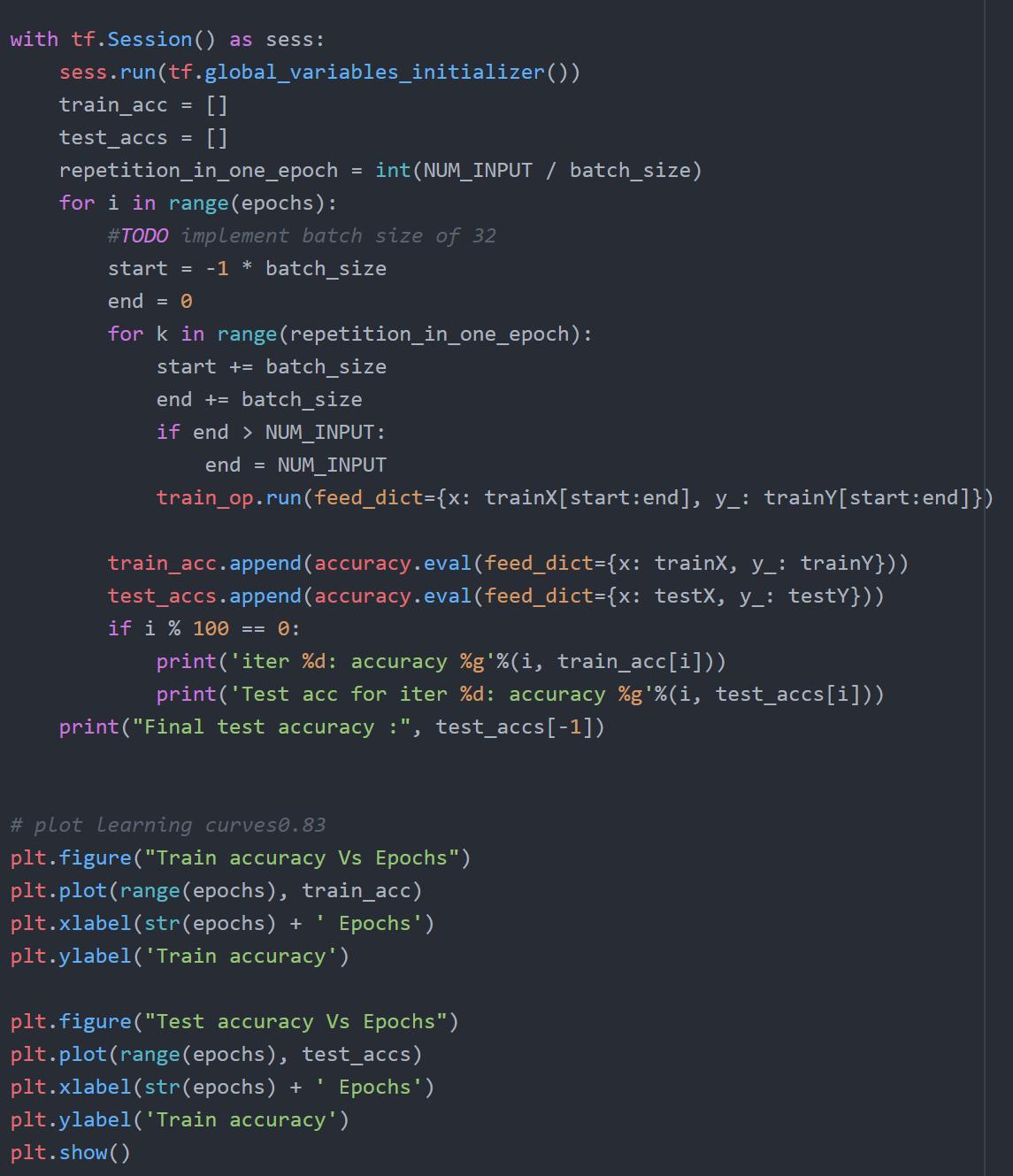
The parameters for the 3 layer neural network is first initialised, as well as a scale function used to scale the input according to the maximum and minimum values



Train and test data is read from the data. The labels of the class are converted into a one hot matrix so that it can be fed into the cross entropy loss function



The neural network consist of 3 layers. The input goes into a 10 neuron hidden layer, where a sigmoid activation function is applied at each neuron. The output of this hidden layer is then sent to the final output layer, which consist of 6 neurons. A softmax function is applied at this layer and the output of each neuron will correspond to the probability that the data point belongs to the class its neuron is representing. The neuron network is trained using a cross entropy loss function applied on the output. L2 regularisation penalty term is added to the loss function with beta parameter of 10-6.



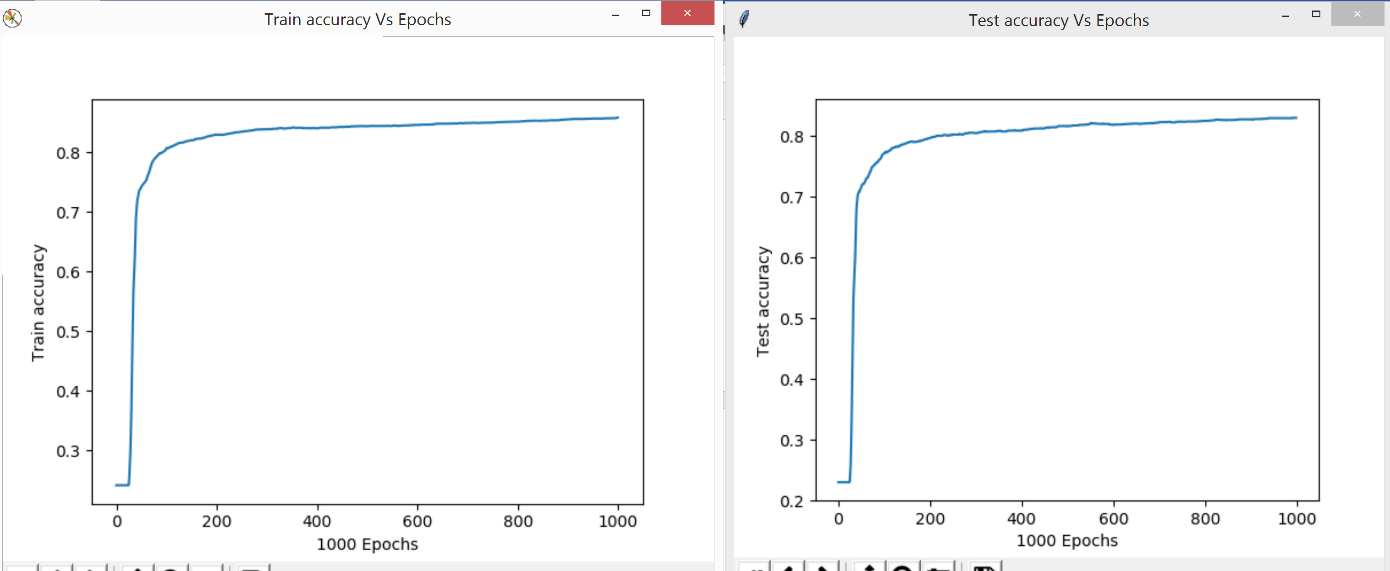
Finally the model is trained using the training data. One epoch consist of all the iterations required for all data points in the training set to be visited. This three layer network was trained for 1000 epochs. The four layer network had the same setup as this three layer one.

# Conclusion and Results

In conclusion, we found that the best performance for this problem was achieved with the following parameters: one hidden layer, batch size of 4, 15 neurons in the hidden layer and decay parameter of 10-6. This yielded a test accuracy of 87%. The results of the questions are contained below

**Question 1**

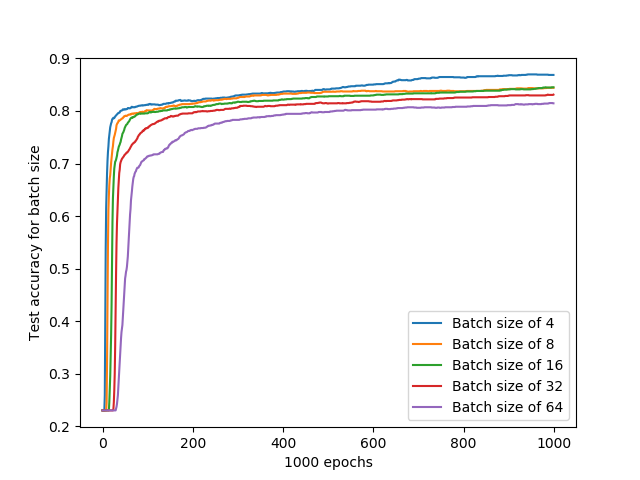
The plots of the training and test accuracy are as below:



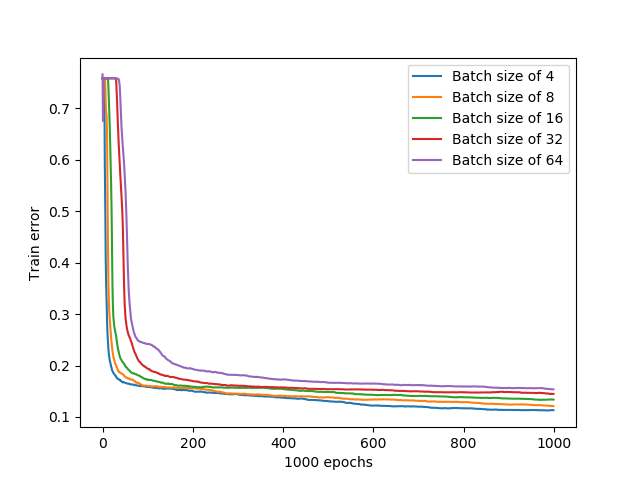
**Question 2**

2a

Test Accuracy Against Epochs for each Batch Size

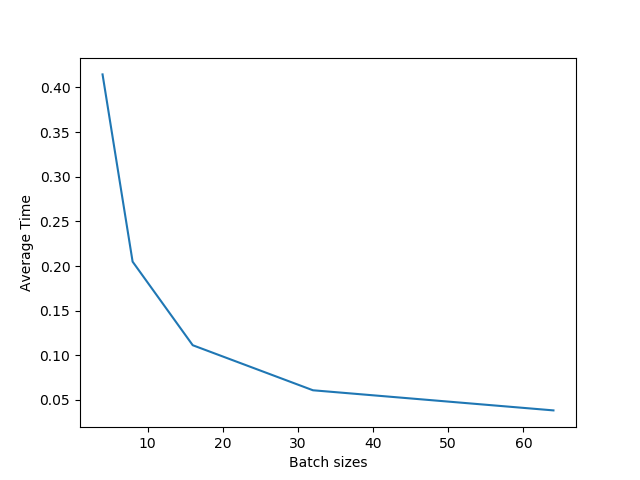


Train Error Against Epochs for each Batch Size



2b

Average Time Taken (seconds) for one Epoch Against Batch Size



Average time for one epoch was derived by taking the time taken to train the network for 1000 epochs and dividing it by 1000 epochs. Values are in seconds

2c

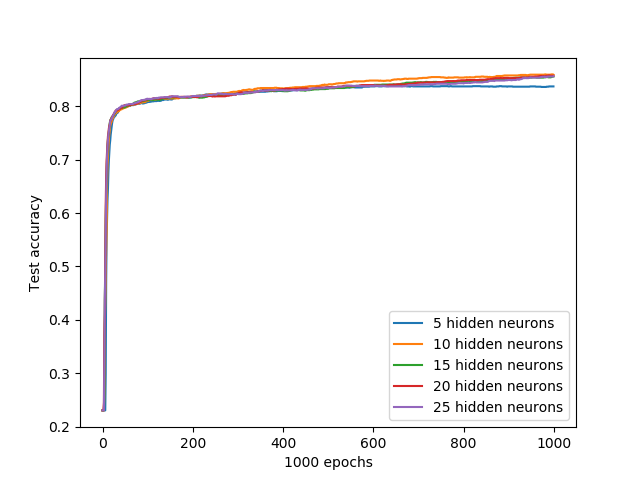
Chosen batch size: 4

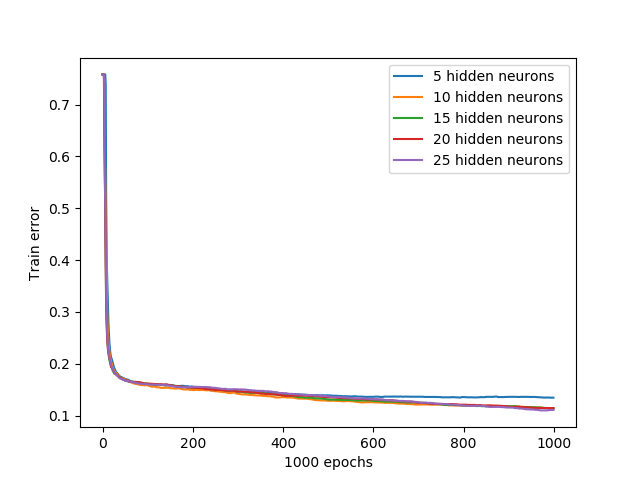
While there is a trade off between accuracy and time as batch size increases, the time taken to train for an epoch with batch size 4 is still tolerable at 0.43 seconds. At 1000 epochs, training with batch size 4 takes around 7 minutes to train which is a relatively small price to pay for the improved accuracy in the model.

**Question 3**

3a

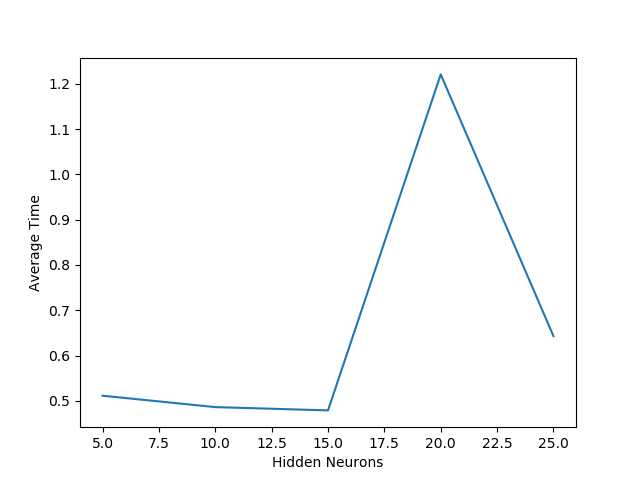
Test Accuracy against Epochs plotted for each number of Hidden Layer Neuron



Training error against Epochs plotted for each number of Hidden Layer Neuron

3b

Average Time taken for one Epoch against Number of Hidden Layer Neurons



Average time here is computed the same way as in question 2.

3c

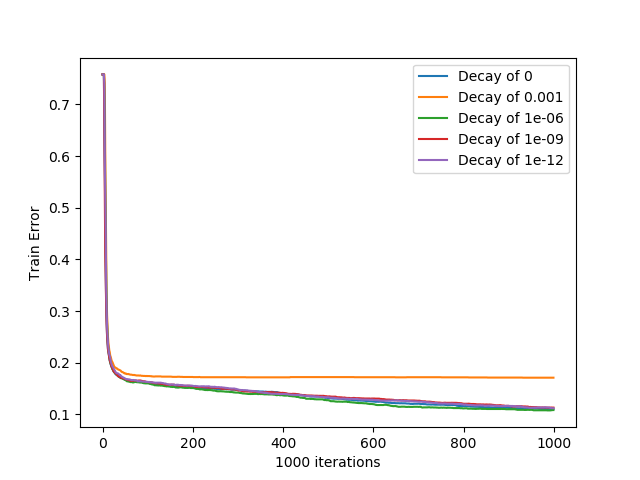
Chosen number of hidden neurons: 15

The test accuracy of the network remained relatively constant from 10 to 25 hidden neurons at 86%. Thus, 15 hidden neurons were chosen since it took the least average time. Nonetheless, this difference was rather insignificant.

**Question 4**

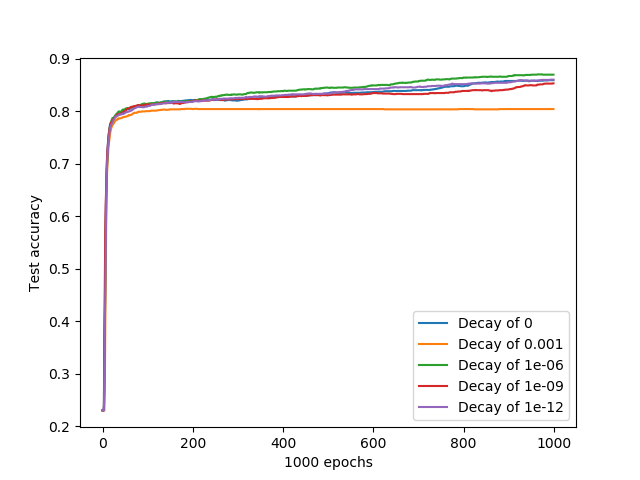
4a

Training error against Epochs plotted for each Beta Value



4b

Test Accuracy against Epochs plotted for each Beta Value

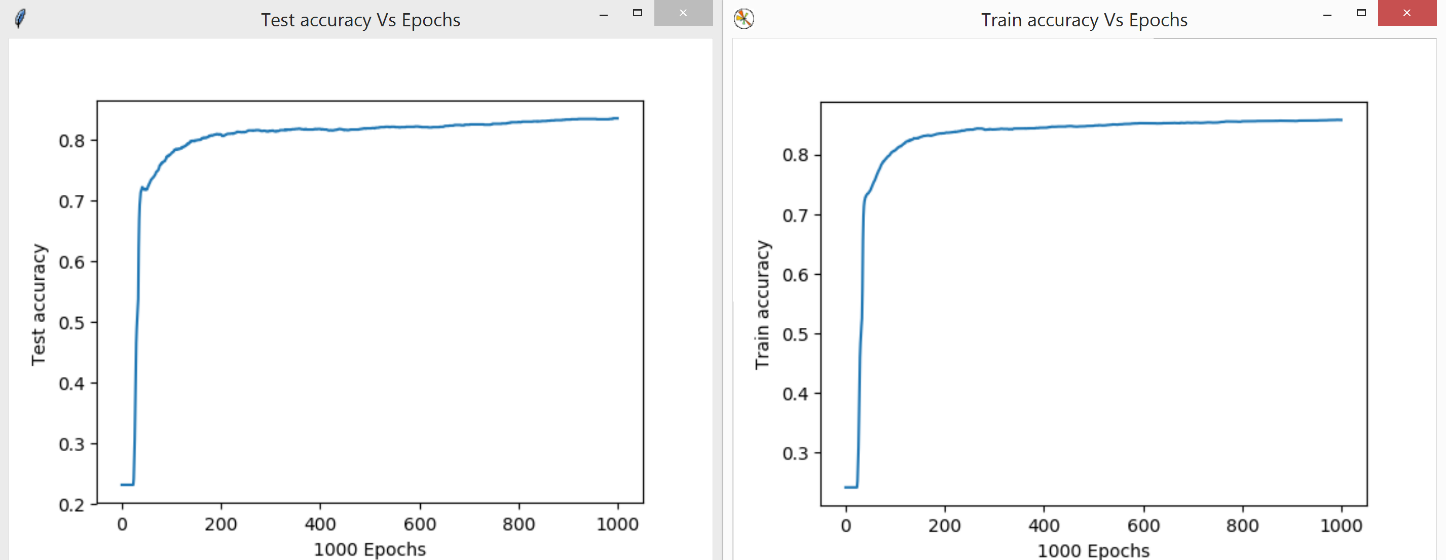


4c

Similar trends were observed for both the training error plot and test accuracy plot for varying decay parameters. As such, the only factor which differentiated each decay parameter was accuracy. Thus, decay parameter of beta = 10-6 was chosen since it yielded the highest test accuracy at 87%.

**Question 5**

5a



5b

The 4 layer neural network gave a slightly higher final test accuracy of 83.5% compared to the 82.7% test accuracy of the 3 layer neural network.

Both 3 and 4 layer neural networks also had similar shaped plots, training and testing plateau after around 300 epochs. In both networks the shapes of the train and test accuracy plots were also similarly shaped.

The similar performance of both networks indicate that one hidden layer was enough to introduced non-linearity into the model for it to learn the parameters of this problem. Hence, the additional non-linearity introduced by the second hidden layer could not be utilised and did not provide much better results.