Report for Machine Learning Assignment-2

Made By:Atharva Vyas
15CS30005

The code for part A corresponds to part1.py and code for part B corresponds to part2.py.Running the code may take half an hour for both of the parts.

Preprocessing

1.The messages were split into tokens using python's split function

2.English stopwords were removed using nltk's stop words library.These stop words were removed from tokens

3.Porter Stemming was applied using nltk's porter stemmer

4.Each message was broken down into vector as given in create_vector function of part1.py and part2.py

5.Training and Test data was split by using random message from the whole set and inserting into test set.

Results of Part A.

1.Two Hidden Layers of 100 and 50 neurons were taken and the output layer had one neuron

2.For part A1(sigmoid) **threshold** of 0.5 was used i.e if output from the neuron of last layer >=0.5 we classify as spam otherwise classified as ham. For part A2(tanh) **threshold** of 0 was used i.e if output from the neuron of last layer >=0 we classify as spam otherwise classified as ham.

3.Stochastic Gradient Descent(SGD) was used as optimisation algorithm and squared error function as optimisation function.learning rate=0.1

Part A1

1.The neural network was ran for 60,000 iterations.The in-sample and out-sample error was plotted after every 2000 iterations The code also plots in-sample and out-sample accuracy.

2.Weights were randomly initialized between -1 to +1

3.The graph of **out-sample error** (squared mean error) with number of iterations is given below saved as figure_1_1.png by code:

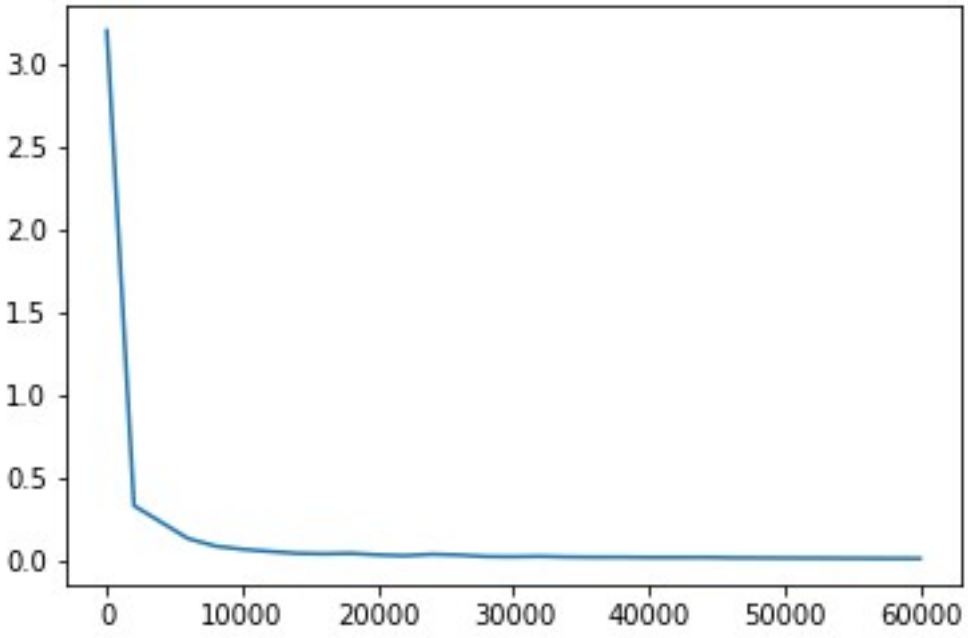


Fig.The squared mean error on y-axis and number of iterations on x-axis

4.The graph of **in-sample error** (squared mean error) with number of iterations is given below saved as figure_1_3.png by code:

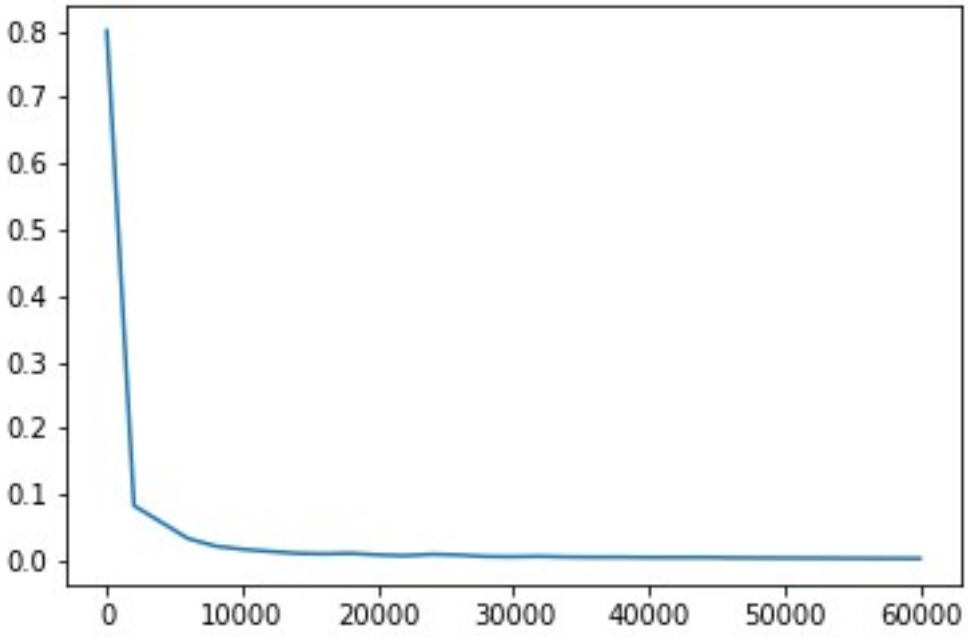


Fig.The squared mean error on y-axis and number of iterations on x-axis

5.The **optimal number of iterations** comes out to be around 20000 as after which very less change in error is observed.

6.The **Accuracy** of classification for out-sample comes out to be 96.4% and Accuracy for in-sample comes out 99.6% after 60000 iterations.

Part A2

1.The neural network was ran for 60,000 iterations.The in-sample and out-sample error was plotted after every 2000 iterations The code also plots in-sample and out-sample accuracy.

2.Weights were randomly initialized between 0 to 1/1000

3.The graph of **out-sample error** (squared mean error) with number of iterations is given below saved as figure_1_5.png by code:

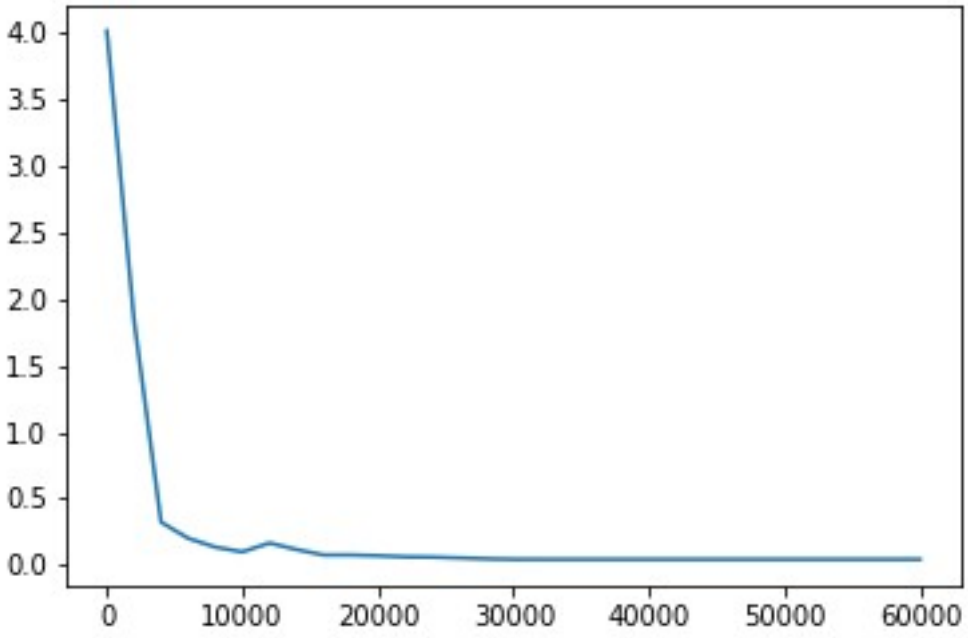


Fig.The squared mean error on y-axis and number of iterations on x-axis

4.The graph of **in-sample error** (squared mean error) with number of iterations is given below saved as figure_1_7.png by code:

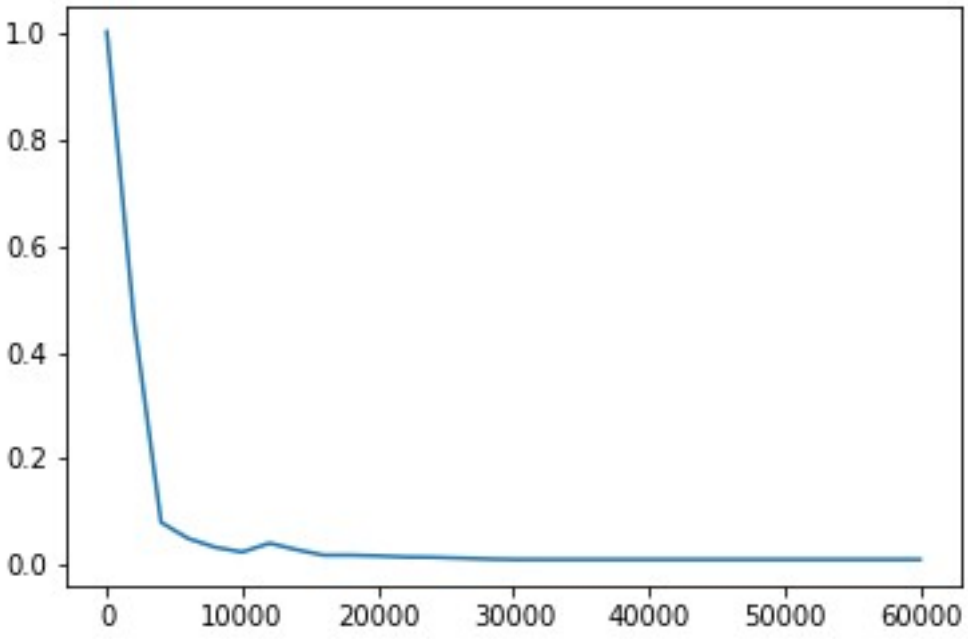


Fig.The squared mean error on y-axis and number of iterations on x-axis

5.The **optimal number of iterations** comes out to be around 20000 as after which very less change in error is observed.

6.The **Accuracy** of classification for out-sample comes out to be 97.4% and Accuracy for in-sample comes out 99.7% after 60000 iterations.

Results of Part B.

1.Two Hidden Layers of 100 and 50 neurons were taken and the output layer had 2 neurons.Softmax was applied on the the two output neurons and not activations functions like sigmoid and tanh.For the inner layers both sigmoid and tanh was applied seperately.

2.Spam is labeled by [0,1] and ham is labeled by [1,0] hence if after applying softmax if output of neuron1 > output of neuron2,neural net classifies the message as ham otherwise spam.

3.Stochastic Gradient Descent(SGD) was used as optimisation algorithm and **cross entropy error** function as optimisation function.learning rate=0.1

4.The neural network was ran for 60,000 iterations.The in-sample and out-sample error was plotted after every 2000 iterations for sigmoid as an activation function for inner hidden layer and 4000 iterations for tanh. The code also plots in-sample and out-sample accuracy.

5.Weights were initialized randomly between -1 and +1

Case1.Sigmoid activation function for hidden layers

i.The graph of **out-sample error** (squared mean error) with number of iterations is given below saved as figure_2_1.png by code:

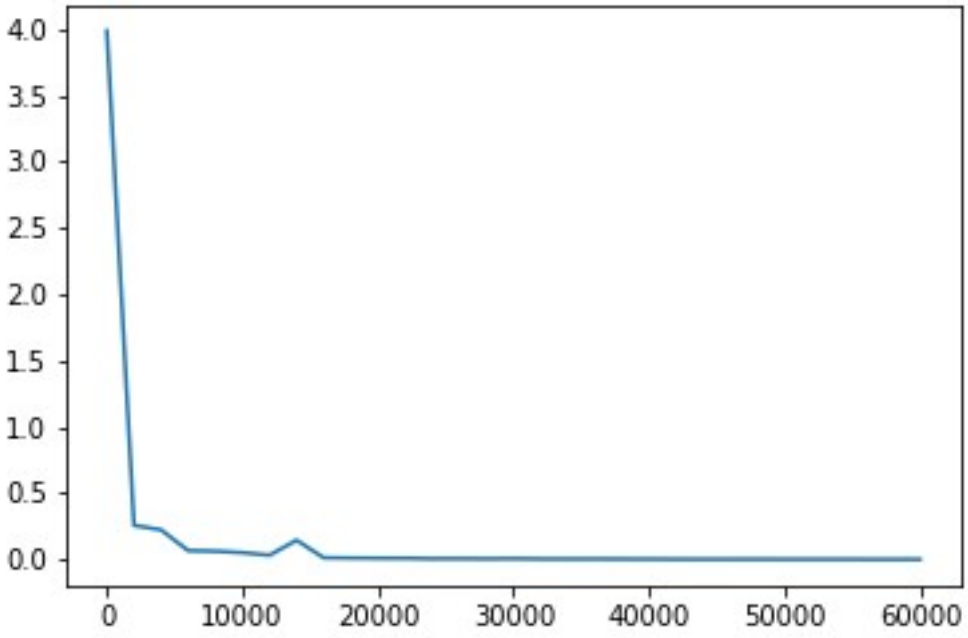


Fig.The squared mean error on y-axis and number of iterations on x-axis

ii.The graph of **in-sample error** (squared mean error) with number of iterations is given below saved as figure_2_3.png by code:

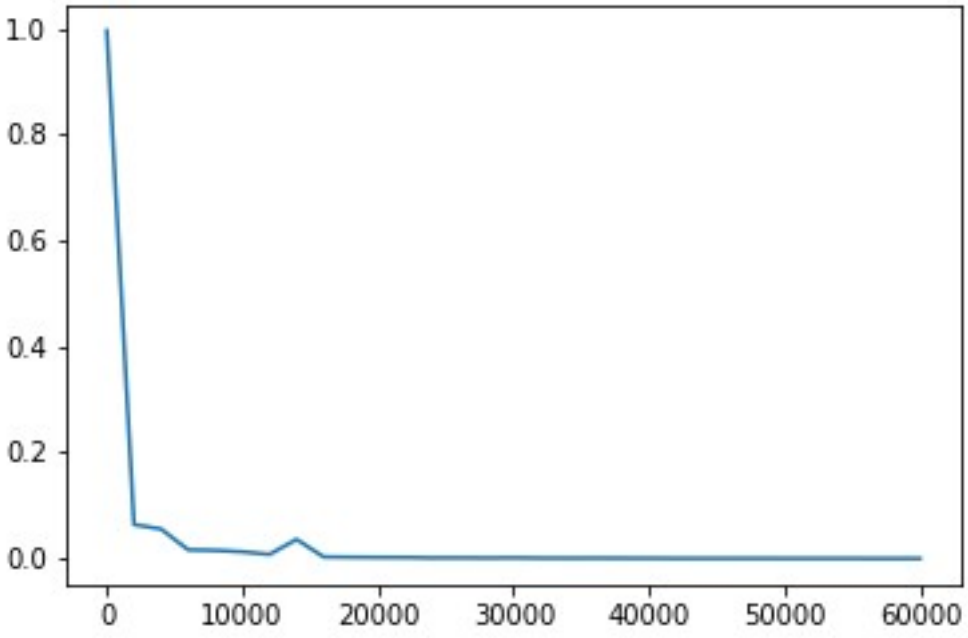


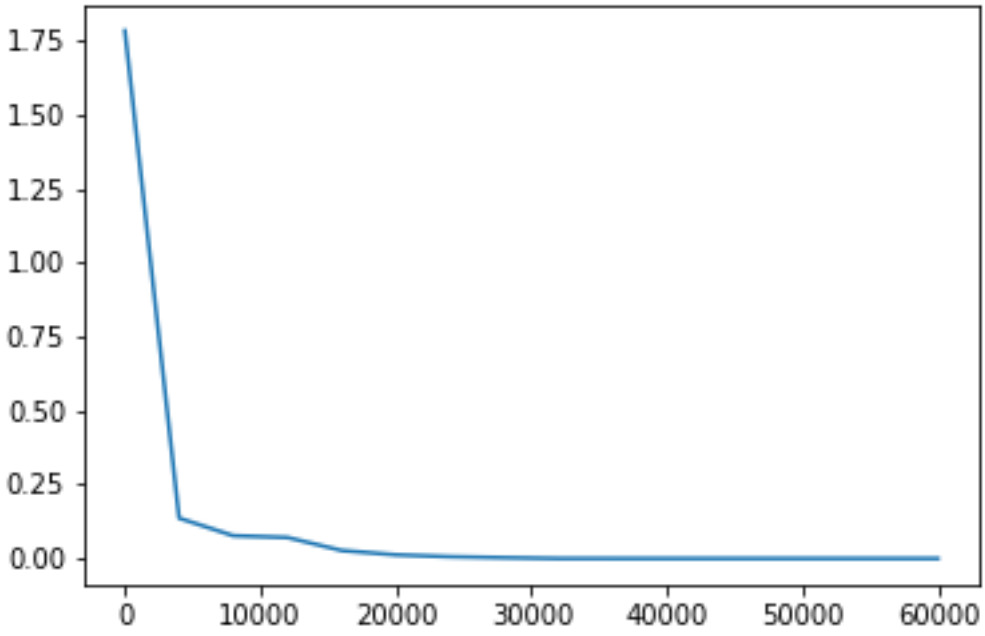
Fig.The squared mean error on y-axis and number of iterations on x-axis

iii.The **optimal number of iterations** comes out to be around 20000 as after which very less change in error is observed.

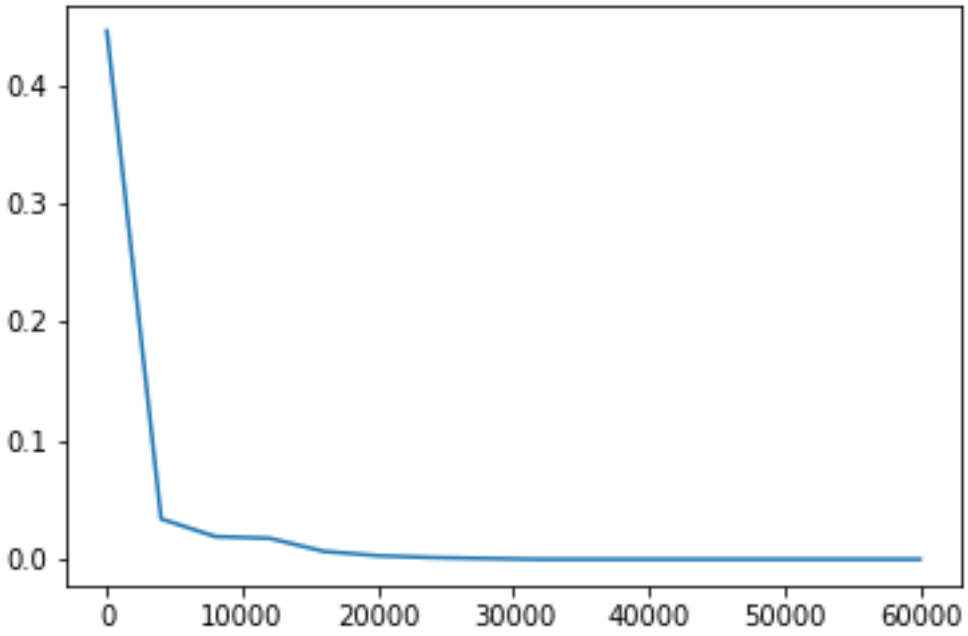
iv.The **Accuracy** of classification for out-sample comes out to be 97.76% and Accuracy for in-sample comes out 100% after 60000 iterations.

Case2.Tanh activation function for hidden layers

i.The graph of **out-sample error** (squared mean error) with number of iterations is given below saved as figure_2_5.png by code:



ii.The graph of **in-sample error** (squared mean error) with number of iterations is given below saved as figure_2_7.png by code:



iii.The **optimal number of iterations** comes out to be around 30000 as after which very less change in error is observed.

iv.The **Accuracy** of classification for out-sample comes out to be 97.2% and Accuracy for in-sample comes out 100% after 60000 iterations.

The Best performer:Comparing the models on the basis of accuracy softmax with cross entropy error and sigmoid as activation performs best with 97.8% out-sample accuracy. Using squared mean error and tanh as activation function also performs good with 97.4% accuracy