MACHINE LEARNING
ASSIGNMENT 2 REPORT
CS60050

Rahul Das
15MA20055

Problem Statement:

Given an email, classify it as a ham or a spam email.

Data:

A labelled Dataset of emails with labels as ham or spam (5574 in total). I randomly sampled 80% of the data (4460 in number) for training purpose and the remaining 20% of the data (1114 in number) for testing purposes.

Data Preprocessing:

All the words in the union of all the emails were brought down in tokens. I removed the standard set of english stopwords. After these preprocessing, I applied Porter Stemming on the tokens. Now these tokens are going to serve as the input to the model by using one hot encoding.

Part 1A:

Neural network Architecture:
Input Layer: 9485 tokens one hot encoding and one bias term
Hidden Layer 1: 100 neurons and one bias term
Hidden Layer 2: 50 neurons and one bias term
Output Layer: One neuron showing chances of being an email being ham or spam
Activation Function: **Sigmoid**
Error Function: Squared Error
Initial Weights: Random Assignment of values
Iterations needed to converge: **10**

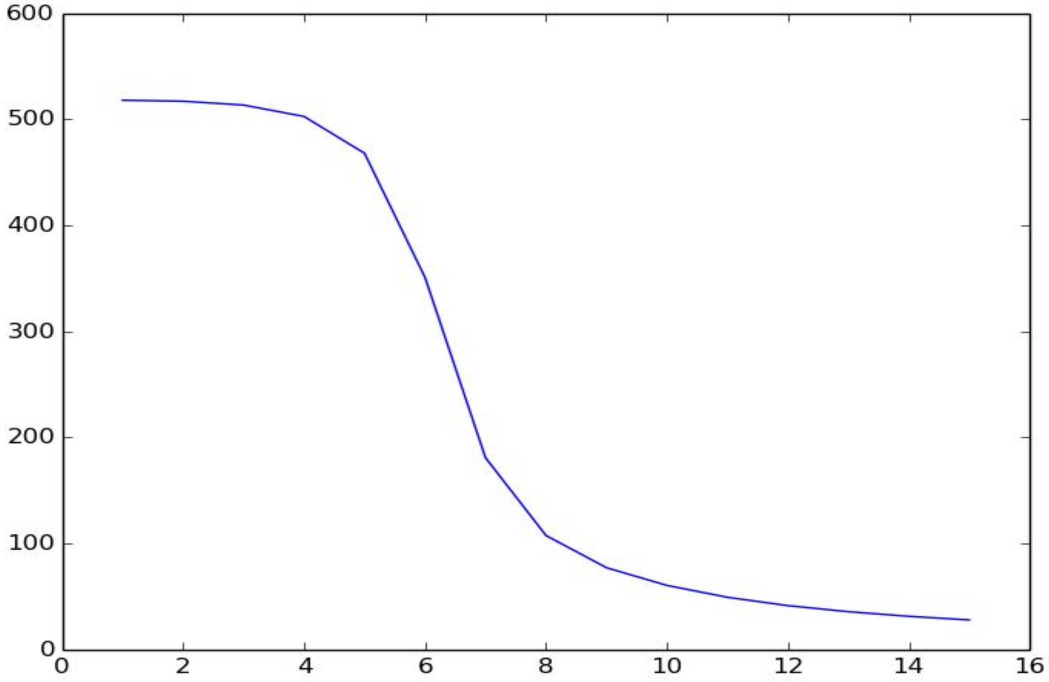

Fig 1. Iterations vs In sample error for sigmoid activation function

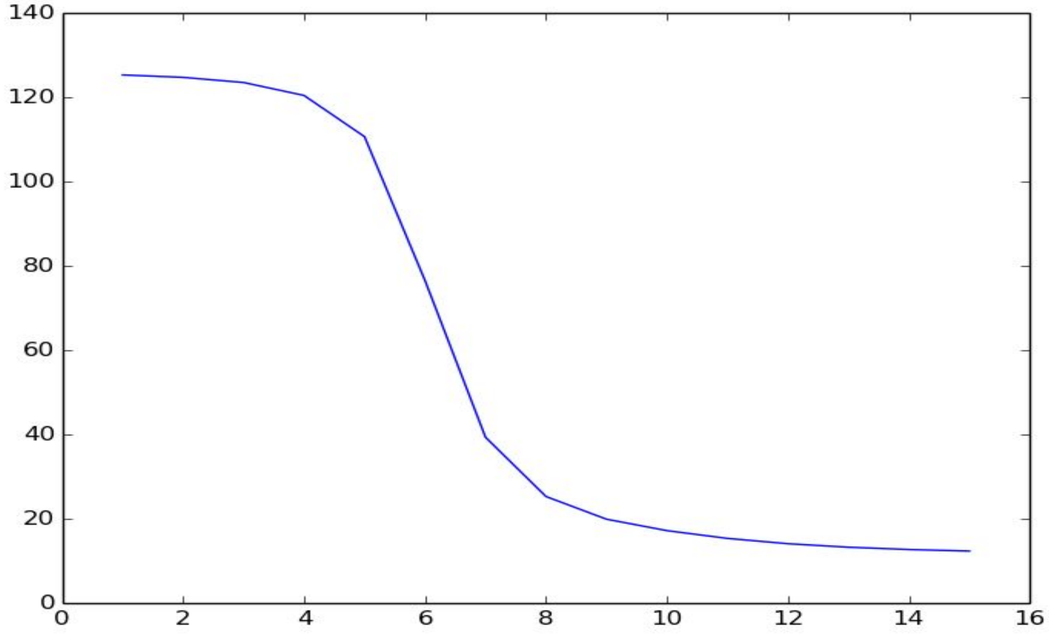

Fig 2. Iterations vs Out Sample error for sigmoid Activation Function

Part 1B:

Neural network Architecture:
Input Layer: 9485 tokens one hot encoding and one bias term
Hidden Layer 1: 100 neurons and one bias term
Hidden Layer 2: 50 neurons and one bias term
Output Layer: One neuron showing chances of being an email being ham or spam
Activation Function: **Tanh**
Error Function: Squared Error
Initial Weights: Random Assignment of values
Iterations needed to converge: **10**

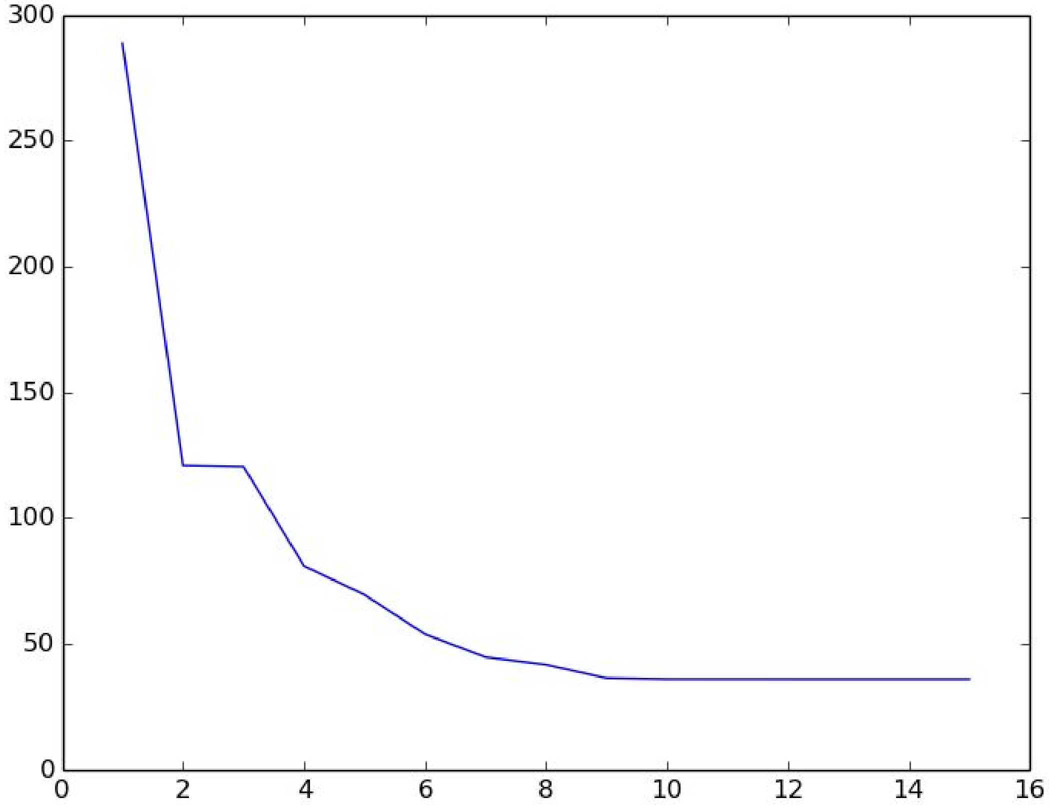

Fig 3. Iteratons vs In Sample error for tanh activation function

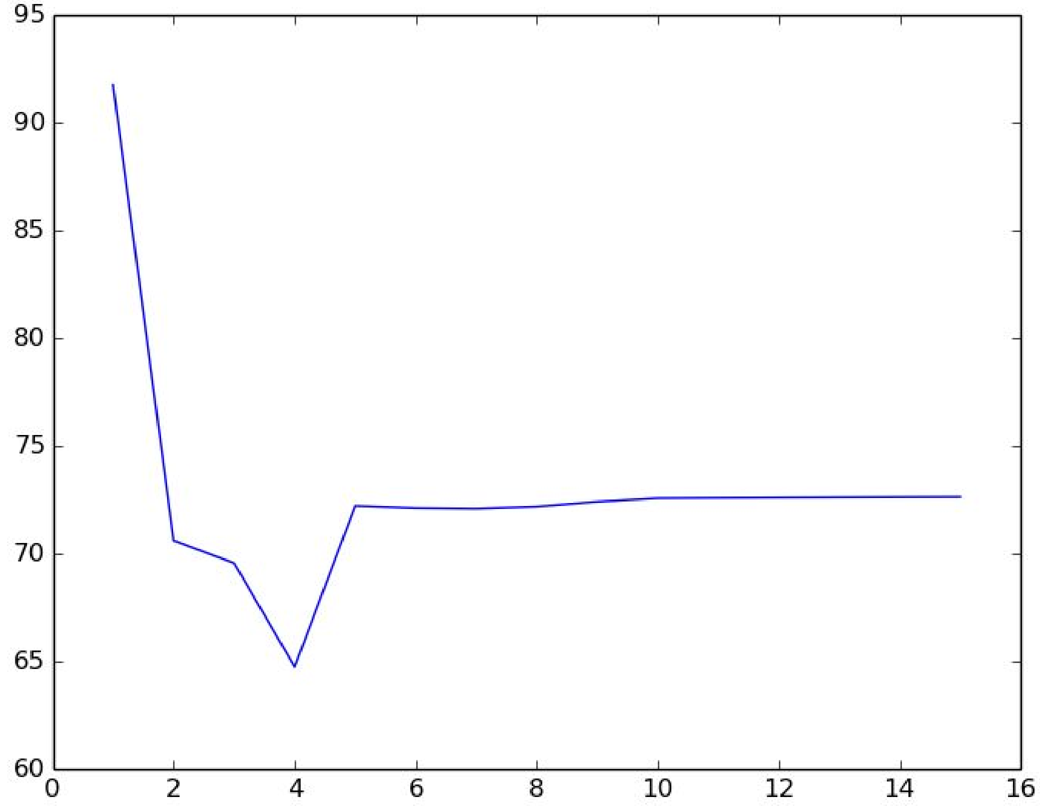

Fig 4. Iterations vs Out Sample Error for tanh activation Function

Part 2:

Neural network Architecture:
Input Layer: 9485 tokens one hot encoding and one bias term
Hidden Layer 1: 100 neurons and one bias term
Hidden Layer 2: 50 neurons and one bias term
Output Layer: 2 neurons each computed taking using **softmax function** for each case spam and ham respectively (1st neuron for spam, 2nd for spam) depicts probabilities
Activation Function: **Sigmoid**
Error Function: Squared Error
Initial Weights: Random Assignment of values
Iterations needed to converge: **10**

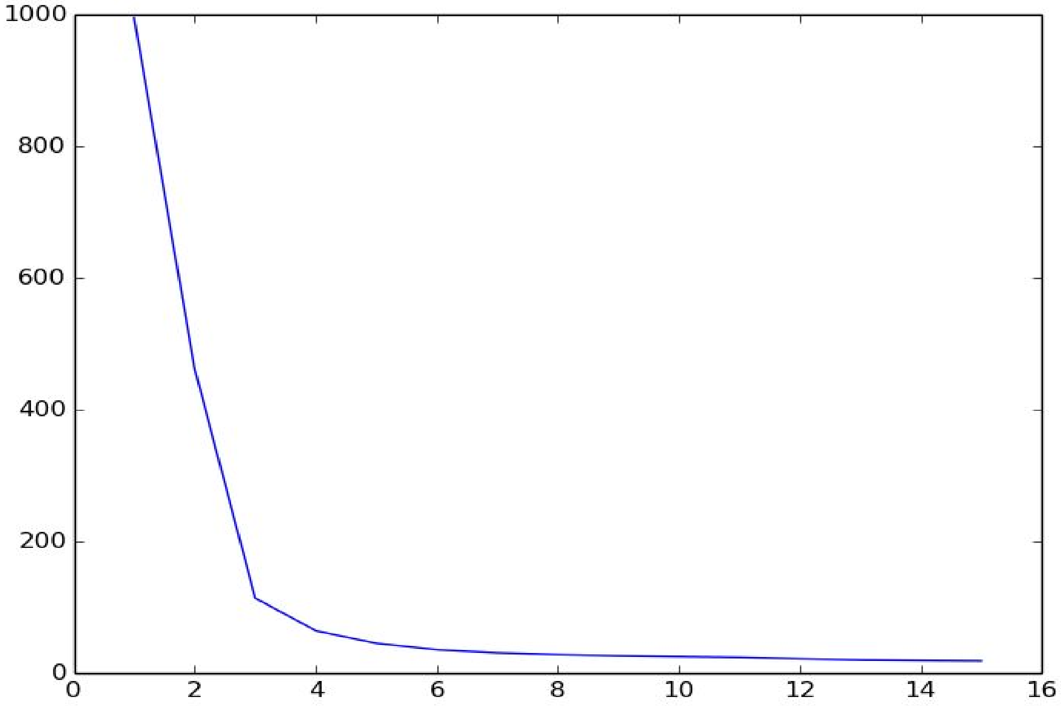

Fig 5. In Sample Error vs number of Iterations for Part 2

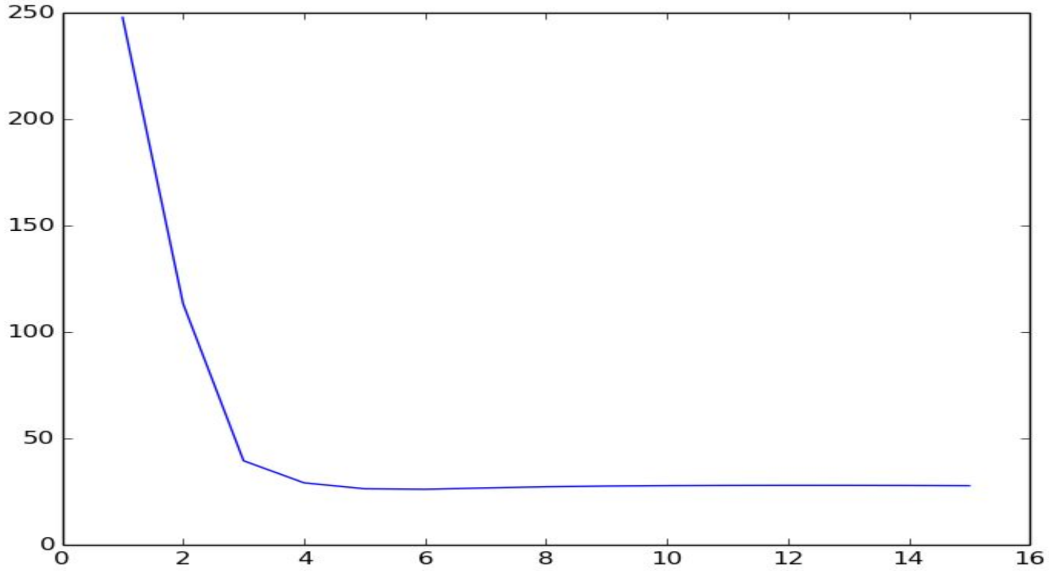

Fig 6. Out Sample Error vs number of iterations for Part 2