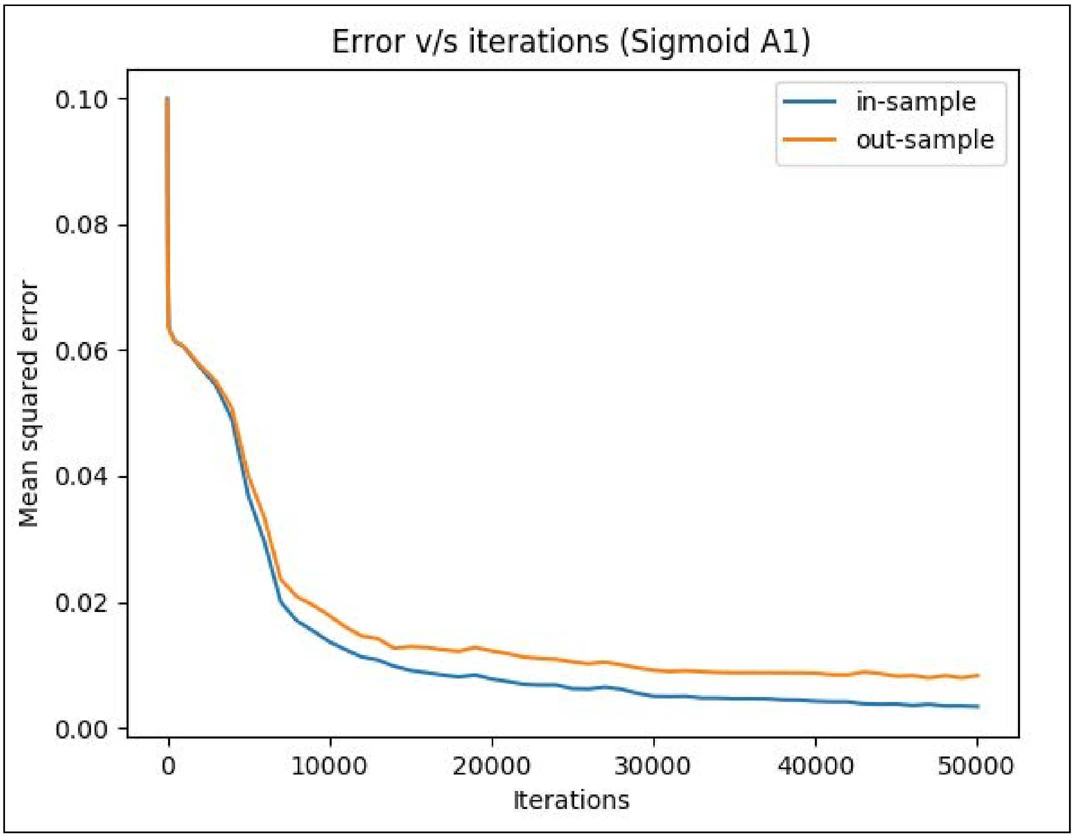
Machine Learning- Assignment 2
Report (15CS10053)

Neural Network for Spam mail classifier
- *2000* most frequent tokens have been used for input vector
- Learning rate of *0.1* used
- Ham mails labelled 0 and spam mails labelled 1
- **Spam is the relevant or important class** for calculating precision and recall
- *80:20* split for training and testing sets
- Both classes distributed proportionally in training and testing sets
- Mean squared error used as error metric
- Weights are initialised to random values in the range *[-0.1,0)*
- Backpropagation algorithm is run for ***50000 iterations***
- A random training instance is picked in every iteration (algorithm discussed in class)

Part A1 (*sigmoid* activation function)
Architecture:
- 2000 features in input vector plus 1 bias term
- 2 hidden layers
  - Layer 1: 100 neurons plus 1 bias node
  - Layer 2: 50 neurons plus 1 bias node
- Output layer consists of 1 neuron
- **Sigmoid** activation function used in every neuron

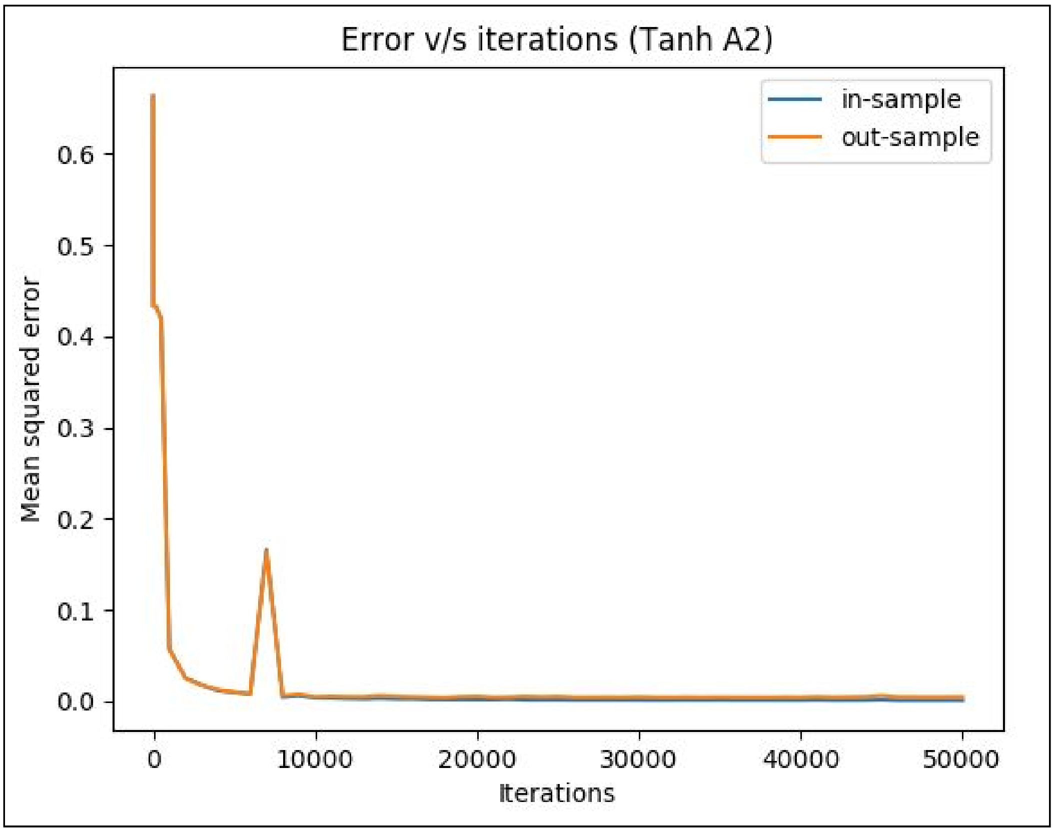


- Minimum **in-sample error** observed is **0.0033944819281865723**
- Minimum **out-sample error** observed is **0.0079653525946106 33**
- Statistics obtained on testing the model on the test set (*threshold set to 0.5*):
    - Precision : 0.9441559440559441
    - Recall : 0.9
    - F1 score : 0.9215017064846417
    - Accuracy : 0.9793906810035843
- Optimal number of iterations ~ **40000**

Part A2 (*tanh* activation function)

Architecture:

- 2000 features in input vector plus 1 bias term
- 2 hidden layers
    - Layer 1: 100 neurons plus 1 bias node
    - Layer 2: 50 neurons plus 1 bias node
- Output layer consists of 1 neuron
- **Hyperbolic tangent** activation function used in every neuron

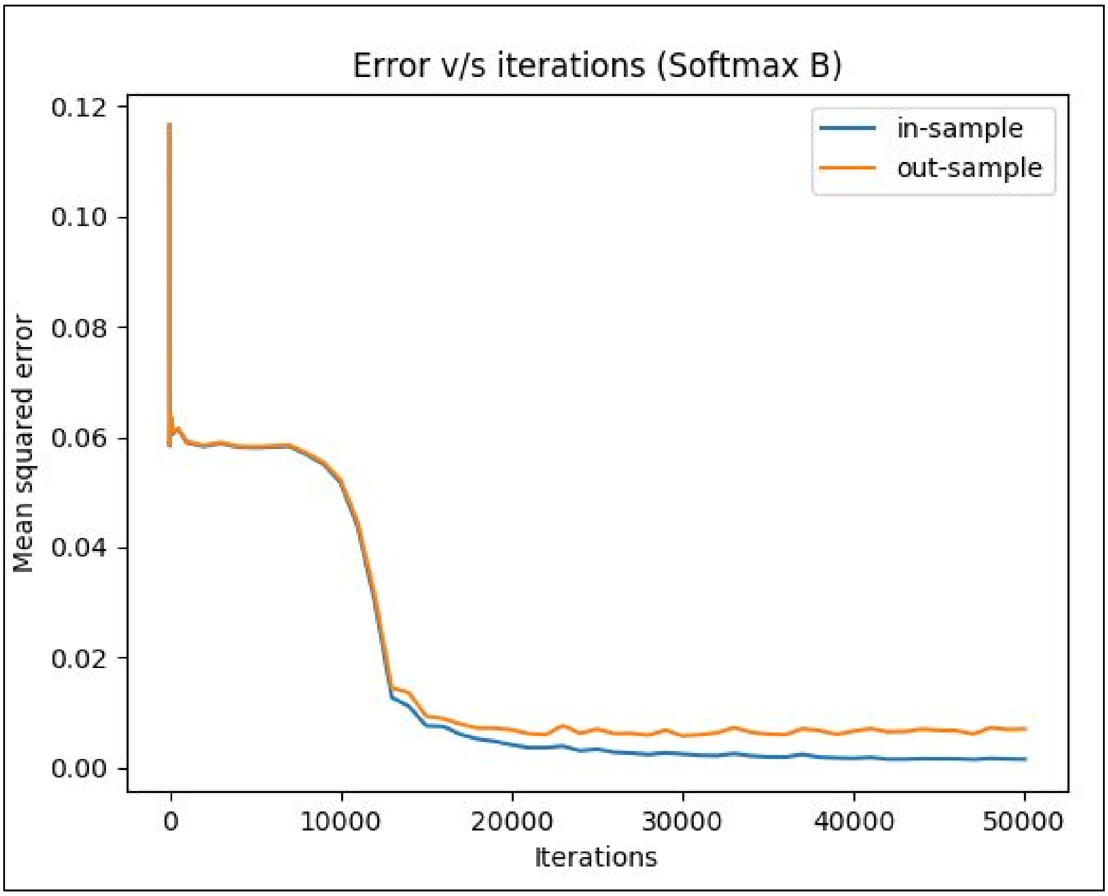


- Minimum **in-sample error** observed is **0.00079107038564751 82**
- Minimum **out-sample error** observed is **0.0038692926619930236**
- Statistics obtained on testing the model on the test set (*threshold set to 0.5*):
    - Precision : 0.9659863945578231
    - Recall : 0.9466666666666667
    - F1 score : 0.9562289562289563
    - Accuracy : 0.9883512544802867
- Optimal number of iterations ~ **10000**

Part B (*sigmoid* - layer 1 & 2 | *softmax* - layer 3)

Architecture:

- 2000 features in input vector plus 1 bias term
- 2 hidden layers
  - Layer 1: 100 neurons plus 1 bias node
  - Layer 2: 50 neurons plus 1 bias node
- Output layer consists of 2 neurons
  - 1st neuron outputs probability of mail being spam
  - 2nd neuron outputs probability of mail being ham
- **Sigmoid** activation function used in every neuron of layer 1 and layer 2
- **Softmax** activation function is used in output layer to convert the output of the network into non-negative probabilities of mail being spam/ham



- Minimum **in-sample error** observed is **0.001480134131550598**
- Minimum **out-sample error** observed is **0.005745730338309037**
- Statistics obtained on testing the model on the test set (*threshold set to 0.5*):
  - Precision *:* 0.9925373134328358
  - Recall *:* 0.8866666666666667
  - F1 score *:* 0.9366197183098592
  - Accuracy *:* 0.9838709677419355
- Optimal number of iterations **~ 40000**

Q: Which of the neural network architectures performs the best?

Architecture of **part A2** performs the best. **Hyperbolic tangent** activation function causes **the quickest convergence** at just 10000 iterations (benefit over sigmoid). It gives the **least mean square error** (both in-sample and out-sample). It also outperforms other architectures in terms of **accuracy, precision, recall** and **F1 score** on the test set.

Other inferences

- tanh activation function leads to ***faster convergence*** than sigmoid.
- Initialization range of weights is important. Sigmoid function gives >= *0.5* value for any positive input. Range *[-0.1,0)* for starting weights works for all parts.
- Too quick convergence implies a very simple model. Complexity can be increased by increasing the size of input vector (adding more tokens)
- Precision denotes the fraction of the mails our model classified as spam that were actually spam. (***precision = true positives/(true positives + false positives)***)
- Recall denotes the fraction of all spam mails that our model classified correctly as spam (***recall = true positives/(true positives + false negatives)***)
- Since our dataset is skewed (747 spam mails out of 5574 ~ 13.4%), simply classifying every mail as ham would land us an accuracy of ~86%
- Precision and recall become important performance metrics in such case