Programming Assignment #1 Report

Name : Kweon Hyuk Jun (권혁준)

Student ID: 2024710898

1. Objectives: Spam message detection code with ML

From known datasets from Kaggle, Differentiate Spam message (abnormal message) and Ham message (normal message)

2. Method

   By using pytorch for training and inferencing model and using scikit-learn for getting and evaluating performance of model

2.1. Detailed Approach

2.1.1. Preprocessing

By using pandas for loading dataset from spam.csv from Kaggle. By using the first row for label, which preprocessed to use 'Spam' as 1, and 'Ham' as 0 value and using second row for message data input and except these rows, all dataset is not used.

2.1.2. Tokenization

When getting dataset, encoded to vocabulary dictionary and replaced to integer value. And after that padded to same length for consistent dimensions.

2.1.3. Model selection

In Model selection, I used embedding layer, LSTM with 128 hidden layers, fully connected layer is 128 hidden layers and output function is out of sigmoid function

2.1.4. Training process

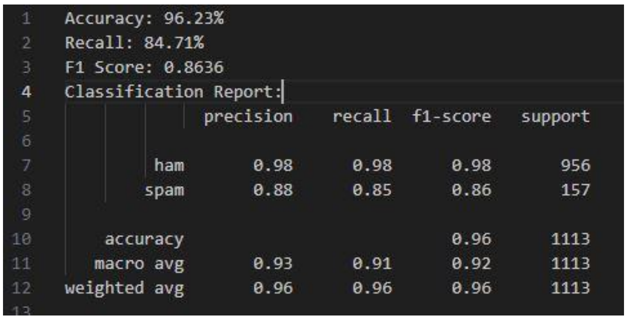
By using MAE loss function and Adam optimizer for backpropagate. MAE loss function is not going negative value than BCE function, so I can choose higher epoch value. So, we choose epoch = 40, as about epoch 35, loss converges

2.1.5. Analysis of model comparison

In model comparison, I used scikit-learn to divide dataset and inference, and using these two datasets, I can get confusion matrix and classification report with accuracy, recall and F1-Score. I chose MAE loss because MAE has accuracy 97.04% and F1-score is 0.8959, but BCE loss has 96.86% and F1-score 0.8837.

3. Result

Classification report result for MAE loss with epoch 40 is

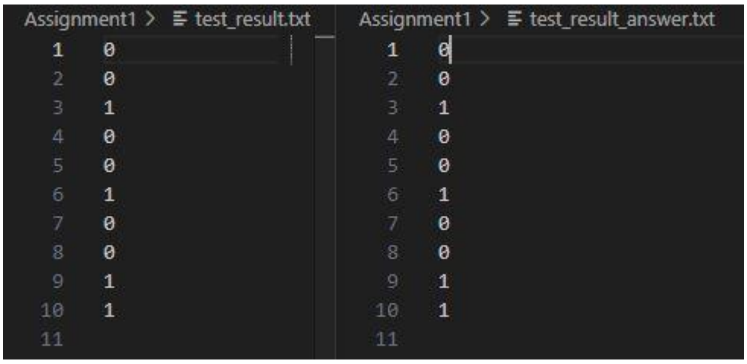


```
Accuracy: 96.23%
Recall: 84.71%
F1 Score: 0.8636
Classification Report:
              precision    recall  f1-score   support

         ham       0.98      0.98      0.98       956
        spam       0.88      0.85      0.86       157

    accuracy                           0.96      1113
   macro avg       0.93      0.91      0.92      1113
weighted avg       0.96      0.96      0.96      1113
```

Comparing Test value from test.csv (Left : expected from my model, Right : ground truth)



Assignment1 > test_result.txt
Assignment1 > test_result_answer.txt

```
0
0
1
0
0
1
0
0
1
1
```

```
0
0
1
0
0
1
0
0
1
1
```

4. Appendix

All of code is on https://github.com/KweonHyukJun/Data_security_HW1