CSC411: Assignment #3 Bonus

Due on Monday, March 24, 2018

Zhiyan Deng, Xin Jie Lee

March 24, 2018

Prefix

The code for this project is written in Python 3

Problem 1

***Part 1 (a)**: Build a better 'Fake News' classifier*

In project 3, we built fake news classifiers using naive bayes (NB), logistic regression (LR) and decision tree (DT) using one-hot encoding of unigrams as input features. For this bonus project, we attempt to improve the performance of the 'fake news' classifier through incoporating a wider range of methods and using an expanded feature set of unigrams, bigrams and trigrams. We started off by experimenting with new classifiers and features on the original project 3 datasets to find the optimal model before further training our selected model on a larger dataset for the competition.

After some experimentation, the multilayer perceptron neural network (MLP) stood out as the best performing single model. The inputs for the MLP are bag-of-words vectors where each vector's dimension is equal to the size of the vocabulary plus a bias unit. The vocabulary includes a combination of unigrams, bigrams and trigrams. Binary flags are used to indicate the presence of the n-gram in each headline. To build the optimal network, experiments were conducted with using different number of network layers, number of hidden units per layer, varying the dropout probability for each layer, using different combinations of unigrams, bigrams and trigrams as input features, and restricting the size of the vocabulary to the most common n-grams. Selected results of the experiments are located in Appendix A. The optimal MLP network used the full vocabulary of unigrams, bigrams and trigrams and 3 1024-dimensional hidden layers each with 40% dropout. After training on just the project 3 dataset, the performance and learning curves of the optimal network are shown below:

Multilayer Perceptron Neural Network Classifier
Final Training Set Performance: 96.28334062090074%
Final Validation Set Performance: 87.32106339468302%
Final Test Set Performance: 85.51020408163265%

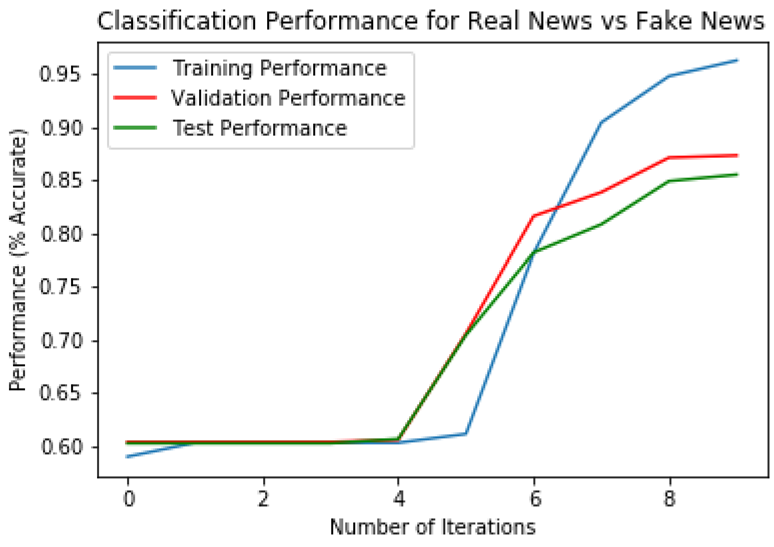

Figure 1: Classification accuracy vs epoch

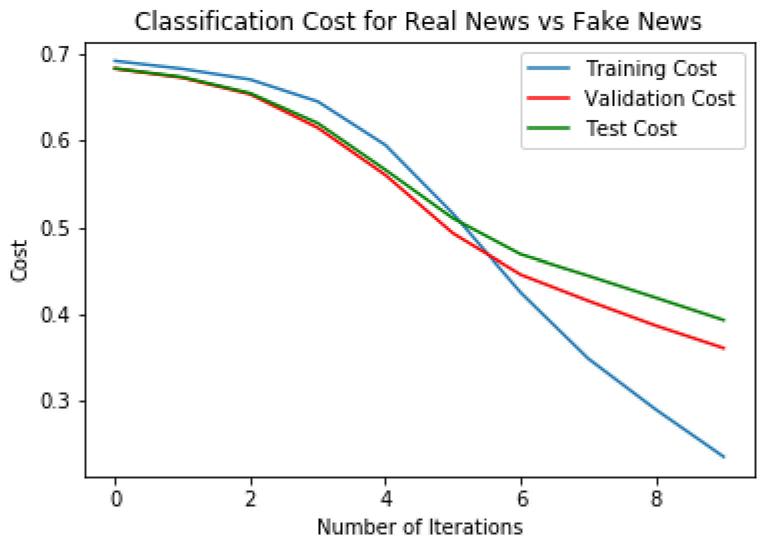

Figure 2: Cost vs epoch

It's performance on the validation and test sets are an improvement over the three classifiers used in Project 3: NB, LR and DT. As a comparison, these are the performance of the three classifiers:

Naive Bayes Classifier
Training Peformance: 96.93922168780061%
Validation Peformance: 87.73006134969326%
Test Peformance: 82.85714285714286%

Logistic Regression Classifier
Training Set Performance: 98.68823786620025%
Validation Set Performance: 86.29856850715747%
Test Set Performance: 82.6530612244898%

Decision Tree Classifier
Training Performance: 98.1198076082%
Validation Performance: 78.118609407%
Test Performance: 76.9387755102%

Among the other new classifiers that we tried, a classiier that exhibited relatively good performance was support vector machine (SVM). The inputs for our SVM model were term frequency inverse document frequency TFIDF vectors that took into account the frequency of the word in the news headline and the number of headlines that contain the word. The TFIDF of a word is the product of the term-frequency (TF) and the inverse document frequency (IDF) of the word. For example, if a word 'the' appears twice in a headline with 7 words, its TF will be $\frac{2}{7}$. However, if the dataset is only comprised of two news headlines and both contains 'the', the IDF of 'the' will be $log(\frac{2}{2}) = 0$. Hence, 'the' TFIDF will be $\frac{2}{7} \times 0 = 0$. If 'the' only appears in one of two documents, its IDF will be $log(\frac{2}{1}) \approx 0.301$, and its TFIDF will be $\frac{2}{7} \times 0.301 \approx 0.086$. In other words, TFIDF 'rewards' words that occur frequently in a document but 'penalizes' them if they are commonly found in other documents. In addition, we experimented with excluding words that have high document frequency and/or low frequency and/or are stopwords. The optimal model used unigrams as features, included english stopwords and ignored words that have document frequencies greater than 0.99 and less than 0.001. It uses a penalty term C = 0.9, gamma=0.5 and the 'rbf' (Gaussian) kernel and its performance on the project 3 dataset is shown below:

Support Vector Machine Classifier
Best Training Performance: 93.7261551505%
Best Validation Performance: 85.6115107914%
Best Test Performance: 84.693877551%

Other classifiers that we tried included both the random forest and XGBoost classifiers, however their performance were not on par with SVM and MLP. For the competition, we decided to use the SVM classifier instead of the MLP classifier since SVM scaled significantly better with a larger dataset and ran more efficiently.

Part 1 (b): *How the classifiers work*

This section would involve a brief discussion on how the two new better perfoming classifiers work, principally the MLP and SVM models.

The MLP is a class of feedforward artificial neural network model that contains at least 3 layers of nodes (1 or more hidden layers), and uses a nonlinear activation function for each node except the input nodes. The advantages of an MLP over a single layer perceptron (SLP) is that a MLP can learn non-linear functions, while a SLP can only learn linear functions. Dropout can be used to remove a percentage of the nodes in any nerual network layer to prevent overfitting of the model.

Support vector machines work by trying to find the decision boundary that maximizes the margin between different classes. By seperating the classes with the maximum margin, the chances of missclassification are reduced if a small error was made on the location of the decision boundary. In addition, doing so will also help to reduce the error on the validation set. For our optimal SVM model which uses a Gaussian kernel, we would classify a news headline x as 'real' news if the hypothesis $h(x) = \sum_k \alpha_k y^k K(x^k, x) + b \geq 0$ and 'fake' news if the hypothesis $h(x) < 0$ where x^k are the support vectors and the kernel is:

$K(x^i, x^j) = exp(\frac{||x^i - x^j||^2}{2\sigma^2})$

Appendix A

Selected results from training on Project 3 dataset

3 Layers (1 Hidden Layers) MLP
Top 10,000 unigrams and bigrams
Best Epochs: 25
Best Hidden Units: 128
Best Dropout: 0.1
Best Training Accuracy: 95.27765631832095%
Best Validation Accuracy: 87.93456032719837%
Best Test Accuracy: 83.6734693877551%

Top 10,000 unigrams, bigrams and trigrams
Best Epochs: 20
Best Hidden Units: 512
Best Dropout: 0.4
Best Training Accuracy: 94.70922606034105%
Best Validation Accuracy: 86.9120654396728%
Best Test Accuracy: 82.6530612244898%

All unigrams and bigrams
Best Epochs: 20
Best Hidden Units: 512
Best Dropout: 0.4
Best Training Accuracy: 97.55137735024049%
Best Validation Accuracy: 87.73006134969326%
Best Test Accuracy: 83.87755102040816%

All unigrams, bigrams and trigrams
Best Epochs: 20
Best Hidden Units: 512
Best Dropout: 0.5
Best Training Accuracy: 97.20157411456056%
Best Validation Accuracy: 87.32106339468302%
Best Test Accuracy: 84.48979591836735%

4 Layers (2 Hidden Layers) MLP
Top 10,000 unigrams, bigrams and trigrams
Best Epochs: 15
Best Hidden Units: 512
Best Dropout: 0.5
Best Training Accuracy: 92.91648447748142%
Best Validation Accuracy: 86.9120654396728%
Best Test Accuracy: 83.87755102040816%

All unigrams, bigrams and trigrams
Best Epochs: 10
Best Hidden Units: 1024
Best Dropout: 0.4
Best Training Accuracy: 96.06471359860078%
Best Validation Accuracy: 87.73006134969326%
Best Test Accuracy: 86.3265306122449%

SVM Including Stopwords
Gaussian 'rbf' Kernel
Best c: 0.9
Best gamma: 0.5
Best gram: (1, 1)
Best mx_df: 0.99
Best mn_df: 0.001
Best Training Performance: 93.7261551505%
Best Validation Performance: 85.6115107914%
Best Test Performance: 84.693877551%

Linear Kernel
Best c: 0.5
Best gamma: 0.1
Best gram: (1, 1)
Best mx_df: 0.99
Best mn_df: 0.001
Best Training Performance: 90.5892327257%
Best Validation Performance: 84.8920863309%
Best Test Performance: 83.8775510204%

SVM Excluding Stopwords

```
Gaussian 'rbf' Kernel
Best c: 1
Best gamma:
1 Best gram: (1, 2)
Best mx_df: 0.7
Best mn_df: 0.001
Best Training Performance: 96.5663416702%
Best Validation Performance: 82.9736211031%
Best Test Performance: 83.8775510204%

Linear Kernel
Best c: 0.5
Best gamma: 0.1
Best gram: (1, 2)
Best mx_df: 0.7
Best mn_df: 0.001
Best Training Performance: 89.7838066978%
Best Validation Performance: 0.822541966427%
Best Test Performance: 0.830612244898%
```

Results from grid search may vary slightly from results obtained from running individual models. This could be due to the fact that the weights were warmed up when successive models were ran continuously in the grid search.