Praneel Trivedy

Professor Yin

Foundations of Machine Learning

06 December 2022

System Description

I began this project by converting the 3 data files from text to csv. I opened the data into Microsoft Excel and converted the data into csv files so I could better operate.

The first step of my research was to understand how to structure my data so that it could be preprocessed and inputted to the neural network. The relevant packages used included: *Numpy, Pandas, Matplotlib, Re, String, NLTK, SKLearn, and CountVectorizer*. I initiated these at the beginning of my code so I could always reference back to them. After these libraries were initiated, I loaded the training and development sets into jupyter notebook. I glanced at the data to ensure it loaded correctly using head().

After a series of trial and error, I cleaned the code to create a function for preprocessing the sentences. The key factors I addressed was the removal of punctuation and stopwords. These two features made it significantly easier to find clarity within the instances.

In order to define a 'universe' which contained all possible words/tokens from all data, I created a new csv file. This file had one column with all sentences. This was tokenized into a list to with length 11,354. This was the number of unique words

The next step was to implement CountVectorizer() to prepare a bag of words representation for the data. I experimented with unigrams,. I observed the dimensionality of my resulting dataframe increased significantly as n increased. The resulting data frame for unigrams consisted of X rows/instances from the original dataset and the number of unique n-grams created using the CountVectorizer() function.

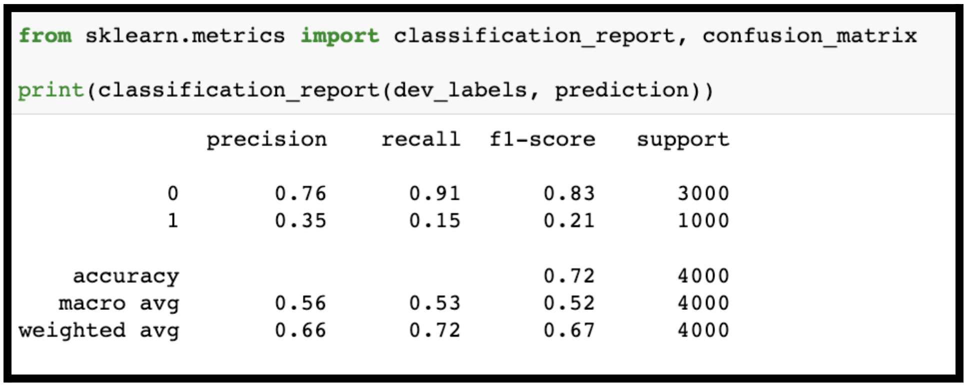
I for sentence 1 and 2 of both the train and dev set, I searched whether each sentence contained a token from the universe of words defined earlier.

- The resulting dataframe reported 0 if a word was not in either sentence for an instance.
- If a word was shared in one or both sentences, the frequency would be updated in a new dataframe.

For both the training and test data, I merged the two data frames resulting from the bag of words representation. The rows represented the number of sentences. The columns represented the number of unique words in the both sentence 1 and 2. These were large and sparse matrices because of the high dimensionality. I observed the the most a single word appeared in both sentences in the training data was 10 times. This was 7 times for the dev set.

In terms of the architecture for the neural network, we used MLP with an input layer, 2 hidden layers, and an output layer. The layers sizes were (100,50,3). Relu was the activation function. Gradient Descent ('Adam') was used. I set alpha = 0.001.

This was the result of my dev set:



```
from sklearn.metrics import classification_report, confusion_matrix

print(classification_report(dev_labels, prediction))
```

```
              precision    recall  f1-score   support

           0       0.76      0.91      0.83      3000
           1       0.35      0.15      0.21      1000

    accuracy                           0.72      4000
   macro avg       0.56      0.53      0.52      4000
weighted avg       0.66      0.72      0.67      4000
```