COEN 346 Natural Language Processing

Programming Assignment 2: Sentiment Analysis with Neural Networks and Word Embeddings (100 pts + 10 bonus pts)

Due on Camino: April 25, 5pm, 2024

In this assignment, you will do sentiment classification on the same dataset of Assignment 1 by using neural networks and word embeddings. Please use 5-fold cross validation to test the performance of your model. Below are the requirements:

1) Implement the logistic regression model as a neural network in PyTorch. Report the 5-fold cross validation accuracy.

2) Use word embeddings for feature representation in your PyTorch logistic regression. The basic idea of word embeddings is that each possible vocabulary word has a specific associated vector with a fixed size (e.g. 50 dimensions or 1000 dimensions). We have made available a large file of pre-trained length-50 embedding vectors for 400,000 possible vocabulary words, using a specific word embedding method called "GloVe". You can download the pretrained embeddings .zip file and example Python code to load the vectors in below. You can represent a sentence/review by averaging the vectors of words in the sentence.

   https://www.cse.scu.edu/~yfang/coen346/p2/glove.6B.50d.txt.zip
   https://www.cse.scu.edu/~yfang/coen346/p2/load_word_embeddings.py

3) Propose your own sentiment classifier by creating different neural networks and applying feature selection. You can experiment with different numbers of layers, different activation functions, and different training tips we introduced in class. Feature selection is to select a good set of words/features for classification, e.g., excluding stopwords such as “the”, “a”, etc., excluding rare words such as appearing in less than 10 documents, considering some bigrams such as “not bad”, etc. Apply your own sentiment classifier to predict sentiment labels on x_test.txt and generate yprob_test.txt. The students who achieve the top 5 accuracy on the test data in class will receive 10 bonus points.

What to submit to Camino:

- yprob_test.txt: plain text file. Please use the exactly same file name in your submission.
  - Each line contains float probability that the relevant example should be classified as a positive example given its features. Please follow the order in the x_test.csv file for the test examples.

- A report to summarize your results.
- Either .py or .ipynb file for the source code.