

## CSEN 346 Natural Language Processing

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

Due on Camino: April 30, 5pm, 2025

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](https://www.cse.scu.edu/~yfang/coen346/p2/load_word_embeddings.py)

You don't have to know how GloVe was implemented for this assignment, but if you are interested in the details, you can find more information about GloVe at <https://nlp.stanford.edu/projects/glove/>

- 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`. You can use whatever methods but please do not directly use off-the-shelf classifiers in ML libraries. The students who achieve the top 5 accuracy on the test data in class will receive 10 bonus points. The students who achieve the top 6-10 accuracy will receive 5 bonus points.

What to submit to Camino:

- A report in PDF to summarize your results with a brief description about your models and whether the results are reasonable.
- yprob\_test.txt: plain text file. Please use the exactly same file name in your submission. o Each line contains float probability that the relevant example should be classified as a positive example **by your best sentiment classifier**. Please follow the order in the x\_test.csv file for the test examples.
- Either .py or .ipynb files for the source code.