DS-GS 1011 Fall 2018
Bag of N-Gram Document Classification

Elman Mansimov and Phu Mon Htut

September 2018

1 Assignment

In this assignment, you are required to build a bag of n-grams model for predicting the sentiment of the movie reviewers given the textual review for the movie.

We will use IMDB Movie review dataset consisting of $25,000$ train and $25,000$ test movie reviews scraped from IMDB website.[1] Each movie review has a positive or negative sentiment, and you are expected to treat this problem as a classification problem. We recommend that you split the train dataset into $20,000$ train examples and $5,000$ validation examples. **Performing hyperparameter exploration on the test set will result in the grade of** 0 (Choose the best model on the validation set and only report the results on test set at the end). You are expected to preprocess the dataset, implement bag of n-grams classifier, do hyperparameter tuning on the validation set, and report the result of the best model on the test set.

We recommend you try different preprocessing and hyperparameters for the model, including but not limited to:

- Tokenization schemes of the dataset.
- Model hyperparameters: Vary n for n-gram (n=$1, 2, 3, 4$), vocabulary size and embedding size.
- Optimization hyperparameters: Optimizer itself (SGD vs Adam), learning rate and whether or not you use linear annealing of learning rate (learning rate is reduced linearly over the course of training).
- You are welcome to experiment with more hyperparameters.

We expect you to do the ablation study and report different results in the table and plot training curves. Also, list 3 correct and 3 incorrect predictions of your model on the validation set.

[1] `http://ai.stanford.edu/~amaas/data/sentiment/`

There is no skeleton code provided but we recommend that you use the starter PyTorch code from lab session on Bag of Words model. You are welcome to use FastText library for sanity-check after you implemented PyTorch version of code, but using purely FastText library (or any other classification library) to report the results will give you mark of 0. Please let us know if you have questions on Piazza or visit us during office hours.

We require that you submit LaTeX formatted PDF report with analysis of the results (max 4 pages) as well as link to public GitHub repo with your code (Jupyter Notebook recommended but you are welcome to submit python files without notebook). To get full mark, we expect that you get at least 70% accuracy on the test set. You will get 0 mark if you do not submit the written report and the link to public github repo. We will take away 15 points if you submit the link to github repo late or commit to the repo after deadline. We will take away 5 points each for every listed hyperparameter that you do not experiment with. We will give bonus points if you additionally predict the rating of the review (between 1-10) and/or show if there is any other hyperparameter not listed that affect the performance of the model.