ASSIGNMENT 4

CS5304 - SENTIMENT ANALYSIS USING CONVOLUTIONAL NEURAL NETWORKS

In this assignment, you'll recreate the CNN for NLP model from Kim EMNLP 2014 for Sentiment Analysis. This CNN model should be trained on the Stanford Sentiment Treebank dataset that can be downloaded from `http://nlp.stanford.edu/~socherr/stanfordSentimentTreebank.zip`. More details on the dataset can be found at the SST site. Please note that the zip file contains several files. You'll be using the **dictionary.txt** file which contains the phrases and their corresponding ids. The sentiment labels for these phrase ids are given in **sentiment_labels.txt**. You'll have to convert the raw score into 5-classes as follows: $[0, 0.2] \rightarrow 0, (0.2, 0.4] \rightarrow 1, (0.4, 0.6] \rightarrow 2, (0.6, 0.8] \rightarrow 3, (0.8, 1.0] \rightarrow 4$. The phrase ids for the **training**, **validation**, and **test** partitions are given in the following links. Please use them in your experiments.

(1) Training
(2) Validation
(3) Test

You'll conduct a total of 6 experiments as follows:

(1) Train a CNN Sentiment Classifier using pre-trained and frozen skip-gram word2vec as your word embedding on the Sentiment data from above
(2) Train a CNN Sentiment Classifier using pre-trained and frozen GloVe vectors as your word embedding instead of skip-grams
(3) Train a 2-channel CNN Sentiment Classifier using both pre-trained and frozen GloVe and skip-grams
(4) Repeat experiment 1 allowing the skip-grams to be fine-tuned for this task
(5) Repeat experiment 2 allowing the GloVe vectors to be fine-tuned for this task
(6) Repeat experiment 3 allowing both GloVe and skip-grams to be fine-tuned

Here are some helpful links:

- `https://github.com/DSKSD/DeepNLP-models-Pytorch`
- `https://github.com/mmihaltz/word2vec-GoogleNews-vectors`
- `https://nlp.stanford.edu/projects/glove/`
- `https://stackoverflow.com/questions/37793118/load-pretrained-glove-vectors-in-python`
- `https://gist.github.com/mrdrozdov/2a5049914db63136199556f18dfbf854`

For your submission, we'll expect the following in CMS.

- Your PyTorch code (**assign4.py**) which contains your implementation of CNN for Sentiment Analysis, including simple instructions on how the code can be used to run the 6 experiments.

- A text file called **results.txt** that contains your class labels for each phrase in the **test** partition. These class labels should take Integer values of $(0, 1, 2, 3, 4)$ as specified above. The format of each row in the file should be
  $<$ **phraseid** $>$ | $<$ **label** $>$
- A writeup called **assign4_writeup.pdf** that discusses your experiments along with accuracy measures (evaluated on your validation set) for all 6 models summarized in a table.

You'll be evaluated as follows:

- Results that exceed expected baseline and experiments done properly (e.g. not trained on the test set). 6 points
- Writeup that demonstrates all 6 experiments along with their results. For full credit, you?ll need to compare the performance of your models in a non-trivial way. Hint: Did one model to especially better than others on a particular label? On a particular phrase length? 6 points.
- Working Python code file that contains no errors. 3 points

To ensure that we are able to recreate your experiments exactly (in case we need to), do hard-code any random initializations that you use in assign4.py. If we are unable to recreate your results using your submitted assign4.py file, we'll deduct points.