Assignment 4: Model Comparison

Homework assignments will be done individually: Each student must hand in their own answers. Use of partial or entire solutions obtained from others or online is strictly prohibited. Electronic submission on Canvas is mandatory.

This assignment focuses on evaluating convolutional neural networks (CNN) and recurrent neural networks (LSTM or bidirectional LSTM) on two NLP tasks: document classification and sentimental analysis.

1. **Document Classification** (50 points) Classify text paragraphs into three categories: *Fyodor Dostoyevsky*, *Arthur Conan Doyle*, and *Jane Austen* by building your own classifiers. The data provided is from Project Gutenberg.

   (a) (5 pts) **Preprocess the data**: build the vocabulary, tokenize, etc. Each training/testing example is a paragraph. Divide the data into train, validation, and test. You can initialize word vectors using pre-trained embeddings such as Glove or FastText.

   (b) (10 pts) **Build your models**: Initialize parameters for each model (a LSTM and a CNN). Implement the forward pass for the model. Use an embedding layer as the first layer of your network (e.g. `tf.nn.embedding_lookup`). For the CNN model, use zero paddings to the input matrix and apply one convolutional layer (includes convolution, activation, and maxpooling). For the LSTM model, try (1) use the element-wise average (or max) of all hidden states as the context vector for prediction and (2) the last hidden state as the context vector for prediction.

   (c) (10 pts) **Backpropagation**: Calculate the loss of the model (cross-entropy loss is suggested). Write the backpropagation process. Set up the training steps: use a learning rate of $1e-3$ and an Adam optimizer.

   (d) (10 pts) **Train you model** and report the recall, precision, and F1 of each class on test data. Tune the parameters to achieve the best performance using the validation set. Report the search range of your hyper parameters (e.g., hidden state size for LSTM, number of filters, the filter size for CNN) and the best values you choose.

   (e) (15 pts) **Compare the four models** you have tried on the test set in a table: (1) Logistic Regression from Assignment 1, (2) Multilayer Perceptron (MLP) from Assignment 1, (3) Convolutional Neural Networks, (4) LSTMs. **Report recall, precision, and f1 score for each category in the test set**. The first two models use TF-IDF features and the last two models use word embedding vectors.

2. **Sentiment Analysis** (50 points)

   The task is to classify multi-domain reviews into positive or negative. We only use the book reviews for this assignment. There are 1000 positive book reviews and 1000 negative book reviews. The original data can be found here.

   Follow the steps from (a) to (e) in the last question to implement and compare models. Specifically, you need to extract text from <`review_text`>. The output layer prediction function can be a sigmoid function and the loss function for this task can be binary cross-entropy loss. If your test set is balanced (same number of positive and negative samples), you can just report accuracy. You can just compare CNN and LSTM in this task.

Submission Instructions You shall submit a zip file named Assignment4_LastName_FirstName.zip which contains: (Those who do not follow this naming policy will receive penalty points)

- python files (.py or .ipynb) including all the code, comments, and results. You need to provide detailed comments in English.
- report(.pdf) for each task: Describe your model: size of the training/validation/test set; hyper-parameters and how you select these values, loss function, learning rate, optimizer, etc. Plot the curves of training and validation loss. Compare different models and report recall, precision, and f1 score for task 1, and accuracy score for task 2 on test data.

Further Reading:

- Yoon Kim. Convolutional Neural Networks for Sentence Classification. ACL 2014. arXiv:1408.5882
- Ye Zhang, Byron Wallace. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. arXiv:1510.03820
- Jenq-Haur Wang, Ting-Wei Liu, Xiong Luo, Long Wang. An LSTM Approach to Short Text Sentiment Classification with Word Embeddings. ROCLING/IJCLCLP 2018
- Lei Zhang, Shuai Wang, Bing Liu. Deep Learning for Sentiment Analysis : A Survey. arXiv:1801.07883