Assignment 4: Convolutional Neural Networks

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 convolutional neural networks. You will need to implement convolutional neural network models for two tasks: document classification and sentimental analysis.

1. **Document Classification** (50 points) Use the same datasets as Assignment 1. 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) (10 pts) Preprocess the data: build the vocabulary, tokenize, etc. Divide the data into train, validation, and test.
   (b) (10 pts) Initialize parameters for the model. Implement the forward pass for the model. Use an embedding layer as the first layer of your network (e.g. `tf.nn.embedding_lookup`). Set zero paddings to the input matrix. Use at least two convolutional layers (each layer includes convolution, activation, and maxpooling).
   (c) (10 pts) Choose and report the number of filters and the filter size for your CNN.
   (d) (10 pts) Calculate the loss of the model (cross-entropy loss is suggested). Set up the training step: use a learning rate of $1e-3$ and an Adam optimizer.
   (e) (10 pts) Train you model and report the recall and precision of each class on test data. Tune the parameters to achieve the best performance you can.

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

   This is a multi-domain sentiment dataset with positive or negative sentiment annotations. We only use the book reviews for this assignment. There are 1000 positive book reviews and 1000 negative book reviews.
   (a) (10 pts) Preprocess the data: extract the review text from <`review_text`>, build the vocabulary, tokenize, etc. Divide the data into train, validation, and test.
   (b) (10 pts) Initialize parameters for the model. Implement the forward pass for the model. Use an embedding layer as the first layer of your network (e.g. `tf.nn.embedding_lookup`). Set zero paddings to the input matrix. Use at least two convolutional layers (each layer includes convolution, activation, and maxpooling).
   (c) (10 pts) Choose and report the number of filters and the filter size for your CNN.
   (d) (10 pts) Calculate the loss of the model (binary cross-entropy loss is suggested). Choose appropriate output function. Set up the training step including learning rate and optimizer.
   (e) (10 pts) Train you model and report the accuracy of each class and the total accuracy on test data. Tune the parameters to achieve the best performance you can.

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) 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 set and validation set, parameters for your model, number of filters, filter size for you CNN model, loss function, learning rate, optimizer, etc. Plot for training and validation loss. Report recall and precision 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