CS 563: Natural Language Processing
Assignment-3: Sentiment Analysis

Deadline: 14 April 2023

- Markings will be based on the correctness and soundness of the outputs.
- Marks will be deducted in case of plagiarism.
- Proper indentation and appropriate comments (if necessary) are mandatory.
- Use of frameworks like scikit-learn, PyTorch etc is allowed.
- *All benchmarks(accuracy etc), answers to questions and supporting examples should be added in a separate file with the name 'report'*.
- *All code needs to be submitted in '.py' format.* Even if you code it in '.IPYNB' format, download it in '.py' format and then submit
- You should zip all the required files and name the zip file as:
    - <roll_no>_assignment_<#>.zip, eg. 1501cs11_assignment_01.zip.
- Upload your assignment (the zip file) in the following link:
    - https://www.dropbox.com/request/RI2PGWmrlpAot9PLzaFP

Problem Statement:

- The assignment targets to implement Feed-Forward NN and RNN for Binary and Multi-class sentiment analysis

Implementation:

Input features:

- Tokenize the dataset and consider words with frequency >= 5
- Assign "UNK" token to all other remaining words
- During testing, if a word is not in vocabulary it should be taken as "UNK"
- Use spaCy English tokenizer for tokenizing the data (link: https://spacy.io/models)

Input to the Neural Network:

- Input to the NN should be one-hot encoding of input tokens
- For example, given the following sentence:

    I love watching anime and reading manga .

- Vocabulary size: **8** (I, love, watching, anime, and, reading, manga, . (including fulls stop at the end))

- The one-hot encoding for the tokens is as follows:

  I: **[1, 0, 0, 0, 0, 0, 0, 0]**
  love: **[0, 1, 0, 0, 0, 0, 0, 0]**
  …
  manga: **[0, 0, 0, 0, 0, 0, 1, 0**]
  . : **[0, 0, 0, 0, 0, 0, 0, 1]**

- The input sentence is now can be represented as tensor of one-hot encoded vectors as:

  [[1, 0, 0, 0, 0, 0, 0, 0],
  [0, 1, 0, 0, 0, 0, 0, 0],
  …,
  [0, 0, 0, 0, 0, 0, 0, 1],]

- The size of the input tensor is: [1, 8, 8] (1: batch size (because the current batch contains only one sentence), 8: sentence length, 8: one-hot vector length (same as the size of vocabulary))
- Since the network takes a fixed length input, longer sentences should be truncated after a maximum length and the smaller sentences should be padded
  - For maximum length: Use the average length of the corpus as maximum length after tokenization. For example if the average length of the corpus is 60 tokens, then the maximum length should be set to 60. Average length of corpus = (Total no. of words in corpus / Total no. of sentences in corpus)
  - For minimum length: A special token "**PAD**" should be added to vocabulary and fill the remaining positions with this PAD token. For example if the maximum length is 10, the following sentence is padded as:
    - **I love watching anime and reading manga . PAD PAD**
  - Previously input has a length of 8, now we padded to the maximum length of 10
- In general, both Feed-Forward NN and LSTM inputs should be prepared in this way

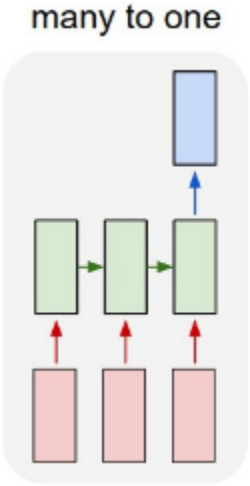
Feed-Forward NN:

- Explain and draw the architecture of Feed-Forward NN that you are proposing with justification. Describe the features of Feed-Forward NN.
- Network should contain **TWO** hidden layers
  - input — hidden_layer_1 (hidden_layer_1 size is 256)
  - hidden_layer_1 — hidden_layer_2 (hidden_layer_2 size is 128)

- Finally, hidden_layer_2 — Output (Output depends upon no. of classes)
- Use non-linearity of your choice (tanh, relu, gelu etc.) between hidden layers
- Clearly discriminate between binary class and multi class loss functions

RNN:

- Use the following architecture for RNN based model (ref: https://karpathy.github.io/2015/05/21/rnn-effectiveness/)



- After reaching the end of the sentence, the last state is used to classify the input (hence the prediction is at the end)
- Conduct experiments on LSTM (not on base RNN model)
- Hidden layer size: 256
- Output size depends upon no. of classes
- Clearly discriminate between binary class and multi class loss functions

Dataset:

- IMDB (binary class) dataset:
    - Dataset consists of movie reviews and each review is tagged with its corresponding sentiment tag (positive or negative)
    - Link:
        - https://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz
    - Example:
        - **Input:** If you like adult comedy cartoons, like South Park, then this is nearly a similar format about the small adventures of three teenage girls at Bromwell High. Keisha, Natella and Latrina have given exploding sweets and behaved like bitches, I think Keisha is

a good leader. There are also small stories going on with the teachers of the school. There's the idiotic principal, Mr. Bip, the nervous Maths teacher and many others. The cast is also fantastic, Lenny Henry's Gina Yashere, EastEnders Chrissie Watts, Tracy-Ann Oberman, Smack The Pony's Doon Mackichan, Dead Ringers' Mark Perry and Blunder's Nina Conti. I didn't know this came from Canada, but it is very good. Very good!
        - **Label: Positive**

- SemEval (multiclass) dataset:
    - Dataset consists of tweets and each tweet is tagged with its corresponding sentiment tag (positive, negative or neutral)
    - Link:
        - https://drive.google.com/drive/folders/1s_KZY0oIyCDkn6lK23_2q_ToEl30HQju?usp=share_link
    - Example:
        - **Input:** Gas by my house hit $3.39!!!! I\u2019m going to Chapel Hill on Sat. :)
        - **Label: Positive**

Evaluation:

- Run each model for 10 epochs
- For **SemEval** dataset use the given **dev** set as validation set and for **IMDB** dataset, use **last 10% samples** of train set as dev set
- Save best model checkpoint based on the **Accuracy of dev set**
- Report overall **Accuracy, Precision, Recall and F-Score** and also for each label on the best model checkpoint on **test** set

Documents to submit:

- Model code
- Model logs (in the form of graph):
    - Accuracy of dev set for each epoch
    - Train loss for each epoch
- Write a report (doc or pdf format) on how you are solving the problems as well as all the results including model architecture (if any).

For any queries regarding this assignment, contact:

Gopendra Singh Vikram (gopendra.99@gmail.com),

Mamta (mamta20118@gmail.com),
Aizan Zafar (aizanzafar@gmail.com),
Ramakrishna Appicharla (ramakrishnaappicharla@gmail.com) and,
Arpan Phukan (arpanphukan@gmail.com)