Cairo University
Faculty of Computers and Artificial Intelligence
Deep Learning Course

Assignment 2 - Twitter Sentiment Analysis

About the Problem:

Twitter sentiment analysis is an application of natural language processing in which the text data of tweets are analyzed and classified according to their polarity *(i.e. positive, negative, or neutral)*. It can be used for a wide range of applications like monitoring people's opinions about various topics, predicting stock market trends based on tweets about certain companies, and so on.

Dataset:

For the Twitter sentiment analysis application, Sentiment140 is a popular dataset. It consists of about 1.6 million tweets with their polarities and some other fields. In the dataset, the polarity of a tweet can be 0 *(negative)*, or 2 *(neutral)*, or 4 *(positive)*.

In this assignment, you will use the Sentiment140 dataset, or just a fraction of it *(depending on your resources)*.

Requirements:

In a notebook, write Python code where you:

1. Load the data, and keep tweets that are either positive or negative; i.e. discard "neutral" tweets.
2. Perform the necessary ***text preprocessing*** on the text of the tweets.
   (You will be asked which preprocessing steps you performed and why)
3. Build ***two*** models that classify tweets using word embeddings and long short-term memory (***LSTM***) such that:
   - Both models have ***one*** LSTM layer.
   - The first model uses ***trainable word embeddings***.
   - The second model uses the ***pre-trained Word2Vec embeddings***.
4. Train each model, evaluate its performance using the ***accuracy***, and the ***F1 score*** and its macro average, and handle ***overfitting*** if it occurs.
   *(At least one of your models should reach an accuracy that is greater than **75%**)*
5. Implement the functions "***predict***" and "***LSTM_forward***" ***from scratch*** such that:
   - The "*predict*" function takes a batch of examples and returns their predicted probabilities/classes. Firstly, it calls the ***trained embedding layer*** *(of one of your models)* on the batch, and extracts the ***masks*** produced with the embedding output as well. Then, it calls the "*LSTM_forward*" function on each example, and returns the outputs when the whole batch has been processed.
   - The "*LSTM_forward*" function performs the ***forward propagation*** of the LSTM and the final dense layer *(using the same model's **trained weights**)*.

   Note that:
   - You will need to extract the weight matrices of the LSTM layer *(and the bias vectors if they were used)*, in addition to the dense layer weights and bias, from the trained model. You can also store them in global variables so that they can be easily used by your functions.
   - When applying forward propagation on an example of the batch, the "*LSTM_forward*" function should only consider timesteps whose corresponding inputs were not masked in that example.
   - **You are allowed to use ChatGPT to help you write the code of this requirement *only***. However, you must understand every part of that code, how it works, why it is needed, etc.
6. Test your functions by taking a small sample of the test data and feeding it to your "*predict*" function, then ***compare the performance*** *(time and memory)* ***and outputs*** to those of the model's "*predict*" method.
   (If your "predict" function works correctly, it will predict the same classes as those predicted by "model.predict")

Deliverables:

You are required to submit **one zip file** containing only the following:

- Your notebook *(**.ipynb**)*.
- Your code as *(**.py**)* file.
- A report *(**.pdf**)* containing the team members' names and IDs, the code with the output of each part, and ***your insights and comments*** on the results of requirements 4 and 6.

The zip file must follow this naming convention:
ID1_ID2_ID3_ID4_ID5_Group

Submission Remarks:

- The ***minimum*** number of students in a team ***is 3*** and the ***maximum is 5***.
- All team members must understand every part of the code during the discussion.
- ***No late submission*** is allowed.
- ***Plagiarism*** is totally prohibited and ***will not be tolerated***. If similarity between your code and any other source is detected, you will be assigned ***ZERO*** without argument, and no excuses will be accepted.
- A ***penalty*** will be ***imposed for violating*** any of the assignment rules.

Grading Criteria:

Data preparation and text preprocessing	1 mark
First model	1 mark
Second model	1.5 marks
Models' evaluation	1 mark
LSTM's forward propagation code and usage	1.5 marks
Total	**6 marks**