Phuong Dinh – 1835957

HW2 – Task 4

BERT Model	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
BERT-Tiny (L-2_H-128_A-2)	0.8146	0.80648	0.824166525638856	0.7792	0.8010527181511639
BERT-Mini (L-4_H-256_A-4)	0.8374	0.83608	0.8614696265702977	0.80096	0.8301135892546223
BERT-Small (L-4_H-512_A-8)	0.8518	0.8468	0.890892696122633	0.7904	0.8376430690970751
BERT-Medium (L-8_H-512_A-8)	0.8444	0.84184	0.9000187230855645	0.76912	0.829436631869554

- For this assignment, I used Jupyter notebook to run my training locally. First, I download the training dataset, test dataset, and stored them under train/ and test/. After which, I used np.load to load in the npz files, converted both to pandas Dataframe. Secondly, for the train data, I used train_test_split to get 80% train data and 20% validation data. For the test data, I left untouched. For both I added in the names of the column which are DATA_COLUMN and LABEL_COLUMN. I used the code given in Steve's lab to process the two Dataframes into tokens to be passed as input into the BERT model.

- Regarding hypermeters, I used optimizer Adam with learning_rate = $3e^{-5}$, epsilon = $1e^{-8}$, clipnorm = 1.0, batch_size for all were 32, loss function was Sparse Categorical Crossentropy, and epoch was 1. I used these hyperparameters since it was provided from the test code and the results were decent so there was no point in fine tuning at the current stage. However, I have created a fine-tuned fifth model specifically to increase F1 score and accuracy.

- After testing four different BERT models from tiny to medium, the result showed that the small BERT model performed the best out of all four. However, it was clear that more complex BERT model performed better overall. Specifically, when moving from the tiny model to the mini model, there was an increase of about 0.02 across train accuracy, test accuracy, and f1 score. There was a slight decrease in performance from the small model to the medium model, but it was minimal and could be contribute to training error. I believe the reason why bigger BERT model is better is due to the increase in parameters in complex BERT model.

- Using the previous training knowledge, I have built a fifth model using a more complex BERT (L-12_H-768_A-12). Moreover, I adjusted the batch down to 16 instead of 32, used 35% validation data, changed to Adamax optimizer at learning_rate = 5e-5. For other hypermeters, I kept the same. Attached is my fine-tuned result:

BERT (L-12_H-768_A-12)	0.8675	0.87356	0.8530812854442344	0.90256	0.8771234207968902

I chose these hyperparameters specifically since it was the best result from my quick grid evaluation which evaluated different batch size, optimizer, and learning rate. I also tried to increase the epoch to 3 but it was obvious after the second epoch that the model was overfitting with high accuracy for train data but high loss and low accuracy for the validation data. Therefore, I decided to not use that model. Overall, the fine-tuned model provided an additional 0.02 increase in accuracy and 0.04 in F1 score from the previous best model (L-4_H-512_A-8).