Task 4 Report

Arno Dunstatter, 1926976

Discussion of Preprocessing

The critical step of preprocessing was primarily achieved via the tokenizer generated for each specific model-type by using the `BertTokenizer.from_pretrained()` method. The respective tokenizers for each model type are used directly to produce tokenized reviews for testing and indirectly within the `convert_examples_to_tf_dataset()` method as the tokenizer's `.encode_plus()` method is utilized to produce the input dictionary containing the input ids, token type ids, and attention masks which are themselves passed to the `InputFeatures()` method to produce each example's features list which is itself later used by the `tf.data.Dataset.from_generator()` method to produce the tensorflow dataset utilized for fine tuning each pre-trained BERT type. From what I can tell from this youtube video, essentially each head pays attention to a set of linear combinations of the input embeddings sumed with positional embeddings and projected into different 'semantic spaces', and each layer transforms the concatenation of the previous layers' embedding into increasingly information rich embeddings.

Results

Set 1

	learning_rate	epochs	epsilon	clipnorm
0	0.0003	1	1.000000e-08	1.0

	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
L-2_H-128_A-2	0.8332	0.79688	0.781734	0.82376	0.802197
L-4_H-256_A-4	0.8224	0.79592	0.787189	0.81112	0.798976
L-4_H-512_A-8	0.7950	0.77080	0.787547	0.74168	0.763926
L-8_H-512_A-8	0.5028	0.50000	0.500000	1.00000	0.666667

Set 2

	learning_rate	epochs	epsilon	clipnorm
0	0.0003	2	1.000000e-08	1.0

	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
L-2_H-128_A-2	0.8296	0.76928	0.734173	0.84424	0.785369
L-4_H-256_A-4	0.8180	0.77324	0.742286	0.83712	0.786856
L-4_H-512_A-8	0.7822	0.65432	0.600921	0.91888	0.726640
L-8_H-512_A-8	0.5028	0.50000	0.500000	1.00000	0.666667

Set 3

	learning_rate	epochs	epsilon	clipnorm
0	0.00003	1	1.000000e-08	1.0

	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
L-2_H-128_A-2	0.8082	0.80092	0.768621	0.86104	0.812210
L-4_H-256_A-4	0.8368	0.83504	0.812584	0.87096	0.840760
L-4_H-512_A-8	0.8518	0.85500	0.851318	0.86024	0.855756
L-8_H-512_A-8	0.8480	0.84936	0.909816	0.77560	0.837364

Set 4

	learning_rate	epochs	epsilon	clipnorm
0	0.00003	2	1.000000e-08	1.0

	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
L-2_H-128_A-2	0.8164	0.81632	0.802942	0.83840	0.820288
L-4_H-256_A-4	0.8380	0.83344	0.869046	0.78520	0.824998
L-4_H-512_A-8	0.8460	0.84720	0.872468	0.81328	0.841835
L-8_H-512_A-8	0.8502	0.85292	0.861628	0.84088	0.851128

Set 5

	learning_rate	epochs	epsilon	clipnorm
0	0.000003	1	1.000000e-08	1.0

	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
L-2_H-128_A-2	0.6202	0.62116	0.603414	0.70696	0.651096
L-4_H-256_A-4	0.7876	0.78164	0.777227	0.78960	0.783364
L-4_H-512_A-8	0.8252	0.81916	0.800573	0.85008	0.824584
L-8_H-512_A-8	0.8430	0.84388	0.838545	0.85176	0.845101

Set 6

	learning_rate	epochs	epsilon	clipnorm
0	0.000003	2	1.000000e-08	1.0

	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
L-2_H-128_A-2	0.6290	0.69616	0.684389	0.72808	0.705559
L-4_H-256_A-4	0.7622	0.80424	0.827958	0.76808	0.796896
L-4_H-512_A-8	0.8196	0.83664	0.845939	0.82320	0.834415
L-8_H-512_A-8	0.8394	0.85496	0.872482	0.83144	0.851466

Full Bert Model

	learning_rate	epochs	epsilon	clipnorm
0	0.00003	1	1.000000e-08	1.0

	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
bert-base-uncased	0.8772	0.87536	0.920505	0.82168	0.86829

Discussion

In an attempt to improve performance I performed a rudimentary grid-search over six combinations of learning rates and epoch counts; for each hyperparameter configuration all four models were attempted. Additionally the bert-base-uncased model was evaluated with Mr. Aigbe's original hyperparameter configuration. The results are shown above. The hyperparameters were chosen to encircle the configuration originally used in the lab since that configuration gave pretty good results to begin with. Ultimately increasing the learning rate had a strong negative impact on performance, presumably overshooting local optima from too large of steps, while decreasing the learning rate also deteriorated accuracy, albeit not as much; the reason for this effect is not yet clear to me. The best hyperparameters were given by Set 4 with the original learning rate of 3e-5 and a total of 2 epochs. While this configuration was better than the original configuration, the improvement was only marginal. The best performance observed was unsurprisingly from the bert-base-uncased which is as expected given that it's the largest model which conceivably utilizes its 12 layers to extract more semantic meaning from the text than the models with 8, 4, or only 2 layers.