Artificial Intelligence II

Nefeli Tavoulari

Fall Semester 2021

1 Vaccine Sentiment Classification

In the first notebook I used a pretrained text classification Bert model, to classify tweets as Neutral, Pro-vax or Anti-vax.

1.1 Preprocessing

First of all, I performed some preprocessing on the data, using the BertTokenizer, in order to get word embeddings. In particular, I used the bert-base-uncased model, which does not make difference between capital and lowercase letters. In this way, I got the token ids of each tweet and also the information which indicates whether it is a word or padding.

1.2 Fine-tuning

Then, using the BertForSequenceClassification model, I experimented with different configurations, in order to find the ones giving the best results.

So, using the following configurations:

- no cleaning in the data
- max length = 100 in tokenization
- batch size = 32
- gradient clipping max norm = 1
- epochs = 3

The precision/recall/f1 scores taken using a different learning rate in Adam optimizer were the following:

lr	*score*
$1e-3$	0.46699
$1e-4$	0.74308
1e-5	**0.76292**
$1e-6$	0.68361

So, using the following configurations:

- no cleaning in the data
- max length = 100 in tokenization
- batch size = 32
- epochs = 3
- learning rate = 1e-5

The precision/recall/f1 scores taken for different gradient clipping max norm values were the following:

clip	*score*
1	0.76292
2	0.76336
3	**0.77081**

The results are pretty close, considering that there is also some randomness. So, using the following configurations:

- no cleaning in the data
- max length = 100 in tokenization
- gradient clipping max norm = 3
- epochs = 3
- learning rate = 1e-5

The precision/recall/f1 scores taken for different batch sizes were the following:

batch_size	*score*
16	**0.77519**
32	0.77081
64	0.75241

It seems that our model needs a small batch size and a relatively small learning rate, which is expected from a Bert pretrained model. So, using the following configurations:

- no cleaning in the data
- batch size = 16

- gradient clipping max norm = 3
- epochs = 3
- learning rate = 1e-5

The precision/recall/f1 scores taken for different max lengths in tokenization were the following:

max_length	*score*
50	0.76511
100	**0.77519**
200	0.76421

In the first case, probably useful information is cut out, while in the third one, not very useful information is passed to the model. I also tried to set the max tweet length as max length but it was too large, it took too much time to train and since the 200 size did not work, that would not work either. So, using the following configurations:

- batch size = 16
- gradient clipping max norm = 3
- epochs = 3
- learning rate = 1e-5
- max length = 100 in tokenization

The precision/recall/f1 scores taken using or not using data cleaning were the following:

cleaning	*score*
yes	0.76906
no	**0.77519**

The cleaning of the data includes special character, emojis, foreign language and url removal and converting capital to lowercase letters. Even though, the models trained in the previous assignments had tremendous result improvements after this cleaning procedure, the Bert model does not seem to need this before getting the word embeddings, since the results in both situations are very similar. Last but not least, using the following configurations:

- no cleaning in the data
- batch size = 16
- gradient clipping max norm = 3
- learning rate = 1e-5
- max length = 100 in tokenization

The precision/recall/f1 scores taken using different numbers of epochs were the following:

epochs	*score*
2	0.76906
3	0.77519
5	0.75185

Therefore, the final configurations of my best model are the following:

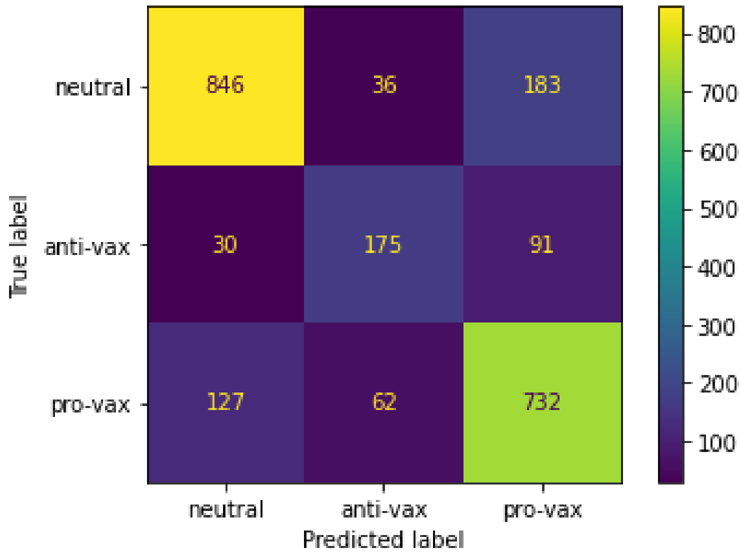
- no cleaning in the data
- batch size = 16
- epochs = 3
- gradient clipping max norm = 3
- learning rate = 1e-5
- max length = 100 in tokenization

1.3 Evaluation - Comparison

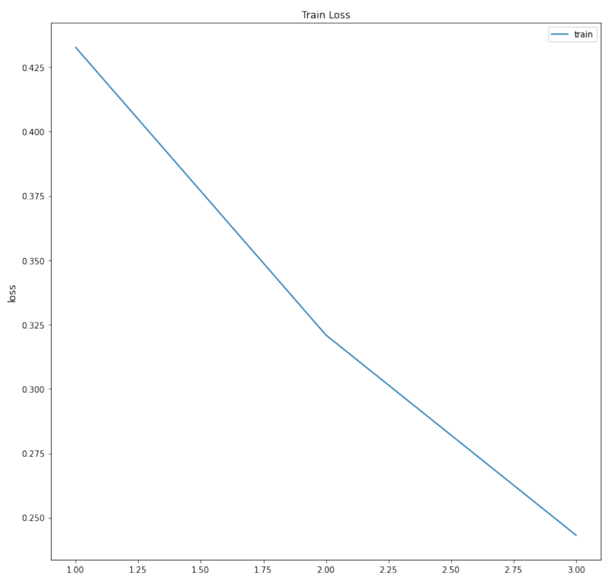
The results taken for all classes:

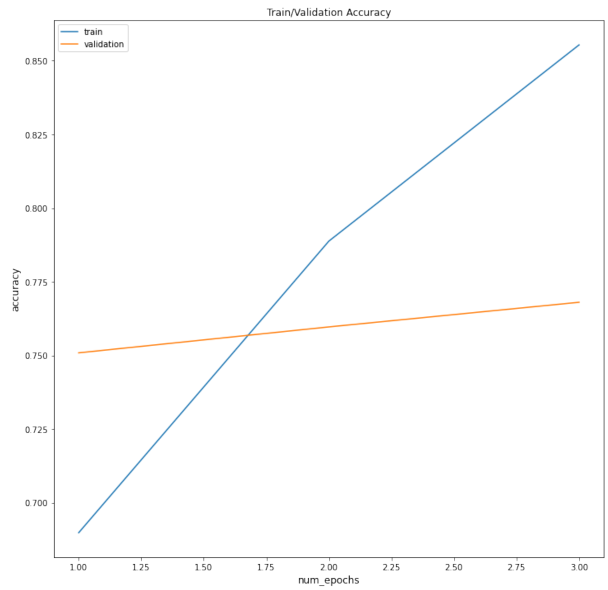
	precision	recall	f1-score	support
0	0.84	0.79	0.82	1065
1	0.64	0.59	0.62	296
2	0.73	0.79	0.76	921
accuracy			0.77	2282
macro avg	0.74	0.73	0.73	2282
weighted avg	0.77	0.77	0.77	2282

From the confusion matrix, I understand that my model makes good enough predictions, since the diagonal has quite bigger numbers even for the anti-vaxers who are very few in the data, in comparison to the other classes.
The confusion matrix ,using the aforementioned model :



Also, the loss and accuracy curves:





We observe that after some point, the model does not learn that much, the accuracy descreases, on the validation set. In general, however, the behavior of the curves is the expected one.That means that the accuracy increases and the loss decreases as the model learns.

Finally, I want to note that the results are better than all the models I trained in the previous assignments(logistic regression,feedforward,RNN), which proves in a way the significance of the transformers attention mechanism.
The results out of the 4 models are the following:

	accuracy	*recall*	*f1*	*precision*
feedforward	0.6520	0.6520	0.6520	0.6520
LSTM	0.7050	0.7050	0.7050	0.7050
logistic regression	0.7230	0.7230	0.7230	0.7230
BertForSequenceClassification	0.77519	0.77519	0.77519	0.77519

The outcome is, of course, very satisfactory.

References

[1] PyTorch Documentation

[2] BERT Documentation

[3] BERT Tokenizer