Yassine Filali
yassine.filali@ensta-paris.fr

1 Question 1

The role of the square mask is to mask the future positions in the sequence because the self-attention layers in nn.TransformerEncoder are only allowed to attend the earlier positions in the sequence. therefore, we setting the masked values to -inf before the softmax step in the self-attention calculation.
For the positional encoding, it is used to get some information about the position of the token in the sequence. It has the same dimension as the embedding vector so that they can be added together.

2 Question 2

We have to replace the classification head because the task has changed, the model was first trained to generate a sentence, therefore, has learned some fixed parameters for this task. So we can't use this model with these same parameters in a order to do a different task which is sentiment analysis. The main difference between the language modeling and the classification task is that language modeling is used to generate texts based on the probability of occurrence of the word and generates a result for every token in the sequence, while classification only uses the last token to generate a result about the sequence.

3 Question 3

For the embedding layer, we have $ntoken * nhid$ parameter, where $ntoken$ is the size of the vocabulary and $nhid$ the embedding dimension, which is equal to $50001 * 200 = 10000200$ parameter for the embedding layer. for each transformer encoder block, the number of parameters is as follows: - for the input to the multihead-attention, we have $3 * nhid * nhid$ weight parameter and $3 * nhid$ bias parameter. - for the output, we have $nhid * nhid$ weight parameter and $nhid$ bias parameter. - for the feed forward layer, we have $nhid * nhid$ for the weights parameter and $nhid$ bias parameter. - for the norm layers, we have $nhid$ weight parameter and $nhid$ bias parameter. The total number of parameter is therefore equal to 242000 parameter. for all the transformer heads we have 968000 parameter. The post task blocks therefore have 10968200 parameter. For the language modeling task head, we have $nhid * ntokens$ parameter, which is equal to $200 * 50001 = 10000200$ weight parameter and 50001 bias parameter. For the classification, we have $nhid * nclasses$ parameter equal to 400 weight parameter and $nclasses = 2$ bias parameter. Therefore for the language modeling task, we have a total of 21018401 parameter, and for the classification task, we have 10968602 parameter.

4 Question 4

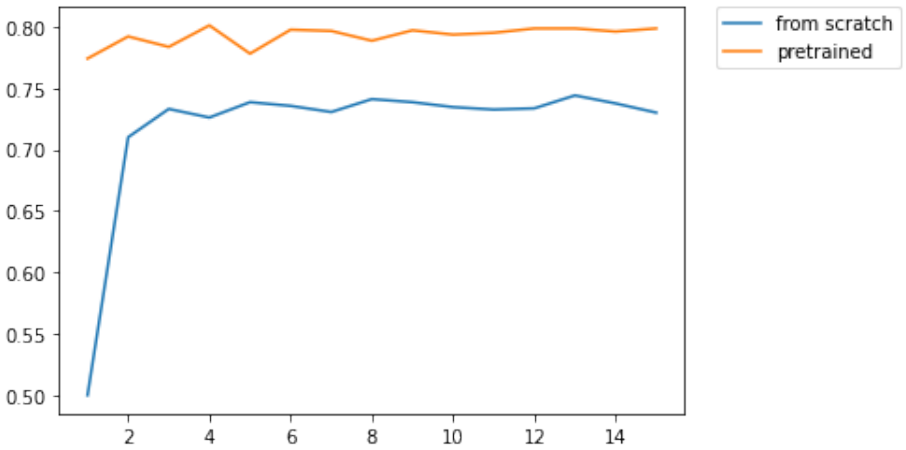

Figure 1: Accuracy: pretrained model vs from scratch model

We can note that the pretrained model leads to a better accuracy than the model trained from scratch. We can also note that the from scratch model needs a few epochs to reach its maximum accuracy. Therefore, the transfer learning approach yields to better results in terms of accuracy, and training time (Since it is only a tuning of the pretrained model).

5 Question 5

One of the limitations of the language modeling objective used in this lab is that it only takes the past context (the left side of the sequence) by using a square mask, as it can lead to some inaccurate translation, as it is sometimes necessary for some languages (french for example). To correct that, we could use a bidirectional architecture (like BERT for example).