Simon Queric
simon.queric@telecom-paris.fr

Question 1

The squared mask forces the self-attention mechanism to be causal. It means that the ith ouput depends only on i first inputs. Positional encoding is a vector representation of the position of a word in the source sentence.

Question 2

We replace the classification head so that it fits with our wanted task. The classification head is a linear layer, so we have to specify the number of classes wanted. In the language modelling task, the model guesses the next token of a group of tokens, so that it performs a classification task with n_{tokens} classes. In the classification task, there are fewer classes. Here for instance, there are two classes : postive and negative book reviews.

Question 3

I took the notations of *Attention Is All You Need* [2]. Without the biases, we have $n_{\text{tokens}} \times d_{\text{model}}$ parameters for the embedding layer. Then, in each attention head, the model learns $3 \times N$ matrices W^Q, W^K, W^V and one matrix W^O which gives us $3 \times d_{\text{model}} \times d_k \times N \times h + h \times d_k \times d_{\text{model}} = 3 \times N \times d_{\text{model}}^2 + d_{\text{model}}^2$ parameters for the attention layers and

$$\begin{cases} d_{\text{model}} \times n_{\text{tokens}} & \text{parameters for the decoding layer in the case of language modelling task.} \\ d_{\text{model}} \times n_{\text{classes}} & \text{parameters for the decoding layer in the case of classification task.} \end{cases}$$

Finally, the number of parameters of our model is :

$$\begin{cases} n_{\text{tokens}} \times d_{\text{model}} + d_{\text{model}} \times n_{\text{tokens}} \times d_{\text{model}} + n_{\text{tokens}} + 3 \times N \times d_{\text{model}}^2 + d_{\text{model}}^2 + d_{\text{model}} \times n_{\text{tokens}} & \text{for language modelling.} \\ d_{\text{model}} \times n_{\text{classes}} + 3 \times N \times d_{\text{model}}^2 + d_{\text{model}}^2 + d_{\text{model}} \times n_{\text{classes}} & \text{for classification.} \end{cases}$$

Concretely, in the case of language modelling task, our model learns roughly 20×10^6 parameters and 10×10^6 parameters in the case of the classification task.

Question 4

We can plot the evolution of accuracy with the pretrained model and without the pretrained model :

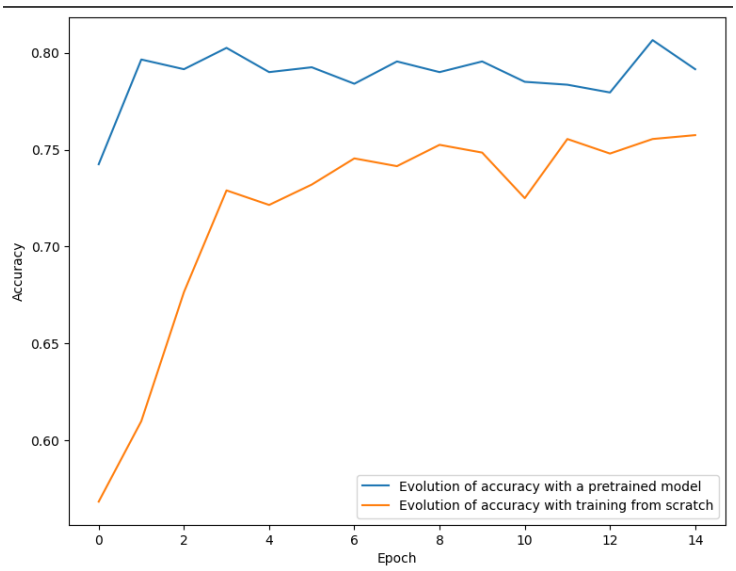

Figure 1: Evolution of accuracy

We can see that the pretrained model requires less epochs to be efficient than the model trained from scratch. Indeed, it reaches an accuracy of 80% after 3 epochs while the model trained from scratch reaches 75% after 10 epochs. The pretrained model starts also with a quite good accuracy of 75% while the model trained from scratch start with a terrible accuracy of 50% (just random guess, which is logical).

Question 5

Our language modelling obective is to predict the word that follows the beginning of a sentence. In the paper [1], 15% of words are masked and one of the language modelling objective is to predict these words. Another objective of BERT is next sentence prediction.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.