MERRHEIM Maïssane
maissane.merrheim@telecom-sudparis.eu

Lab session # 3
ALTEGRAD 2023

10/31/23

1 Question 1

The role of the square mask implementation :

In our implementation, we use a mask in the Transformer model, to ensure that the model doesn't cheat by looking ahead. It is used to compute the self-attention scores. For a better understanding, here is an example : Let's consider the sentence : ' I see white flowers'. During the training, when the model will want to predict the third word of this sentence 'white', we will mask the words 'white' and the word 'flowers', so that the model doesn't 'cheat' by looking ahead.

The role of the positional encoding :

In a sentence, the order of the words is essential. In fact, without taking into account the order of the words in a sentence, the signification can change drastically. Therefore, in the architecture of a transformer, we need to take into account the position of the words in a sentence. To do so, we need positional encoding. Unlike RNNs, which inherently capture the word order in their input sequences, the Transformer relies on multi-head attention, where attention scores are computed simultaneously for each word in the input. To incorporate this information, we use positional encoding.

2 Question 2

Language modeling and Text classification have different purposes. For the language modeling module, the purpose is to predict the next tokens from a prompt(in input) and to generate words, whereas the text classification module takes in input a sentence and predict whether this sentence is positive or negative, in the case of sentiment analysis. In transfer learning, we take a pretrained model, that was not trained on the same task or on the same data, and then we use it to learn another task, or the same task on another data. Here, we use the same type of data (words), but instead of predicted the next tokens and generate sentences, we want to predict whether the sentence in input is positive or negative. However, we don't want to train from scratch another model. Therefore, we use the first layers of the model (the Transformer block), and then we adapt it to another task, which is the sentiment classification analysis.

3 Question 3

To compute the number of parameters the models have for both tasks, text classification and language modeling, we first need to compute the number of parameters for the common block : the base model. The base model is composed of :

- an embedding layer
- a positional encoding layer
- 4 transformers

Let's compute the number of parameters layer per layer in the base model :

- Number of parameters for the embedding layer : $ntokens \times nhid$
- Number of parameters for a transformer encoder :
  - Number of parameters for the Multi-Head Attention layers : $Param_{MHAL} = 4 \times (nhid^2 + nhid$
  - Number of parameters for the 2 Normalization layers : $Param_{LN} = 2 \times (nhid + nhid) = 4 \times nhid$
  - Number of parameters for the FFNN : $Param_{FFNN} = 2 \times (nhid^2 + nhid)$

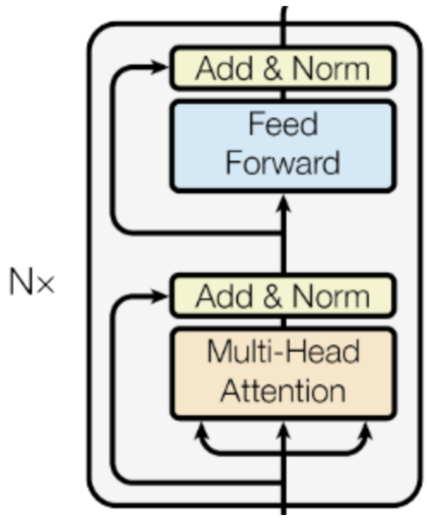

Figure 1: Architecture of en encoder in a Transformer [1]

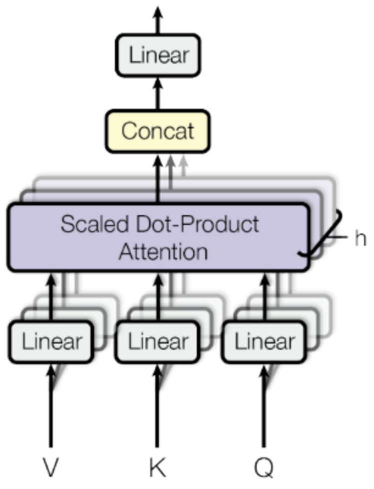

Figure 2: Architecture of a Multi-Head Attention Layer[1]

In conclusion, the number of parameters in total for a transformer blocks :

$$\begin{aligned} Param_{transfomer} &= Param_{LN} + Param_{MHAL} + Param_{FFNN} \\ &= (4 \times nhid) + (4 \times (nhid^2 + nhid) + (2 \times nhid^2 + nhid) \\ &= 6 \times nhid^2 + 10 \times nhid) \\ &= 6 \times 200^2 + 10 \times 200 \\ &= 242000 \end{aligned} \tag{1}$$

For the language modeling task, the number of parameters for the classifier (task 1 language modeling) is equal to $Param_{task2} = nhid \times nclasses + nclasses = 200 \times 100 + 100 = 20100$.
What's more $Param_{Embedding} = (ntokens \times nhid) = 2000$.
Therefore, the total number of parameters for the model, for the classification task is equal to :

$$\begin{aligned} Param_{classification} &= Param_{Embedding} + 4 \times Param_{transformer} + Param_{task2} \\ &= (100 \times 200) + 4 \times 242000) + (200 \times 100 + 100) \\ &= 1008100 \end{aligned} \tag{2}$$

For the classifier task :

The number of parameters for a classifier (task 2 finetuning) is equal to $Param_{task1} = nhid \times nclasses = 200 \times 2 = 400$. The total number of parameters for the model, for the language modeling task is equal to :

$$\begin{aligned} Param_{language_modeling} &= Param_{Embedding} + 4 \times Param_{transformer} + Param_{task1} \\ &= (100 \times 200) + 4 \times 242000 + (400) \\ &= 988400 \end{aligned} \tag{3}$$

4 Question 3

According to the 3rd figure, the accuracy is higher when the model doesn't learn from scratch specially for the first epochs. Therefore, it is more interesting to adapt a model and use a pretrained model, than build a model from scratch.

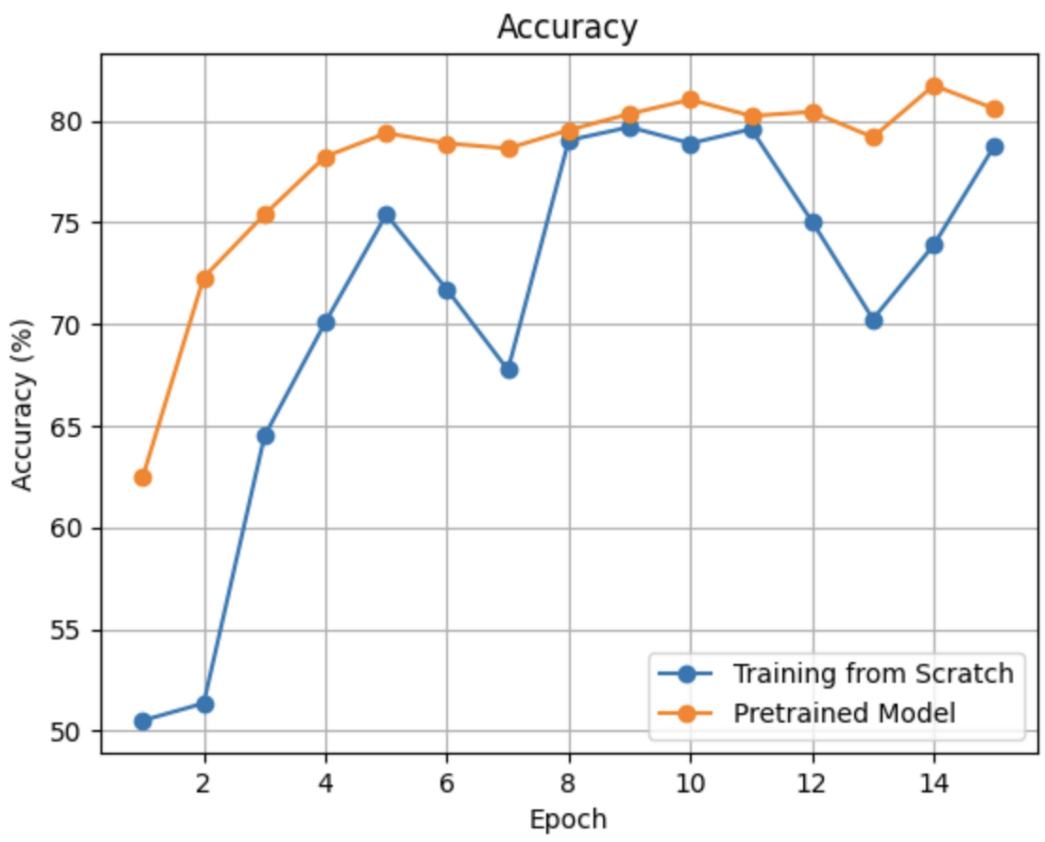

Figure 3: Accuracy of the models over 16 epochs

5 Question 4

In our notebook, we only masked the words that become after the words in the input Therefore, the Transformer is one directional. However, in Jacob Devlin and al paper[2], the authors study another way tot train the model which allows to have better performances. In fact, unlike our language modeling training, the authors decide to train the model so that it has to guess the masked word, no matter where it is in the sentence. If we take our previous example of ”I love white flowers”, in the model used by the authors called MLM, it can mask the word ”white” only, and with the sentence ” I love ¡masked word¿ flowers” as input, predict the word ”white”.

Source : ” Unlike left-to- right language model pretraining, the MLM objective enables the representation to fuse the left and the right context, which allows us to pretrain a deep bidirectional Transformer.”[2]

Therefore, our model is limited in a way in cannot predict masked words,if those words do not immediately follow the input sentence.

References

[1] NikiParmar-JakobUszkoreit LlionJones AidanNGomez Łukasz Kaiser AshishVaswani, NoamShazeer and Illia Polosukhin. Attention is all you need. In *In Advances in neural information processing systems,*, page 5998–6008, 2017.

[2] Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *arXiv preprint arXiv 1810.04805*, 2018.