NAIM Mouna
mouna.naim@ens-paris-saclay.fr

Lab session 2
ALTEGRAD 2024

10/21/24

1 Question 1

In the transformer implementation, the **square mask** is important as it ensures that each position in the sequence can only attend to previous positions or itself. The transformer aims to predict the next token based on the tokens seen so far, so it is essential that it does not have access to future tokens. However, when we construct the dataset, this constraint isn't respected by default. The role of the square mask is to prevent the attention mechanism from focusing on future tokens by setting their weights to -infinity, effectively blocking them from being attended to and ensuring predictions are made only based on past and present information.

The **positional encoding** is used to provide the transformer with information about the relative and absolute positions of tokens within the sequence. Positional encodings are added to the input embeddings to indicate each token's position, enabling the transformer to capture the order of the tokens in the sequence.

2 Question 2

Here, we focus on two tasks. In the first task, the transformer predicts the next word (language prediction), and in the second task, the transformer classifies a sentence as positive or negative. The transformer is first trained on the language prediction task so that it can understand the language before moving to the classification task. Therefore, the classification head helps us adapt the model to each of these tasks.

3 Question 3

Let's compute the trainable parameters for each task (language modeling and classification) and clarify the notations used in the code:
Language Modeling Task

- Emnedding Layer(nn.Embedding): The numbre of the parameters for the embeddig layer is ntokens $\times$ nhind.
- self attention mechanism of 3 $\times$ nhid $\times$ (nhid + 1). + nhid $\times$ (nhid + 1), for projections.
- Two linear layers: nhid $\times$ (nhid + 1). parameters.
- Two normalization layers: nhid $\times$ (nhid + 1) parameters.

Thus, the modeling language task: ntoken $\times$ nhid + 8 $\times$ nhid $\times$ (nhid + 1).

Classification
The classification has only one linear layer with nhid: nhid $\times$ nclasses

4 Question 4

The plot compares the accuracy progression over epochs between two models: one trained from scratch and another using a pretrained model. The pretrained model (orange line) begins with a higher initial accuracy, around 65%, compared to the from scratch training model (blue line), which starts at a lower point of 55%. This initial advantage is expected, as pretrained models have learned general features from previous tasks, leading to better early performance. Both models show rapid improvement during the first few epochs, with the pretrained model reaching its peak accuracy of approximately 77% as early as epoch 1, while the from scratch model catches up to around 75% by epoch 2.

The pretrained model demonstrates a significant advantage in terms of faster convergence and statibity throughout training. Its accuracy is more consistent, while the scratch model experiences greater instability with more pronounced fluctuations. This suggests that pretrained models are particularly useful when stable and rapid performance improvements are desired, especially in the early stages of training.

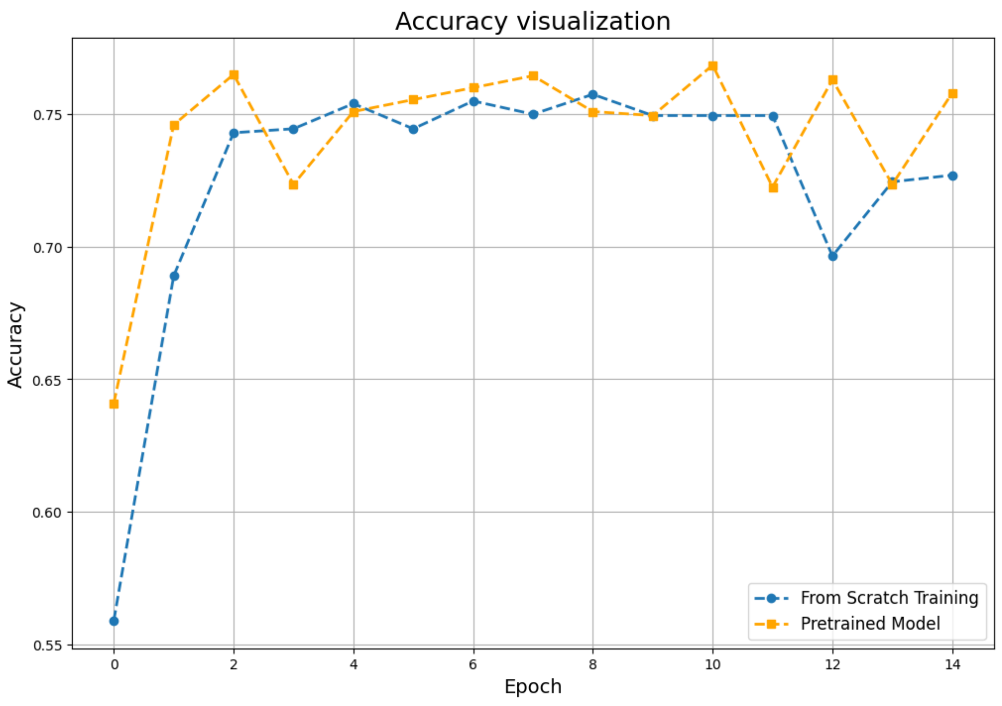

Figure 1: : Accuracy of a pretrained model and a model trained from scratch over the 15 epochs

5 Question 5

In this approach, the model learns only from left to right, and each piece of information can be used only for future understanding. This is due to the unilateral mask. In the paper , the authors propose a new mask (bilateral mask). This mask uses the entire context for prediction, incorporating both the left and right parts of the word to be predicted.