SIMON Hugo
hugo.simon@telecom-paris.fr

Lab session # 3
ALTEGRAD 2021

29/11/2021

1 Question 1

The role of the mask is to prevent transformer to rise attention on future words in language modeling task. This mimics one aspect of unidirectional recurrent neural networks (RNN) previously used though the processing is not Markov nor stationary anymore.

However, unlike RNN, transformers in itself don't leverage any information about the ordering of the words. So a positional encoding is added to input such that distance between any two time-steps is consistent across sentences with different lengths.

2 Question 2

The *language modeling* task consists of predicting the probability distribution of the next word in a sentence so the classification head must have as many outputs as the size of the vocabulary. For the *classification* task we look at the whole sentence and output a binary value saying if it's a positive or negative review, this requires having only two output neurons.

3 Question 3

- The embedding layer is a matrix of size. $n\_token \times n\_hidden = 50,001 \times 200$
- For positional encoding, we have matrices of sines and cosines values with no parameters learned.
- The rest of the encoder consists of $n\_layers = 4$ transformer blocks (see Vaswani et al. [2]) each with
  - A multi-head attention layer with Key, Value, and Query matrices plus a linear layer (all with bias). $3 \times (n\_hidden \times n\_hidden + n\_hidden) + (n\_hidden \times n\_hidden + n\_hidden) = 4 \times (200 \times 200 + 200)$
  - 2 batch normalization layers (means and variances). $2 \times n\_hidden \times 2 = 4 \times 200$
  - 2 feed forward linear layers (with bias). $2 \times (n\_hidden \times n\_hidden + n\_hidden) = 2 \times (200 \times 200 + 200)$
- Finally the classification head which consists of
  - Tokens' size linear layer with bias in the case of *language modeling* task. $n\_token \times n\_hidden + n\_token = 50,001 \times 200 + 50,001$
  - Sentiments' size linear layer with bias in the case of *classification* task. $2 \times n\_hidden + 2 = 2 \times 200 + 2$

Total number of trainable parameters :

- *language modeling* task : $10,000,200 + 968,000 + 10,050,201 = 21,018,401$ parameters
- *classification* task: $10,000,200 + 968,000 + 402 = 10,968,602$ parameters
- *classification* task with pretrained base model: 402 parameters

4 Question 4

In figure 1 we report our validation accuracy for both pretrained model and trained-from-scratch model. The pretrained model easily outperforms it's counterpart. This is especially significant because our training dataset is small (around 3000 examples).

The warm-started model reached it's peak performance after only 3 epochs while the cold-started one takes around 15 epochs to reach it's peak.

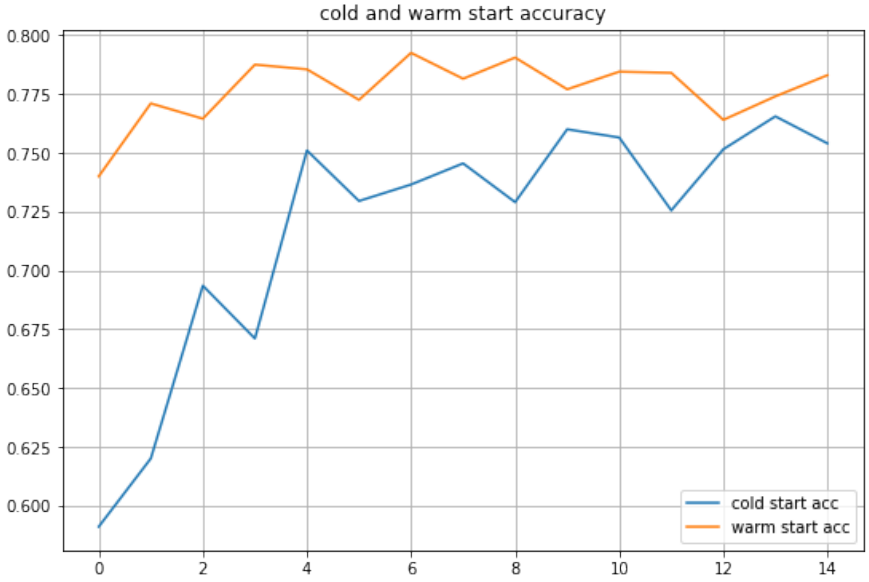

Figure 1: Warm-started and cold-started accuracies among training epochs

5 Question 5

In the previously used transformer network, each token can only attend to previous tokens. Such restrictions are sub-optimal for sentence-level tasks, and could be very harmful when applying fine tuning based approaches to token-level tasks such as question answering, where it is crucial to incorporate context from both directions.

Devlin et al. [1] solve this problem by training a Bidirectional Encoder Representations from Transformer (BERT) on two objectives. The masked word objective where the goal is to predict a randomly masked word from a sentence using the rest of the sentence as context, and the `IsNext` objective where the goal is to predict if two sentences are consecutive or not.

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.