KU Leuven - NLP 2024-2025
22 October 2024

Exercise Session 1 - TA: Nathan Cornille

Language Modeling, Fine-tuning, Self-Attention

For questions contact:
Toledo Forum for Questions (under Discussions) or nlp@ls.kuleuven.be

1 Transformer Language Models

The original Transformer paper is [4].

1.1 Example of self-attention

To illustrate the idea behind the attention mechanism, consider the following two sentences:
"Juventus lost from Ajax because they were too strong"
"Juventus lost from Ajax because they were too weak".
Note that "they" in the first sentence refers to Ajax, while it refers to Juventus in the second sentence. The word "strong" resp. "weak" informs us of this. This toy exercise will illustrate how the attention mechanism can mimic this reasoning.
Assume that the words are encoded by the following vectors:

"$Juventus$"	"$lost$"	"$from$"	"$Ajax$"	"$because$"	"$they$"	"$were$"	"too"	"$strong$"	"$weak$"
$\begin{bmatrix}1\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}-1\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\1\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}-1\\1\end{bmatrix}$	$\begin{bmatrix}1\\1\end{bmatrix}$

Question 1 Now, apply attention twice. Use the following matrices to transform the word encodings into keys, queries and values:

$$W_K = \begin{bmatrix}1 & 0\\0 & 1\\0 & 0\end{bmatrix}$$

$$W_Q = \begin{bmatrix}1 & 0\\0 & 1\\0 & 0\end{bmatrix}$$

$$W_V = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$$

Don't forget to apply softmax to the attention weights before applying them to the values.
First, write down the correct formulas. Second, because this involves quite some numerical calculations even for this toy exercise, and because those are not the point of this exercise, you can also make the calculations in Python. Some template code (in which you still need to complete the attention function) is provided on Toledo. You can run it locally, or e.g., in a google colab file.
Compare the words with high attention values for the query "they" in both sentences: does the difference make sense?

1.2 Multi-head self-attention

Question 2 Make a detailed sketch of the *multihead*-self-attention layer in a Transformer for a sequence of length S. More specifically, if the embedding dimension is E, there are H heads, the key-query-matching dimension of each head is M_{QK}, and the hidden dimension of each head for the values is M_V, sketch both the parameter matrices and the intermediate representations at each step, annotating each with their correct dimension, as well as the non-parametric computations (e.g., slicing, concatenating, dot product). Next to each computation, write down the formula to go from its inputs to its outputs. In addition to the lecture slides, you can check out https://jalammar.github.io/illustrated-transformer/ to understand the self-attention part of the transformer.

1.3 Pretraining, Freezing and Finetuning

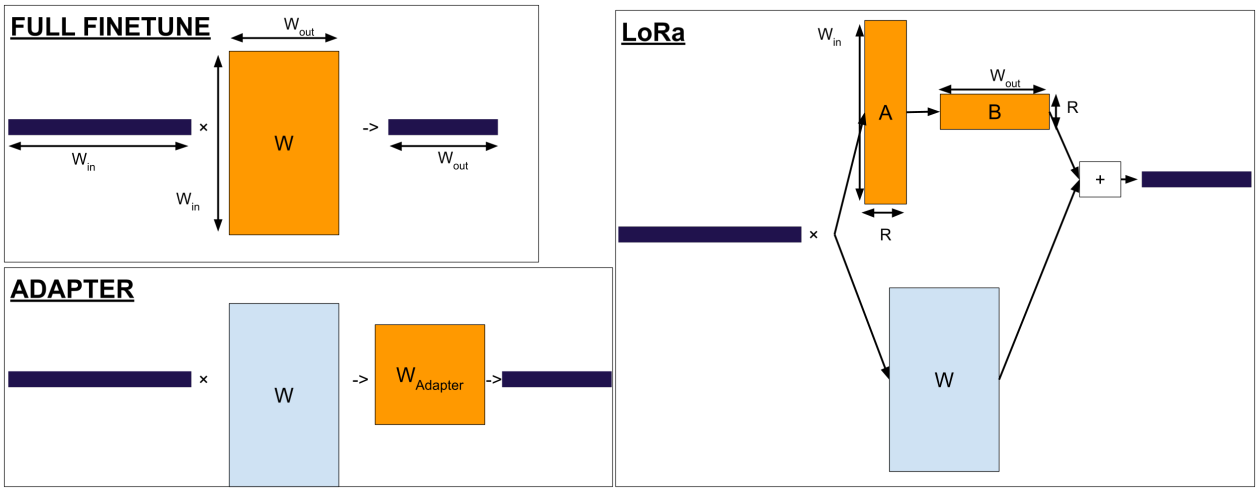
Question 3 Why do we call objectives such as Language Modelling, Masked Language Modelling or Next Sentence Prediction 'self-supervised'? What is the benefit of pretraining with a self-supervised objective?

Question 4 Suppose you want to do sentiment classification on tweets. You let people annotate tweets to create supervised data, but you've used up your annotation budget after about 10 000 labeled tweets. You do however have access to 2 000 000 *unlabeled* tweets. How could you leverage those unlabeled tweets to create a better sentiment classifier?

Question 5 Consider the three finetuning strategies shown in figure 1.3. Discuss which methods are most suited for the following scenarios:

- You want the model to be as fast as possible during inference
- You want to store fine-tuned variants of the model for many different tasks, but only have limited storage space
- You want to finetune the model, but only have limited memory available to store parameters and gradients during training

If $W_{in}, W_{out}, R = 200, 50, 5$, what is the ratio between the number of fine-tuned parameters using LoRa [2] and the number of parameters when doing full fine-tuning.



2 Byte-pair encoding

Question 6 In this exercise, you'll apply byte pair encoding [3][1]. Consider the following toy corpus:

$$abbc\ abb\ abc\ abb$$

We will consider a single character as one 'byte'. Encode it using byte pair encoding, which consists of the following steps:

1. Split the text into a list of words and their frequency
2. Add a symbol to the end of each word that indicates that it is the end of a word (e.g., the symbol '\w')
3. Count the frequency of each byte pair: within each word, and multiplied by the frequency of that word
4. Select the pair with the highest frequency, and replace it with a novel symbol (here for example, we can use capital letters (A,B, ...) as novel symbols).
5. Iterate from step 3 until there is no byte pair with a frequency higher than 1

[1] A good explanation can be found here

3 Cross-modal search

Question 7 Given are a matrix $\boldsymbol{I} \in \mathbb{R}^{2\times4}$ of image representations and a matrix $\boldsymbol{T} \in \mathbb{R}^{3\times2}$ of sentence representations. If we have learned an image projection $\boldsymbol{W}_i \in \mathbb{R}^{4\times3}$ and text projection $\boldsymbol{W}_t \in \mathbb{R}^{2\times3}$ to a three-dimensional multimodal embedding space, then find for each of the images which sentence is the most suitable annotation. Show the computation.

$$\boldsymbol{I} = \begin{bmatrix} 0.2 & -1.5 & 0.6 & 0.3 \\ -0.4 & -0.5 & 1.2 & -0.1 \end{bmatrix} \qquad \boldsymbol{T} = \begin{bmatrix} 0.9 & 0.3 \\ -0.8 & -0.1 \\ 0.3 & -0.5 \end{bmatrix}$$

$$\boldsymbol{W}_i = \begin{bmatrix} 0.6 & 0.3 & -0.2 \\ 1.3 & 0.3 & -0.6 \\ 0.8 & 0.2 & -0.3 \\ 0.4 & 0.4 & 0.7 \end{bmatrix} \qquad \boldsymbol{W}_t = \begin{bmatrix} -1.2 & 0.5 & 0.8 \\ 0.4 & -0.1 & -1.6 \end{bmatrix}$$

Question 8 Say that our goal was not to find the sentence that is the most suitable annotation, but to update the image representations by using cross-attention on the text representations. Which components of the cross-modal search could you reuse for that? And which components would still need to be added?

4 OPTIONAL: LSTM gates

The original LSTM paper is [1]

Question 9 Given the intermediate values given in Figure 1, calculate $\boldsymbol{c}_t$ and $\boldsymbol{h}_t$. Note that you don't need to do the calculations for the blocks in yellow.

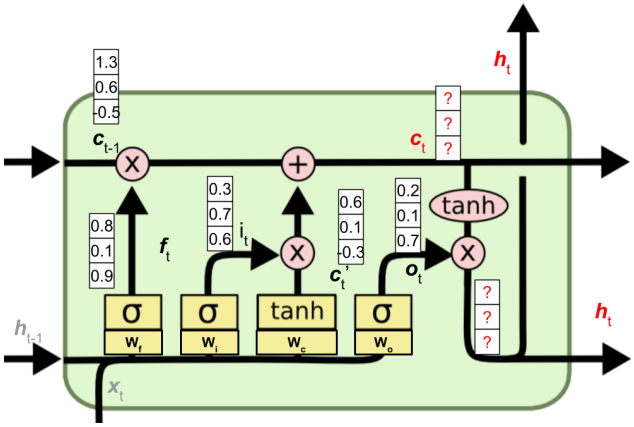


Figure 1: Image based on https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

References

[1] S Hochreiter. "Long Short-term Memory". In: *Neural Computation MIT-Press* (1997).

[2] Edward J Hu et al. "Lora: Low-rank adaptation of large language models". In: *arXiv preprint arXiv:2106.09685* (2021).

[3] Rico Sennrich. "Neural machine translation of rare words with subword units". In: *arXiv preprint arXiv:1508.07909* (2015).

[4] A Vaswani. "Attention is all you need". In: *Advances in Neural Information Processing Systems* (2017).