KU Leuven - NLP 2021-2022
8 November 2021

Exercise Session 2
Language Modelling, Pretraining, Self-Attention

For questions contact:
Toledo Forum for Questions or
nlp@ls.kuleuven.be

1 LSTM gates

Question 1 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.

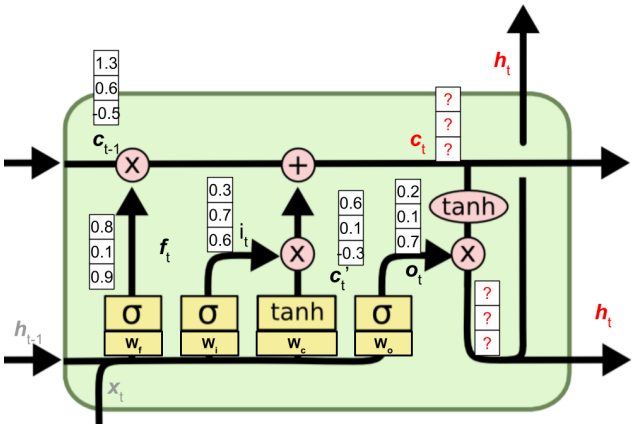


Figure 1: Image based on https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

2 BERT MLM for pretraining

2.1 The attention mechanism

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

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

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

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

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?

2.2 Multi-head self-attention

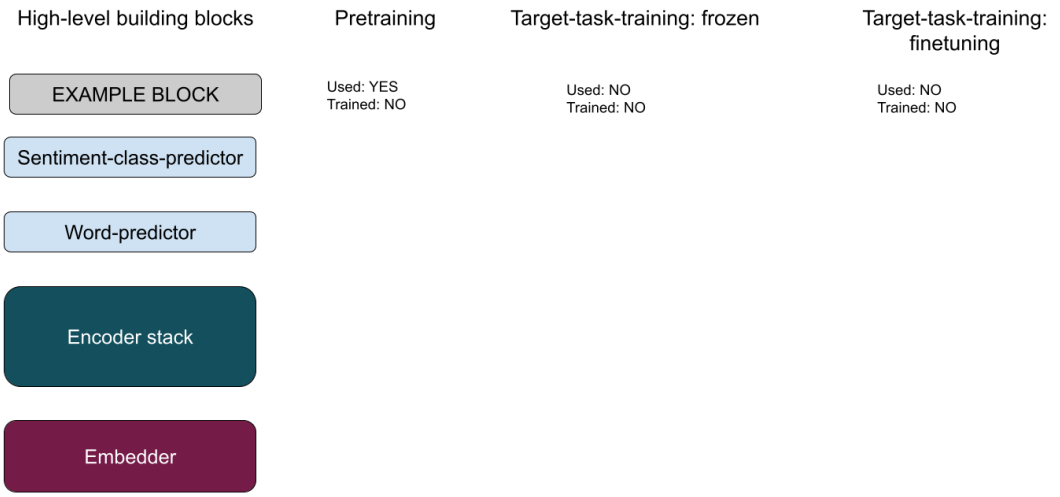
Question 3 Make a detailed sketch of the *multiheaded*-self-attention layer in a Transformer for a sequence of length S. More specifically, if the embedding dimension is E, there are H heads, and the key-query-matching dimension of each head is M, sketch both the parameter matrices and the intermediate representations at 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. Besides the lecture slides, you can check out `https://jalammar.github.io/illustrated-transformer/` to understand the self-attention part of the transformer.

2.3 Pretraining, Freezing and Finetuning

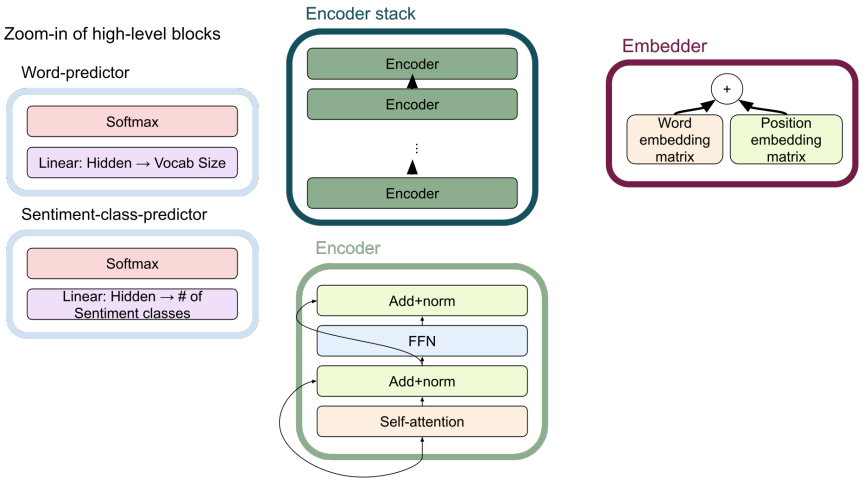
Question 4 Why do we call objectives such as Language Modelling, Masked Language Modelling or Next Sentence Prediction 'self-supervised'? What is the benefit of pretraining with a self-supervised objective?

Question 5 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 6 Given the following high-level building blocks, indicate which are 1) used 2) trained during resp. pretraining, target-task-training with freezing, and target-task-training with finetuning. What can be the benefit of freezing layers (at least for some time) during target-task-training?



Just for reference, this is how the high-level building blocks are composed of lower-level building blocks for BERT pretraining and finetuning:



3 OPTIONAL: Frequentist n-gram language model

3.1 Language Model

Question 7 First write out the equations for estimating probabilities with a trigram model that (1) does not use smoothing (2) uses Laplace smoothing. Then use these equations to estimate the probabilities from the following toy corpus. Ignore casing (e.g., don't distinguish between The and the). Add a special token $< unk >$ to the vocabulary to represent words that are not in the toy corpus.

<s> The room was quiet when the girls played the minuets on the piano.</s>
<s> It was quiet for minutes afterwards. </s>

a. $P(played|the, girls)$

b. $P(sang|the, girls)$

c. $P(the| < s >)$

d. $P(the, room, was, noisy| < s >)$

no smoothing: $P(w_i \mid w_{i-2}, w_{i-1}) = \frac{C(w_{i-2},w_{i-1},w_i)}{C(w_{i-2},w_{i-1})}$

Laplace smoothing : $P(w_i \mid w_{i-2}, w_{i-1}) = \frac{C(w_{i-2},w_{i-1},w_i)+1}{C(w_{i-2},w_{i-1})+V}$

3.2 Spelling correction

Question 8 For correcting real-word spelling errors (i.e., the incorrectly spelled word is an actual word of the language) and for resolving cases where an edit distance yields several valid solutions, we need context-sensitive spelling correction. Given the following sentence, where the word $minutes$ is spelled incorrectly: <s>The girls played the minutes</s>. Design a method and prove that $minutes$ in this sentence better is replaced by $minuets$ using a language model based on the toy corpus of question 1. Use a trigram model, but smooth by interpolating with a bigram and unigram model (see slide 28 of the lecture (from prof. Yin) on Language models). The interpolation weight of a trigram is 0.70, of a bigram is 0.20, of a unigram 0.10.
In what cases is this way of smoothing not sufficient?

4 OPTIONAL: Word embeddings

Question 9 Indicate if the following statements are true or false and explain why.

1. Word embeddings are the weights of a neural network between the input layer and the hidden layer.

2. The goal of the CBOW model is to assign probabilities to word sequences.
3. The goal of the continuous Skip-gram model is to learn vector representations that can be used in NLP tasks.
4. The softmax function converts a vector of real valued numbers to a probability distribution.
5. When you can train your data on a small text corpus, it is a good idea to use embeddings with a large dimension, since training speed is not an issue.
6. Words that never occur in the same sentence won't have similar embeddings.

Question 10 Given the following toy vocabulary $V = \{cat, dog, on, mat, sat, the\}$. Make a sketch of the continuous Skip-gram architecture that trains embeddings of dimension 3. From your sketch it should be clear what the dimensions are in each layer of the network. Next to each layer write down the formula to go from its inputs to its outputs.

Question 11 Given the toy vocabulary of the previous question and the weight matrices below, calculate the probability that dog occurs in the context of mat (that is, mat is the input word). Each word is represented by its index of the vocabulary. For example, cat is the first word in the vocabulary and hence corresponds to the first row of $\boldsymbol{W}$ and the first column of $\boldsymbol{V}$.

$$\boldsymbol{W} = \begin{pmatrix} 1 & -0.8 & 0.25 \\ 0.3 & 1.1 & -0.25 \\ 0.1 & 0.2 & 0.3 \\ 0.5 & 1 & -0.5 \\ 0.75 & 0.5 & 0.25 \\ 0.2 & -0.75 & -1.25 \end{pmatrix}$$

$$\boldsymbol{V} = \begin{pmatrix} 0.5 & 1 & -1 & 1 & 0.1 & 0.75 \\ 2 & 1 & -0.5 & 1.1 & 1 & 1.1 \\ -0.2 & -0.1 & 1 & -1 & 0.5 & 1 \end{pmatrix}$$