Adriano Del Gallo
adriano.delgallo@telecom-paristech.fr

Lab session # 7
ALTEGRAD 2019

01/22/20

1 Question 1

In the classical RNN encoder decoder framework, Q is the source x's hidden state, K is the target y's hidden state. V is the source's hidden state again.

In the self attention framework, Q,K, and V are embedding matrices of the input sequence x.

2 Question 2

For the self-attention layer, we compute the output from the entire sentence (which is in $O(n.d)$) for each word, which gives us $O(n^2.d)$. There is no sequential order for the operations, so we get a complexity of $O(1)$. The maximum path length is $O(1)$ because every operation is done in one go

For recurrent layer, the output y_t in $O(d)$ operations from h_t which we also compute in $O(d)$ from h_{t-1} and x_t. By recursion, for a sequence of length n the total complexity is $O(n.d^2)$. To compute y_t we need to preserve the order of the sequence so we get a complexity of $O(n)$.The maximum path length is $O(n)$ because the information is passed trough n units.

For convolutional layer, the embedded sequence is of size $n.d$ with which we multiply k filters. We have a complexity in $O(k.v.d)$. There is no order to apply these operation, hence, we have a complexity in $O(1)$.The maximum path length is $O(log_k(n))$ because this is the number of windows of size k required to cover a sequence of length n.

3 Question 3

The point of having multiple heads of the same dimension is that we produce different representations for each head, thus uncovering different types of dependencies within the sequences. By concatenating we get a rich representation.

4 Question 4

We set $\omega_{2i} = \frac{1}{10000^{2i/d_{model}}}$, we have

$$\begin{aligned} PE_{(pos+k,2i)} &= \sin(pos\,\omega_{2i})\cos(k\,\omega_{2i}) - \cos(pos\,\omega_{2i})\sin(k\,\omega_{2i}) \\ PE_{(pos+k,2i+1)} &= \cos(pos\,\omega_{2i})\cos(k\,\omega_{2i}) - \sin(pos\,\omega_{2i})\sin(k\,\omega_{2i}) \end{aligned}$$

Which we can write for all k, i

$$\begin{aligned} \begin{bmatrix} PE_{(pos+k,2i)} \\ PE_{(pos+k,2i+1)} \end{bmatrix} &= \begin{bmatrix} \cos(k\,\omega_{2i}) & -\sin(k\,\omega_{2i}) \\ \sin(k\,\omega_{2i}) & \cos(k\,\omega_{2i}) \end{bmatrix} \begin{bmatrix} PE_{(pos,2i)} \\ PE_{(pos,2i+1)} \end{bmatrix} \\ &= M_i^k \begin{bmatrix} PE_{(pos,2i)} \\ PE_{(pos,2i+1)} \end{bmatrix} \end{aligned}$$

By writing,

$$M^k = \begin{bmatrix} M_1^k & & 0 \\ & \ddots & \\ 0 & & M_{d_{model}/2}^k \end{bmatrix}$$

we get that $PE_{pos+k} = M^k PE_{pos}$

For all k, M^k is invertible and position invariant. Thus we get a suitable encoding because every position vector is unique while also giving us a consistent way to account for the relative positions between words, independently of the input length.

5 Question 5

In a test situation, we do not have access to the future words, the only available information are the relationships between the current and past words and past words with one another. During training, however, since we are in teacher forcing mode, the model has access to all the words in the target sentence. To replicate the test situation, we must only use the available knowledge contained in the lower triangular part of the attention matrix QK^T. The 'illegal' zone for which we are not allowed to use information is the upper triangular part.

Our goal is to thus to assign an attention score of 0 to the illegal zone in order to force the attention mechanism to ignore these words. By setting the upper diagonal components of the matrix QK^T to $-\infty$ we ensure that the Softmax Normalization will assign probability 0.

Indeed Since there is at least one non-zero element per line i, we get for $a_{ij} = -\infty$, $\text{Softmax} = \frac{e^{a_{ij}}}{\sum_{j'}^{T} e^{a_{ij'}}} = 0$ which is the desired outcome.

6 Question 6

In the original model from the paper there are 65 million parameters. Considering that the authors trained the model for several days on high end GPUs, we cannot possibly imagine to train this model properly on our own machines.

7 Question 7

8 Question 8

BERT model tries to improve on other models by taking into account bidirectional context to represent sentences, instead of simply the natural flow of the sentence. The entire sentence (sequence of word embeddings) will be fed as input, and then represented using the self attention mechanism in the multi-attention layers. This means that the sentence will be encoded as all the relationships between the words it contains, independently of their position in the sentence (this is why it is said 'bi-directional'). Ultimately, the BERT representation of a sentence captures the relationships between every word and its context (i.e. every other word) to produce an embedding.

We know that the (non-masked) multihead attention layers enable us to do just this. By computing a softmax score between identical keys and queries matrices, we are, for every word, computing its relevance with all the words in the context, regardless of word order, before applying these scores the the sentence. Therefore we can uncover complex dependencies between words within the sentence. Thus, I believe that leveraging multihead attention is an efficient way of represent the meaning and the internal structures of a sentence.