Лекция 6. Transformers. GPT. BERT. T5.

Transformers

① Self-Attention

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{H'}}\right) V \in \mathbb{R}^{T \times H}$$, T – длина посл-ти

$Q, K \in \mathbb{R}^{T \times \tilde{H}}$, $V \in \mathbb{R}^{T \times H}$

$Self\text{-}Attention = Attention(E \cdot W_Q, E W_K, E W_V)$,

$W_Q, W_K \in \mathbb{R}^{H \times \tilde{H}}$, $W_V \in \mathbb{R}^{H \times H}$ – обучаемые матрицы

② Multihead Self-Attention

$$head_i = Attention(\underset{H \times \hat{H}}{Q W_i^Q}, \underset{H \times \hat{H}}{K W_i^K}, \underset{H \times \hat{H}}{V \cdot W_i^V})$$,

$$MultiHead\,Att(Q, K, V) = concat\underbrace{[head_1, \dots, head_n]}_{T \times n\hat{H}} \cdot \underset{n\hat{H} \times H}{W^O} \in \mathbb{R}^{T \times H}$$ – изначальная размерность

На практике:

1) Нам не хочется заводить 3 матрицы и умножать их отдельно на $E \in \mathbb{R}^{T \times H}$

   ∃ внутренняя раз-ть h, $Q, K, V \in \mathbb{R}^{H \times h}$

   $W \in \mathbb{R}^{H \times 3h} \Rightarrow [QKV] = E \cdot W \in \mathbb{R}^{T \times 3h}$

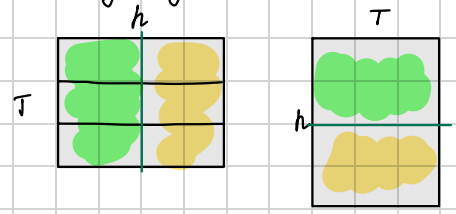
2) Теперь разделим их

   $Q, K, V = split([QKV])$

   $\Rightarrow Q, K, V \in \mathbb{R}^{T \times h}$

3) Теперь нужно разделить матрицы между головами, но опять будет не оптимально умножать.

   Подходит $A = Q \times K^T$?



Тогда очевидно не подходит.

reshape $Q, K, V: T \times h \to n\_heads \times T \times h'$

$A = \frac{Q K^T}{\sqrt{h'}} \in n\_heads \times T \times T$

$A \leftarrow A + M$

4) $A = Softmax(A)$

Также часто на этом этапе применяется Dropout

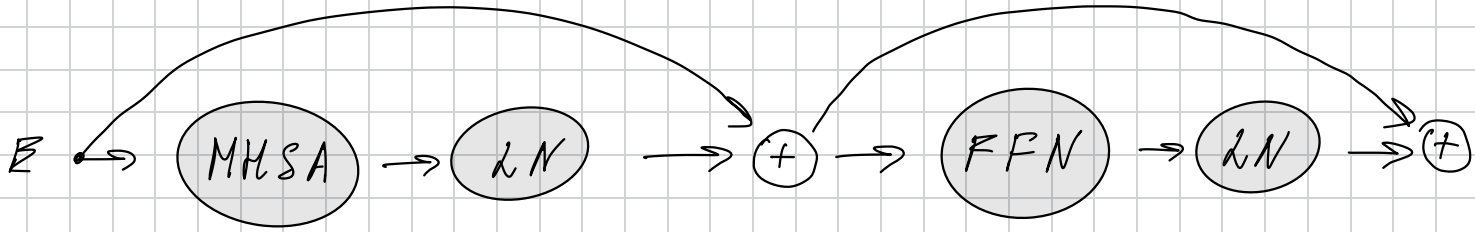
5) $y = A \cdot V \in n\_heads \times T \times h'$

reshape $n\_heads \times T \times h' \to T \times n\_heds \times h' \to T \times h$

6) $y = y \cdot W \to T \times H$

Тут тоже обычно стоит Dropout.

3) Transformer block



* Post / Pre Normalization

4) Position Encoding

$PE_{pos} \in \mathbb{R}^d$, pos - позиция в посл-ти

$PE(pos, 2i) = \sin\left(\frac{pos}{f_i}\right)$ $\quad f_i = 10000^{\frac{2i}{d}}$

$PE(pos, 2i+1) = \cos\left(\frac{pos}{f_i}\right)$ $\quad i = \overline{1, \frac{d}{2}}$

Такие эмбеддинги обладают очень хорошим свойством, что pos + k эмбеддинг является линейной функцией от pos эмбеддинга:

$PE(pos+k, 2i) = \sin\left(\frac{pos+k}{f_i}\right) = \sin\left(\frac{pos}{f_i}\right)\cos\left(\frac{k}{f_i}\right) + \cos\left(\frac{pos}{f_i}\right)\sin\left(\frac{k}{f_i}\right) =$

$= Linear\left[PE(pos, 2i), PE(pos, 2i+1)\right]$

Transfer Learning

Рассказ про GPT, BERT, ERNIE, XLNET, Roberta