BERT-Base Transformer Forward Pass

Ana Marasović

Initialize:

$W_T \in \mathbb{R}^{\text{vocab size} \times d} = \mathbb{R}^{\text{vocab size} \times 768}$ ... token embeddings
$W_P \in \mathbb{R}^{\text{max input length} \times d} = \mathbb{R}^{512 \times 768}$ ... positional embeddings
$h \in \{1, \ldots, n_{\text{heads}}\}, n_{\text{heads}} = 12$
$l \in \{1, \ldots, n_{\text{layers}}\}, n_{\text{layers}} = 12$
$W^Q_{h,l} \in \mathbb{R}^{d \times d_q} = \mathbb{R}^{768 \times 64}$ ... query *weight* matrices
$W^K_{h,l} \in \mathbb{R}^{d \times d_k} = \mathbb{R}^{768 \times 64}$ ... key *weight* matrices
$W^V_{h,l} \in \mathbb{R}^{d \times d_q} = \mathbb{R}^{768 \times 64}$ ... value *weight* matrices
$W^{ffnn}_l \in \mathbb{R}^{d \times d_{ffnn}} = \mathbb{R}^{768 \times 3072}$ ... feedforward layer's weight matrix
$b^{ffnn}_l \in \mathbb{R}^{1 \times d_{ffnn}} = \mathbb{R}^{1 \times 3072}$ ... feedforward layer's bias vector
$W^{out}_l \in \mathbb{R}^{d_{ffnn} \times d} = \mathbb{R}^{3072 \times 768}$ ... output layer's weight matrix
$b^{out}_l \in \mathbb{R}^{1 \times d} = \mathbb{R}^{1 \times 768}$ ... output layer's bias vector
$W^{final} \in \mathbb{R}^{d \times d} = \mathbb{R}^{768 \times 768}$ ... final layer's weight matrix
$I = (i_1, \ldots, i_{512}) \in \mathbb{N}_0^{1 \times \text{max input length}} = \mathbb{N}_0^{1 \times 512}$ ... input vocab indices
$T = \texttt{lookup}(W_T, I) \in \mathbb{R}^{\text{max input length} \times d} = \mathbb{R}^{512 \times 768}$ ... input token embeddings
$X = T + W_P \in \mathbb{R}^{\text{max input length} \times d} = \mathbb{R}^{512 \times 768}$ ... input embeddings
$Z_0 = X$

Forward algorithm:

for ($l = 1;\ l \leq n_{layers} = 12;\ l++$) {

  for ($h = 1;\ h \leq n_{heads} = 12;\ h++$) {

    $Q_{h,l} = Z_{l-1} W^Q_{h,l} \in \mathbb{R}^{\text{max input len} \times d_q} = \mathbb{R}^{512 \times 64}$ ... query matrix

    $K_{h,l} = Z_{l-1} W^K_{h,l} \in \mathbb{R}^{\text{max input len} \times d_k} = \mathbb{R}^{512 \times 64}$ ... key matrix

    $V_{h,l} = Z_{l-1} W^V_{h,l} \in \mathbb{R}^{\text{max input len} \times d_v} = \mathbb{R}^{512 \times 64}$ ... value matrix

    $A_{h,l} = \texttt{Softmax}(\frac{Q_{h,l} K^T_{h,l}}{\sqrt{d_k}}) \in \mathbb{R}^{\text{max input len} \times \text{max input len}} = \mathbb{R}^{512 \times 512}$

    $Z_{h,l} = A_{h,l} V_{h,l} \in \mathbb{R}^{\text{max input len} \times d_v} = \mathbb{R}^{512 \times 64}$

  }

  $\tilde{Z}_l = \texttt{concat}(Z_{1,l}, \ldots, Z_{n_{\text{heads}},l}) \in \mathbb{R}^{\text{max input len} \times (d_v \cdot n_{\text{heads}})} = \mathbb{R}^{512 \times (64 \cdot 12)} = \mathbb{R}^{512 \times 768}$

  $\bar{Z}_l = \texttt{LayerNorm}(Z_{l-1} + \tilde{Z}_l) \in \mathbb{R}^{512 \times 768}$

  $Z^{ffnn}_l = \max(0, \bar{Z}_l W^{ffnn}_l + b^{ffnn}_l) \in \mathbb{R}^{\text{max input len} \times d_{ffnn}} = \mathbb{R}^{512 \times 3072}$

  $Z^{out}_l = Z^{ffnn}_l W^{out}_l + b^{out}_l \in \mathbb{R}^{\text{max input len} \times d} = \mathbb{R}^{512 \times 768}$

  $Z_l = \texttt{LayerNorm}(\bar{Z}_l + Z^{out}_l) \in \mathbb{R}^{512 \times 768}$

}

Pass $\tanh(W^{final} Z_{n_{\text{layers}}}[0, :])$ to the final `Softmax` that predicts the class, where $Z_{n_{\text{layers}}}[0, :]$ is the hidden state corresponding to the first token.