Assignment Nº 4: Transformer with PyTorch

Daniel Jader Pellattiero
고려대학교
Seoul, Republic of Korea
danieljaderpellattiero@outlook.com

Abstract

The goal of this assignment is to develop a Transformer to perform classification on a dataset of IMDb reviews, consisting of both 15k training samples and 10k for validation and testing.

1. Implementation

1.1. Positional encoding

To supply the model with sequential-order information that pure token embeddings lack, we add a deterministic sinusoidal signal to each representation.

For position $p \in \{0, \ldots, L-1\}$ and embedding coordinate $k \in \{0, \ldots, d_{\text{model}} - 1\}$,

$$\text{PE}_{p,k} = \begin{cases} \sin\bigl(p\, 10000^{-k/d_{\text{model}}}\bigr), & k \text{ even}, \\ \cos\bigl(p\, 10000^{-(k-1)/d_{\text{model}}}\bigr), & k \text{ odd}. \end{cases}$$

These sinusoids give every position a unique phase vector while ensuring that any fixed shift in p can be expressed as a linear function of the encodings. In the forward pass we add the $\text{PE}_{p,k}$ to the learned token embeddings, followed by dropout, leaving dimensionality unchanged.

1.2. Multi Head Attention

Scaled dot-product attention projects the query Q, key K and value V tensors into h parallel subspaces of dimension $d_k = d_{\text{model}}/h$. Each head computes:

$$\text{Attention}(Q, K, V) = \text{softmax}\Bigl(\frac{QK^\top}{\sqrt{d_k}}\Bigr)V,$$

optionally masking illegal positions before softmax. The h outputs are concatenated and mapped back to the model dimension by a learned matrix W_O. Splitting attention this way lets the network attend to multiple relational patterns simultaneously without increasing asymptotic complexity beyond $\mathcal{O}(L^2 d_{\text{model}})$.

1.3. Encoder

A single encoder block alternates self-attention and position-wise feed-forward sub-layers, each wrapped in residual connections and layer normalisation. Denoting the input by x, we have:

$$\begin{aligned} x' &= x + \text{Drop}\bigl(\text{MultiHead}(x)\bigr), & \tilde{x} &= \text{LayerNorm}(x') \\ y' &= \tilde{x} + \text{Drop}\bigl(W_2\, \sigma(W_1 \tilde{x})\bigr), & y &= \text{LayerNorm}(y') \end{aligned}$$

where W_1 and W_2 form a two-layer MLP of hidden size d_{ff} and σ is either ReLU or GELU. This architecture enables global context mixing followed by non-linear feature transformation, while residual paths stabilise gradients in deep stacks.

The full encoder is a stack of N identical layers that share the structure above but not parameters:

$$x^{(0)} = \text{Embed}(w) + \text{PE},$$

$$x^{(l+1)} = \text{EncoderLayer}^{(l)}\bigl(x^{(l)}\bigr),\ l = 0, \ldots, N-1.$$

An optional final layer normalisation yields $x^{(N)}$, a contextualised sequence representation suitable for classification or other tasks. Depth N and the feed-forward width d_{ff} control model capacity, while the number of heads h governs how many relational patterns can be modelled in parallel.

2. Experimental results

The implemented model, characterized by a 4-level encoder, 12 attention heads and a dimensionality of 768, showed an accuracy value of 62.38% on the test set after one epoch of training with the Adam optimizer (learning rate of $5e-5$) and Cross Entropy as a loss function.

Using the same training modes, the BERT model (pre-trained) demonstrated to have a higher accuracy of 82.66%.