Attention is All You Need

Attention mechanism allows modeling of dependencies without regard to their distance in the input or output sequences. Transformer relays entirely on the attention mechanism to draw global dependencies between input and output.

1. Archtecture

Encoder-decoder structure, encoder maps an input sequence of symbol representations $(x_1, x_2, ..., x_n)$ to a sequence of continuous representations $z = (z_1, z_2, ..., z_n)$. Given z, the decoder generates an output sequence $y = (y_1, ..., y_m)$ of symbols *one element at a time*. At each step model *auto-regressively* consumes the previously generated symbols as additional input.

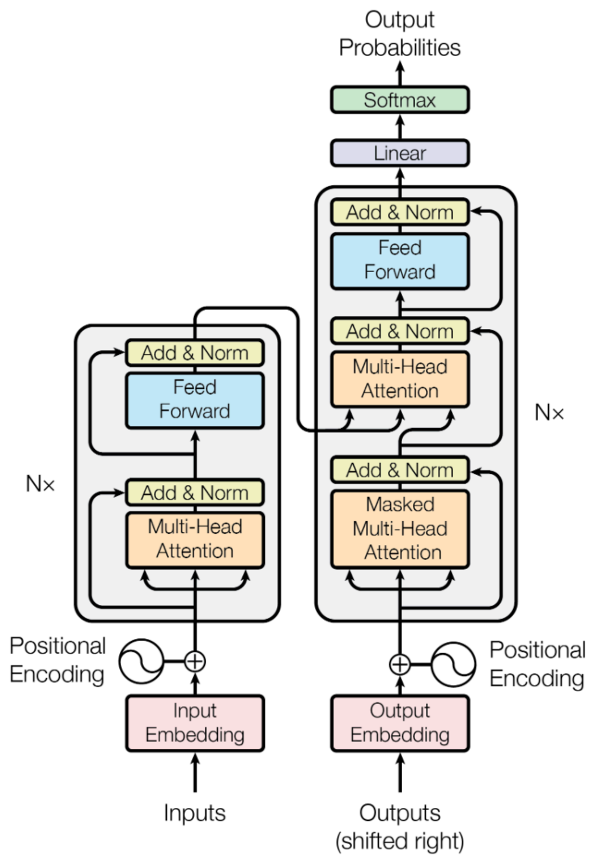


Figure 1: The Transformer - model architecture.

1.1. Encoder

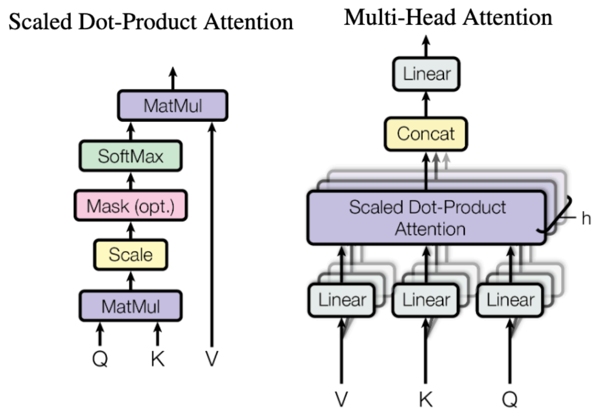
The encoder is composed of a stack of $N = 6$ identical layers, each with two sublayers. First a multi-head self-attention mechanism, second a simple position-wise fully connected feed-forward network. Residual connections are employed arounf these sublayers. All sublayers produce the same size representations so that these residual connections work easily.

1.2. Decoder

Similarly employs $N = 6$ identical layers, but has a third sublayer that performs multi-head attention over the output of the encoder. In the decoder, positions subsequent to the last input do not get attended, they are masked.

1.3. Attention

Attention function maps a query and a set of key-value pairs to an output, where each query, key and value are vectors. Output is the weighted sum of the values, weights are assigned by a compatiility functions of the query with the corresponding key.



1.3.1. Scaled Dot-Product Attention

The input consists of queries and keys of dimension d_k and values of dimension d_v. Dot products of query is computed for all keys and each result is divided by $\sqrt{d_k}$, softmax is applied to obtain the weights of values.

In the implementation, it makes sense to gather all queries, keys and values into corresponding matrices Q, K, V. This way attention operation can be shown as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Without scaling, dot product attention is outperformed by additive attention. But dot-ptoduct attention can utilize matrix multiplications and is much faster.

1.3.2. Multi-head Attention

Instead of having a single attention operation, queries, keys and values are projected to h different vectors. Attention is performed in parallel over these. projections. All outputs from these operations are then concatanated and projected once again.

Multi-head attentio allows the model to jointly attend to information from different representation subspaces at different positions.

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \\ \text{where head}_i &= \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^K\right) \end{aligned}$$

$$W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k} W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k} W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$$

1.3.3. How is it used?

- In encoder-decoder attention, queries come from the previous decoder layer, keys and values come from the encoder. This way decoder can attend to all input values.
- In encoder attention, self-attention layers are used. In a self-attention layer all of the keys, values and queries come from the same place, output of the previous encoder layer.
- In decoder attention, self-attention layers allow each position in the decoder to attend to all positions in the decoder up to and including that position. Masking is done via setting all values of in the input of the softmax to $-\infty$.

1.4. Posiiton-wise Feed-forward Networks

MLP is applied to each position separetely and identically.

1.5. Embedding and Softmax

Learned embeddings are used to convert the input tokens and output tokens to vectors of dimension d_{model}.

1.6. Positional Encoding

To let the model know the order of the sequence positional encodings are injected into the input embeddings. These embeddings have the same dimensions as the model so that they can sum up easily. Sine and cosines of different frequencies can be used for these positional encoding vectors.

2. Why Self-attention?

- All positions are connected by a constant number of operations.
- To connect all positions CNNs require depth, increasing the length between to positions.