Attention is all you Need

💡 We propose a new simple network architecture, the **Transformer**, based solely on attention mechanisms

Introduction

- Recurrent models (RNN) are mainly used for language modeling & machine translation
    - + Typically factor computation along the symbol positions of the input and output sequences
    - - fundamental constraint of sequential computation

→ Transformer, a model architecture **eschewing recurrence** and instead **relying entirely on an attention mechanism** to draw global dependencies between input and output.

Background

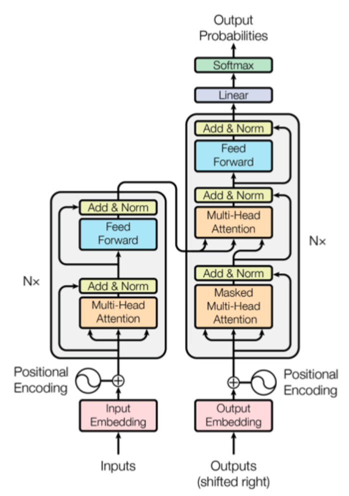
- CNN
    - - difficult to learn dependencies between distant positions
- Self-attention
    - attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence

Novel Idea proposed

Transformer is the first transduction model **relying entirely on self-attention** to compute representations of its input and output without using sequence- aligned RNNs or convolution.

Model Architecture

- Encoder- Decoder Architecture



Attention

Mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors

- **Scaled-Dot-Product Attention**

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

→ Why Scaled? because to counteract the effect; where large values of d_k performs worse when compared to additive attention

- **MultiHead Attention**

$$\begin{aligned} \mathrm{MultiHead}(Q, K, V) &= \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O \\ \text{where } \mathrm{head_i} &= \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned}$$

    - used in encoder-decoder attention layers, inside encoders, and decoders

Position-wise Feed-Forward Networks

- Each of the layers in encoder and decoder contains a fully connected feed-forward network, which is applied to each position separately and identically.
- **ReLU** as activation

Positional Encoding

Inject some information about the relative or absolute position of the tokens in the sequence.

- Add "positional encodings" to the input embeddings at the bottoms of the encoder and decoder stacks
- use sine and cosine functions of different frequencies

Why Self-Attention

1. Total Computational Complexity per layer
2. Amount of computation that can be parallelized
3. Path length between long-range dependencies in the network
4. Yield more interpretable models