Name: Suhaila Ahmed Hassan
Track: AI
Branch: Alexandria

Attention Is All You Need

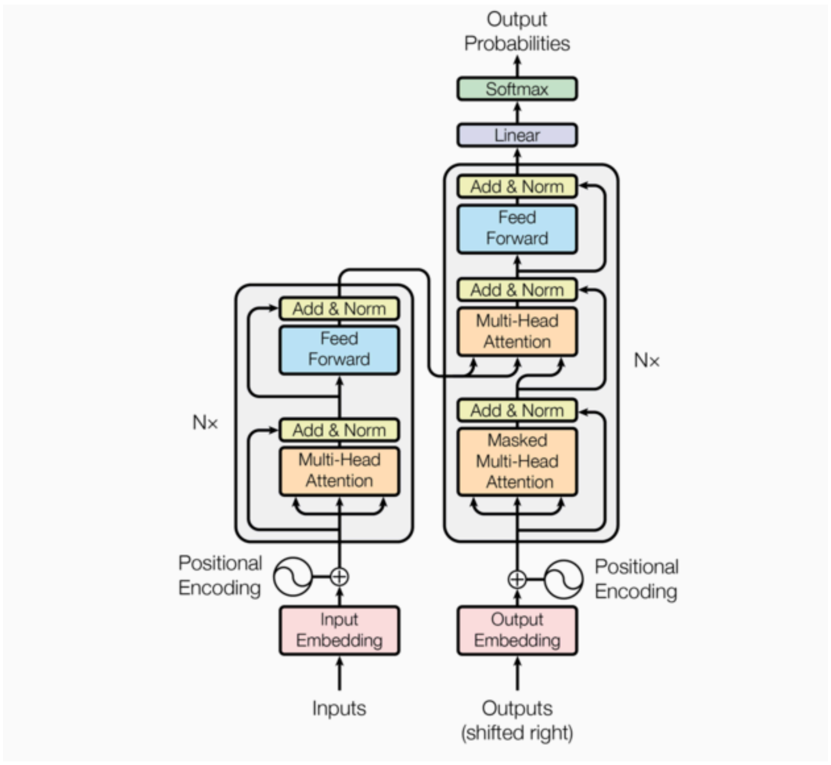
Introduction

This paper introduces the Transformer, a novel architecture that eliminates recurrence and relies solely on attention mechanisms, enabling greater parallelization and faster training, achieving state-of-the-art results.

Model Architecture

The Transformer is based on an encoder-decoder structure where:

- Encoder converts the input sequence into continuous representations.
- Decoder generates the output sequence one token at a time, using encoder outputs and previously generated tokens.
- Both encoder and decoder use stacked self-attention and point-wise feed-forward layers.



1. Encoder and Decoder Stacks

Encoder:

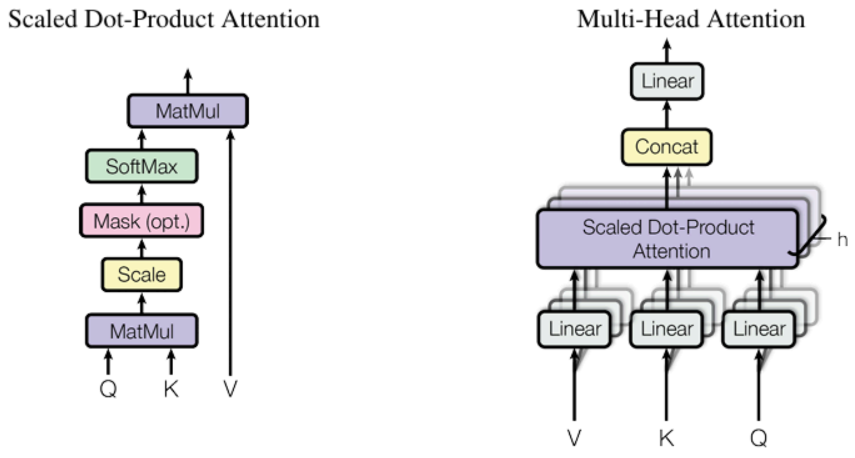
- Composed of 6 identical layers.
- Each layer has two sub-layers:
    A. Multi-head self-attention.
    B. Position-wise fully connected feed-forward network.
- Residual connections and layer normalization are applied to each sub-layer.

- All layers output vectors of dimension d_model = 512.

Decoder:

- Composed of 6 identical layers.
- Each layer has three sub-layers:
  A. Masked multi-head self-attention.
  B. Position-wise fully connected feed-forward network.
  C. Multi-head attention over the output of the encoder stack.
- Residual connections and layer normalization are applied to each sub-layer.
- Masking prevents the decoder from seeing future positions.

2. Attention Mechanism



Attention Function:

- Maps a query and a set of key-value pairs to an output.
- Output is a weighted sum of values based on query-key similarity.

Scaled Dot-Product Attention:

- The input consists of queries, keys of dimension dk, and values of dimension dv
- Computes attention scores using dot product of query and key, scaled by $\sqrt{d}$□.
- Softmax applied to obtain attention weights.

Multi-Head Attention:

- Uses multiple attention "heads" in parallel (h = 8).
- Each head has a lower dimension (d_k = d_v = 64).
- Final output is the concatenation of all heads, followed by linear projection.
- Enables the model to attend to different positions and representation subspaces simultaneously.

Applications of Attention in Transformer:

The Transformer uses multi-head attention in three ways:

- Encoder-Decoder Attention: Queries come from the decoder, and keys/values from the encoder output. This lets each decoder position attend over all encoder positions, mimicking typical encoder-decoder attention in sequence-to-sequence models.

- Encoder Self-Attention: All queries, keys, and values come from the encoder's previous layer. Each encoder position attends to all positions in the previous encoder layer.
- Decoder Self-Attention: Each decoder position attends to all positions up to and including that position. To preserve the auto-regressive property, leftward information flow is blocked by masking out (setting to -inf) illegal connections in the softmax input of scaled dot-product attention.

3. Position-wise Feed-Forward Networks

- Applied independently to each position.
- Consists of two linear transformations with a ReLU activation.
- Input/output dim (d_model) = 512; hidden layer dim (d_ff) = 2048.

4. Embeddings and Softmax

- Learned embeddings convert input/output tokens to vectors of dimension d_model.
- In our model, same weight matrix is shared between input embeddings, output embeddings, pre-softmax linear transformation
- Embedding layers weights are multiplied by the square root of d_model.

5. Positional Encoding

- Injects information about token relative or absolute position since the model has no recurrence or convolution.
- Uses fixed sinusoidal positional encodings:
    A. PE(pos, 2i) = sin(pos / 10000^(2i/d_model))
    B. PE(pos, 2i+1) = cos(pos / 10000^(2i/d_model))
- Positional encodings are added to input embeddings.
- Fixed encodings generalize better to longer sequences than learned ones.

Why Self-Attention

Compared to RNNs and CNNs, self-attention:

- Has lower complexity
- Allows maximum parallelism
- Has shortest path lengths for long-range dependencies
- Is more efficient when sequence length is less than hidden dimension

Model Variation

- Attention heads: Too few or too many heads degrade performance.
- Key/Value dimension: Smaller dimensions hurt quality.
- Model size: Bigger models perform better.
- Dropout: Helps prevent overfitting.
- Positional Encoding: Learned and sinusoidal encoding performed similarly.

Conclusion

Transformer model outperforms previous architectures on standard machine translation benchmarks.
The Transformer model, using only attention mechanisms, produces translation with better quality, has faster training and with lower cost.