Attention and Transformers

Yegor Kuznetsov

Sequence to Sequence

this cat is very large

↓

este gato es muy grande

Sequence to Sequence

this cat is very large

↓

este gato es muy ____

The Problem with RNNs

Whiteboard time!

- Lots of information crammed into a single vector
- Information takes a long path through the system
    - Long range dependencies are hard
    - Vanishing or exploding gradients likely

“Attention Is All You Need”

- Attention is all you need
- Recurrence free
- Enables large models

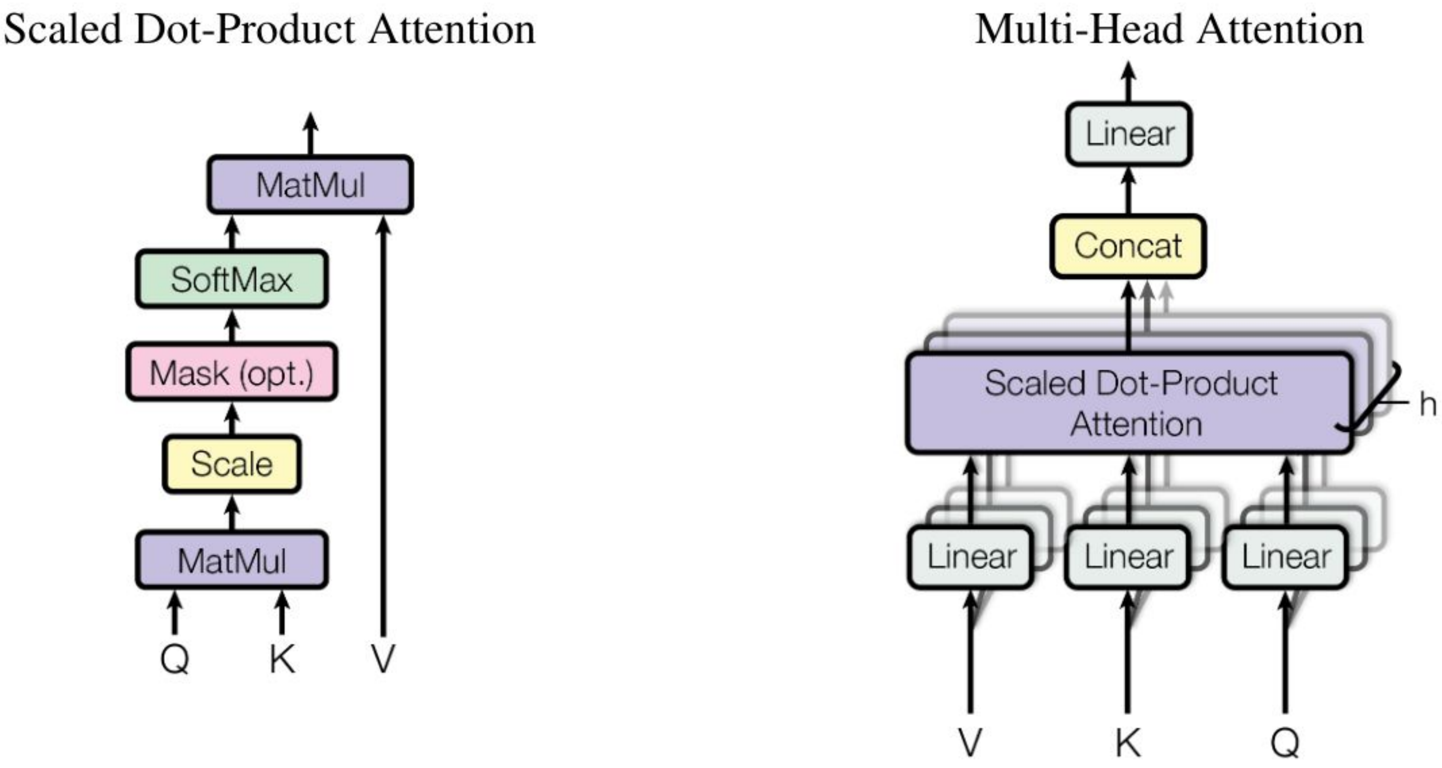


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

Leveraging existing resources

https://courses.cs.washington.edu/courses/cse447/22sp/assets/slides/lec13.pdf#page=29

Attention

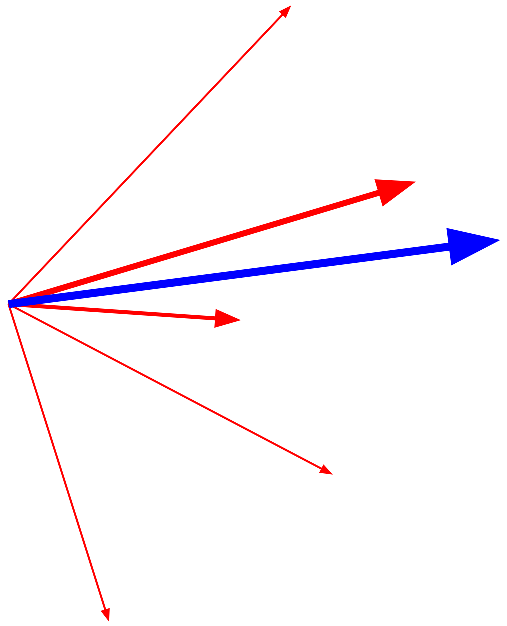
- Pick and choose which word has to do with which other word(s)

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

Query, Key, Value

Whiteboard time!

- Word Embeddings
  - What does each word mean?
- Q,K,V projections
  - Extract and organize information
- Scaled dot product using Q,K
  - Match keys to queries
- Softmax and multiply by V
  - Weighted average of values

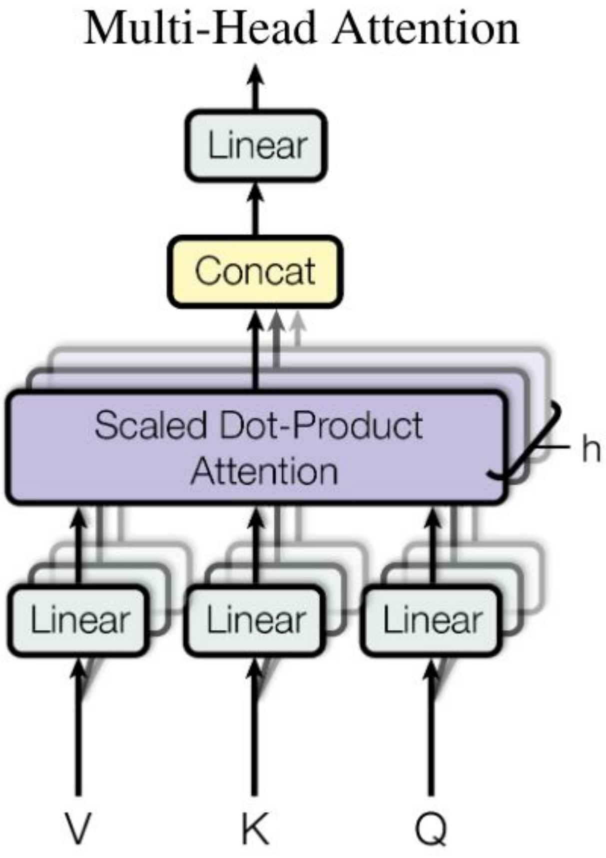


More heads

3.2.2 Multi-Head Attention

Instead of performing a single attention function with d_{model}-dimensional keys, values and queries, we found it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively. On each of these projected versions of queries, keys and values we then perform the attention function in parallel, yielding d_v-dimensional output values. These are concatenated and once again projected, resulting in the final values, as depicted in Figure 2.

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



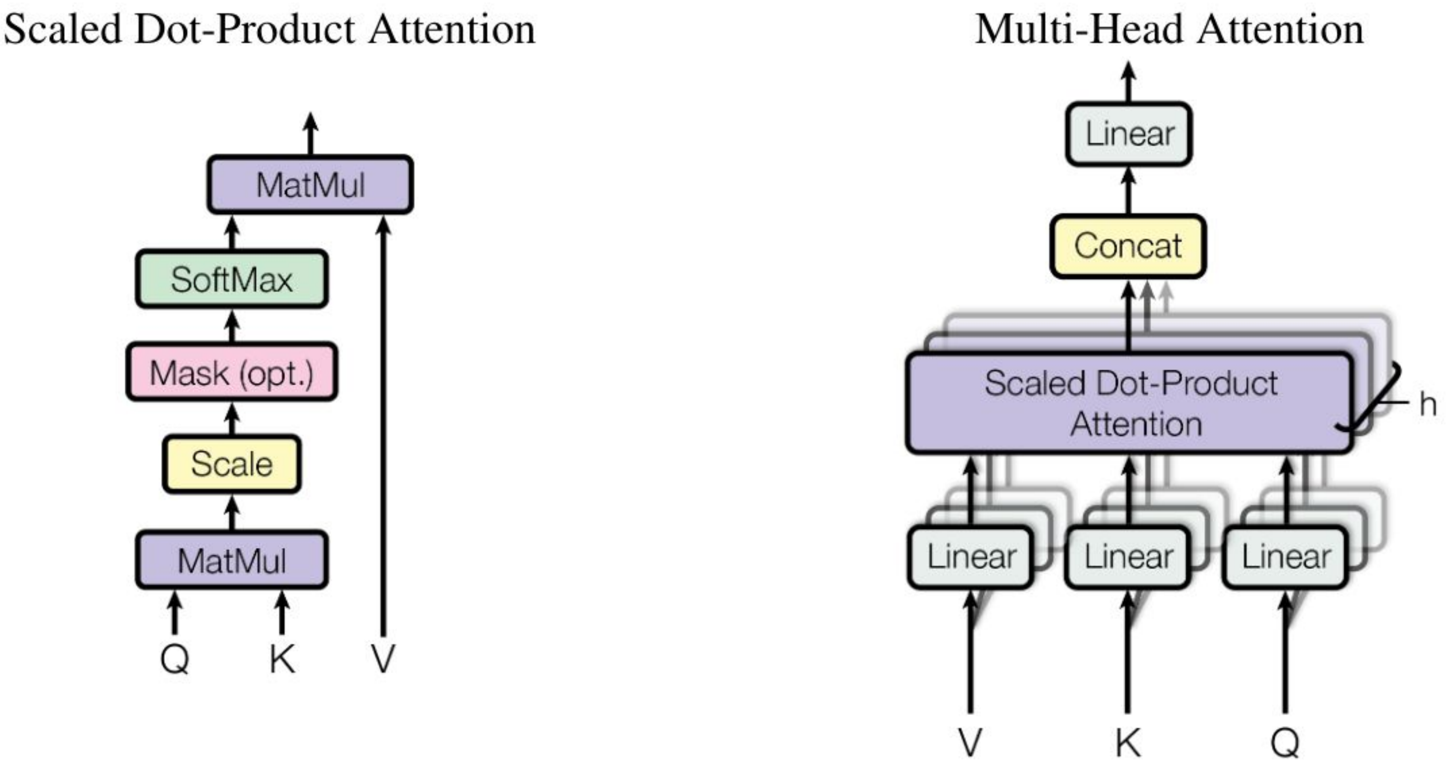


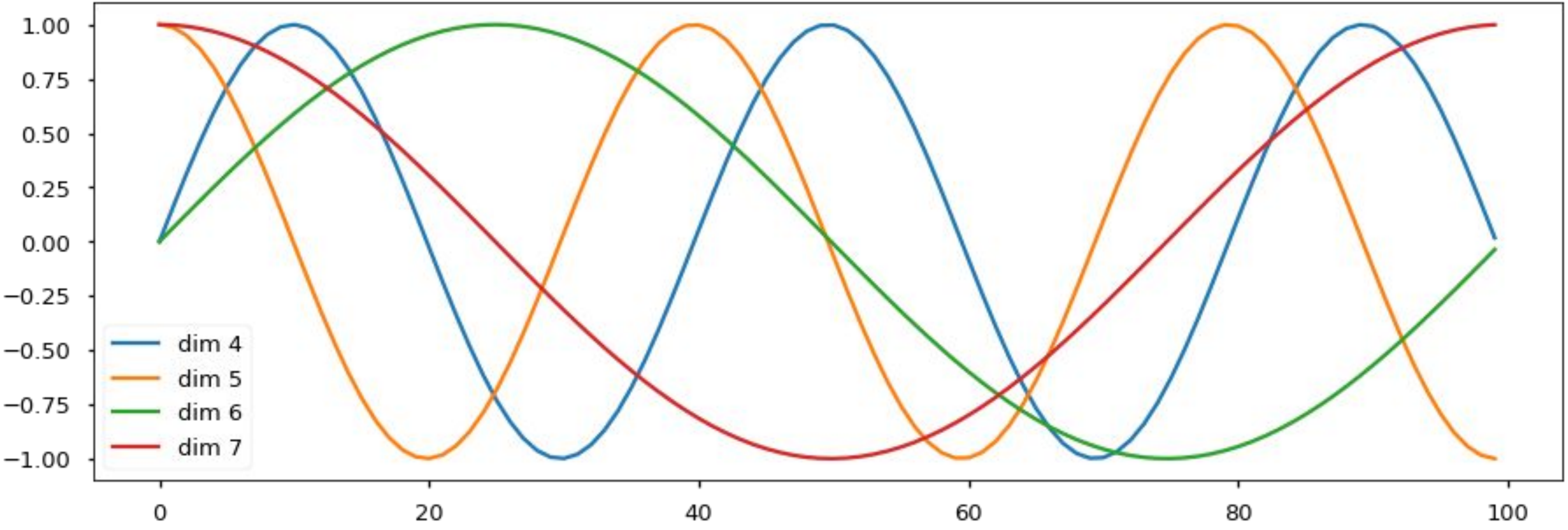
Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

A slight problem

- Weighted average of words → all positional information lost
    - Effectively, we just have a very advanced bag of words
- Fixed by directly providing positional information
    - Arguably a hack, but it does work

Positional Encoding

Whiteboard time!



Click to add tilte

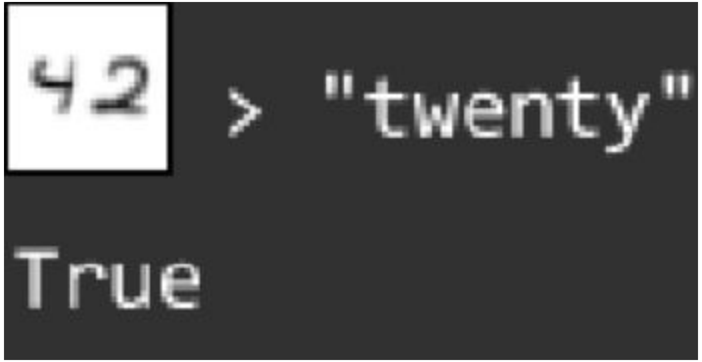
Sorting:

Comparisons.

10 > 4

Cilck to add title

Clikc to add title



Turksort: Sorting with Human Intelligence

http://sigbovik.org/2020/proceedings.pdf#subsection.0.25

RISE: Randomized Input Sampling for Explanation of Black-box Models

http://sigbovik.org/2020/proceedings.pdf#subsection.0.25

A project proposal

- “RISE works on black-box models”
- The natural insight: Explaining human perception via RISE
  - If RISE works on black-box models, we should be able to apply it to Amazon Mechanical Turk

The Transformer

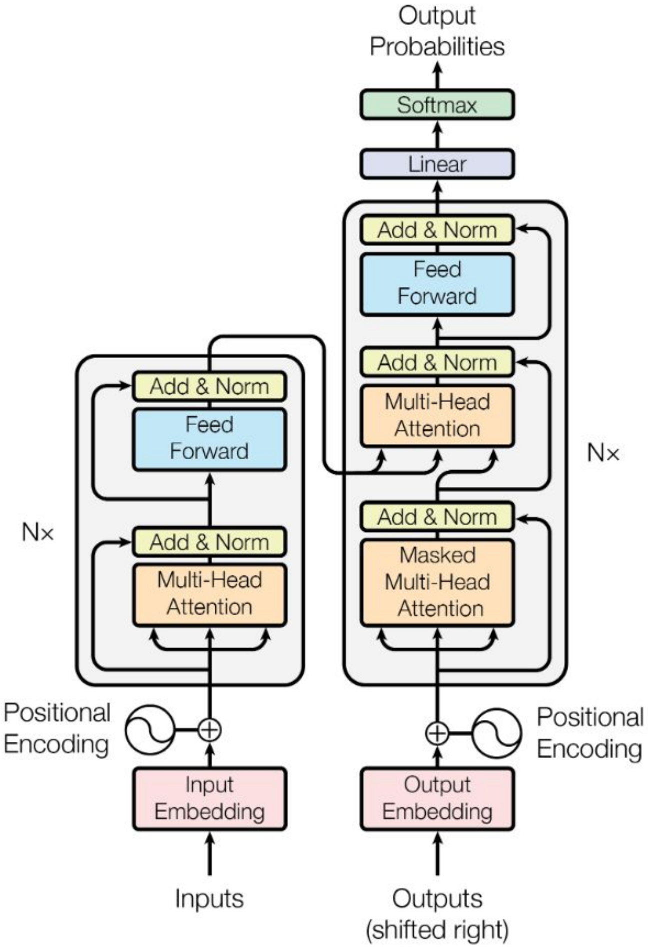
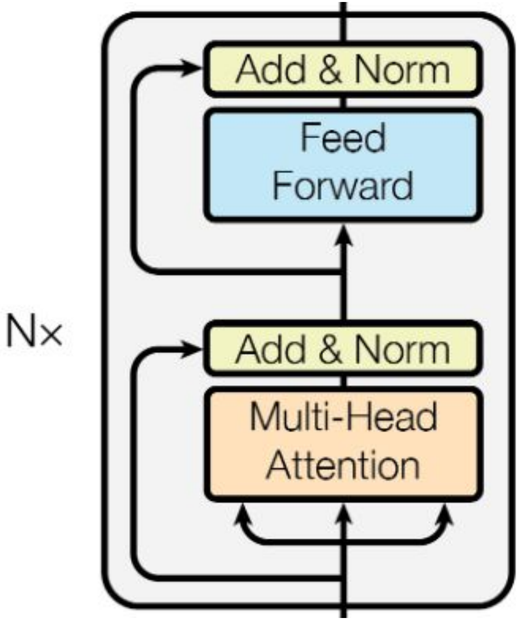


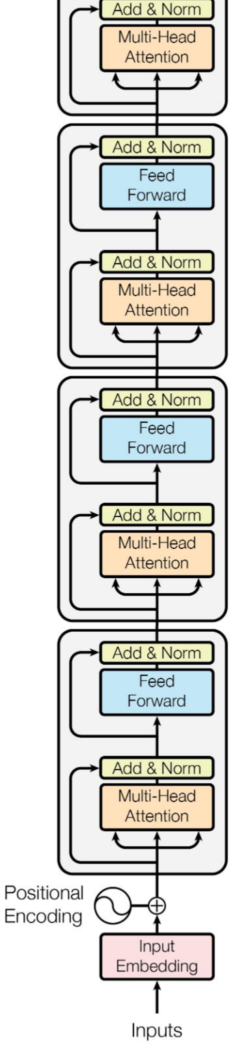
Figure 1: The Transformer - model architecture.

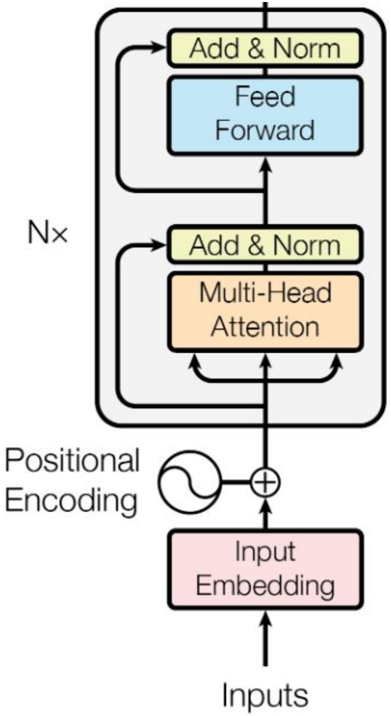
The Transformer – Encoder



The Transformer – Encoder

- Stack many of those attention layers on top of each other





The Transformer

- “Encoder” extracts relevant information, organized into K,V
- “Decoder” constructs relevant queries to ‘search’ that information.

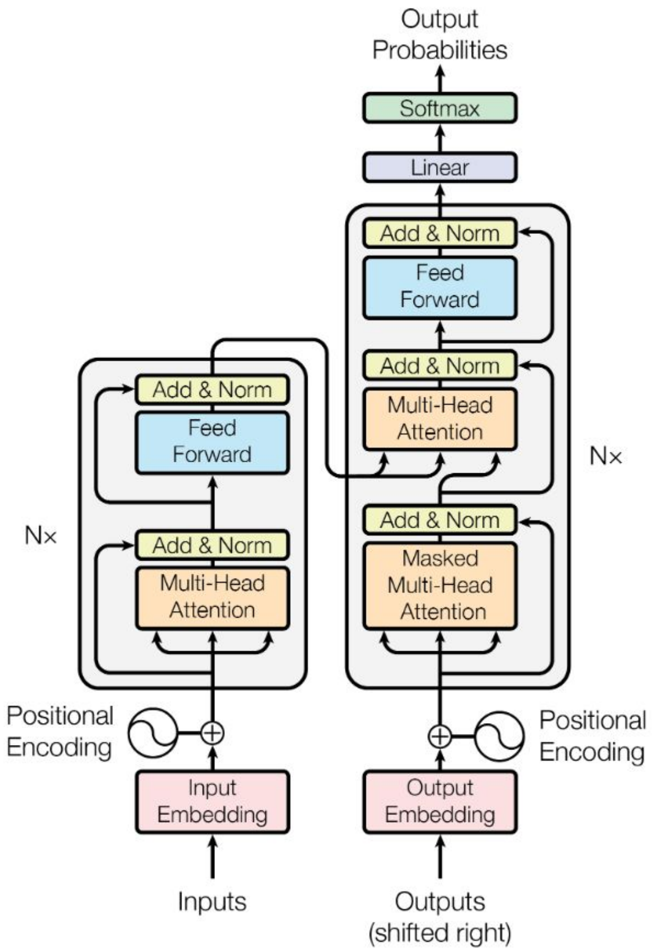


Figure 1: The Transformer - model architecture.

But why is this any good?

- Path length
  - Equal path lengths enable learning long range dependencies
- Speed
- Avoids gradient vanishing/explosion

Attention is all you need

http://sigbovik.org/2020/proceedings.pdf#subsection.0.25