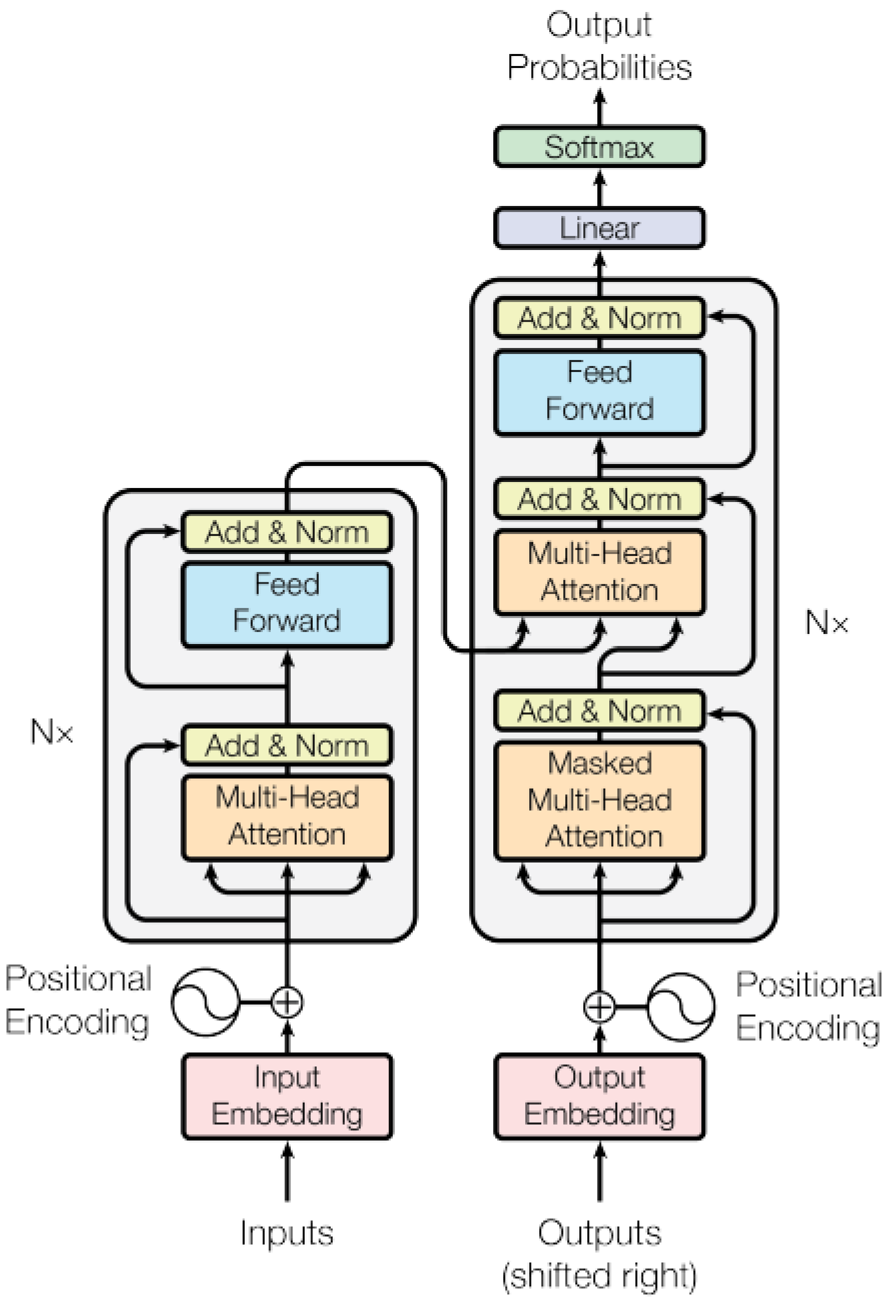
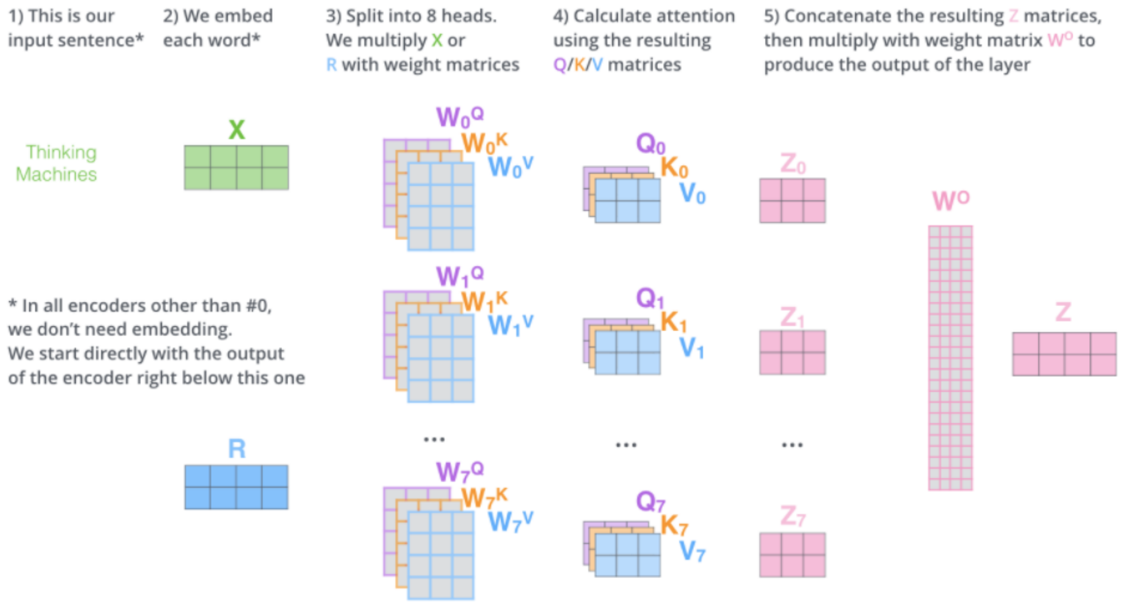
- #paper/read ~ 2017 CE ~ Transformer, Attention Mechanism, Natural Language Processing, NLP
  - **Attention Is All You Need**
  - https://arxiv.org/abs/1706.03762
  - https://jalammar.github.io/illustrated-transformer/
  - https://nlp.seas.harvard.edu/2018/04/03/attention.html
  - Mentioned papers:
    - Gated Recurrent Unit, GRU
    - Sequence to Sequence Learning, Seq2seq
    - Using Attention to Align and Translate
    - RNN Encoder-Decoder
    - Google's Neural Machine Translation
    - Attention-based Neural Machine Translation
    - Exploring the Limits of Language Modeling
    - LSTM for Machine Reading
    - Convolutional Seq2seq
    - Generating Sequences With RNNs
  - Mentioned topics:
    - LSTM
    - Beam Search

- # Summary
  - ## Architecture
    - The model consists of encoder and decoder stacks comprised of sub-layers.



  - Neither encoders nor decoders in this stack share weights, although they are identical in structure.
  - The bottom encoder gets a sequence of N embeddings of size D as an input.
- Self-Attention is used where all W_K, W_V, and W_Q are applied to the input embeddings.
  - This allows the model to consider the most crucial parts of the context while encoding each word.
  - Before Softmaxing, the attention scores are divided by $\sqrt{\text{key size}}$ which leads to more stable Gradients.

        - The default size in the paper is 64 (compared to the size of 512 used for the input embeddings).
    - Attention key size is important for determining the query-key compatibility.
        - Its shortening leads to worse performance and its lengthening leads to higher Compute requirements.
        - A more sophisticated Function than dot product may be beneficial.
    - The weighted sum of the *value* vectors is the output of the self-attention layer (it is fed to the fully-connected network within the current encoder).
- Moreover, self-Attention in Transformers is **multi-headed**.
    - There are h sets of W_K, W_V, and W_Q inside each encoder.
        - In the paper, the default $h = 8$.
    - The resulting weighted sums of *values* are concatenated and fed to an additional weight matrix W_O (in order to shrink them back to the standard input size).
    - It gives the model several *representational subspaces* to avoid dominance of a single word while calculating attention scores.
- Here is a recap of a singe **multi-headed self-attention** layer:


1) This is our input sentence*
2) We embed each word*
3) Split into 8 heads. We multiply X or R with weight matrices
4) Calculate attention using the resulting Q/K/V matrices
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer
Thinking Machines
X
R
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one
W0Q
W0K
W0V
W1Q
W1K
W1V
...
W7Q
W7K
W7V
Q0
K0
V0
Q1
K1
V1
...
Q7
K7
V7
Z0
Z1
...
Z7
WO
Z