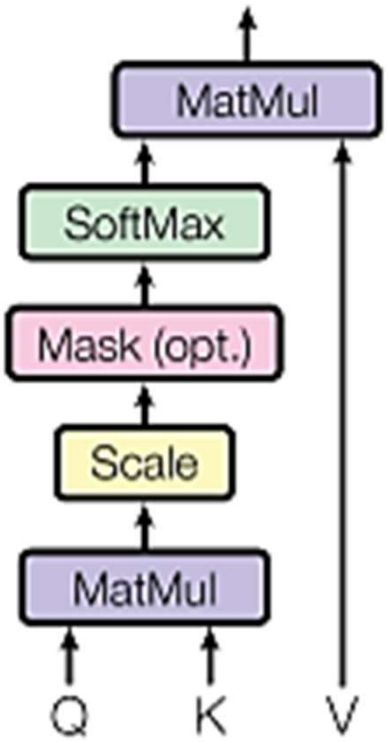
Self Attention Model

Architecture

Working and intuition

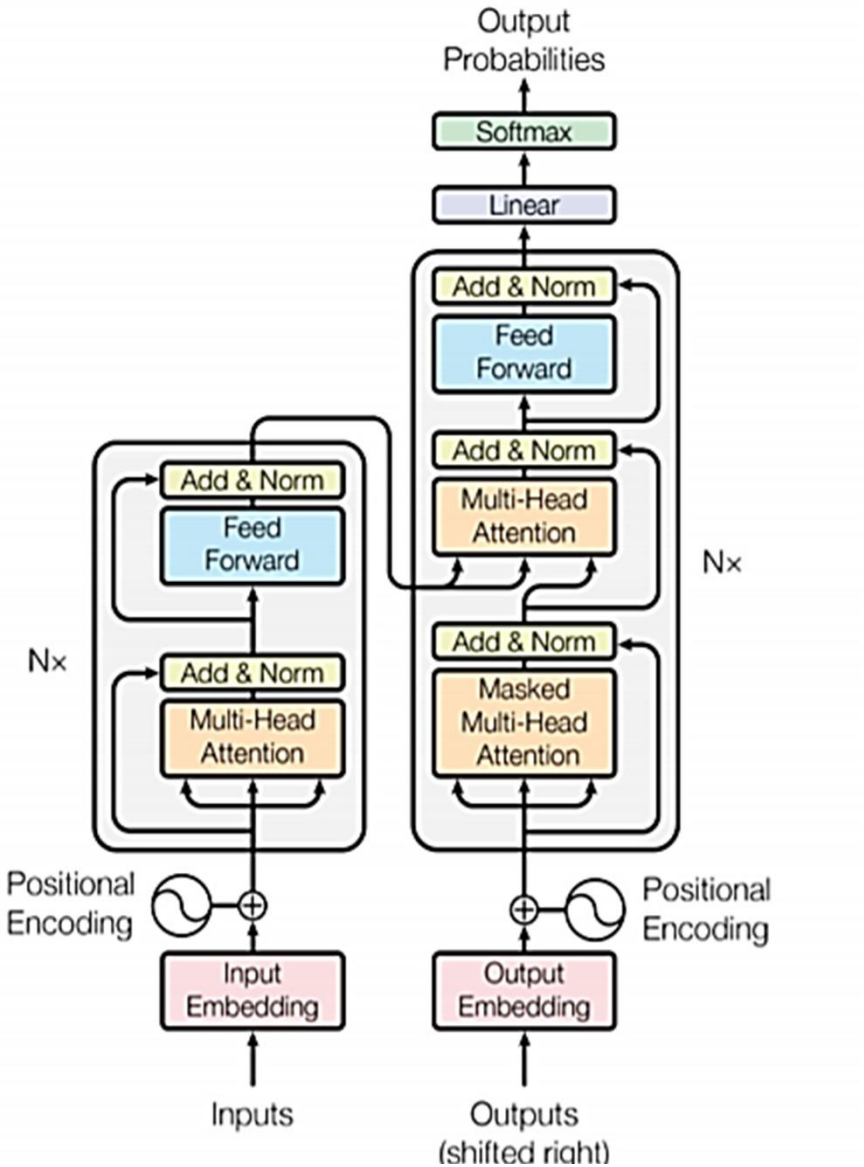
Self Attention Mechanism



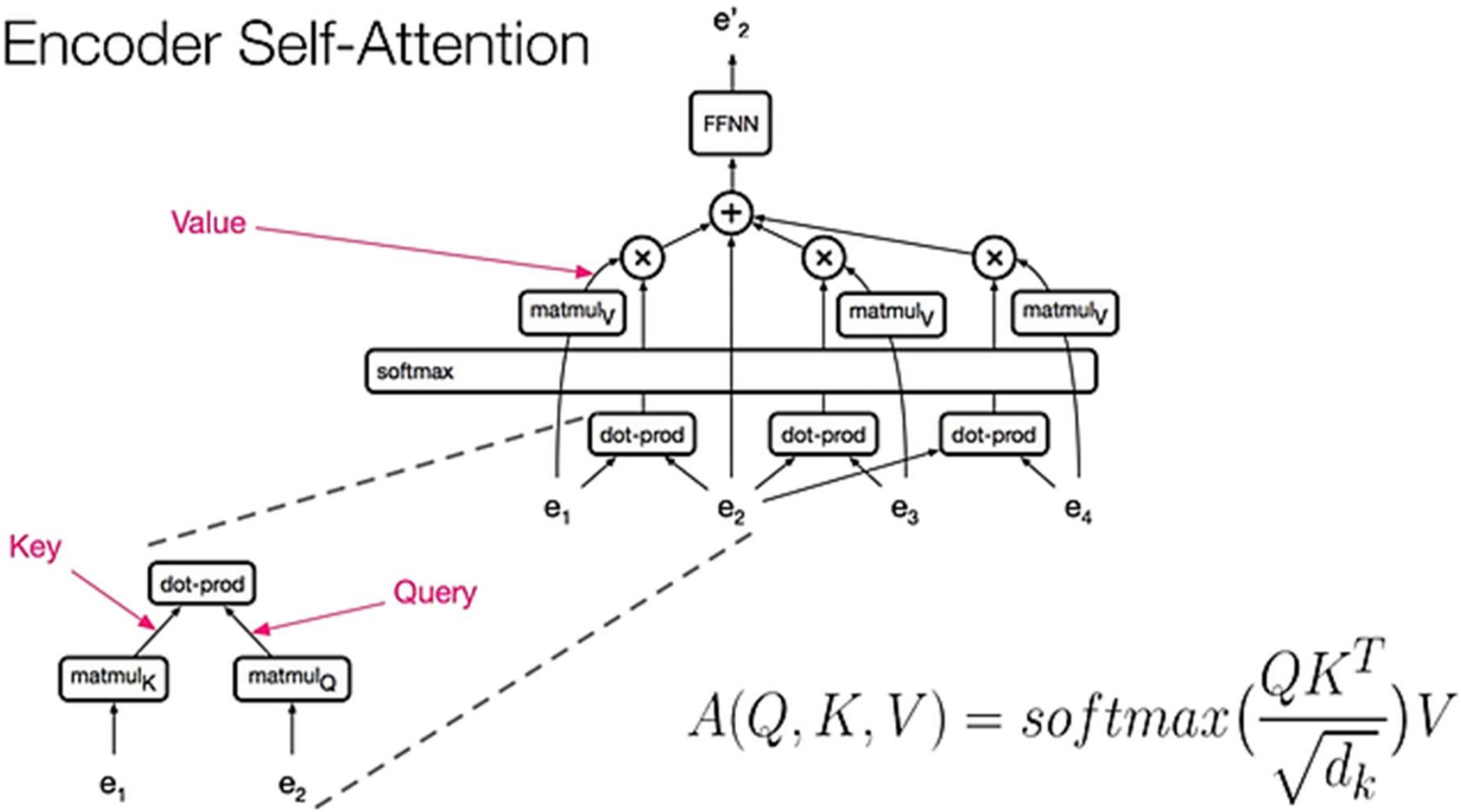
- Multiplying each of the encoder input vectors with three weights matrices (W(Q), W(K), W(V)) that we trained during the training process.
- Multiply the Query vector of the current input with the key vectors from other inputs.
- Divide the score by square root of dimensions of the key vector (d_k).
- Softmax function on all self-attention scores we calculated wrt the query word.
- We multiply the value vector.
- Sum up the weighted value vectors.

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

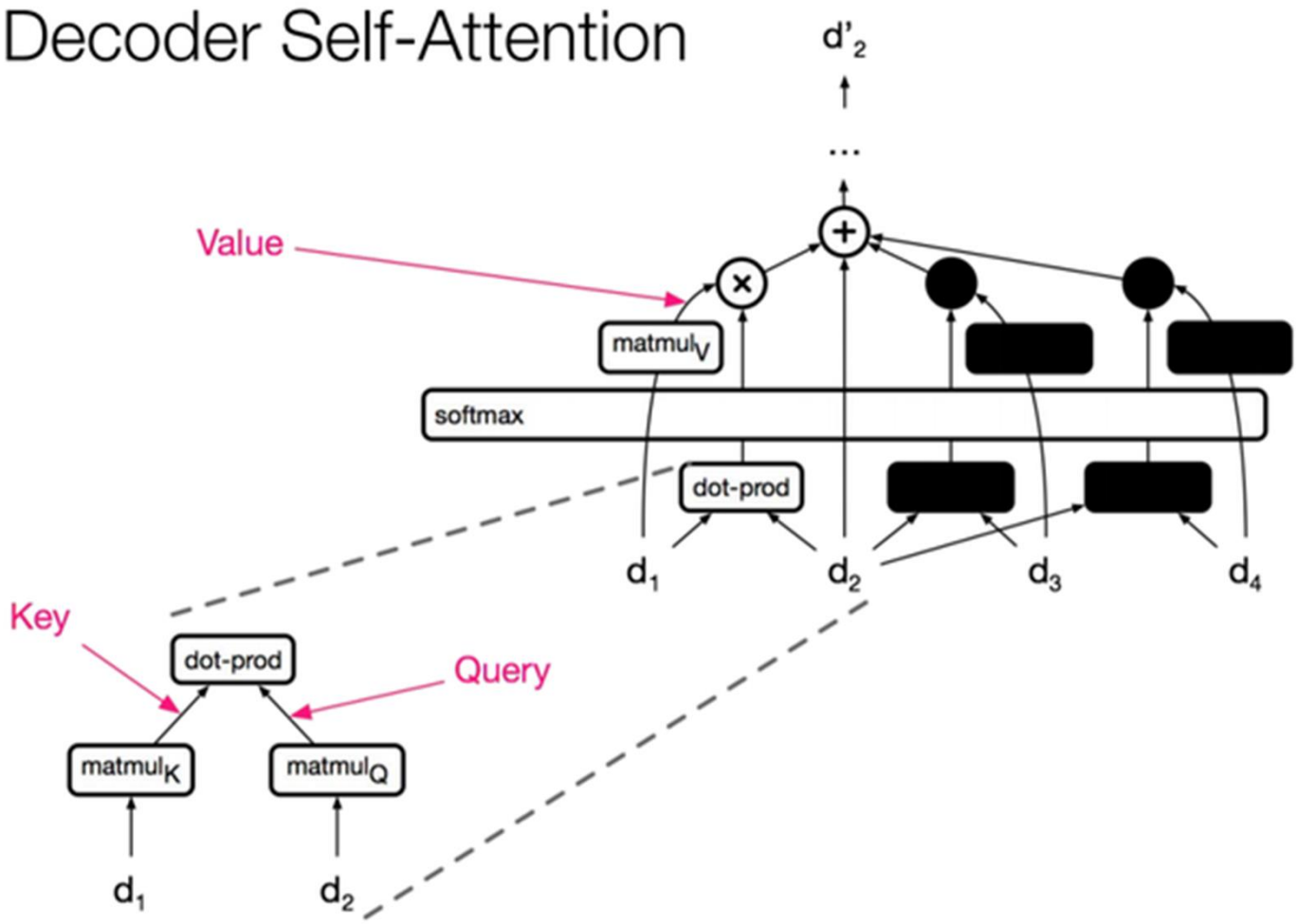
Encoder Decoder Attention



In encoder-decoder attention layer the queries come from the previous decoder layer while the keys and values come from the encoder output. This allows each position in the decoder to give attention to all the positions of the input sequence.



Encoder Self-Attention layer receives key, value, and query input from the output of the previous encoder layer. Each position in the encoder can get attention score from every position in the previous encoder layer.



Decoder Self-Attention is similar to self-attention in encoder where all queries, keys, and values come from the previous layer. The self-attention decoder allows each position to attend each position up to and including that position. The future values are masked with (-Inf). This is known as masked-self attention.