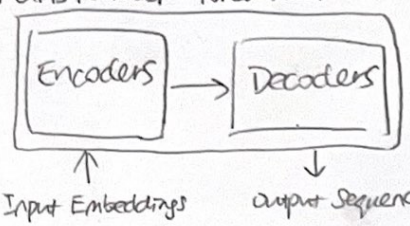
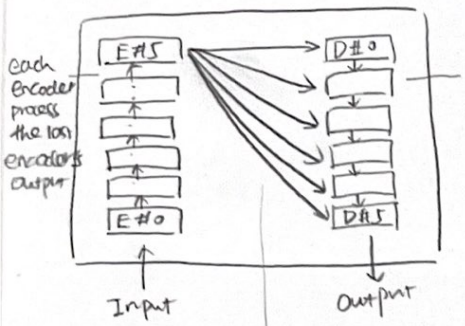
1 Transformer Architecture Encoder



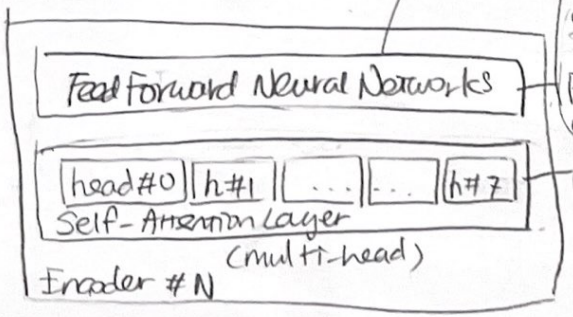
① Same nº of encoders & decoders stacked (6 in the original paper)



each decoder also receives the output of the last decoder

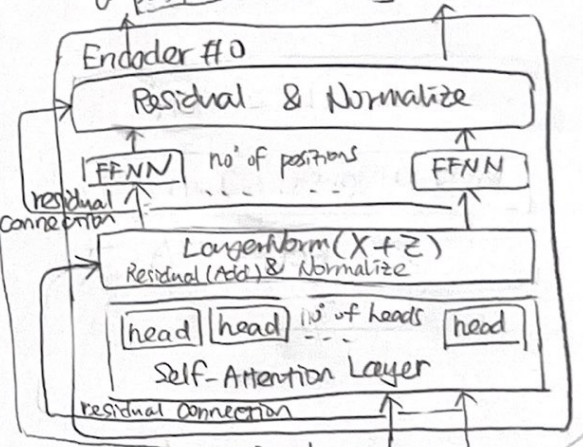
output of the last encoder goes to each decoder

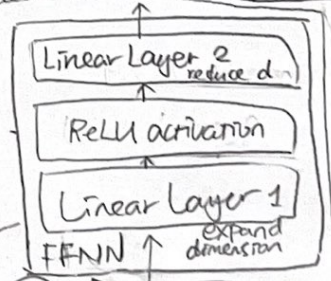
② Encoder layers



②-2

- more detailed Encoder architecture with Residual Connection & Layer Norm & positional encoding





positional encoding

Input Embeddings

pos 1 pos #

②-1 Same FFNN applied to each position's self-attention layer output

parallel processing all positions independently

(calculate all positions inter-dependently) all heads are different

②-3 positional encoding

- gives the model info about the position of each token in the sequence (and relative positions)
- same dimension as embedding (so they can be added together)
- use sine and cosine functions of different frequency

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right)$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d}}\right)$$

i: vector index, $\in (0, \ldots, \frac{d}{2} - 1)$

d: dimensionality of embedding

- a way to generate unique encoding for each position
- can scale to sequence lengths that are bigger than max sequence length in training data