MULTI HEAD ATTENTION

Attention: $\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$

$$\mathrm{Softmax}\left(\frac{\boxed{Q}_{(seq,d_k)} \times \boxed{K^T}_{(d_k,seq)}}{\sqrt{d_k}}\right) \times \boxed{V}_{(seq,d_v)} = \boxed{\text{attention}}_{(seq,d_v)}$$

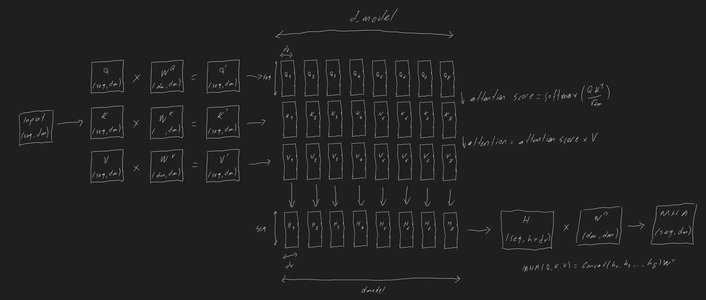
where, $d_q = d_k = \frac{d_{model}}{h} = 64$, $h=8$

$d_{model} = 512 \rightarrow d_q = d_k = 512$

Seq \ Seq	I	am	Karan	?
I	0,3	0,4	0,2	1
am	0,15	0,33	0,49	1
Karan	0,51	0,09	0,42	1

$$\times \boxed{V}_{(seq,d_v)} = \boxed{\text{attention}}_{(seq,d_v)}$$

$$\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_m}}\right)$$

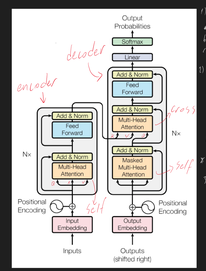


LAYER NORMALIZATION

$$y_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \cdot \gamma + \beta$$

where $\underbrace{\gamma(\text{multiplication}), \beta(\text{addition})}_{\text{learnable parameters}}$

ENCODER



1) Input Embedding(learned during training)
   Tokenizing the input split the sentence into word
2) Input id increasing rise
3) embedding of input id mapped into the vector (d_model)
4) Positional Embedding(fixed, sin and cos training formula)
   provide the positional info of each w sentence
   & keep spatial distortion

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

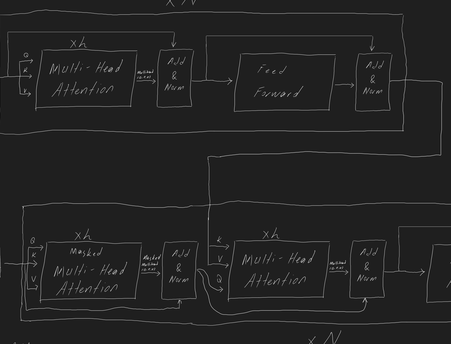
$$PE(pos,2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

* sin/cos function are continuous pattern

5) Multi-head Attention:
   self attention relate words to each other

Attention: $\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$

$$\frac{\boxed{Q}_{(seq,d_k)} \times \boxed{K^T}_{(d_k,seq)}}{\sqrt{d_k}} \qquad d_{model} = 512$$



Ben Karan <EOS> → this is total

↓

Cross Entropy Loss

Hyperparameter:
- d_model embedding dim → 512
- seq_len max token length → 128
- h number of attention heads → 8
- d_k dimension of key matrix
- d_v dimension of value matrix } $d_k = d_v = \frac{d_{model}}{h} = 64$
- d_ff dimension of feed matrix
- FFN inner hidden layer → 2048
- N number of layers in encoder/decoder → 6

Positional Encoding:
- PE has the same dimension as d_model
- $PE(pos,2i) = \sin(pos/10000^{2i/d_{model}})$
- $PE(pos,2i+1) = \cos(pos/10000^{2i/d_{model}})$

Scaled Dot Product Attention:

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

Multi-Head Attention:

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O$$

where $\mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

Position-wise Feed-Forward Networks:

$$FFN(x) = \mathrm{ReLU}(xW_1 + b_1)W_2 + b_2$$

Residual Connections:

$$\mathrm{layer_norm}(x + \mathrm{sublayer}(x))$$

1) Input:
   words are mapped to token based on vocabulary
   tokenized words are better than to others
2) Input Embedding:
   tokens are converted into embeddings → a dense vector
3) Positional Encoding:
   transformers are permutationally invariant, meaning that the order of word order doesn't change the prediction
   Positional encoding provides awareness of the position of each word
4) Multi-head Attention:
   a) Self attention
      Attention between each word for understanding
      & kept the global value and they are ignored by
      mapping - not out to calculate any attention
      for decoder input only the tokens until recent
      prediction is given and rest is masked/ignored
   b) Cross attention
      Q and V matrices from encoder is fed through multi-head attention block of decoder and K comes from masked multi-head self attention block of decoder.
5) Residual connection
   After attention block the input is added to output of sub-layer and normalized.
6) Feed Forward layer
   After attention block, the output is fed to linear layers.

Masked Attention

Seq \ Seq	I	am	Karan
I	0,3	-∞	-∞
am	0,15	0,33	-∞
Karan	0,51	0,09	0,42

* applied before softmax: $-\infty \rightarrow 0$

x N

DECODER