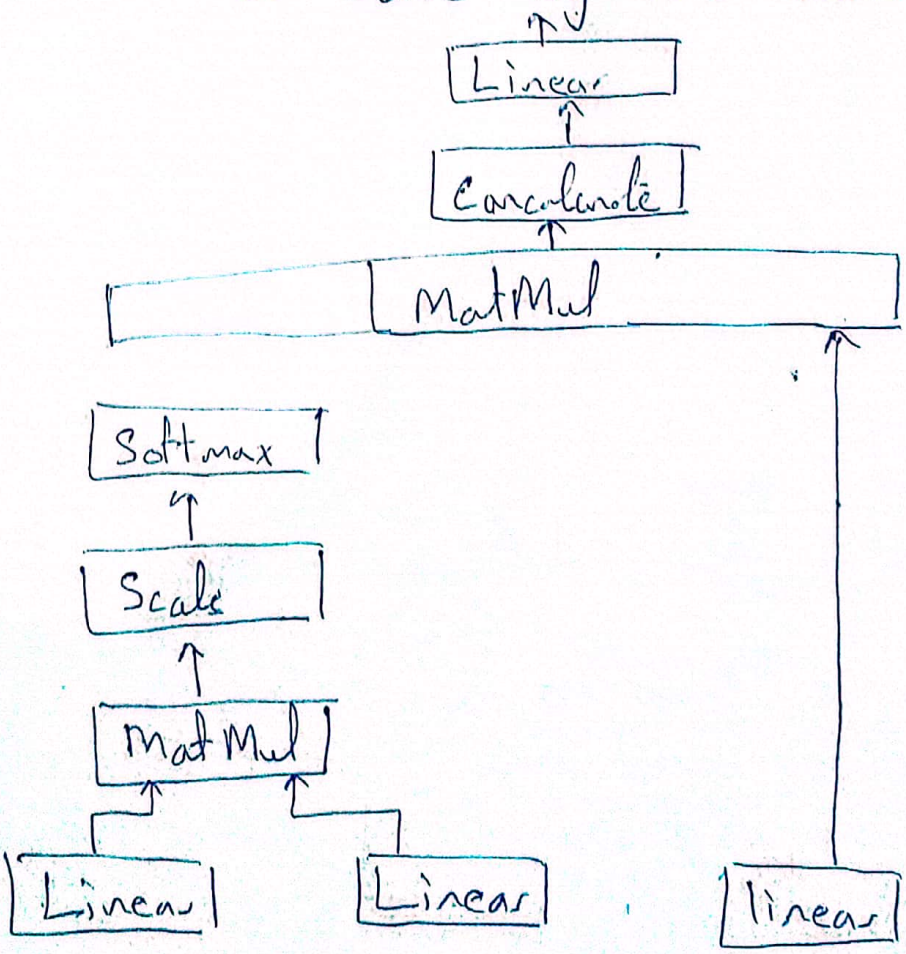
Our mind will not pay equal attention to all the words in the sentence

Attention: Focus on specific words in a sentence

"I went to a bank"

and

"I live by river bank"



(2)

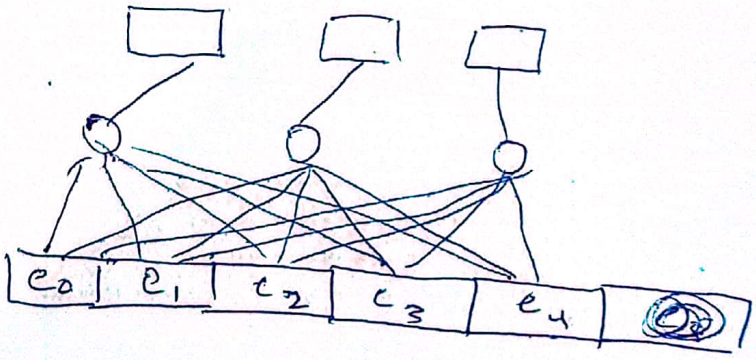
Three liner layers:

a combination of neurons without activation function

Purposes:

1) Mapping inputs on to the outputs
2) Changing the dimensions.

different sets of weights.



Size shrunk 5 to 3

save computation

These weights are fed as a matrix

Why 3 layers: Each has a specific function Q, K, V

request titles contents

Similarity b/w Q and K

Cosine Similarity ? -1 to +1

↓ ↓

opposite direction Same direction

$$\text{Cos}(A, B) = \frac{A \cdot B}{\text{Scaling}}$$

$$\text{Similarity using matrix} = \frac{A \cdot B^T}{\text{Scaling}}$$

$$\text{Similarity}(Q, K) = \frac{Q \cdot K^T}{\text{Scaling}}$$

~~Refer~~.

Query layer:.. Position-aware Embeddings
- two more copies to key and value layer.
- Rely on train - self attention for improvement
- Multiply with linear layers transpose with embedding matrix. each linear layer has its own weights

1) focus on Q and K

Compute dot product of Q and K

o/p of dot product → Attention filte

Once the training period is over

7x3 . 3x7

weights are more meaningful.

Scale attention Scores:

- Divide the attention scores by the dimension of key vector.
- Apply Softmax $\text{Softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right)$

We now have

- Value matrix.
- Attention filter (K, Q). - multiply them $\text{Softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V$

analogy with Computer Vision.

filtered value matrix o/p of multi-head attention.

Why need multi-head

focus on diffent liguistic phenomenon.

Orignal Paper: 8 heads.

- Concatenate the three heads together.

- Apply linear to shrink size

(6)

Residual Connection (information is not lost)

- Knowledge Recursion
- Vanishing gradient

Add & Norm:

Simple addition which gets fed

- Normalization: layer normalization standardization along axis # features

$$x_i = \frac{x_i^d - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

↓

does not add 0

Linear layer with RELU

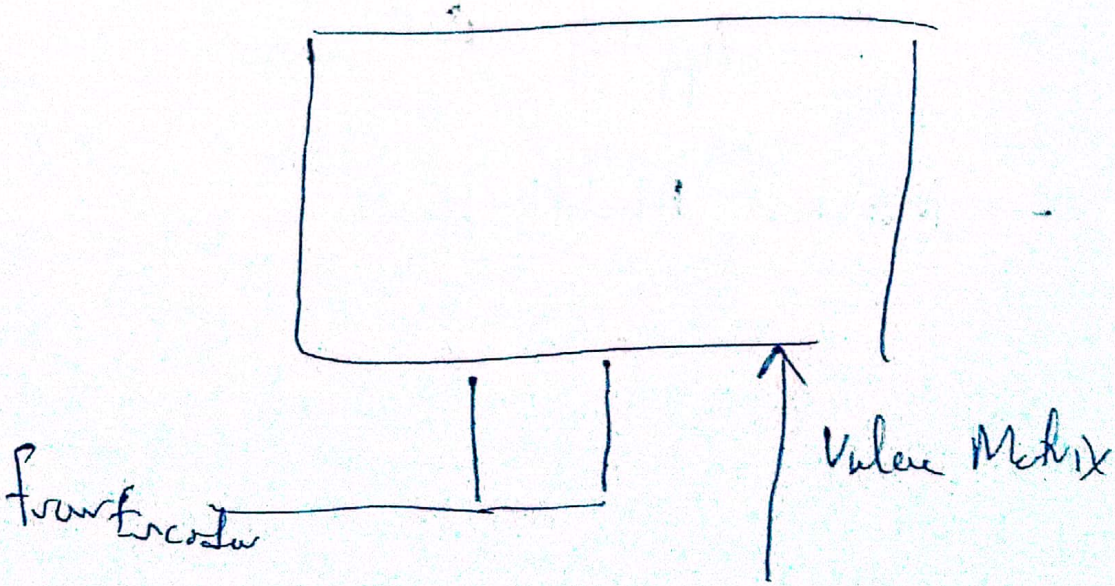
Encoder : Takes input and convert to vectorized o/p

Decoder :- Two inputs

o/p of encoder is split in to two copies K, & 2nd input to decoder

First input of Decoder:

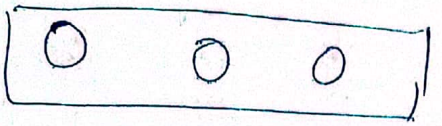
- 1st word : <Start>
- Marked Multi-head
- Add & Norm
- Value Matrix



final linear layer

— fully connected Neurons

# of classes.



dialogue generation

total vocabulary size

the final linear layer we need score

— Multiply the matrix

- We get single scores for each word (logit)

— Apply Softmax

— Masked Multi-head

Unmask

During training
Mask is shown correct word
is shown teacher forcing

loss (prob (Pred), prob (Truth))

- Cross entropy loss
- Masking the input

Difference b/w train and testing
- We mask in both
- During train we apply actual words to decoder
- By inference we input predicted words