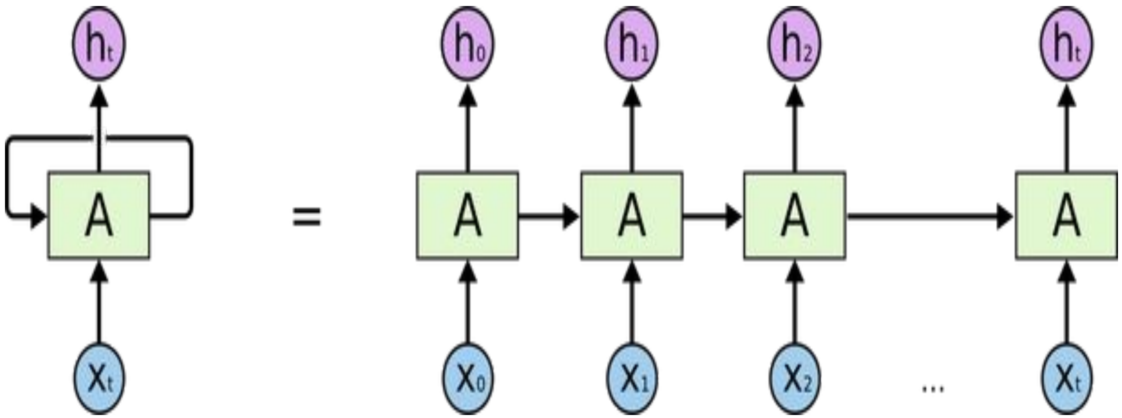
NLP Session

Topics

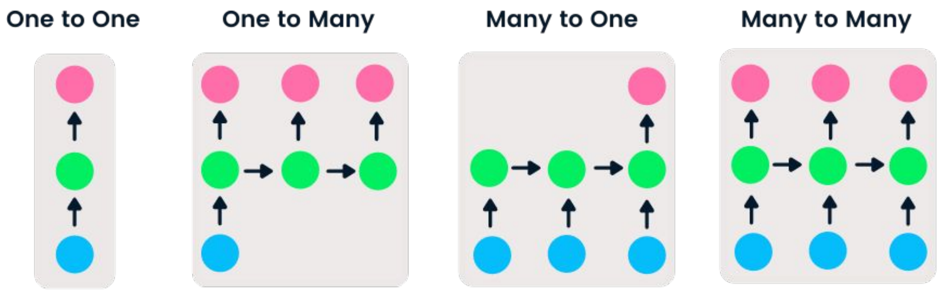
- RNN
- LSTM
- Transformers

RNN

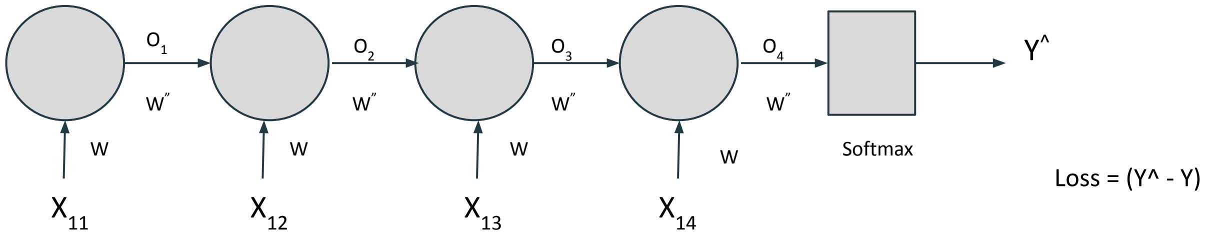


Types of RNN

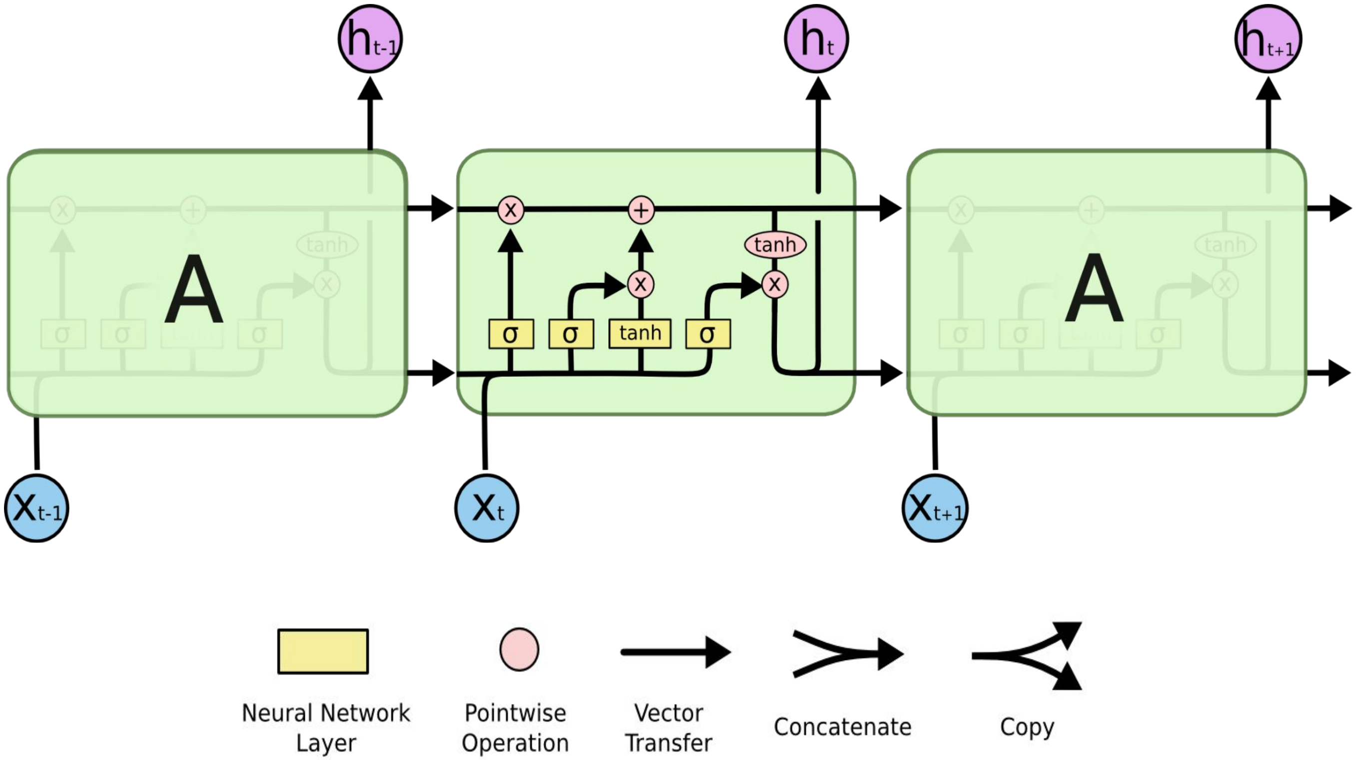
- One to One RNN
- One to Many RNN
- Many to One RNN
- Many to Many RNN



Forward and Backward Propagation



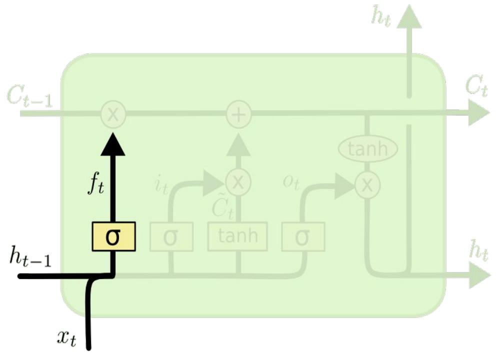
LSTM



h_{t-1}
h_t
h_{t+1}
A
A
σ
σ
tanh
σ
tanh
X_{t-1}
X_t
X_{t+1}
Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy

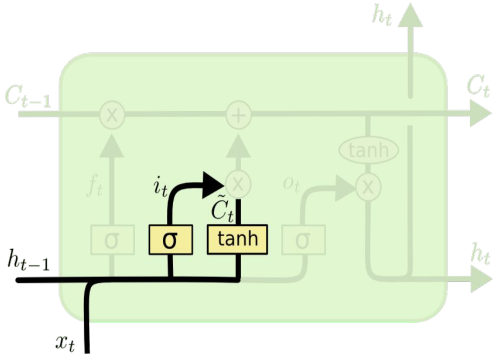
Forget Gate Layer

Example 1 - Rohan likes pizza but he does not like burger.
Example 2 - Rohan likes pizza but his friend likes burger.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

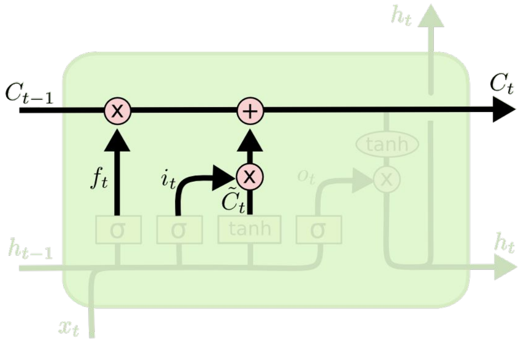
Input Gate Layer



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

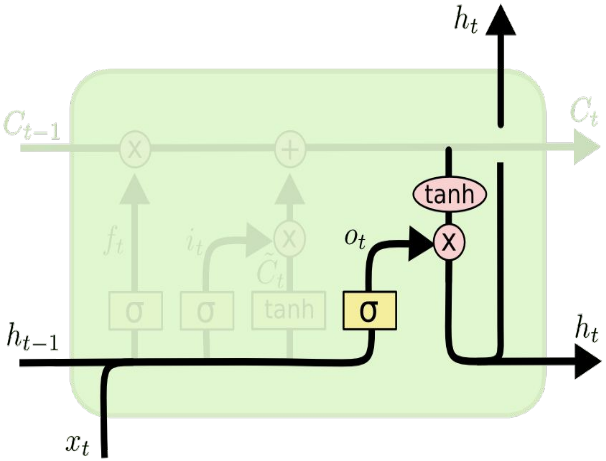
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Combining



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

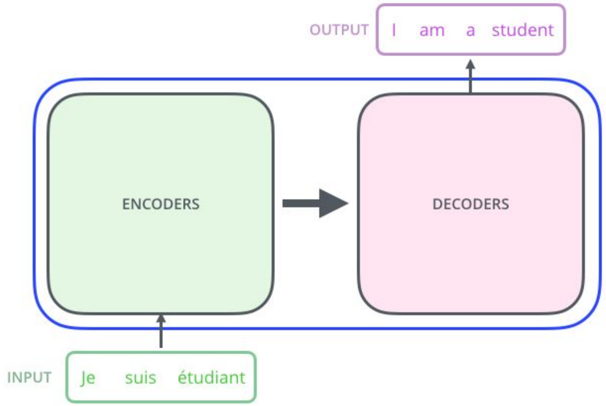
Output Gate Layer

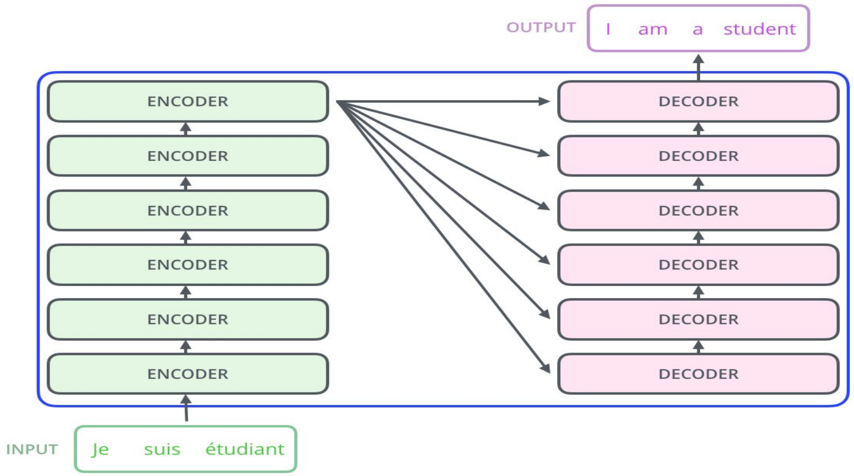


$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] + b_o\right)$$

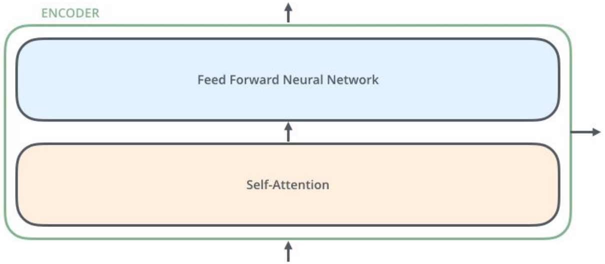
$$h_t = o_t * \tanh\left(C_t\right)$$

Transformers





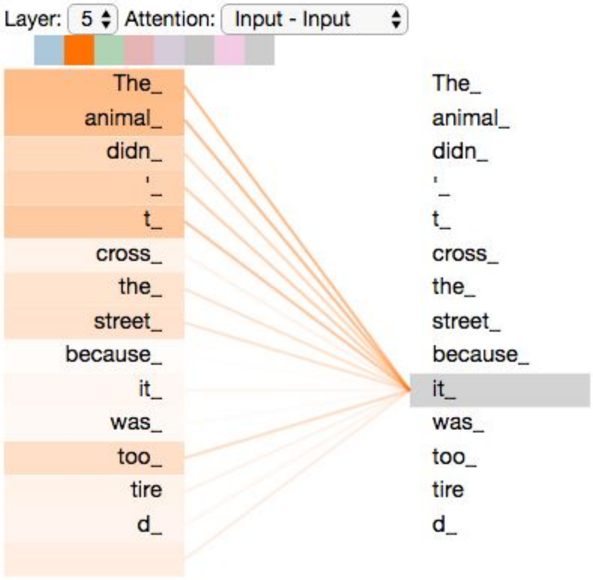
In Every Encoder -



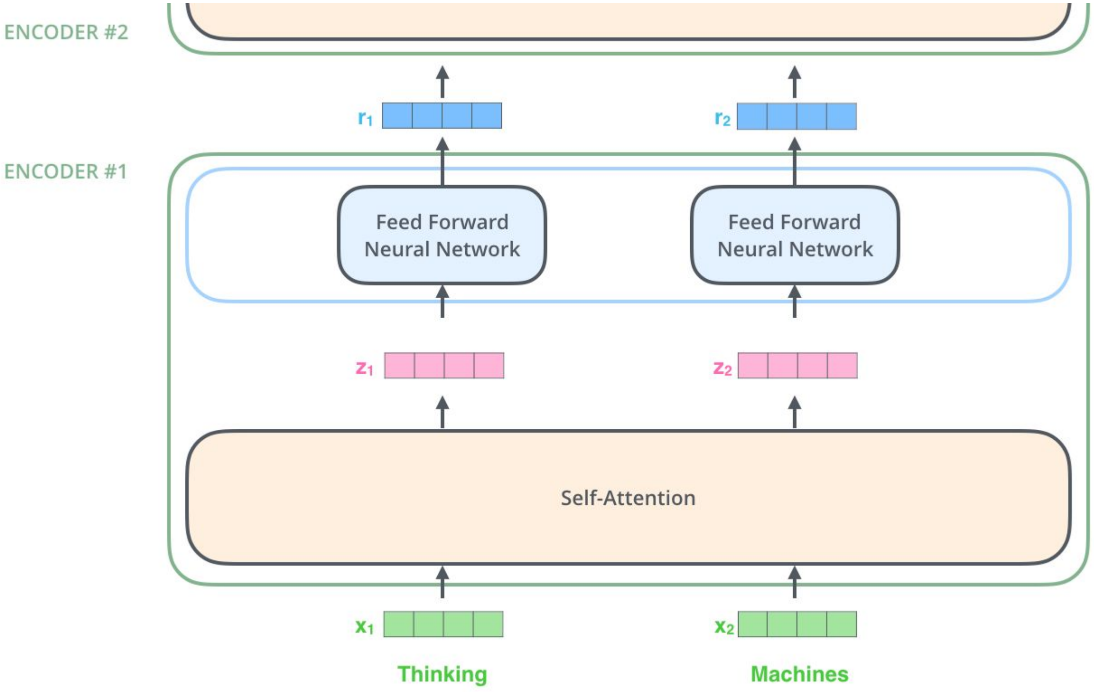
Self - Attention

Why to use ?

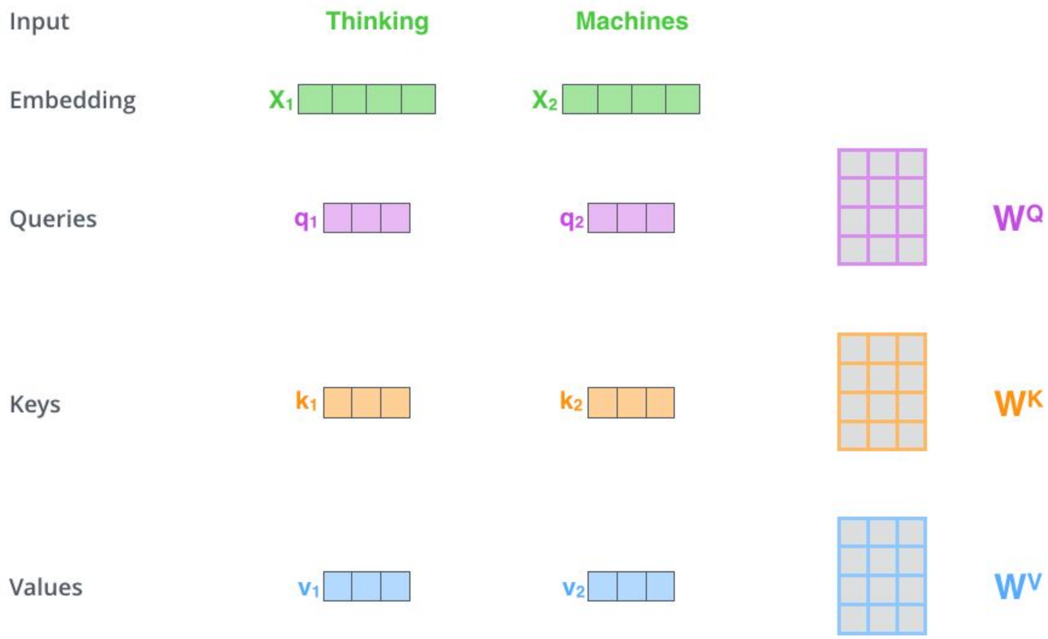
Example - The animal didn't cross the street because it was too tired



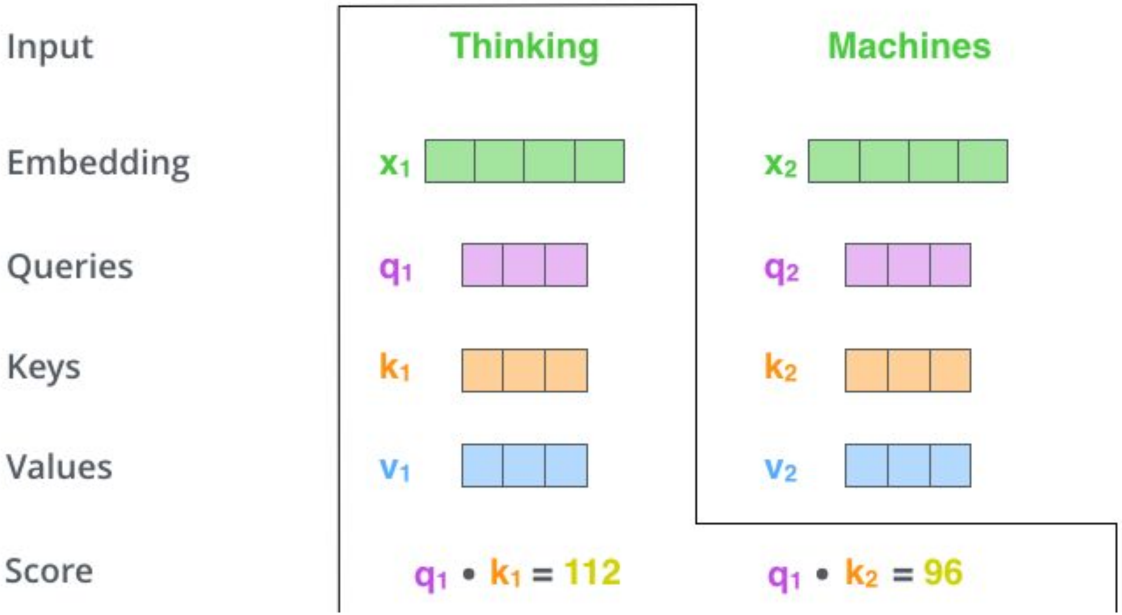
Understanding self-attention



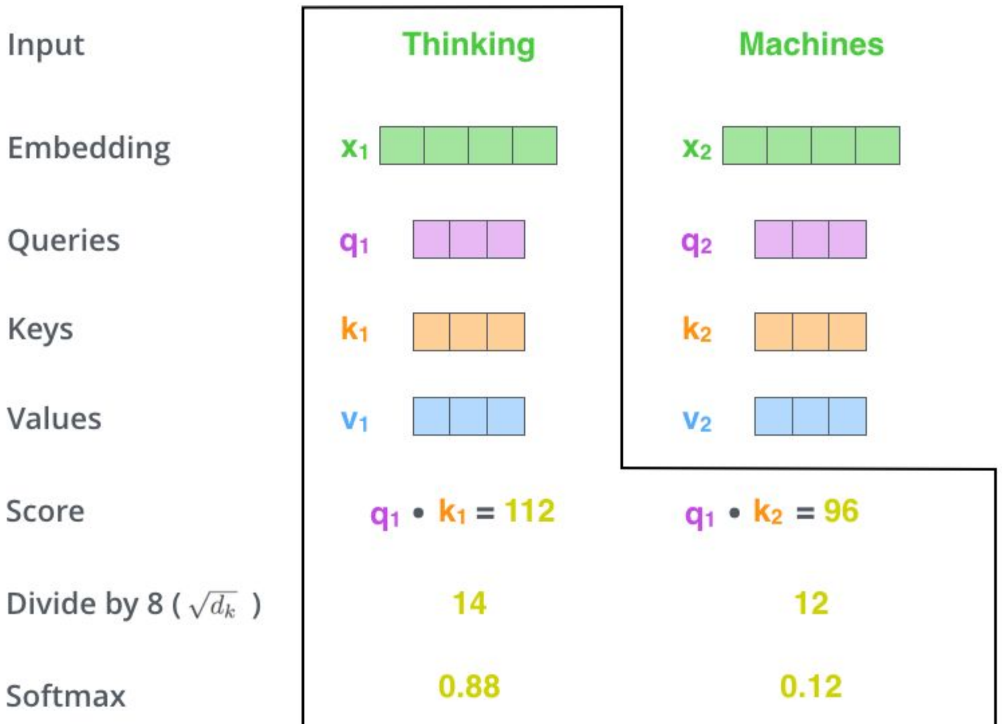
Step 1 - To create Queries, key and values vector



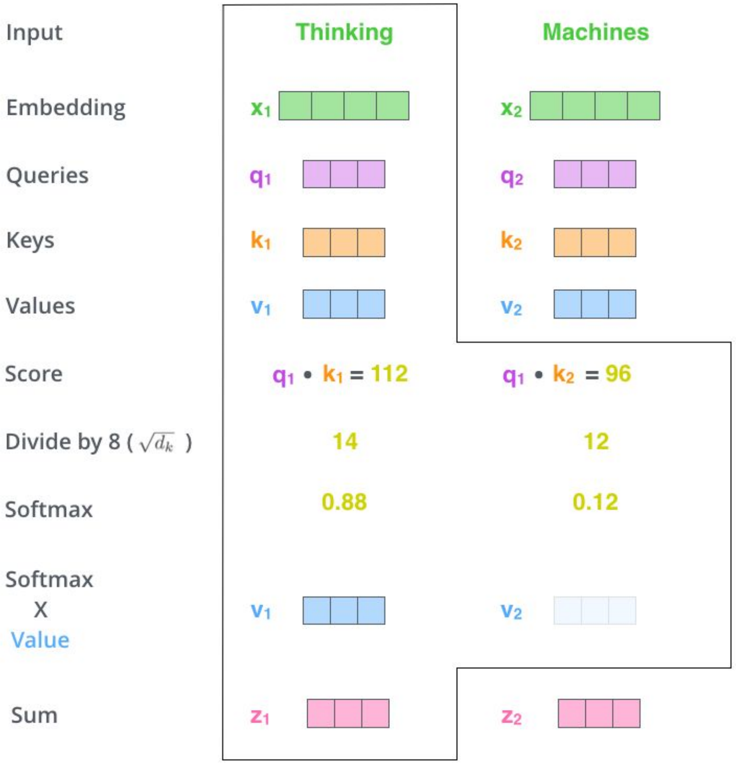
Step 2 - To calculate the score



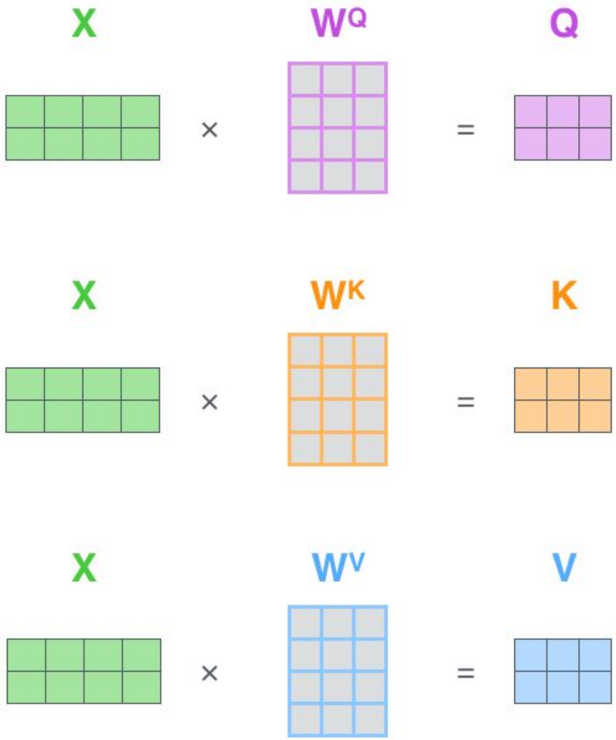
Step 3 - Divide the score by 8 and calculating the softmax



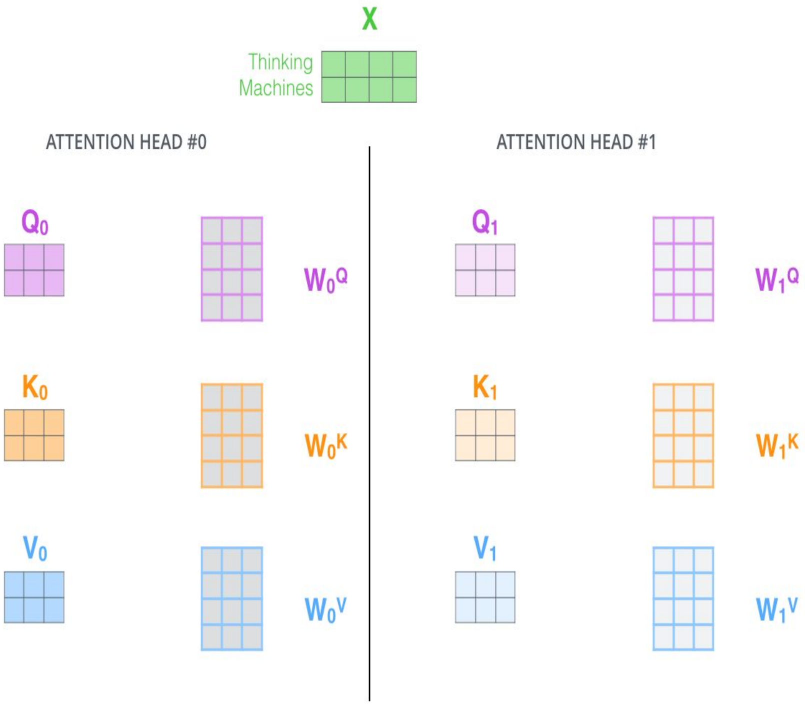
Step 4 - Multiply the value vector with softmax

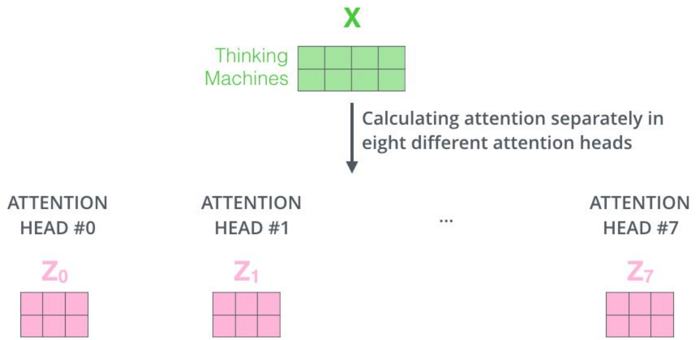


Matrix Calculation of self-attention



Multi-headed Attention



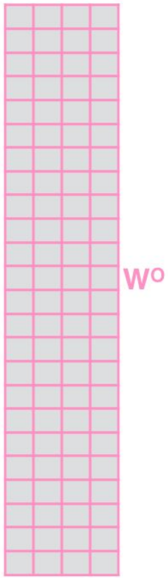


Input to feed-forward neural network

1) Concatenate all the attention heads

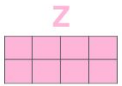
2) Multiply with a weight matrix W^O that was trained jointly with the model

X



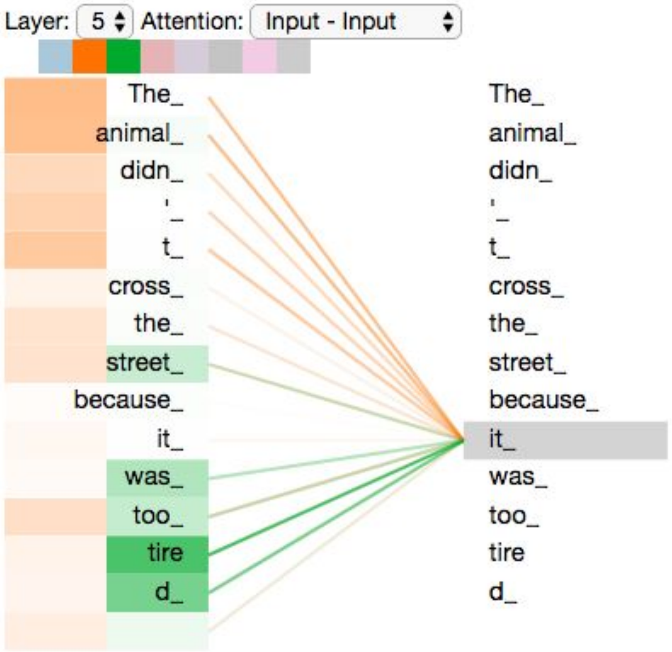
3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

=



After Applying multi-headed attention

Example - The animal didn't cross the street because it was too tired



Entire Process

1) This is our input sentence*

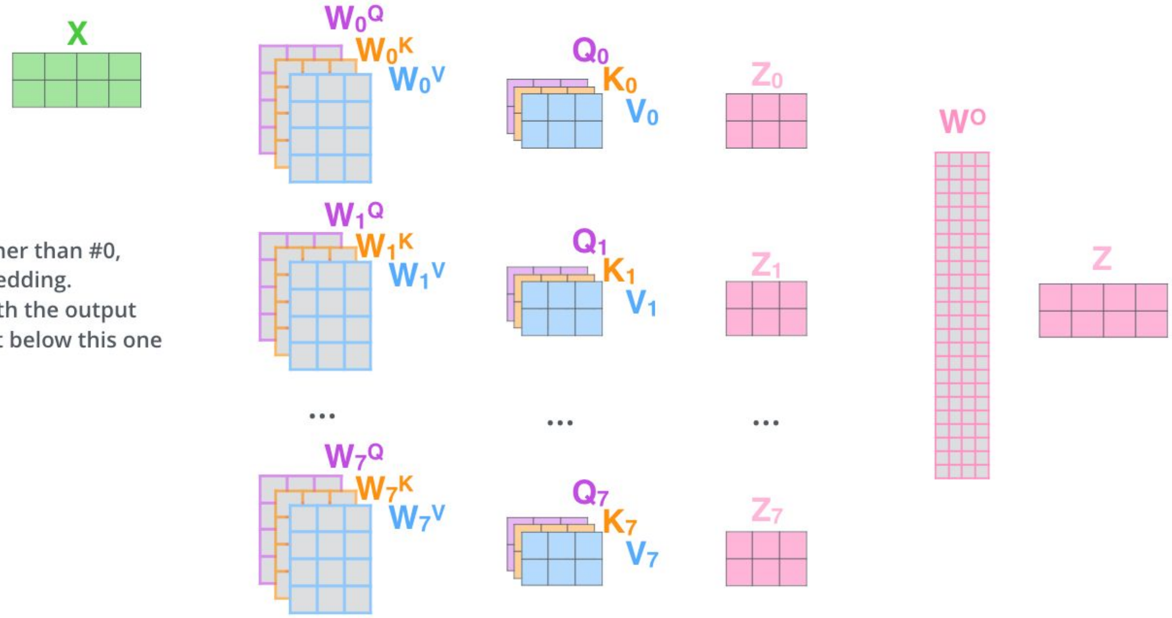
2) We embed each word*

3) Split into 8 heads. We multiply X or R with weight matrices

4) Calculate attention using the resulting Q/K/V matrices

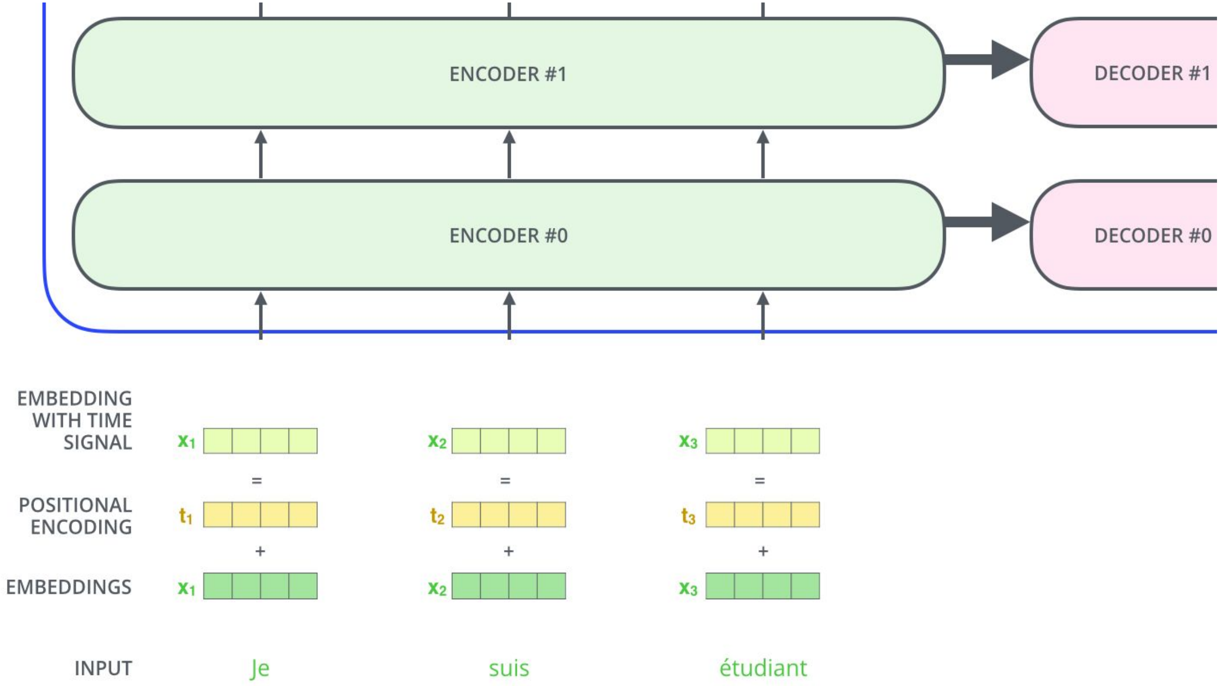
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

Thinking
Machines

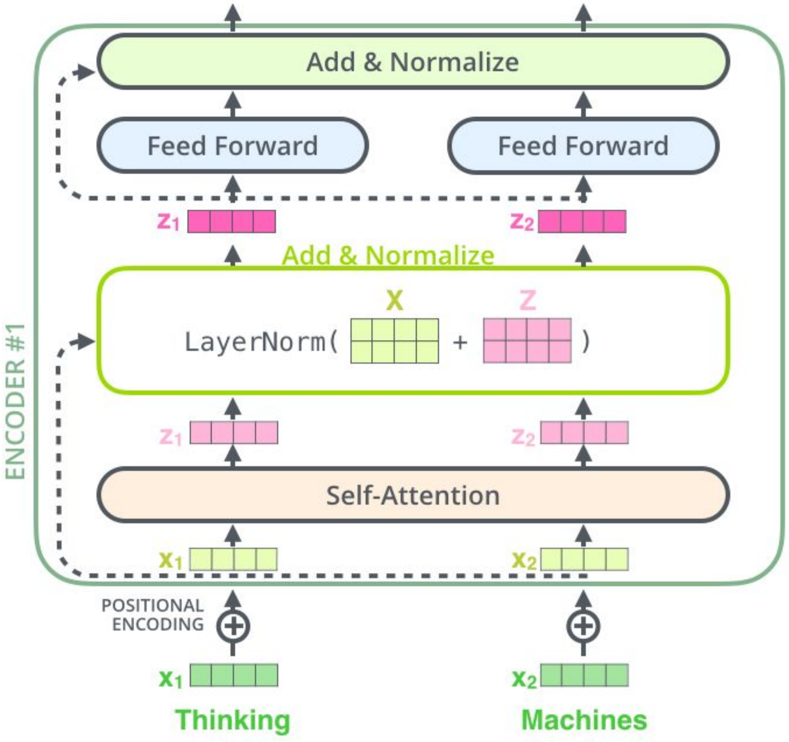


* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

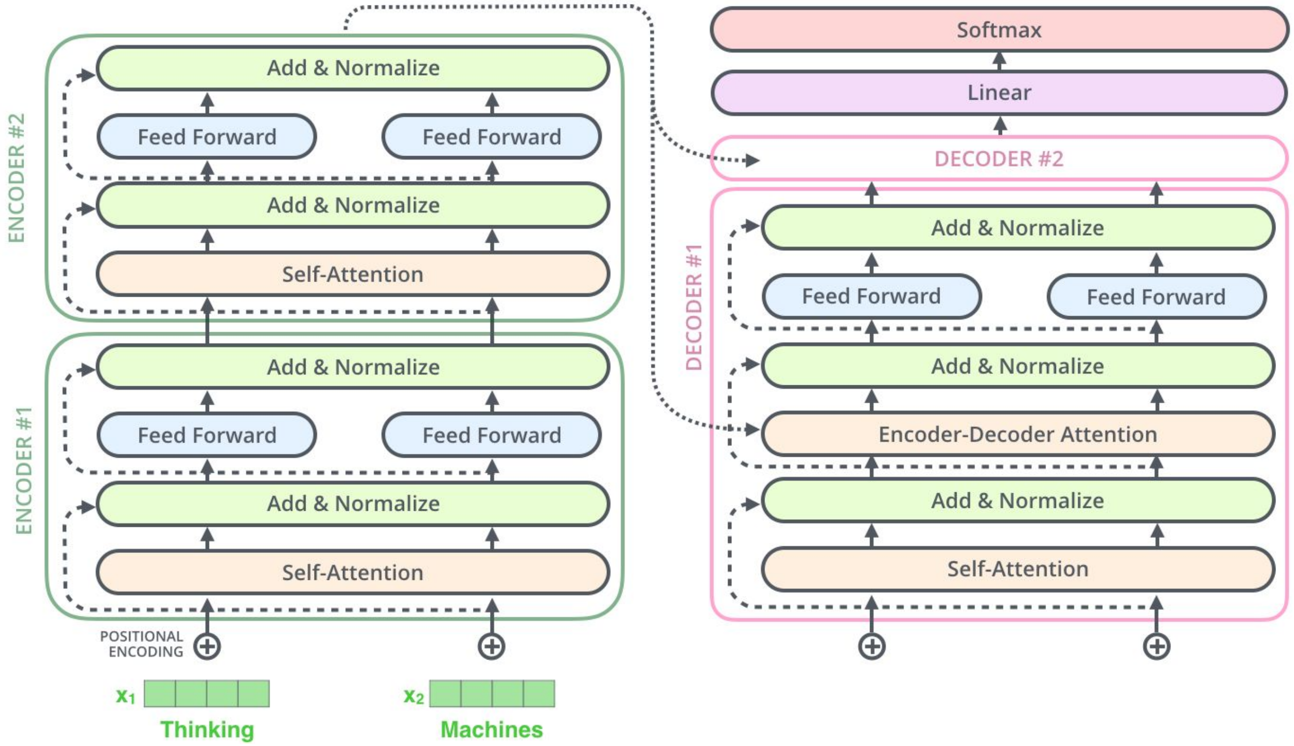
Positional Encodings



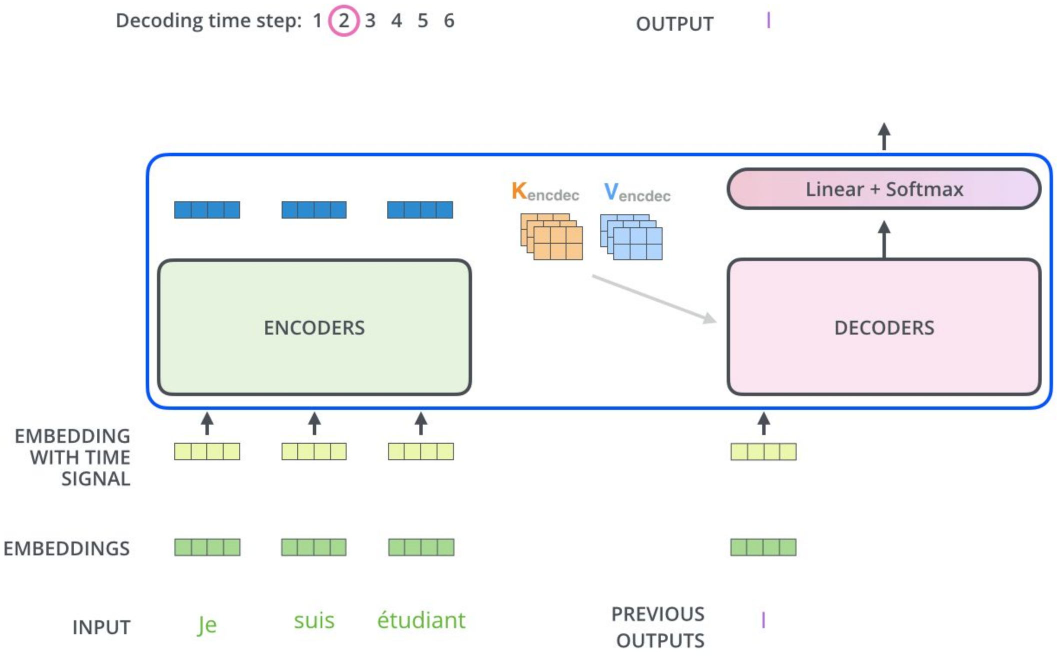
Layer Normalization



Entire Process



Output Visualization



Thank you