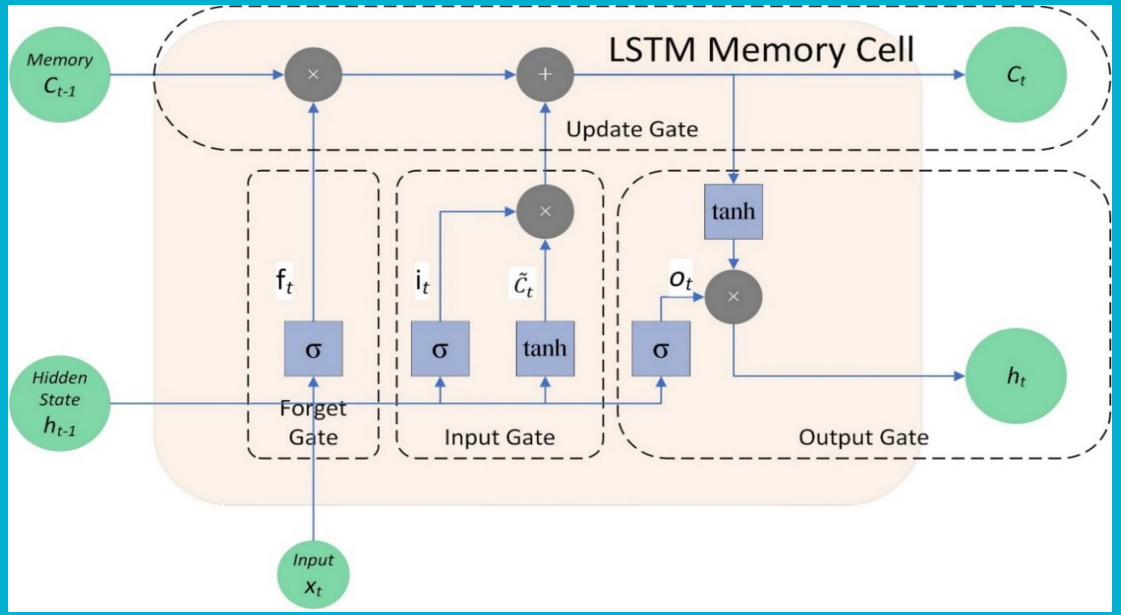
NLP - Section 8

—

LSTMs and Sequence2Sequence

LSTMs

- RNNs but with fancy math.
- The main idea is conserving long term memory and choosing what to remember
- To do that we have 3 main gates and two outputs.



https://www.nature.com/articles/s41598-025-06809-w

Memory Cell

- There are no weights in the memory cell. it can be modified by two operations only (sum and multiplication).
- The lack of weights is to counterfeit the vanishing/exploding gradients problem
- If the memory cell doesn’t contain weights, then it can be transferred directly to the last input without being diminished

Forget Gate

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

- Forget gate determines what to remember based on the input and the hidden state.
- Since sigmoid returns values of mostly 0 or 1, if the value returned for the forget gate is 0, then the memory cell is wiped out.
- If the value returned is 1, then the memory cell doesn't change.

Input Gate

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

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

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

- The input gate determines how much of the current input and the hidden state (short term memory) should be added to the memory cell (long term memory).
- i_t represents the same “choosing” technique displayed in the forget gate.
- ~C_t represents the combination between the hidden state and the input.
- C_t has now i_t*~C_t which represents the “choosing” by processing the combination through 0 or 1

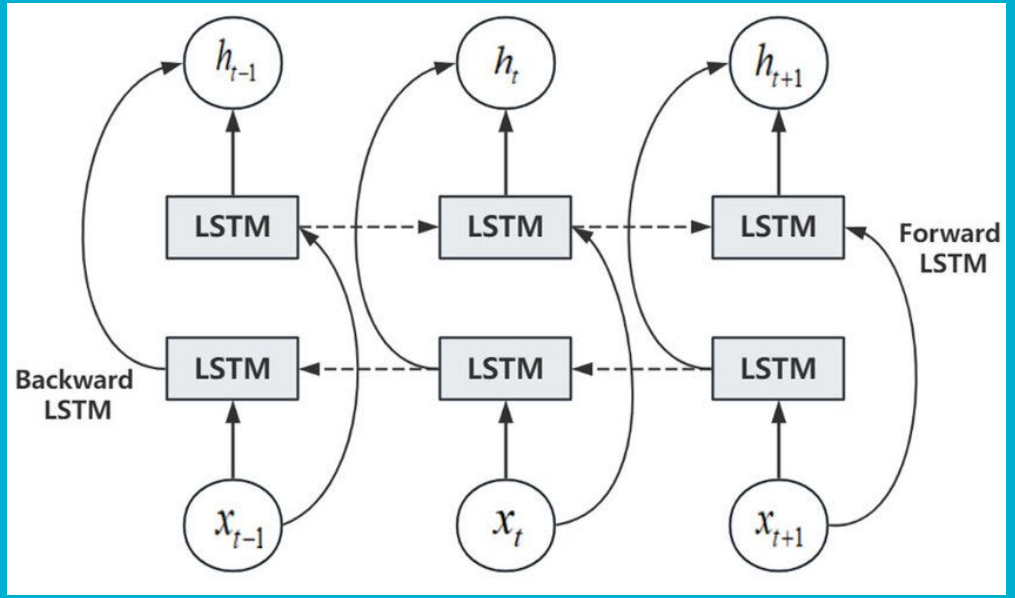
Output Gate

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

- This gate outputs the hidden state which is also the short term memory for the next input.
- It does the same “choosing” technique and outputs the hidden state based on what it selects from the memory cell after the activation function tanh

Bi-LSTM

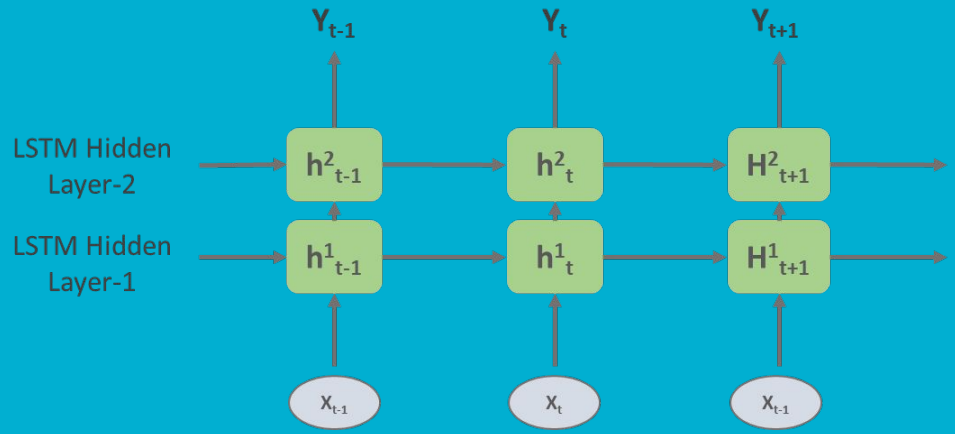
Run the input forward and backward through the same LSTM and concatenate or sum the hidden states of both directions.



https://www.researchgate.net/figure/BiLSTM-model-structure-diagram_fig2_381367239

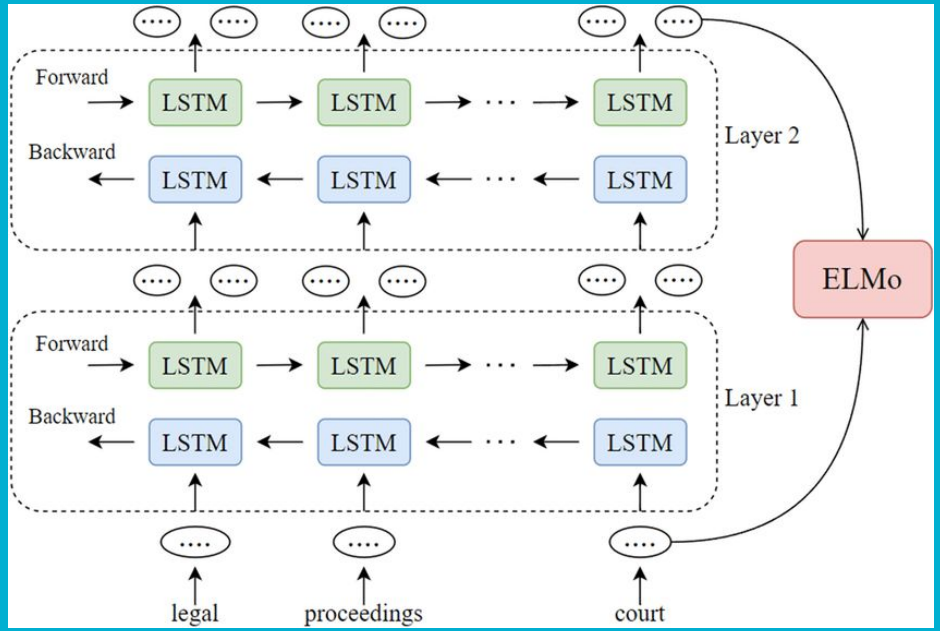
Stacked LSTM

- Here we pass the hidden state t through another LSTM layer.
- Can stack infinitely but would require skip connections.



ELMo

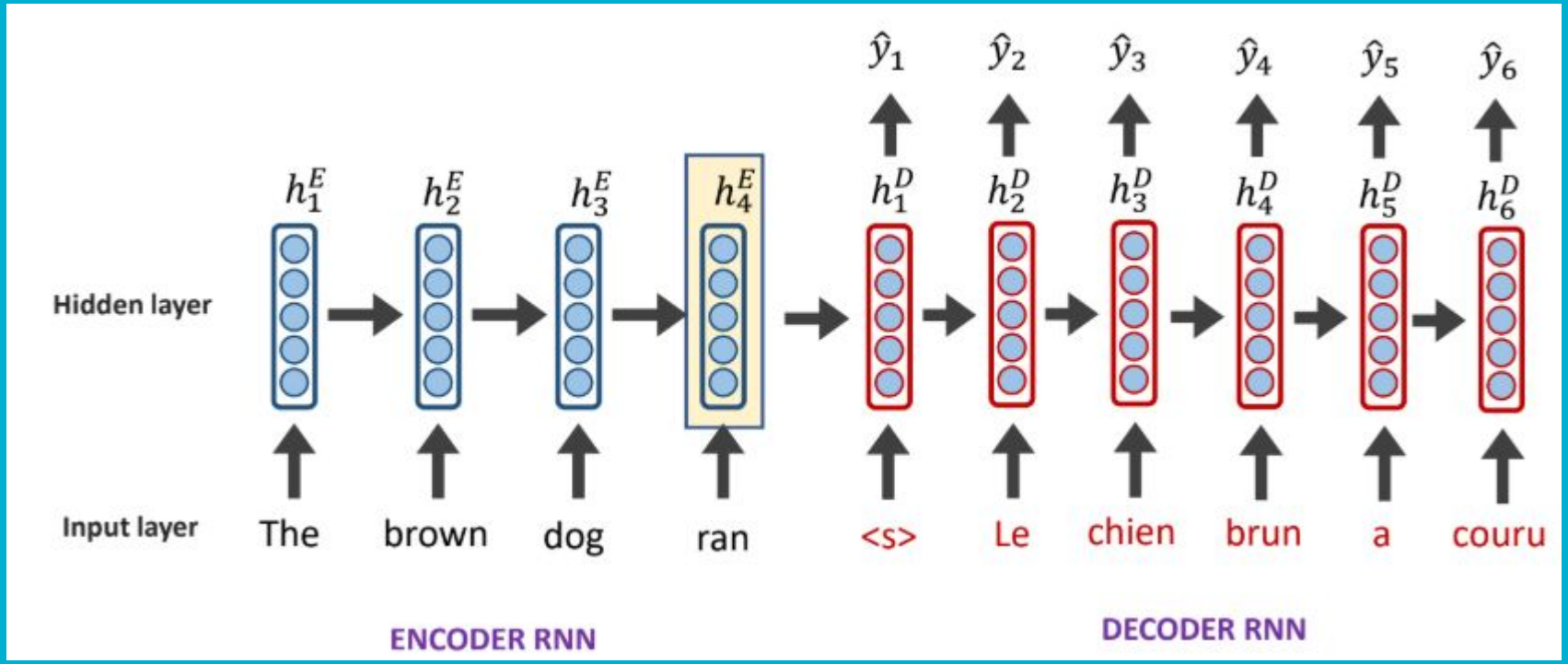
- ELMo is a combination of Bidirectional and Stacked LSTMs
- ELMo yields incredibly good contextualized embeddings.



Seq2Seq Modeling

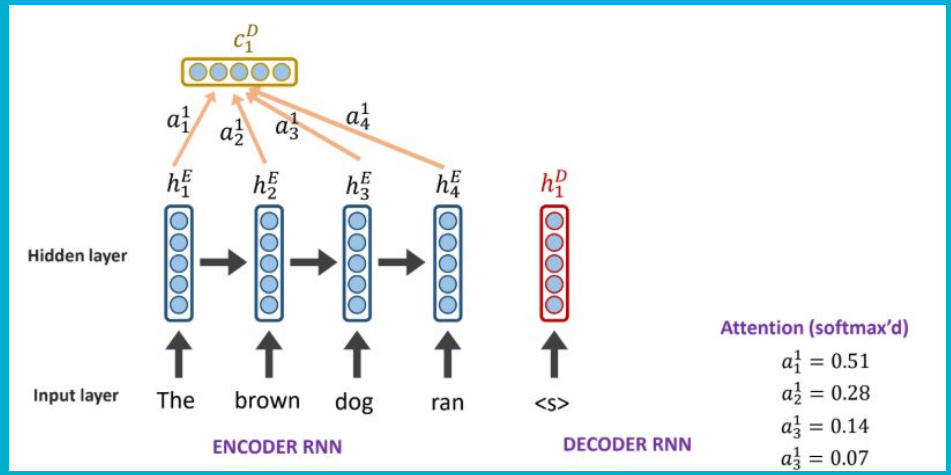
- Tasks such as: Machine Translation and Question Answering
- Takes n and outputs m. (different sizes)
- Would need two models: encoder and decoder.
- Encoder would output a hidden state htE which is the last hidden state output.
- This state would be the first input to the decoder h1D.
- The same concept can be applied to transformers.

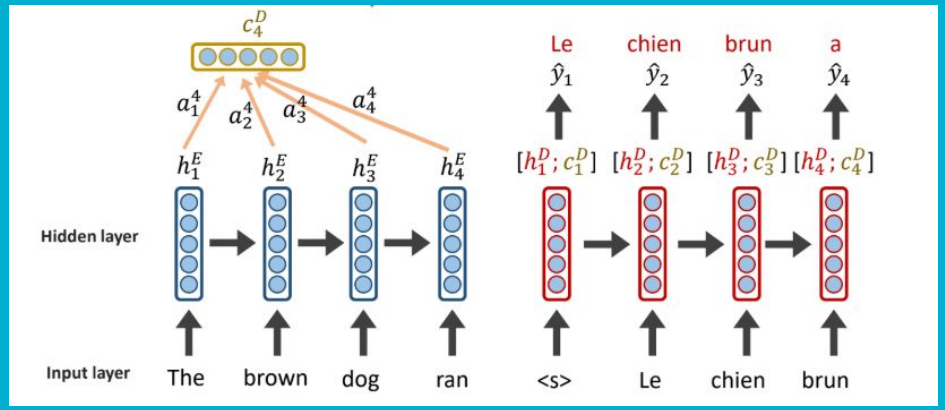
Seq2Seq RNN


$\hat{y}_1$ $\hat{y}_2$ $\hat{y}_3$ $\hat{y}_4$ $\hat{y}_5$ $\hat{y}_6$
h_1^E h_2^E h_3^E h_4^E h_1^D h_2^D h_3^D h_4^D h_5^D h_6^D
Hidden layer
Input layer
The brown dog ran <s> Le chien brun a couru
ENCODER RNN
DECODER RNN

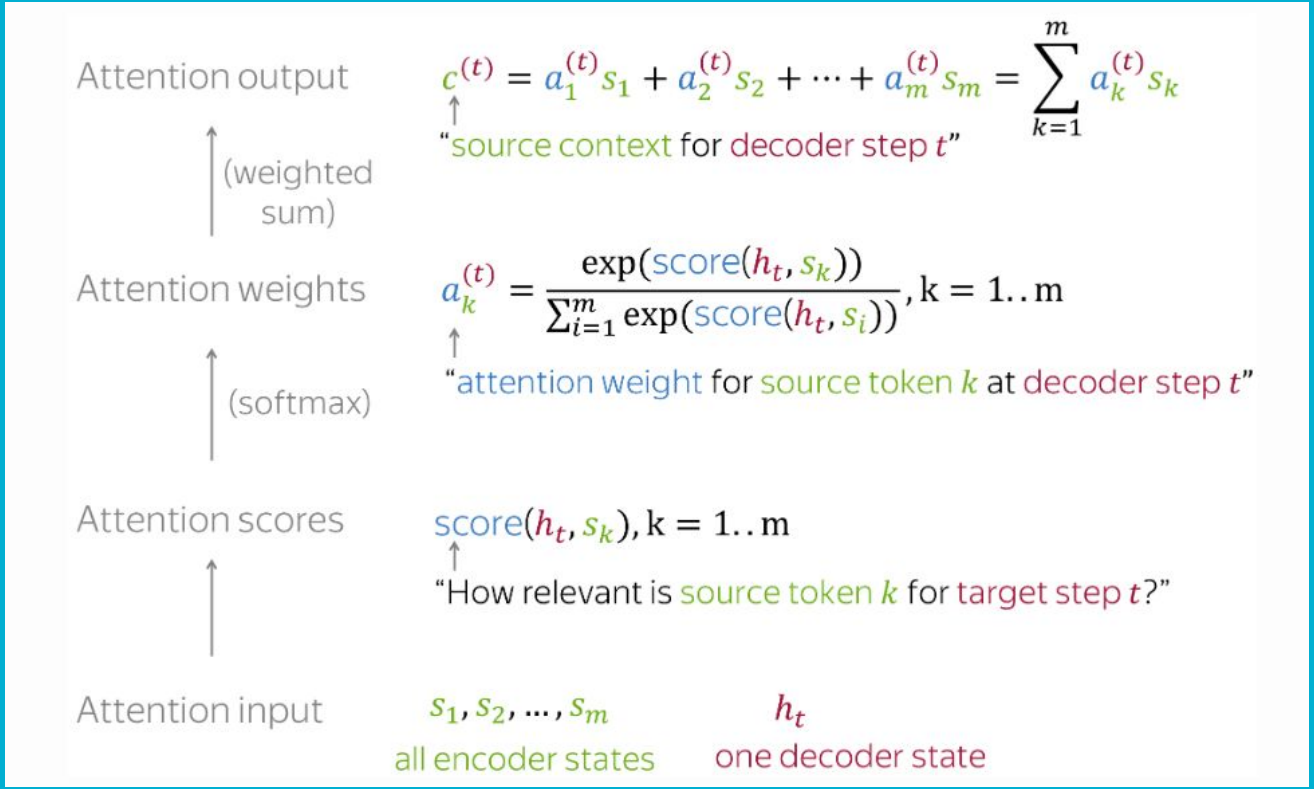
Attention

- Determines how much focus should the decoder pay to each hidden state from the encoder.
- Can be calculated by multiple ways, even a feed forward layer is sufficient.





Attention calculation



https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html