Attention

Motivating Attention for Translation

If you were translating a paragraph of text, how would you do it?

- **Memorise the paragraph and then translate**

or

- **Translate segments of the paragraph as you read it**

Motivating Attention for Translation

If you were translating a paragraph of text, how would you do it?

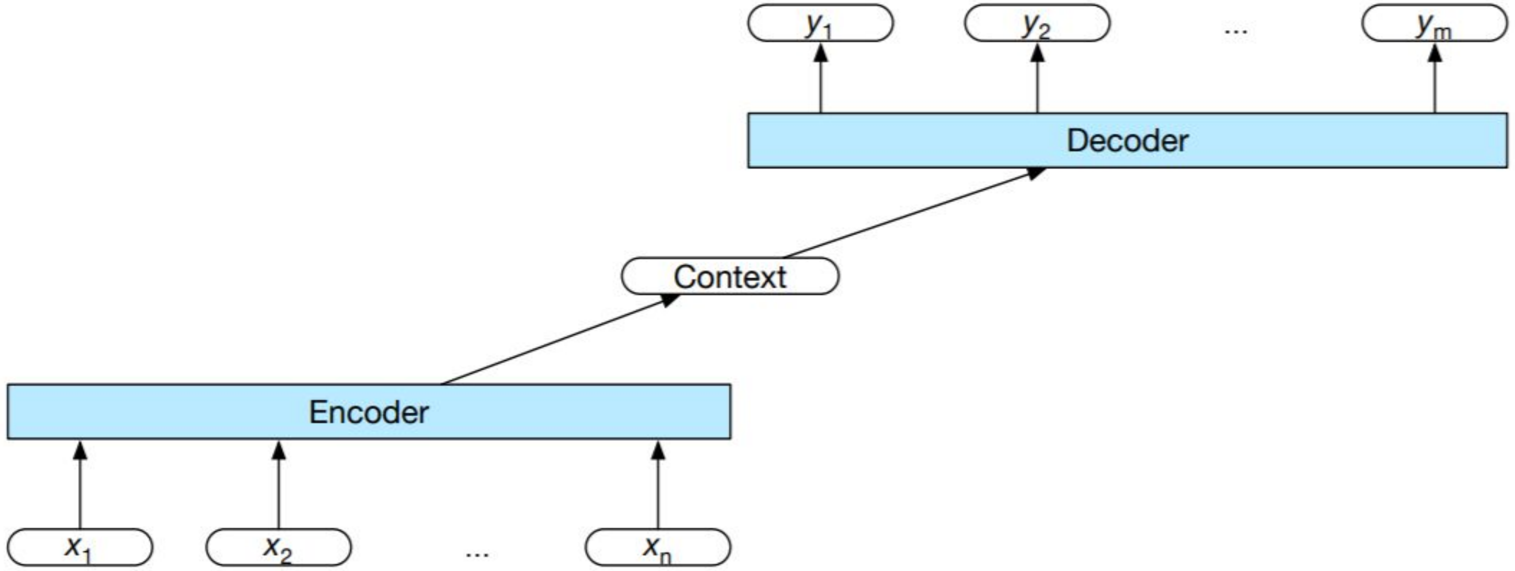
- **Memorise the paragraph and then translate**

or

- **Translate segments of the paragraph as you read it**

RNN Translation

How does a regular RNN/LSTM do it?



Focusing RNN/LSTM Attention

- For longer sentences RNNs/LSTMs have worse performance as it is difficult for the model to remember everything through the encoded vector.

- Want to feed the context vector (encoded vector) directly for each translated word prediction.

- Should the context vector be the same for each translated word prediction?

Relevant Context

Ideally the context vector should contain information about words that are relevant to the output word.

Example:

The farmer wrote that the sheep lives in a pen

$$\downarrow$$

Le fermier a écrit que le mouton habite à un _____

Relevant Context

What would happen if the model focused on this part of the sentence?

$\boxed{\textit{The farmer wrote}}$ *that the sheep lives in a pen*

$\downarrow$

Le fermier a écrit que le mouton habite à un _____

Relevant Context

Might get this output, that the sheep lives in a writing pen

$\boxed{\textit{The farmer wrote}}$ *that the sheep lives in a pen*

$\downarrow$

Le fermier a écrit que le mouton habite à un stylo

Relevant Context

What would happen if the model focused on this part of the sentence?

The farmer wrote that $\boxed{\textit{the sheep lives in}}$ *a pen*

$$\downarrow$$

Le fermier a écrit que le mouton habite à un _____

Relevant Context

Might get this (correct) output, that the sheep lives in an animal pen

The farmer wrote that $\boxed{\textit{the sheep lives in}}$ *a pen*

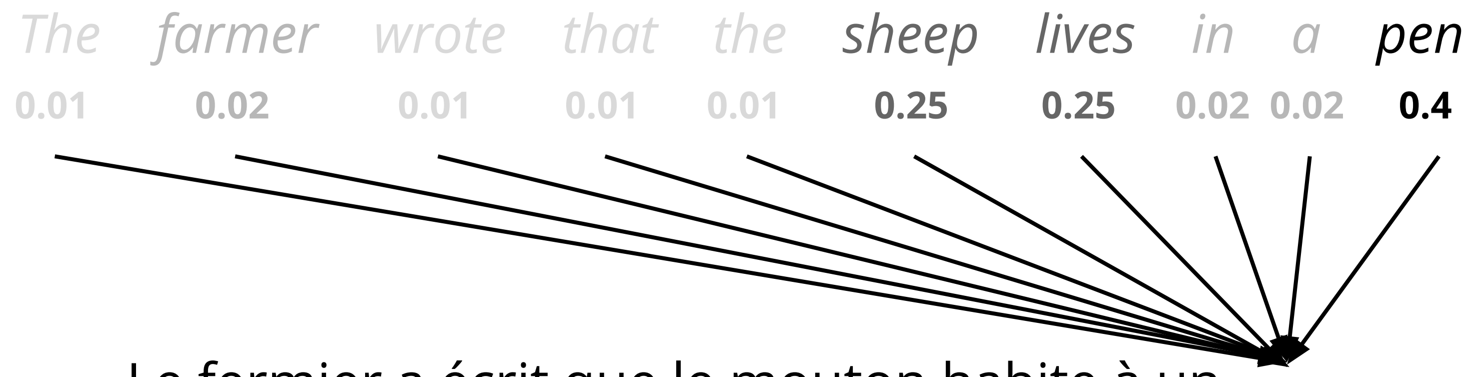
$\downarrow$

Le fermier a écrit que le mouton habite à un parc

Attention

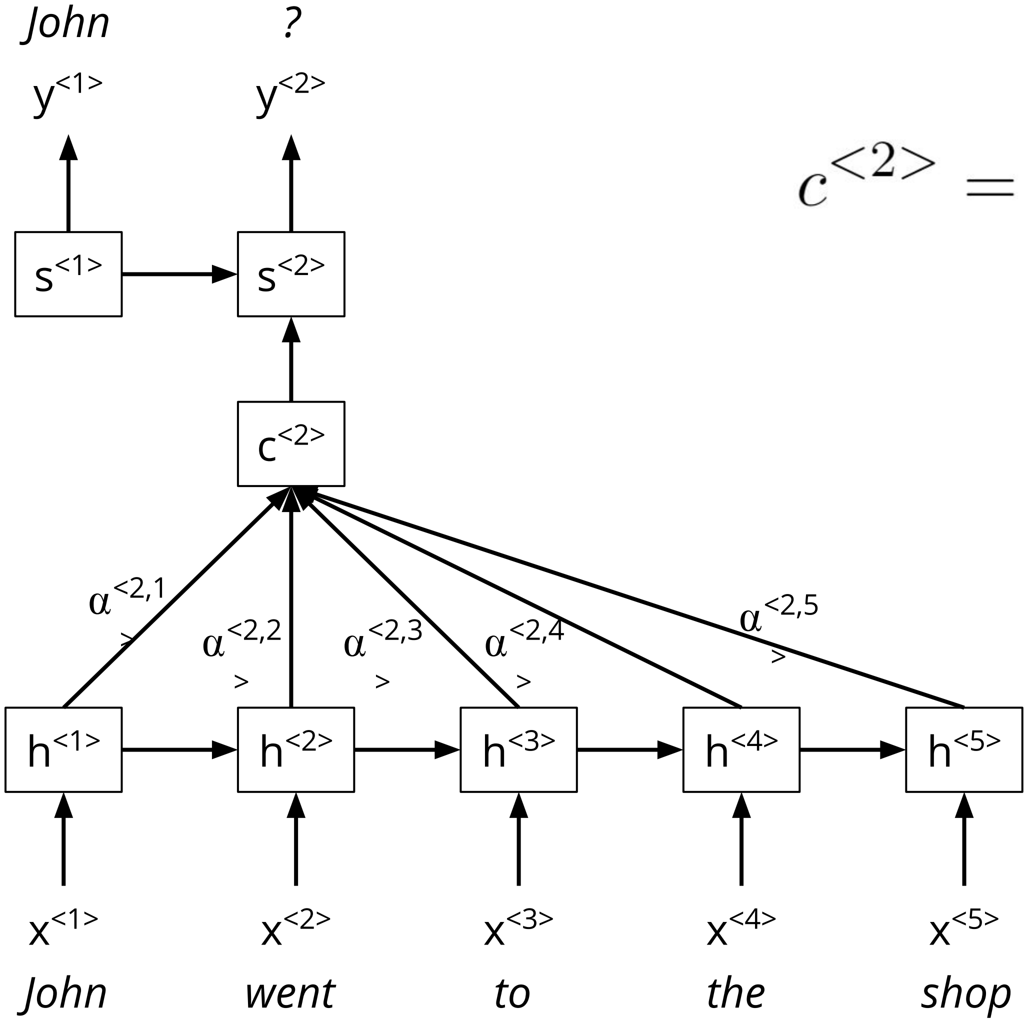
Attention is a way of training an RNN/LSTM to focus on relevant parts of the original input for performing a translation.

Attention involves creating a vector of weights which indicates how much of an influence each word in the original sentence has on an output word.



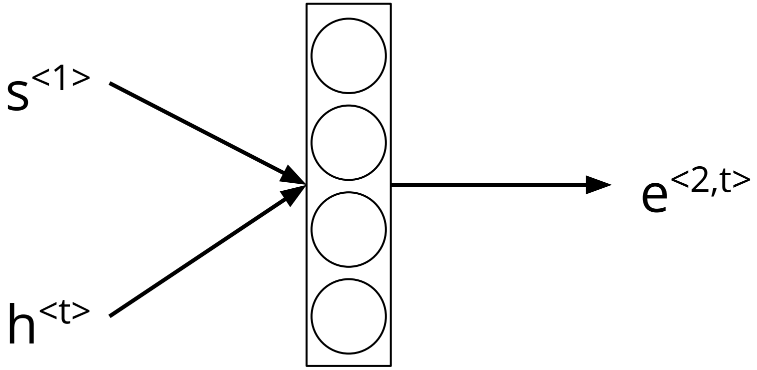
The	*farmer*	*wrote*	*that*	*the*	*sheep*	*lives*	*in*	*a*	*pen*
0.01	0.02	0.01	0.01	0.01	0.25	0.25	0.02	0.02	0.4

Le fermier a écrit que le mouton habite à un _____



$$c^{<2>} = \sum_t \alpha^{<2,t>} h^{<t>}$$

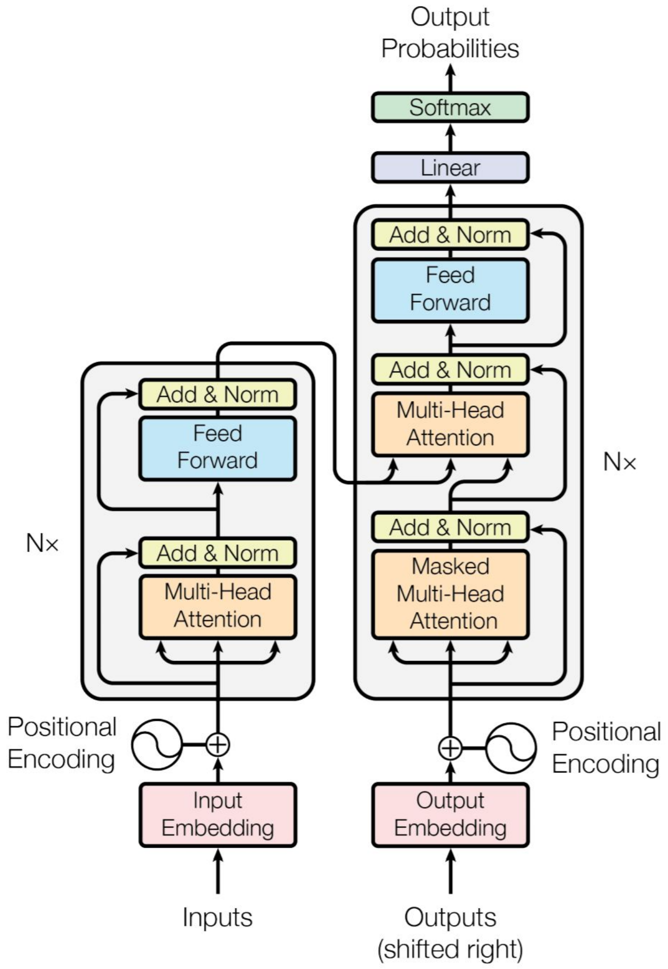
Determining Attention Weights



Previous time step's hidden state $s^{<1>}$ and the input word's embedding $h^{<t>}$ are given to a small neural network to produce the attention weight

$$\alpha^{<2,t>} = \frac{\exp(e^{<2,t>})}{\sum_{t'=1}^{T} \exp(e^{<2,t'>})}$$

The attention weight is then normalised using the softmax function


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)