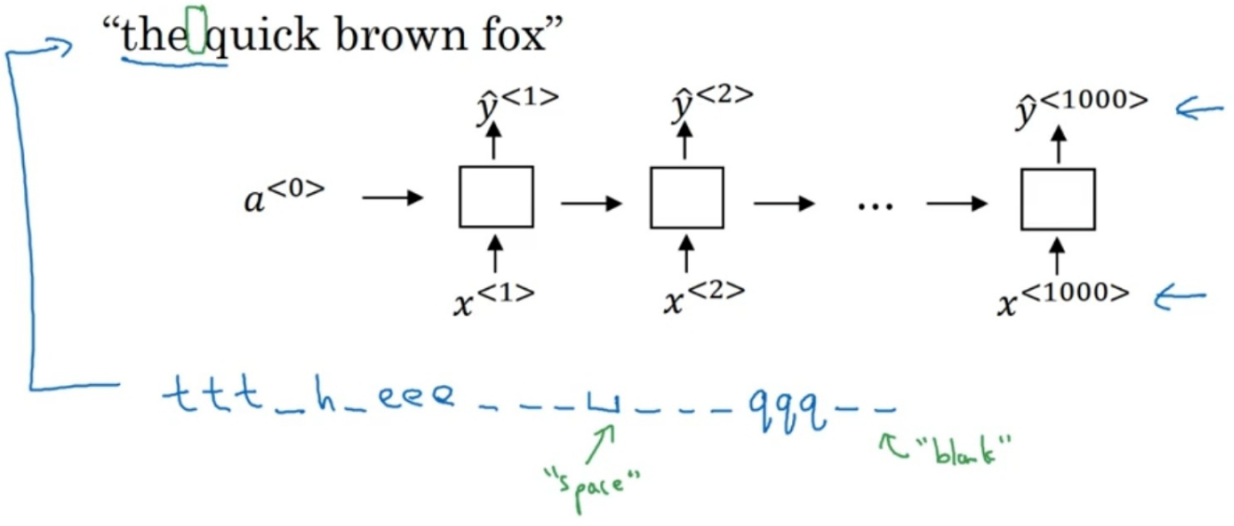
Speech recognition:

CTC cost for speech recognition

(Connectionist temporal classification)



Basic rule: collapse repeated characters not separated by "blank",

⇓

the q (...)

2 - Neural Machine Translation with Attention

- If you had to translate a book's paragraph from French to English, you would not read the whole paragraph, then close the book and translate.
- Even during the translation process, you would read/re-read and focus on the parts of the French paragraph corresponding to the parts of the English you are writing down.
- The attention mechanism tells a Neural Machine Translation model where it should pay attention to at any step.

2.1 - Attention Mechanism

In this part, you will implement the attention mechanism presented in the lecture videos.

- Here is a figure to remind you how the model works.
  - The diagram on the left shows the attention model.
  - The diagram on the right shows what one "attention" step does to calculate the attention variables $\alpha^{\langle t,t' \rangle}$.
  - The attention variables $\alpha^{\langle t,t' \rangle}$ are used to compute the context variable $context^{\langle t \rangle}$ for each timestep in the output ($t = 1, \ldots, T_y$).

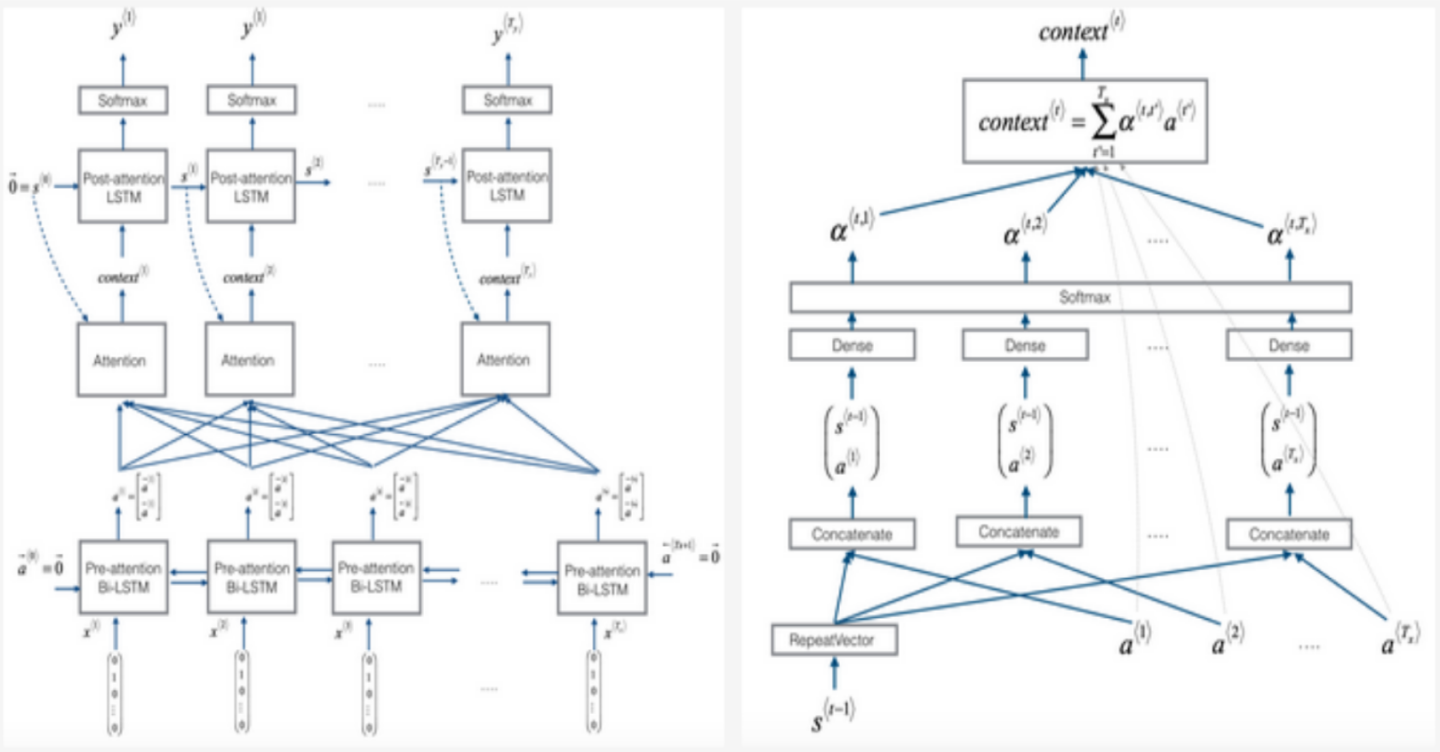


Figure 1: Neural machine translation with attention

Here are some properties of the model that you may notice:

Pre-attention and Post-attention LSTMs on both sides of the attention mechanism

- There are two separate LSTMs in this model (see diagram on the left): pre-attention and post-attention LSTMs.
- *Pre-attention* Bi-LSTM is the one at the bottom of the picture is a Bi-directional LSTM and comes *before* the attention mechanism.
  - The attention mechanism is shown in the middle of the left-hand diagram.
  - The pre-attention Bi-LSTM goes through T_x time steps
- *Post-attention* LSTM: at the top of the diagram comes *after* the attention mechanism.
  - The post-attention LSTM goes through T_y time steps.
- The post-attention LSTM passes the hidden state $s^{\langle t \rangle}$ and cell state $c^{\langle t \rangle}$ from one time step to the next.

An LSTM has both a hidden state and cell state

- In the lecture videos, we were using only a basic RNN for the post-attention sequence model
  - This means that the state captured by the RNN was outputting only the hidden state $s^{\langle t \rangle}$.
- In this assignment, we are using an LSTM instead of a basic RNN.
  - So the LSTM has both the hidden state $s^{\langle t \rangle}$ and the cell state $c^{\langle t \rangle}$.

Each time step does not use predictions from the previous time step

- Unlike previous text generation examples earlier in the course, in this model, the post-attention LSTM at time t does not take the previous time step's prediction $y^{\langle t-1 \rangle}$ as input.
- The post-attention LSTM at time 't' only takes the hidden state $s^{\langle t \rangle}$ and cell state $c^{\langle t \rangle}$ as input.
- We have designed the model this way because unlike language generation (where adjacent characters are highly correlated) there isn't as strong a dependency between the previous character and the next character in a YYYY-MM-DD date.

Concatenation of hidden states from the forward and backward pre-attention LSTMs

- $\overrightarrow{a}^{\langle t \rangle}$: hidden state of the forward-direction, pre-attention LSTM.
- $\overleftarrow{a}^{\langle t \rangle}$: hidden state of the backward-direction, pre-attention LSTM.
- $a^{\langle t \rangle} = [\overrightarrow{a}^{\langle t \rangle}, \overleftarrow{a}^{\langle t \rangle}]$: the concatenation of the activations of both the forward-direction $\overrightarrow{a}^{\langle t \rangle}$ and backward-directions $\overleftarrow{a}^{\langle t \rangle}$ of the pre-attention Bi-LSTM.

Computing "energies" $e^{\langle t,t' \rangle}$ as a function of $s^{\langle t-1 \rangle}$ and $a^{\langle t' \rangle}$

- Recall in the lesson videos "Attention Model", at time 6:45 to 8:16, the definition of "e" as a function of $s^{\langle t-1 \rangle}$ and $a^{\langle t \rangle}$.
  - "e" is called the "energies" variable.
  - $s^{\langle t-1 \rangle}$ is the hidden state of the post-attention LSTM
  - $a^{\langle t' \rangle}$ is the hidden state of the pre-attention LSTM.
  - $s^{\langle t-1 \rangle}$ and $a^{\langle t \rangle}$ are fed into a simple neural network, which learns the function to output $e^{\langle t,t' \rangle}$.
  - $e^{\langle t,t' \rangle}$ is then used when computing the attention $a^{\langle t,t' \rangle}$ that $y^{\langle t \rangle}$ should pay to $a^{\langle t' \rangle}$.

- The diagram on the right of figure 1 uses a `RepeatVector` node to copy $s^{\langle t-1 \rangle}$'s value T_x times.
- Then it uses `Concatenation` to concatenate $s^{\langle t-1 \rangle}$ and $a^{\langle t \rangle}$.
- The concatenation of $s^{\langle t-1 \rangle}$ and $a^{\langle t \rangle}$ is fed into a "Dense" layer, which computes $e^{\langle t,t' \rangle}$.
- $e^{\langle t,t' \rangle}$ is then passed through a softmax to compute $\alpha^{\langle t,t' \rangle}$.
- Note that the diagram doesn't explicitly show variable $e^{\langle t,t' \rangle}$, but $e^{\langle t,t' \rangle}$ is above the Dense layer and below the Softmax layer in the diagram in the right half of figure 1.
- We'll explain how to use `RepeatVector` and `Concatenation` in Keras below.

Implementation Details

Let's implement this neural translator. You will start by implementing two functions: `one_step_attention()` and `model()`.

one_step_attention

- The inputs to the one_step_attention at time step t are:
  - $[a^{<1>}, a^{<2>}, \ldots, a^{<T_x>}]$: all hidden states of the pre-attention Bi-LSTM.
  - $s^{<t-1>}$: the previous hidden state of the post-attention LSTM
- one_step_attention computes:
  - $[\alpha^{<t,1>}, \alpha^{<t,2>}, \ldots, \alpha^{<t,T_x>}]$: the attention weights
  - $context^{\langle t \rangle}$: the context vector:

$$context^{<t>} = \sum_{t'=1}^{T_x} \alpha^{<t,t'>} a^{<t'>}$$

Clarifying 'context' and 'c'

- In the lecture videos, the context was denoted $c^{\langle t \rangle}$
- In the assignment, we are calling the context $context^{\langle t \rangle}$.
  - This is to avoid confusion with the post-attention LSTM's internal memory cell variable, which is also denoted $c^{\langle t \rangle}$.