Attention Models:

The attention mechanism was developed first to improve the performance of machine translation task in an encoder-decoder network. At every segment of the prediction, this mechanism allows the algorithm to identify and base the prediction on the most relevant parts of the input vectors by assigning weights to each input vector.

The encoder-decoder network introduced in **Figure 1** has a profound weakness and that is that all of the information in the input of the encoder network is given in a fixed-length vector as the initial hidden layer to the decoder network. This means that no matter how complex or simple the input of the encoder network is, its output representation is still a fixed-length vector which might not suffice to carry all the required information to the decoder network to make accurate predictions. To tackle this problem, (Bahdanau et al., 2014) proposed an encoder-decoder network equipped with an attention mechanism [3].

Let us refer to the sequences in the encoder network with t' and in the decoder network with time $t.$ Note that $t' = 1,2,\dots,T_x$ and $t = 1,2,\dots,T_y.$ The encoder network outputs two parameters at each time t'; one is the encoded vector, $a^{<t'>}$, representing the input $x^{<t'>}$ and the other is an attention parameter, $\alpha^{<t,t'>}$, which weighs the encoded vector $a^{<t'>}$ based on its relevance to make a prediction at time t. A context vector, $c^{<t>}$, is an input to the decoder network such that:

$$c^{<t>} = \sum_{t'=1}^{T_X} \alpha^{<t,t'>} \, a^{<t'>}$$

Where $\alpha^{<t,t'>}$ is the attention parameter corresponding to t'th word in the input to predict the tth word of the output sentence and $a^{<t'>}$ is the encoded vector representing the input $x^{<t'>}$.

To determine the weights of the attention parameters, we run a Softmax function as the following:

$$\alpha^{<t,t'>} = \frac{e^{\,u<t,t'>}}{\sum_{t'=1}^{T_x} e^{\,u<t,t'>}}$$

Note that the sum of the attention weights inputted to make a prediction at time t should equal to 1 due to using the Softmax function. $u < t, t' >$, called energies or alignment score, is a feedforward neural network which takes the hidden layer of the decoder network at the previous time step, $S^{<t-1>}$, and the output of the encoder network at time step $t', a^{<t'>}$, to output the energies or alignment scores for the encoded vector $a^{<t'>}$.

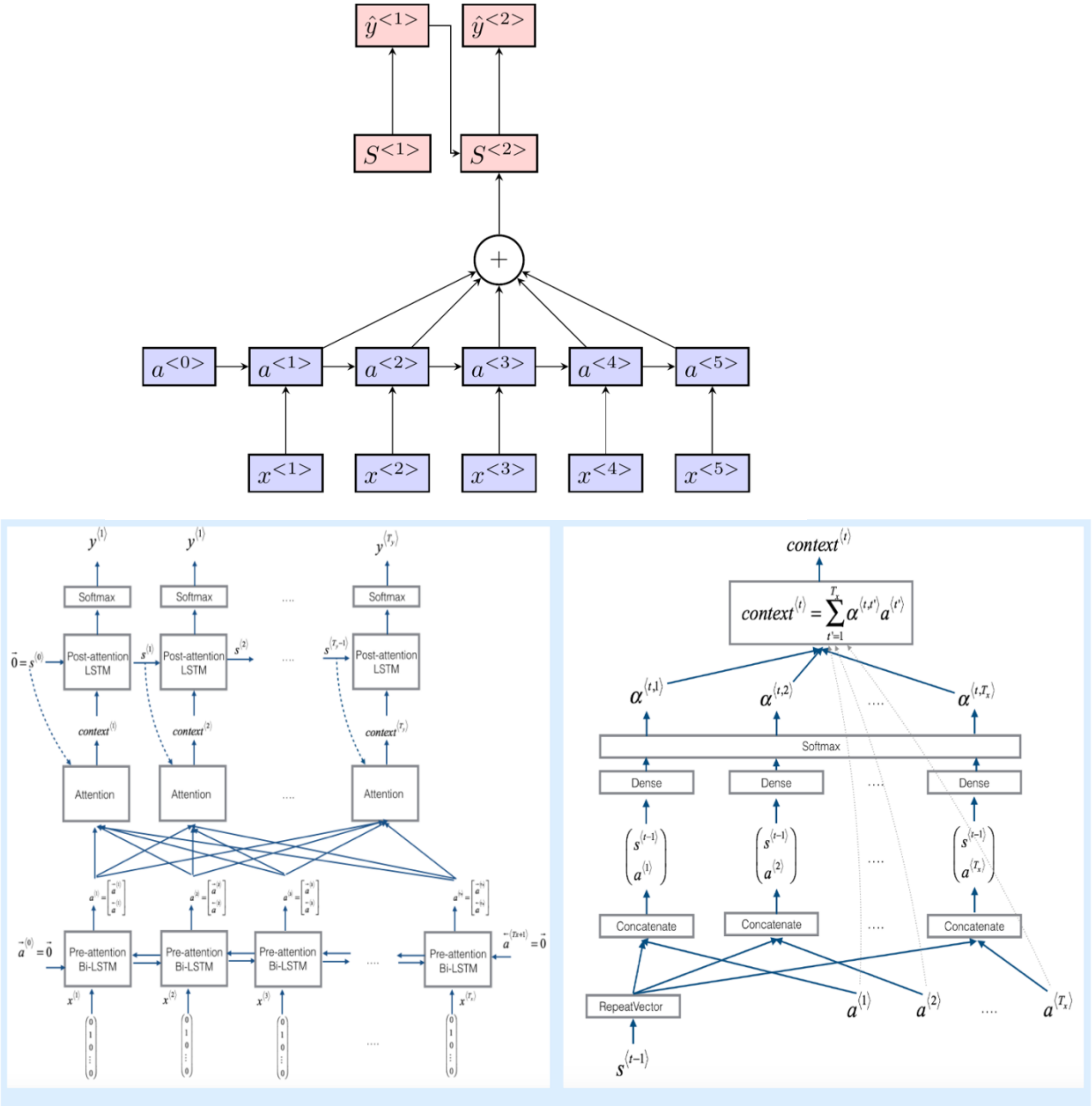
$\hat{y}^{<1>}$
$\hat{y}^{<2>}$
$S^{<1>}$
$S^{<2>}$
+
$a^{<0>}$
$a^{<1>}$
$a^{<2>}$
$a^{<3>}$
$a^{<4>}$
$a^{<5>}$
$x^{<1>}$
$x^{<2>}$
$x^{<3>}$
$x^{<4>}$
$x^{<5>}$
$y^{\langle 1 \rangle}$
$y^{\langle 1 \rangle}$
$y^{\langle T_y \rangle}$
Softmax
Post-attention LSTM
$\vec{0} = s^{\langle 0 \rangle}$
$s^{\langle 1 \rangle}$
$s^{\langle 2 \rangle}$
$s^{\langle T_y - 1 \rangle}$
$context^{\langle 1 \rangle}$
$context^{\langle 2 \rangle}$
$context^{\langle T_y \rangle}$
Attention
Pre-attention Bi-LSTM
$\vec{a}^{\langle 0 \rangle} = \vec{0}$
$\overleftarrow{a}^{\langle T_x + 1 \rangle} = \vec{0}$
$x^{\langle 1 \rangle}$
$x^{\langle 2 \rangle}$
$x^{\langle 3 \rangle}$
$x^{\langle T_x \rangle}$
....
$context^{\langle t \rangle}$
$context^{\langle t \rangle} = \sum_{t'=1}^{T_x} \alpha^{\langle t,t' \rangle} a^{\langle t' \rangle}$
$\alpha^{\langle t,1 \rangle}$
$\alpha^{\langle t,2 \rangle}$
$\alpha^{\langle t,T_x \rangle}$
Softmax
Dense
$s^{\langle t-1 \rangle}$
$a^{\langle 1 \rangle}$
$a^{\langle 2 \rangle}$
$a^{\langle T_x \rangle}$
Concatenate
RepeatVector
$s^{\langle t-1 \rangle}$

Figure 2. Recurrent Neural Network using Encoder – Decoder Structure and Attention model. Diagram is created using Latex. The encoder and decoder network are respectively shown with colors purple and red. https://www.coursera.org/learn/nlp-sequence-models/programming/L0BBe/neural-machine-translation/lab?path=%2Fnotebooks%2FW3A1%2FNeural_machine_translation_with_attention_v4a.ipynb