A Quick Summary:
Sequence to Sequence Learning with Neural Networks

Original Paper: https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

23rd Feb 2019

1 Ideas:

(a) Using a multilayered deep LSTM to encode an input sentence, and using another LSTM to decode the encoding. This method was used for language translation.

2 Explanations:

(a) There are two components to the model:

i Encoder: The encoder takes in an input sentence such as “I have a dog” and encodes it using a RNN (in this case, the authors used LSTM cells). The encoding stops when an <EOS> (end of sentence) symbol is reached. Think of this symbol as just another “word” in the sentence.

ii Decoder: If we are working with English to Spanish translation (assuming the model has been trained), then the first LSTM cell of the decoder should output ”Tengo”, then this word will be fed into the next decoder cell, which takes as input the word ”Tengo” and the hidden representation from the first decoder cell, to output ”un”, and so on. Similarly, the decoding terminates when an <EOS> symbol is reached.

The advantage of this model is that it can output translations of variable length, e.g. the model can process an English sentence which is 3 words long to a Spanish sentence which is 5 words long - there are no theoretical limitations on the input/output lengths (aside from practical issues such as exploding/vanishing gradients).

3 Model:

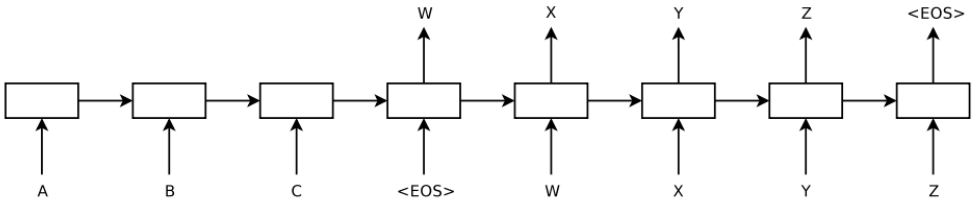

Figure 1: How the attention mechanism works (image is from the original paper).

4 Results:

(a) Slightly better than the (then) state of the art baseline system.

5 Notes:

(a) If I wanted to translate a paragraph, must the number of output sentences in be the same as that of the input? If not, what can we do to ensure this variability? Add a <EOP> (End of Paragraph) token?