Lost in Translation

9. (10 points) We want to make an RNN to translate English to Martian. We have a training set of pairs $(e^{(i)}, m^{(i)})$, where $e^{(i)}$ is a sequence of length $J^{(i)}$ of English words and $m^{(i)}$ is a sequence of length $K^{(i)}$ of Martian words. The sequences, even within a pair, do not need to be of the same length, i.e., $J^{(i)}$ need not equal $K^{(i)}$. We are considering two different strategies for turning this into a transduction or sequence-to-sequence learning problem for an RNN.

Method 1: Construct a training-sequence pair (x, y) from an example (e, m) by letting

$$x = (e_1, e_2, \ldots, e_L, \mathit{stop})$$
$$y = (m_1, m_2, \ldots, m_L, \mathit{stop})$$

In Method 1, we assume that if the original e and m had different numbers of words, then the shorter sentence is padded with enough time-wasting words ("ummm" for English, "grlork" for Martian) so that they now have equal length, L. Any needed padding words are inserted at the end of $e^{(i)}$, and at the start of $m^{(i)}$.

Method 2: Construct a training-sequence pair (x, y) from an example (e, m) by letting

$$x = (e_1, e_2, \ldots, e_J, \mathit{stop}, \mathit{blank}, \ldots, \mathit{blank})$$
$$y = (\mathit{blank}, \ldots, \mathit{blank}, m_1, m_2, \ldots, m_K, \mathit{stop})$$

In Method 2, blanks are inserted at the end of e and start of m such that the length of x and y are now both $J + K + 1$.

(a) Assume an element-wise loss function $L_{elt}(p, y)$ on predicted versus true Martian words. What is an appropriate sequence loss function for **Method 1**? Assume that the predicted sequence p has the same length as the target sequence y.

Solution:

$$L_{seq} = \sum_{i=1}^{L+1} L_{elt}(p_i, y_i)$$

The RNN should seek to output the correct Martian words, as well as the *stop* indicator.

(b) Assume an element-wise loss function $L_{elt}(p, y)$ on predicted versus true Martian words. What is an appropriate sequence loss function for **Method 2**? Assume the predicted sequence p has the same length as the target sequence y.

Solution:

$$L_{seq} = \sum_{i=J+1}^{J+K+1} L_{elt}(p_i, y_i)$$

It's really only necessary that the RNN correctly outputs the whole Martian sequence and the final *stop* indicator. But, it's okay if you sum starting from the first token, $i = 1$.

(c) Which method is likely to need a higher dimensional state? Explain why.

Solution: Method 2 likely needs to have a larger state to hold a representation of the full input sentence e, while Method 1 might have a shorter state that enables mapping of individual words or shorter sub-sequences of words to corresponding output words or sub-sequences.

(d) Which method is better if English and Martian have very different word order? Explain why.

Solution: Method 2 since it can first parse the entire input sentence, and then output in a different word order.

(e) Martian linguist Grlymp thinks it is also important to pad the original English and Martian sentences with time-wasting word to be of the same length for Method 2 (i.e., so that $J = K$), but English linguist Chome Nimsky disagrees. Who is correct, and why?

Solution: Chome Nimsky is right: Method 2 already has full flexibility in processing the entire sentence e before outputting m, so additional time-wasting words would not help (and may hurt) in expressiveness and/or training.