Lab 7 Exercise - Transforming Sequences

Luke McClure
29573904

1 SEQUENCE-TO-SEQUENCE MODELLING

The source text must first be passed through the embedding layer that maps a word into a dense vector that the LSTM is able to use.

As the decoder only requires the hidden and cell state of the previous timestep, we can disregard the LSTM output that includes all previous timesteps. By splitting the hidden state tuple into the hidden state and cell state it is easier to pass back for use by the decoder network.

```
def forward(self, src):
    embedded = self.embedding(src)
    _, (h, c) = self.rnn(embedded)

    return h, c
```

The model shows a very clean training curve in Figure 1, with the test and training loss closely following each other.

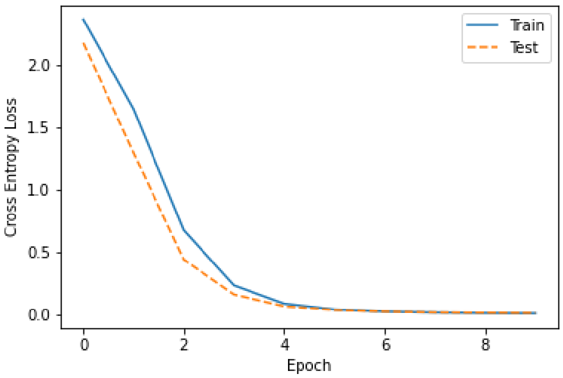

Figure 1: Training curve of seq2seq model

1.1 Complete and train a seq2seq model

1.2 now use it!

1.2.1 Why is the order of the output reversed? The output is reversed as the model is trained to predict the output in reverse. Due to the difficulty LSTM's have in learning long range dependancies, reversing the output means that the end characters of the input sequence will be beside the corresponding correctors in the target sequence. This shortens the dependencies the LSTM needs to learn at first in order to build up the translation model.

1.2.2 What is the point of teacher forcing? During normal output the RNN would use the previous character to predict the character at timestep t, so if the model were to predict the character incorrectly this would carry through for the rest of the sentence. Teacher forcing provides the model with the correct output before prediction of the next character during training. This means that instead of trying to build off an incorrect prediction the model is given another opportunity to correctly learn, speeding up backpropogation and therefore speed up convergence.

1.3 Sequence lengths

The first string in subsection 1.2 was chunked to factors of it's original length, these factors were 1, 3, 23 & 69.

1 et te eee ett e ete t eeee e eete ttt etee etee ttt ett ee te tte
3 t amenm̂st a tet aee e tm̂seeeet ôumett môreeeêdt m tet m iemenn̂g
23 ^^^^^^^^^^^^^ĝavert ^^^^^^^^^^^^^^^ĥe fol^^^^^^^^^^^^^^^l̂owing
69 ^^^ e lovelwing

The model starts to break down with longer chunks. While single letters only produces outputs consisting of the most common letters in the English language, chunks of half or the length of the sentence start to only get predicted as the starting character. With a mean input length of 12.4 and target length of 3.5 this is not too surprising, the training data consists of several character slices long of text so chunks closely matching that training size will perform better.