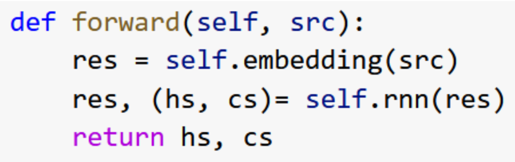
COMP6248 Lab 7 Exercise

Adrian Azzareli, aab1g18@soton.ac.uk

1 Sequence-to-sequence Modelling

The completed function if displayed in Figure 1. The LTSM layer method provided by *Pytorch* allows us to call the hidden and cell states directly.



```
def forward(self, src):
    res = self.embedding(src)
    res, (hs, cs)= self.rnn(res)
    return hs, cs
```

Fig. 1. Snapshot of *forward* method for the LTSM module

Consequently, we were able to plot loss-curves of the training provided in the incomplete code, Figure 2

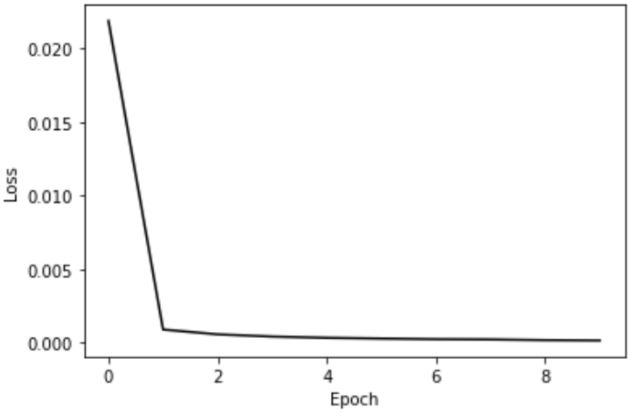


Fig. 2. Loss plots for training the sequence-to-sequence model

We are then able to decode the message as:

> *answer the following*
>
> ** why is the order of the output reversed*
>
> ** what is the point of teacher forcing*

Answering this: (1) Reversing the output is a technique developed in *Ilya Sutskever et Al (2014)*[1] which can be used to introduce "short-term dependencies between the source and target sequence which made the optimization problem easier". Hence, if inputs are reversed then so will our outputs. (2) *Teacher forcing* is a strategy to train networks on group truths as opposed to direct results from prior time step. Thus, the input at $t+1$ is a result of the group truth rather than generated value from step t. The expectation is that by forcing the network to learn the sequence of ground truth we can address problems such as slow convergence or instability.

Finally, chunks were varied from lengths 1 to 7 (each chunk is a collection of sequential codes separated by $n-1$ spaces. Therefore, chunks size $n=2$ consist of parts of encoding separated by 1 space). We found that past 5 the results proved errorsome, which we understand is a result of the default maximum length for training being 5.

[1] *Sequence to Sequence Learning with Neural Networks*, Ilya Sutskever et Al, 2014