

## Lab 7 Exercise - Transforming Sequences

Amishapriya Singh  
Student id: 33333904  
as6u21@soton.ac.uk

## 1. Sequence-to-Sequence Modelling

### 1.1. Complete and train a sequence-to-sequence model

```
def forward(self, src):
    embedding = self.embedding(src)
    output, (hidden_state, cell_state) = self.rnn(embedding)
    return hidden_state, cell_state
```

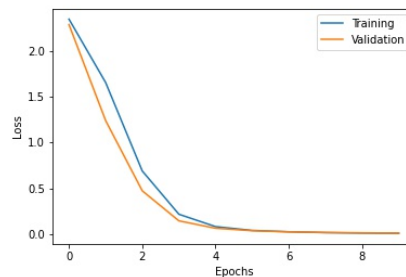


Figure 1. Training loss curve

## 1.2. Now use it!

### 1.2.1 Why is the order of the output reversed?

The idea comes from *Sequence to Sequence Learning with Neural Networks*.<sup>1</sup> According to the paper, Seq2Seq models give better performance when the input is reversed due to “stronger communication link” wherein the first few words of the source might be closer to the first few words of the target. Reversing the output essentially achieves the same result as reversing the input before feeding into the model.

### 1.2.2 What is the point of teacher forcing?

Teacher forcing is used for training RNNs using ground truth as input, instead of model output from a prior time step. It uses the actual or expected output from the training dataset at the current time step  $y(t)$  as input in the next time step  $X(t+1)$ , rather than the output generated by the network. If we use predicted output as input in the next step, it may lead to a huge loss resulting model settling in local maxima or slow convergence. Although teacher forcing allows faster training, we use a ratio to control its extent as an extensive amount of teacher forcing might cause small prediction compound in the conditioning context.<sup>2</sup>

### 1.3. Sequence Lengths

```
def decode_long_chunks(code):
    out = ''
    tokens = code.split(' ')
    long_chunked_code = [tokens[i] + ' ' + tokens[i+1] for i in range(0, len(tokens)-2, 2)]
    for chunk in long_chunked_code:
        num = ds.encode_morse('^ ' + chunk + ' $').unsqueeze(1)
        pred = model(num.cuda(), maxlen=2)
        pred = pred[1:].view(-1, pred_dim).argmax(-1)
        out += ds.decode_alpha(pred.cpu())[:-1]
    return out
```

When given the encoded input ‘-.- .... -.- / .. ... / - .... / — .-. -. .-. / — .-. / -.... / — .-. -. -. -. -. / .-. . .... .-. ... .-.’, the decoder output is *h steodro eotu ees*. Clearly, for long chunks, the model fails to capture the correct sequence. The maximum length of sequence in training data is 6 which could be a possible reason why the seq2seq model does not work well on long chunks.

<sup>1</sup>Ilya Sutskever, Oriol Vinyals, and Quoc V Le. “Sequence to Sequence Learning with Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014.

<sup>2</sup>Alex Lamb et al. *Professor Forcing: A New Algorithm for Training Recurrent Networks*. 2016.