A Basic Encoder-Decoder Model

Amitai Yacobi

June 5, 2022

Architecture, Loss function and Optimizer

As for the architecture, I used unidirectional LSTM network for both Encoder and Decoder, and I used another simple feed forward network for the attention.
After some experiments with the network dimensions I found that the best results were for the following dimensions:

1. Embedding dimension of 128 for both Encoder and Decoder.
2. Hidden dimension of 256 for both Encoder and Decoder.

For the attention network I used simple network with input layer and output layer only. I tried to add hidden layers to this network but it didn't improve the results.
As for the loss function, I used the Negative Log Likelihood loss like we saw in the lectures. In practice, I summed the negative log likelihoods for each correct output token in the output sequence, as computed by the softmax function.
As for the optimizer, I tried, like in the previous part, different types of optimizers like RMSprop, Adam and AdaDelta and I found that Adam gave best results.

Hyper parameters

After some tuning of the hyper parameters I found that the following values gave me the best results:
Learning rate - 0.0005 for both Encoder and Decoder
Dropout - 0.1 for both Encoder, Decoder and attention network.

Results

At the beginning, I implemented the attention mechanism exactly how we saw in class and it did give better results (Bleu score of 87 on test set) then in the first part, but after I read the paper mentioned in the exercise, I saw that they caculated the score value little bit different, i.e., they added another learned vector v_a and they multiplied it by the value $tanh(W_a[h_t; \overline{h}_s])$. I added this small addition to the implementation and I got the following results:
Loss on the train set - 0.3344
Loss on the dev set - 1.3070
Bleu score on the dev set - 95.6036
Bleu score on the test set - 98.53

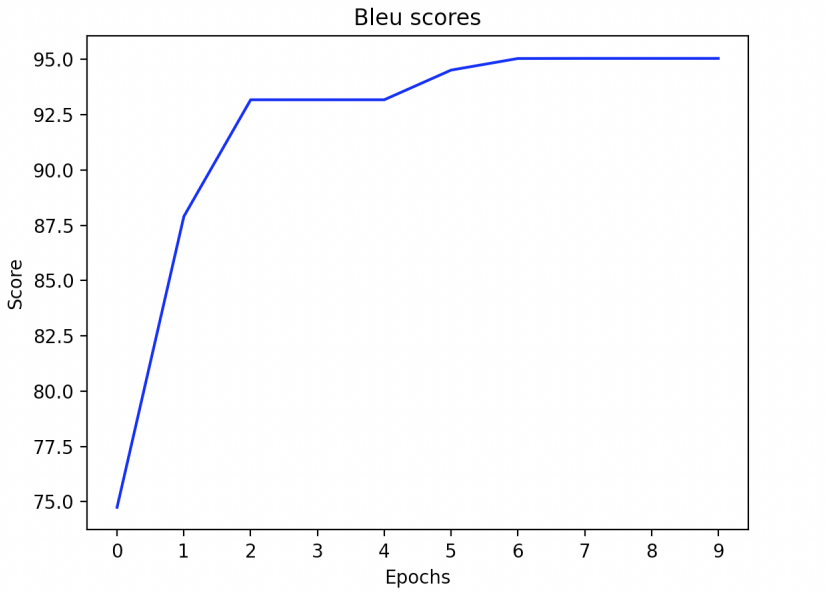

Figure 1: Bleu score on dev per epoch

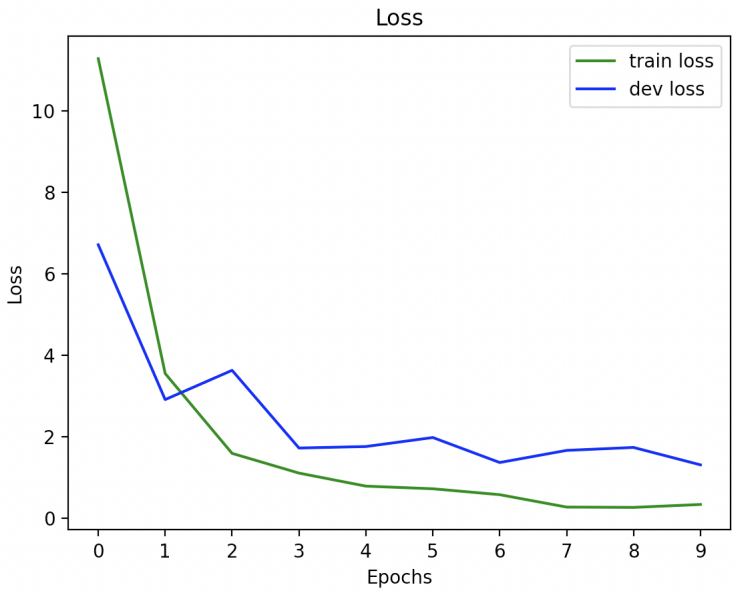

Figure 2: Loss on train and dev per epoch

As expected, the attention-based model gave much better results because of the fact that in this mechanism, the decoder can learn a different representation of the source sentence in every step.