P4 Translation

Name: 赵子涵
Major: Computer Science
Student ID: 2023E8013282148

Task

In this project, we are trying to design and train a transformer model for machine translation task. Specifically, we try to translate Chinese into English.

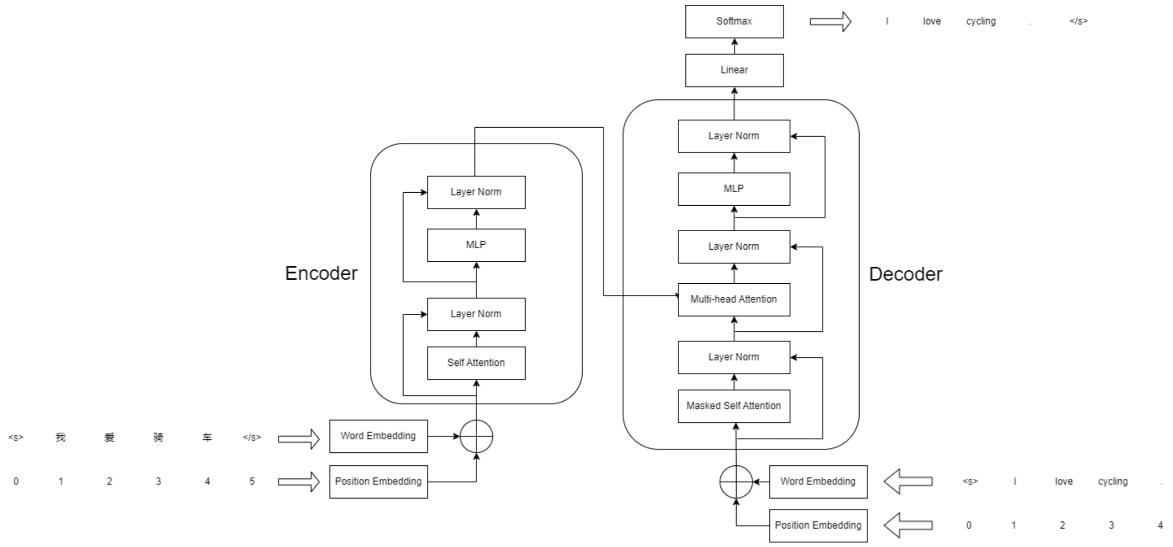
Dataset

We use the dataset from NiuTrans, which contains corresponding Chinese and English corpus of size 100k .

Model Design

In this section, we present the overall architecture of the model, and then discuss the issue of masking within the model, which confused me for quite a while.

The model architecture follows the classical transformer, consist of an encoder and a decoder. Each layer of encoder has one self attention layer and one MLP layer, and each layer of decoder has one self attention, one multi-head attention, and one MLP layer. Beyond that, we add word and position embedding layer for input, and an linear layer for output. See below for an illustration of our model.



Masking Issue

It is a little confusing for a newbie to understand the illustration, why we take the ground truth result as input and expect the same as output? Firstly, we have to understand what decoder actually do: predict the token from all previous tokens. To prevent the model cheating, we add a mask in the self attention layer. Specifically, suppose the sequence length is n, then the attention weight $C = softmax(QK)$ is an $n \times n$ matrix. To prevent the model looking ahead, we could just set the entries of upper triangular zero. Moreover, to make this process suitable by BP algorithm, we could do an addition instead.

$$C' = softmax(QK + U) \tag{1}$$

where the upper triangular entries of U is $-\infty$, and the rest is 0. This force the model to look at the previous value only.

Still, one may argue that, since we adopt residual connection, will the data stream bypass masked attention layer and let the full information accessible to the next layer? The answer is no, here is a little trick in this. As we train the decoder, we do a 'shift-right' operation on the target sequence, which means we expect the model to predict the $(k+1)$ th token on the k th position. Therefore, although the residual connection do bypass the masked attention layer, the k th position still could only access the information in $[0, k]$, and try to predict the $(k+1)$ th token.

Metric

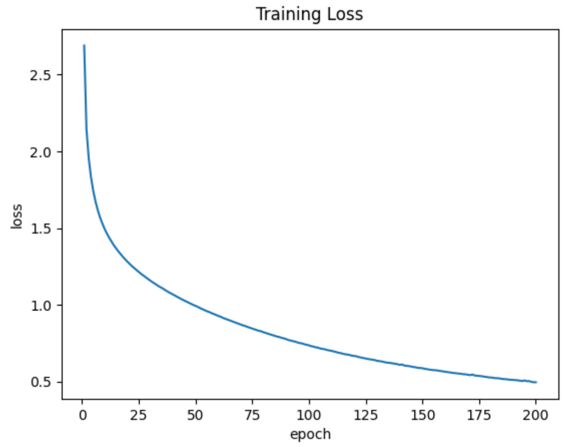
We use BLEU score to evaluate our model. To avoid any mistakes in implementing, we adopt the function by pytorch `torchtext.data.metrics.bleu_score()`.

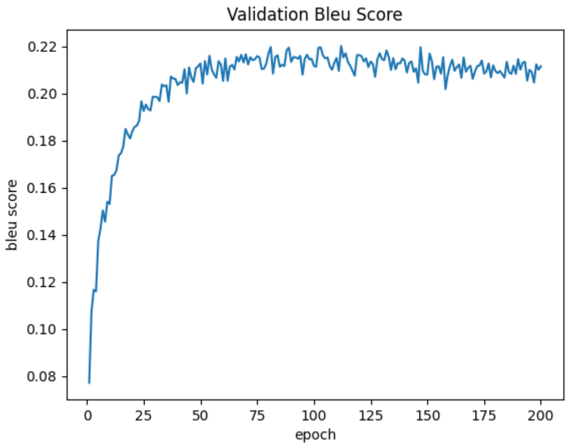
Result

After processing the data and implementing the above model, we set the hyper-parameter as

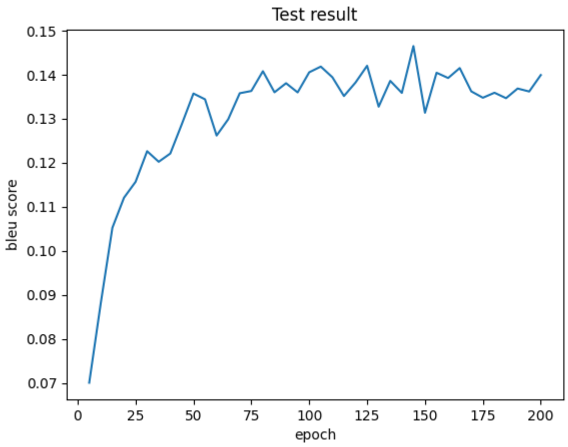
```
LR=0.001

h_dim=256
enc_n_heads=8
dec_n_heads=8
enc_ff_dim=256
dec_ff_dim=256
enc_n_layers=4
dec_n_layers=4
dropout=0.1
```

The train and valid result is as follows





After about 150 epochs, the BLEU score on test set reached `0.146`.



Note: It is worth to clarify that as we *test* the model, we send the whole Chinese text into the encoder, and send the previous-generated k-length English text into the decoder to predict the $(k+1)$ th word in each iteration. Unlike what we do in *valid* process, send the whole reference English text into the decoder to predict the right-shifted text, in which the model actually predict each word based on the previous ground truth.