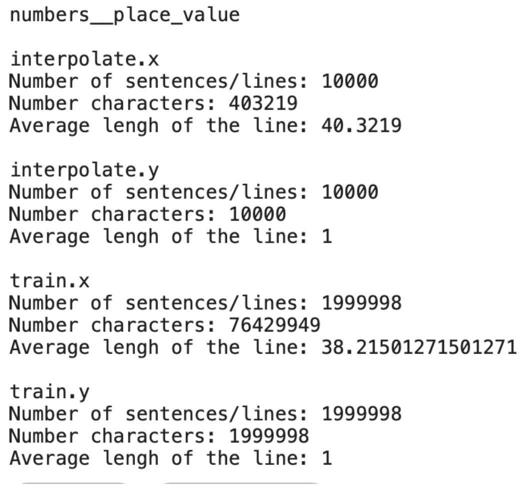
Deep Learning Assignment 4
Terekhin Daniil

1. Preliminaries and Reading Comprehension

1.1. The number of sentences and characters in the training and validation sets (Figure 1)



```
numbers__place_value

interpolate.x
Number of sentences/lines: 10000
Number characters: 403219
Average lengh of the line: 40.3219

interpolate.y
Number of sentences/lines: 10000
Number characters: 10000
Average lengh of the line: 1

train.x
Number of sentences/lines: 1999998
Number characters: 76429949
Average lengh of the line: 38.21501271501271

train.y
Number of sentences/lines: 1999998
Number characters: 1999998
Average lengh of the line: 1
```

(Figure 1)

2. Dataloader

2.1. It creates distinct vocabularies for the source and target sets.

2.2. The unk token, with the id of 1, is present in all vocabularies and serves as a placeholder for any unknown characters encountered during training and validation. When creating the vocabulary, any unknown characters are added, and the unk token is not taken into account

3. Model

3.1. I implemented function forward_separate that implements this second option that makes use of self.encoder and self.decoder instead of self.transformer.

4. Greedy search

4.1. I implemented function greedy_search

4.2. One issue I see with the implementation of nn.Transformer is the need for separate functions for calling the encoder and decoder, which require the creation of new masks each time. This process of creating new masks and transferring them to the device for each new letter predicted significantly reduces the performance of the search, especially as the length of the sequence increases.

4.3. I specified 2 stopping criteria: the model outputs the end-of-sentence token eos, when length of the output sequence exceeds that of the target sequence.

4.4. I implemented it in the greedy_search function

5. Accuracy computation

5.1. I implemented the function accuracy

6. Training

6.1. I chose to use the Cross Entropy Loss because it is suitable for class-matching problems, where the classes are the identifiers of the target vocabulary.

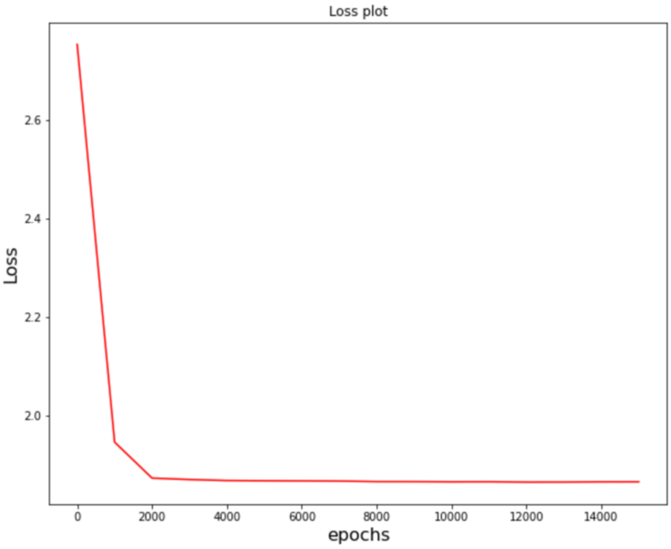
6.2. I implemented code for training the model. During the training we can see training and validation losses and accuracy for training and validation subsets. I print it after each 1000 iterations.

6.3. I implemented gradient accumulation in the training loop.

7. Experiments

7.1. Using the training pipeline i trained a model for the module numbers - place value using the mentioned hyper-parameters.

7.2. I reported loss plot (Figure 2). I could not figure out, why my accuracy function does not work, so I cannot plot accuracy for training and validation.



(Figure 2)