AI539 Spring 2024 Homework I

Rigved Naukarkar – naukarkr@oregonstate.edu

May 6, 2024

Task 1.1

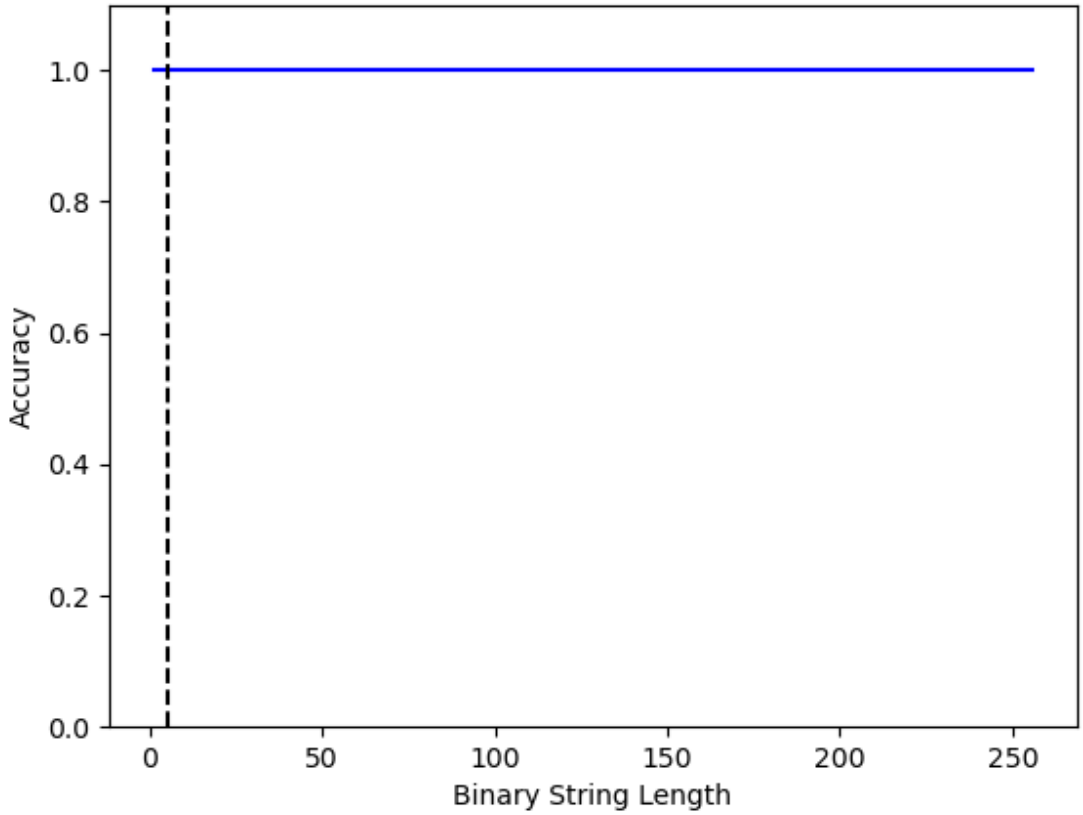
Please check the Python file with the weights and biases chosen for making the LSTM work like an XOR gate.

Task 2.1

The architecture chosen for this problem is an LSTM (used internal Pytorch function) and a linear layer with 2 scores for each sequence - even and odd parity scores. LSTMs are used because they work better than vanilla RNNs for sequential inputs.

Task 2.2

The model shows an accuracy of 100%. This might be because the architecture is quite good for the problem at hand. As we were manually able to find the weights for an univariate LSTM in the last section, similarly the neural network can also do it. While training, it was showing some loss for some initial epochs but then learned the parameters.



Task 2.3

I ran these 3 experiments.

- First, I tried to decrease the hidden_dim param from 16 to 8. saw that the original learning rate of 0.003 was showing less accuracy as the sequence length started to increase.
- Next, I tried to make the last experiment better and surprisingly, after increasing the learning rate to 0.01, the accuracy increased to 100% for all sequence lengths.
- Lastly, I tried to decrease the number of epochs from 2000 to 500. This was on top of the last experiments. This change led to more drop in accuracy than the first experiment, that too early in the sequence length than the first experiment.

Task 2.4

The RNNs suffer from vanishing or exploding gradients problem. LSTMs learn better than RNNs because of their structure of forget-gate, input-gate and output-gate. Even with the residual connections, LSTMs are shown to work better until some length of sequences.

The structure of the problem is a DFA. The vanilla RNNs seem to memorize the training data but cannot understand the underlying patterns of the DFA, to predict correctly with the new data.

Task 3.1

The dataset seems to have news-related data. Examples:

- however, differences between india and ltte soon surfaced and led to clashes between tiger guerrillas and the indian peace keeping force.
- but there are strong hints in the country that a new indo - sri lanka defense deal could be in the making.

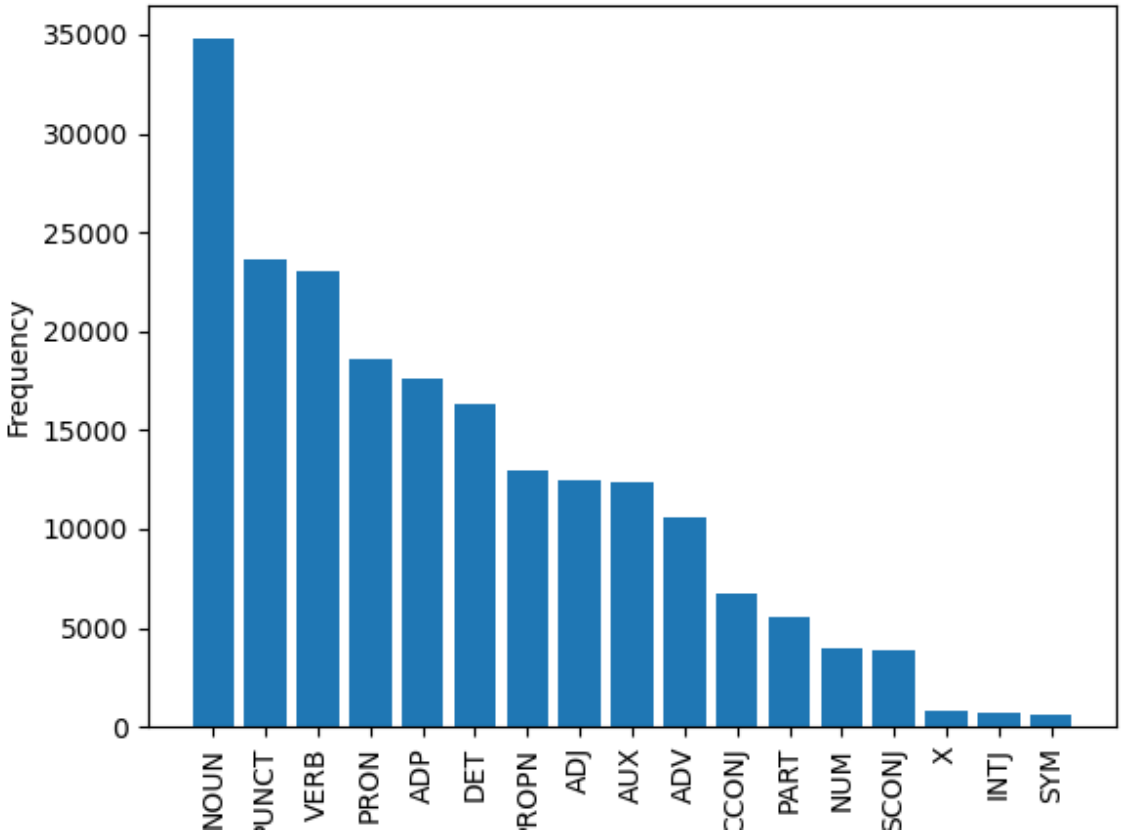
Also I see that some conversational data is present like emails. Example:

- this email is intended to be reviewed by only the individual or organization named above.

This dataset seems to have a type of data, like news-related, consecutively. So we need to shuffle it. Also, removing the suffixes and prefixes will make this data bad for our problem. The part of speech is many times dependent on the affixes. We even need the punctuation as we are tagging that as well. So we just need to lowercase and split based on spaces.

The count of the tags is displayed below. The "NOUN" tag seems the most prevalent tag with exactly 17% of the total tags. so if all the text is predicted as "NOUN" then 17% accuracy will be achieved.

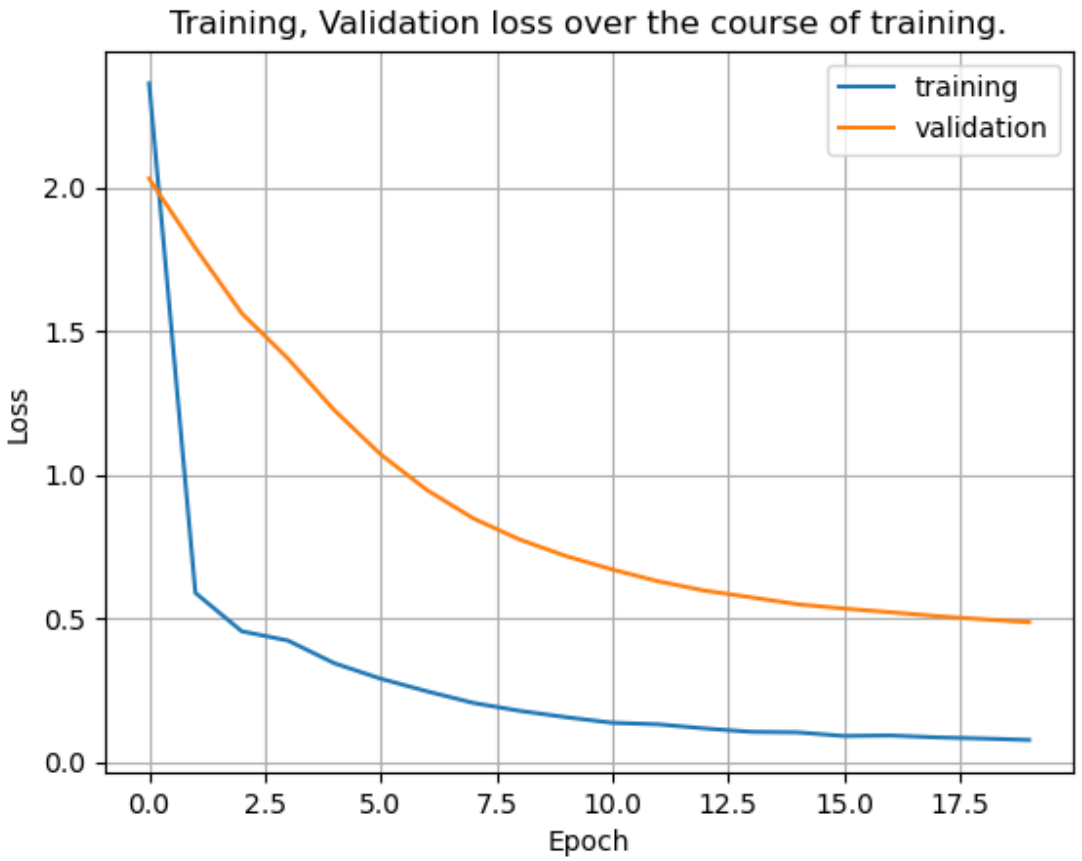
Please find the count of each tag in the attached file "tag_count.txt". Below is the histogram of the tag counts.



Task 3.2

In this part, I have used pretrained token embeddings from GloVe, implementing a bidirectional LSTM with an output distribution over the tags. I have used the adam optimizer as in the parity code. First, I ran 150 epochs and saved the model weights where the validation loss was the least. I also tried to change the embedding dimensions and 100 seems best. Below is the graph of training and validation losses after 20 epochs.

The model achieved the test accuracy of 87.088% and test loss of 0.308.



Task 3.3

The tag_sentence function returned the following output. The output is not as expected.

Listing 1: Count of tags in the dataset

```
1
2  ------------------------------------------------------------
3  TAG         TOKEN
4
5  DET         the
6  ADJ         old
7  NOUN        man
8  DET         the
9  NOUN        boat
10 PUNCT       .
11 ------------------------------------------------------------
12 TAG         TOKEN
```

PROPN The
ADJ **complex**
NOUN houses
VERB married
CCONJ **and**
ADJ single
NOUN soldiers
CCONJ **and**
PRON their
NOUN families
PUNCT .

TAG TOKEN

ADJ The
NOUN man
PRON who
VERB hunts
VERB ducks
ADP out
ADP on
NOUN weekends
PUNCT .