AI 539 - Spring 2024

Zexian Li
lizexi@oregonstate.edu
ONID: lizexi

Assignment 2

1 Demystifying Recurrent Neural Networks

1.1

$$w_{ix} = 10, w_{ih} = 0, b_i = -5$$

$$w_{fx} = 0, w_{fj} = 0, b_f = 10$$

$$w_{ox} = 0, w_{oh} = 0, b_o = 10$$

$$w_{gx} = 10, w_{gh} = -10, b_g = 0$$

2 Learning to Copy Finite State Machines

2.1

The architecture of the model consists of 3 layers and a forward function. The model uses the default hidden dimension of 16 units and takes in the input bit by bit. The output tensor has two layers for even and odd parity. The forward feeds the "pack_padded_sequence" batch into the LSTM layer and the 2 dimension tensor is extracted from the linear layer.

2.2

With the default number of hidden dimensions of 16, I think that the model achieved great results. It has an accuracy of 100% throughout the different lengths

2.3

I first started with a number of hidden dimensions equal to 2. With 2 hidden dimensions, it still achieves great results and almost 100% accuracy. For a solid 100% accuracy, the smallest number of hidden dimensions was 4. I also tried experimenting with a hidden dimension of 5 and found that the accuracy decreases as the binary string length increases.

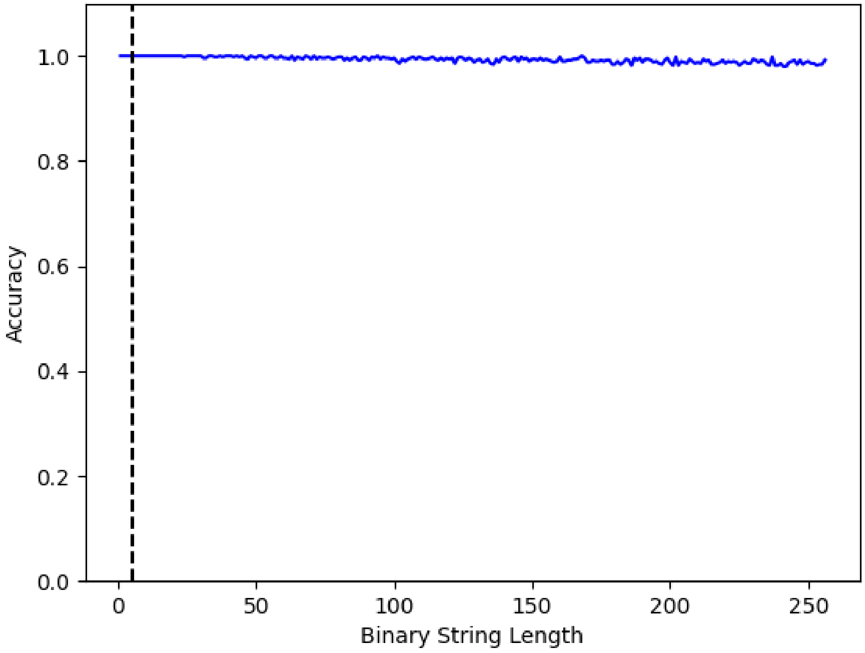

Figure 1: LSTM with 2 hidden dimensions

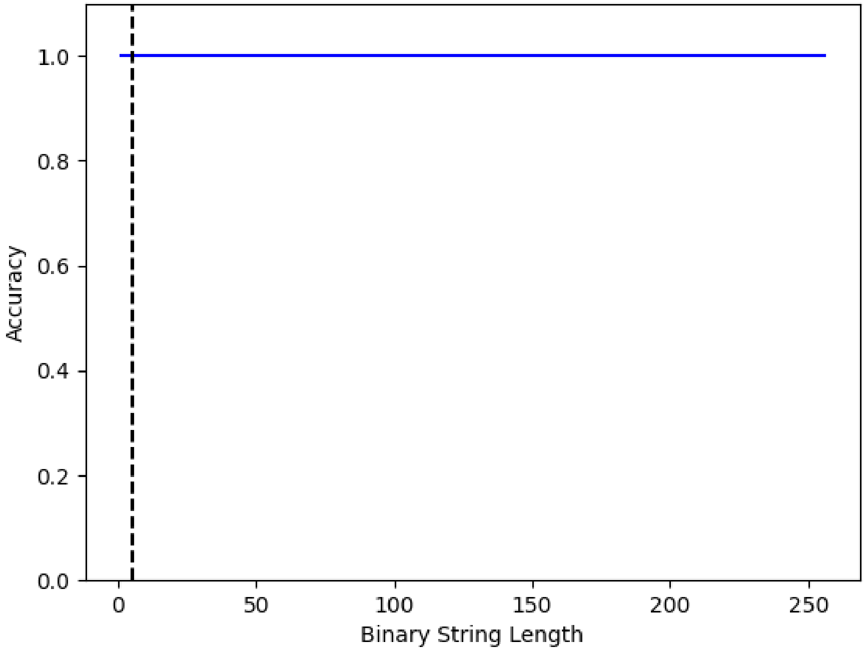

Figure 2: LSTM with 4 hidden dimensions

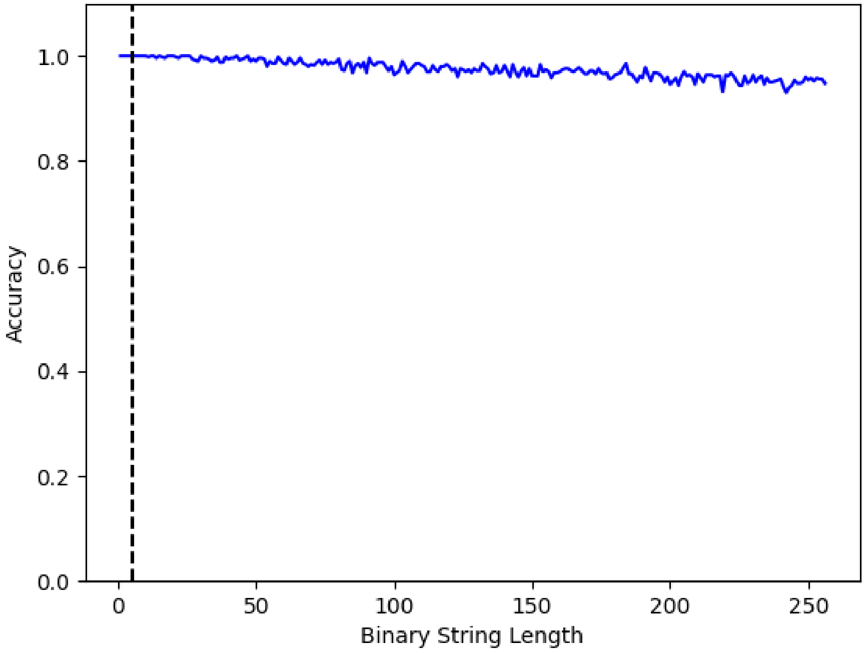

Figure 3: LSTM with 5 hidden dimensions

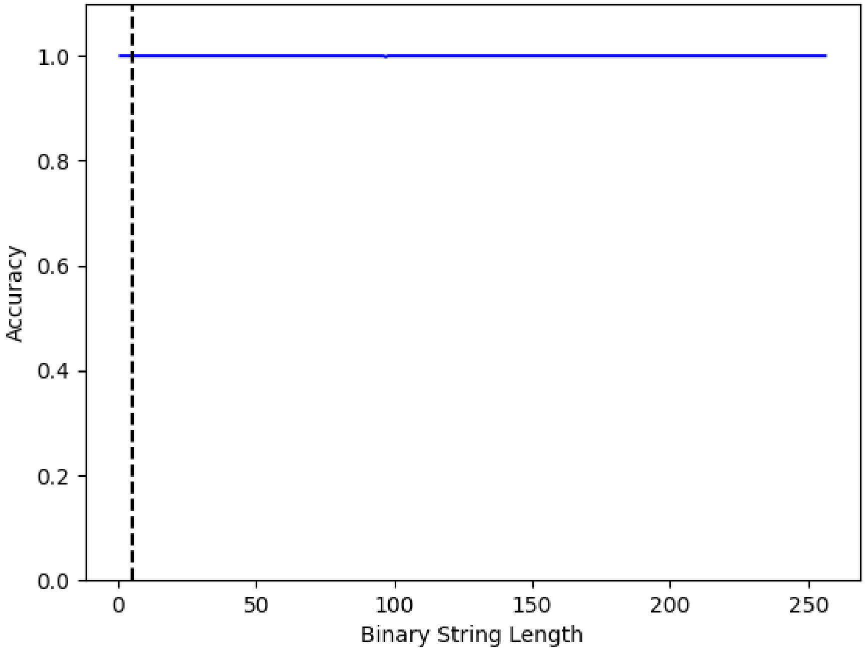

Figure 4: LSTM with 16 hidden dimensions

2.4

RNNs have a hard time classifying whether strings were generated by an ERG, and LSTMs perform better because of the vanishing gradient problem. LSTMs with their architecture, include gates that regulate the information allowing it to selectively remember or forget information. This way the LSTM can maintain a stable gradient over longer sequences. Additionally, LSTMs can support a longer memory using the cell states.

3 Part-of-Speech Tagging

3.1

The dataset contains sentences that are broken down and fit within 17 tags. The data is just sentences that could be present in a conversation. I wouldn't say that it is balanced as NOUN appears almost two times more that the next most common tag. However, this could just be because of how sentences are structured in the English language.

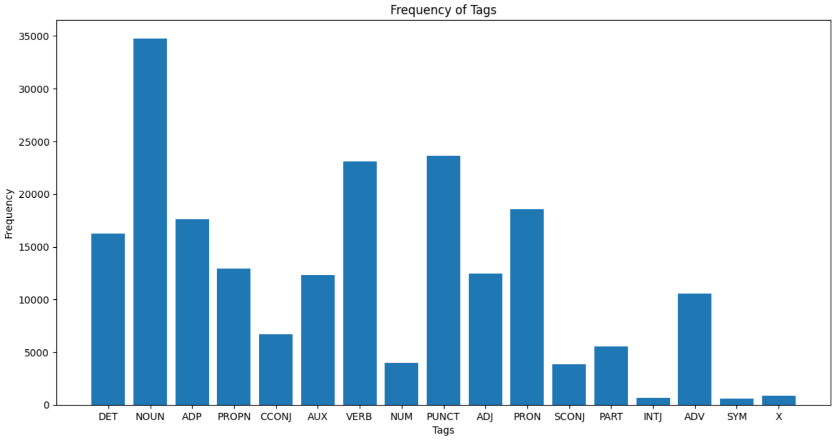

Figure 5: Histogram of tags

3.2

The model was created using the LSTM function provided by the library PyTorch. Following the directions, this LSTM is bidirectional. The final validation loss achieved was close to 0.014 in a short training of 5 epochs.

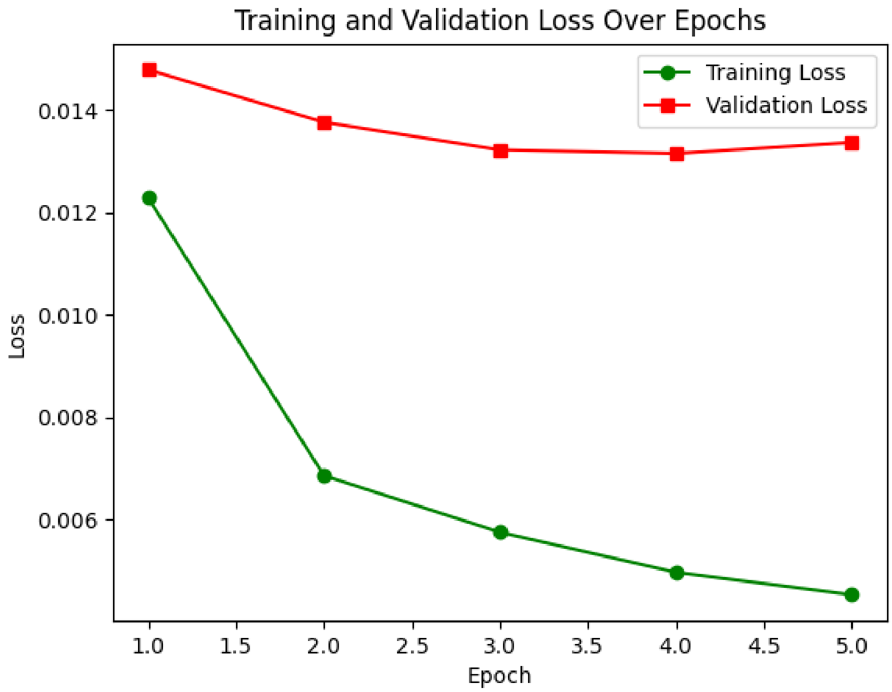

Figure 6: Training and Validation loss

3.3

1. Input: "The old man the boat."
   Token: [’The’, ’old’, ’man’, ’the’, ’boat’, ’.’]
   Output: [’AUX’, ’ADJ’, ’NOUN’, ’DET’, ’NOUN’, ’PUNTC’]

2. Input: "The complex houses married and single soldiers and their families."
   Token: "[’The’, ’complex’, ’houses’, ’married’, ’and’, ’single’, ’soldiers’, ’and’, ’their’, ’families’, ’.’] "
   Output: [’AUX’, ’ADJ’, ’NOUN’, ’VERB’, ’CCONJ’, ’ADJ’, ’NOUN’, ’CCONJ’, ’PRON’, ’NOUN’, ’PUNCT’]

3. Input: "The man who hunts ducks out on weekends."
   Token: [’The’, ’man’, ’who’, ’hunts’, ’ducks’, ’out’, ’on’, ’weekends’, ’.’]
   Output: [’<unk>’, ’NOUN’, ’PRON’, ’AUX’, ’AUX’, ’ADP’, ’ADP’, ’NOUN’, ’PUNCT’]