Assignment 3 - RNN Acceptors and BiRNN Transducers

Itay Etelis, 209041474,
Matan Avitan, 318287455

June 2023

1 Part 2: Acceptor Capabilities: Attempt to Induce Failure

The second part of our study aimed to explore the limits of the LSTM acceptor model by exposing it to a more challenging task. We used a context-sensitive language, specifically PRIME_1, which represents prime numbers in unary notation. Generating and validating prime numbers is a task that has become considerably easier in the modern era due to the advent of probabilistic primality tests, such as the Miller-Rabin algorithm.

1.1 Problem Definition: PRIME_1

The PRIME_1 problem involves a language that consists of all prime numbers represented in unary notation. Formally, we define PRIME_1 as follows:

$$\text{PRIME}_1 = \{w \mid w \text{ represents a prime number in unary notation}\} \tag{1}$$

1.2 Problem Selection

Selecting an appropriate problem is crucial for testing the model's capabilities. We initially considered a NP-complete problem, SAT, but generating an appropriate dataset and encoding CNF formulas would require substantial effort. Instead, we opted for the PRIME_1 problem, which still poses significant challenges for the LSTM but is easier to implement.

1.3 Dataset Generation

The dataset was generated using the sympy library to create and validate prime numbers. We created 1000 random samples for each class and partitioned them into training and test sets with an 80:20 split.

1.4 Model Architecture

The LSTM model used in the first part of our study was also used here, with some adjustments to the parameters. The new parameters are summarized in Table 1.

Parameter	**Value**
Batch Size	32
Learning Rate	0.001
Embedding Dimension	10
LSTM Hidden Units	64
Fully Connected Hidden Units	32
Training Epochs	300

Table 1: Model parameters for Part 2

1.5 Training Results

The model struggled to learn the pattern of the sequences in the $PRIME_1$ language. The accuracy on both the training and test sets was approximately 50%, suggesting that the model was unable to outperform random guessing. These results are summarized in Table 2.

Metric	**Value**
Training Accuracy	50%
Test Accuracy	48.2%

Table 2: Training results for Part 2

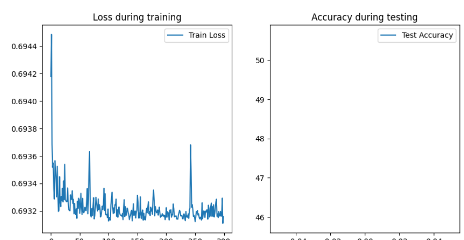

Figure 1: Training Testing on PRIME_1 *problem*

1.6 Discussion

These results underscore the challenges associated with applying LSTM models to context-sensitive languages. Such languages require more than simple counting abilities, pushing the limits of what LSTMs can achieve.