Assignment 3 - RNN Acceptors and BiRNN Transducers

Itay Etelis, 209041474,
Matan Avitan, 318287455

March 2023

1 Writing the RNN Acceptor Network

In this section, we discuss the process of developing a recurrent neural network (RNN) acceptor for the task of language classification. The model employs a Long Short-Term Memory (LSTM) network, followed by a fully connected layer.

1.1 Dataset Generation and Preprocessing

Our dataset comprised 1000 samples, evenly split between two classes.
The samples were randomly generated, with the following regex form:

$$[0-9]+a[0-9]+b[0-9]+c[0-9]+d$$

When the $[0-9]+$ term was constructed with an upper bound of 50 characters, hence, the longest length is 254 chars. The dataset was then shuffled and partitioned into training and test sets with a ratio of 0.8:0.2, respectively.

Subsequently, we tokenized the dataset and the labels. Padding was applied to ensure consistency in sequence length, matching the length of the longest sequence in the batch.

1.2 Model Architecture

Our model consists of an embedding layer, an LSTM layer, and a linear classifier. The specific parameters are illustrated in Table 1.

The model was trained using the AdamW optimizer with a learning rate of 0.001 and weight decay of 0.01.

1.3 Handling Padding

To handle the padded sequences in our model, we ran the LSTM until the start of the padding and used the output of the LSTM at that point.

Layer	Parameters
Embedding	Size = 4
LSTM	Hidden units = 32
Linear classifier	Weight matrix size = 32 x 2 + 2
Output	Neurons = 2

Table 1: Model parameters

1.4 Training Results

Remarkably, the model achieved 99% accuracy on both the training and test sets. The total training time, conducted on a CPU using Colab, was 3.34 seconds with 1000 iterations, as shown in Table 2.

Metric	Value
Training Accuracy	100%
Test Accuracy	100%
Training Time	8.64 seconds
Iterations	1000

Table 2: Training results

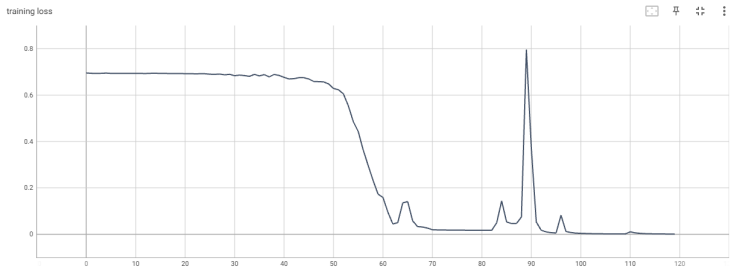

Figure 1: Train Loss

1.5 Discussion

The network exhibited a notable transition during the 60th epoch, initiating the convergence process that ultimately resulted in a 99% accuracy. Additionally, a remarkable occurrence was observed during the 88th epoch, characterized by an unusual degradation followed by a gradual convergence towards a stable accuracy plateau of 99%.

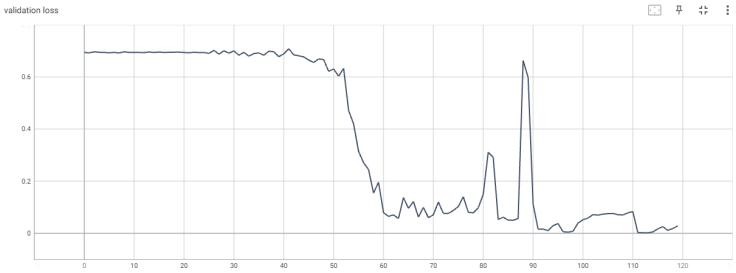

Figure 2: Validation Loss

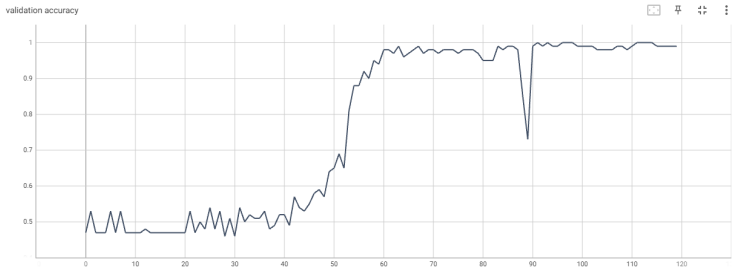

Figure 3: Validation Loss