Report 1: RNN Acceptor Experiment Summary

Submitters:

Ori Braverman 318917010

Elie nedjar 336140116

Dataset

- **Positive Examples**: Sequence format - `[1-9]+a+[1-9]+b+[1-9]+c+[1-9]+d+[1-9]+`

- **Negative Examples**: Sequence format - `[1-9]+a+[1-9]+c+[1-9]+b+[1-9]+d+[1-9]+`

Also, for each subsequence there is a maximum length of 20 digits/letters.

Higher thresholds for the subsequence means more difficult training data to use.

- **Number of Examples**:

- Positive examples: 500

- Negative examples: 500

- **Training and Test Sets**:

- There are 1000 samples which are divided into 900 samples for the training set and 100 samples for dev set.

- There are 500 samples to the test set.

Experiment Details

-**Hyperparameters**

- Dev ratio = 0.1
- Batch size = 16
- Epochs = 25
- Embedding dim = 30
- Lstm hidden dim = 32
- Mlp hidden dim = 16
- Learning rate = 0.003

- **Network Architecture**:

- LSTM RNN using pytorch nn.LSTMCell

- MLP with one hidden layer

Results

- **Training Performance and duration**:

  - Accuracy on training set = 99%

  - Duration on training set = 4 seconds

  - Accuracy on test set = 96%

  - Duration on test set = less than a second

The training was done on CPU so we assume the durations will be much longer compared to training on a GPU.

- **Observations**:

  - The network was able to distinguish between positive and negative examples effectively.

  - The network performed well on both training and test sets.

  - Steps taken to improve performance included optimizing hyperparameters and adjusting the network architecture.