**Part 2 – report**

For this part of the assignment, we defined 3 languages that we thought could make the LSTM model fail and we tested our model from part 1 of the assginment on them.

**Language 1: Palindromes over {a,b}**

We defined the language of palindromes as sequences that read the same forward and backword

**Formal Definition:**  
L = { w ∈ {a,b}\* : w = reverse(w) }

**Difficulty for RNNs:**

We hypothesized that the palindrome language will be difficult for RNNs since it require remebering the entire first part of the sentence and perfroming a comparison against the second part. Also, the model will have a difficult time understanding when the first part of the sentence ends.

**Test set**

Our train set contained 40,000 examples. 50% of them were positive examples and 50% were negative examples which were originally palindormes where one letter is altered. Our test set had 2,000 samples with a 50% positive and negative split.

**Model Performace**

The RNN model failed in this task. We trained the model for 15 epochs. Our model’s accuracy for both the train and test sets did not go over 50% (chance) accuracy throughout the entire model training.

**Language 2: Copy language**

We defined the copy language as sequences that are made of 2 repeats of the same sequence

**Formal Definition:**  
L = { ww : w ∈ {a,b}\* }

**Difficulty for RNNs:**

Like the palidnrome language, we thought that the copy language will be difficult for RNNs since it require remebering the entire first repeat of the sequence and matching it against the second repeat. Also, the model will have no prior knowledge of where the first repeat of the sentence ends and the second repeat start.

**Test set**

Our train set contained 40,000 examples. 50% of them were positive examples and 50% were negative examples which were based on sequences of the copy language where one letter was altered. Our test set had 2,000 samples with a 50% positive and negative split.

**Model Performace**

The RNN model failed in this task. . We trained the model for 15 epochs. Our model’s accuracy for both the train and test sets did not go over 50% (chance) accuracy throughout the entire model training.

**Language 3: Shuffle language**

We defined the function shuffle(w,w) as an interleaving of the two copies of w that preserves the order of symbols in each copy)

**Definition:**  
L = { shuffle(w,w) : w ∈ {a,b}\* }

**Difficulty for RNNs:**

We thought that the shuffle language will be difficult for RNNs since it require tracking two sequences in parallel, ensuring each symbol from the first and second copy appears exactly once and in the right order. Also, the model will have no prior knowledge of what the 2 sequences are.

**Test set**

Our train set contained 40,000 examples. 50% of them were positive examples and 50% were negative examples which were based on sequences of the shuffle language where one letter was altered. Our test set had 2,000 samples with a 50% positive and negative split.

**Model Performace**

The RNN model failed in this task. . We trained the model for 15 epochs. Our model’s accuracy for both the train and test sets did not go over 50% (chance) accuracy throughout the entire model training.