Yaniv Sheena

308446764

1) **Experiment 1**:

    a. The language is composed of all the sequences over {a,b} where the amount of the occurrences of ‘a’ is greater or equals the amount of occurrences of ‘b’.
    b. I thought that the counting task will be an obstaclefor the RNN, since the RNN cannot use the order of the occurrences (it is not important in this task).
    c.
        i. I used training and test-set each composed of 10,000 examples
        ii. After 3 iterations over the training examples the LSTM succeeded both on the train-set and the test-set (95-97% accuracy)

2) **Experiment 2**:

    a. The language is composed of all the sequences over {a,b} where the amount of the occurrencesof the first character is greater or equals the occurrences amount of the other character. (e.g. ‘abaab‘ and ‘bbaab’ are both in the language, ‘abb’ and ‘baa’ are not).
    b. Here I added a twist to the first language, which forces the LSTM to behave differently dependingon the first input. I thought this twist can make it fail.
    c.
        i. I used training and test-set each composed of 10,000 examples
        ii. After 6 iterations over the training examples the LSTM succeeded both on the train-set and the test-set (93-94% accuracy)

3) **Experiment 3**:

    a. The language is composed of all palindromesover {a,b}. The model should distinguish good sequences(palindromes) and bad sequences (almost palindromes) in which inversion of exactly one character will make the sequence a palindrome. (e.g. good= “abaabaaba”, bad = “abaabaa**a**a” - almostpalindrome).
    b. The task of accepting palindromes is very complex when the sequence size is not constant. Because the LSTM does not know what will be the sequence length in prediction time, it cannotknowin which point the sequence is starting to appear in reversed order. Hence, solving this problem requires selective memory of the exact index of every character, and also an ability to compare every char to all the chars that appeared before along with remembering all the substring that have a potential to be the prefix of the reverse ordered part.
    c.
        i. I used training and test-set each composed of 40,000 examples (after I tried 10,000 – I increased to 4 times bigger than previous tries)

ii. The LSTM is failed – after 20 iterations the accuracy and loss did not improve. The accuracy on both training set and test set was ~50%.