Israel Cohen – 205812290 Hananel Hadad – 313369183

Part1 – deceiving RNN acceptor:

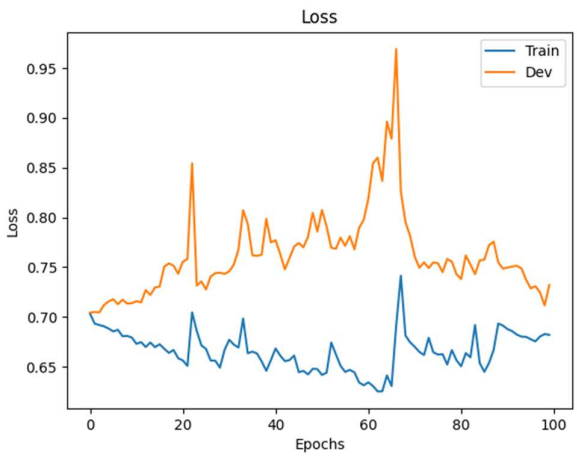
In all 3 methods, we used the letters set of (a,b,c,d, 1) so we can still use our old model without too much headache. We neglected the other numbers as they are treated the same.

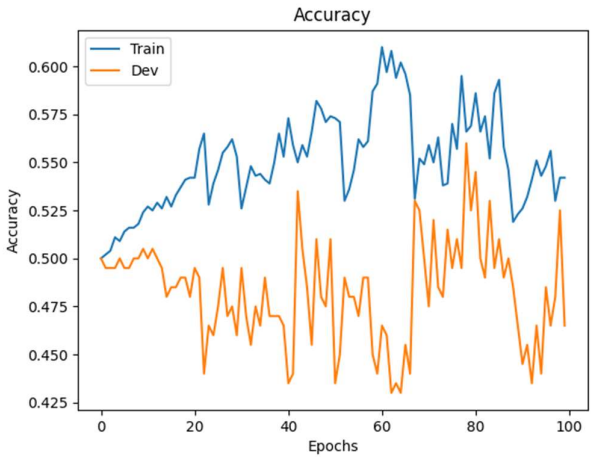
In all 3 methods, the length of samples is random number between 1 to 40.

We run 100 epochs, to be sure the models would never learn more

palindrome:

we took the memory of order, which the RNN does, to the extreme: pos samples are palindromes, and neg samples are palindrome after changing single random letter.
The model was not able to learn at all.



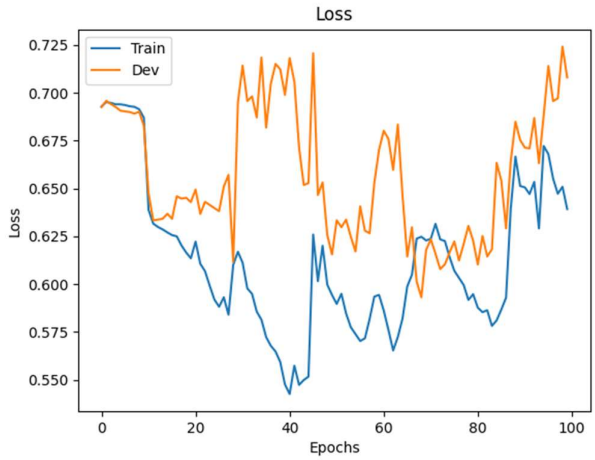


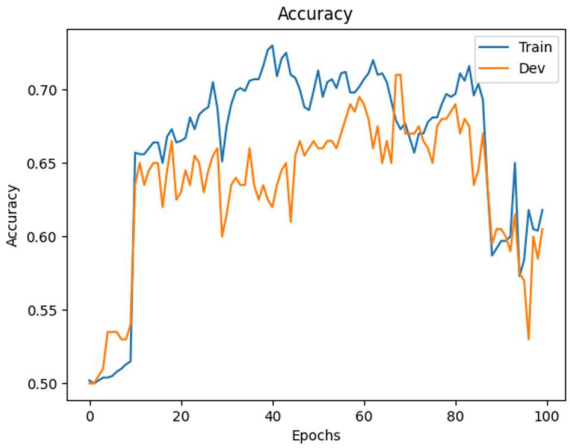
10NoA:

We came up with this rule: pos samples have 'a' in their first 10 places (any number of 'a', anywhere in indexes 0-9), whereas neg samples may have 'a' only form the 11's character and afterward.

All samples still have random length, so they will not have same proportions: one neg sample must not have 'a' at all (shorter than 10), while another neg sample is restricted only in its first quarter (length of 40).

Interestingly, we see that the model was able to learn something and partly succeed, with the peak being around 60 epochs





quarterNoA:

we wanted to see what happens when the sample are proportional with the previous rule. This time the rule is that regardless of length, each neg sample must not have 'a' in its first quarter, and pos samples must have.

The model was not successful in this task either.