Dafna Magid 315635870 Oved Nagar 302824875

**The 3 most challenging cases we came up with:**

| | 1- <u>Long end</u> |
|---|---|
| description of the languages | 1. a[1-9]b[1-9]c[1-9]d[1-9]{20}<br>2. a[1-9]c[1-9]b[1-9]d[1-9]{20} |
| Why did you think the language will be hard to distinguish? | The distinguishing is done according to the last output, and because of the vanishing gradients problem, distant changes have less of an effect. For a higher number of epok (we tried 100) the network will be able to identify the our languages with the addition of the 20/21 numbers at the end, but for an addition which is sufficiently long, the network will certainly fail. |

| | 2- <u>Length</u> |
|---|---|
| description of the languages | 1. [1-9]{20}<br>2. [1-9]{21} |
| Why did you think the language will be hard to distinguish? | A normal RNN network is unable to distinguish between different lengths, so the difference between languages can't be identified. |

| | 3- <u>Order</u> |
|---|---|
| description of the languages | 1. a[a+b+]{4}<br>2. b[b+a+]{4} |
| Why did you think the language will be hard to distinguish? | The network consists of a matrix that preserves the locale changes all along the way.<br>For both languages there are the same changes (ab / ba) and probably therefore it is difficult to distinguish. |

**For all 3 languages:**

| | |
|---|---|
| train and test set sizes | Train-800<br>Set-200 |
| Number of iterations we trained | 100 |
| did it manage to learn the train but did not generalize well to the test, or did it fail also on train? | We have not encountered such case. |
| Did you manage to fail the LSTM acceptor? | Yes. |