

Report 1:

David Samuelson, 208788851

I did not experiment with this part too much, because it worked very fast.

Here is the parameters I used which worked:

- Input dim: 100
- Lstm dim: 100
- Epochs: 10
- Train size: 250 samples, randomly generated:
 - Sample length:

The language is $[n+ a+ n+ b+ n+ c+ n+ d+]$, where n is a number.
I generated the samples in the following way:
There is a function called 'generate sample', which receives order as parameter (this order is used to generate bad samples as well).
pseudo code:
 - Sample = " # empty string
 - for each letter 'c' in the order:
 - len = random number, in the range $1 \leq len \leq 50$
 - if $c == 'n'$:
 - sample += random number in length of len
 - else:
 - sample += $c * len$

samples were generated in the following way:
good sample: generate_sample('nanbncnd')
bad sample: generate_sample('nancnbnd')

I noticed that when the lstm dim was too small in relation to the sample length, the acceptor could not learn the language.

for example – lstm dim < 50, and the variable MAX_LEN was around 100, the samples were around 500 in size, and the acceptor sometimes failed to learn.

So I decreased MAX_LEN to 50, and increased input dim and lstm dim to 100.