Exploring SORNs Performance Using Multiple Plasticity Configurations

Ludovico Deponte, Caroline Kellner, Chloe Maas

1 Introduction

To properly understand time-dependent information processing in the brain, it is crucial to grasp the dynamics of recurrent neural networks in the cortex. Cortical plasticity is the ability of the brain to learn and change in response to experience and use, and it involves a variety of different synaptic and cellular mechanisms. Implementing these should help bring us closer to biologically plausible artificial neural networks, while providing precious insights into the working of our brain.

The self-organizing recurrent neural network (SORN) by Lazar et al. (2009) was proposed to tackle sequential tasks, such as predicting the length of a sequence while receiving its symbols successively as input, by implementing biologically-inspired plasticity mechanism between its neurons. It uses a combination of excitatory (blue) and inhibitory (orange) neurons to create an inner representation of the input sequence, and a readout layer (white) that interprets the activation of the network and outputs the network's prediction for the next symbol in the sequence.

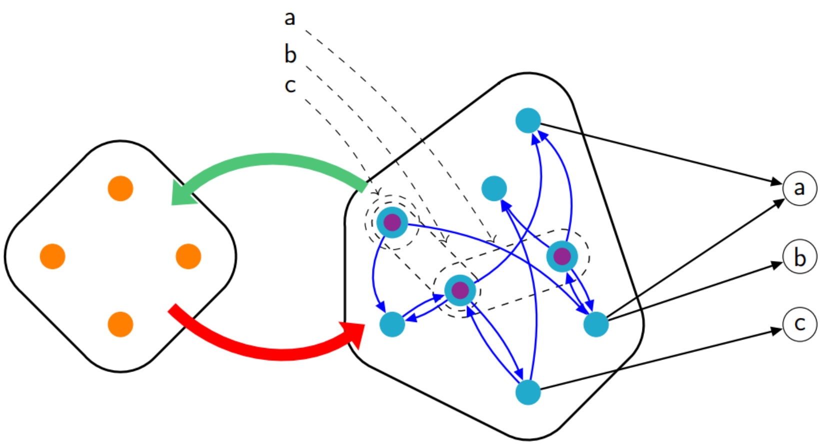

Figure 1: Architecture of SORN. SORN is comprised of excitatory (blue) and inhibitory (red) neurons, as well as an output layer.

The sparse connections between the excitatory neurons are trained using three different forms of plasticity: spike-timing-dependent plasticity (STDP), synaptic normalization (SN), and intrinsic plasticity (IP).
The first, being inspired by Hebbian plasticity and the motto 'neurons that fire successively wire together', strengthens by a small factor η the connection from neuron i to neuron j if i fires before j does, while it weakens it by the same factor if i is not active before j is.[1] The second adjusts the values of incoming connections so that the signal that

[1]Notice that this nicely implements a reinforcement of a 'causal' relation between neurons' activations.

each neuron receives adds up to 1.[2] The latter adjusts the firing threshold of each neuron so that, on average, the excitatory neurons fire at fixed rate.[3]

Lazar et al. (2009) show that SORN is able to learn time- and context-sensitive representations of its input sequences. This means that SORN is able to distinguish different repetitions of the same input from each other and map them onto different activation states. Compared to a static recurrent neural network SORN is able to correctly predict longer input and shows a higher variance in activity patterns. Lazar et al. (2009) interpret the combination of all plasticity mechanisms as the main driving factor behind this advantage of SORN over static networks. They explore the role of different plasticity mechanisms by switching off either IP or SN and find that this interacts with the network dynamics. However, while doing this, they use random input instead of the experiment task. This means that the question of how well the network performs on the actual task, when either one or more plasticity mechanisms are switched off, remains. In our project we attempt to answer it.

2 Methods

Our experiments relay on the implementation of SORN by Del Papa et al. (2017).[4] After creating a Jupyter notebook on CoLab, we added new `param.py` files to store SORN's new parameters; we run a baseline simulation with the original parameters for SORN with 100, 200, 400 and 800 excitatory neurons. The simulation loops over the Counting Task of the original paper for sequences of $4, 6, 8, 10, 12, 14, 16, 18, 20, 22$ symbols. For each sequence length a total of 10 simulations is performed. We use a short shell script to perform the loops and parameter modifications, as we find it the easiest way to run the simulations systematically without extensive changes in the original code.

After obtaining a baseline, we perform for each plasticity mechanism two simulations: the first by switching that plasticity mechanism off while keeping the other two active, the second one by doing the contrary, i.e. switching said plasticity on and keeping the other two off. For each of those simulations we perform the same loops described for the baseline.
After a good amount of time, we collect enough datapoints to plot the graphs in the next section.
Finally, we redefine the plotting function in the Jupyter notebook, adding a colour palette, a small offset on the horizontal axis and transparency over error bars to improve readability.

3 Results

We include three different figures, each containing, in blue, the results for the baseline (original parameters), in red the simulation with only one plasticity mechanism active and in green the simulation with the same mechanism turned off and the other two turned on. By plotting this we provide evidence of the relative importance of each plasticity.

[2]This keeps the proportions of the incoming signals, but regulates the total incoming drive.
[3]Again, this is inspired by the observation that the brain keeps a mostly stable firing rate.
[4]Github repository at: `https://github.com/delpapa/SORN_V2`.
Our repository can be found at: `https://github.com/LudovicoDeponte/FoNCM_SORN_project`

Intuitively, with this graphs we are comparing SORN's baseline performance, where all three plasticity mechanisms are active, with versions of the same network without some of the plasticity mechanism. We do this to confirm the claim that all three mechanisms are needed for a good performance.

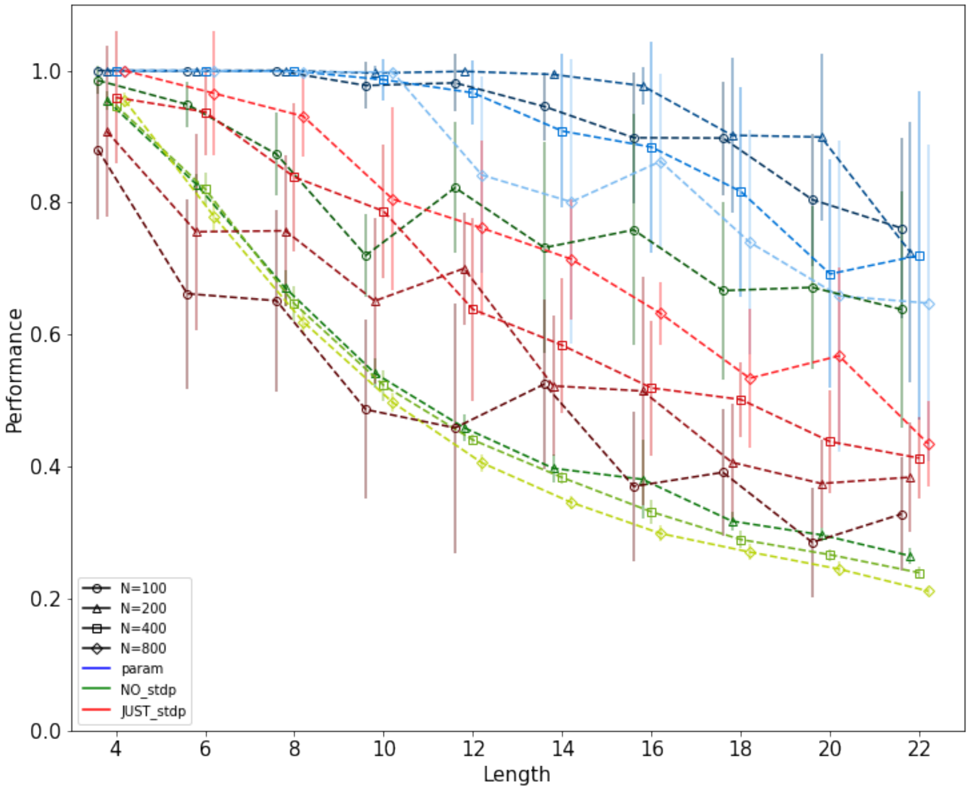

Figure 2: SORN's performance for the baseline initialization (blue), when using SN and IP but no STDP (green) , and when using only STDP (red).

By switching off the STDP (green plots figure 2) we observe that the network is worse at predicting the length of the input sequences. The performance of SORN utilizing only STDP is improved when compared to the green plots, falling almost exactly between them and the baseline with big network sizes (800 excitatory neurons) and being slightly better then the green plots with smaller network sizes. Notably the variance of the performance for the green plots is really low, while for both the baseline and the SORN using only STDP we have wider error bars.

In the original paper the authors switch off the synaptic plasticity or the intrinsic plasticity (corresponding to the green plots of figure 3 and figure 4), observing that both worsen the stability of the network dynamics.
Indeed, looking at the green plot in figure 3, reporting the performance of SORN with SN switched off, we see that the capacity to correctly predict sequences is drastically worsened and dropping to chance level. In contrast to the previous plot, the variance in this case is also high.
As previously, if SORN is initialized using only SN it performs better then an initialization with STDP and IP, with the caveat that when using only SN, the network size seems to be more correlated with the overall performance, as bigger networks achieve almost always a better performance.

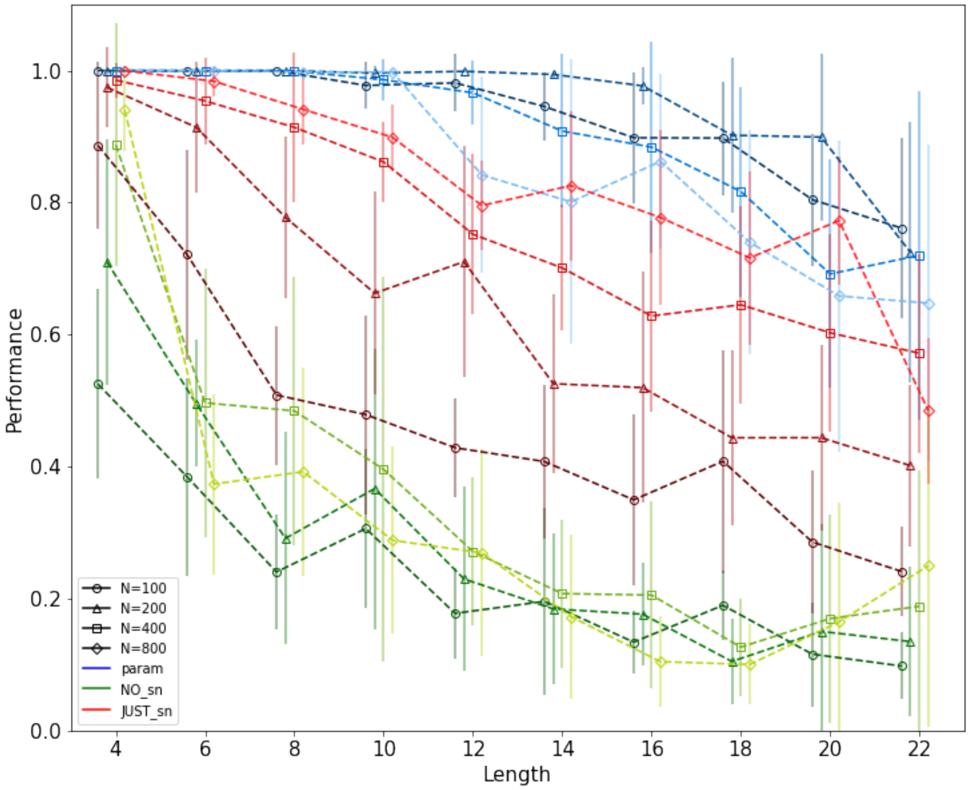

Figure 3: SORN's performance for the baseline initialization, when using STDP and IP but no SN, and when using only SN.

Finally, in figure 4 we see the performance of SORN with only IP active in red, while in green we see its performance with STDP and SN active. This is the only plot in which two mechanisms together perform better then the last one alone, suggesting that the most important plasticities are STDP and SN. Notably, the performance with only IP has low variance.

In figure 2 and figure 4 there are unexpected results for network of size 100: in figure 2, while networks with STDP inactive generally perform badly and have low variance, the smallest network performs similarly or better then any simulation with just STDP, while having high variance.

Similarly, in figure 4 a SORN of 100 neurons perform better than any other size when only using IP, reaching a performance comparable to SORN using STDP and SN, while for all other sizes the performance with only IP is lower then the one for STDP and SN. To verify this we run further experiments with size 60, 100 and 200, using SN and IP in the green plot and activating IP for the red plots (figure 5). Notably the results replicate, with an even smaller network performing even better.

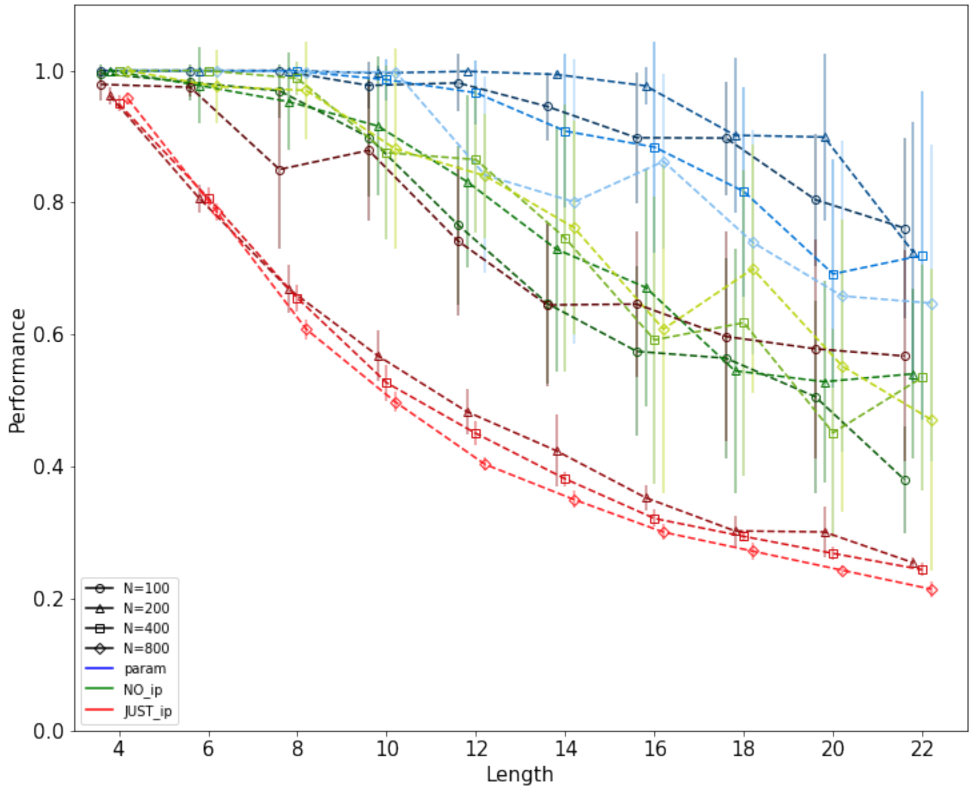

Figure 4: SORN's performance for the baseline initialization, when using STDP and SN but no IP, and when using only IP.

4 Conclusion

In all of the three plots, the baseline outperforms the other two simulations, verifying that the interaction between all the three plasticities is fundamental for a good performance, as claimed in the original paper.

Notably, the STDP and SN seem to be more important than IP for a high performance in the Counting Task, while IP seems to have a normalizing effect, reducing the performance's variance and thus making the network's inference more consistent.
In general we observe that if the red plots are above the green ones, the corresponding single plasticity performs better then a network with only the other two plasticities, hence implying that said plasticity is more relevant; if conversely the red lines are below the green ones, the corresponding plasticity is not crucial for a high performance. That is the case only for IP. We observe that the baseline model is always the best model, hence confirming the claim that the interaction between all three mechanisms is fundamental for achieving better results.
Intuitively, we expected STDP and SN to be closely linked to performance, as those plasticities act directly upon neuron's weights, while the bad performance of SORN with only IP can be explained interpreting IP as a normalization mechanism, that tries to keep the network's total activity fixed. This intuitively restricts the total amount of neurons that can be active, forcing less variance in the proportion of active versus inactive neurons; in turn this means that bigger networks, which in theory could achieve a bigger number

of different proportions between active and inactive neurons are limited only to some of them. This might explain why we see that indeed, when using IP with either STDP or SN (green plots in figure 2 and 3), the variance among different network sizes diminishes, even leading in figure 2 to more consistent performance across the same size (narrow error bars).

As previously noted, SORN 100 in the simulation with SN and IP but no STDP, and in the simulation with only IP, performs consistently better then the same simulations with higher numbers of excitatory neurons. Repeating the experiment while using an even smaller size reproduces the results (figure 5), crossing out the possibility of a 'lucky' initialization. However, in both cases the variance of the performance is really high, while being almost non-existent for bigger sizes.

One drawback of our experiments is that they are based on the original implementation of the SORN model. Since then, more complex models have been introduced. Those implement additional features such as plasticity of the inhibitory connections and membrane noise. These additions change the dynamics of SORN further and for example push it closer to criticality (Del Papa et al., 2017). In future research it might be interesting to explore which of the different plasticity mechanisms are actually needed in which phase of learning to reach optimal performance.
Additionally our experiments are based on the Counting Task introduced by Lazar et al. (2009). This means that the input data as well as the task on which the model is trained is rather minimal and artificial, limiting the external validity. To overcome this, more and more complex tasks such as the ones introduced by Orhan and Ma (2017) could be used for training SORN.
Since networks with only one plasticity switched on often performed better than networks with two types of plasticity it could be worthwhile to explore the dynamics between the two plasticities that lead to a worse performance.
Finally, it might be interesting to further explore models with lower network sizes, in order to explain the anomalies of figure 5.

References

Del Papa, B., Priesemann, V., & Triesch, J. (2017). Criticality meets learning: Criticality signatures in a self-organizing recurrent neural network. *PloS one*, *12*(5), e0178683.

Lazar, A., Pipa, G., & Triesch, J. (2009). Sorn: A self-organizing recurrent neural network. *Frontiers in computational neuroscience*, 23.

Orhan, A. E., & Ma, W. J. (2017). Efficient probabilistic inference in generic neural networks trained with non-probabilistic feedback. *Nature Communications*, *8*.

A Influence of network size

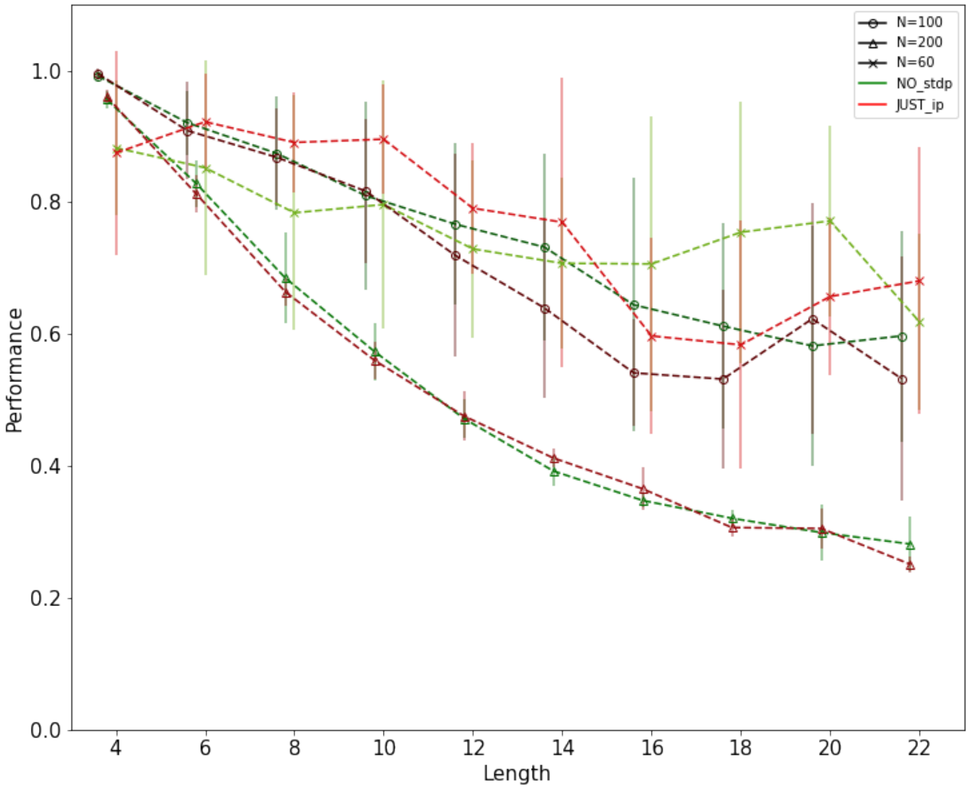

Figure 5: SORN's performance when using SN and IP, in green, and when using only IP for different network sizes; with those plasticities, smaller network perform better.