Parallel Processing in a Sparsely-connected Recurrent Neural Network: a Model for Ultra-short Term Memory

Julien Mayor and Wulfram Gerstner

Laboratory of Computational Neuroscience
EPFL, CH-1005 Lausanne, Switzerland

Abstract

A model for ultra-short term memory is analysed and its storage capacity is investigated. We demonstrate that it is possible to process online many superimposed streams of input. This is despite the fact that the stored information is spread throughout the network. The role of the underlying statistics of the inputs appear to play a crucial role in the way the information is extracted out of the network.

Key words: recurrent network, ultra-short term memory

1 Introduction

Recently, information processing in recurrent neural networks has gained much attention. Those theories are known as Liquid State Machines (LSM, (1; 2)) or echo state networks (3). They are good candidates to model ultra-short term memories. The idea underlying those models is that the instantaneous state of the network provides a rich reservoir of non-linear spatio-temporal transformations of the inputs. Information about past input can then be read out with simple, efficient and adaptive readouts. In general learning only acts on the readout structures, the network itself remaining fixed. Our set of non-linear transformations of the input is achieved through the use of a sparsely-connected neural network of integrate-and-fire neurons. Tuning the network to an asynchronous irregular firing state allows us to have the needed rich dynamics. In this paper we establish a link between liquid state machines on the one side and the theory of sparsely-connected networks (4) on the other side.

2 The Model

The system we study is a sparsely-connected network of leaky integrate-and-fire (IF) neurons. Such networks are known to have a complex dynamics ((4; 5)). Our network is made up of 200 IF neurons, 80% of which are excitatory and 20% inhibitory. Both excitatory and inhibitory neurons are modelled with a membrane time constant of 20ms. They are weakly (connection probability = 0.2) and randomly connected through simple static synapses. We carefully chose the synaptic strengths and an external drive such that the network (without additional external input) is in an asynchronous and irregular firing regime based on the phase diagram described in (4). With a weaker or a stronger input, the system reaches a phase of synchronous firing (either regular or irregular). This absence of synchrony at the working point of the liquid state machine is important in order to avoid limit cycles (periodic patterns of activity) and therefore indistinguishable moments that share the same phase within these cycles.

We assess the information processing capacity of the network with a procedure analogous to (1) and (3). We inject simultaneously N independent inputs to N disjoint groups of randomly chosen neurons (see figure 1 left). The inputs are derived from a bounded random walk so that each share the same underlying statistics. Their auto-correlation profiles can easily be measured, an analysis of which will be done in section 3.3. N readout structures, 'seeing' all neurons of the network are trained to retrieve the amplitude of their corresponding signal a given time T in the past (see figure 1 right). The outputs of the readout structures are simple linear combination of all the membrane potentials i.e.

$$Output(t) = \sum_k w_k u_k(t)$$

.

Only the weights w_k of the readout structures are tunable, the network itself remaining fixed. We minimise the error :

$$E = [Output(t) - Input(t-T)]^2$$

by an optimal regression on the training set. This $Input(t-T)$ plays the role of a target value for optimisation. After a training period, the weights of the readout structures are frozen and N new input signals are introduced in the network. Outputs of the readout structures are then compared with their corresponding targets.

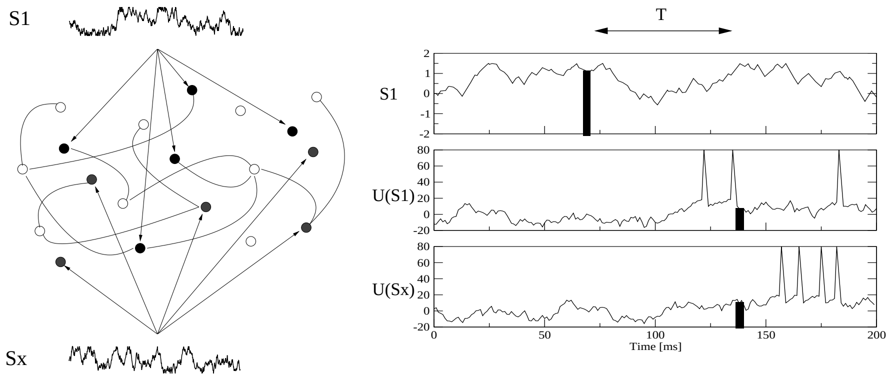


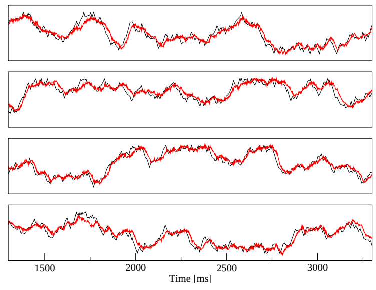
Fig. 1. Left: The different signals are introduced to randomly chosen interconnected neurons. Every neuron receives only one input from the exterior. Only two input signals are shown in the figure. Right: Based on the momentary state of all membrane potentials (those receiving directly the signal S1 *and* those receiving any other signal Sx), a readout structure is trained to guess the amplitude of its corresponding input a time T before (referred as delay thereafter).

3 Results and Discussion

In the following section we will discuss the results we obtained by the procedure described above. Firstly we will show that it is possible to process online many superimposed streams of input. In the second subsection we will show that this is possible despite the fact that information has diffused within the network. In the last subsection we investigate the role of the underlying statistics of the input and see that the output structures use all the available information in order to minimise the distance of its output to the target.

3.1 Multiple Inputs

Following the procedure described in the previous section, we inject simultaneously eight independent signals. The training time is 50000 time steps. Performances are obtained on a test set of 5000 time steps. In figure 2 (left), only four of the eight input-output pairs are shown. In order to allow a qualitative visual assessment of the performance, inputs are shifted back in time so that they can be directly compared with the output. The time delay shown here is 10ms. In figure 2 (right), the cross-correlations between the output of the readout structures and their corresponding targets are shown as a function of the delay. A significant amount of information is still present up to about 150ms. This number should be compared with the membrane time constant of 20ms. Although the eight signals excite the network simultaneously, the high dimensionality of the system allows the readout structures to extract any individual signal.



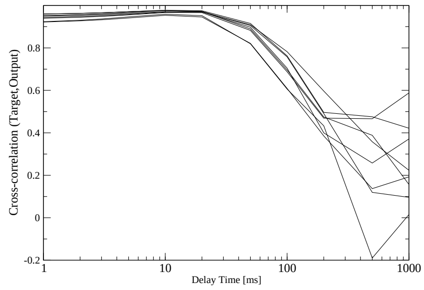


Fig. 2. Left: A sample of four target(black)-output(grey) pairs. Trained delay: 10ms. Right: Cross-correlation curves for the eight target-output pairs.

3.2 Information Diffusion

A control has yet to be done. Although the network is generated randomly, local loops might dominate on a short length scale. Information might then be stuck in only part of the network, where it was injected. To rule out this possibility, a closer look at the weights of the readouts was taken. This time we injected simultaneously four different signals. For three different delays and four different readout structures, the sixteen weights with the largest value have been identified. Neurons of the network have been grouped into four groups according to the label of the input they receive. Those weights have then been counted in each of the zones. The results are shown as grey-level plots: the horizontal axis is the label of the zones and the vertical axis is the label of the readout structures, the grey-level being the count of large weights that belong to a given readout structure (y-axis) and that 'read' neurons that belong to a given zone (x-axis). The clearer the zone is, the more weights lie in that zone. A diagonal light ridge implies that information is located where it was injected, whereas a rather homogeneous coloration indicates that information is spread out over the network. In figure 3, three different plots are shown. They correspond to delays of 2ms, 20ms and 50ms. Although cross-correlation plots of the figure 2 (right) indicate that information is still extractable up to more than 100ms, the plots corresponding to the 20ms and 50ms delay do not show any significant diagonal trace, compared to the one clearly present for the 2ms delay. The information therefore diffuses in the network without being predominantly stuck in local reverberative loops.

3.3 Auto-Correlation versus Cross-Correlations

A step towards a deeper analysis of the characteristics of the network is to have a look at the full cross-correlation curves instead of only sampling the values for the trained delay as done in figure 2 (right). In what follows, only one

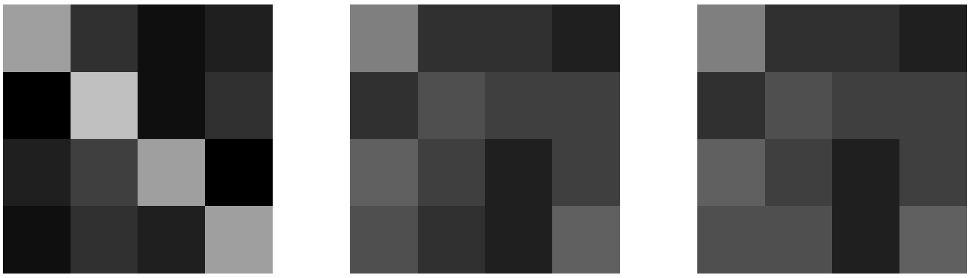


Fig. 3. Greyscale plots giving insight to where the information is (see text). Light areas indicate regions where a given readout structure extracts most of the information. Diagonal clear squares signify information stays where it has been injected (as seen on the left for a delay of 2ms) whereas rather uniform colour distribution means information is spread out among different network regions (as seen for the 20ms (middle) and 50ms (right) delays.

single input with a short auto-correlation profile excites the network. In figure 4, the thick curves correspond to the cross-correlations between the input and the outputs of readouts trained for different delays. If we have to guess what was the trained delay was by only looking at the input and the output of the network, we might be tempted to say it corresponds to the location of the peak of the cross-correlation curve. This would in fact give us an incorrect answer. A quick examination shows us that the curves do not peak around values that correspond to the trained delays. Yet another important observation is that for a given shift, the highest curve for this shift correspond to the readout trained for a delay equal to that shift. Although it is in apparent contradiction, the explanation is rather simple. Because the weights are chosen such that they minimise the error $L2$ between the output and the target, they use all the available information. On the one hand they will set their guess by looking in the past, but on the other hand they will also use the fact that the auto-correlation of the target has a certain width. Reading the present and saying this was the past is not completely incorrect, because of this auto-correlation. Minimising the error $L2$ is equivalent to finding the optimal balance between extracting degraded past information and online reading of the present input (correlated with the input some time T ago). Performances are then strongly dependent on the underlying statistics of the target.

3.4 Conclusion

We have seen that a sparsely-connected recurrent neural network tuned to be in an asynchronous irregular firing regime has enough degrees of freedom for simple output structures to extract simultaneously informations coming from independent sources, even though patterns are superimposed. This type of

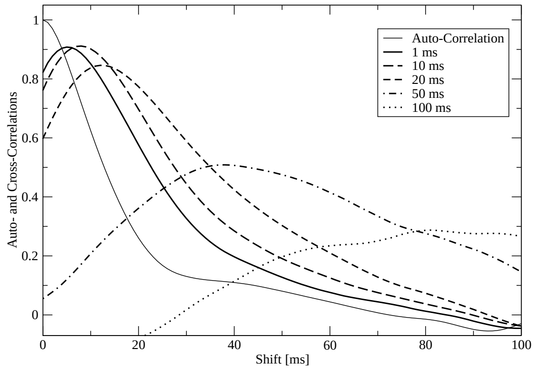


Fig. 4. Auto-correlation curve (thin line starting at the point(0,1)) of the input signal and cross-correlation curves target-output. Note that the cross-correlation curves peak at a shift value that is not the trained delay (eg for the dot-dashed curve; the trained delay is 50ms but the curve peaks around 37ms). For a given shift though, the highest curve has the correct corresponding delay (eg for a 50ms shift, the highest curve is the one trained for a 50ms delay). These curves are in this perspective optimal.

lateral spread of depolarization has been observed in cortical optical imaging experiments. We also have seen that statistics of the targets play an important role in the performances of the network. In order to minimise the distance to the target, an optimal trade-off is made between extracting old corrupted information and retrieving not-so-old information correlated in time to the desired target.

References

[1] Maass, W., Natschläger, T. and Markram, H.: Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural. Comp. **14(11)** (2002) 2531–2560

[2] Maass, W., Natschläger, T. and Markram, H.: Computational models for generic cortical microcircuits. In J. Feng, editor, Computational Neuroscience: A Comprehensive Approach. CRC-Press, 2002. to appear

[3] Jäger, H.: Short term memory in echo state networks. GMD Report 152, German National Research Center for information technology, 2002.

[4] Brunel, N.: Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. J. Comp. Neuro. **8** (2000) 183–208

[5] Brunel, N. and Hakim, V.: Fast global oscillations in networks of integrate-and-fire neurons with low firing rates. Neural. Comp. **11** (1999) 1621–1671