Topological Choices, Sliding Thresholds, and STDP Learning Variants Can Make Reservoir Computing Appropriate For Spatio-Temporal Pattern Recognition

Hananel Hazan and Larry M. Manevitz
{hhazan01, manevitz}@cs.haifa.ac.il
Department of Computer Science
University of Haifa

***Abstract*—Humans are known to be even better at temporal pattern recognition than static pattern recognition. On the other hand, artificial systems are much better in the reverse state; and in fact, much artificial temporal pattern recognition is performed by transforming time into space in one fashion or another. This is quite evident in artificial neural networks and related methods, which stands in large contrast to the known temporal properties of biological neurons. In this paper we discuss how abstractions of known biological temporal properties in the basic model can allow for good spatio-temporal pattern recognition using methods that are essentially spatio-temporal. In particular, we look at several versions of abstractions of biological neurons and their use in various biologically motivated configurations of the liquid state machine (LSM). We show how (1) the generalization capability of the liquid state machine is greatly improved by adding a sliding threshold to the leaky integrated and fire neurons and a learning mechanism related to spike-timing-dependent plasticity (STDP) to training the connectivity of the liquid. A novel and alternative training method ASTDP ("anti-STDP") is also investigated. (2) Robustness of the network to internal damage or noise is greatly improved by adding topological constraints to the random connectivity of the network . This makes the network fault tolerant and thus restores its plausibility as a model of biological activity. Finally, we also point out, that in this way, the network maintains the firing-order based encoding method discovered by Marom et al in natural neuronal networks.**

Introduction And Background

Most applications and classical theory of artificial neurons and networks are based on the 1948 McCullough and Pitts abstraction of neurons (with slight extensions, e.g. to sigmoidal non-linearities.) However, since this abstraction has no element of time; it is not surprising that these networks are more appropriate for static applications (e.g. pattern recognition or associative memories) than for dynamic spatio-temporal patterns. On the other hand, neurophysiologists and their modelers have long been aware of the many different and extensive temporal aspects of biological neurons (e.g. dynamic voltage, history dependent channels in neuronal gates, dynamic thresholds, and more recently dynamic synapses as well as the more classical synaptic plasticity). Modelers have been aware for some time that many aspects of mental processing depend precisely on these aspects; and so it makes sense for modelers to investigate these dynamic properties.

In the last ten years, several groups have made an advance in this direction by suggesting an innovative model Liquid State Machines or Reservoir computing that uses dynamic properties of a fixed network to store spatio-temporal patterns and there was some hope that this sort of model can help explain human capabilities. In particular, simple examples of spatio-temporal pattern recognition (such as simple spoken word recognition) were shown to be possible.

We have recently been looking at these models with several goals in mind:

- Judging the biological plausibility of the models:
  - By seeing how well the models work for generalization.
  - Seeing how fault-tolerant the models can be made (a sine qua non for explanatory biological models).
- Seeing how useful these ideas are for applications, i.e. we have looked at data based modeling of voxels in fMRI studies (i.e. without an a priori HRF model) and voice recognition.

Maass and Jaeger introduced Reservoir computing (Liquid State Machines [1–5] and Echo-State Machines [6–8]) as a natural way to create a system that has temporal properties. The liquid separates input signals, because of its recurrent feedback properties into temporal long-lasting patterns while standard trained detectors serve to identify the patterns. However, partially because of the intrinsic recurrent nature of the systems we find that the system is fragile in the sense that small damages can cause the system to diverge wildly from its trained behavior, thereby limiting its effectivity for practical applications. In [9], [10] we argued that this also limits its explanatory ability for biological systems.

We have approached this problem in several directions:

A. To reduce the sensitivity of the system, we modify the topology of the liquid.
B. To make the system maintain a signal for a long time, we have added sliding thresholds; which on the one hand, helps keep the feed-back from exploding while, importantly, keeps the network sufficiently active over time to maintain the memory of the signal as long as desired.
C. For cyclic input, we found this methodology satisfying.
D. For non-cyclic one-time input, the system was not reliable under any degree of noise (either in the liquid or in the input) and thus also had extremely limited ability to recognize generalized patterns.
E. Adding Hebbian learning in the form of either STDP or a novel "anti-STDP" restored the resilience of the network to noise. This worked well only in conjunctioin with a “sliding threshold” mechanism (explained below).
F. We were also curious how the system would work with a dynamic synapse [11–13] arrangement, which means the underlying network is continually changing. We investigated this in the simplest situation, where the synapses become inactive for a time period after sufficient reception of spikes. Somewhat to our surprise, the systems above also work in this setting although the training data needs to be increased about four-fold.
G. Under the arrangement of dynamic synapses, following Marom et al [14] we see that order based firing was sufficient for pattern recognition. This somewhat augments Marom et al's claim that this is a naturally emergent encoding of signals.
H. We also discovered that the STDP addition makes the network more sensitive without losing robustness. This means that a smaller network can replace a larger one.

As a first application together with additional colleagues, we have shown [15] that we can train an artificial "voxel" in a model-free fashion solely from fMRI data in a cognitive task to respond with a BOLD-type signal. This voxel response is produced by replacing the detector in an LSM with a trained generator hooked up to the liquids described above.

As a second application, we show that over a dynamic synapse liquid, the order based firing information in a window is sufficient to identify input spatio-temporal patterns. This further verifies the results of Marom et al [14] in an artificial setting; which strengthens the hypothesis that the order-based encoding is a natural emergent property.

Basic Background And Summary Of Results

In slightly earlier work [9], [10] it was shown that the original models of Maass and Jaeger [2], [16], [17] are not in fact, fault-tolerant; and that small errors in the network itself causes immediate degradation in the networks functioning; and thus generalization properties are not sufficient. However, we did discover that under certain topological constraints (such as small-world graph topologies) the network can function appropriately. In this paper, we focus on how the generalizability of spatio-temporal patterns recognizers can be substantially strengthened by adding certain capabilities to the underlying neurons. We have investigated this in variants of basic neurons (such as Izhikevich [18] and LIF (leaky integrate and fire) neurons) and under both cyclic patterns and one-time patterns. (In the latter case we are also interested in the length of time the memory of the pattern is maintained in the liquid). After extensive experimentation, we found that the best results are obtained if a sliding threshold, determined by recent history of the neuron, and the network is itself adaptive with a learning rule suggested by biological STDP (a form of a Hebbian learning rule). Note that this is different from the other presentations of this sort of model in the literature (e.g. [11], [19], [20]). An alternative also giving good results was a somewhat novel “anti-STDP” learning rule. In these cases the memory can be maintained almost indefinitely, and the generalizability recognizing ability of the networks are substantially increased in the paper below, we make these issues precise and we report on our experimental findings.

Applications:

We have shown with other colleagues that these systems can be used to produce data trained models of active voxels from fMRI scans (i.e. BOLD signal production) [15]. Furthermore the generated models of such voxels can be used to identify “relevant” voxels involved in cognitive tasks. We have also done some preliminary investigations of spoken word identification using these models and found, surprisingly, that the networks can generalize even between different speakers. Because of space restrictions, the applications will be only briefly described in this paper.

Experimental Set-Up

We ran all experiments on an LSM style set-up with the following variations:

A. The networks had 250 neurons.
B. The detector was a three level standard feed-forward neural network trained with back-propagation.
C. The neurons in the liquid were initially of leaky integrate and fire neurons (with parameters leak rate of 0.6, with connectivity weights of 0.2, threshold = 30, and each neuron was connected on the average to 20% of the liquid) and then modified as described below to have sliding thresholds. Similar experiments were run with Izhikevich neurons [11], [18] and qualitatively similar results.
D. The liquid was connected with various topologies as described below. The best result for topology was one we call "directional" and we pursued the later results with this topology.
E. We then added sliding thresholds as described below to the neurons in the liquid,
F. Since results with non-cyclic (i.e. one time presentation) of input were not satisfactory), we added various learning techniques to the connections in the liquid and measured resilience of the network.
G. We then added dynamic synapses to the neurons in the liquid and investigated the capabilities under the learning.
H. We then saw if the detectors could still recognize patterns when they were trained solely with first order firing patterns in the windows

Topology Selection

We performed experiments with different architectures. In this paper, we suppress this search as it appears elsewhere [9], [21] and all the experiments are posted at our laboratory website[1]. For this paper, the most significant ones are "hub" or "directional" and "small worlds" (types C and D below) and we report our results mostly on these topologies. The significant fact is that we now have a topology based generally on the idea of creating bottlenecks between parts of the network to limit the effect of errors during recurrency feedback. (We include results in the tables as well for some others topologies for comparison.).

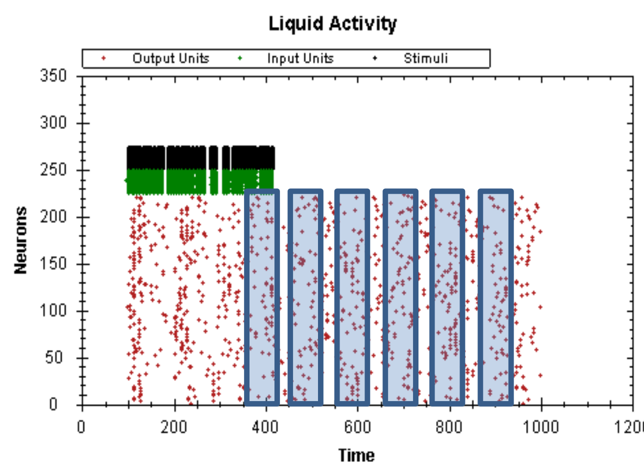

Figure 1 Example of Liquid Activity, Blue Square are the slice window of the output to the readout unit

A. Random Connectivity as a baseline. (Note: this is not a "straw dog". This is actually the basic definition of the LSM.)
B. Varying the amount of connectivity in the liquid. Lowering the average degree of connectivity shows decreased sensitivity in all architectures. Unfortunately, lowering the connectivity also decreases the strength the network has in representability and, importantly, in the persistence of the signal. (That is, a low degree of connectivity causes the activity to die down quickly because of the lack of feedback. Thus the network is bounded in time and cannot recognize an "older" input signal.) Thus we see, as is to be expected from the analysis in [1–3], [5], [7], [8], [22], [23] that a higher connectivity gives a larger set of "filters" that separate signals, but on the other hand makes it more sensitive to changes. In any case, even with low connectivities, the random topology was not robust; nor was the Maass topology [2], [8], [22]. (While not at random levels of identification, as we have seen it suffered very substantial decays with even small amounts of damages. In addition, our experiments with connectivities below 15% - 20%, show that the networks do not maintain the trace for very long. (Not shown here.)
C. "Hub" topologies. Here we designed by hand topologies with essentially one hub. In this case, the robustness was substantially increased but the persistence was weak; and under the algorithm chosen, there were substantial disconnected components in the liquid. In this paper this is also called "directional"
D. Finally, we designed an algorithm to allow distinct input and output connectivity but both obeying the same power law and Small world topologies. (See algorithm 1 and algorithm 2 below).

[1] http://cri.haifa.ac.il/neurocomputation

```
  Algorithm 1: Generate a random
number between min and max value
with Power law distribution
Input: min,max, size
How_many_numbers
counterArry = array
Magnify = 5
for  i = 1  to How_many_numbers
 index = random(array.start ,
array.end)
 end_array = array.end
 candidate = array[index]
 AddCells(array , Magnify);
 for t = 0 to Magnify
  array[end_array+t]=candidate
 end for
 shuffle(array)
 output_Array[i] = candidate
 counterArry[candidate]++
end for
shuffle(counterArry)
Output output_Array,counterArry
```

```
  Algorithm 2: Create the
connectivity matrix for the liquid
network using the algorithm 1
Input weight_Matrix
use algorithm 1 to creart
(arraylist, counterArry)
counter = 0
for i=1 to counterArry.lenght
 for t=1 to counterArry[i]
  weight_Matrix[i,
arraylist[counter]]=true
  counter++
 end for
end for
```

One problem with the various algorithms for designing power law connectivity is that under a "fair" sampling, the network might not be connected. This means that such a network actually has a lower, effective connectivity. Since we already knew that lower connectivity results in less sensitivity to noise, we decided to eliminate this problem by randomly connecting the disconnected components (either from an input or output perspective) to another neuron chosen randomly but proportionally to the connectivity. (This does not guarantee connectivity of the graph, but makes it unlikely, so that the effective connectivity is not substantially affected.)

The Learning / Testing Process

Initially we start with connecting 250 units according to the topology chosen. The global connectivity of the entire network is 20% of its size; each connection initially has a weight of 0.25 which corresponds to ¼ of the threshold needed for firing.

We start by running the network without any input for duration of 100 iterations; then we start the input to the net as described above. From 1000 iterations of activity the readout units received 6 window slices of 50 iterations each; from designated neurons that were marked as output neurons (see for example Figure 1). We tested whether the detector identified the pattern correctly at each window.

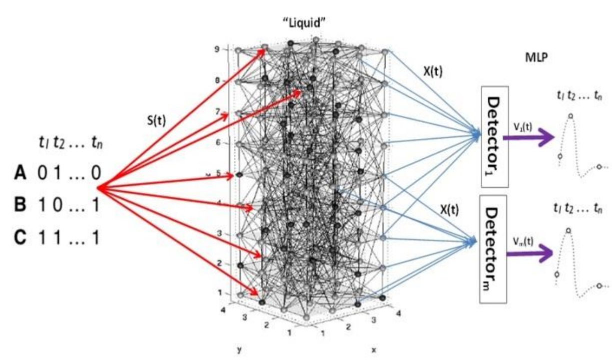

Figure 2: Diagram of the Liquid / Echo State Machine with the input afferents and read out detectors. In our work, the elements of the liquid have been examined with different properties as has the connectivity and strength between the members of the liquid.

In our implementation of the learning in the Liquid / Echo State machine we used some adjustments to biological learning rules. For example according to the Hebbian rule "those who fire together, wire together". In our model we implement a learning phase that uses a Spike Timing Dependent Plasticity (STDP) [24] hat is a temporally asymmetric form of Hebbian learning induced by tight temporal correlations between the spikes of pre- and postsynaptic neurons. The synapses increase their efficiency if the synapse persistently causes the postsynaptic target neuron to generate action potentials.

In our model we actually successfully used two kinds of simplified Hebbian rules: STDP and Anti STDP, with each one we implemented as short version and the "standard" version (see below).

Since we obtained good results for the cyclic input, we used the STDP/ASTDP rules both short and "standard" only in the

Non-Cyclic (one-time) Input test. The rationale is: when the input to the network ceases, damages (either dead neuron that failed to respond or noise inside the input), the error is amplified with every iteration of the network. Therefore we needed a mechanism that will be able to learn or to amplified the relevant connection between neurons to maintain the right signal or to correct it somehow.

Sliding Threshold

The problem addressed here was that the neurons in the liquid do not regulate their activity and as can be seen in Figure 3 the activity dies out quickly once input ceases. The sliding threshold [13] habituates the firing making the neuron harder to fire once it is firing at about 60% of its maximal firing capacity. In addition, the sliding threshold is adjusted in the other direction thus making the neuron more sensitive to firing when the rate of spikes reaching it is very low. See Figure 3 compare with Figure 4.

The problem with the sliding threshold is that it makes the neurons much more sensitive to noise in the liquid itself or in the input to the liquid. (The results are random (about 50% recognition) in almost all cases of even small noise. Because of this we needed another method that will strengthen the relevant and important connections between the neurons. That this is why we use in addition the STDP rule (see below).

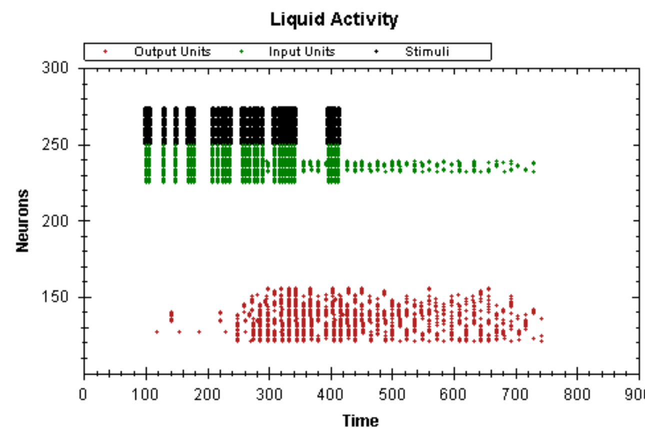

Figure 3, Liquid activity without STDP and without sliding threshold. We can see that the activity did not persist for all 1000 iteration; furthermore it's only on a small part of the liquid.

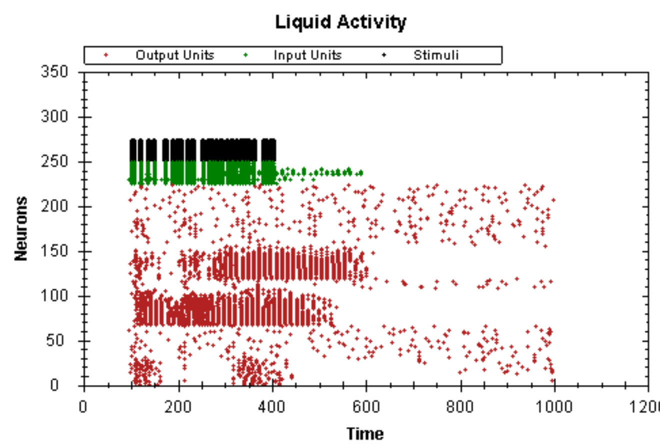

Figure 4 Liquid activity with the same parameters and the same input from Figure 3 but with sliding threshold and without STDP

Cyclic Input Results

In this case the input to the liquid was cyclic; therefore it reinforced the signal in the liquid and became somewhat "resonant" in the liquid. By "cyclic" we mean that a spatio-temporal pattern was continually repeated. As a result, it was able to modify the effect of damages inside the liquid; and thus the liquid remained robust to damages. (Note that this was only true under certain topologies.) In other words the damage as compared to the input wasn't strong enough to create a feedback reaction and the input to the liquid immediately overcomes the "damage" signal. (See Table 1, Table 2) under the topologies of "directional" and "small worlds".

Input generalization recognition

Topology	Generalization Recognition Ability Versus Topology				
	Variability / Differences				
	1%	3%	5%	7%	10%
Random	0.95	0.90	0.89	0.86	0.86
"directional" style network	0.97	0.97	0.96	0.96	0.95
Maass	0.93	0.87	0.84	0.85	0.85
Small worlds	0.98	0.97	0.97	0.96	0.97

Table 2 Liquid performance on cyclic input

Noise damage

Topology	Recognition Ability Versus Topology				
	Variability / Differences				
	1%	3%	5%	7%	10%
Random	0.61	0.55	0.54	0.53	0.52
"directional" style network	0.86	0.77	0.7	0.65	0.62
Maass	0.55	0.53	0.51	0.51	0.49
Small worlds	0.88	0.74	0.72	0.68	0.64

Table 1 Liquid performance on cyclic input.

Non-Cyclic Failure Results

Under the same set-up as the above but with one-time non-cyclic input, we find that the network has very poor fault tolerance (see Table 3) and poor generalization abilities (See Table 4.)

Noise damage

Topology	Recognition Ability Versus Topology				
	Variability / Differences				
	1%	3%	5%	7%	10%
Random	0.53	0.54	0.53	0.5	0.51
“directional” style network	0.51	0.52	0.51	0.5	0.5
Maass	0.51	0.51	0.53	0.52	0.52
Small worlds	0.51	0.53	0.51	0.51	0.52

Table 3 Liquid performance on non-cyclic input with Anti STDP and sliding threshold

Input generalization recognition

Topology	Generalization Recognition Ability Versus Topology				
	Variability / Differences				
	1%	3%	5%	7%	10%
Random	0.7	0.58	0.57	0.58	0.61
“directional” style network	0.79	0.72	0.68	0.7	0.71
Maass	0.74	0.63	0.61	0.62	0.63
Small worlds	0.72	0.6	0.61	0.61	0.62

Table 4 Liquid performance on non-cyclic input with Anti STDP and sliding threshold

STDP And Anti STDP And Non-Cyclic Patterns

In order to make the liquid capable of handling the non-cyclic input, before we modify the readout neurons on the signal we first create a study phase. In that phase the network received the input one time (as in the testing phase) but during this phase the synapses are allowed to change according to the STDP rule (see: Figure 6: STDP rule)

Each neuron increases or decreases the weight between pre and post synaptic neuron according to the time that it fires: If the post neuron fires, it checks how close it is to pre synaptic firing, and increases its magnitude of the weights in proportion to the time it elapsed (i.e. a neuron that fires relatively close to the pre synaptic firing receives more weight than synapse that transferred a spike a long time ago but still in the window time frame of change).

The different between the “standard” and the short is that the standard models takes all history of firing from each neuron and changes the weights accordingly, and the short version only takes into account the last firing from the pre synaptic neuron. For example Figure 6

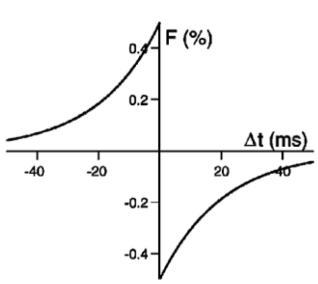

Figure 6: STDP rule[2]

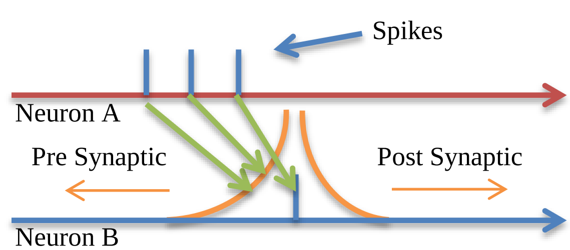

Figure 5: Pre and Post synaptic firing. The Standard model taken in to accounts all three spiking and change the weight between neuron A to B accordingly. The short model only takes the last spike and changes the weight according to the only last spike.

[2] The figure taken from [24]

We also use as a variant a different STDP that is illustrated in Figure 7 and Figure 6. The goal of this "anti"- STDP is to perform similarly to the standard STDP effect of spikes that comes before and weakens the spikes that come after.

Secondly, both rules decrease the effect of presynaptic neurons that fire after the postsynaptic firing, again in relation to the time of firing (i.e. for STDP the effect of a presynaptic neuron that fires slightly after a post synaptic firing neuron decreases under STDP more than one that fires a long time before the post synaptic firing, but still in the window time frame of change) whereas for "anti" STDP this is reversed.

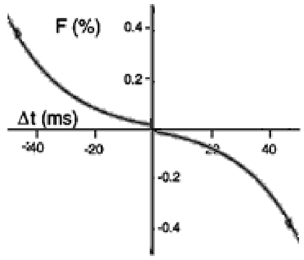

Figure 7: Anti - STDP rule[3]

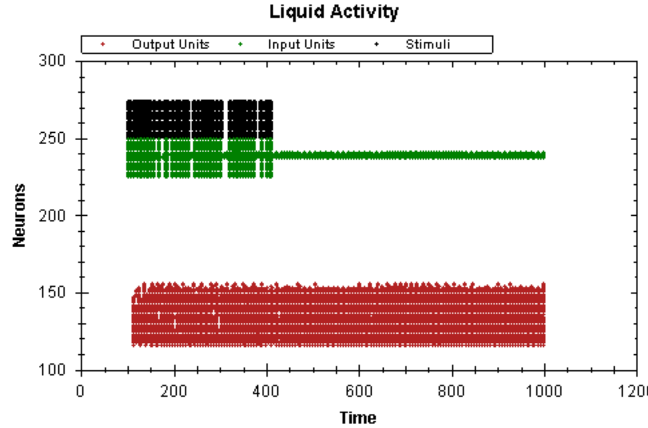

Figure 8 Liquid activity with the same parameters and the same input from Figure 3 but with STDP only

We note here that, as can be seen in Figure 8, the result of activity as a result of using a liquid with only STDP and without the sliding threshold is that we have long activation of the liquid, but the activation is not easily separable for classification or very hard to do any separation because the activity is monotonic and repeated throughout the time course.

Figure 9 shows the result of using the best arrangement with the same arrangement as Figure 3 but using both the sliding threshold and the STDP.

Table 5 and Table 6 show the final results of this paper; using both STDP and the sliding threshold.

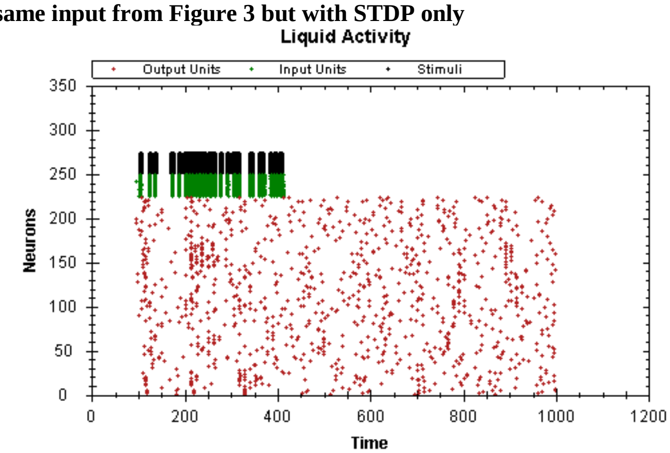

Figure 9 Liquid activity with the same parameters and the same input from Figure 3 but with STDP and sliding threshold

Dead Neurons damage

Topology	Recognition Ability Versus Topology				
	Variability / Differences				
	1%	3%	5%	7%	10%
Random	0.80	0.74	0.72	0.69	0.66
"directional" style network	0.87	0.76	0.72	0.66	0.64
Maass	0.83	0.74	0.72	0.70	0.66
Small worlds	0.82	0.74	0.71	0.68	0.65

Table 5 Result on non-cyclic input with standard STDP and sliding threshold

Damage on the input – Generalization recognition

Topology	Generalization Recognition Ability Versus Topology				
	Variability / Differences				
	1%	3%	5%	7%	10%
Random	0.92	0.87	0.84	0.85	0.86
"directional" style network	0.94	0.89	0.89	0.89	0.88
Maass	0.92	0.86	0.86	0.84	0.85
Small worlds	0.93	0.87	0.85	0.80	0.82

Table 6 Result on non-cyclic input with standard STDP and sliding threshold

[3] The figure is modified from Figure 6: STDP rule [24]

Dynamic Synapses And Pattern Recognition

Following the work of [14] we wondered if the ideas presented above would also persist when the network has dynamic synapses. We tested this only for the simplest case where synapses become unavailable after frequent firing for a set period of time. Somewhat to our surprise, the system retained its resilience to noise and its generalization ability as long as we used both the sliding threshold and the STDP properties. However, the training time for STDP had to be lengthened to five times as long.

Dead Neurons damage

Topology	Generalization Recognition Ability Versus Topology				
	Variability / Differences				
	1%	3%	5%	7%	10%
Random	0.82	0.76	0.70	0.70	0.66
"directional" style network	0.85	0.76	0.71	0.69	0.64
Maass	0.85	0.75	0.69	0.66	0.67
Small worlds	0.80	0.73	0.66	0.65	0.64

Table 7 Dynamic Synapses. Results for non-cyclic input with standard STDP and sliding threshold

Damage on the input – Generalization recognition

Topology	Generalization Recognition Ability Versus Topology				
	Variability / Differences				
	1%	3%	5%	7%	10%
Random	0.92	0.85	0.86	0.83	0.84
"directional" style network	0.94	0.87	0.88	0.85	0.88
Maass	0.91	0.86	0.84	0.85	0.84
Small worlds	0.90	0.82	0.81	0.79	0.79

Table 8 Dynamic Synapses. Results for non-cyclic input with standard STDP and sliding threshold

Dynamic Synapses And Order Based Representation

One of the main insights of Marom et al [14] was that under natural biological networks order based pattern encoding is sufficient for pattern identification, and it is claimed, an emergent property of the system. We tried the same attempt for our best system and did in fact show that this works both for static synapses and dynamic synapses.: Table 9 and Table 10 show the result for static synapses networks while Table 11 and Table 12 shows the results for dynamic synapses. Thus these results help support the claims of [14]

Dead Neurons damage

Topology	Recognition Ability Versus Topology				
	Variability / Differences				
	1%	3%	5%	7%	10%
Random	0.68	0.64	0.61	0.59	0.57
"directional" style network	0.73	0.62	0.58	0.57	0.55
Maass	0.67	0.62	0.57	0.59	0.56
Small worlds	0.65	0.61	0.60	0.57	0.55

Table 9 Result on non-cyclic input with standard STDP and sliding threshold

Damage on the input – Generalization recognition

Topology	Generalization Recognition Ability Versus Topology				
	Variability / Differences				
	1%	3%	5%	7%	10%
Random	0.83	0.73	0.7	0.72	0.66
“directional” style network	0.83	0.78	0.7	0.73	0.69
Maass	0.83	0.74	0.71	0.71	0.69
Small worlds	0.84	0.74	0.68	0.68	0.69

Table 10 Result on non-cyclic input with standard STDP and sliding threshold

Dead Neurons damage

Topology	Recognition Ability Versus Topology				
	Variability / Differences				
	1%	3%	5%	7%	10%
Random	0.67	0.65	0.59	0.59	0.56
“directional” style network	0.67	0.63	0.60	0.55	0.56
Maass	0.67	0.64	0.59	0.58	0.57
Small worlds	0.66	0.62	0.59	0.56	0.55

Table 11 Result on non-cyclic input with standard STDP and sliding threshold

Damage on the input – Generalization recognition

Topology	Generalization Recognition Ability Versus Topology				
	Variability / Differences				
	1%	3%	5%	7%	10%
Random	0.84	0.74	0.70	0.71	0.69
“directional” style network	0.83	0.75	0.74	0.72	0.75
Maass	0.84	0.75	0.71	0.70	0.72
Small worlds	0.81	0.73	0.68	0.68	0.64

Table 12 Result on non-cyclic input with standard STDP and sliding threshold

Summary

We have investigated the LSM under variations designed to make the system applicable for spatio-temporal pattern identification both under cyclic and non-cyclic input. We discovered that this can be done under the choice of a "directional" topology, and with a sliding threshold and a learning method applied to the connections in the liquid.

References

[1] W. Maass, T. Natschläger, and H. Markram, “A Fresh Look at Real-Time Computation in Generic Recurrent Neural Circuits,” *submitted for publication*, 2002.

[2] W. Maass, T. Natschläger, and H. Markram, “A Model for Real-Time Computation in Generic Neural Microcircuits,” *Proc. of NIPS 2002*, vol. 15, pp. 229-236, 2002.

[3] W. Maass and H. Markram, “On the Computational Power of Recurrent Circuits of Spiking Neurons,” *Journal of Computer and System Sciences*, vol. 69(4), no. 4, pp. 593-616, 2004.

[4] T. Natschläger, W. Maass, and H. Markram, “The ‘Liquid Computer’: A Novel Strategy for Real-Time Computing on Time Series,” *Special Issue on Foundations of Information Processing of TELEMATIK*, vol. 8, no. 1, pp. 39–43, 2002.

[5] W. Maass, T. Natschläger, and H. Markram, “Computational Models for Generic Cortical Microcircuits,” in *Computational Neuroscience: A Comprehensive Approach*, J. Feng, Ed. CRC-Press, 2002.

[6] H. Jaeger, “Adaptive nonlinear system identification with echo state networks,” 2002.

[7] H. Jaeger, “Short term memory in echo state networks,” German National Research Center for Information Technology, GMD Report 152, 2001.

[8] H. Jaeger, “The ‘echo state’ approach to analysing and training recurrent neural networks,” German National Research Center for Information Technology, GMD Report 148, 2001.

[9] L. Manevitz and H. Hazan, “Stability and Topology in Reservoir Computing,” in *Advances in Soft Computing*, vol. 6438, G. Sidorov, A. Hernández Aguirre, and C. Reyes García, Eds. Springer Berlin / Heidelberg, 2010, pp. 245-256.

[10] H. Hazan and L. M. Manevitz, “The Liquid State Machine is not Robust to Problems in Its Components but Topological Constraints Can Restore Robustness,” in *IJCCI (ICFC-ICNC)*, 2010, pp. 258-264.

[11] E. M. Izhikevich, “Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling,” *Cerebral Cortex*, vol. 17, no. 10, pp. 2443 -2452, Oct. 2007.

[12] Ö. B. Artun, H. Z. Shouval, and L. N. Cooper, “The effect of dynamic synapses on spatiotemporal receptive fields in visual cortex,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, no. 20, pp. 11999 -12003, 1998.

[13] S. Song, K. D. Miller, and L. F. Abbott, “Competitive Hebbian learning through spike-timing-dependent synaptic plasticity,” *Nat Neurosci*, vol. 3, no. 9, pp. 919-926, 2000.

[14] G. Shahaf, D. Eytan, A. Gal, E. Kermany, V. Lyakhov, C. Zrenner, and S. Marom, “Order-Based Representation in Random Networks of Cortical Neurons,” *PLoS Comput Biol*, vol. 4, no. 11, Nov. 2008.

[15] P. Avesani, H. Hazan, E. Koilis, L. Manevitz, and D. Sona, “Learning BOLD Response in fMRI by Reservoir Computing,” in *2011 International Workshop on Pattern Recognition in NeuroImaging (PRNI)*, 2011, pp. 57-60.

[16] T. Natschläger and W. Maass, “Spiking Neurons and the Induction of Finite State Machines,” *Theoretical Computer Science: Special Issue on Natural Computing*, vol. 287, pp. 251–265, 2002.

[17] W. Maass and H. Markram, “Temporal Integration in Recurrent Microcircuits,” in *The Handbook of Brain Theory and Neural Networks*, 2nd ed., M. A. Arbib, Ed. MIT Press (Cambridge), 2002.

[18] E. M. Izhikevich, “Simple model of spiking neurons,” *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569-1572, Nov. 2003.

[19] C. Clopath and W. Gerstner, “Voltage and spike timing interact in STDP – a unified model,” *Front. Syn. Neurosci.*, vol. 2, p. 25, 2010.

[20] J.-C. Zhang, P.-M. Lau, and G.-Q. Bi, “Gain in sensitivity and loss in temporal contrast of STDP by dopaminergic modulation at hippocampal synapses,” *Proceedings of the National Academy of Sciences*.

[21] H. Hazan and L. M. Manevitz, “Topological constraints and robustness in liquid state machines,” *Expert Systems with Applications*, vol. 39, no. 2, pp. 1597-1606, Feb. 2012.

[22] W. Maass, T. Natschläger, and H. Markram, “Real-time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations,” *Neural Computation*, vol. 14, no. 11, pp. 2531-2560, 2002.

[23] W. Maass, R. A. Legenstein, and H. Markram, “A New Approach towards Vision suggested by Biologically Realistic Neural Microcircuit Models,” in *Proc. of the 2nd Workshop on Biologically Motivated Computer Vision*, 2002.

[24] S. Song, K. D. Miller, and L. F. Abbott, “Competitive Hebbian learning through spike-timing-dependent synaptic plasticity,” *Nat Neurosci*, vol. 3, no. 9, pp. 919-926, 2000.