2012 IEEE 27-th Convention of Electrical and Electronics Engineers in Israel

Temporal Pattern Recognition via Temporal Networks of Temporal Neurons

Alex Frid
Edmond J. Safra Brain Research Center for the Study of Learning Disabilities,
University of Haifa, Israel
and
Department of Computer Science,
Tel-Hai Academic College,
Upper Galilee, Israel
alex.frid@gmail.com

Hananel Hazan
Department of Computer Science
University of Haifa, Israel
hhazan01@cs.haifa.ac.il

Larry Manevitz
Department of Computer Science,
University of Haifa, Israel
and
The Rotman Research Institute at Baycrest, Toronto, Canada
manevitz@cs.haifa.ac.il

***Abstract*— We show that real valued continuous functions can be recognized in a reliable way, with good generalization ability using an adapted version of the Liquid State Machine (LSM) that receives direct real valued input. Furthermore this system works without the necessity of preliminary extraction of signal processing features. This avoids the necessity of discretization and encoding that has plagued earlier attempts on this process. We show this is effective on a simulated signal designed to have the properties of a physical trace of human speech. The main changes to the basic liquid state machine paradigm are (i) external stimulation to neurons by normalized real values and (ii) adaptation of the integrate and fire neurons in the liquid to have a history dependent sliding threshold (iii) topological constraints on the network connectivity.**

***Index Terms*—Liquid State Machine (LSM), Classification, Signal Processing, Temporal Networks.**

I. Introduction

Recently, we showed that liquid state machines can be adapted to give robust pattern recognition of temporal patterns [1], [2]. However, under this idea, complex continuous real valued patterns seem to require a discretization and digital encoding in order to input them to the liquid. In attempting to apply this technique to signal processing on phoneme recognition from a continuous voice signal, we found this to be intractable; because of problems in both making the system accurate while maintaining good generalizability properties. It seems that the combination of the separability of the liquid with the digital encoding conflict with the generalizability.

In this paper, we show how to rectify this problem, by making three fundamental changes in the system. (i) We input the signal directly to the neurons as normalized real values (instead of already encoded discrete "firings"). (ii) We modify the integrate and fire neurons in the liquid to have history dependent sliding thresholds. (iii) We maintain the small world topology constraints as described in [1], [2] in order to get more diverse activity in the firing patterns of the neurons.

A. The Liquid State Machine

The liquid state machine is a recurrent neural network. In its usual format [3], [4], each neuron is a biologically inspired artificial neuron such as an "integrate and fire" (LIF) neuron, Hodgkin-Huxley [5] an "Izhikevich" style neuron [6]. The connections between neurons define the dynamical process, and the recurrence connections define what we call the "topology" in this paper. The properties of the artificial neurons, together with these recurrences, results in any sequence of history input being transformed into a spatio-temporal pattern activation of the liquid. In this work we used uncorrelated Scale Free networks [7] that have small world properties [8], [9] to create an architecture of the recurrent connections between neurons.

Its nomenclature comes from the fact that one can intuitively look at the network as if it was a "liquid" such as a pond of water, the stimuli are rocks thrown into the water, and the ripples on the pond are the spatio-temporal patterns.

The "detectors" (see Stage 4 in Fig 1) are classifier systems that receive as input a state (or in large systems a sample of the elements of the liquid) and are trained to recognize patterns that evolve from a given class of inputs. Thus a detector could be an SVM [10] or an adaline [11], percepton [12], or back propagation neural networks, etc.

As Maass et al. postulate [13], [14] the recurrent network serves as a kind of “kernel” to separate spatio-temporal patterns and in addition serves as a memory for spatio-temporal information. Thus the standard detector has no need to transform time to space, the liquid itself serve as a memory for temporal information in a natural way.

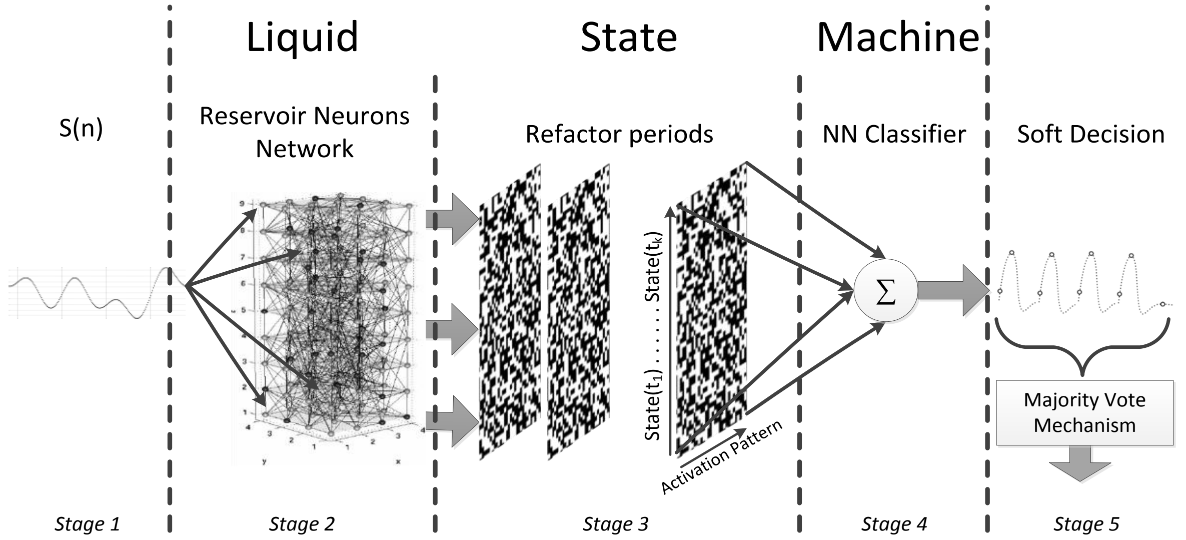

Fig 1. Diagram of the setup Liquid / Echo State Machine as used in the paper. The real valued temporal signal input into the liquid, and the classifier uses the digital firing patterns of the reservoir with consecutive iteration times (synchronized to the neuron refactor period) to classify the signal. Thus each "state" in Stage 3 has a single row (liquid state), which is a snapshot of the reservoir. Thus entry to the classifier is consecutive snapshots of the reservoirs firing pattern. Eventually the final classification of the signal S(n) is done by a weighted voting between all the soft decisions over time.

For comparison the traditional methods of actual signal classification schemes, usually consist of the following basic parts as shown in Fig 2: (1) Resampling and Quantization, (2) Windowing, (3) Feature Extraction, (4) applying classification algorithm, and then decision algorithm may applied. The goal of the first and the third steps are to reduce the data explosion that is derived from real life signals.

The second part, signal windowing, is used to attack the issue of global and local field of view on the data; in other words to incorporate time as a feature. A short window will be able to look on short temporal issues while a long period window can catch long period tendencies. However, in both cases the perception of time in this way is somewhat artificial and will never be accurate enough. In addition, the windows always need to be adjusted to a specific task. Furthermore, the process of windowing itself eliminates the option to look on all the possibilities of cross window events while trying to pass through all the possibilities of window size. Part of the problem can be solved using overlapping windows, but this has its computational cost.

The feature extraction part will always be dependent on the window size selection. Moreover it requires substantial experience with the specific data task and the success of finding good features can be affected by personal experience, extensive analysis of the data and time-frequency manipulations as described in [15]. All of these are time demanding processes, and usually computationally ineffective and somewhat not natural [16]. Their main goal is data dimensionality reduction for the classification algorithms.

The last part, the classification strictly depends on the previous one. Successful feature extraction process will result in efficient and fast learning, affect its accuracy and generalization (and the problems in achieving this goal as stated above). Moreover, as mentioned, the windowing process restricts the classifier's ability to use time as a feature.

In this work we show that using the LSM setup, a somewhat more natural way of temporal dependent data classification can be done. In addition this setup, can successfully replace all those processes of traditional classification scheme, such as windowing, feature extraction and finding sufficient classification algorithms. We also show that this task can be done with a relatively small amount of neurons. We believe it may be more suitable for real-time applications (where windowing and feature extraction can't be properly implemented). Moreover we show that our network has the ability to convert a human dependent task of the selection of signal processing features into an automated computational task of finding LSM parameters (which in turn can be optimized using parallel programming).

II. Methods

A. Liquid state configuration

Our configuration of the liquid state mechanism is illustrated in Fig 1. The Liquid network used twenty five LIF (leaky integrate and fire) neurons with 20% connectivity between neurons, arranged with a topology of uncorrelated Scale Free network [7]. Two of the neurons were inhibitory neurons and the rest (23 neurons) were excitatory neurons. The weights between neurons were set to a constant value of 0.25. The Liquid was configured to have four input neurons from Stage 1 to Stage 2 dedicated for signal transition into the network, as shown in Fig 1. All the remaining 21 neurons were considered as output neurons. The input neurons were fed by the input value and those neurons were not taken in account during the snapshot that was exported to the detectors (see activation pattern in Fig 1). The input to the liquid in our configuration was real numbers between 0 to 95 volts. The input neurons converted this current into binary firing sequences that in turn were propagated in the liquid.

However, under the standard LSM arrangement, the "neurons" of the liquid receive input by external “spikes". Thus to apply this method to real continuous signals (such as speech), one has to first discretize and encode the signal.

This creates a conflict between the separability and the generalizability properties of the network. We indicate further the needed changes to avoid this problem by directly inputting normalized real values to the liquid.

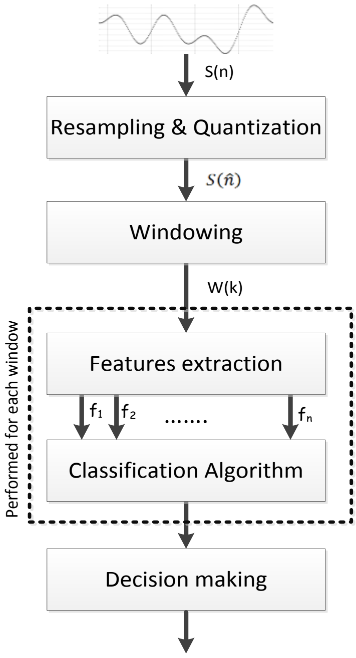

Fig 2. A general diagram of a traditional classification algorithm, where the dashed area is performed for every input window generated by the windowing procedure. In case that there is more than one classifier used a decision making algorithm can be applied.

As a show-case of this method we present an example of such a network that successfully classifies synthetically generated auditory data having relatively similar signal processing properties as human speech without any of the steps needed for the traditional classification algorithm (i.e. windowing, preliminary data analysis and features seeking and selection of classification algorithm).

Each neuron has resting state of -65mv, threshold of 30mv, refectory period after spike is -95mv and decay rate of 60%. Each individual neuron has its own sliding threshold factor, initially set randomly between zero and one, which decreases the threshold after each iteration that the neuron hasn’t fired. Once a neuron fires, the threshold is reset to 30mv. The detector received all consecutive snapshots of liquid activations as input (a single row in Fig. 1. Stage 3), where the multiple snapshots were chosen to be synchronized with the average neuron refactor period (see the division into activation windows in stage 3 in Fig. 1.), in order to maintain proper activation level of the signal input to the classifier. For long signals input, this procedure is repeated until the signal length is exhausted and the ultimate classification is a voting result of those separate classifications for each snapshot (as depicted in stage 5 of Fig. 1.).

B. Data analysis

We built up our results gradually. Starting by training the network using signals that passed through various encoded binary (spike or not spike) sequences as described in [1], [2]. This was somewhat not successful for this type of data, because despite very extensive experimentation with methods of encoding the continuous real valued signal into a discrete binary one, such as translation of signal magnitude into its matching firings number sequence, convolutional filtering and division of the signal into its sub-bands (logarithmically scaled) description of activations in each of the sub-bands separately. But we could not reach acceptable levels of generalization.

We then decided to eliminate the encoding and discretization problem on the input by modifying the network to receive its input as normalized real values (as opposed to spikes).

While this solved the encoding problem, as a result of using real values in the input instead of spikes, the activation in the liquid was very scarce. We then decided to introduce to each neuron a sliding threshold mechanism [17] to make the neuron more sensitive to the input by change the sensitivity of the neuron according to his history of firing, if a neuron fire more than 60% of its capability the threshold grow by constant value, make the neuron less sensitive to inputs, if the neuron didn’t fire long time his threshold get lower by constant value, making the neuron become more sensitivity to future inputs. The initial threshold of all neurons was set randomly between zero to 30mv.

We then continued with three tests on different variations of real valued continuous signals, each one designed to approximate more closely natural human speech signals:

1. Long harmonic signals composed of 7 base frequencies with 5% random noise. Each of the two classes has disjoint frequencies.
2. Long harmonic and 5% random noised with 5 of the 7 frequencies mutual in both sets.
3. Step (2) + all the signals had random length and random phase shift

The last test has similar properties of voiced phonemes in English.

Since we eventually succeeded in all these tasks, we report here only on the last and most interesting one.

In order to test the network, each time four hundred (400) samples from each of the two groups were generated, where 50% of the dataset randomly selected as the training set (both groups had equal representation in the training set) and the rest used as testing set. Each time the data is cross-validated 30 times and confusion matrices were calculated similar to Table I to perform the True Positive (TP) and True Negative (TN) analysis (the bold marked diagonal in Table I).

In order to show the effect of the sliding threshold, we display the results with and without the sliding threshold (as show in TABLE II) on the most successful topology (uncorrelated scale free network).

TABLE I. CONFUSION MATRIX OF THE 3RD EXPERIMENT

	Classified as group 1	Classified as group 2
Group 1	**0.91± 0.03**	0.09%
Group 2	0.06%	**0.92± 0.08**

Note that the results shown in TABLE I and TABLE II are average of cross validation on 100 trials.

TABLE II. UNCORRELATED SCALE FREE TOPOLOGY WITH AND WITHOUT SLIDING THRESHOLD

Configuration #	1	2
True positive	0.98 ±0.02	0.89± 0.14
True Negative	0.28 ±0.42	0.95± 0.11
Accuracy	0.63± 0.20	**0.92± 0.08**

General Liquid configuration:
Configuration 1:

- 25 LIFs Neurons
- 60% decay
- 20% connectivity
- 10% Inhibitory neuron
- 90% excitory neurons
- 15% of the neuron are input from external source
- The strength between neuron is 0.25.
- Threshold of each was set randomly between zero to 30mv

Configuration 2:

In addition to the previous configuration we added to all neurons a sliding threshold as described above.

III. CONCLUSIONS

This paper shows that the LSM can be modified in order to classify real valued continuous data, with different length that typically arrive in natural temporal patterns such as human speech, and to do so in a natural manner. The major modifications were (1) the direct input of real values to the liquid in the form of adjustments to the pre-spike activation level of the individual neurons and (2) the use of sliding thresholds.

Using LSM for this task has several advantages: (1) the network itself can act as encoder of the real data into its binary representation; (2) it uses a relatively small set of neurons to do so; (3) it manages to control the noise by itself, (4) it incorporates the time feature into classification and (5) it can be easily adjusted for N-group classification by adding detector for each group.

ACKNOWLEDGMENT

We thank the Caesarea Rothschild Institute of the University of Haifa for its support.

Larry Manevitz thanks the Ben and Hilda Katz Foundation for its support during a visit to the Rotman Institute at Baycrest in Toronto.

REFERENCES

[1] 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.

[2] 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.

[3] M. Lukosevicius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Computer Science Review*, vol. 3, no. 3, pp. 127–149, Aug. 2009.

[4] 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.

[5] A. L. Hodgkin and A. F. Huxley, “A quantitative description of membrane current and its application to conduction and excitation in nerve,” *J. Physiol. (Lond.)*, vol. 117, no. 4, pp. 500–544, Aug. 1952.

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

[7] M. Catanzaro, M. Boguna, and R. Pastor-Satorras, “Generation of uncorrelated random scale-free networks,” *arXiv:cond-mat/0408110*, Aug. 2004.

[8] S. Achard, R. Salvador, B. Whitcher, J. Suckling, and E. Bullmore, “A Resilient, Low-Frequency, Small-World Human Brain Functional Network with Highly Connected Association Cortical Hubs,” *The Journal of Neuroscience*, vol. 26, no. 1, pp. 63 –72, Jan. 2006.

[9] D. S. Bassett and E. Bullmore, “Small-World Brain Networks,” *The Neuroscientist*, vol. 12, no. 6, pp. 512 –523, Dec. 2006.

[10] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[11] B. Widrow and M. Hoff, “Adaptive Switching Circuits,” in *1960 {IRE} {WESCON} Convention Record, Part 4*, 1960, pp. 96–104.

[12] W. Pitts and W. S. McCulloch, “A logical calculus of the ideas immanent in nervous activity.,” *Bulletin of Mathematical Biology*, vol. 52, no. 1–2, pp. 99–115; discussion 73–97, 1943.

[13] 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.

[14] 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.

[15] A. Frid and Y. Lavner, “Acoustic-phonetic analysis of fricatives for classification using SVM based algorithm,” in *2010 IEEE 26th Convention of Electrical and Electronics Engineers in Israel (IEEEI)*, 2010, pp. 000751 –000755.

[16] 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.

[17] E. L. Bienenstock, L. N. Cooper, and P. W. Munro, “Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex,” *J. Neurosci.*, vol. 2, no. 1, pp. 32–48, Jan. 1982.