Expert Systems with Applications 39 (2012) 1597–1606

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Topological constraints and robustness in liquid state machines

Hananel Hazan *, Larry M. Manevitz

Department of Computer Science, University of Haifa, Mount Carmel, Haifa 31905, Israel

ARTICLE INFO

Keywords:
Liquid State Machine
Reservoir computing
Small world topology
Robustness
Machine learning

ABSTRACT

The Liquid State Machine (LSM) is a method of computing with temporal neurons, which can be used amongst other things for classifying intrinsically temporal data directly unlike standard artificial neural networks. It has also been put forward as a natural model of certain kinds of brain functions. There are two results in this paper: (1) We show that the Liquid State Machines as normally defined cannot serve as a natural model for brain function. This is because they are very vulnerable to failures in parts of the model. This result is in contrast to work by Maass et al. which showed that these models are robust to noise in the input data. (2) We show that specifying certain kinds of topological constraints (such as "small world assumption"), which have been claimed are reasonably plausible biologically, can restore robustness in this sense to LSMs.

1. Introduction

Processing in artificial neurons typically is a-temporal. This is because the underlying basic neuronal model, that of Pitts and McCulloch (1943) is a-temporal by nature. As a result, most applications of artificial neural networks are related in one way or another to static pattern recognition. On the other hand, it has long been recognized in the brain science community that the McCullough–Pitts paradigm is inadequate. Various models of differing complexity have been promulgated to explain the temporal capabilities (amongst other things) of natural neurons and neuronal networks.

However, during the last decade, computational scientists have begun to pay attention to this issue from the neurocomputation perspective as well, e.g. Fern and Sojakka (n.d.), Jaeger (2001a, 2001b, 2002), Lukosevicius and Jaeger (2009) and Maass, Natschläger, and Markram (2002a, 2002b, 2002d), and investigations as to the computational capabilities of various models are being investigated.

One such model, the Liquid State Machine (LSM) (see Fig. 1) (Maass et al., 2002a), has had substantial success recently. The Liquid State Machine is a somewhat different paradigm of computation. It assumes that information is stored, not in "attractors" as is usually assumed in recurrent neural networks, but in the activity pattern of all the neurons which feed-back in a sufficiently recurrent and inter-connected network. This information can then be recognized by any sufficiently strong classifier such as an Adaline (Widrow & Hoff, 1960), Back-Propagation, SVM[1] or Tempotron (Gutig & Sompolinsky, 2006). (The name "liquid state" comes from the idea that the history of, e.g. timings of rocks thrown into a pond of water, is completely contained in the wave structure.) Moreover, the "persistence of the trace" (or as Maass put it, the "fading memory" (Lukosevicius & Jaeger, 2009)) allows one to recognize at a temporal distance the signal that was sent to the liquid; and sequence and timing effects of inputs.

The Liquid State Machine is a recurrent neural network. In its usual format (Lukosevicius & Jaeger, 2009; Maass et al., 2002a), each neuron is a biologically inspired artificial neuron such as an "integrate and fire" (LIF) neuron or an "Izhikevich" style neuron (Izhikevich, 2003). 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. The 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 pattern.

In the context of LSM the "detectors" 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 a SVM or an Adaline (Widrow & Hoff, 1960), perceptron (Pitts & McCulloch, 1943), or three level back propagation neural networks, etc.

* Corresponding author. Tel.: +972 4 8288337; fax: +972 4 8288181.
E-mail addresses: hhazan01@cs.haifa.ac.il (H. Hazan), manevitz@cs.haifa.ac.il (L.M. Manevitz).

[1] SVM = support vector machine.

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.06.052

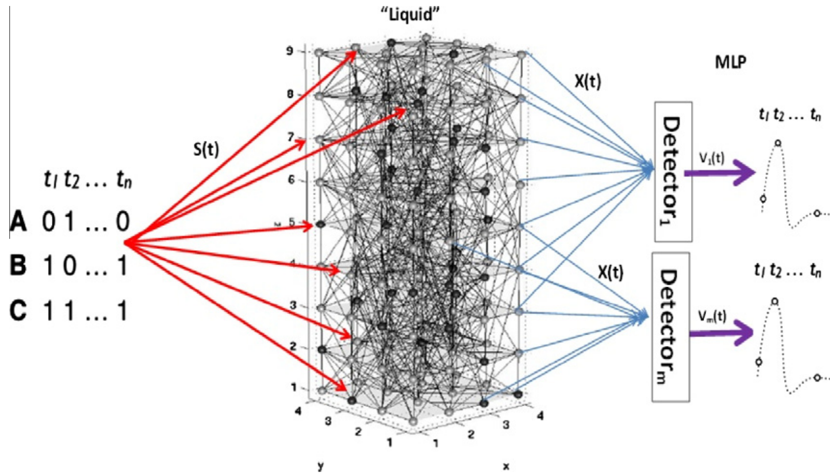

Fig. 1. Liquid State Machine framework.

The term detector is standard in the LSM community and date back to Maass et al. (Jaeger, 2001a; Lukosevicius & Jaeger, 2009; Maass, 2002; Maass & Markram, 2004; Maass et al., 2002b) the idea is that the "detectors" are testing whether the information for classification resides in the liquid; and thus are not required to be biological. In this way, it is theoretically possible for the detectors to recognize any spatio-temporal signal that has been fed into the liquid; and thus the system could be used for, e.g. speech recognition, or vision, etc.

This is an exciting idea and, e.g. Maass and his colleagues have published a series of papers on it. Amongst other things, they have recently shown that once a detector has been sufficiently trained at any time frame, it is resilient to noise in the input data and thus it can be used successfully for generalization (Bassett & Bullmore, 2006; Fern & Sojakka, n.d.; Maass et al., 2002b).

Furthermore, there is a claim that this abstraction is faithful to the potential capabilities of the natural neurons and thus is explanatory to some extent from the viewpoint of computational brain science. Note that one of the underlying assumptions is that the detector works without memory; that is the detector should be able to classify based on instantaneous static information; i.e. by sampling the liquid at a specific time. That this is theoretically possible is the result of looking at the dynamical system of the liquid and noting that it is sufficient to cause the divergence of the two classes in the space of activation.

Note that the detector systems (e.g. a back propagation neural network, a perceptron or a support vector machine (SVM)) are not required to have any biological plausibility; either in their design or in their training mechanism, since the model does not try to account for the way the information is used in nature. Despite this, since natural neurons exist in a biological and hence noisy environment, for these models to be successful in this domain, they must be robust to various kinds of noise. As mentioned above, Maass et al. (Lukosevicius & Jaeger, 2009; Maass, Legenstein, & Markram, 2002; Maass et al., 2002b; Maass & Markram, 2004) addressed one dimension of this problem by showing that the systems are in fact robust to noise in the input. Thus small random shifts in a temporal input pattern will not affect the LSM's ability to recognize the pattern. From a machine learning perspective, this means that the model is capable of generalization.

However, there is another component to robustness; that of the components of the system itself.

In this paper we report on experiments performed with various kinds of "damage" to the LSM and unfortunately have shown that the LSM with any of the above detectors is not resistant, in the sense that small damages to the LSM neurons reduce the trained classifiers dramatically, even to essentially random values (Hazan & Manevitz, 2010; Manevitz & Hazan, 2010).

Seeking to correct this problem, we experimented with different architectures of the liquid. The essential need of the LSM is that there should be sufficient recurrent connections so that on the one hand, the network maintains the information in a signal, while on the other hand it separates different signals. The models typically used are random connections; or those random with a bias towards "nearby" connections. Our experiments with these topologies show that the network is very sensitive to damage because the recurrent nature of the system causes substantial feedback.

Taking this as a clue, we tried networks with "hub" or "small world" (Albert & Barabási, 2000; Barabási, 2000; Barabási & Albert, 1999) architecture. This architecture has been claimed (Achard, Salvador, Whitcher, Suckling, & Bullmore, 2006; Bassett & Bullmore, 2006; Varshney, Chen, Paniagua, Hall, & Chklovskii, 2011) to be "biologically feasible".

The intuition was that the hub topology, on the one hand, integrates information from many locations and so is resilient to damage in some of them; and on the other hand, since such hubs follow a power rule distribution, they are rare enough that damage usually does not affect them directly. This intuition was in fact borne out by our experiments.

2. Materials and methods

We simulated the Liquid State Machine with 243 integrate and fire neurons (LIF) in the liquid following the exact set up of Maass and using the code available at the Maass laboratory software "A neural Circuit SIMulator".[2] To test variants of topology we re-implemented the code, available at our website.[3] The variants of the topologies implemented are described in the paper below as are the types of damages. Input to the liquid was at 30% of the neurons, the same input at all locations in a given time instances. The detectors of the basic networks were back propagation networks with three levels with 3 neurons in the hidden level and one output neuron. In most experiments, the input was given by the output of all non-input neurons of the liquid (i.e. 170 inputs to the detector). In some experiments (see section below) the inputs to the detector were given over 20 time instances and so the detector had 3400

[2] http://www.lsm.tugraz.at/csim/.
[3] http://www.cri.haifa.ac.il/neurocomputation.