Co-Evolutionary Learning of Liquid Architectures

Igal Raichelgauz
Department of Electrical Engineering
Technion, Haifa 32000, Israel
E-mail: igal.raichelgauz@intel.com

Karina Odinaev
Department of Bio-medical Engineering
Technion, Haifa 32000, Israel
E-mail: karinao@tx.technion.ac.il

Yehoshua Y. Zeevi
Department of Electrical Engineering
Technion, Haifa 32000, Israel
E-mail: zeevi@ee.technion.ac.il

***Abstract*— A large class of problems requires real-time processing of complex temporal inputs. These are difficult tasks for state-of-the-art techniques, since they require capturing complex structures and relationships in massive quantities of low precision, ambiguous noisy data. A recently-introduced Liquid-State-Machine (LSM) paradigm provides a computational framework for applying a model of cortical neural microcircuit as a core computational unit in classification and recognition tasks of real-time temporal data. We extend the computational power of this framework by closing the loop. This is accomplished by applying, in parallel to the supervised learning of the readouts, a biologically-realistic reward-based learning within the framework of the neural microcircuit (NM). This approach is inspired by neurobiological findings from ex-vivo multi-cellular electrical recordings and injection of dopamine to the neural culture. We show that by closing the loop we obtain a much more effective performance with the new Co-Evolutionary Liquid Architecture. We illustrate the added value of the closed-loop approach to liquid architectures by executing, as an example, a speech recognition task.**

I. Introduction

Of the various alternatives, large, random, vastly connected cortical networks are the best candidates for a core of biologically-motivated computational architectures. Moreover, even a relatively simple model composed of 100 leaky-integrate-and-fire neurons connected by dynamic synapses with stochastic heterogeneous parameters has an interesting computational power in a domain of parallel processing of temporal noisy data in real-time. A new computational paradigm, called Liquid-State-Machine (LSM), recently presented by [1], provides a theoretical basis for applying a model of neural microcircuit to generic computational tasks. The LSM system is composed of two parts: (1) Liquid - a model of neural microcircuit is used as a "reservoir" of complex dynamics to transform the input time series *u(.)* into "liquid states" *x(t)*. (2) Readout - memory-less function which maps the liquid state *x(t)* at time *t* onto the output *v(t)*. Readout may be implemented by a simple one-layer network of perceptron, trained by linear algorithm to build a function mapping liquid-states onto desired outputs. It was shown [2] by the means of simulations that such a system is computationally effective in executing parallel tasks of recognition and classification of temporal data. In the framework of computational LSM, a neural microcircuit is used as an efficient generic filter transforming different temporal inputs into significantly different liquid states. The task-dependent part is executed by the readout after being trained by supervised-learning algorithm to map these states onto predefined output. Turning back to neurobiological facts, the plasticity and learning ability of real cortical networks should not be neglected in the biologically-motivated computational framework. A feedback from the environment drives the learning process in neurobiological systems and allows the success in tasks varying in time rather than being predefined. In this study we extend LSM computational framework to a closed-loop setup wherein feedback from the environment drives the learning process of the computational core liquid unit - Neural Microcircuit (NM). However, it is not a straightforward task to define a learning algorithm to such a large and randomly constructed network, therefore an inspiration from neurobiological findings is required again. We use two neurobiological paradigms for implementing the learning of the NM - Reward based learning [3] and Dopamine induced learning by dispersion mechanism [4]. The learning process of the NM is composed of two stages - the exploration of various states of NM and the recognition of the appropriate one. In the proposed closed-loop framework, this biologically-motivated learning of the NM is done in parallel to the supervised learning of the readout, i.e. there is a co-evolutionary learning process of NM and readout until the best performance of the overall system is reached.

II. Neural Microcircuit as a Generic Computational Unit

The neocortex is characterized by precise structure of columns and layers. Within neocortical layers neurons are mapped into each other, where anatomical and physiological properties are unique for each type of pre- and post-synaptic combination. However remarkable morphological, electrophysiological and spatial stereotypy exists in these networks, in addition to very stereotypical connectivity and patterning of synaptic connections between neighboring cells. This clear stereotypy exists across different regions of the brain, suggesting that there is a generic template of microcircuit and that all neocortical microcircuits are merely subtle variations of that common microcircuit template. Such templates could subserve the apparent omnipotent functional capacity of the neocortical microcircuitry [5]. A computational model of generic neural microcircuit is inherently endowed with powerful and versatile information processing capabilities. We used a similar model to [2], composed of a 3-dimensional recurrent network of 135 Leaky-Integrate-and-Fire (LIF) neurons with random connectivity, and similarity to generic cortical microcircuit, 20% of the neurons are randomly chosen to be inhibitory and, accordingly, 80% excitatory. The probability of connection between two neurons depends on the distance between them according to,

$$C \cdot exp(D(i,j)/\lambda^2,$$

wherein λ and C are parameters that determine the average number of connections for a certain Euclidean distance D between the neuron *i* and neuron *j*. This connectivity characterization by primary local connections and a few longer connections is biologically realistic. Long range connections will be incorporated, and their functional effects on the computational properties of the network will be investigated within a context of a different study. Random, heterogeneous parameters of NM model fit neurobiological data from rat somatosensory cortex [2]. Synaptic short-term plasticity of the NM is implemented by dynamic synapses in which the amplitude of each post-synaptic-current depends on the spike train that is impinging on the synapse [6], and causes facilitation and depression processes. The model was implemented using CSIM simulator [7].

III. Learning by Dispersion

Learning process drives a neural microcircuit to a desired state defined by configuration of sets of associations between stimuli and responses. This dynamical process begins with exploration of various network's states through modification of neuronal correlations. Two mechanisms which may be responsible for changing neuronal correlations are driving stimuli and neuromodulation by dopamine. Experiments on ex-vivo culture have shown [8], [4] that both mechanisms enhance changes in neuronal correlations by dispersing existing correlations, i.e. decorrelating previously acquired correlated activity. It is assumed that both mechanisms that cause decorrelation (dispersion) are mediated by a biophysical jittering of the synaptic strengths at polysynaptic level. This has led to the idea of modeling both mechanisms by what Eytan and Marom [10] coined as ”Dispersing Mechanism”. The second phase of learning, the recognition, is responsible for ”freezing” the NM state by stopping the exploration process. In recent years, a major effort was devoted to mapping of the behavioral concept of reward to neural mechanisms that change the functionality of a given NM based on its past performance [9]. The regulation of exploration process, driven by dopamine neuromodulation, is enabled by reward prediction error (RPE) signals. Dopamine neurons appear to emit RPE signal, as they are activated by rewards that are better than predicted, uninfluenced by rewards that occur exactly as predicted and depressed by rewards that are worse than predicted [9]. Learning by reward can occur by associating a stimulus or an action with a reward [3]. The learning is a function of RPE, defined by Schultz as a scalar difference in value (magnitude x probability) between a delivered (DR) and a predicted reward (PR):

$$RPE = DR - PR = f(error\ in\ task)$$

We apply a constant delivered reward, i.e. p(DR)=1, as long as there is any success in task execution. The predicted reward is a function of the system's previous success in executing the task, i.e. PR=f(success in task execution). Since the performance of the system at the beginning of learning is lower than 100%; the predicted reward is lower than the delivered; dopamine neurons should be activated and emit dopamine to the system. We implement a feedback mechanism based on this reward mechanism in our Co-Evolutionary Learning of Liquid Architecture. According to this ”exploration and recognition” paradigm the dopamine jitters network's formed associations and thus enables state transition across the NM states space. In other words, the mechanism of jittering the synaptic efficacies, discovered by Eytan and Marom, is instrumental in avoiding trapping into a fixed point. When the best state dictated by the environment is found, the system reaches the recognition phase, and by stopping the dopamine emission, network's associations are ”frozen”. A mathematical model of this process, in which the synaptic efficacies are randomly jittered by regulation of RPE is formulated by:

$$\Delta W = \psi(W_0 \cdot K \cdot PRE)$$

wherein ψ is uniformly distributed between positive and the negative values of the argument, W_0 is the previous value of the synaptic strength, K is a constant, and ΔW is the change in the strength of the synapse. The model illustrates exploration and recognition processes, by dispersion of the NM synaptic strengths, regulated by the success in achieving the task of the overall system.

IV. Co-Evolutionary Learning in a Closed-Loop Framework

We propose a new closed-loop liquid architecture based on a NM as a core computational unit. The components of the system, illustrated in Fig.1, are NM, Readout function and a Decorrelator. In the open-loop setup the system is equivalent to recently-proposed general theoretical model, called Liquid-State-Machine [1]. LSM presented a convenient framework for neural computations in real time for rapidly-time-varying continuous input functions. NM stores information about past inputs with high dimensional dynamics in its internal perturbations. Different input streams to the microcircuit cause different internal sates (liquid states) of the system and enable the inputs to be separated. Liquid states of the NM are read by memoryless Readout. Readout is trained by supervised learning algorithms to transform high-dimensional transient liquid states of the NM onto desired outputs. After the Readout learns to define a needed class of equivalence, it can perform the learned task on novel inputs. The separation property (SP) requirement of the NM for functionality of LSM framework was illustrated in [2]. Within LSM framework the learning process is applied to the readout only, while the function of the NM as a generic filter is not changed. We propose an extended closed loop framework in which we apply to NM a previously-described learning-by-dispersion, driven by a feedback from the environment. The overall framework is described in Fig. 1. Time-varying stimuli from the environment excite NM with a continuous input stream $(P_i(t))$. At any time t_0, the internal liquid state of the microcircuit $(Q_i(t_0))$ holds a substantial amount of information about recent inputs $P_i(t < t_0)$. Memoryless readout neurons are trained to map liquid states $Q_i(t_0)$ onto discrete predefined values *j*. Discrete value *j* is a decision/action of the system in its environment. If the system succeeds in the task, i.e. *i=j* for classification task, reward signal is sent by the environment to the system. Reward signals, injected by the environment, are determined by system's performance and activate the Decorrelator by setting the value of RPE. Decorrelation mechanism modifies the NM synaptic strengths according to previously defined algorithm and drives the exploration phase of learning.

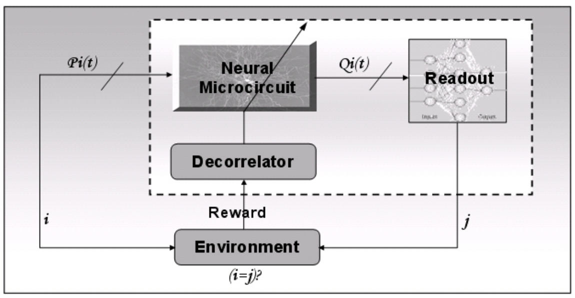

Fig. 1. Closed-loop liquid architecture implemented in a classification task of time-varying inputs. NM is composed of 135 LIF neurons. Time-varying stimuli $P_i(t)$ are transformed by NM onto liquid states, $Q_i(t)$, defined as firing patterns of NM at time t_0. Readout neurons are trained by supervised learning to identify the input applied to the system by transforming NM liquid states onto discrete value *j*. A feedback on system's performance is sent by the environment in form of reward signals to determine the RPE. Decorrelation, regulated by RPE, enables the co-evolution of the Readout and NM until a desired performance is obtained

During the exploration of NM states, the Readout is trained by supervised learning to transform the new formed liquid states onto system's output. When system's performance is sufficient, RPE is low, the recognition phase is reached and NM state is "frozen" by stopping the dispersion of the synaptic strengths. We applied this co-evolutionary learning of the liquid architecture in general computational task of classification time varying stimuli. Randomly generated Poisson spike trains were injected to the system with a certain noise. Analysis of system's performance in a closed-loop versus an open-loop setup will be described in the next section.

V. Computational Analysis of the Closed-Loop Framework

The added value of a closed-loop setup is examined in a general computational task of classification of a Poisson spike train. The error-in-task of the open-loop setup remains almost constant, since the optimal performance of the system is reached after the first supervised learning of the Readout is completed. In a closed-loop setup, in parallel to the supervised learning of the Readout, we apply a learning-by-dispersion of the NM. This co-evolutionary learning, of NM and Readout, generates an exploration process until the optimal performance of an overall system is obtained. The learning curve of a closed-loop versus an open-loop setup is illustrated in Fig. 2.

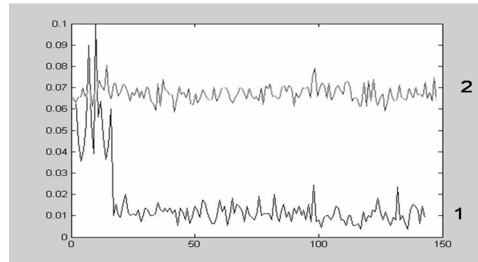

Fig. 2. Learning curve of a closed-loop *(1)* versus open-loop *(2)* setup implemented in classification task of time-varying stimuli

As the computational results depicted in Fig.2 indicate, the curve of co-evolutionary learning in a closed-loop setup does not converge gradually to the optimal point, since there is no *a-priori* knowledge of such a point. Various states of the NM are explored. This type of exploration is manifested by "jumps" characteristic of the learning curve. The exploration continues until a sufficient performance is obtained, at which time the NM state is "frozen". The experimental results indicate that an optimal learning curve is achieved at a certain reward strategy, as indeed predicted by equation (3).
It has been proposed that extensive computational capabilities are achieved by systems whose dynamics is neither chaotic nor ordered, but somewhere in between order and chaos. Computation on the edge of chaos of randomly connected recurrent neural networks in the domain of real-time processing was analyzed in [14]. Two types of dynamics, ordered and chaotic, in the proposed NM model, are illustrated in Fig. 3. Ordered dynamics is characterized by strong correlation of NM response at given time, to the input presented to the NM at the same time. There is no fading-memory in ordered systems, thus separation of temporal inputs is not efficient. Chaotic systems are driven by their internal dynamics with low correlation to the inputs, thus the extraction of the information from NM liquid-states about the applied inputs is not efficient as well.

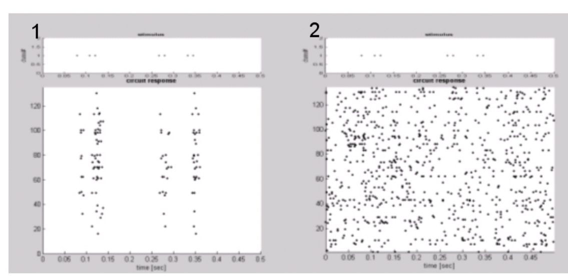

Fig. 3. NM ordered *(1)* and chaotic *(2)* response to the stimulus. Identical stimulus (depicted above) applied to the NM in both, ordered and chaotic, dynamical states.

There is an optimal dynamical behavior of the NM from computational viewpoint, wherein the NM holds substantial information about past stimuli. This dynamics is characterized by rich excitability relative to ordered state, and strong correlation to the stimuli relative to chaotic state. Co-evolutionary learning changes NM dynamical behavior by modifying its synaptic strengths during the exploration process. Fig. 4 depicts NM response to the stimuli at the beginning of exploration process versus its response after the optimal NM state was obtained. The change in response illustrates a transition of NM state from ordered to more chaotic. In fact, this co-evolutionary learning drives NM dynamics towards the edge of chaos. The rise of NM excitability, resulted from transition to more chaotic state, plays an important role in system's robustness to noise. Excitability rise of NM is manifested by rich responses to stimuli, in terms of firing rate and number of firing neurons, and thus provides the Readout more information about past stimuli. This robustness of liquid states to noise results in robustness to noise of the overall system.

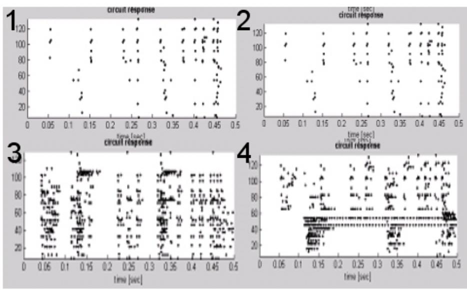

Fig. 4. Pre-learning *(1,2)* versus post-learning *(3,4)* NM response to stimuli illustrates a rise in NM excitability.

The closed-loop architecture exhibits superior performance, compared with the open-loop, insofar as the signal-to-noise ratio (SNR) is concerned. As the data depicted in Fig. 5 illustrates, the SNR of the closed-loop setup is by far lower than that characteristic of the open-loop setup.

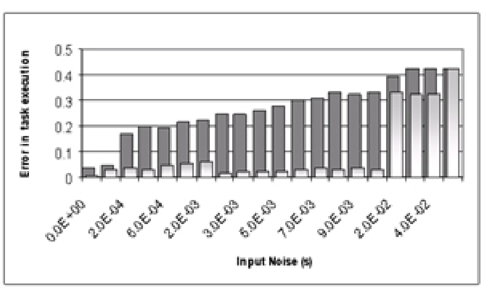

Fig. 5. Error-in-task for a closed-loop *(bright bars)* and open-loop *(dark bars)* setups versus noise in input.

Whereas the error increases with noise level in the open-loop, as expected, in the closed-loop it even decreases, until at a certain noise level this advantage of the closed-loop breaks down. This abrupt shift in performance is due to a network's phase transition to a chaotic state, which results from saturation by the reward. This phenomenon should be checked and confirmed in ex-vivo experiments on tissue culture such as those reported in [4].

VI. Voice recognition task

Co-evolutionary learning of liquid architecture was applied in a well-studied computational benchmark task for which data had been made publicly available - a speech recognition task [11]. The dataset consists of 500 input files: the words ”zero”, ”one”, ”two”, . . . , ”nine” are spoken by 5 different speakers, 10 times by each speaker. The task was to construct a network of I&F neurons that could recognize each of the spoken words. The waveforms of the input sound were preprocessed by performing Fourier transform. Each of the frequency bands was composed of one or more of the following three events: onset (the start of the phase of significant energy), offset (the end of this phase), and peak (the first maximum of energy). The entire waveform is normalized to have maximum amplitude of 0.7, the sampling rate used in this case is 12000 samples/sec. The running average power and its second derivative are subsequently used in identification of events in the sound's spectrogram. This sound preprocessing converts the sound signal into a spatiotemporal sequence of events, suitable for recognition. Monosyllabic words are encoded into such sequences by retrieving features in different frequency bands in their spectrogram. Finally, sound waveform is converted into a list of 40 single events that are converted in turn into their respective times of occurrence [13]. Internet competition was publicized on this dataset for finding a network with the best classification performance. The best performance in this competition exhibited an error of 0.15, and was accomplished by a network with 800 pools of neurons [12]. The same task was solved by Maass, Natchlaeger and Markram in 2002 [2] using LSM framework with 145 I&F neurons. The average error in this classification task, achieved by this network, was 0.14. We tested the co-evolutionary learning of the liquid architecture on the same task and the same dataset. A randomly chosen subset of 300 input files was used for training and the other 200 for testing. A previously described, randomly generated NM was implemented in a co-evolutionary learning of a closed-loop setup. The average error in this classification task, achieved by this closed-loop system, was 0.06.

VII. Discussion

Liquid architectures embed interesting computational learning features in NM model. These emerging architectures are motivated by neurobiological findings obtained in experiments with neural culture [4], [8], [10]. The common component of these liquid architectures is a core computational unit implemented by a generic heterogeneous model of NM. The proposed feedback mechanism adds a significant computational power to liquid architectures, illustrated for example in our simulations comparing the performance of the open and a closed-loop as a function of a noise level. Liquid architecture exhibit a broad spectrum of solutions obtained under the condition of an identical task, manifested by its internal parameters. Co-evolutionary learning, illustrated in this study, provides a robust mechanism that exploits this computational feature, by randomly exploring the states space. A feedback mechanism regulates the exploration process until a sufficient solution is obtained. Converging the ideas of liquid architecture, feedback mechanism and learning by exploration reveals a powerful paradigm for real-time, parallel computation in a rapidly varying environment.

REFERENCES

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

[2] W. Maass, T. Natschlaeger, and H. Markram, "Computational models for generic cortical microcircuits," *In J. Feng, editor, Computational Neuroscience: A Comprehensive Approach,* chapter 18, pp. 575-605, Chapman & Hall/CRC, Boca Raton, 2004.

[3] W. Schultz, "Neural coding of basic reward terms of animal learning theory, game theory, microeconomics and behavioural ecology," *Curr Opin Neurobiol*, vol. 14(2), pp. 139-47, Apr 2002.

[4] D. Eytan, A. Minerbi, N. Ziv and S. Marom, "Dopamine-induced Dispersion of Correlations Between Action Potentials in Networks of Cortical Neurons," *J. Neurophysiol.*, vol. 92, pp. 1817-1824, 2004.

[5] Silberberg, A. Gupta and H. Markram, "Stereotypy in neocortical microcircuits," *Trends Neurosci.*, vol. 25(5), pp. 227-30, May 2002.

[6] M Tsodyks, K. Pawelzik, H. Markram, "Neural networks with dynamic synapses," *Neural Computation*, vol. 10, pp. 821-835, 1998.

[7] T. Natschlaeger, H. Markram, and W. Maass, "Computer models and analysis tools for neural microcircuits," *In R. Kotter, editor, A Practical Guide to Neuroscience Databases and Associated Tools,*, chapter 9, Kluver Academic Publishers (Boston), 2002 in press. Website: http://www.lsm.tugraz.at

[8] G. Shahaf and S. Marom, "Learning in networks of cortical neurons," *J. of Neuroscience*, vol. 21(22), pp. 8782-8788, Nov. 15, 2001

[9] W. Schultz "Predictive reward signal of dopamine neurons," *J. of Neuroscience*, vol. 80, pp. 1-27, 1998

[10] D. Eytan and S. Marom, "Learning in Ex-Vivo Developing Networks of Cortical Neurons," *Progress in Brain Research*, vol. 147, "Development, Dynamics and Pathology of neural Networks", van Pelt et al, Editors, 2004

[11] J. Hopfield and C. Brody, The mus silicium (sonoran desert sand mouse) web page. Base: http://moment.princeton.edu/m̃us/Organism

[12] J. Hopfield and C. Brody, "What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration," *Proc. Natl. Acad. Sci. USA*, vol. 98(3), pp. 1282-1287, 2001

[13] Prashant Joshi, "Synthesis of a liquid machine with Hopefield/Brody transient synchrony," M.SC Thesis, (2002)

[14] N. Bertschinger and T. Natschlaeger, "Real-time computa-tion at the edge of chaos in recurrent neural networks," *Neural Computation,* , vol. 16(7), pp. 1413-1436, 2004