DYNAMICS OF NEURONAL PLASTICITY: SIMULATION OF A COMPENSATION ALGORITHM FOR SMALL-SCALE NETWORKS

DANIEL RIORDAN WUELLNER

Abstract. We explore a single morphogenetic algorithm to model neural plasticity in small-scale networks. Using randomly-generated McCulloch-Pitts networks and the simulation outlined by Dammasch et al (1986), we attempt to simulate the effects of a compensation algorithm on long-term network behavior and morphogenetic changes. The results obtained indicate further development is required to properly analyze these effects.

1. INTRODUCTION

One of the most fascinating aspects of biological nervous systems is the capacity to simultaneously learn new information while remembering existing information over time. This capacity entails two opposed processes: the ability for a nervous system to change in response to novel stimuli and the capacity to maintain existing structures. In other words, the nervous system maintains a level of 'neuronal homeostasis' as it acquires knowledge. [2]

This simulation attempts to model this process by using a simple compensatory rule for modulating the connection strength between neurons in a small network based on past activity. The anticipated goal of this project was to develop a mechanism for stabilizing the connectivity of randomly-generated networks which could then be coupled to other morphogenetic algorithms which encode the associative learning mastered by biological nervous systems. While the results indicate we have not yet met that goal, the simulation developed should a provide a flexible framework for future study.

2. BACKGROUND

An understanding of the biology of impulse transmission across synapses as well as the specific mechanisms of synaptogenesis are not critical to this project; a basic neurobiology text was used for reference [7]. The critical features of the neural system are summarized here:

(1) A neural network is composed of individual neurons which interact with each other electrically across synapses. Information flows in one direction on each neuron; that is, it has specific 'postsynaptic' elements to receive inputs (dendrites) from other cells and specific 'presynaptic' elements which send outputs (axonal elements) to other cells.
(2) Each neuron encodes a certain spatial and temporal integration process which maps a large set of inputs into a single binary output. In particular, the inputs a neuron receives changes its membrane potential and that neuron fires an 'action potential,' a single output event if and only if the inputs exceed some threshold potential.
(3) Some neurons cause the membrane potential of their output targets to increase closer to threshold ('excitatory' neurons) and others cause the membrane potential to decrease away from threshold ('inhibitory' neurons).
(4) Following an action potential, a neuron is relatively insensitive to inputs during a short time period, the 'refractory period.'
(5) The time scale for synaptic transmission, integration, and firing is on the order of msec, while the gradual changes in connectivity occur on the order of hours or days, thus these processes may be considered independently for the purpose of simulation.
(6) The morphogenetic changes a neuron experiences, including synaptic growth and degradation as well as cell proliferation and death, are

correlated with its electrical activity. Other factors including hormones and other chemicals also play a role in morphogenesis, but these will not be considered here.

3. DYNAMICAL SYSTEM

We use the classical McCulloch-Pitts network model to simulate small neural networks and their subsequent evolution over time according to a morphogenetic rule (limited in the scope of this project to a simple compensation type rule) [5]. In this formulation there are a number of key parameters and relevant state variables:

3.1. State Variables.

z^t: Activity vector indicating which of the N neurons are active at time t.

$\boldsymbol{C}$: Connectivity matrix (N by N) where c_{ij} encodes the strength of the input from neuron j on neuron i.

psp^t: Vector of the current postsynaptic potential (PSP) for the N neurons

3.2. Parameters.

The following summarizes the key parameter space with default values tuned to within the range where most randomly generated connectivity matrices and random initial activities reach a self-sustaining limit cycle.

Parameter	Definition	Default Value
N	Number of Neurons	30
NE	Number of Excitatory Neurons	27
$NI = N - NE$	Number of Inhibitory Neurons	3
θ	Threshold	1.0
ϕ	Weight of Inhibitory Connections	2.0
K	Percent Initial Network Connectivity	60%
Ξ	Distribution of Initial Connection Strength	$N(.5, .1)$
I	Percent Neurons Active at Time $t = 0$	10%

To clarify, at time $t = 0$, the number of nonzero entries in each row of $\boldsymbol{C}$ equals $(N \cdot K)$, and the value c_{ij} is generated using the distribution Ξ. Also, the initial activity vector z^0 is generated by uniformly distributing $(N \cdot I)$ 1 entries across an otherwise 0-valued N-tuple.

Additionally, we impose several physiologically-based constraints on the system: first, after a neuron fires (i.e. $z_i^t = 1$), it undergoes a transient 'refractory' period when it is insensitive to input as a consequence of hyperpolarization of the neural membrane following depolarization. Throughout this simulation, we have assumed that the length of this refractory period is 1 time step, so $z_i^{t+1} = 0$ necessarily, but z_i^{t+2} may be 1 or 0. Consequently, neurons cannot self-stimulate and we do not allow $c_{ii} > 0$.

3.3. **Equations of State.** At each time step, the current state of activity of the network is given by the N-tuple z_t where for each neuron $k, 1 \leq k \leq N$, the k-th entry z_k^t is calculated by:

$$psp_k^t = \sum_{j=1}^{NE} c_{kj} \cdot z_j^{t-1} - \sum_{j=NE+1}^{N} \phi \cdot c_{kj} \cdot z_j^{t-1}$$

$$z_k^t = \begin{cases} 1 & \text{if } psp_k^t \geq \theta \text{ and } k\text{-th neuron is not refracting} \\ 0 & \text{otherwise} \end{cases}$$

In other words, if the weighted sum of synaptic inputs to any (nonrefracting) neuron at time $t-1$ exceeds the threshold θ, assumed equal for all neurons in the network, that neuron is activated at time t, i.e. $z_k^t = 1$.

3.4. **Equations of 'Motion': Compensatory Morphogenesis Rule.** After an initial transient period (typically 10 steps is sufficient since most randomly-connected networks stabilize to a limit cycle within 5 iterations), we begin to impose a morphogenesis rule which changes the connectivity matrix $\boldsymbol{C}$ based on prior activity (the algorithm is described below). Physiologically, synaptic transmission and neural integration occur on a time scale of msec, whereas synaptogenesis and cell proliferation occur on a time scale of hours or days [3]. Despite this fact, we used a slow time scale equivalent to 10 times the fast time scale for computing efficiency since a longer slow time scale does not seem to effect the dynamics (the network restabilizes within 10 'fast' time steps).

The specific compensation rule used in the simulation is a variation on that outlined in Dammasch et. al. 1986 and was originally intended only as a first step to generating 'physiologically stable' networks from initially random connected matrices; see below for further discussion of the somewhat surprising results that curtailed this plan.

This compensation algorithm is based on the assumption that a highly plastic neuron has some ideal window of average postsynaptic potentials near its threshold value; otherwise, excessively stimulated neurons (where a PSP which generated an action potential is significantly higher than threshold) or understimulated neurons (which rarely receive sufficient inputs to fire) are not sensitive to small changes in input strengths and hence remain unresponsive to small perturbations in the network. On the other hand, this assumption is not expected to hold for mature, stable neural structures which have already been tuned to respond to certain activities.

The compensation algorithm is as follows: first, the current PSP for each neuron is compared to upper and lower bounds for deviations from the threshold θ:

If $psp_k^t - \theta > \theta \cdot \sigma/2$	Then compensate down
If $\theta - psp_k^t < \theta \cdot \sigma/2$	Then compensate up
Otherwise	No morphogenesis

Here the deviation from the threshold is encoded as σ and set at 0.3 throughout this simulation. If the PSP of a neuron meets these criteria, the connectivity matrix entries for the row corresponding to that neuron are modulated in proportion to that entry's fraction of the row sum. If the k-th neuron needs to be compensated down, we decrease all of the nonzero incoming excitatory entries c_{kj} by the following algorithm:

$$\delta = k^H_{epo} \cdot \frac{c_{kj}(t)^2}{\sum_{j=1}^{NE} \cdot c_{kj}(t)}$$

$$c_{kj}(t+1) = \begin{cases} c_{kj}(t) - \delta & \text{if } c_{kj}(t) - \delta > 0 \\ 0 & \text{otherwise} \end{cases}$$

If the k-th neuron needs to be compensated up, we decrease all of the nonzero incoming inhibitory entries c_{kj} by the following algorithm:

$$\delta = k^L_{ipo} \cdot \frac{c_{kj}(t)^2}{\sum_{j=NI}^{N} \cdot c_{kj}(t)}$$

$$c_{kj}(t+1) = \begin{cases} c_{kj}(t) - \delta & \text{if } c_{kj}(t) - \delta > 0 \\ 0 & \text{otherwise} \end{cases}$$

Additionally, we decrease all of the nonzero output entries c_{kj} (in this case we adjust the entries on the $k-th$ column in proportion to the column sum):

$$\delta = k^L_{pr} \cdot \frac{c_{kj}(t)^2}{\sum_{j=1}^{N} \cdot c_{jk}(t)}$$

$$c_{jk}(t+1) = \begin{cases} c_{jk}(t) - \delta & \text{if } c_{jk}(t) - \delta > 0 \\ 0 & \text{otherwise} \end{cases}$$

The constants $k^H_{epo}, k^L_{ipo}, k^L_{pr}$ encode the degree of morphogenesis at each step; throughout this project we set all of these to 0.1.

3.5. **State Space.** The state space of the system is the set of connectivity matrices

$$\{[c_{ij}] : 1 \leq i \leq N, 1 \leq j \leq N, c_{ij} \in [0,1]\}$$

Individual trajectories $\boldsymbol{C}(t)$ correspond to morphogenetic evolutions of the network's connectivity.

4. METHODS

Following the simulation outlined in Dammasch et al [3], we created a network simulator in Python. The simulator outputs data log files containing the network's connectivity matrix at the beginning and end of the simulation as well as individual time logs of the network activity vector at each time step of the simulation. Several tools were written to analyze these data statistically as well as graphically; some of these results are included below.

For each simulation, a randomly connected N x N matrix is generated. Static and plastic modes are run over this same network using a set of randomly generated initial activities (the simulation is run in static and plastic mode over the same set of initial conditions). During the plastic mode, the connectivities of the network are updated synchronously.

5. RESULTS

First, the simulation of this compensation algorithm led to some unanticipated results that require further analysis. The results obtained up to this point are summarized below:

(1) Regardless of the parameter settings, the activity vectors of the static simulations quickly settle into the single fixed point $v = (0, 0, \dots, 0,$ corresponding to no activity or a limit cycle where activity oscillates among specific subpopulations of neurons. These limit cycles seem to be exclusively period 2, indicating further tuning of the parameters could be illustrative since nothing in the algorithm necessitates this. Fortunately, these results reproduce those from Dammasch et al 1986 [3].

(2) Several parameters have a large impact on the proportion of matrices and initial conditions which reach a limit cycle rather than the zero vector fixed point, namely the connectivity strength Ξ and the weight of inhibitory connections ϕ. In either case, increasing the parameter tends to result in more 'inactive' networks.

(3) The compensation algorithm yielded surprising results. From plots of average activity of the networks over time (see figs 1-4 below in appendix), the compensation algorithm actually tended to destabilize the activity of the network in an unpredictable fashion. Periods of relative stability are interrupted by brief periods of aperiodic activity and eventually the activity of all the tested networks ended at the zero fixed point.

6. CONCLUSION

While these results show that further work is needed to develop the compensation algorithm to effectively stabilize randomly-generated McCulloch-Pitts networks, we have developed a framework to test the effects of various morphogenetic algorithms on small-scale neural networks. Next steps on this project include the following: reevaluation of the parameter space, in particular nondimensionalization, in order to better model the physiological range, and testing of other morphogenetic rules, such as Hebbian learning algorithms.

References

[1] Butz M, Lehmann K, Dammasch IE, Teuchert-Noodt G (2006) A theoretical network model to analyse neurogenesis and synaptogenesis in the dentate gyrus. *Neural Networks*, 19, 1490-1505.

[2] Lehmann K, Butz M, Teuchert-Noodt G (2005) Offer and demand: proliferation in the dentate gyrus. *European Journal of Neuroscience*, 21, 3205-3216.

[3] Dammasch IE, Wagner GP, Wolff JR (1986) Self-stabilization of neuronal networks: The compensation algorithm for synaptogenesis. *Biol. Cybern.*, 54, 211-222.

[4] Dammasch IE, Wagner GP, Wolff JR (1988) Self-ntabilization of neuronal networks: Stability sonditions for synaptogenesis. *Biol. Cybern.*, 54, 211-222.

[5] Dammasch IE, Wagner GP (1984) On the properties of randomly connected McCulloch-Pitts Networks: Differences between input-constant and input-variant networks. *Cybernetics and Systems*, 15, 91-117.

[6] Cromme LJ, Dammasch IE (1989) Compensation type algorithms for neural nets: stability and convergence. *J. Math. Biol.*, 27, 327-340.

[7] Purves D, et al, eds. (2004) *Neuroscience*, Sunderland, MA: Sinauer Assoc., inc.

7. APPENDIX: Figures

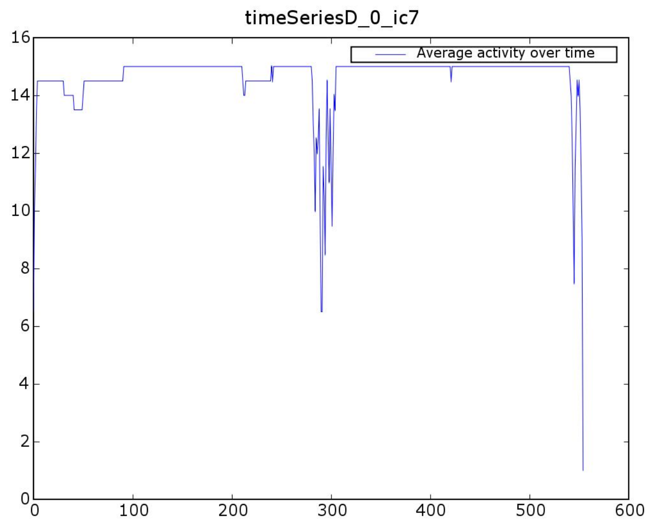

Figure 1. Example of stable behavior interrupted by transient deviations

Graduate Group in Applied Mathematics, University of California - Davis
E-mail address: `dwuellner@math.ucdavis.edu`

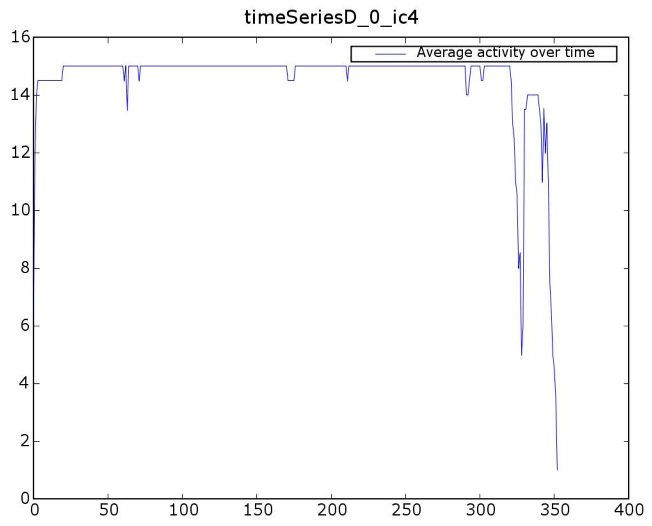

FIGURE 2. Example of stable behavior interrupted by transient deviations

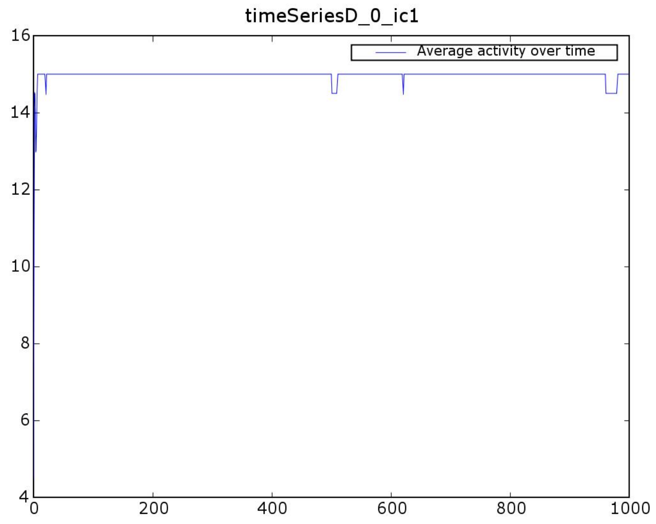

FIGURE 3. Example of maintained stable behavior with small deviations

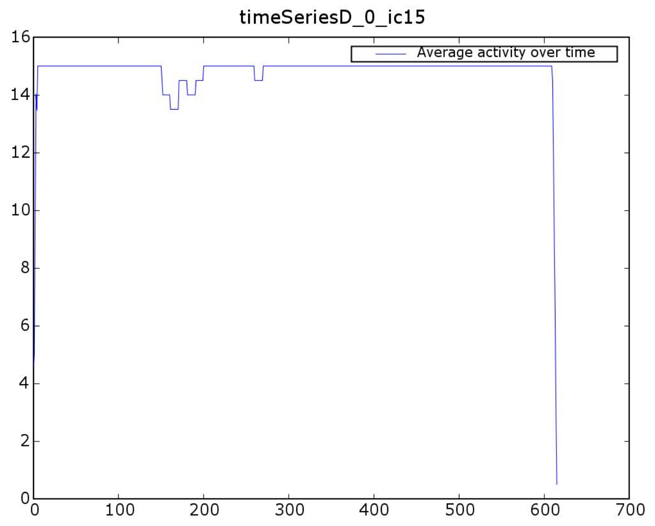

FIGURE 4. Example of stable behavior with sudden termination