Journal of Statistical Physics, Vol. 90, Nos. 1/2, 1998

Significance of Updating Schemes in Computational Models: Dynamics of Neutral Networks

Larry S. Liebovitch[1,3] and Michal Zochowski[1,2]

Received May 30, 1997; final August 21, 1997

A Hopfield neural network was constructed with relevance to protein dynamics. The dynamics of this network was analyzed by determining the distribution of first passage times between memories and its dependence on temperature. The distribution depended on the updating scheme. This illustrates the importance of choosing an updating scheme that leads to physically meaningful results in computational models of dynamic processes, such as in neural networks or molecular dynamics.

KEY WORDS: Monte Carlo updating schemes; neural networks; protein dynamics.

There are numerous examples in physics of complex systems, such as neural networks,[1] biomolecules[2] and spin glasses,[3] that have many degrees of freedom with many interactions that impose conflicting constraints. Computer simulations are an important tool in understanding the physics of these systems. They are used to determine the stable conformations that these systems reach in long times as well as the dynamics of the approach to these states. The computational form of the simulation must correctly model the physics if it is to lead to correct results. This is especially true for understanding the dynamical properties of these systems.

[1] Center for Complex Systems, Florida Atlantic University, Boca Raton, Florida 33431; e-mail: liebovitch@walt.ccs.fau.edu, zochowski@walt.ccs.fau.edu.
[2] Center of Theoretical Physics, Polish Academy of Science, 02-784, Warsaw, Poland.
[3] To whom correspondence should be addressed.

0022-4715/98/0100-0253$15.00/0

We used a simple, parallel distributed, neural network to study the properties of such computational models. We have been interested in neural networks as a way of modeling how ion channel proteins change their conformational shape from ones that are open to ones that are closed to the passage of ions, such as sodium and potassium, through the cell membrane. This problem involves understanding the dynamics of the protein switching from one formed conformational shape to another rather than the protein folding problem which studies how an unformed structure folds into its stable shape. Proteins have many of the characteristics of neural networks, such as parallel distributed structure, frustration, and ultrametricity.[4–7] It may therefore be possible to use neural networks to compute the dynamics of these proteins.[5] The spatial shape of the protein can be encoded in the values of the nodes of the network and the forces or energies in the connection strengths. As the neural network is updated the changing values of the nodes may be able to compute the changing shape of the corresponding protein. We ultimately want to assess if the dynamics computed by such a network matches that of its corresponding protein. However, we must first understand how the dynamics computed by the network, or any computational simulation, depends on how the simulation is carried out, which is the goal of the studies presented here.

There has been much study of static properties of neural networks, such as the nature of the energy surface and whether a given set of initial conditions converges to a desired configuration in the long term. However, there has been little study about the dynamics of how such networks reaches those memories. In particular, does the dynamics of the network depend on the scheme used to update the computation?

In this article we report the results of our studies of a Hopfield type neural network.[8] To assess the dynamical properties we determined the distribution of first passage times from one state to another. We found that the updating scheme can have a crucial affect on these distributions. These results have important implications for the dynamics computed by neural networks and by other simulations as well, such as molecular dynamics. It raises the question as to whether some dynamical properties computed by such simulations are artifacts of the updating scheme used.

The PDF distribution for one stored memory was derived in ref. 5. However, we could not solve for PDF analytically for two or more stored memories. Therefore, our results presented here are based on numerical simulations.

The network we studied had $N = 100$ nodes which were all connected to each other. The values of the nodes $\xi_i = \pm 1$. There were two memories ($\mu = 1, 2$) analogous to the open and closed states of the ion channel protein that consisted of the set of N nodes ξ_i^μ which were given random

values so that the two memories are orthogonal. The connection matrix **J** was constructed by using the Hebbian algorithm:

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{2} \xi_i^\mu \xi_j^\mu$$

The state of the network corresponding to the open or closed state of the ion channel was defined by the largest overlap:

$$m^\mu(t) = \frac{1}{N} \sum_{i=1}^{N} \xi_i^\mu S_i(t)$$

where $S_i(t)$ are the values of the nodes. When the state with the largest overlap changed from $1 \to 2$ or from $2 \to 1$, the network passed through the boundary between the two states. The first passage times are the number of consecutive time steps spent in each state. We used the distribution of first passage times to characterize the dynamics of the network since this corresponds to the distribution of the open and closed times of the ion channel protein that are measured by patch clamp experiments.[4] We determined these distributions as a function temperature. The temperature was parametrized by $\beta = 1/kT$, where T is temperature in °K, and k is the Boltzmann constant.

There are two major ways that the values of the nodes of the network can be updated:

1. *Synchronous updating*: all the elements are updated in every time step.

2. *Asynchronous updating*: one randomly chosen element is updated at every time step.

In either method, at each step, the probability that a given neuron assumes a new state is given by:

$$Pr(S_i = \pm 1) = \frac{1}{1 + \exp(-2\beta h_i)}$$

where:

$$h_i = \sum_{j=1}^{N} J_{ij} S_j$$

These different updating schemes resulted in different first passage time distributions as shown in Figs. 1, 2. For a synchronous updating, at high

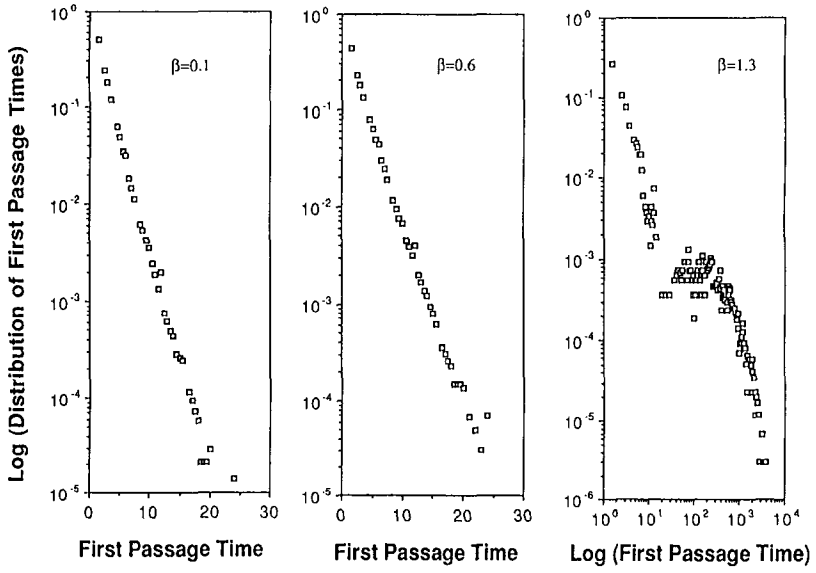

Fig. 1. Synchronous updating: Distribution of first passage times between the two memories in the neural network. High temperature ($\beta = 0.1$), medium temperature ($\beta = 0.6$), and low temperature ($\beta = 1.3$). As the temperature decreases the distribution changes from a single exponential to a power law with an exponential tail. This is consistent with the dynamics expected of a neural network.

temperature, the distribution is a single exponential. At lower temperature, the distribution changes to a power law with a single exponential tail. For synchronous updating, at both high and low temperature, the distribution is always a power law.

These results can be understood in the following way. The evolution of the network in time can be represented as the motion of a point on an N-dimensional hypercube whose vertices are given by the values $S_i(t)$ of nodes. The motion of this point on N-dimensional hypercube has a random component determined by the temperature and may have an additional component at each location driven by the connection strengths and the values of the nodes. The vertices of that hypercube are split into two distinct regions which are representing open and closed conformational states of the protein. Thus not all vertices of the hypercube lie on the boundary between those two regions.

The FPT distribution for such a random walk in a finite region of an M-dimensional space (in our case $M = N/2$ for both regions) is a power law at short times and a single exponential at longer times.[9, 10] The transition

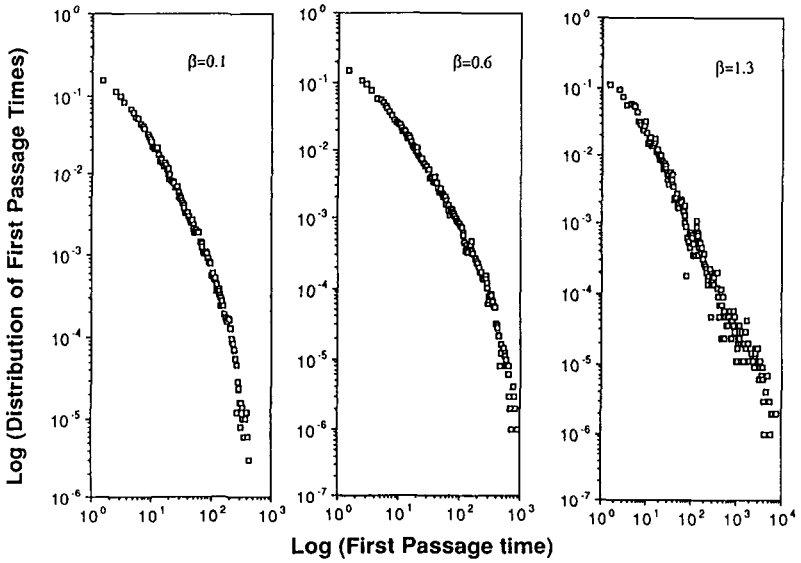

Fig. 2. Asynchronous updating: Distribution of first passage times between the two memories in the neural network computed by asynchronous updating. High temperature ($\beta = 0.1$), medium temperature ($\beta = 0.6$), and low temperature ($\beta = 1.3$). The distribution is a power law at all temperatures. This form may represent an artifact of the asynchronous updating.

between these two regimes occurs at the time expected for the random walker to reach the boundary of the region.

For synchronous updating, at high temperature, the heat energy is much above the energy surface of the network. The configuration of the network performs an unrestricted random walk at high enough energy to reach the edge of the configurational space in short enough time to make the distribution of first passage times a single exponential. At lower temperature, the random walk is slower, it takes longer to reach the edge of the configurational space, and thus the distribution of first passage times consists of a power law with an exponential tail.

The behavior of the synchronous updating can also be thought of in another way. At high temperature there is enough energy so that the network freely wanders over the energy surface producing a single exponential distribution. As the temperature is lowered there is a spin-glass phase where the network encounters many local minima and wanders between them. The heat energy is of the same order as the energy of the network. This produces constraints on the walk among these minima which leads to the power law component in the distribution of first passage times.

As the temperature is lowered further the network reaches one stable configuration, corresponding to one memory, and there is not enough heat for it to switch out of that memory. These results are consistent with the derived physical properties of the T-α (temperature-storage capacity) phase diagram [1, p. 305]. As the temperature is lowered (in the limit of few memories) the network passes from an ergodic phase where it passes through the entire configurational space, to a spin-glass phase where it passes through many local minima, to a stable state where the network converges to the minimum of a true memory. We also found that the rate constant of the single exponential at high temperature had the expected physical characteristic of an Arrhenius ($e^{-\text{constant}/kT}$) temperature dependence. It is worth mentioning that the spin-glass phase is a well defined finite region in these simulations because the network is of finite size ($N = 100$). For a fixed number of memories, the size of this spin-glass region scales with the number of nodes. For a network with two stored memories, in the limit as $N \to \infty$ there is no spin-glass phase because $\alpha \to 0$.

For asynchronous updating, at both high and low temperature, the distribution of first passage times is the same power law. Since only one node is updated at a time, the network can only pass through a limited number of points adjacent in the configurational space. Thus, the random

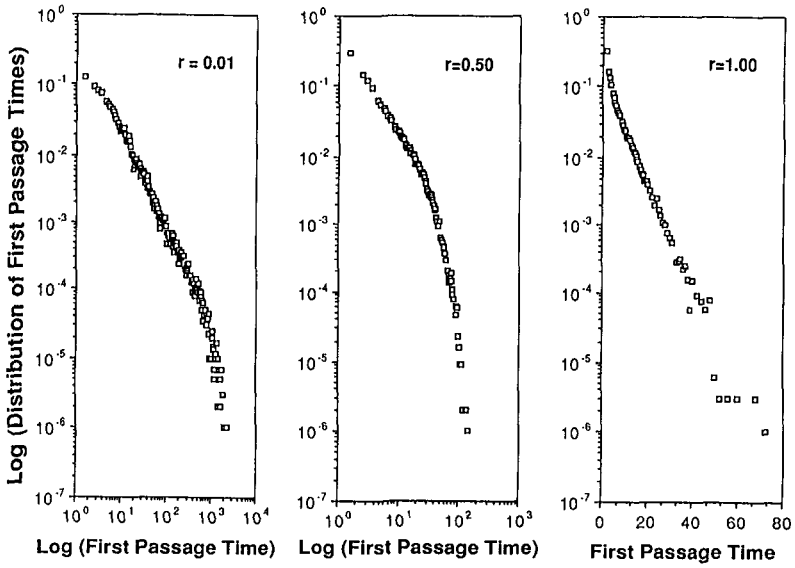

Fig. 3. As the updating was varied from asynchronous ($r = 0.01$) to synchronous ($r = 1$) the distribution of first passage times varied from a power law, to a power law with an exponential tail, to a single exponential. These simulations were performed for $\beta = 1$.

Table I. Summary of the Obtained Results: Changes of the FPT Distributions Dependent on the Updating Scheme

	High Temperature Limit	Low Temperature Limit
Synchronous Updating	single exponential	power law with single exponential tail
Asynchronous Updating	power law	power law

walk is constrained, it takes much longer to reach the edge of the configurational space, and thus the distribution of first passage times is dominated by the power law component. The restrictions introduced in the motion in the configurational space cause correlations in the dynamics that are due to the updating scheme and are not physically meaningful. In essence, updating isolated nodes generates local, physically distinct regions and the complexity introduced by such separate regions leads to a power law distribution.

We also devised updating schemes that are intermediate between synchronous and asynchronous updating. At every time step a fixed proportion r of randomly chosen elements are updated. As shown in Fig. 3, when the updating was varied from asynchronous ($r = 1/N$) to synchronous ($r = 1$), the distribution of first passage times varied continuously from a power law, to a power law with an exponential tail, to a single exponential.

As summarized in Table I, synchronous and asynchronous updating schemes result in different dynamics of the computational model. The dynamics computed by synchronous updating is consistent with other physical properties of neural networks. The dynamics computed by asynchronous updating appears to produce nonphysical results because the updating generates physically distinct regions that evolve separately and then interact, instead of the whole system evolving globally and interacting in parallel. This problem with asynchronous updating has implications for all complex computational models that use such updating schemes. It raises the question as to which dynamical results of such calculations are due to the physical properties of the model and which are artifacts of the computational method. In particular, it raises the concern that some of the dynamical properties computed in molecular dynamics simulations by asynchronous updating may arise from the nature of the updating scheme and may not be physically meaningful.[2]

ACKNOWLEDGMENT

This work was supported by NIH grant EY6234.

REFERENCES

1. D. J. Amit, *Modeling Brain Function* (Cambridge University Press, Cambridge, 1989).
2. J. A. McCammon and S. C. Harvey, *Dynamic of Proteins and Nucleic Acid* (Cambridge University Press, Cambridge, 1987).
3. K. H. Fischer and J. A. Hertz, *Spin Glasses* (Cambridge University Press, New York, 1991).
4. J. B. Bassingthwaighte, L. S. Liebovitch, and B. J. West, *Fractal Physiology* (Oxford University Press, Oxford, 1994).
5. L. S. Liebovitch, N. D. Arnold, and L. Y. Selector, Neural networks to compute molecular dynamics, *J. Biol. Sys.* **2**:193 (1994).
6. H. Fraunfelder, in *Structure and Dynamics of Nucleic Acids, Proteins, and Membranes*, E. Clementi and S. Chin, eds. (Plenum Press, New York, 1986).
7. H. Fraunfelder, S. G. Siglar, and P. G. Wolynes, *Science* **254**:1598 (1991).
8. J. J. Hopfield, *Proc. Natl. Acad. Sci, USA* **79**:2554 (1982).
9. L. S. Liebovitch, L. Y. Selector, and R. P. Kline, *Biophys. J.* **63**:1579 (1992).
10. P. Hänggi and P. Talkner, *Rev. Mod. Phys.* **62**:251 (1990).