Self-modeling in Continuous-time Hopfield Neural Networks

Mario Zarco and Tom Froese

Centro de Ciencias de la Complejidad, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
Universidad Nacional Autónoma de México

Abstract

Hopfield neural networks can exhibit many different attractors of which most are local optima. Hopfield networks are applied mainly in two cases: associative memory (Hopfield, 1982) based on a training set of patterns, and optimization (Hopfield & Tank, 1985) based on a predefined weight space which represents a constraints satisfaction problem. Watson, Buckley, and Mills (2011) have demonstrated that the discrete-time Hopfield network process of learning its own attractors – hence 'self-modeling' its previous dynamic - enlarges the basin of attraction of globally optimal attractors - hence leading to self-optimization of network connectivity. However, this approach is limited to networks with symmetric and self-recurrent connections. We are interested in knowing which topological constraints can be relaxed. Therefore, the self-modeling process is applied to a continuous-time Hopfield neural network with asymmetric and self-recurrent connections.

Introduction

The self-modeling process (SMP) is grounded in two properties of the Hopfield neural network. First, there exists a positive correlation between the width and depth of a basin of attraction (Kryzhanovsky & Kryzhanovsky, 2008). Second, associative memory allows generalizing over patterns learned. Learning is able to cause a simple form of generalization producing attractors which are new combinations of similar patterns. These spurious attractors are possible solutions of the original optimization problem. The reinforcement of an attractor at the same time can reinforce attractors with lower energy given that subpatterns that are common to many local optima can be common to superior optima (Watson et al., 2011). Thus, better attractors are reinforced more frequently due to learning, hence increasing the size of the basins of attraction, even if some of them have not been visited previously.

Watson et al. (2011) use the following iterative algorithm: (1) the network states are initialized randomly, (2) the states are updated so as to the network converges into an attractor, and (3) after reaching the attractor, small changes in the weights are applied using Hebbian learning. Watson and colleagues point out three conditions for the process to work: (C1) the initial dynamics of the system exhibit multiple point attractors; (C2) the system configurations are repeatedly relaxed from different random initial conditions such that the system samples many different attractors on a timescale where connections change slowly; (C3) the system spends most of its time at attractors. They mention also two requirements: (R1) the learning rate must be small; (R2) the time of convergence into attractors during relaxation periods must be less than τ.

Continuous-time Hopfield Neural Network

A continuous-time Hopfield neural network is a fully-connected recurrent neural network, usually with symmetric connection matrix $\Omega(t)$, and with self-recurrent connections. The network consists of N continuous states, s_i, which are updated according to the following equation (Beer, 1995):

$$\tau_i \dot{s}_i = -s_i + \sum_{j=1}^{N} \omega_{ij} \sigma(g_j(s_j + \theta_j))$$

where τ_i is the time constant of neuron i, ω_{ij} is the weight between neuron i and j, with $\omega_{ij} \in [-1, 1]$, g_j is the gain of neuron j, σ is the following activation function (Hoinville, Siles, & Hénaff, 2011):

$$\sigma(x) = \frac{2}{1 + e^x} - 1,$$

and θ_j is the next bias term (Golos, Jirsa, & Daucé, 2016):

$$\theta_i = \frac{1}{2} \sum_{j=1}^{N} \omega_{ij}.$$

This term allows the continuous version of the Hopfield network to satisfy condition (C1).

Self-modeling in Continuous-time Hopfield neural networks

We use the following algorithm: (1) the states configuration is initially randomized such that $R = [-1, 1]^N$, (2) the network is relaxed, during τ time steps, from a random configuration into an attractor using the weights modified due to learning, (3) from this last states configuration, the network is relaxed again, during τ time steps, into an attractor using the original weights, and (4) after finalizing the second relaxation period, small changes in the weights are applied using the next Hebbian rule:

$$\omega_{ij}(t+1) = \theta_{LTF}\left[\omega_{ij}(t) + \delta V_i(t) V_j(t)\right]$$

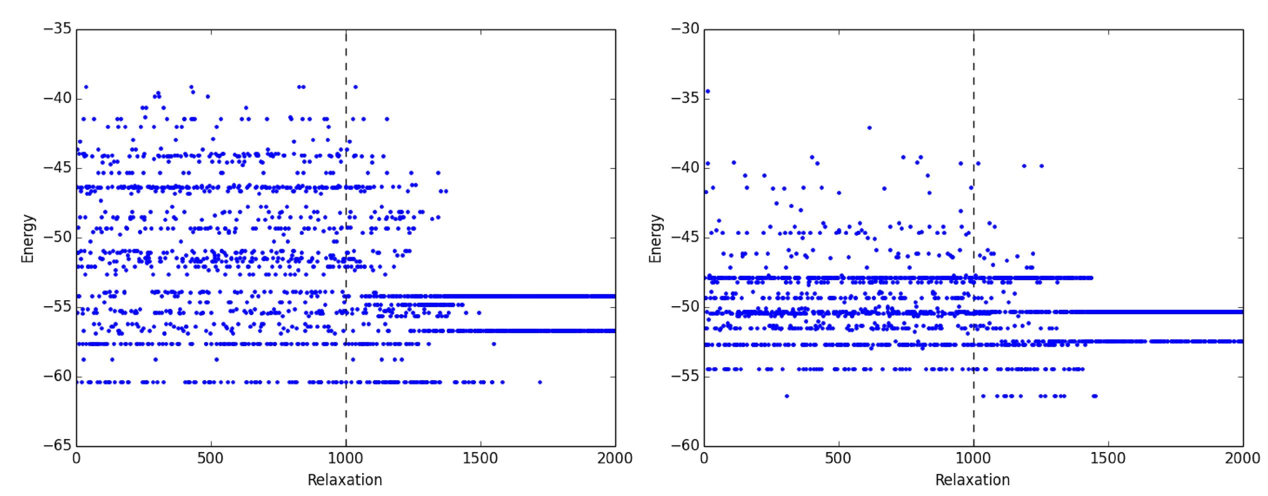
for all ω_{ij}, where δ is the learning rate, $V_j = \sigma(g_j(s_j + \theta_j))$, and θ_{LTF} is a linear threshold function. So, if $x > 0$ then $\theta_{LTF}(x) = 1$; if $x < 0$ then $\theta_{LTF}(x) = -1$; else $\theta_{LTF}(x) = x$.

An energy function for a continuous-time Hopfield network was defined by Hopfield (1984). The original energy, E^O, is used to compute the degree to which a states configuration obtained by the SMP successfully resolves the original constraints. It is calculated with the states configuration at the end of the second relaxation period, $H_O(t)$, that is

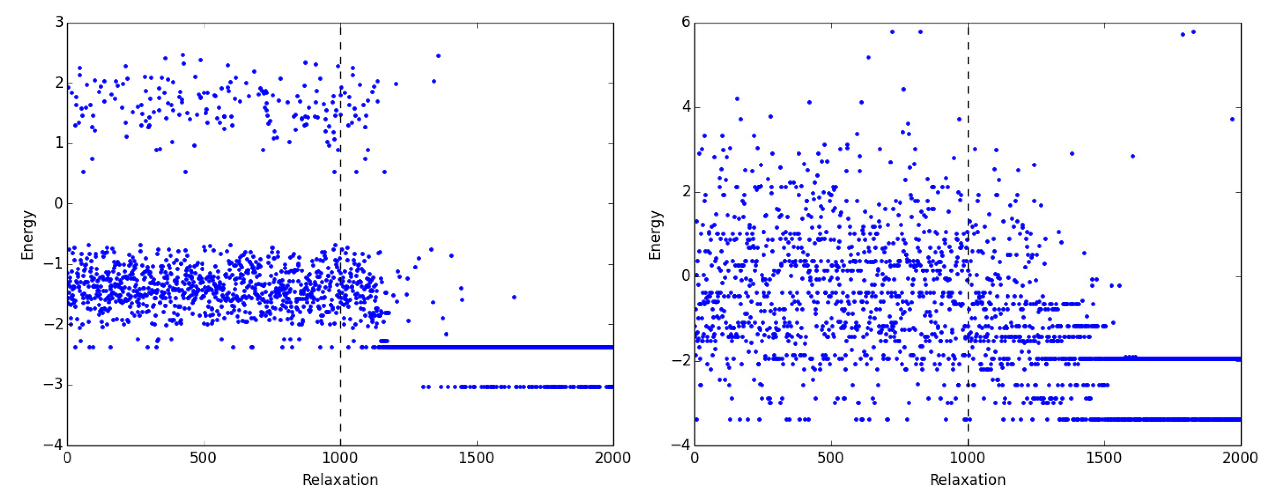
$$E^O(H(t), \Omega(t=0)) = -\frac{1}{2} \sum_{ij}^{N} \omega_{ij} V_i(t) V_j(t) + \sum_{i}^{N} \int_0^{V_i(t)} \sigma^{-1}(\xi) d\xi$$

where $\alpha_{ij} \equiv \omega_{ij}(t = 0)$. If the process works properly, the energy of the attractors will be lower over time. Finally, (C1), (C2), (C3), (R1), and (R2), are still mandatory.

Results



Symmetric (left) and asymmetric (right) random constraints.
1-1000 energy of attractors without SMP; 1001 - 2000 energy of attractors during SMP.
$N = 30$, $\tau = 1000$, $\delta = 0.0001$, $g_i = 1$



Symmetric (left) and asymmetric (right) modular constraints.
1-1000 energy of attractors without SMP; 1001 - 2000 energy of attractors during SMP.
$N = 30$, $\tau = 1000$, $\delta = 0.0001$, $g_i = 1$

References

Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, *3*(4), 469-509.

Golos, M., Jirsa, V., & Daucé, E. (2016, 12). Multistability in large scale models of brain activity. *PLOS Computational Biology*, *11*(12), 1-32.

Hoinville, T., Siles, C. T., & Hénaff, P. (2011). Flexible and multistable pattern generation by evolving constrained plastic neurocontrollers. *Adaptive Behavior*, *19*(3), 187-207.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the national academy of sciences of the united states of america* (p. 2554-2558).

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*.

Hopfield, J. J., & Tank, D. W. (1985). "neural" computation of decisions in optimization problems. *Biological Cybernetics*, *52*(3), 141–152.

Kryzhanovsky, B., & Kryzhanovsky, V. (2008). Binary optimization: On the probability of a local minimum detection in random search. In L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, & J. M. Zurada (Eds.), (pp. 89–100). Springer Berlin Heidelberg.

Watson, R. A., Buckley, C. L., & Mills, R. (2011). Optimization in "self-modeling" complex adaptive systems. *Complexity*, *16*(5), 17 - 26.