Dynamics at the Horsetooth Volume 1, 2009.

Storing Heteroclinic Cycles in Hopfield-type Neural Networks

Chuan Zhang
Department of Mathematics
Colorado State University zhang@math.colostate.edu

Report submitted to Prof. P. Shipman for Math 540, Fall 2009

Abstract. This report demonstrates how to use the pseudoinverse learning rule to store patterns and pattern sequences in a Hopfield-type neural network, and briefly discusses the effects of two parameters on the network dynamics.

Keywords: heteroclinic cycles, Hopfield-type neural networks, pseudoinverse learning rule

1 Introduction

To reveal the mystery of how a neural system stores and processes information is one of the main goals of mordern neuroscience. Recently, a self-organized criticality phenomenon, which was called "neuronal avalanche", was observed in both cultured and acute cortical slices [1, 2], and has been often associated to the capability of the network to enhance information transmission and store and retrieve information [3, 4]. In the past several decades, as a typical example of coupled nonlinear dynamical systems, formal neural networks have attracted much interest in mathematics [5, 6, 7, 8, 9]. In this report, we demonstrate how patterns and patterns sequences (heteroclinic cycles) can be stored in a hopfield-type neural network, and briefly discuss the effects of two parameters on the network dynamics.

The paper is organized as follows: in section 2, we introduce the hopfield-type neural network, pseudoinverse learning rule, and heteroclinic cycles. In section 3, we demonstrate the storage of both fixed points and heteroclinic cycles in networks. In section 4, we briefly discuss the effects of two parameters on the network dynamics.

2 Heteroclinic Cycles and Pseudoinverse Learning Rule

2.1 Neural Network Model

In this paper, we consider the following Hopfield-type neural network with continuous time [5, 6]:

$$\begin{cases} \frac{d}{dt}\mathbf{u} = -\frac{1}{\tau}\mathbf{u} + \mathbf{J}\cdot\mathbf{v} + \mathbf{I} \\ \mathbf{v} = \theta(\lambda\mathbf{u}) \end{cases} \tag{1}$$

where $\mathbf{u}$ denotes mean internal potential of neurons in the network. It is a N-dimensional column vector, and N denotes the number of neurons in the network. τ is the relaxation time of a single neuron. λ is an arbitray parameter which controls the steepness of the sigmoid activation function. $\mathbf{J}$ describes the synaptic connection among neurons, and is called the *synaptic matrix*[10, 14]. For a network of N neurons, $\mathbf{J}$ is an $N \times N$ matrix. $\theta : \mathbf{R^N} \to \mathbf{R^N}$ is an invertible function, which is bounded above and below, and usually is taken as a sigmoid function. Since it converts internal potential into firing rate output of the neurons, it is called the *activation function.* $\mathbf{I}$ is the direct external inputs, such as sensory inputs and so on, to the neurons in the network. For the dynamical system described by the equation (1), the equilibria are solutions of the equations

$$\mathbf{u} = \tau(\mathbf{J} \cdot \theta(\lambda \mathbf{u}) + \mathbf{I}). \tag{2}$$

In section 3, we consider spontaneous evolution of the network. Accordingly we neglect the direct external input term $\mathbf{I}$ by combining the external input into the initial conditions of the network. If we rescale the relaxation time τ to 1, then the conditions for the equilibria of the system become:

$$\mathbf{u} = \mathbf{J} \cdot \theta(\lambda \mathbf{u}) \tag{3}$$

2.2 Heteroclinic Cycles

Let $\xi_1, \xi_2, \cdots, \xi_m$ be the hyperbolic equilibria of a dynamical system. If there exist trajectories $\{y_1(t), y_2(t), \cdots, y_m(t)\}$ with the property that $y_i(t)$ is backward asymptotic to ξ_i and forward to ξ_{i+1}, then the collection of trajectories $\{\xi_i, y_i(t)\}$ is called a *heteroclinic cycle* [12]. Figure 1 illustrates an example of a heteroclinic cycle.

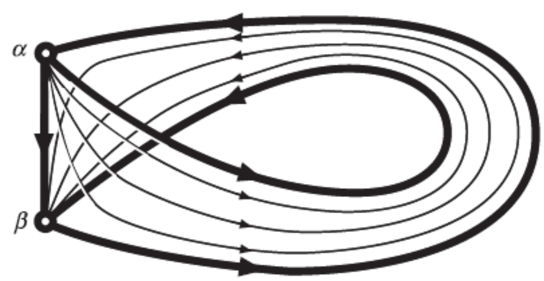

Figure 1: Sketch of a flow on a Möbius band that is a *heteroclinic cycle* from α to β and back. Taken from [12]

2.3 Storage of Patterns and Pattern Sequences

For a network designed to store and retrieve information, the storage takes place in the synapses. Several ways, which are called *learning rules*, can be used to determine the synaptic matrix [13, 14, 15, 16]. Inspired from biological nervous systems, a simple but elegant and widely used learning rule, which was called the *Hebbian learning rule* was introduced in 1949 [13]:

$$J_{ij} = \frac{1}{N} \sum_{\mu} \xi_i^{\mu} \xi_j^{\mu}. \tag{4}$$

However, as is known, this learning rule is well-suited for random uncorrelated patterns only, whereas in real world patterns entering the network can be correlated. To address this issue, Personnaz et al [17, 18] proposed the pseudo-inverse (PI) learning rule. The PI learning rules, which are mainly used in this paper, can be implemented with two different procedures [14].

The synaptic matrix produced by the first procedure, which requires that the patterns to be memorized are orthogonal to each other, is called the standard PI synaptic matrix; and the synaptic matrix produced by the second procedure, which does not require the patterns to be orthogonal to each other, is called the sequential PI synaptic matrix. Here we refer to the two procedures as the *standard PI learning rule* and the *sequential PI learning rule* respectively. In section 3, all simulations are implemented by the *standard PI learning rule.*

2.3.1 Standard PI learning rule

In the Hopfield-type neural network model (1), the memorized patterns are read from the output firing rates of the neurons in the neural network. Since the output firing rate of each neuron is denoted by v_i, $i = 1, 2, \ldots, N$, we have $v_i = \theta(\lambda u_i)$, i.e., $u_i = \frac{1}{\lambda}\theta^{-1}(v_i)$. Thus the memorized patterns $\{\xi^{(\mu)}\}$ are actually the stationary solutions (equilibria) to the following equation:

$$-\frac{1}{\lambda}\theta^{-1}(\mathbf{v}) + \mathbf{J}\cdot\mathbf{v} = 0. \tag{5}$$

Suppose $\xi^{(i)}\cdot\xi^{(j)} = 0$ for $i \neq j$, and the patterns $\xi^{(\mu)}$ form a matrix Σ, where $\Sigma_{i,j} = \xi_i^{(j)}$. Then

$$\frac{1}{\lambda}\theta^{-1}(\Sigma) = \mathbf{J}\cdot\Sigma. \tag{6}$$

Approximate the inverse sigmoid activation function $\frac{1}{\lambda}\theta^{-1}(\xi_i^{(\mu)})$ by a linear function $\beta\xi_i^{(\mu)}$, then (6) can be solved as:

$$\mathbf{J} = \beta\Sigma\cdot\Sigma^{+} \tag{7}$$

where Σ^{+} is the *pseudoinverse* of Σ, and can be calculated by $\Sigma^{+} = (\Sigma^T\Sigma)^{-1}\Sigma^T$.
Since the patterns stored in the network are equilibria of the coupled dynamical system 1, in order to adjust the synaptic matrix to memorize pattern sequences, extra conditions must be imposed. Let $\{\xi^{(\mu)}\}_{\mu=1}^{p}$ be the sequence to be stored in the network. Due to the symmetry of the activation function θ, the sequence can form a cycle in two ways: $\xi^{(1)} \to \xi^{(2)} \to \cdots \to \xi^{(p)} \to \xi^{(1)}$ and $\xi^{(1)} \to \xi^{(2)} \to \cdots \to \xi^{(p)} \to -\xi^{(1)} \to -\xi^{(2)} \to \cdots \to -\xi^{(p)} \to \xi^{(1)}$. Here we take the second way. Thus, the conditions for storing pattern sequences are:

$$\frac{1}{\lambda}\theta^{-1}(\xi_i^{(\mu+1)}) = \sum_{j=1}^{N} J_{ij}\xi_j^{(\mu)} \ (1 \leq \mu \leq p) \tag{8}$$

where $\xi_i^{(p+1)} = -\xi_i^{(1)}$. Similarly, the equation (8) can be rewrite in matrix form:

$$\mathbf{J} = \beta\Sigma_1\cdot\Sigma^{+}, \tag{9}$$

where $\Sigma_1 = (\xi^{(2)}\ \xi^{(3)}\ \cdots - \xi^{(1)})$.

2.3.2 Sequential PI learning rule

Suppose the initial synaptic matrix has stored a pattern sequence $\{\xi^{(\mu)}\}_{\mu=1}^{p}$, in which $\xi^{(i)}\cdot\xi^{(j)} = 0$ for $i \neq j$, and the patterns $\xi^{(\mu)}$ form a matrix Σ, where $\Sigma_{i,j} = \xi_i^{(j)}$. Since the memorized patterns are equilibria of the network, following the condition 3, we have $\mathbf{J}\cdot\Sigma = \theta^{-1}(\Sigma)$, i.e.,

$$\mathbf{J} = \Sigma\cdot\Sigma^{+} \tag{10}$$

Suppose an arbitrarily given pattern ζ is adding to the sequence which has been memorized, then the updated synaptic matrix is given by the *sequential PI learning rule*

$$\tilde{\mathbf{J}} = \mathbf{J} + \frac{(\zeta - \mathbf{J}\zeta) \otimes (\zeta - \mathbf{J}\zeta)}{N - \zeta^T J \zeta}, \tag{11}$$

where $\otimes$ denotes the direct product of two arbitrarily given vectors:

$$[\mathbf{A} \otimes \mathbf{B}]_{ij} = \mathbf{A}_i \mathbf{B}_j$$

The updated synaptic matrix satisfies the following conditions:

$$\tilde{\mathbf{J}}\Sigma = \Sigma \tag{12}$$

$$\tilde{\mathbf{J}}\zeta = \zeta \tag{13}$$

Following the same reasoning in the preceding section, the condition for storing pattern sequences can be obtained. Since in the next section, only the standard PI learning rule will be used in simulations. So we are not going to discuss the sequential PI learning method in more details.

3 Simulations

In this section we demonstrate how to save patterns and pattern sequences in equilibria and heteroclinic cycles. First, we demonstrate storing of patterns as equilibria in a network. Consider a continuous Hopfield-type neural network:

$$\frac{du_i}{dt} = -u_i + \sum_{j=1}^{N} J_{ij} \tanh(\lambda u_j) \tag{14}$$

In order to combine the conditions for storing patterns and pattern sequences, we rewrite the equations in terms of the firing rate $v_i = \tanh(\lambda u_i)$:

$$\frac{dv_i}{dt} = \lambda(1 - v_i^2(t))[-\frac{1}{\lambda}\text{arctanh}(v_i(t)) + \sum_{j=1}^{N} J_{ij} v_j(t)] \tag{15}$$

3.1 Storing Patterns as Equilibria

Let $\xi^{(1)}, \xi^{(2)}, \cdots, \xi^{(p)}$, be p patterns (binary vectors) to be stored in the network. Then $v_i = \beta_1 \xi_i^{(\mu)}$ with $0 < \beta_1 \leq 1$ are equilibria of the system (15), i.e.,

$$\lambda(1 - (\beta_1 \xi_i^{(\mu)})^2)[-\frac{1}{\lambda}\text{arctanh}(\beta_1 \xi_i^{(\mu)}) + \beta_1 \sum_{j=1}^{N} J_{ij} \xi_i^{(\mu)}] = 0$$

Since $\xi^{(\mu)}$ is an arbitrarily given binary vector, we have

$$-\frac{1}{\lambda}\text{arctanh}(\beta_1 \xi_i^{(\mu)}) + \beta_1 \sum_{j=1}^{N} J_{ij} \xi_i^{(\mu)} = 0$$

i.e.,

$$\frac{1}{\lambda}\text{arctanh}(\beta_1 \xi_i^{(\mu)}) = \beta_1 \sum_{j=1}^{N} J_{ij} \xi_i^{(\mu)}$$

Since for $|kx| < 1$ and $0 < k \leq 1$, $\text{arctanh}(kx) \approx \text{arctanh}(k)x$, we have:

$$\frac{1}{\lambda}\text{arctanh}(\beta_1)\xi_i^{(\mu)} = \beta_1 \sum_{j=1}^{N} J_{ij}\xi_i^{(\mu)}$$

$$\frac{\text{arctanh}(\beta_1)}{\lambda\beta_1}\xi_i^{(\mu)} = \sum_{j=1}^{N} J_{ij}\xi_i^{(\mu)}$$

Let $\beta_K = \frac{\text{arctanh}(\beta_1)}{\lambda\beta_1}$, we get:

$$\beta_K \xi_i^{(\mu)} = \sum_{j=1}^{N} J_{ij}\xi_i^{(\mu)}$$

Since the patterns can be written in matrix form: $\Sigma = [\ \xi^{(1)} \quad \xi^{(2)} \quad \cdots \quad \xi^{(p)}\ ]$, the above equation can be rewritten as:

$$\beta_K \Sigma = \mathbf{J} \cdot \Sigma$$

i.e.,

$$\mathbf{J} = \beta_K \Sigma \Sigma^{+}$$

where Σ^{+} is the *pseudoinverse* of Σ. Given a group of patterns to be memorized Σ, β_1 and λ, the temporal evolution of an arbitrarily given pattern can be shown numerically by solving the differential equations (15). Our numerical simulations indicates that the larger λ is, the more stable the equilibria are, and the faster the orbits approach the equilibria.
Here we illustrate the results of one simulation. We choose

$$\Sigma = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & 1 & 1 \end{pmatrix}$$

i.e., we choose $N = 4$, $p = 3$, and $\xi^{(1)} = (1\ \text{-1}\ \text{-1}\ \text{-1})^T$, $\xi^{(2)} = (1\ 1\ \text{-1}\ 1)^T$, and $\xi^{(3)} = (1\ \text{-1}\ 1\ 1)^T$. Here $\xi^{(1)}, \xi^{(2)}$, and $\xi^{(3)}$ are orthogonal to each other, so we can determine the synaptic matrix by $\mathbf{J} = \beta_K \Sigma \Sigma^{+}$ directly. Here we set $\beta_1 = 0.99$, $\lambda = 2$, we get:

$$\mathbf{J} = \begin{pmatrix} 1.0025 & -0.3342 & -0.3342 & 0.3342 \\ -0.3342 & 1.0025 & -0.3342 & 0.3342 \\ -0.3342 & -0.3342 & 1.0025 & 0.3342 \\ 0.3342 & 0.3342 & 0.3342 & 1.0025 \end{pmatrix}$$

Figure 2 shows the results of one simulation with Euler method, where $dt = 10$ ms, and $\mathbf{v}(0) = (\ 0.9611 \quad -0.9982 \quad 0.2913 \quad -0.9837\ )^T$. Figure 2 (A) shows the firing rate of each of the four neurons in the network as a function of time. It is easy to see that although the initial pattern is very different from any one the three memorized patterns, the firing rate of the four neurons quickly evolves into the closest memorized pattern. In order to describe the similarity between the pattern of the firing rate of the four neurons and each one of the memorized patterns, "overlap" $\mathbf{m}(t)$ is defined as:

$$\mathbf{m}(t) = (\ m_1(t) \quad m_2(t) \quad \cdots \quad m_p(t)\ )$$

where

$$m_\mu(t) = \frac{1}{N}\sum_{i=1}^{N} \xi_i^{(\mu)} v_i(t).$$

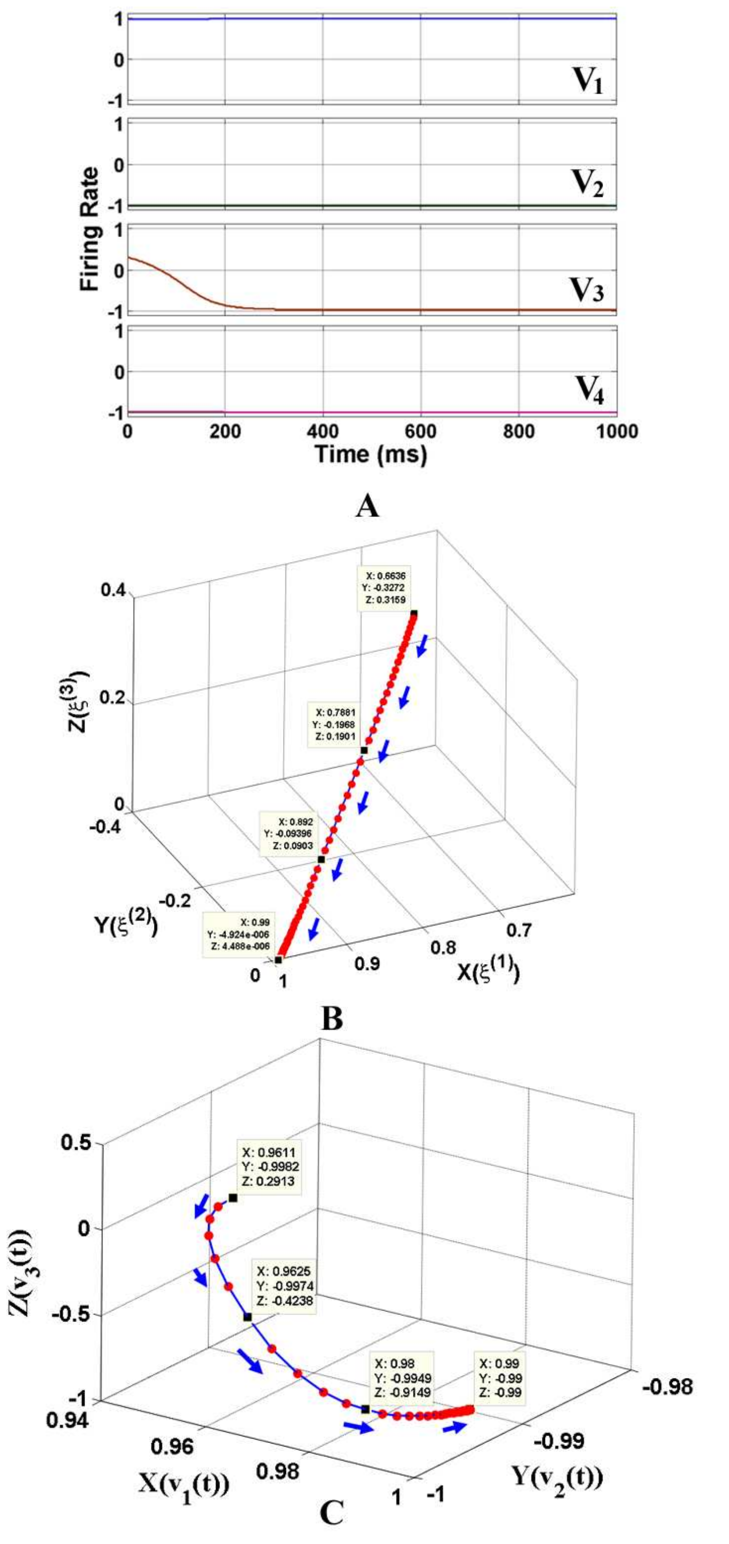

Figure 2: Example of retrieving one of the patterns (1 -1 -1 -1) stored as equilibria. (A) the temporal evolution of firing rate $v_i(t)$ of the four neurons in the network; (B) overlap $\{m_1(t), m_2(t), m_3(t)\}$; (C) the 3D projection of the phase portrait into $(v_1 - v_2 - v_3)$ subspace. The blue arrows indicates the state evolution direction of the network.

Figure 2 (B) shows that the pattern of the firing rate of the four neurons becomes more and more similar to the memorized pattern $\xi^{(1)}$ directly. In figure 2 (C) we illustrate the three dimensional projection of the four dimensional phase space. From this projection, we also can see that the phase trajectory of the network moves to the equilibrium, which corresponds to the memorized pattern $\xi^{(1)}$ directly.

3.2 Storing Pattern Sequences as Heteroclinic Cycles

Let $\xi^{(1)}$, $\xi^{(2)}$, $\cdots$, $\xi^{(p)}$, be p patterns (binary vectors) memorized in the network as hyperbolic equilibria. Suppose these equilibria can be linked by heteroclinic orbits. Then, the pattern sequences can be memorized as heteroclinic cycles. According to the transition condition (9), we have

$$-\frac{1}{\lambda}\text{arctanh}(\beta_1 \xi_i^{(\mu+1)}) + \beta_1 \sum_{j=1}^{N} J_{ij}\xi_i^{(\mu)} = 0$$

i.e.,

$$\frac{1}{\lambda}\text{arctanh}(\beta_1 \xi_i^{(\mu+1)}) = \beta_1 \sum_{j=1}^{N} J_{ij}\xi_i^{(\mu)}$$

Since for $|kx| < 1$ and $0 < k \leq 1$, $\text{arctanh}(kx) \approx \text{arctanh}(k)x$, thus:

$$\frac{\text{arctanh}(\beta_1)}{\lambda\beta_1}\xi_i^{(\mu+1)} = \sum_{j=1}^{N} J_{ij}\xi_i^{(\mu)}$$

Let $\beta_K = \frac{\text{arctanh}(\beta_1)}{\lambda\beta_1}$, we get:

$$\beta_K \xi_i^{(\mu+1)} = \sum_{j=1}^{N} J_{ij}\xi_i^{(\mu)}$$

Since $\xi^{(p+1)} = -\xi^{(1)}$, the patterns can be written in matrix form: $\Sigma_1 = [\ \xi^{(2)} \quad \xi^{(3)} \quad \cdots \quad \xi^{(p)} \quad -\xi^{(1)}\ ]$. Thus the above condition can be rewritten as:

$$\beta_K \Sigma_1 = \mathbf{J} \cdot \Sigma$$

i.e.

$$\mathbf{J} = \beta_K \Sigma_1 \Sigma^{+}.$$

However, here in order to combine both the fixed-point and transition behaviors in one network, a pair of weighting factors C_0 and C_1 are introduced and the pseudoinverse learning rule becomes:

$$\mathbf{J} = \frac{\beta_K}{2}(C_0 \Sigma\Sigma^{+} + C_1 \Sigma_1 \Sigma^{+}) \tag{16}$$

Numerical simulations indicates that the larger β_K and λ respectively are, the closer to equilibria the orbits will be, and the longer the orbits will stay around the equilibria. Next, we demonstrate a simulation of storing and retrieving a given pattern sequence in the network. In this simulation we use the network used in the preceding example. First we set $\beta_1 = 0.9999$, $\lambda = 10$, and use the same patterns: $\xi^{(1)} = (1\ \text{-1}\ \text{-1}\ \text{-1})^T$, $\xi^{(2)} = (1\ 1\ \text{-1}\ 1)^T$, and $\xi^{(3)} = (1\ \text{-1}\ 1\ 1)^T$. Thus

$$\Sigma = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & 1 & 1 \end{pmatrix}$$

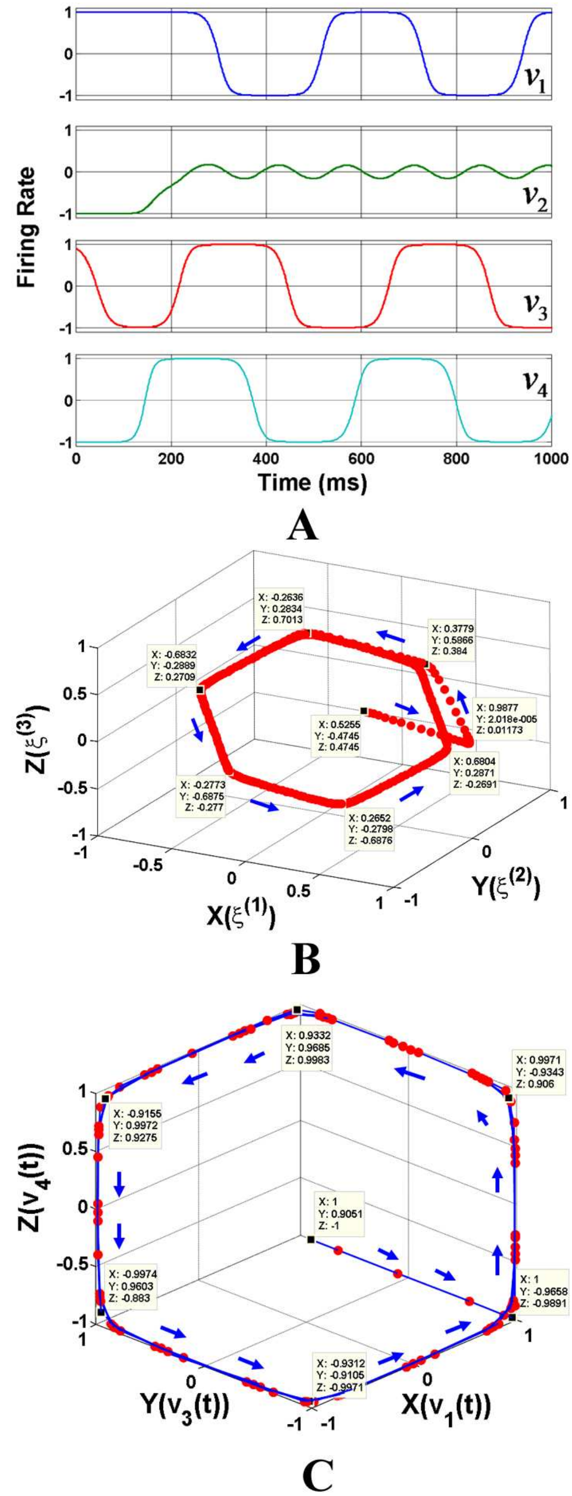

Figure 3: Example of retrieving one of the pattern sequences ($\xi^{(1)} \rightarrow \xi^{(2)} \rightarrow \xi^{(3)} \rightarrow -\xi^{(1)} \rightarrow -\xi^{(2)} \rightarrow -\xi^{(3)} \rightarrow \xi^{(1)}$) stored as a heteroclinic cycle. (A) the temporal evolution of firing rate $v_i(t)$ of the four neurons in the network; (B) overlap $\{m_1(t), m_2(t), m_3(t)\}$; (C) the 3D projection of the phase portrait into ($v_1 - v_3 - v_4$) subspace. The blue arrows indicates the state evolution direction of the network.

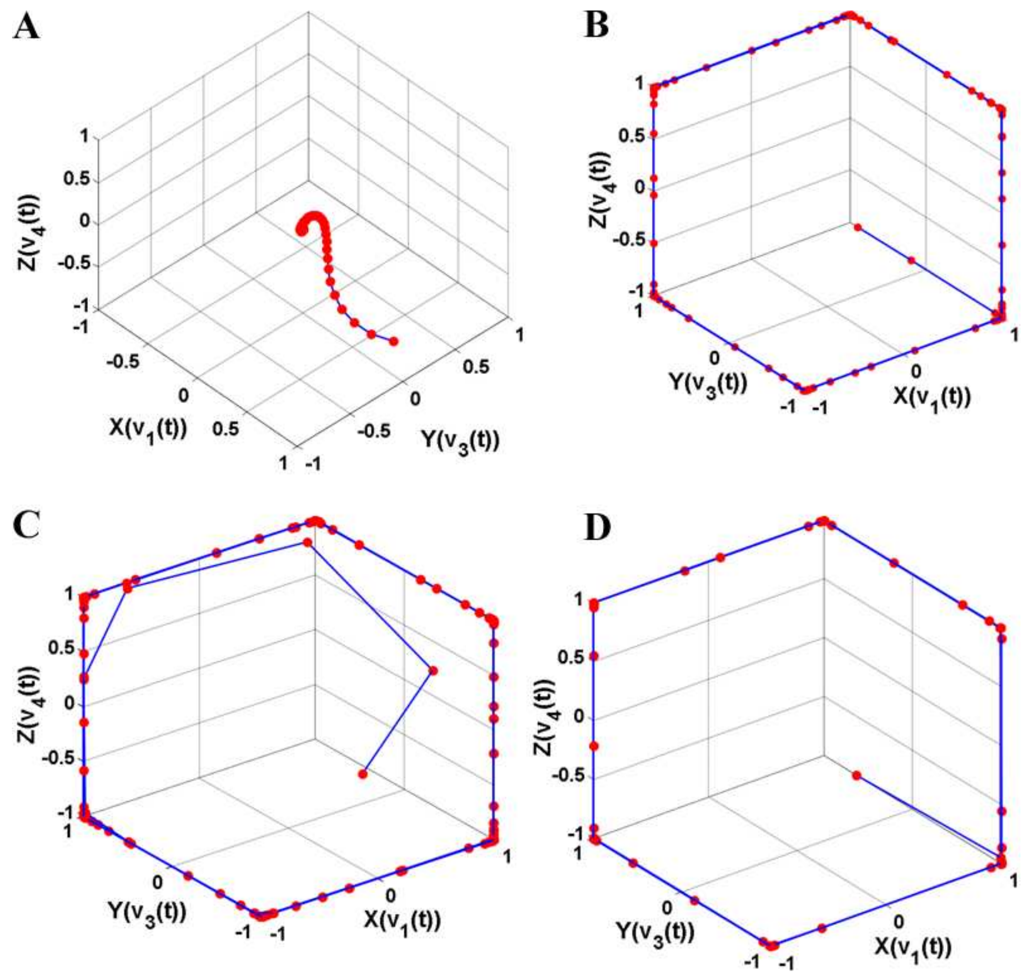

Figure 4: Effects of the parameters β_K and λ on the network dynamics. (A) $\beta_K = 1$, $\lambda = 1$; (B) $\beta_K = 1$, $\lambda = 10$; (C) $\beta_K = 10$, $\lambda = 1$; (D) $\beta_K = 5$, $\lambda = 5$.

and

$$\Sigma_1 = \begin{pmatrix} 1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

We set $C_0 = 0.1$, $C_1 = 1.9$, then by formula (13), we get:

$$\mathbf{J} = \begin{pmatrix} 0.1362 & 0.1114 & -0.3590 & -0.1114 \\ 0.1114 & -0.3343 & 0.1114 & -0.1114 \\ 0.1114 & 0.1114 & 0.1362 & 0.3590 \\ 0.3590 & -0.1114 & -0.1114 & 0.1362 \end{pmatrix}$$

In figure 3, we set initial firing rate as: $\mathbf{v}(0) = (\ 0.9611 \quad -0.9982 \quad 0.2913 \quad -0.9837\ )^T$, and choose $dt = 10$ ms as the iteration step size, and use the Euler method to numerically solve the differential equations (15). Figure 3(A) illustrates the firing rate of the four neurons, figure 3(B) illustrate the overlap of the orbit starting from $\mathbf{v}(0)$ as a function of time, and figure 3(C) illustrate the three dimensional projection of the four dimensional phase space. From this projection, we can see that the phase trajectory of the network asymptotically converges to the heteroclinic cycle stored in the network. From figure 3(C) we can see that the six equilibria the orbit visits are,

$v_1(t)$	$v_2(t)$	$v_3(t)$	$v_4(t)$
1	?	−1	−1
1	?	−1	1
1	?	1	1
−1	?	1	1
−1	?	1	−1
−1	?	−1	−1

By checking the simulation results, we fill in the second column and see that these six points are exactly $\xi^{(1)} \to \xi^{(2)} \to \xi^{(3)} \to -\xi^{(1)} \to -\xi^{(2)} \to -\xi^{(3)}$.

4 Discussions

Finally, we discuss the effects of the two parameters, β_K and λ. Since in the system of firing rate, β_1 will not appear, β_K and λ actually become independent of one another. And in numercial simulations, we saw that those two parameters affect the rate that the orbit converges to the heteroclinic cycle. So in this section, we briefly illustrate that increasing each or both of these two parameters will increase the rate of the convergence of the orbit starting at some prescribed initial point to the memorized heteroclinic cycle.
Figure 4(A) shows that when both β_K and λ are small ($\beta_K = 1$, $\lambda = 1$), the orbit failed to converge to the heteroclinic cycle; in other words, the attempt of retrieving the stored pattern sequence failed. Figure 4(B), (C), and (D) show that when each or both of the two parameters are increased, the orbit converges to the stored heteroclinic cycle, and the effect of increasing λ seems more significant than β_K.

5 Conclusion

In summary, in this report, we have shown how patterns or pattern sequences can be stored and retrieved in a Hopfield-type neural network, by using the standard PI learning rule. When combined with the sequential PI learning rule, the network can be used to store and retrieve patterns and pattern sequences in real-time.

References

[1] Beggs, J.M. and Plenz, D. Neuronal avalanches in neocortical circuits. *J Neurosci* 23(35), 11167 11177, 2003.

[2] Beggs, J.M. and Plenz, D. Neuronal avalanches are diverse and precise activity patterns that are stable for many hours in cortical slice cultures. *J Neuronsci* 24(22), 5216 5229, 2004.

[3] Baruchi, I. and Ben-Jacob E. Towards neuro-memory-chip: Imprinting multiple memories in cultured neural networks. *Physical Review E.* 75, 050901-1 050901-4, 1984.

[4] Plenz, D. and Thiagarajan T.C. The organizing principles of neuronal avalanches: cell assemblies in the cortex? *TRENDS in Neurosciences* 30(3), 101-110, 2007.

[5] Hopfield, J.J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. USA* 81(10), 3088-3092, 1984.

[6] Hopfield, J.J. and Tank D.W. Computing with neural circuits: a model. *Science* 233, 625-633, 1986.

[7] Golubitsky, M., and Stewart, I. Nonlinear dynamics of networks: the groupiod formalism. *Bull. Amer. Math. Soc.* 43(3), 305-364, 2006.

[8] Field, M. Combinatorial dynamics. *Dyn. Sys.* 19(3), 217-243, 2004.

[9] Ashwin P. and Borresen J. Encoding via conjugate symmetries of oscillations for globally coupled oscillators. *Phys. Rev. E* 70, 026203, 2004.

[10] Lappe, M. Storage of patterns and pattern sequences in artificial neural networks. Master Degree Thesis, Tübingen Universtiy 1989. (In German)

[11] Genic, T. Lappe, M. Dangelmayr, G. and Güttinger W. Storing cycles in analog neural networks. *Parallel Processing in Neural Systems and Computers* Eckmiller, R. Hartmann, G. and Hauske, G., Eds. Elsevier Science Publishers, 445-450, 1990.

[12] Ashwin, P. and Field, M. Heteroclinic Networks in Coupled Cell Systems. *Arch. Rational Mech. Anal.* 148, 107-143, 1999.

[13] Hebb, D.O. The organization of behavior, New York: Wiley, 1949

[14] Linkevich , A.D. A sequential pseudo-inverse learning rule for networks of formal neurons. *J. Phys. A: Math. Gen.* 25, 4139-4146, 1992

[15] Van Hulle, M.M. Globally-ordered topology-preserving maps achieved with a learning rule performing local weight updates only. *Proc. IEEE NNSP* 95, 95-104, 1995

[16] Lapedes, A. and Farber, R. A self-optimizing, nonsymmetrical neural net for content addressable memory and pattern. *Physica D* 22, 247-259, 1986

[17] Personnaz, L. Guyon, I. and Dreyfus, G. Information storage and retrieval in spin-glass like neural networks. *J. Physique Lett.* 46, 359-365, 1985

[18] Personnaz, L. Guyon, I. and Dreyfus, G. Collective computational properties of neural networks: New learning mechanisms. *Physical Review A* 34(5), 4217-4228, 1986