2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Retrieval Characteristics of a Dynamical Associative Memory Using Nonlinear Support Vector Machines

Akinori KATO† and Masaharu ADACHI‡

†Department of Electronic Engineering, Graduate School of Engineering, Tokyo Denki University
2-2 Kanda–Nishiki–cho, Chiyoda–ku, Tokyo, 101–8457, Japan
Email: †08gmd05@ms.dendai.ac.jp, ‡adachi@eee.dendai.ac.jp

Abstract—In this paper, we introduce chaotic model neurons as the constituents of the associative memory model with nonlinear SVMs. We examine the retrieval characteristics of the dynamical associative memory model through numerical experiments. As for the stored patterns used in our previous paper[1], we have demonstrated that a dynamic associative memory can be realized by the proposed model. In this paper, we investigate retrieval characteristics of recall and transition frequencies of the stored patterns with considering correlation between them.

1. Introduction

An associative memory retrieves the stored information that is closest to the cue input information. Such information processing may be realized in the brain. Therefore, associative memory models have been studied not only for engineering applications but also for studying the mechanism of memory in the brain. Deterministic chaos has been considered to play an important role in the brain. Therefore, the relationship between the information processing function of the brain and deterministic chaos has been investigated by many researchers. As an example, a dynamical associative neural network model has been proposed which incorporates a chaotic neuron [2] as a constituent element of the network [1].

In contrast to a static associative network, a dynamical associative network does not always retrieve the closest stored pattern but recalls other stored patterns in a periodic sequence [3]. Most conventional dynamical associative memory determines synaptic connection weights w using Hebb learning with patterns to be stored or using an auto-correlation matrix of the patterns [1]. In this paper, we introduce the pattern classification technique of nonlinear Support Vector Machines (SVMs) as a learning method for determining the synaptic connection weights and the biases. We construct a dynamical associative network by connecting chaotic neurons with these synaptic weights. Then, we investigate retrieval characteristics of recall and transition frequencies of the stored patterns with considering correlation between them.

2. Support Vector Machine

A Support Vector Machine [4] is an effective classifier for binary classification problems. We consider a set of two classes of classification problems for $(\mathbf{x}^{(m)}, y^{(m)})$, $m = 1, ..., M$, where M is the number of training data. $\mathbf{x}^{(m)} \in R^N$ and $y^{(m)} \in \{-1, +1\}$ are the learning data and class label, respectively. SVMs can be realized as neural networks, the synaptic connection weights $\mathbf{w} \in R^N$ and the bias b of the constituent neurons in the network can be determined by the following procedures using training data and their class labels.

$$y^{(m)}\left(\sum_{i=1}^{N} w_i x_i^{(m)} + b\right) > 0 \tag{1}$$

Given an unknown input vector $\mathbf{x}^{(\mathrm{u})}$ and its class label, we classify it as follows:

$$f(\mathbf{x}^{(\mathrm{u})}) = \sum_{i=1}^{N} w_i x_i^{(\mathrm{u})} + b \begin{cases} > 0 & (class\ y^{(\mathrm{u})} = +1) \\ < 0 & (class\ y^{(\mathrm{u})} = -1) \end{cases} \tag{2}$$

The hyperplane $f(\mathbf{x}) = 0$ that represents the class boundary is called the separation plane. The distance between the training data point which is the nearest to the separation plane (the Support Vector) and the plane is termed the margin. The most important feature of an SVM is that it can determine $\mathbf{w}$ and b so that they maximize the margin between two classes. Therefore, the SVM can obtain generalization performance for unknown data. In addition, the separation plane $f(\mathbf{x}) = 0$ is not distinguished from the rest of the data.

For learning data which are not separable by a plane in their input space, one can use a nonlinear map $\Phi(\cdot)$ to map them in a higher dimensional feature space and construct a separation plane in the higher dimensional space.

Margin maximization in this feature space can be obtained by solving a convex quadratic programming problem, which is a standard optimization problem. Concretely, the constraint conditions for this problem are

$$\sum_{m=1}^{M} \alpha^{(m)} y^{(m)} = 0 \tag{3}$$

$$\alpha^{(m)} \geq 0, \quad m = 1, ..., M. \tag{4}$$

Subject to Eq. (3) and Eq. (4), we maximize the objective function

$$J(\alpha) = \sum_{m=1}^{M} \alpha^{(m)} - \frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{M} \alpha^{(m)} \alpha^{(n)} y^{(m)} y^{(n)} K\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right) \tag{5}$$

where $\alpha^{(m)}$ is a Lagrange multiplier. A kernel function $K(\cdot)$ gives a measure of the similarity between two vectors $\Phi(\mathbf{x}^{(m)})$ and $\Phi(\mathbf{x}^{(n)})$ that are mapped in the higher dimensional feature space. Utilizing the kernel function avoids the calculation of the inner product of $\Phi(\mathbf{x}^{(m)}) \cdot \Phi(\mathbf{x}^{(n)})$ in the high dimensional space and has the advantage that one can calculate the measure of similarity in a space with the dimension of the input space. This is the kernel trick, and it makes it possible for an SVM to classify data having nonlinear separation planes.

With the Lagrange multiplier $\alpha^{(m)}$ that maximizes the objective function of Eq. (5), we assume that connection coefficient $w^{(m)}$ is

$$w^{(m)} = \alpha^{(m)} y^{(m)} \Phi(\mathbf{x}^{(m)}). \tag{6}$$

In addition, the bias value b describes an arbitrary support vector in the following equation for sample data $(\mathbf{x}^{(s)}, y^{(s)})$.

$$\begin{aligned} b &= y^{(s)} - \sum_{m=1}^{M} \alpha^{(m)} y^{(m)} \Phi(\mathbf{x}^{(m)}) \Phi(\mathbf{x}^{(s)}) \\ &= y^{(s)} - \sum_{m=1}^{M} W^{(m)} K\left(\mathbf{x}^{(m)}, \mathbf{x}^{(s)}\right) \end{aligned} \tag{7}$$

where $W^{(m)} = \alpha^{(m)} y^{(m)}$ and $\Phi(\mathbf{x}^{(m)})\Phi(\mathbf{x}^{(s)}) = K(\mathbf{x}^{(m)}, \mathbf{x}^{(s)})$. Furthermore, the discriminant function of a nonlinear SVM is given by

$$f(\mathbf{x}^{(\mathrm{u})}) = \sum_{m=1}^{M} W^{(m)} K\left(\mathbf{x}^{(\mathrm{u})}, \mathbf{x}^{(m)}\right) + b \begin{cases} > 0\,(class\ \ y^{(\mathrm{u})} = +1) \\ < 0\,(class\ \ y^{(\mathrm{u})} = -1) \end{cases} \tag{8}$$

The nonlinear SVM is represented by a three-layer neural network. The first layer is the input layer. This layer just transfers input data to the second layer which performs the calculation of the measure of similarity between the input data and the training data by using the kernel function. The output layer operates as in Eq. (8) and classifies the input data.

In this paper, we use the Gaussian (RBF) kernel function $K(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}) = \exp(- \| \mathbf{x}^{(m)} - \mathbf{x}^{(n)} \|^2 / 2\sigma^2)$. The variance σ^2 of the Gaussian kernel function is used to determine the dimension of the space in which to perform planar separation of the training data. In addition, we use the Sequential Minimal Optimization (SMO) method [5] for maximization of the objective function in Eq. (5).

3. Chaotic Neural Networks

A chaotic neural network is a neural network model that consists of model neurons that exhibit deterministic chaos. Such a network can avoid being trapped in a local minimum which conventional static neural networks can easily fall into. This means that chaotic neural networks have the possibility of achieving the globally optimal solution. Therefore, such nonequilibrium dynamics can be also used for realizing dynamical associative memory. The state updating of the ith neuron in the network is given by

$$\begin{aligned} x_i(t+1) &= f_i(\eta_i(t+1) + \zeta_i(t+1) + A), && (9) \\ \eta_i(t+1) &= k_f \eta_i(t) + \sum_{j=1}^{N} w_{ij} x_j(t), && (10) \\ \zeta_i(t+1) &= k_r \zeta_i(t) - \alpha x_i(t) + b_i, && (11) \\ f(y) &= \frac{2}{1 + \exp(-y/\epsilon)} - 1 && (12) \end{aligned}$$

where $x_i(t)$ is the output of the ith neuron at discrete time t, and $\eta_i(t)$ and $\zeta_i(t)$ are internal states for feedback input to the neuron from other neurons in the network and for refractoriness of the neuron, respectively. The sigmoid function $f(y)$ is the output function of the neuron. Equation (12) is different from the sigmoid function used in Refs.[2] and [1]. The reason is that each element of the stored patterns used in this paper consists of binary vectors $\{+1, -1\}$. Therefore, the output function has to be chosen so that the chaotic neuron takes values from -1 to $+1$. The parameter ϵ determines the steepness of the sigmoid function and affects the sensitivity of firing. The parameters k_f and k_r are the decay constants of η_i and ζ_i, respectively. The parameter α is a refractory scaling parameter. A denotes an external input to the chaotic neuron. All chaotic neurons in the network have a common value of A in this study. The parameters w_{ij} are synaptic weights (to the ith chaotic neuron from the jth one), and b_i is the bias of the ith chaotic neuron.

4. Proposed method

4.1. Determination of the connection coefficients and the bias by SVM

Using an SVM for determining synaptic weights to store patterns can be seen as a classification problem of the N unit with training data $(\mathbf{x}^{(m)}, x_i^{(m)})$ (for, $m = 1, ..., M$), where $x_i^{(m)} \in \{-1, +1\}$, $i = 1, ..., N$) represents the class label for the input $\mathbf{x}^{(m)}$. We can solve each of these N pair classification problems by using an SVM. The synaptic weight vector $\mathbf{w}$ and the bias b that distinguishes stored patterns can be obtained by performing the above operation N times. Here, it should be noted that the size of the synaptic weight matrix $\mathbf{w}$, which is obtained using a linear SVM for storing patterns, is $N \times N$, while the one obtained using a nonlinear SVM is $N \times M$. Therefore, we rewrite Eq. (10) for updating η_i of the chaotic neuron as follows:

$$\eta_i(t+1) = k_f \eta_i(t) + \sum_{m=1}^{M} W_i^{(m)} K\left(\mathbf{x}^{(m)}, \mathbf{x}(t)\right). \tag{13}$$

The difference between the matrix sizes of linear and nonlinear SVMs is caused by the fact that the nonlinear SVM uses a kernel function $K(\cdot)$. Using the kernel function ensures that a stored pattern whose separation plane is nonlinear can be embedded into W and b.

4.2. Structure of proposed dynamical associative network

We show the structure of our model in Figure 1. The first layer is the input layer which transfers the input pattern to the second layer. The second layer is the kernel function layer. The boldface **1** in Figure 1 represents the fact that all the synaptic weights from the input layer to the kernel function layer are fixed at one. The kernel function layer computes a measure of similarity between the input pattern and each stored pattern. These measures of similarity are given by $K(\mathbf{x}^{(1)}, \mathbf{x}(t))$, $K(\mathbf{x}^{(2)}, \mathbf{x}(t))$, ..., $K(\mathbf{x}^{(M)}, \mathbf{x}(t))$. The third layer is the output layer, which consists of chaotic neurons. Furthermore, there is feedback from the output layer to the input layer in order to repeat the retrieval process. The matrix $\mathbf{W}$ of synaptic weights between the kernel function layer and the output layer is an $N \times M$ matrix whose elements are determined by $W_i^{(m)} = \alpha_i^{(m)} y_i^{(m)}$. The bias value of the output layer is denoted by $\mathbf{B} = [b_1, b_2, ..., b_N]$.

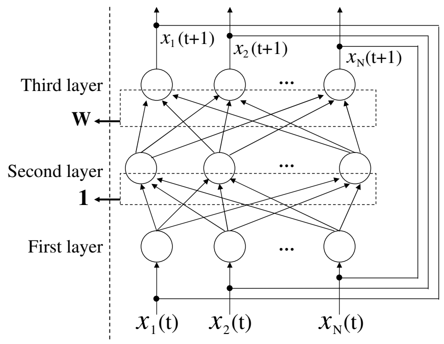

Figure 1: Dynamical associative memory using a nonlinear support vector machine.

5. Numerical experiments

To investigate retrieval characteristics of recall and transition frequencies of the proposed associative memory model, we prepare ten sets of four stored patterns ($M = 4$) as shown in Figure 2. Each stored pattern used in this study has a size of 100 pixels ($N = 100$). The stored patterns are random pattern whose number of firing is varying from 49 to 60. The black and white rectangles denote firing (+1) and resting (−1), respectively.

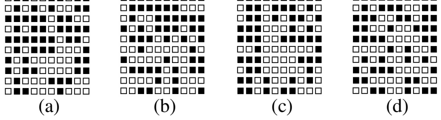

Figure 2: The stored patterns.

In addition, the patterns are generated so that they have certain amount of correlation between them. These correlations are as follows; Correlation coefficients between Pat.(a) and Pat.(b), between Pat.(a) and Pat.(c), and between Pat.(a) and Pat.(d) are set to 0.1, 0.2, and 0.3, respectively. Moreover, correlation coefficients of between Pat.(b) and Pat.(c), and between Pat.(a) and Pat.(c) are set to 0.3, and 0.2, respectively. Lastly, correlation coefficient of between Pat.(c) and Pat.(d) is set to 0.1.

At the beginning of numerical experiments, the synaptic weights and biases are determined by the nonlinear SVM from the stored patterns. Then, we carry out simulation of the dynamical associative memory for 150000 iterations. To avoid the influence of a transient phase, the number of recalls in the first 100000 iterations is not counted. The experiment is carried out for the dynamical associative memory for with in 50000 iterations with 100 initial input patterns and computed the average recall and transition frequencies of the stored patterns for these 100 trials. In addition, to count the recall and transition frequencies of the stored patterns, the output values of the chaotic neurons are binarized with a threshold of 0.5. The parameter values of the chaotic neurons are set to $k_f = 0.533$, $k_r = 0.826$, $\alpha = 0.54$, and $\epsilon = 1.566$. In addition, the standard deviation of Gaussian kernel function is set to $\sigma = 15$. The average number of recalls of each stored pattern set and the value of external input A are shown in Table 1. The external input A in Table 1 are chosen so that the numbers of recalls of all the stored patterns are almost the same among the sets of stored patterns and the dynamics of the proposed model shows the orbital instability. From Table 1, we see that the proposed model recalls all the stored patterns with almost the same frequencies for all stored pattern sets. Then, we show the average of frequencies of transitions among recalled patterns and transition intervals among them. The frequencies of the transition intervals are evaluated by the following standardization:

Table 1: The average number of recalls.

Pattern	A	Pat.(a)	Pat.(b)	Pat.(c)	Pat.(d)
set1	-0.011	903.5	944.2	898.3	846.4
set2	-0.001	887.4	890.5	891.1	900.8
set3	-0.427	962.3	838.9	1331.8	1267.5
set4	0.014	1384.7	1390.8	1438.9	1447.7
set5	-0.0055	901.7	910.7	886.9	887.6
set6	0.009	1403.2	1442.5	1402.6	1435.6
set7	-0.0035	1401.4	1433.5	1570.2	1600.4
set8	0.015	1435.0	1454.2	1405.5	1387.6
set9	0.018	768.4	780.8	758.0	722.1
set10	0.083	1482.9	1465.2	1407.7	1434.2

$$s = \frac{T - \bar{T}}{\sigma_T} \tag{14}$$

where s denotes standardized value of frequencies of the

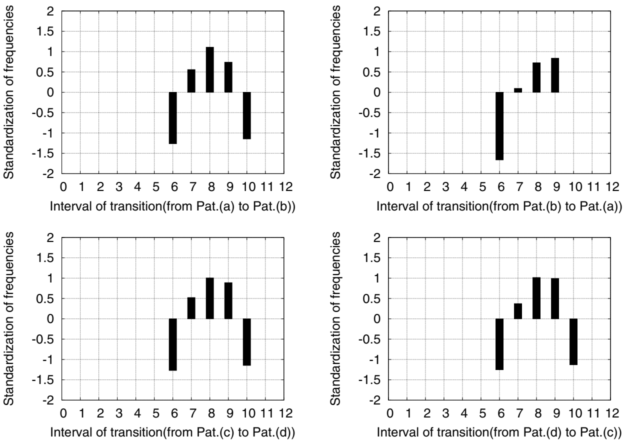

Figure 3: The standardized frequencies of the transitions interval(for correlation coefficient of 0.1).

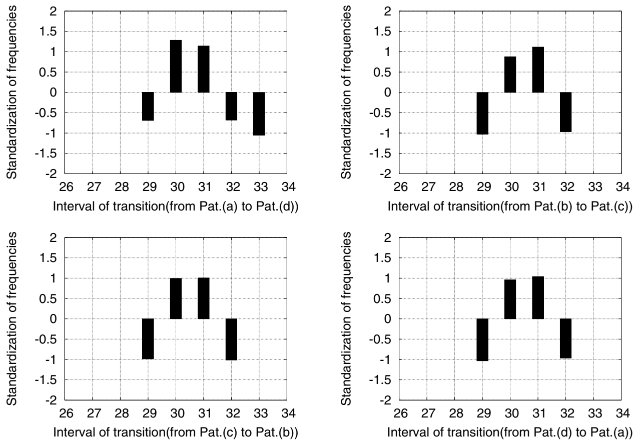

Figure 4: The standardized frequencies of the transitions interval(for correlation coefficient of 0.3).

transition intervals, T denotes frequencies of the transition intervals, $\bar{T}$ and σ_T are the average value and standard deviation of T, respectively. The average numbers of transitions among recalled patterns for the stored pattern set6 are shown in Table2.

Table 2: The average numbers of recall transitions among recalled patterns.

From \ To	Pat.(a)	Pat.(b)	Pat.(c)	Pat.(d)
Pat.(a)	4.11	224.16	229.62	127.07
Pat.(b)	219.32	6.24	134.4	235.28
Pat.(c)	227.3	125.64	5.25	224.2
Pat.(d)	134.2	239.22	213.07	5.47

There is dispersion in the transition frequencies between pairs of patterns, however we see that the transitions of every combination of stored patterns were occurred. In addition, The standardized frequencies of the transition intervals for the case of correlation coefficient of 0.1 and 0.3 among stored patterns are shown in Figs, respectively. Upper left panel of Figure 3 shows the standardized frequencies of the transition intervals from Pat.(a) to Pat.(b). Upper right, lower left, and lower right panels in Figure 3 show the same ones from Pat.(b) to Pat.(a), is the same one from Pat.(c) to Pat.(d), and from Pat.(d) to Pat.(c), respectively. Figure 4 is the same figure as in Figure 3 but with the correlation coefficient of 0.3. As in the case that the correlation coefficient is 0.1 between stored patterns, the transitions of all the stored occur in 6-10 iterations as shown in Figure 3. In particular, we find that the proposed model tends to transit the stored patterns in 8 iterations. As in the case that the correlation coefficient is 0.3 between stored patterns, the transitions of all the stored occur in 29-32 iterations as shown in Figure 4. As in the case of Figure 3, the proposed model tends to transit specific iterations in the case of Figure 4. It seems that the model has specific periods to be a period for recall transition even the model shows orbital instability. This periodicity of recall transition is also observed for other stored pattern sets.

6. Conclusions

We construct a dynamical associative network by determining the synaptic weights **W** and the biases **B** of associative chaotic neural networks with a nonlinear SVM. We investigate retrieval characteristics of recall and transition frequencies of the stored patterns with considering correlation between them. As for the proposed model, it became clear that recall transition shows specific periods even the network shows orbital instability. A future problem is to investigate retrieval characteristics for lager the size of network with changing the correlation coefficient between stored patterns.

References

[1] M. Adachi. K. Aihara. Associative dynamics in a chaotic neural network. *Neural Netw.*, vol. 10, no. 1, pp. 83-98, 1997.

[2] K. Aihara. T. Takabe. M. Toyoda. Chaotic Neural Networks. *Physics Letters A*, vol. 144, pp. 333–340, 1990.

[3] I.Tsuda. Dynamic link of memory–Chaotic memory map in nonequilibrium neural networks. *Neural networks*, 5,313–326

[4] Nello Cristianini. John Shawe-Taylor. An introduction to support vector machines. 2006.

[5] J.C.Platt. Fast training of support vector machines using sequential minimal optimization. *MIT Press*, Advances in kernel methods, pp.185–208, 1999.