数理モデル化と問題解決 31－7
(2000. 9. 21)

カオスニューロンを用いた階層型ニューラルネットワークによる重なったパターンの分離

福原 義久
慶應義塾大学 政策・メディア研究科
〒252-2580 神奈川県藤沢市遠藤 5322

武藤 佳恭
慶應義塾大学 環境情報学部
〒252-2580 神奈川県藤沢市遠藤 5322

概要

人工ニューラルネットワークを用いてパーセプトロンや連想メモリをはじめとするさまざまなパターン認識のための技法が提案されてきた。入力パターンをある一つのカテゴリーに分類することはこれらの技法で可能である。しかし複数のパターンが重なっている場合に、それらを的確に分離し認識するためには複雑な機構を必要としてきた。提案手法はカオスニューロンとパターン認識のための古典的な学習法であるバックプロパゲーション技法を組み合わせることにより、これを実現した。非常に扱いやすい単純なモデルにもかかわらず、カオスのダイナミクスを用いることにより、効果的に重なったパターンを分離できる。

Chaotic pattern segmentation of mixed figures

Yoshihisa Fukuhara
Graduate School of Media and Governance
Keio University
5322, Endo, Fujisawa, Kanagawa, 252-2580, JAPAN

Yoshiyasu Takefuji
Faculty of Environmental Information
Keio University
5322, Endo, Fujisawa, Kanagawa, 252-2580, JAPAN

Abstract

Many pattern segmentation systems have been developed by artificial neural networks. Those systems, such as Perceptron and Associative memory are very effective to map an input to one of the memorized patterns. However, if we try to extract plural patterns from a mixed figure at once, we have to embed a specialized mechanism in their system.

This paper proposes a new pattern segmentation system using chaotic dynamics. This system can discriminate several patterns at once, even if they are overlapped to each other. And the proposed system is also reconfigurable, because the system consists of two simple components, 'back-propagation' and 'chaotic neuron model'.

1. Introduction

The purpose of this research is to discriminate plural patterns from a mixed pattern by using back-propagation and chaotic dynamics. Many scientists have developed various kinds of pattern segmentation systems, for example, Perceptron[1] and Associative memory[2]. These methods work accurately only if the input data does not include noise. However, in the real world, information contains a lot of noises, where the input data is often incomplete and several patterns are overlapped. Especially, to discriminate the overlapped objects is an extremely difficult problem for information processing, because every pattern interferes each other. To solve this problem, we have developed a simple neural network by using chaotic dynamics. The proposed system consists of well-known supervised learning algorithm 'Back-propagation[3]' and a new artificial neuron model, called 'chaotic neuron model[4]'. By adding a chaotic dynamism into the back-propagation networks, the proposed system shows a dynamic behavior and solves more difficult pattern segmentation problems. Our system can extract mixed patterns more than four patterns simultaneously with the simple structure, and can be applied to practical information processing applications. The proposed system shows an advantage of the chaotic dynamism without instability from chaos. In the following section, we explain the mechanism of our system and state the advantage of our system based on the simulation results.

2. Method

The proposed system consists of two mechanisms, chaotic neuron model and back-propagation algorithm. In this section, we introduce the back-propagation algorithm and chaotic neuron model. Finally, we explain how to manipulate the system.

2.1 Back-propagation

Back-propagation is a supervised learning algorithm proposed by Rumelhart in 1986[3]. It enables the multi-layer neural network (known as Perceptron) to realize the segmentation of any nonlinear pattern. It gradually minimizes average error based on the difference between an output and a teaching signal. The average error between the output and the teaching signal is given by eq.(1). T is a teaching signal and O is a system output. The weight between jth neuron and ith neuron is updated by eq.(2). Note that α and η are constant values respectively.

$$E_p = \frac{1}{2}\sum_j (T_{pj} - O_{pj})^2 \qquad 1.$$

$$\Delta_p W_{ji}(n+1) = \eta\left(-\frac{\partial E_p}{\partial Wji}\right) + \alpha \Delta_p W_{ji}(n) \qquad 2.$$

2.2 Chaotic neural networks

K.Aihara proposed a chaotic neuron model [4] which imitated a chaotic behavior caused by the refractoriness of biological neuron. And they proposed chaotic neural networks model based on the chaotic neuron. The dynamics of chaotic neural networks are given by eq.(3) and eq.(4), where $x_i(t+1)$ is an output between 0 and 1 at time $t+1$, M is the number of given data, N is the number of neurons, $A_j(t)$ is jth input value, Vij is a weight from Aj to ith neuron, Wij is a weight from ith neuron to jth neuron, and α and θ are the refractory scaling parameter and the threshold. Ke, K_f, and K_r are the refractory decay parameters from external inputs, the feedback inputs, and the refractoriness, respectively. eq.(4) is a sigmoid function and ε defines slope of sigmoid curve.

$$x_i(t+1) = f\Big(\sum_{j=1}^{M} V_{ij} \sum_{d=0}^{t} K_e^d A_j(t-d) + \sum_{j=1}^{N} W_{ij} \sum_{d=0}^{t} K_f^d x_j(t-d) - \alpha \sum_{d=0}^{t} K_r^d x_i(t-d) - \theta i \qquad 3.$$

$$f(u) = 1/(1+\exp(-u/\varepsilon)) \qquad 4.$$

They also simplified eq.(3) into the following formulas. Although they used this network model for an associative memory, we applied this model in multi-layered neural network.

$$x_i(t+1) = f(\eta_i(t+1) + \zeta_i(t+1))$$
$$\eta_i(t+1) = k_f \eta_i(t) + \sum_{j=1}^{N} w_{ij} x_j(t) \qquad 5.$$
$$\zeta_i(t+1) = kr\zeta_i(t) - \alpha x_i(t) + ai$$

where the term ζ is an input value and the term η is a mutual interactions, and a_i is the sum of the threshold and temporally constant external inputs to the ith neuron. A chaotic neuron outputs an analog value. In the output layer, the output of neuron $x_i(t)$ has an alternative value based on eq.(6).

$$x_i(t) = \begin{cases} 1 & (x_i(t) \geq 0.5) \\ 0 & (x_i(t) < 0.5) \end{cases} \qquad 6.$$

2.3 Learning and recognition

At the first stage, we use the back-propagation algorithm to teach several patterns to the network. In this stage, we use the sigmoid neuron as described in eq.(4). We do not use chaotic neuron model at the learning stage, because we do not need instability of chaotic dynamism for learning. Chaotic dynamism is used when the system discriminate several patterns after learning.

In the next stage, we use chaotic neurons instead of sigmoid neurons for pattern segmentation in order to recognize complex mixed patterns (Figure-1). Note that the learning weights are kept as fixed values after learning.

Outputs of the system must be recorded as sequential values during a certain period for one input. During the calculation period, we must not initialize the internal states in the chaotic neurons. By keeping the internal states of neurons, the chaotic dynamism enables the system to discriminate mixed patterns.

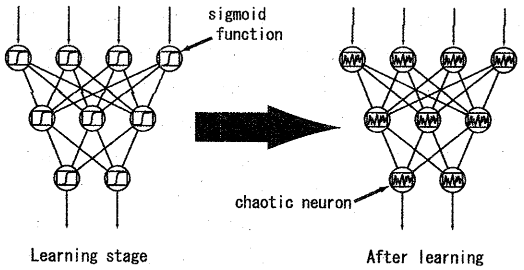


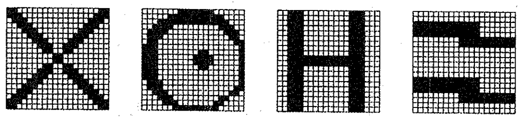
Figure 1: The exchange of elements

2.4 Structure of the network

The proposed system consists from a simple multi-layered neural network. In this paper, we use three layered neural networks, however you can use different numbers layers and neurons. To avoid interfering plural results each other, we should put a number of the neurons in the output layer and a number of memorized patterns in the same. And we should apply one memorized pattern to one output.

3. Simulations and results

To show the effectiveness of our system, we experimented three simulations on computer. In this simulation, the network has 400 neurons in the input-layer, 40 neurons in the hidden-layer and 4 neurons in the output-layer. And the system was trained to memorize four patterns as shown in Figure-2. We continued the learning until the average error $E < 0.02$. The parameters of the chaotic neurons are shown in Table-1. These parameters were obtained by genetic algorithm [5]. As a result, we recorded 100 sequential outputs for one static input.



PATTERN 1 PATTERN 2 PATTERN3 PATTERN4

Figure 2: Stored patterns

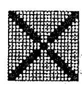
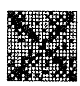
Table 1: The Parameters of chaotic neuron

$kf=0.202551$ $ai=0.797982$
$kr=0.327443$ $\varepsilon=0.015000$
$\alpha=2.136770$

3.1 Simulation-1: Input un-mixed pattern

When we gave pattern-1 to the network, the system shows a correct answer perfectly as shown in Table-2. Figure-3 shows the state of neurons in output layer from the time scale 0 to 30. Although we gave noisy input, we obtained a good result.

Table 2: Results of the simulation-1

Input	Persentages of correct answeres	Persentages of spikes
 PATTERN 1	PATTERN 1 = 100% PATTERN 2 = 0% PATTERN 3 = 0% PATTERN 4 = 0% Unknown = 0%	Output 1 = 100% Output 2 = 0% Output 3 = 0% Output 4 = 0%
 PATTERN 1 with 20% noise	PATTERN 1 = 100% PATTERN 2 = 0% PATTERN 3 = 0% PATTERN 4 = 0% Unknown = 0%	Output 1 = 100% Output 2 = 0% Output 3 = 0% Output 4 = 0%

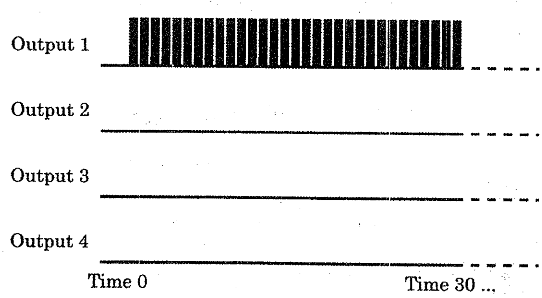
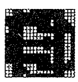


Figure 3: Spikes of output neurons (sim-1)

3.2 Simulation-2: Segmentation of 3 patterns

When we gave a mixed pattern of pattern-2, pattern-3, pattern-4, the proposed system did not show a good performance seemingly. However, you can observe that the proposed system recognize each of three patterns from four memorized patterns, when we observe the spikes of the neurons in the output layer as shown in Figure-4. In Figure-4, output neuron-3 and output neuron-4 spiked simultaneously. The proposed system shows an excellent answer from the view of rate of spikes as shown in Table-3. Finally, we gave a mixed pattern of the three patterns with noises, the proposed system showed accurate answers from the difficult input value as shown in Table-4 and Figure-5.

Table 3: Results of the simulation-2

Input	Persentages of correct answeres	Persentages of spikes
 PATTERN 2+3+4	PATTERN 1 = 0% PATTERN 2 = 32% PATTERN 3 = 0% PATTERN 4 = 0% Unknown = 68%	Output 1 = 0% Output 2 = 41% Output 3 = 50% Output 4 = 50%

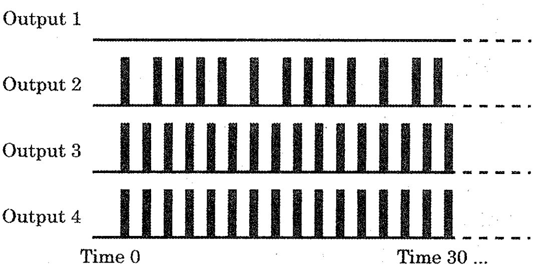
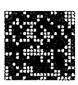


Figure 4: Spikes of output neurons (sim-2)

Table 4: Results of the simulation-2 (with noise)

Input	Persentages of correct answeres	Persentages of spikes
 PATTERN 2+3+4 with 20% noise	PATTERN 1 = 0% PATTERN 2 = 20% PATTERN 3 = 17% PATTERN 4 = 17% Unknown = 46%	Output 1 = 0% Output 2 = 28% Output 3 = 34% Output 4 = 30%

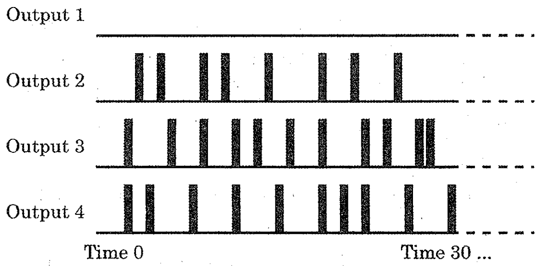


Figure 5: Spikes of output neurons (sim-2 with noise)

4. Conclusion

The proposed system can discriminate several patterns simultaneously from a mixed pattern. Generally, neural network systems using chaotic dynamism show an unstable behavior. However the proposed system shows a good performance by tuning the values of parameters to control chaotic dynamism. Since our system consists of 'back-propagation' method, we can simply handle and modify it. Our system shows a breakthrough of back-propagation by using chaotic components. We can apply this system for complex information processing such as video observation system and speech recognition, and text recognition.

References

[1] Minsky,M and Papert,S, "Perceptrons", *MIT Press*, 1969

[2] J.J.Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proc. Of National Academy of Science*, U.S.A., 79, 2445-2558, 1982

[3] Rumelhart,D.E.,Hinton,G.E.and Williams,R.J. Learning representaions by back-propagating errors. *Nature*,Vol.323-9, pp.533-536. 1986

[4] K.Aihara, T.Takabe and M.Toyoda, "Chaotic Neural Networks", *Phys. Lett. A*, 144, 6, 7, 333-340, 1990

[5] J.H. Holland, "Adaptation in Natural and Artificial Systems", *University of Michigan Press*, 1975