The Chaotic Self-Organizing Map

Alison A. Dingle[1] John H. Andreae[2] Richard D. Jones[1]

1. Department of Medical Physics and Bioengineering, Christchurch Hospital
2. Department of Electrical and Electronic Engineering, University of Canterbury
Christchurch, New Zealand.

Abstract

A chaotic self-organizing map can be produced by replacing the linear neural units of the conventional self-organizing map with neural units capable of producing chaos. The introduction of chaos into the self-organizing map is shown to improve the ability of the network to cluster input patterns.

1 Introduction

A fundamental limitation of neural networks is their rigid behaviour compared with even the simplest biological organisms [1]. By introducing chaos into neural networks the problem of their rigid behaviour can be overcome. In this paper, a chaotic neural unit based on a simple chaotic system is introduced into the self-organizing map, and is shown to improve the ability of the network to cluster input patterns.

2 Deterministic chaos

Deterministic chaos, or simply chaos, is the phenomenon whereby unpredictable, seemingly random behaviour is produced by a completely deterministic system. Such activity is sensitively dependent on initial conditions and, therefore, unpredictable — small differences in initial conditions give rise to huge differences in the outputs within a finite time.

One of the simplest systems capable of displaying deterministic chaos is the logistic equation :

$$y(n+1) = 4\,g\,y(n)\,[1 - y(n)] \qquad (1)$$

The behaviour of this first order difference equation changes dramatically as the bifurcation parameter g is altered (Fig. 1). When $g < 0.25$ the output or population dies away to zero. For $0.25 < g < 0.75$ the output converges to a single non-zero value. As g is increased beyond 0.75 the output begins to oscillate first between 2 values, then 4 values, then 8 values and so on, until for $g > 0.89$ the output becomes chaotic.

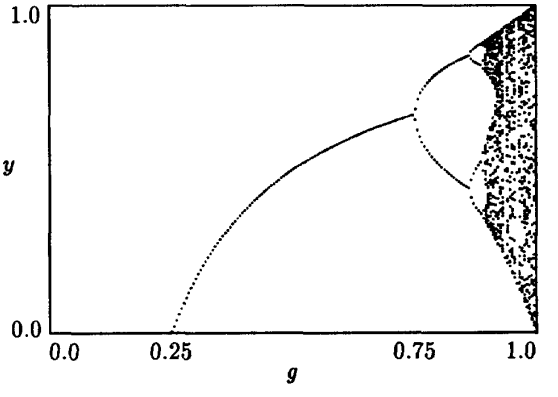


Figure 1: Bifurcation diagram for the logistic map.

3 The chaotic neural unit

The chaotic neural unit (fig. 2) employed in this study is based on the inverted logistic equation [2, 3]. The net input to the neural unit is given by :

$$net = \sum_i w_i\,x_i \qquad (2)$$

where x_i are the inputs to the neural unit and w_i are their associated weights. The next output $y(n+1)$ of the neural unit depends on the net input and the previous output as follows :

$$y(n+1) = 1 - 4\,[1 - net]\,y(n)\,[1 - y(n)] \qquad (3)$$

0-8186-4260-2/93 $03.00 © 1993 IEEE

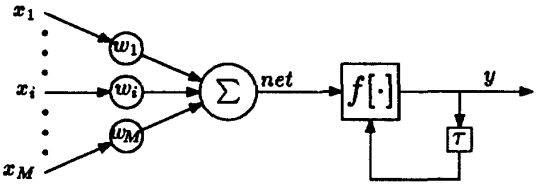


Figure 2: The chaotic neural unit with inputs x_i, weights w_i, output y, a unit time delay τ and $f[net, y] = 1 - 4[1 - net]\, y\, [1 - y]$.

The behaviour of this neural unit depends on its net input (Fig. 3). When $net < 0.11$ the neural unit displays chaotic activity, for $0.11 < net < 0.25$ it produces periodic activity, while for $net > 0.25$ it produces a single value which increases with increasing net to be unity for $net > 0.75$.

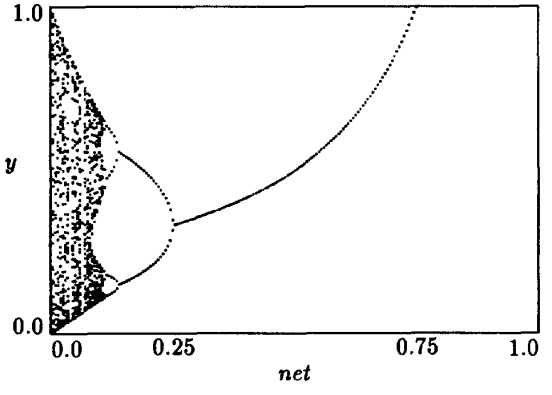


Figure 3: The inverted bifurcation diagram of the logistic equation displays the output y of the chaotic neural unit as a function of the net input net.

4 The chaotic self-organizing map

Chaotic neural units can be incorporated into the conventional self-organizing map to produce a chaotic neural network which is able to learn from experience. The conventional self-organizing map [4] is made up of linear neural units :

$$y_{pj} = net_{pj} = \sum_{i=1}^{M} w_{ji} x_{pi}$$

$$= \mathbf{w}_j \bullet \mathbf{x}_p = \|\mathbf{w}_j\| \, \|\mathbf{x}_p\| \cos \phi_{pj} \qquad (4)$$

where ϕ_{pj} is the angle between the input pattern $\mathbf{x}_p$ and the weight vector $\mathbf{w}_j$. Typically the vectors $\mathbf{x}_p$ and $\mathbf{w}_j$ are of unit length and, therefore, $y_{pj} = \cos \phi_{pj}$. When an input pattern $\mathbf{x}_p$ is presented to the network, neural units compete for the privilege to learn — the winning neural unit being the one with the largest output y_{pj}. Because $y_{pj} = \cos \phi_{pj}$, the winning neural unit is the one whose weight vector is closest to the input pattern. The weight vectors of the winning neural unit and its neighbours are modified according to the following rule, which rotates $\mathbf{w}_j$ towards the input pattern $\mathbf{x}_p$:

$$\mathbf{w}_j = \frac{\mathbf{w}_j^{old} + \eta \mathbf{x}_p}{\|\mathbf{w}_j^{old} + \eta \mathbf{x}_p\|} \qquad (5)$$

The learning rate η, typically, decreases linearly over the training session from an initial value $\eta(0)$ to a final value η_{min}.

Chaos can be introduced into the self-organizing map by replacing the linear neural units of the conventional network with chaotic neural units (3).

When an input pattern $\mathbf{x}_p$ is presented to the network, each neural unit produces a sequence of outputs $y_{pj}(n)$. As with the conventional self-organizing map, neural units compete for the privilege to learn but, in this case, the winning neural unit is chosen to be the one with the greatest average output $\overline{y_{pj}}$ calculated over T elements of the output sequence :

$$\overline{y_{pj}} = \frac{1}{T} \sum_{n=1}^{T} y_{pj}(n) \qquad (6)$$

In practice, T is relatively small and the average output $\overline{y_{pj}}$ depends on both the net input net_{pj} and the initial value $y_{pj}(0)$. Thus, the competition to learn effectively becomes probabilistic, with the neural unit with the largest net input (i.e., closest to the input pattern) usually having the greatest probability of winning. As with the conventional self-organizing map, the winning neural unit and its neighbours adjust their weight vectors according to (5).

The self-organizing map is renown for its ability to cluster input patterns, model their probability density function, and generate a spatial order. The following simulations illustrate the ability of the chaotic self-organizing map to perform each of these tasks.

4.1 Clustering input patterns

Consider a self-organizing map where only the neural unit that wins the competition to learn modifies its

weight vector. Although such a network cannot develop a spatial order (because all neural units learn independently), it is still able to cluster input patterns. During training the weight vector of each neural unit approaches the average of the cluster of input patterns for which it won the competition to learn (Fig. 4). Thus, each neural unit becomes sensitively selective to a cluster of input patterns.

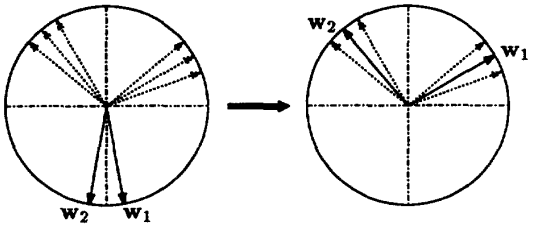


Figure 4: Optimal clustering of input patterns, weight vectors move to a cluster of inputs.

In order for a self-organizing map to cluster input patterns optimally, all neural units must participate in learning. When input patterns from two well-separated clusters are presented to a conventional self-organizing map consisting of only two neural units, the weight vectors typically move to the average of the nearest cluster of input patterns (Fig. 4). However, if the initial positions of the weight vectors are such that one weight vector is closer to all input patterns than the other, then, regardless of which input pattern is presented, the same neural unit wins the competition and its weight vector moves to the average of all input patterns while the other weight vector remains in its initial position (Fig. 5).

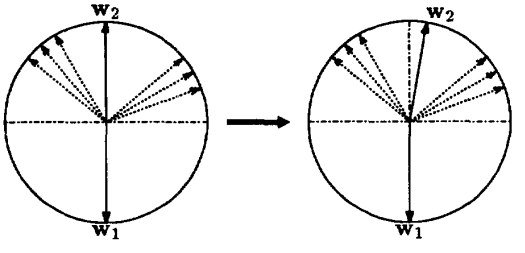


Figure 5: A clustering problem arises with the conventional self-organizing map when one weight vector is closer than the other to all input patterns; one weight vector moves to the average of all input patterns while the other remains in its initial position.

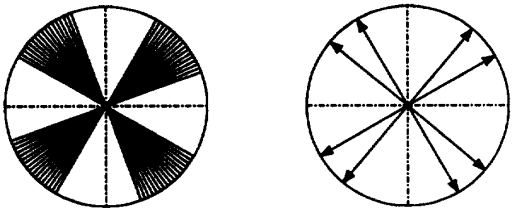
In the chaotic self-organizing map, the neural unit whose weight vector is closest to the input pattern does not necessarily win the competition to learn and, thus, the clustering problems encountered with the conventional self-organizing map should, at least to some extent, be overcome.

In order to test this hypothesis, input patterns were selected from two well-separated clusters and presented to a self-organizing map consisting of only two neural units. The initial positions of the weight vectors were pseudo-randomly selected from the unit circle and the network trained from 15 different initial states. As expected, the ability of the conventional self-organizing map to cluster input patterns depended on the initial positions of the weight vectors. In fact, the conventional self-organizing map was able to cluster input patterns correctly for only 8 of the 15 initial states, with one weight vector being closer to all input patterns than the other for the remaining 7 initial states. When the same input patterns were presented to a chaotic self-organizing map, input vectors were correctly clustered regardless of the initial positions of weight vectors and of the value of T. Because the neural unit whose weight vector was closest to the input pattern did not necessarily win, both neural units were, at some stage, able to win the competition and, thus, participate in learning.

4.2 Probability density functions

In order to examine the ability of the chaotic self-organizing map to model the probability density function of input patterns, input patterns were pseudo-randomly selected from the unit circle according to a given probability density function ρ and presented to a self-organizing map, where again only the winning neural unit modified its weight vector. The accuracy to which the network modelled ρ was measured by the root mean square (r.m.s.) error σ between the final positions of weight vectors and their ideal positions.

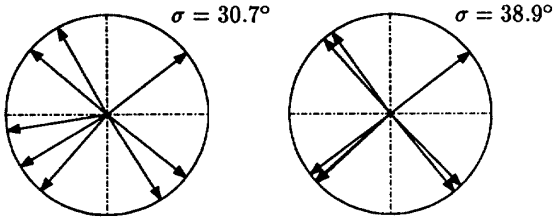
Input patterns were selected from four equiprobable clusters (Fig. 6a) and presented to a network consisting of eight neural units. Ideally, the weight vectors of the neural units would have moved to the positions indicated by Fig. 6b. However, during training, the weight vectors of the conventional self-organizing map tended to move from their initial positions towards the nearest cluster of input patterns, with some weight vectors remaining in their initial positions. Thus, the accuracy with which the network modelled ρ tended to be poor (e.g., Fig. 7a) and depended on the initial positions of the weight vectors. The performance of the chaotic self-organizing map also depended on the initial weight vectors but, in this case, all neu-



(a) (b)

Figure 6: (a) Four equiprobable clusters of input patterns and (b) the corresponding ideal positions of weight vectors.

ral units participated in learning. Although relatively poor models of $\rho(x)$ were produced, the chaotic self-organizing map again displayed its ability to cluster input patterns, with all neural units becoming sensitive to a single cluster of input vectors (e.g., Fig. 7b).



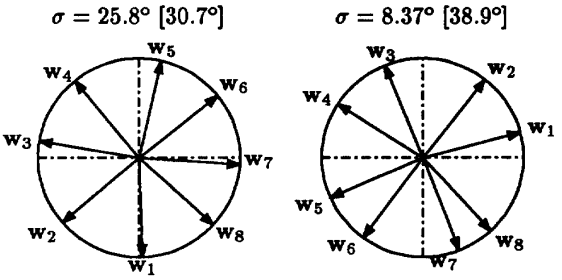
(a) CONVENTIONAL (b) CHAOTIC

Figure 7: Final positions of weight vectors (a) for the conventional network and (b) for the chaotic network (T=10).

4.3 Generating a Spatial Order

The formation of a spatial order relies on the interaction of neural units during learning, which is achieved by allowing neighbours of the winning neural unit to modify their weight vectors.

In the following simulation, eight neural units were arranged in a circle and two neural units either side of the winner were allowed to adjust their weight vectors. While the winning neural unit learned at a rate η, its immediate neighbours learned at a reduced rate $\eta_1 = \eta/4$ and their neighbours at a rate $\eta_2 = \eta/16$. Input patterns were again selected from four equiprobable clusters (Fig. 6a). The interaction of neural units not only resulted in a spatial order but also significantly improved the models of ρ produced, with the chaotic self-organizing map developing more accurate models than the conventional network (Fig. 8).



(a) CONVENTIONAL (b) CHAOTIC

Figure 8: Final positions of weight vectors (a) for the conventional network and (b) for the chaotic network with T=10. Brackets indicate r.m.s. error for the same network when only the winning neural unit modified its weight vector.

5 Conclusion

A chaotic neural unit based on a simple chaotic system, the logistic equation, can be used to introduce chaos into neural networks. In particular, chaotic neural units were incorporated into the self-organizing map to produce a chaotic neural network which can learn from experience. Chaos introduces a controlled randomness into the self-organizing map, which results in an improved ability to cluster input patterns.

References

[1] M. Zak, "Creative dynamics approach to neural intelligence," *Biological Cybernetics*, vol. 64, pp. 15–23, 1990.

[2] A. A. Dingle, J. H. Andreae, R. D. Jones, "A chaotic neural unit," *Proceedings of IEEE International Conference on Neural Networks*, San Francisco, USA, pp. 335–340, 1993.

[3] A. A. Dingle, "Engineering in Brain Research : Processing electroencephalograms, and chaos in neural networks," PhD thesis, Department of Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, 1992.

[4] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.