ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 401-406

Learning in a Chaotic Neural Network

Nigel Crook and Tjeerd olde Scheper

Neurocomputing Research Group
School of Computing and Mathematical Sciences
Oxford Brookes University

Abstract. Previous research has shown how the Unstable Periodic Orbits (UPOs) embedded in a chaotic attractor can be made to correspond to self-organised dynamic memory states in a chaotic neural network [2]. This paper demonstrates how this chaotic neural network model can be extended to enable it to adapt to dynamic input patterns using two unsupervised learning rules. The proposed learning rules are designed to modify model parameters in order to support the network's dynamics from which the memories emerge. This means that input weights and feedback delays are adapted so that the network will stabilise an appropriate UPO in response to each input signal.

1 Introduction

The research presented in this paper attempts to identify and model ways to store information in dynamic, chaotic neural networks. The justification for this research is given by both biological as well as theoretical motivations [1, 3, 2]. Firstly, there is substantial support for the use of nonlinear dynamics to study more complex and interesting behaviours of neuronal networks. Secondly, even though chaos may seem to be generally undesirable, it has important properties that may be exploited to store and retrieve information [5]. These include space filling, the possibility of control via delayed feedback, synchronisation and the sensitive dependence on initial conditions. In this paper we demonstrate how these unique properties may be exploited to store information in the dynamic behaviour of a neural network. Furthermore, we present a novel approach to neural network adaptation which is based on supporting the dynamics from which memory states emerge during pattern recognition.

2 Network Architecture

In this paper we present a neural implementation of the continuous delayed feedback method of chaos control [4]. This implementation is defined by a set of discrete-time equations which model a three layered network. The first layer

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 401-406

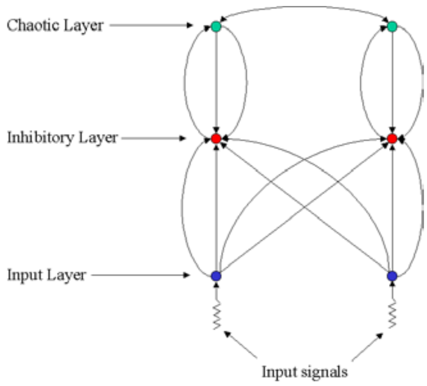

Figure 1: An overview of the network architecture

is composed of units which receive dynamic input signals from the *environment*. This input layer is fully connected to the second layer which is made up of inhibitory units. Each unit in the inhibitory layer is connected to one chaotic unit in the third layer. Units in the chaotic layer are connected to their immediate neighbours via lateral connections. A complete description of this network is given in [2]. An overview of the network architecture is shown in Figure 1.

The activation $y_i(t)$ of unit i on the chaotic layer is determined by the following equations:

$$g_i(t+1) = (1-\phi)(\omega y_i(t) - \alpha f(y_i(t)) + a) + \frac{\phi}{N_i}\sum_{j=1}^{N_i} y_j(t) \tag{1}$$

$$y_i(t) = g_i(t) + k_i(t)z_i(t) \tag{2}$$

The chaotic layer can be organised either linearly, so that each unit has at most two lateral connections with its immediate neighbours, or it can be organised as a rectangular lattice so that each unit has at most four lateral connections. The organisation of the chaotic layer becomes significant with the application of the learning rules given below which enable the network to develop *localised responses* so that units in the same region of the chaotic layer will respond to similar input patterns. The right most term of equation (1) sums up the input from these lateral connections, with N_i denoting the number of neighbours for unit i. The constant ϕ determines the strength of the lateral connections relative to the units own chaotic dynamics. The right most term of equation (2) introduces the control to be applied to this unit (see below). When this term is zero for a number of time steps, the dynamics of $y_i(t)$ are governed by a chaotic attractor.

Each chaotic unit is associated with one unit in the inhibitory layer. The

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 401-406

purpose of each inhibitory unit is to apply feedback control to stabilise the associated chaotic unit into an unstable periodic orbit (UPO). The activation $z_i(t)$ of inhibitory unit i is given by:

$$z_i(t) = \begin{cases} 0 & : \sum_{j=1}^{M} w_{ij}(t) I_j(t) = 0 \\ g_i(t) - \sum_{j=1}^{D} \xi_j y_i(t - j\langle\tau_i\rangle)) & : \sum_{j=1}^{M} w_{ij}(t) I_j(t) \neq 0 \end{cases} \quad (3)$$

where M is the number of input units, $w_{ij}(t)$ ($w_{ij}(t) > 0$) is the weight on the connections from input unit j to inhibitory unit i, $I_j(t)$ is the activation of the jth input unit, D is the number of delayed feedback connections from chaotic unit i to inhibitory unit i, τ_i is the characteristic delay for inhibitory unit i and $\langle\tau_i\rangle$ denotes the value of τ_i rounded to the nearest integer.

The characteristic delays for the inhibitory units are set to random real values at initialisation. When an input signal is presented to the network, the inhibitory units are enabled and can apply control to the chaotic units. The network will select the unit in the inhibitory layer whose characteristic delay best matches the dominant period of the input sequence. This selection of the *winning* unit is made by finding the unit which has the smallest value of $h_i(t)$:

$$h_i(t) = \sum_{j=1}^{M} w_{ij}(t)(I_j(t) - I_j(t - \langle\tau_i\rangle)) \quad (4)$$

The Characteristic delay of the winning unit is denoted by τ_{win}. All inhibitory units i whose values of $\langle\tau_i\rangle$ are equal to $\langle\tau_{win}\rangle$ can apply control to their chaotic units. This is achieved through the following equation:

$$k_i(t) = \begin{cases} 0 & : \langle\tau_i\rangle \neq \langle\tau_{win}\rangle \\ \gamma & : \langle\tau_i\rangle = \langle\tau_{win}\rangle \end{cases} \quad (5)$$

where γ ($\gamma < 0$) is the optimum value that k_i can have for effective control to be applied to the chaotic unit(see [4] for a discussion of optimal values for the feedback strength k_i).

3 Adaptation

The network architecture presented in Section 2 generates internal dynamic recognition states (UPOs) which are associated with the dynamics present in the input signals. In this approach, memories are not stored as distributed patterns of weights between units, as is commonly used with artificial neural networks. Instead, *memory states* emerge from the dynamics of the network.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 401-406

Consequently, conventional approaches to network adaptation, which are centred on weight adaptation, cannot be applied in this model. In this section we present a novel approach to adaptation which is based on modifying network parameters in order to support the dynamics from which the memory states emerge.

The profile of characteristic delays across the inhibitory units determines which UPOs will be stabilised when input is presented to the network. Initially these delays are randomly selected. Two competitive learning rules are introduced which enable the network to (i) adapt the weights on the connections from the input layer to the inhibitory layer so that input signals which are more commensurate with the characteristic delays of the inhibitory units can be given a stronger weighting, (ii) tune the characteristic delays to match the frequency profiles of the input signals, and (iii) develop a localised response on the inhibitory layer so that neighbouring units have similar characteristic delays. Both learning rules use the concept of a neighbourhood around the winning unit. This neighbourhood is delimited by a radius $\mathcal{R}$ and a maximum reach $\mathcal{M}(t)$. Learning is applied to all units which are within a distance $\mathcal{M}(t)$ from the winning neuron. The distance d_{ij} from unit j to unit i in the inhibitory layer is defined as the minimum number of connections required to connect them. The function $\rho(d_{ij})$ calculates the direction and magnitude of the changes to be made to the characteristic delay and input weights of an inhibitory unit i based on its distance d_{ij} from the winning unit j: $\rho(d_{ij}) = \frac{\eta(\mathcal{R}-d_{ij})}{\mathcal{R}*d_{ij}}$

The first learning rule, LR1, is concerned with adapting the weights on the connections from the input layer to the inhibitory layer. It enables an inhibitory unit to shift weights away from input signals which are not commensurate with its characteristic delay, and towards units which are commensurate. LR1 is expressed by the following equation (Note that for each inhibitory unit $\sum_{j=1}^{M} w_{ij} = M$ and $m_{ij}(t) = |I_j(t) - I_j(t - \langle\tau_i\rangle)|$):

$$w_{ij}(t+1) = \begin{cases} w_{ij}(t) & : \quad d_{ij} > \mathcal{M}(t) \\ w_{ij}(t) - \rho \left[\frac{w_{ij}(t)}{N} - 1 + \frac{w_{ij}(t) m_{ij}(t)}{\sum\limits_{p=1}^{N} w_{ip}(t) m_{ip}(t)} \right] & : \quad d_{ij} <= \mathcal{M}(t) \end{cases} \tag{6}$$

The second learning rule, LR2, is responsible for tuning the characteristic delays to match the frequency profiles of the input signals and developing a localised response on the inhibitory layer so that neighbouring units have similar characteristic delays. This is achieved at each iteration by identifying the period τ_{ij} of the strongest input j to the winning unit i. The characteristic delay of the winning unit is then modified to be closer to the value of τ_{ij}.

At each time step of the evolution of the system the winning inhibitory unit finds the period τ_{ij} of the input connection with the largest weight $w_{ij}(t)$. All units within the maximum reach of the learning rule then modify their characteristic delays according to the following equation, where i is the index

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 401-406

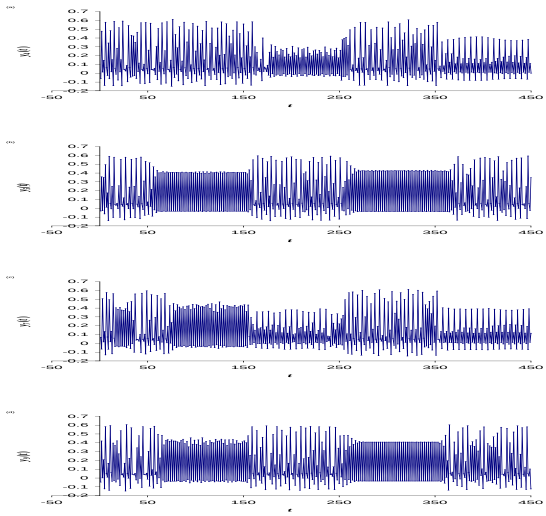

Figure 2: The activation time series of selected units

of the winning unit:

$$\tau_k(t+1) = \begin{cases} \tau_k(t) & : \quad d_{ij} > \mathcal{M}(t) \\ \tau_k(t) + \rho(d_{ij})(\tau_{ij}(t) - \tau_k(t)) & : \quad d_{ij} <= \mathcal{M}(t) \end{cases} \quad (7)$$

The following experiment demonstrates how learning rules LR1 and LR2 adapt a 16 unit rectangular layered network (further experimental result are given in [2]). The network was presented with two input sequences: the first was period 2 (i.e. input unit I_1 was presented with the sequence (0, 1, 0, 1, ...), and input unit I_2 with (1, 0, 1, 0, ...)), the other was period 3 (i.e. I_1 was presented with (0, 0.5, 1, 0, 0.5, 1, ...) and I_2 with (1, 0.5, 0, 1, 0.5, 0, ...)). In each case the input was started at $t = 51$, and consisted of alternating 100 of the period 2 iterations with 100 of the period 3 iterations. The activations of units 4, 5, 7 and 9 (shown from the top down) from the chaotic layer are plotted in Figure 2.

The final values of the characteristic delays for the 16 unit rectangular network are shown in Figure 3. This figure clearly shows that the inhibitory layer has been partitioned into units which have $\langle \tau_i \rangle = 2$ and units which have $\langle \tau_i \rangle = 3$. Figure 2 shows the activation of two units from each partition. Unit 4 and 7 have $\langle \tau_i \rangle = 3$, and so develop a response to period 3 input by stabilising a period 6 orbit. Units 5 and 9, on the other hand, have adapted their characteristic delays so that $\langle \tau_i \rangle = 2$, enabling them to respond to a period 2 input.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 401-406

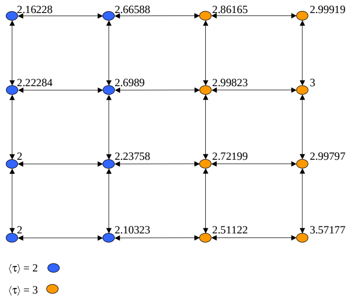

Figure 3: The final values of the characteristic delays

4 Conclusion

The results presented in this paper demonstrate the ability of a discrete neural network to respond to periodic input by stabilising into an unstable periodic orbit. Delayed feedback control is used to respond to the period of the input and the behaviour of the network is affected accordingly. The spatial organisation of the network makes it possible to allow subsets of neurons to stabilise into different orbits depending on the presented input. By dynamically modifying the characteristic delays of the inhibitory units, the system can be trained to stabilise appropriate orbits. In addition the input weights may be modified to stabilise the sensitivity of a subset of neurons for a particular input frequency.

References

[1] K. Aihara, T. Takabe and M. Toyoda, 1990, *Chaotic neural networks*, Physics Letters A, **144**(6,7):333-339.

[2] N.T. Crook, T.V.S.M. olde Scheper, 2002, Adaptation Based on Memory Dynamics in a Chaotic Neural Network, To appear in *Systems and Cybernetics*, vol. 33(4).

[3] W.J. Freeman and J.M. Barrie. Chaotic oscillations and the genesis of meaning in cerebral cortex. In: *Temporal Coding in the Brain*, eds. G. Buzsaki, R. Llinas, W. Singer, A. Berthoz, and Y. Christen. Berlin: Springer-Verlag, 1994.pp. 13-37.

[4] K. Pyragas, Continuous control of chaos by self-controlling feedback, *Physics Letters* A, vol. 170, 1992.

[5] S. Sinha and W.L. Ditto, 1999, *Computing with distributed chaos*, Physical Review E, **60**(1):363-377.