ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 333-338

A Self-Organising Neural Network For Modelling Cortical Development

Mike Spratling   Gillian Hayes
Department of Artificial Intelligence
University of Edinburgh
mikes@dai.ed.ac.uk   gmh@dai.ed.ac.uk

Abstract.
This paper presents a novel self-organising neural network. It has been developed for use as a simplified model of cortical development. Unlike many other models of topological map formation all synaptic weights start at zero strength (so that synaptogenesis might be modelled). In addition, the algorithm works with the same format of encoding for both inputs to and outputs from the network (so that the transfer and recoding of information between cortical regions might be modelled).

1. Introduction

Topological map formation is a ubiquitous feature of cortical development and many algorithms have been proposed as models (see [7] for a review of models of the visual cortex). These models generally represent the cortex as a single sheet of neurons each of which receives afferent excitation from a receptive field (RF). The RFs are refined by a learning algorithm to generate a topologically organised representation of the input space. Nearly all these models are based on the same underlying principles [7]:

- Patterned afferent activity.
- Pseudo-Hebbian synaptic modification.
- Normalisation of the total synaptic strength of each neuron.
- Fixed lateral connections between neurons in the cortical sheet which are locally excitatory, and inhibitory at greater distances.

It is obvious that the form of lateral connections used between neurons in the cortical sheet will encourage nearby neurons to be active for similar input patterns, while competition between more distant neurons will result in them coming to represent dissimilar inputs. In this way such a network can come to form a topologically organised map. The algorithm presented here (table 1) uses similar lateral interaction to form topological maps in the same way[1], however, it differs in: Not using weight normalisation; Allowing both afferent and lateral synapses to grow from an initial strength of zero, and; Using a consistent encoding scheme for both inputs and outputs.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 333-338

1. Initialisation:

   $q_{ij} = 0 \; \forall j, i$ where q_{ij} is the synaptic weight from input i to node j,
   $A_j = 0 \; \forall j$ where A_j is the amplitude of lateral inhibition from node j,
   $\bar{x}_i^t = 0 \; \forall i$ where $\bar{x}_i^t$ is the time trace of past activity for input i,
   $\bar{y}_j^t = 0 \; \forall j$ where $\bar{y}_j^t$ is the time trace of past activity for node j,

2. For each input, i, get the new value x_i, and update the time trace,
   $$\bar{x}_i^t = \tau_x * x_i + (1 - \tau_x) * \bar{x}_i^t$$

3. For each node, j calculate the activation due to the current input, y_j^{in}, and the activation modified by habituation, y_j^{bid},
   $$y_j^{in} = b + \sum_i q_{ij} x_i$$
   $$y_j^{bid} = y_j^{in} \left(1 + \mu \left(\frac{I}{N} - P_j\right)\right)$$
   where: b is the bias,
   I is the total number of iterations,
   P_j is the number of iterations for which node j was the winning node,
   N is the number of nodes in the network.

4. Choose the winning node, win, such that;
   $$y_{win}^{bid} + noise_{win} > y_j^{bid} + noise_j, \; \forall j \neq win$$
   where: $noise_j = rand * \bar{y}_{ave}^{bid}$,
   $rand$ is a random number between $-\nu$ and $+\nu$,
   $\bar{y}_{ave}^{bid}$ is the mean of $y_j^{bid} \; \forall j$.

5. For each node, j, calculate the output activation, after lateral inhibition, by applying a Gaussian inhibition function centred about the winner's location, $\vec{z}_{win}$, in the neural array,
   $$y_j^{out} = y_j^{in} + A_{win} y_{win}^{in} \left(e^{\frac{(\vec{z}_{win} - \vec{z}_j)^2}{-2\sigma^2}} - 1 \right)$$
   and update the time trace,
   $$\bar{y}_j^t = \tau_y * y_j^{out} + (1 - \tau_y) * \bar{y}_j^t$$

6. Modify synaptic weights,
   $$q_{ij} = q_{ij} + \beta \left(1 - y_j^{out}\right) \frac{lr_{post}\left(\bar{y}_j^t, y_j^{out}\right)}{\hat{lr}_{post}} \frac{lr_{pre}\left(\bar{x}_i^t, x_i\right)}{\sum_k abs\left(lr_{pre}\left(\bar{x}_k^t, x_k\right)\right)}$$
   $$A_j = A_j + \alpha \frac{lr_{post}\left(\bar{y}_j^t, y_j^{out}\right)}{\hat{lr}_{post}}$$
   where: $lr_{post}(t, a) = \left(\frac{a}{t} - 1\right)_+$ is the post-synaptic learning rate,
   $lr_{pre}(t, a) = (a - t)$ is the pre-synaptic learning rate,
   $\hat{lr}_{post}$ is the maximum value of $lr_{post}\left(\bar{y}_j^t, y_j^{out}\right) \; \forall j$.

7. Continue from 2.

Table 1: The details of the proposed learning algorithm.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 333-338

2. Avoiding Normalisation

Normalisation provides competition between nodes, in addition to that provided by the lateral connections, since it strengthens connections to one part of the input space while weakening connections to another [7]. Because the algorithm presented here was designed to model synaptic weights starting from zero strength it does not use normalisation. Instead the learning rule provides activity-dependent decreases as well as increases in synaptic strength (step 6), which proves to be sufficient to refine the receptive fields. The learning rule is a variant of the covariance rule in which time traces of the input and output activities have been used as the thresholds. It is scaled to limit the synaptic changes that can occur in any one learning step, but this does not provide weight normalisation since weight changes for each node will differ.

Normalisation also helps to prevent any single node coming to represent a disproportionally large region of the input space. In this algorithm 'habituation' is used for this purpose. Nodes compete to represent each input (step 3), however, nodes which have won the competition too often have their ability to compete reduced. Modifying competitiveness as a function of the number of times a node has won has been used previously to produce equiprobable clustering of the input space [2, 1]. However, in order to model habituation it is necessary to use a function (step 3) in which the modification made to the competitiveness of the node is independent of the total number of iterations that have taken place.

The selection of the winning node is also modified by noise (step 4). Noise helps the formation of a well ordered map, since it enables re-organisation before nodes become committed to representing particular inputs. As the nodes develop different RFs noise has less effect.

3. Initial Conditions

Both afferent and lateral connections have an initial strength of zero (step 1). Since there is always a winning node which adjusts its receptive field to become a better representation of the input, and because habituation ensures that nodes are uniformly active, the entire input space becomes represented, even though all nodes initially have identical synapses.[2] Many other models give

[1] Figure 1(a) shows a very simple map with each node in the network projected onto the input space at the position of the preferred values of its RF. Figure 1(b) shows a map for a slightly more complicated input distribution. Because RFs are developed purely through activity dependent modification (rather than through the use of a 'neighbourhood' function [4, 3] which forces neighbouring nodes to have similar RFs) the network can successfully represent distinct input distributions. Figure 1(c) shows the preferred stimulus orientation for nodes trained with lines at 9 orientations. All results in this paper have been generated with identical parameter values.

[2] Figure 2 shows the formation of a simple map over time. The map becomes larger as RFs form and are differentiated by increased competition from the lateral inhibition. The resulting network shows the same topology as in figure 1(a) mapped with a quarter as many nodes, to demonstrate the robustness of the algorithm to changes in network size.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 333-338

random initial values to the afferent synapses to ensure coverage of the input space, and to enable normalisation to be used. For a real cortical map it would be inefficient for the innate mechanisms which create the rough connectivity of the cortex, prior to sensory invoked neural activity, to form random connections of arbitrary weight, only for these connections to be substantially modified by the subsequent input activity. The output of such a map would also be incorrect until considerable re-organisation had taken place. There, thus, seems to be neither biological nor computational justification for using random initial weights, and so we do not consider the inability of our model to 'untangle' a randomly initialised map as a major disadvantage. On the contrary, the growth of synapses from an initial strength of zero allows synapse formation to be modelled and makes the output representation inaccurate, rather than wrong, during learning.

4. Coding Requirements

It is known that many cortical maps make use of population coding [5], in contrast to the winner-takes-all encoding used by many models. Population codes are representations which are distributed over the activity of a population of neurons each of which respond over a range of inputs and have overlapping receptive fields [5]. The output of one cortical region will form (part of) the input to other regions. In order for a second region to form topologically organised representations, similar events in the first region must be encoded in a similar way. Population coding satisfies this requirement since similar events are represented by overlapping populations of active nodes. It would thus seem that the output activations of a model cortical region should form a population coded representation and that the region should receive inputs as population codes. The algorithm presented here has been designed to do this. Weak local inhibition means that nodes in the neighbourhood of the winner remain active (step 5). The output of the network is, thus, the activity of a population of nodes centred around the winner.[3]Such a common format for input and output encoding allows the output from one neural network to directly form the input to another to model the transfer [5, 6] and recoding of information between regions.

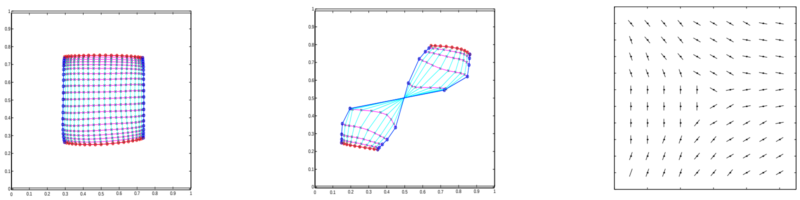
5. Conclusions

Some desirable features for a model of the organisation of the cerebral cortex include the ability to:

1. Form topologically organised representations.
2. Model synapse formation.

[3]Figure 3 shows (a) typical RFs, and (b) typical output activations for nodes in a network. It can be seen that both are population codes.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 333-338



(a) (b) (c)

Figure 1: **Simple mappings.** (a) Trained with data uniformly distributed over the unit square of the plane (after 4000 iterations). (b) Trained with data uniformly distributed within the top-right and bottom left quadrants of the unit square of the plane (after 10000 iterations). (c) Orientation preference when trained with the data used by von der Malsburg [7] (after 4000 iterations).

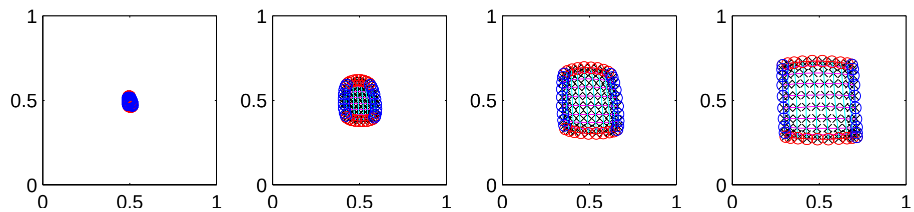
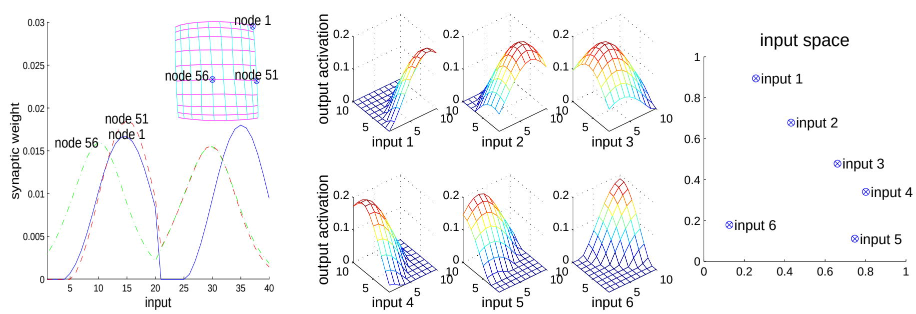


Figure 2: **Map development**. The maps are trained with data uniformly distributed over the unit square of the plane and are shown (from left to right) at 1000, 2000, 3000, and 4000 iterations.



(a) (b)

Figure 3: **Examples of population encoding of input and output signals.** (a) Typical receptive fields of nodes in figure 2 after 4000 iterations. The synaptic weights are shown for 3 nodes whose positions are shown on the inset. The 1st 20 inputs ($x_1 - x_{20}$) are supplied with a population coded representation of the x-axis coordinate value, while the 2nd 20 inputs ($x_{21} - x_{40}$) represent the y-axis coordinate value. (b) Typical output activations (y^{out}) for nodes in the same map. These activations are generated in response to the inputs shown on the right.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 333-338

3. Process population coded inputs and generate population coded outputs.

The algorithm described here has all of these properties. Other models of topological map formation are much more efficient (*e.g.* the Kohonen algorithm [4] will form the simple map shown in figure 1(a) with less than one-tenth of the training data), but generally only have the the first property.

References

[1] S. C. Ahalt and J. E. Fowler. 'Vector Quantization Using Artificial Neural Network Models'. In D. Docampo and A. R. Figueras, editors, *Proceedings of the International Workshop on Adaptive Methods and Emergent Techniques for Signal Processing and Communications*, pages 42–61, 1993.

[2] D. Desieno. 'Adding a Conscience to Competitive Learning'. In *International Conference on Neural Networks, I*, pages 117–124. IEEE Press, 1988.

[3] G. J. Goodhill. 'Topography and Ocular Dominance: A Model Exploring Positive Correlations'. *Biological Cybernetics*, 69:109–118, 1993.

[4] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. 'SOM_PAK: The Self-Organizing Map Program Package'. Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996.

[5] E. Salinas and L. F. Abbott. 'Transfer of Coded Information from Sensory to Motor Networks'. *Journal of Neuroscience*, 15:6461–6474, 1995.

[6] M. W. Spratling and G. M. Hayes. 'Learning Sensory-Motor Cortical Mappings Without Training'. In *6th European Symposium on Artificial Neural Networks*, submitted.

[7] N. V. Swindale. 'The Development of Topography in the Visual Cortex: A Review of Models'. *Network: Computation in Neural Systems*, 7(2):161–247, 1996.