5 Summary and Conclusions

This paper has introduced Competitive Hebbian Learning 2, and presented a couple simple demonstrations of its ability as an algorithm for unsupervised learning in artificial systems. The other claim made for CHL 2, that it represents a simplified model for learning in biological systems, rests upon the nature of the algorithm and the characteristics of the systems for which it is valid:

1. It is a monopolar system, with no negative node outputs required.
2. Learning occurs on positive weights only.
3. It is a local learning rule.
4. No arbitrary, ad-hoc rules are imposed.
5. It is a feedforward system unless time-sequences of inputs are significant.
6. It may be extended with trainable lateral connections to learn from time-sequences of inputs.

References

Bienenstock, E. L., Cooper, L. N., & Munro, P. W. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, **2**, 32 - 48, reproduced in: Anderson, J. A. and Rosenfeld, E. (eds.), *Neurocomputing: Foundations of Research* (chap. 26, pp. 439-455). Cambridge, MA: The MIT Press.

Hebb, D. O. (1949). *The organization of behavior.* New York: J. Wiley, reproduced (in part) in: Anderson, J. A. and Rosenfeld, E. (eds.), *Neurocomputing: Foundations of Research* (chap. 4, pp. 45-56). Cambridge, MA: The MIT Press.

Leen, T. K. (1991). Dynamics of learning in linear feature-discovery networks. *Network*, **2**, 85 - 105.

Lenz, R. & Österberg, M. (1992). Computing the Karhunen-Loeve expansion with a parallel, unsupervised filter system. *Neural Computation*, **4**, 382 - 392.

Linsker, R. (1988). Self-organization in a perceptual network. *Computer,* **21**, 105 -117, reproduced in: Anderson, J. A., Pellionisz, A., & Rosenfeld, E. (eds.), *Neurocomputing 2: Directions for Research* (chap. 31, pp. 528 - 540). Cambridge, MA: The MIT Press.

Linsker, R. (1992). Local synaptic learning rules suffice to maximize mutual information in a neural network. *Neural Computation*, **4**, 691 - 702.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology,* **15**, 267 - 273.

White, R. H. (1992a). Competitive Hebbian learning: Algorithm and Demonstrations. *Neural Networks*, **5**, 261 - 275.

White, R. H. (1992b). Recurrent Competitive Hebbian Learning, in *Proceedings of the International Joint Conference on Neural Networks, Baltimore*, (IV:767 - IV:772). Piscataway, NJ: IEEE Press.

w_{ij}								
	Input j							
Node i	1	2	3	4	5	6	7	8
1	1.20	0.00	0.00	1.20	0.00	0.00	1.20	1.20
2	0.00	1.20	1.20	0.00	1.20	1.20	0.00	0.00

Table 1: Weights learned by a pair of nodes. This pair learned a left-shift set of weights.

Qualitatively similar results were obtained for a system of 4 nodes. In this case the symmetry of the system lead to each node learning to respond strongly toward a different corner of the input space. Effective training occurred with $\lambda = 0.5$ for the first 15000 training cycles, with 10000 subsequent training cycles at $\lambda = 1.0$ increasing the separation of the node responses.

Except for the two-node system, these results appear quite similar to the results found with CHL in White (1992a). One must remember that these are non-negative systems, and the training is therefor different, as are the weights which are learned. For two nodes CHL 2 produces a pair of nodes which very precisely divide the input space between them, and which are analogous to the results for three- and four-node systems. CHL, White (1992a), produced a pair of nodes which took advantage of the bipolar nature of the system, producing a pair with very large overlap which were also very precisely orthogonal, with canceling positive and negative contributions to the product-of-node-outputs sum. There are computational advantages to bipolar systems, and when these advantages are important, bipolar systems such as CHL will have an advantage over monopolar systems such as CHL 2.

4 Stereo Disparity

For a second demonstration of CHL 2, we consider a simplified stereo-disparity task. Each input vector has eight components. The first four components, treated as if the first and last components were contiguous, have a contiguous pair of components ON and the other pair OFF. The second four components are set to a one-component shift of the first four, either a shift to the left or a shift to the right, the shift also being chosen randomly. So there are in fact eight distinct possible input vectors in the set, the four right-shift cases, 0011-1001, 0110-0011, 1100-0110, and 1001-1100; plus the four left-shift cases, 0011-0110, 0110-1100, 1100-1001, and 1001-0011. This test demonstrates the ability of CHL 2 to learn interesting responses to a set of non-orthogonal input vectors.

Ten sets of two CHL 2 nodes were trained, starting with small random weights, on random sequences of these input vectors. The best results were obtained with $\lambda = 1.11$ in equation 6. Using a learning-rate parameter, α, of 0.15, each pair of nodes learned to respond *with precision* to either a complementary pair of right-shift or to a complementary pair of left-shift inputs after an average of 2500 training cycles. A typical result is shown in Table 1. In each case the learned weights were qualitatively similar and had been learned to two decimal places precision by between 1300 and 6200 learning cycles in the ten test cases. Further training caused the weights to converge ever more precisely to the values indicated in the table. In six of the cases, the pair of nodes learned a pair of left shifts, and in the remaining four cases a pair of right shifts.

Somewhat analogous responses were learned by systems of four CHL 2 nodes. Using a λ of 0.40, five cases learned a set of four left-shift nodes, one case learned a set of four right-shift nodes, four cases learned a complementary pair of left-shift plus a complementary pair of right-shift nodes, and one set learned a duplicated pair of left-shift nodes.

previously. In this case, however, the competitive aspect clearly appears as a threshold. As such it is an alternative to the non-linear threshold for Hebbian learning introduced by Bienenstock, Cooper, and Munro (1982).

One should note the basic characteristics of the CHL 2 learning rule, equation 6. It is a modified Hebbian learning rule and as such it shares the optimizing characteristics of these rules. It is a very simple rule, and shares many of the desirable characteristics of Competitive Hebbian Learning, or CHL. There are no arbitrary, ad-hoc rules imposed, as in common competitive systems which utilize a winner-take-all or similar scheme. In addition, the competitive threshold can be implemented in a local system if the threshold is produced by non-learning inhibitory connections from the competing nodes. Furthermore, the system described is a purely feedforward system in its signal path. In addition to CHL, the systems proposed by Lenz and Österberg (1992) and by Linsker (1992) share some of the important characteristics of CHL 2, though the learning rules proposed by Lenz and Österberg and by Linsker are distinctly more complex.

3 Random Spots

For the first demonstration of CHL 2, take the problem of the system learning to respond to randomly placed, Gaussian-shaped spots. The system and training was very similar to that used in the CHL paper, White (1992a). The system consists of one to four nodes with a common set of 100 inputs. The inputs were treated as forming a 10-by-10 square array. Each input presentation, or input vector, was a Gaussian spot centered at a random position within the input array, but at least two input units away from the nearest edge of the input array. The strength of each component of an input vector was determined by how far that input unit was from the center of a spot. The Gaussian spot for these demonstrations was given by $0.5e^{-1.0(r-r_o)^2}$, where r_o is the center location, and neighboring components of an input vector are unit distance apart. The brightness of the spots was constant, and the background brightness was zero. (This contrasts with the CHL paper; there the average brightness of each input vector was zero in the bipolar system used there.) Weights were initialized to small random positive values. The training set consisted of a relatively large number of distinct input vectors, typically 5000.

The idea of this test is that the system should learn to respond to the structure present in the set of input vectors. With random Gaussian spots the only structure is the two-dimensional structure of the input array, along with the weakness of the training vectors near the edges of the array. A single Hebbian node should learn just this, an essentially symmetric response which is strong in the center of the input array, and weak around the edges. This is illustrated in Figure 1, which shows the weights learned by a single Hebbian node after 15000 training cycles, with little change after 5000 training cycles.

In contrast to the situation for a single Hebbian node, multiple CHL 2 nodes must also learn to avoid each other effectively. For this simple test, that means that they must share the input space and develop distinct regions of strong response. Weights learned by systems of two and three nodes are shown in Figures 2 and 3, respectively.

For two nodes the training was done using a value of 1.0 for λ, the CHL 2 separation parameter in equation 6. The weights shown in Figure 2 are for 25000 cycles of training, though again there were only small changes in weights after 5000 training cycles. The overlap of the pair of nodes is small; the average product of responses of the pair of nodes to a single input component is less than 5 % of the average squared node-response.

Results for a system of three nodes are shown in Figure 3. In this case the system was trained for 10000 cycles with $\lambda = 0.67$, and then trained for an additional 10000 cycles with $\lambda = 1.0$. If training was started with $\lambda = 1.0$, then nodes did not start to learn effectively; the competitive threshold was too high. Only after the nodes had learned to respond strongly to separate regions of the input could λ be increased to 1.0. For the three nodes shown the average product-of-response of any two of the nodes to a single input component was again less than 5 % of the average squared node-response. Note that the square symmetry of the input array forced an unsymmetric response for the three-node system.

of the node activities,

$$F = \sum_{i=1}^{N} y_i^2. \tag{1}$$

In this expression the y_i are the outputs of each node i of a set of N nodes. The y_i are squashed functions of a common set of I inputs,

$$y_i = f(\sum_{j=1}^{I} w_{ij} x_j). \tag{2}$$

Each w_{ij} is the connection strength, or weight, connecting input x_j to node y_i. For biological plausibility the squashing function f should be a sigmoid, though for single-layer model studies a linear function may also be used for y.

For a single node a desirable property for unsupervised learning is to maximize F; for a system of multiple nodes, what is desired is that something like F be maximized subject to constraints that the individual y_i respond as distinctly as possible. As in White (1992a), such constraints may be expressed as $g_{ik} = (y_i y_k)^S = 0$, where i and k are any distinct members of the set of nodes, and S is some constant.

Form a constrained system response function by subtracting from F the complete set of constraints g_{ik}, with each g_{ik} multiplied by a Lagrange multiplier λ_{ik},

$$G = F - \sum_{i=1}^{N} \sum_{k=i+1}^{N} \lambda_{ik} g_{ik}. \tag{3}$$

Symmetry suggests that all the λ_{ik} should be equal; we substitute the single parameter 2λ below. The constraints are subtracted, rather than added, in order that λ turn out positive in the final algorithm. In contrast to the situation in the CHL paper, since all node outputs are to be non-negative, there is no need for an even power of S in the definition of the constraints. The simplest form for G now is with $S = 1$, and with this choice for S the expression for G may be written

$$G = \sum_{i=1}^{N} y_i^2 - 2\lambda \sum_{i=1}^{N} \sum_{k=i+1}^{N} y_i y_k = \sum_{i=1}^{N} y_i^2 - \lambda \sum_{i=1}^{N} \sum_{\substack{k=1 \\ k \neq i}}^{N} y_i y_k. \tag{4}$$

The next step is to find the partial derivatives of G with respect to each w_{ij},

$$\frac{\partial G}{\partial w_{ij}} = f' \times [2 x_j y_i - 2\lambda x_j \sum_{k \neq i} y_k], \tag{5}$$

where f' is the derivative of the squashing function, and λ is a still undetermined parameter.

As in the CHL paper, we optimize the constrained output function G by requiring that the change in each weight, Δw_{ij} be made to be proportional to the corresponding partial derivative of G, first divided by f', which remains the same over a specific node. If we introduce a learning-rate parameter α this gives us:

$$\Delta w_{ij} = \alpha x_j (y_i - \lambda \sum_{k \neq i} y_k). \tag{6}$$

A final consideration for the learning algorithm is that weights should not be allowed to grow indefinitely. As in the CHL papers, we constrain the sum of weights into each node to an explicit limit, either setting the sum of weights or else the square of the sum of weights to a constant value. In numerical simulations the w_{ij} are also limited to non-negative values.

This Competitive Hebbian Learning 2 algorithm is a modified Hebbian-learning rule, but there is a competitive aspect to the learning rule, very like the Competitive Hebbian learning rule, CHL, introduced

Poster to be presented at WCNN '93, Portland, July 12, 1993.

Competitive Hebbian Learning 2: an Introduction

Ray H. White
Departments of Physics and Computer Science
University of San Diego
5998 Alcalá Park
San Diego, California 92110

Abstract

In this paper the Competitive Hebbian Learning 2, or CHL 2, learning rule is introduced. CHL 2 is an unsupervised learning rule with characteristics which are plausible as a simplified model of biological learning, as well as useful learning properties for artificial systems. It combines a modified Hebbian learning rule with a competitive learning, and shows promise for unsupervised-learning tasks such as feature detection.

1 Introduction

There is a continuing interest in unsupervised-learning algorithms due to a pair of distinct questions, the question of effective feature extraction, and the question of effective learning in biological systems. In general, the learning algorithms developed may be viewed as modified Hebbian learning algorithms, Hebb (1949), and these algorithms share to some extent the optimal-response characteristic of Hebbian learning, as shown, for example, in Oja (1982) and Linsker (1988).

In earlier papers, White (1992a) and White (1992b), a learning rule based upon Hebbian learning, Competitive Hebbian Learning (CHL), was introduced and demonstrated to be effective in extracting useful features in both feedforward and recurrent systems. (A very similar learning rule was considered by Leen (1991) also.) However, the nodes in CHL are inherently bipolar; that is, output values may be both positive and negative. This bipolar characteristic was fundamental to the development of the CHL learning rule.

The purpose of this paper is to introduce and to present some preliminary demonstrations of Competitive Hebbian Learning 2 (CHL 2), a modified Hebbian learning algorithm designed for systems of nodes with monopolar, non-negative outputs. Such monopolar nodes more closely conform to the characteristics of biological neurons, which may viewed from a simplified point of view as simply producing an output firing rate which is inherently non-negative. A consequence is that a simpler learning rule is allowed, CHL 2, which still retains the competitive character of CHL.

The next section develops the CHL 2 learning algorithm, and then subsequent sections present simulations which show something of the applicability of CHL 2 for a couple typical artificial neural network demonstration problems.

2 The Competitive Hebbian Learning 2 Algorithm

The purpose of this section is to develop a modified Hebbian-learning algorithm which is especially suitable for a system of monopolar nodes which have non-negative outputs, and which effectively compete with each other, producing distinct, useful responses. The reason for requiring that the outputs be positive (or zero) is to model something like a simplified biological neuron. The output of a model node may be taken to represent the firing rate of a biological neuron.

As in the paper by Oja (1982), as well as the paper which introduced Competitive Hebbian Learning, White (1992a), the starting point is an overall system response function, F, defined as the sum of squares

Figure 1

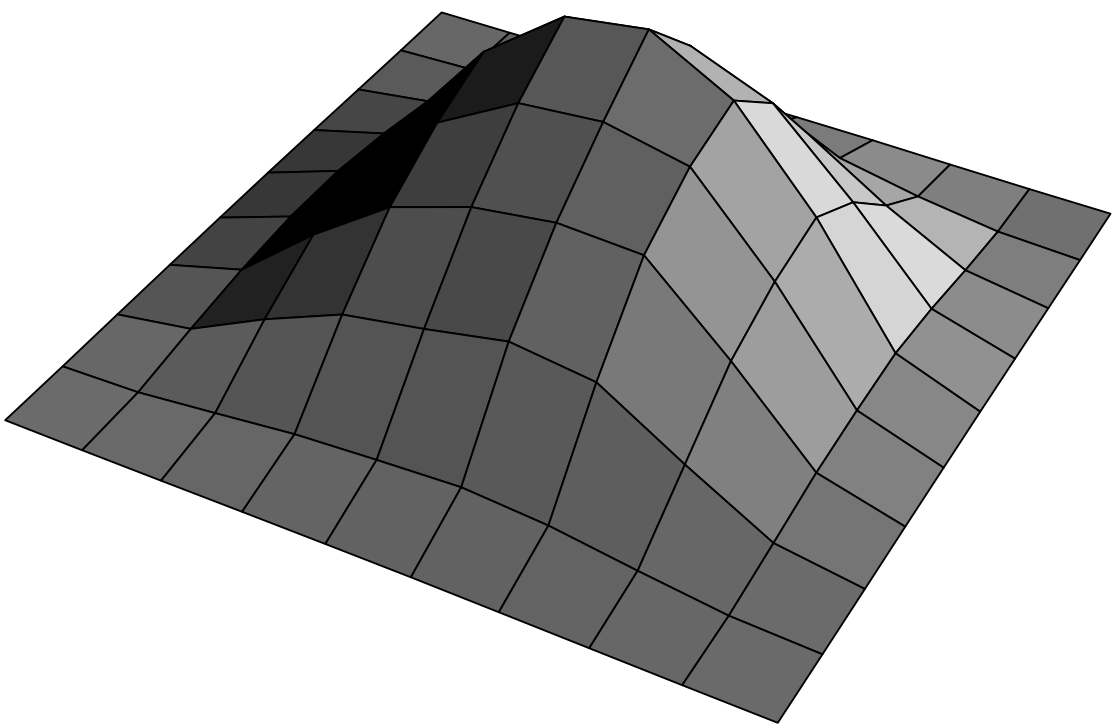

Figure 2

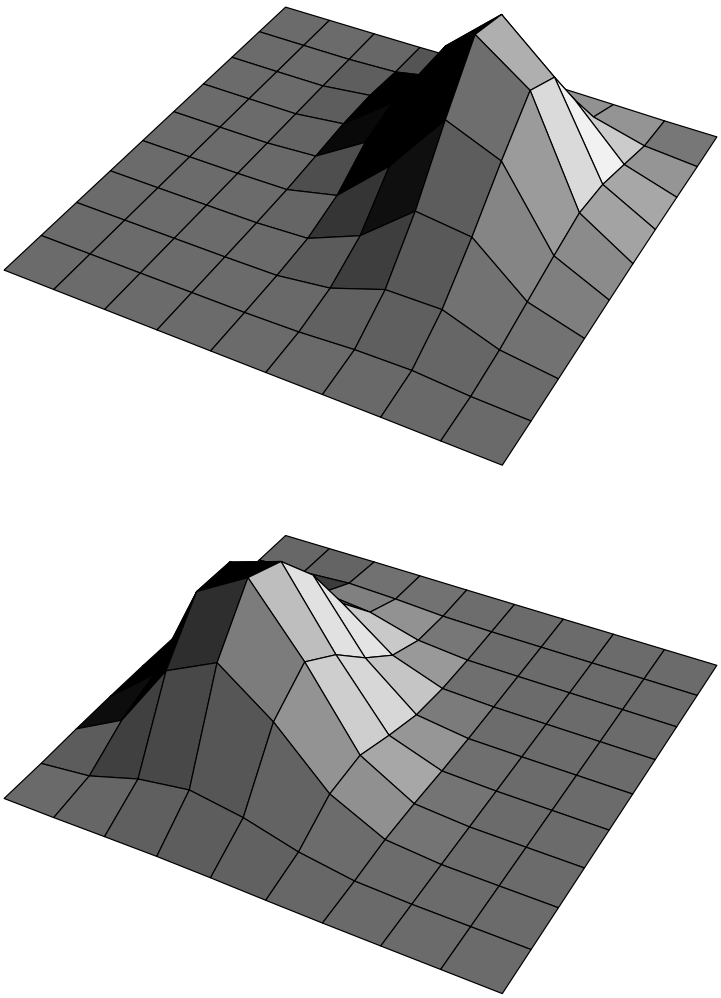
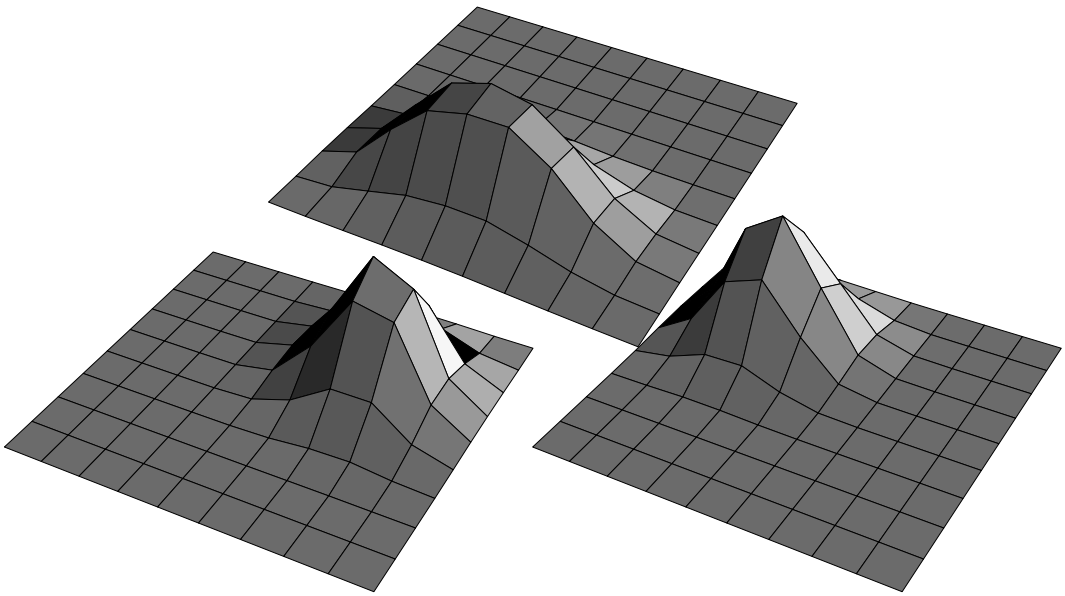
Figure 3