NEURAL ARCHITECTURE IN 3-D

The idea behind 3-D neural architecture is to create a pattern recognition system using neural components. The brain is taken as a model, and although little is known about how pattern recognition is accomplished there, much more is known about the cells that constitute the lowest levels, which process and analyze the features of an environment most directly. By constructing cells with similar properties to the biological cells, we may gain an advantage in information conservation and proper utilization of neural architectures.

The most important characteristic of brain cells is their receptive field (RF). With this in mind, we can search for an adaptive mechanism that, by changing connective strengths, can give the desired RFs. Then, since we will know what information the algorithmic components are providing, when a method is found that provides the desired cell types, we may be able to trace back via the algorithm to see what information the neurons give. In this article, a new neural network architecture is presented that encompasses functions similar to those in a biological brain, such as lateral and feedback connections of neurons. The neurons are randomly distributed on 2-D–planes. Each neuron on each plane can connect to a neighborhood of neurons at the next layer (plane), as well as receive feedback from neurons on that layer, or any other layer in the immediate or distant vicinity. In addition, lateral inhibitory connectivity within a layer adds to the flexibility and generalization abilities of the neural network.

ARCHITECTURE

The architecture developed is that of a hierarchy of 2-D cell layers, each successive layer farther removed from the environment (Fig. 1). The first layer receives inputs from the external world, and each successive layer from the preceding layers. In addition, the cells may receive lateral connections from neighboring cells within the same layer, depending on the particular choice of the architecture. The interlayer feedforward connections are chosen so that a cell feeds its connections onto a neighborhood of cells in the lower layer. This neighborhood may have definite bounds, so that all cells within it make connections, or it may have indefinite bounds, so that the probability of a connection decreases as a Gaussian with distance.
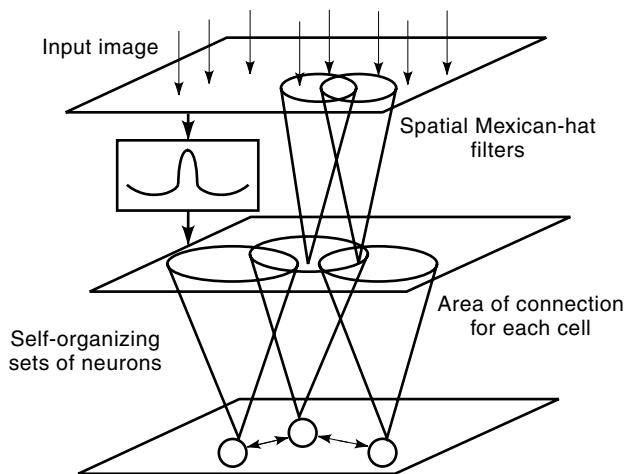
Figure 1. A schematic representation of the neural architecture.

The component cells themselves choose their outputs according to a weighted sum of all inputs passed through a function σ, such as

$$O_i(t) = \sigma \left(\alpha_i \sum_j C_{ij} O_j(t-1) \right) \qquad (1)$$

where $O_i(t)$ is the output of neuron i at time interval t; C_{ij} are the connection strengths, bounded in $[-\beta, \beta]$, where β is usually 1.0; and α is a constant. In simulations, σ is usually a sigmoid of the form

$$\sigma(x) = 0.5a[1 + \tanh(bxc)] \qquad (2)$$

where a, b, c, are constants that fix the maximum value, steepness, and bias of the sigmoid respectively. However, if we wish to allow the inhibitory components of the RF to be used by subsequent layers, then the sigmoid function must have a nonzero firing level for those negative inputs. This suggests the use of spontaneous firing activity for all neurons. An additional requirement for keeping the neurons useful and responsive is to keep them from being pushed too far into the saturation level. Since each neuron receives several inputs, that can easily occur. If it does, input deviations will not be sensed well, if at all. To prevent saturation from happening, α is usually chosen equal to the reciprocal of the number of connections to neuron i, so that the neuron simply passes a weighted average of the inputs through the sigmoid.

SIMULATIONS

A simulation usually consists of a sequence of presentations of random input patterns to the first layer, and a learning rule imposed on the connections by analysis of the firings of the neurons. A random input is chosen so as to prevent the cells from being biased towards any specific environmental feature. Since neighboring inputs are uncorrelated, first-layer cells that receive their influences are expected to have synapse patterns that will similarly wander aimlessly in the learning process. The first layer provides a spatial average of the overlying inputs. Since neighboring cells have the great-

est overlap in their neighborhoods, they tend to have firing patterns which are most similar. This will cause cells in layer 2, which have synapses originating from nearby cells, to want to be alike.

The actual training of the connections can be done in different ways:

1. Synapses can be changed according to a variation of Hebb's rule (1) as follows:

$$C_{ij} = \delta O_i O_j \qquad (3)$$

where δ is a small positive constant. Due to the correlation between neighboring level 1 cells, the synapses to the cells in subsequent layers will tend to want to be all alike without additional constraints. In order to guarantee both positive and negative synapses to every cell, an additional *resource* constraint is imposed, which takes the form

$$\sum_j C_{ij} = 0 \qquad (4)$$

The third restriction is a bounding of the connections to the interval $[-1, 1]$. A synapse is allowed the freedom to switch from positive to negative and vice versa. This is not expected to alter the main results, but only to prevent many of the synapses from disappearing with zero strength. Convergence usually occurs within 1000 to 5000 iterations, and faster convergence can be achieved with larger δ. Usually the final state of the synapses is at either the excitatory or the inhibitory limit. A network was created with three layers and 128 cells per layer. A square stimulus was assumed with 32×32 pixels size. The maximum distance that these cells can affect is r, with minimum weight -1 and maximum weight $+1$. The network had a total of 7071 connections. In the training mode, the minimum stimulus value was assumed to be zero and the maximum equal to 10. No noise was imposed on the system. The results obtained show the emergence of cells with edge-type RFs in layer 2 as is the case in the biological visual cortex [Fig. 2(a)]. The orientation of the edge appears to be totally arbitrary, even between neighboring cells. In layer 3, these edge cell RFs often conflict to form RFs that have oblong centers and surrounds of the opposite polarity, but many times these centers draw to the edges with further "learning." Thus the final RFs often look like an elliptical center touching the outside of the field, mostly surrounded by a horseshoe-shaped region of opposite polarity [Fig. 2(b)]. Figure 3 shows the results from a similar network, except that the minimum weight value is -0.5, that is, there is less inhibitory effect. Notice that excitation spreads more and that the maximum amplitudes are much larger. Also notice that the layer 3 RF is much longer than the one in layer 2. In the frequency domain, these RFs show more fine tuning as we move to deeper layers of the system. Figures 4 and 5 represent the power spectra of Figs. 2 and 3 respectively. Also notice that the edge effects are more obvious in the spectra of layer 3, again as is the case in the visual cortex.
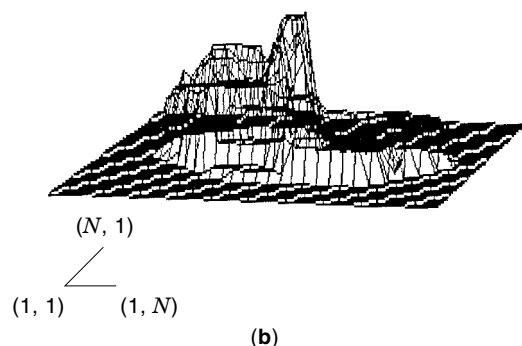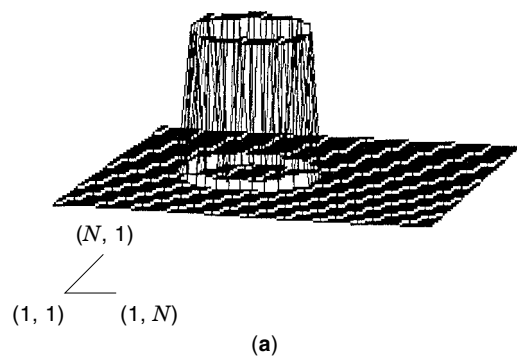
(N, 1)

(1, 1)   (1, N)

(a)

(N, 1)

(1, 1)   (1, N)

(b)

Figure 2. Receptive field characteristics for the neurons described in the text. (a) RF of layer 2. (b) RF of layer 3. Notice the center–surround organization of layer 2 and the elongated character of layer 3.

(N, 1)

(1, 1)   (1, N)

(a)

(N, 1)

(1, 1)   (1, N)

(b)

Figure 3. Receptive field organization for (a) layer 2 and (b) layer 3 when the inhibitory effects are less than in Fig. 2. Compare the amplitudes and the spread of the RFs with those of Fig. 2.

2. The wider the variance of the firing rate of the cells, the more information the cells can carry. With such a supposition, we can use an optimization routine to find the values of the synapses to a cell such that the variance in the firing rate of the cell is maximized. The optimization system is a variation of the ALOPEX (algorithmic logic of pattern extraction) process (2). In this process two random connection patterns are presented, and the variance V of the cell output is estimated with a number of random input patterns. Since we want the pattern of connection strengths to affect the variance and not the strength of the connections themselves, the variance is modified as

$$V_i = \frac{(1/N)\sum_j (O_i^j - O_i^{\mathrm{avg}})}{\sum_j C_{ij}} \qquad (5)$$

The connections are then changed according to the relation between the last change in connections and the last change in the variance, with an added noise term to prevent local minima, as follows:

$$C_{ij} = \beta[C_{ij}(t) - C_{ij}(t-1)][V_i(t) - V_i(t-1)] + \text{noise term} \qquad (6)$$

Remarkably, with this modification, the same edge-sensitive cell RFs emerge after only about 100 iterations and remain the same until about 400 iterations. This shows that the combination of Hebb's rule and ALOPEX
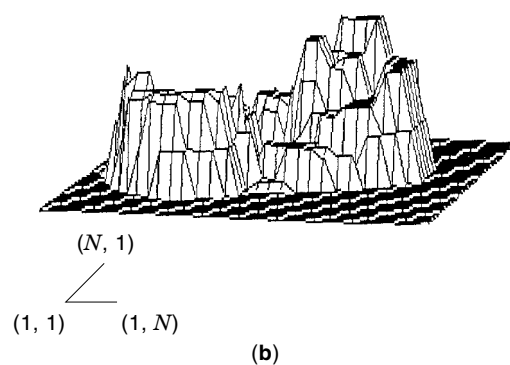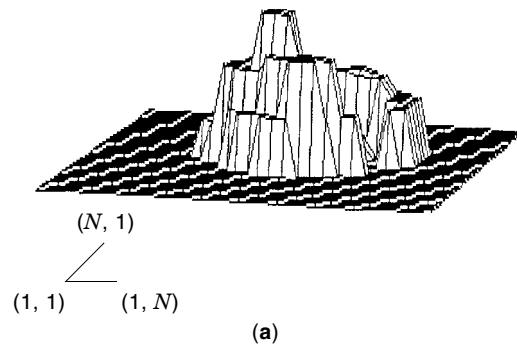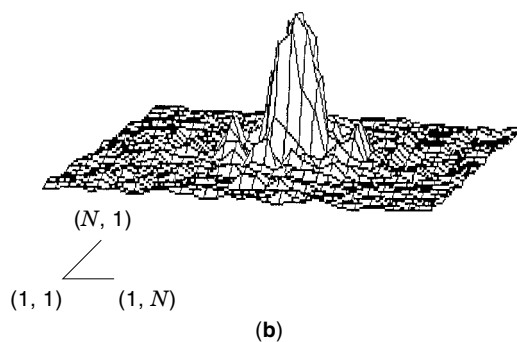

(N, 1)

(1, 1)   (1, N)

(a)

(N, 1)

(1, 1)   (1, N)

(b)

Figure 4. Power spectrum of the RF in Fig. 2: (a) layer 2, (b) layer 3. Notice the fine tuning in layer 3.

(N, 1)

(1, 1)   (1, N)
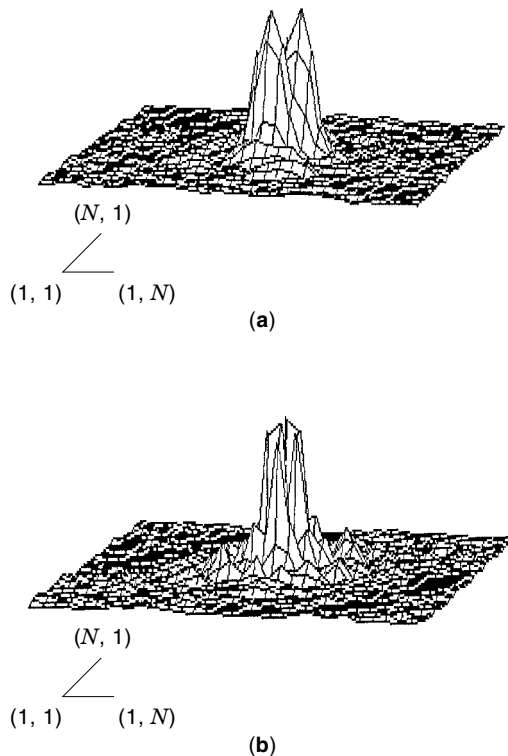
(a)

(N, 1)

(1, 1)   (1, N)

(b)

Figure 5. Power spectra of RFs in Fig. 3: (a) layer 2, (b) layer 3. Compare with Fig. 4. The edge effect is much more pronounced.

is something desirable. It might also mean that the way in which the architecture of the network is set up biases it towards neurons with edge detection capabilities. Work by others (3) has indicated that certain forms of Hebb's rule can be used to perform principal-component analysis, a a variance maximization of sorts.

3. In addition, both feedforward and feedback connections can be used, with feedback having a wider connective neighborhood than the feedforward connections. All connections are variable. If the inhibitory connections are spread over a much wider area, they tend to cancel the excitatory influence, making the Hebb changes ineffective. In future work we will include feedforward connections of cells with a Gaussian distribution, and with inhibitory connections and excitatory connections having a different spatial standard deviation. The present number of maximum synapses allowed does not give us the ability to obtain statistical significance for initial random strength generation.

4. Lateral connections on each layer are allowed and used, thus adding an extra feature of similarity to the biological system. If each input signal value is thought of as a dimension in parameter space, any particular input will constitute a point in that space. The synapses of a neuron can then be thought of as describing a vector in the same space, and the output of the neuron as the projection of the input point onto the synapse vector. If the choice of the synapses is initially random, chances are that the projections of many different inputs will lie close to one another, giving the neuron a response profile. Consider this to be the response profile of a neuron

before optimization. In order to better distinguish between inputs, the synapses should be changed so that more of the neuron range can be utilized. An intriguing choice is for the neuron to perform a type of *principal-component analysis* (Karhunen–Loeve feature extraction). Principal-component analysis may be approximated by a search for the vector (described by the connection weight values) that maximizes the variance of the cell firing level. The choice of this property may serve to partition the input space into recognizable categories at the output. This analysis approximates the Karhunen–Loeve search for the eigenvector with the maximum eigenvalue. For layers of neurons that have a large amount of information in common with near neighbors, the use of low-level lateral inhibition should prevent the system from settling on the same vector for each neuron, providing instead a graded topography to the layer.

Depending on the partitioning of the input space, this processing mode of neurons could provide many different behaviors. If the input space has clusters, the neuron may provide classification. If, on the other hand, the inputs are randomly distributed in space, the neuron can choose any feature vector, but might be constrained by near-neighbor interactions as to how it forms topographic maps.

DISCUSSION

In the neural network architecture presented, certain assumptions are made and various constraints are imposed, so that it resembled as much as possible the biological equivalents of feature detectors and edge detectors. A consequence is that the neural network can "learn" from stimuli alone, without a set of templates to compare the stimuli with. In particular, the neural network can implement unsupervised training with a variation of Hebb's learning rule (2). The connection strengths (weights) among the neurons of this network thus become the means of storing memories of the presented stimuli, so that the same stimulus, if reapplied, will bring the same output to the neural network. These outputs can become the templates for a new neural network—in a different region or even the same region implementing a different function. In recollection, external stimuli must be correlated with memories already stored as templates. In the case of using another neural network for this purpose, the ALOPEX training algorithm (1) can be applied with supervision in the form of previously stored memories.

The storage–recollection process is a dynamic one, and these networks need be coordinated well in order that new "experiences" can affect both networks in a proper fashion. Damage within a network will affect storage or recognition or both.

BIBLIOGRAPHY

1. E. Micheli-Tzanakou et al., Comparison of neural network algorithms for face recognition, *Simulation,* **64** (1): 15–27, 1995.

2. D. Hebb, *The Organization of Behavior: A Neurophysiological Theory,* New York: Wiley, 1949.

3. E. Oja, A simplified neuron model as a principal component analyzer, *J. Math. Biol.,* **15**: 267–273, 1982.

Reading List

P. J. Kruk and A. Wrobel, Spatio temporal organization of the receptive fields of retinal ganglion cell in the cat: A phenomenological model, *ACTA Neurobiol. Exp.,* **46**: 153–169, 1986.

E. Micheli-Tzanakou, When a feature detector becomes a feature generator, *IEEE Eng. Med. Biol. Mag.,* special issue on neural networks, **9** (3): 19–22, 1990.

E. Tzanakou, R. Michalak, and E. Harth, The ALOPEX process: Visual receptive fields by response feedback, *Biol. Cybernetics,* **35**: 161–174, 1979.

J. G. Daugman, Two dimensional spectral analysis of cortical receptive field profiles, *Vision Res.*, **20**: 847–856, 1980.

J. P. Jones and L. A. Palmer, The two dimensional spatial structure of simple receptive fields in cat striate cortex, *J. Neurophysiol.,* **58** (6): 1187–1211, 1987.

S. Marcella, Mathematical description of the responses of simple cortical cells, *J. Opt. Soc. Amer.,* **70** (11): 1297–1300, 1980.

I. Marsic and E. Micheli-Tzanakou, Distributed optimization with the ALOPEX process, *Proc. 12th Conf. IEEE Eng. Med. Biol. Soc.,* **12**: 1415–1416, 1990.

T. Poggio, V. Torre, and C. Koch, Computational vision and regularization theory, *Nature,* **317** (6035): 314–319, 1985.

D. A. Pollen and S. F. Ronner, Phase relationships between adjacent simple cells in the visual cortex, *Science,* **212**: 1409–1411, 1981.

E. Micheli-Tzanakou, Non-linear characteristics in the frog's visual system, *Biol. Cybern.,* **51**: 53–63, 1984.

E. Micheli-Tzanakou, Methods and designs: Visual receptive fields and clustering. *Behav. Res. Meth. Instrum.,* **15** (6): 553–560, 1983.

D. Zahner and E. Micheli-Tzanakou, Artificial neural networks: Definitions, methods and applications, in J. Bronzino (ed.), *The Biomedical Engineering Handbook,* Boca Raton: CRC Press, Chapter 184, pp. 2689–2705, 1995.

Evangelia Micheli-Tzanakou
Rutgers University

Timothy J. Dasey
MIT Lincoln Laboratories