SELF-ORGANIZING FEATURE MAPS

Neurons are the basic building blocks of the nervous system, and they successfully communicate information and perform rather complex pattern processing and recognition. Neural processes are characterized by intensive connections, inherent parallelism, self adaptation, and organization.

The growing scientific field of artificial neural networks uses mathematical modeling and computer simulation to achieve robust learning and pattern information processing analogous to the nervous system by interconnecting simple yet nonlinear computational elements.

Several potential application areas have been considered recently, ranging from control systems to speech synthesis and image processing. The preliminary results are promising and have opened up new research alternatives to the conventional computer programming paradigm and even artificial intelligence and expert systems.

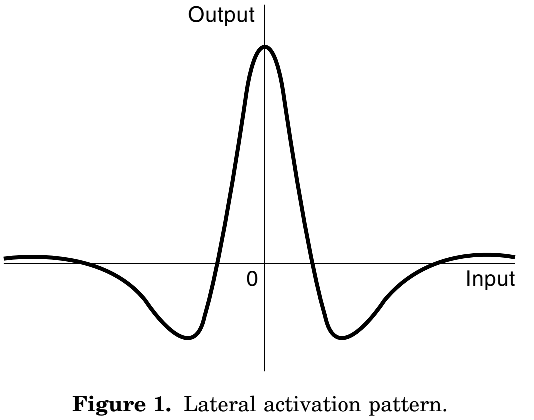

Figure 1. Lateral activation pattern.

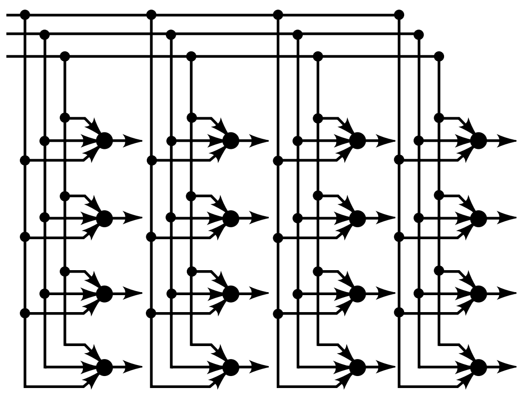

Figure 2. Output neuron lattice.

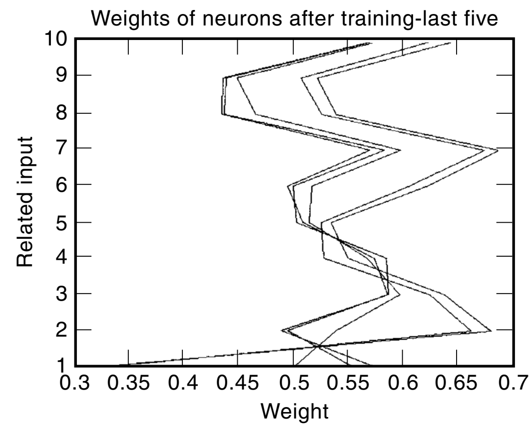

Figure 4. Initial weights, second class.

Artificial neural networks are broadly classified by the type of their connective structure, input–output transfer function, and learning paradigm, which describes how the connective weights are adjusted to adapt the network's dynamic performance to achieve application goals.

The so-called self-organization and feature mapping in unsupervised neural networks is typically associated with the special adaptive behavior of connective weights in a training phase, intended to selectively extract salient input features under either a deterministic or a stochastic environment. A self-organized learning style capitalizes on the competition among output neurons and their surrounding neighborhoods to code input distribution which consistently improves with training experience.

The network weights asymptotically approach exemplars of distinguished input clusters, which is reminiscent of simulated annealing and similar approaches in the related field of global optimization.

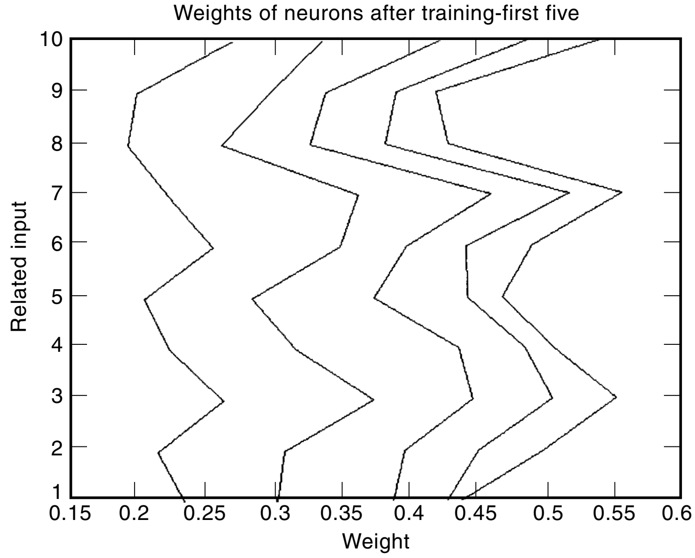

Figure 3. Initial weights, first class.

SELF-ORGANIZATION PRINCIPLES

Self-organizing feature maps topologically emulate salient features in the input signal space at the neural network output without explicitly using supervision or even reinforcement of correct output behavior.

The network's output neurons are usually conveniently arranged in single one-dimensional or two-dimensional layers. Full connectivity to the inputs is tacitly assumed. Lateral positive and negative feedback connections are also applied to help in convincingly deciding the outcome of competitive learning. Winning a competition lets a specific output neuron reach "on state" and thus updates its weights and the weights of its surrounding neighborhood. Normalization of all weights, in addition to controlling the size of surrounding neighborhoods, usually improves the network performance by equalizing the relative changes in weight connections.

Neuron activities and interactions can be represented by a set of discrete nonlinear mathematical equations, as proposed by Kohonen.

Therefore, the strengths of interconnective weights are expressed in an $n \times m$ weight matrix $W(k)$, and the lateral feedback coefficients are similarly collected in an $n \times n$ matrix C, which has a symmetrical band structure. Furthermore, the

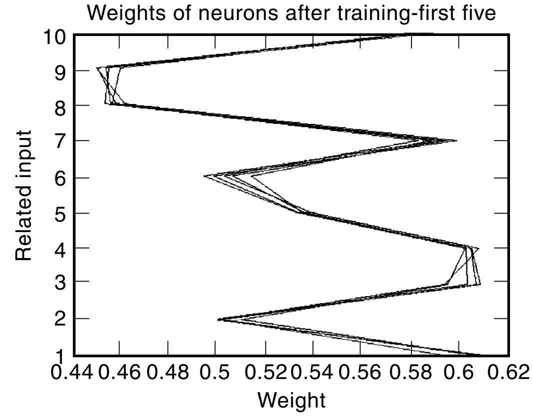

Figure 5. Converged weights, first class.

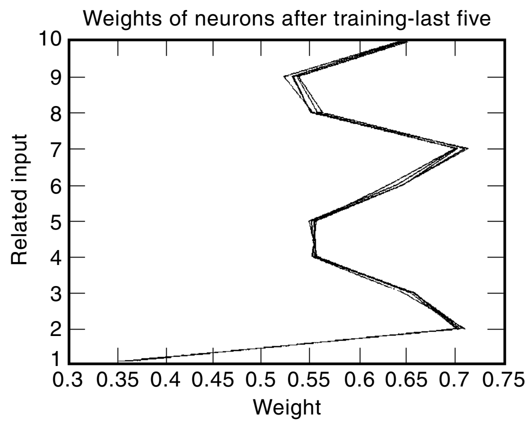

Figure 6. Converged weights, second class.

width of this band structure determines the effective size of neighborhoods surrounding each output neuron: Let n be the total number of output layer neurons, and let $Y(k) \in R^n$ be the neuron outputs at the kth discrete itration step. Let $\boldsymbol{X}(k) \in R^m$ and $\boldsymbol{U}(k) \in R^n$ be the input stimuli vector and the net weighted sum.

Finally, consider a nonlinear activation function designated by Φ: $R^n \to R^n$

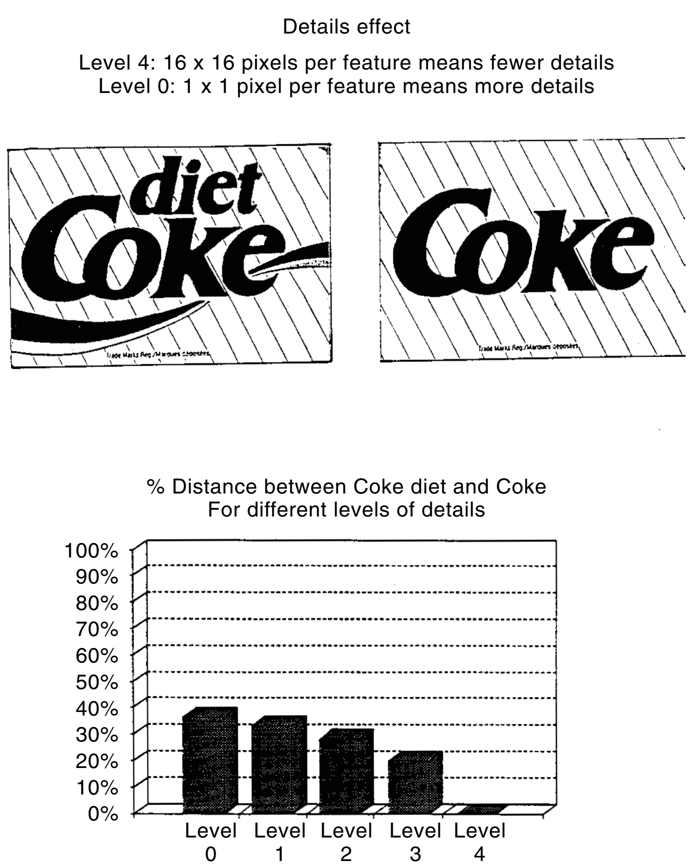

Figure 7. Effect of image details on recognition.

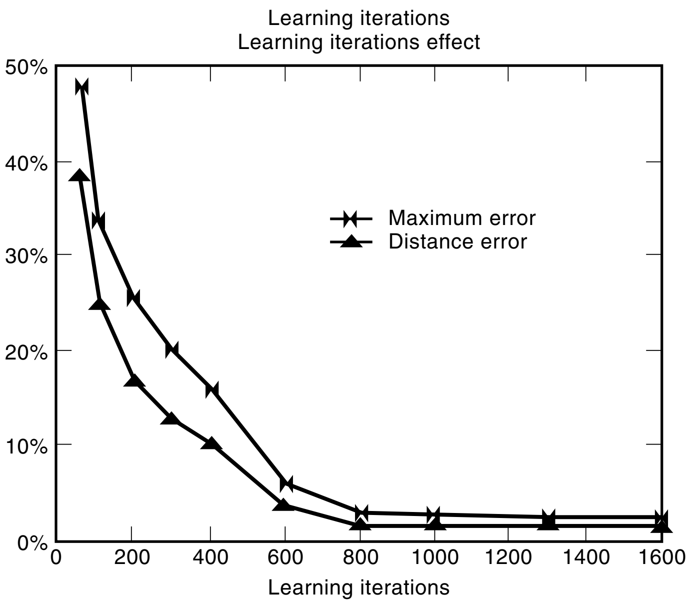

Figure 8. Convergence of learning algorithm.

Then the output neuron activity is modeled by

$$\boldsymbol{Y}(k+1) = \Phi[\boldsymbol{V}(k)] \tag{1}$$

$$\boldsymbol{V}(k) = \boldsymbol{U}(k) + \beta C(k) Y(k) \tag{2}$$

$$\boldsymbol{U}(k) = W(k)\boldsymbol{X}(k) \tag{3}$$

and β reflects a scalar relaxation factor that increases or decreases the effect of lateral feedback connections.

The set of Eqs. (1)–(3) may be solved assuming typical center-surrounding input vector patterns $\boldsymbol{X}(k)$. Considerable sim-

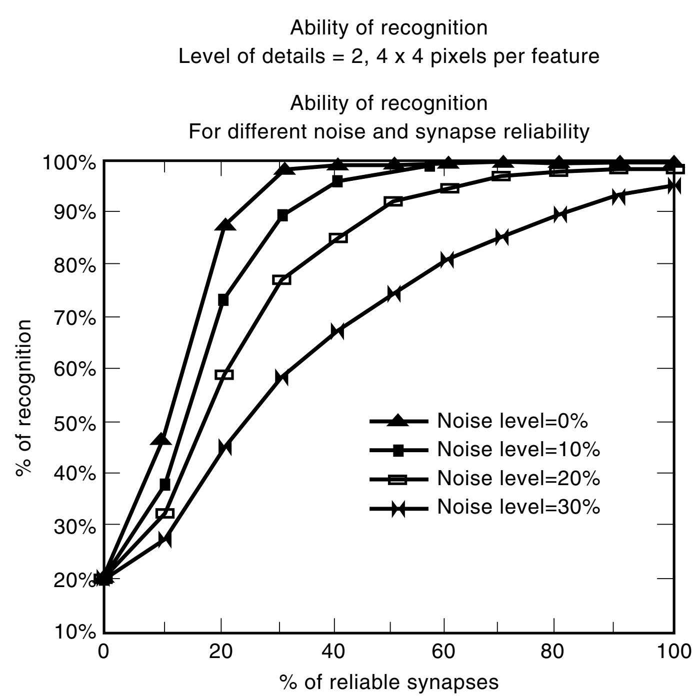

Figure 9. Effect of noise on recognition.

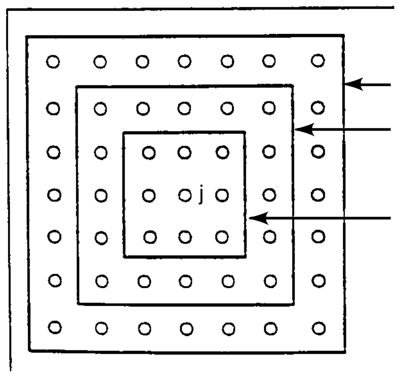

Figure 10. Topology of neighborhoods.

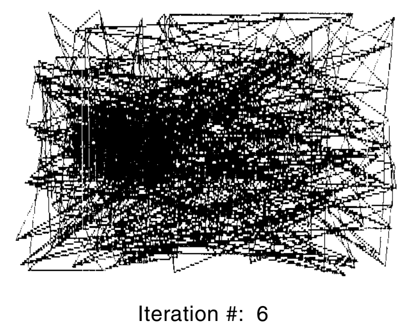

Figure 12. Initial lattice weights.

plification is effected if Φ is taken to be piecewise linear and if

$$C(k) = C \tag{4}$$

These assumptions produce an emergent output neuron behavior that amounts to ignoring lateral feedback and using a variable-size surrounding neighborhood that depends on k. The concept of neighborhood allows gradually decoupling topological groups of output layer neurons, which is similar to fuzzy system membership functions.

LEARNING ALGORITHMS

Self-organized feature mapping is essentially a transformation from the input signal space to a topologically ordered but reduced-dimensional output neuron activity pattern. $\boldsymbol{X} \in R^{mn}$ is the input vector, $\boldsymbol{W}_i \in R^m$ is the associated connection weight.

The dot vector product $\boldsymbol{W}_i\boldsymbol{X}$ is generally a scalar measure of the geometric projection of the input vector on a subspace spanned by the weights.

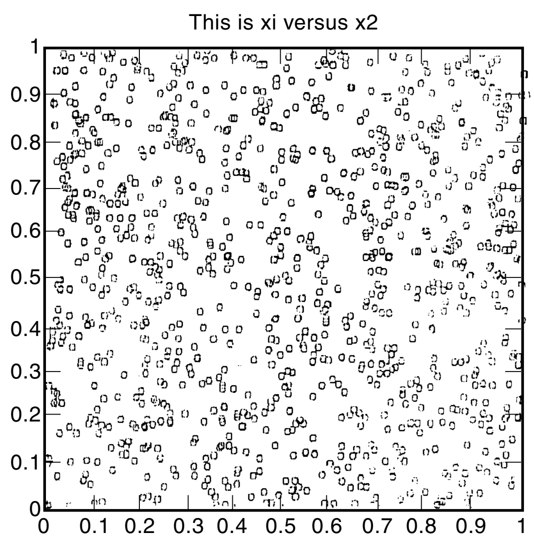

Figure 11. Input distribution.

$$(\boldsymbol{W}_i, \boldsymbol{X}) = \|\boldsymbol{W}_i - \boldsymbol{X}\|_2 = \sum_{j=1}^{m} (W_{ij} - X_j)^2 \tag{5}$$

$i^*(\boldsymbol{X}) \equiv$ number of neuron widths $(\boldsymbol{W}_i, \boldsymbol{X})$ = global minimum

$$\boldsymbol{W}_i(k+1) = \boldsymbol{W}_i(k) + \alpha(k)[\boldsymbol{X} - \boldsymbol{W}_i(k)] \tag{6}$$

where $\boldsymbol{W}_i(k) \in R^m$ is the weight vector at the kth iteration, $\boldsymbol{X} \in R^m$ is the input vector, and $\alpha(k)$ is a scalar learning rate.

Several possible learning rate expressions are as follows:

$$\alpha(k) = \alpha \tag{7}$$

$$\alpha(k) = \frac{1}{k} \tag{8}$$

$$\alpha(k) = \alpha_0 \exp\left(\frac{-k}{J_\alpha}\right) \tag{9}$$

$$\alpha(k) = \alpha_0 \exp\left(\frac{-k^2}{J_\alpha}\right) \tag{10}$$

$$\alpha(k) = \alpha_0 \frac{1}{\ln(1+k)} \tag{11}$$

$x \in R^{10}$

Therefore, a designated neuron has the most match, expressed in its connective weight, to the input features. The same idea can be conceptualized by letting the output neurons compete for replication of input vectors, by using the Euclidean distance.

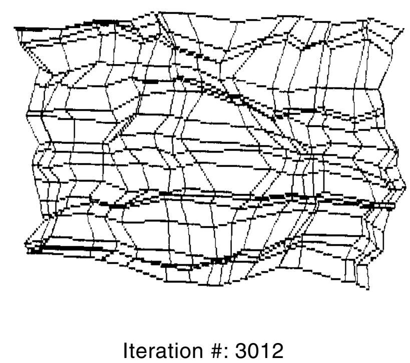

Figure 13. Trained lattice weights.

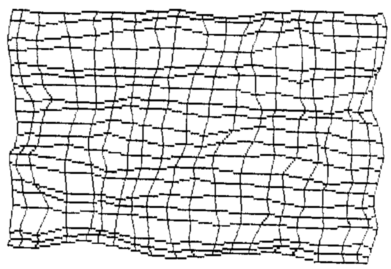

Figure 14. Converged lattice weights.

Now an appropriate neighborhood size is postulated a priori and later on allowed to be relaxed by size reduction.

These parametric neighborhood indicator functions are useful

Rectangular
Trapezoidal
Gaussian

Each of these functions allows the neuron activity patterns, discussed earlier, to various approximation degrees.

This adaptive learning updates the weights of the winning neuron and its surrounding neighbors according to a modified Hebbian rule.

COMPUTER SIMULATIONS

To gain insight into the self-organized feature map and the effect of some parameters used in the algorithm, we consider two distinct illustrative computer simulations.

Clustering of Two-Dimensional Data

Here, two-dimensional distributed data sets are addressed. The objective is to extract and approximately replicate the salient features in input vector topological clusters by using 100 output neurons arranged in a 10×10 two-dimensional layer.

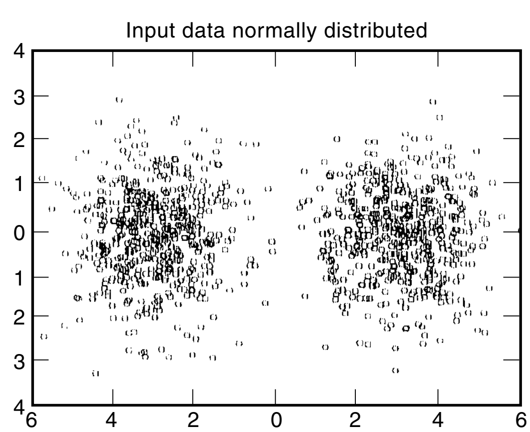

Figure 15. Overlapping inputs distribution.

A training set of 1000 input vectors is presented to the neural network without the benefit of supervision or reinforcement. Initial weights are random and have small values. The results obtained after several iterative epochs are presented along with the corresponding network parameters. Several statistically independent simulations are run to ensure that the network orients its weights toward the important input distribution characteristics. The renormalization of weights is effective in keeping the network on track. Self-organization is evident following sufficient presentations of input and learning iterations.

Classification of Scanned Two-Dimensional Images

Input images exhibit considerable center-surround correlation, which aids in forming output neuron prototypes that distinguish among image classes. Each scanned image is sliced into N parts. The reliability of classification depends on N. Subsequently each part is subdivided into N segments, and then the average image intensity over all parts is calculated.

Ten output neurons are assumed, arranged in a one-dimensional layer. Four image classes were considered with 10% noise added as an illustration. More than 80% classification accuracy was possible after tuning the experimental network parameters. The observations at neural network outputs are independent and identically distributed, so that a statistical estimate of performance is valid.

The connective weight quantization to reduce the overall computational task does not affect performance, especially under the continuous control of neighborhood size reduction.

The presentation of results to visualize input–output mapping requires plotting a set of piecewise linear curves that show how the output neurons are affected by input classes. Because the output layer in this case is required to be one-dimensional, the algorithm gave only suboptimal results.

CONCLUSION AND ADVANCED APPLICATIONS

The self-organizing feature map neural network has been illustrated to provide an effective and natural approach for topological classification of sensory input signals. The network performance hinges, however, on the neighborhood indicator function and the learning rate to achieve proper convergence. Normalization and preprocessing of input vectors also enhances the extraction of invariant and salient features embedded in the input space.

The recognition abilities are robust to noisy corruption of inputs and to inaccuracies representing connective weights.

Applications of self-organizing feature maps abound in engineering. They have been exploited in robot control, equalization of communications channels, texture classification, vehicular radar navigation, biomedical diagnosis, and detecting overlaps in manufacturing group technology.

Advanced applications involve combining self-organization learning paradigms with least mean squares supervision to achieve high performance. The role of the feature map is to distill key features from input space and to ease the classification task, especially in the presence of complicated boundaries between classes.

Hierarchical self-organization also presents opportunities for future advances in streamlining computation on multiprocessor systems.

BIBLIOGRAPHY

D. Kleinfeld, Sequential state generation by model neural networks, *Proc. Natl. Acad. Sci. USA,* **83**: 9469–9473, 1986.

E. Knudsen, S. DuLac, and S. Easterly, Computational maps in the brain, *Annu. Rev. Neurosci.,* **10**: 41–65, 1987.

T. Kohonen, Correlation matrix memories, *IEEE Trans. Comput.,* **C-21**: 353–359, 1972.

T. Kohonen, Self-organized formation of topologically correct feature maps, *Biological Cybern.,* **43** (1): 59–69, 1982.

T. Kohonen, An introduction to neural computing, *Neural Networks,* **1**: 3–16, 1988.

T. Kohonen, Physiological interpretation of the self-organizing map algorithm, *Neural Networks,* **6**: 895–905, 1993.

M. Kuperstein, Neural model of adaptive hand-eye coordination for single postures, *Science,* **239**: 1308–1311, 1988.

A. Lapedes and R. Farber, A self-optimizing non-symmetrical neural net for content addressable memory and pattern recognition, *Physica,* **22D**: 247–259, 1986.

Y. Linde, A. Buzo, and R. Gray, An algorithm for vector quantizer design, *IEEE Trans. Commun.,* **28**: 84–95, 1980.

W. Little and G. Shaw, A statistical theory of short term memory, *Behav. Biol.,* **14**: 115–133, 1975.

S. Luttrell, Self-organization: A derivation from first principles of a class of learning algorithm, *Proc. IJCNN 89,* **2**: 1989, pp. 495–498.

C. von der Malsburg, Outline of a theory for ontogenesis of iso-orientation domains in visual cortex, *Biological Cybern.,* **45** (1): 49–56, 1982.

H. S. ABDEL-ATY-ZOHDY
M. A. ZOHDY
Oakland University

SELF-TIMED CIRCUITS. See ASYNCHRONOUS CIRCUITS; ASYNCHRONOUS SEQUENTIAL LOGIC.