The Effect of Stochastic Interconnects in Artificial Neural Network Classification

Robert J. Marks II, Les E. Atlas, Dong C. Park and Seho Oh
Interactive Systems Design Laboratory
University of Washington, FT-10
Seattle, WA 98195

The storage capacity of any viable artificial neural network classifier increases with the number of available neurons. Assuming that each neural state is in some sense uncorrelated with those remaining, each neuron represents a computational degree of freedom available to the network. The number of degrees of freedom can be artificially increased through the use of neurons in a hidden layer, the states of which can be almost any nonlinear combination of the stimulus neural states. In this paper, such nonlinearities are generated with stochastically chosen interconnects between the input and hidden neural layers with a sigmoidal nonlinearity at each hidden neuron. The hidden to output interconnects are chosen to be a (trainable) projection matrix whose values are a function of the stochastically chosen interconnects and the training data. Preliminary simulations of such networks show an approach to fixed generalization boundaries as the number of hidden neurons becomes larger.

Although the use of hidden neurons with arbitrarily determined nonlinear states is potentially applicable to a large number of artificial neural networks, we limit our investigation here to the projection artificial neural network. In this section, we briefly review the network. A more detailed explanation can be found elsewhere[1].

A set of stimuli vectors $\{s_n \mid 1 \le n \le N\}$ is to be made to correspond to a set of response vectors $\{r_n \mid 1 \le n \le N\}$. That is, we wish to design a classifier that will output, say, r_3 when the input is s_3. We define the stimulus and response matrices respectively as

$$\mathbf{R} = [\, r_1 \mid r_2 \mid \ldots \mid r_N \,]$$

and

$$\mathbf{S} = [\, s_1 \mid s_2 \mid \ldots \mid s_N \,].$$

The hidden states will be denoted by $\{\, h_n \mid 1 \le n \le N \,\}$ where

$$h_n = \varphi\, s_n ; \quad 1 \le n \le N.$$

where φ is some nonlinear operation. The hidden layer matrix follows as

$$\mathbf{H} = [\, h_1 \mid h_2 \mid \ldots \mid h_N \,].$$

In an artificial neural network architecture, the number of input neurons is equal to the length of a stimulus vector. Each input neuron is connected to each hidden neuron in order to achieve a nonlinear mapping. The interconnects between the hidden and output neurons are given by elements of the projection matrix

$$\mathbf{C} = \mathbf{R}\, [\, \mathbf{H}^T \mathbf{H} \,]^{-1} \mathbf{H}^T. \tag{1}$$

In practice, the hidden to output interconnects are trained using an updating rule that requires examination of the training data only once [1].

Once trained, the network output, o, corresponding to an input vector, i, is given by

$$o = \mathbf{C}\,\varphi\, i \tag{2}$$

A nonlinearity φ that is useful in artificial neural network architectures is

$$\varphi\, s_n = \eta\, \mathbf{T}\, s_n$$

where $\mathbf{T}$ is the matrix of input-to-hidden interconnects and η is a nonlinear pointwise vector operator (e.g. sigmoid). A hidden neuron adds the contribution from all of the inputs and adopts a state equal to that sum passed through the hidden neuron nonlinearity. Equation (2) then becomes

$$o = \mathbf{C}\,\eta\,\mathbf{T}\, i$$

Almost any nonlinearity will allow the trained artificial neural network to respond correctly to training data. The manner in which the network responds to data outside of the training set (i.e. how the network *generalizes*) is determined by the choice of nonlinearity. This is illustrated in the following example.

EXAMPLE 1: Consider the two bit parity problem with

$$\mathbf{S} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix} \tag{1}$$

and

$$\mathbf{R} = \begin{bmatrix} 1 & -1 & -1 & 1 \end{bmatrix} \tag{2}$$

The parity problem is geometrically depicted in Figure 1. The coordinate pair (1,1), for example, is assigned a value of 1 shown as a small square. The small circles denote a value of -1. In order to perform the parity operation, we require a minimum of four hidden neurons corresponding to the four columns of both $\mathbf{S}$ and $\mathbf{R}$. We choose the input-to-hidden interconnect matrix

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ a & a & a & a \end{bmatrix} \tag{3}$$

(where $a = 3/2$) and the four hidden layer nonlinearities

$$\eta_1(z) = \exp(z) \qquad \eta_2(z) = \exp(-z)$$
$$\eta_3(z) = \tanh(z) \qquad \eta_4(z) = \operatorname{sech}(z) \tag{4}$$

The resulting network generalization is shown in Figure 2. The plot, is that of the sign of the single output neural state, o, versus the two input neural states i_1 and i_2. Note that the response to the training data is at the corners of the square and is correct. Note also that the partition in the generalization plot is parallel to the 45° line $i_1 = -a\, i_2$. This is due to our choice of the $\mathbf{T}$ matrix in (3). The input to each of the hidden neurons is $i_1 + a\, i_2$. A plot of this function on the (i_1,i_2) plane results in equal potential contours parallel to the line $i_1 = -a\, i_2$. Any function of $i_1 + ai_2$ will have these same contours (with, of course, different values on each contour). Thus, independent of our choice of functions in (4), our generalization is restricted to have partition boundaries parallel to the line $i_1 = -ai_2$.

EXAMPLE 2: We repeat the parity problem using the interconnect matrix

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ a & a & -a & -a \end{bmatrix} \tag{5}$$

and the nonlinearities $\eta_3(z) = \exp(z)$ and $\eta_4(z) = \exp(-z)$. The remaining two nonlinearities remain as in (4). Two of the hidden neurons now see an input of $i_1 - ai_2$. The resulting generalization, shown in Figure 3, is now close to being a least-mean-square partition of the training data.

The preceding two examples clearly illustrate the consequences of the choice of the input-to-hidden interconnects on the resulting generalization. The highly regularized structure in (?) gave an architecturally constrained generalization. The interconnects in (4) would generate similar artifacts in a more complex partitioning requirement. In the next example, we empirically explore the use of a larger number of hidden neurons and a stochastically chosen **T** matrix and its effects on the network's generalization.

EXAMPLE 3: We continue with our running parity example as defined by the training matrices in (1) and (2). We will use a total of P hidden neurons and a P by 2 input-to-hidden interconnect matrix, **T**, the elements of which are uniform random variables over the interval $(-1/2, 1/2)$. The nonlinearities at each of the neurons were chosen to be

$$\eta_p(z) = \exp(-z) - 1 \quad ; 1 \leq p \leq P \tag{6}$$

Generalization results are shown in Figures 4 a, b, c, and d for of $P = 4$, 10, 18 and 50 respectively. As in our other simulations, the generalization reached a steady state for large P.

EXAMPLES 4 and 5: Shown in Figure 5 is the generalization of parity training using 50 hidden neurons with input-to-hidden interconnects chosen from a zero mean unit variance normal distribution. The nonlinearity in (6) was used at each hidden neuron. The result of using a uniform distribution on (0,1) to solve the same problem is shown in Figure 6.

EXAMPLE 6: The $P = 50$ hidden neuron network was trained on the data corresponding to

$$\mathbf{T} = \begin{bmatrix} 1 & -1 & 1 & -1 & 0 \\ 1 & 1 & -1 & -1 & 0 \end{bmatrix}$$

and

$$\mathbf{R} = \begin{bmatrix} -1 & -1 & -1 & -1 & 1 \end{bmatrix}$$

This is geometrically illustrated in Figure 7 with the small circle denoting -1 and the square +1. Using interconnects randomly chosen from a uniform distribution on $(-1/2, 1/2)$ and the neural nonlinearity in (6), the generalization in Figure 8a was obtained. The same simulation was repeated for

$$\mathbf{R} = \begin{bmatrix} -1 & -1 & -1 & -1 & 4 \end{bmatrix}$$

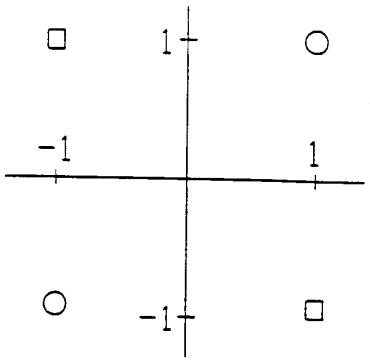

FIG. 1

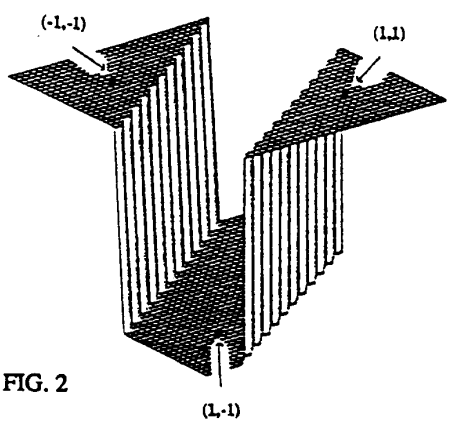

FIG. 2

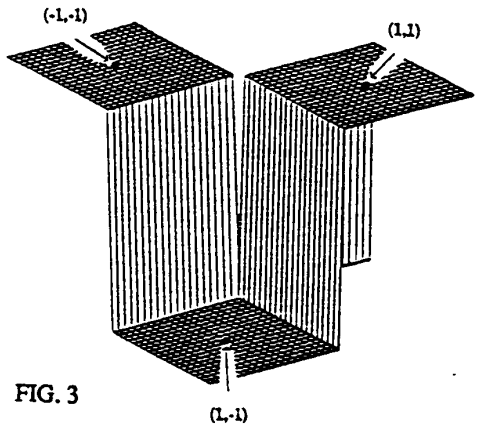

FIG. 3

In other words, the square at the origin in Figure 7 was given a more dominant value. This dominance is reflected in Figure 8b where the area of the resulting generalization for +1 is shown to be significantly increased.

The examples in this paper illustrate the potential use of stochastically chosen interconnects to increase the storage capacity and the discrimination ability of classification artificial neural networks. Future research should be directed at understanding the effects of the interconnect probability distribution and the hidden layer nonlinearities on the network's generalization characteristics.

REFERENCE

1. R.J. Marks II, Les E. Atlas and Seho Oh "Generalization in layered classification neural networks", **Proceedings of the IEEE International Symposium on Circuits and Systems**, Helsinki, June 7-9, 1988.

ACKNOWLEDGEMENTS

This work was supported by the Washington Technology Center at the University of Washington and, in part, by the SDIO/IST administered by ONR through the Optical Systems Lab at Texas Tech University in Lubbock. A discretionary gift from the Physio control Corporation for the authors' work in neural networks is also gratefully appreciated. In addition, L.E. Atlas was supported by an NSF Presidential Young Investigator Award.

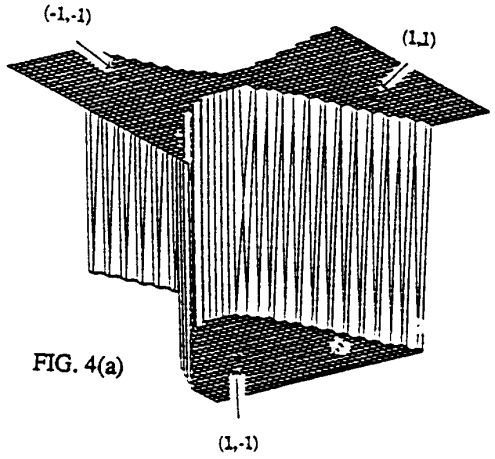

FIG. 4(a)

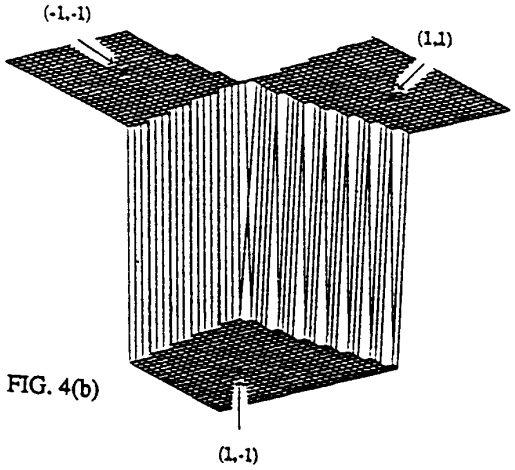

FIG. 4(b)

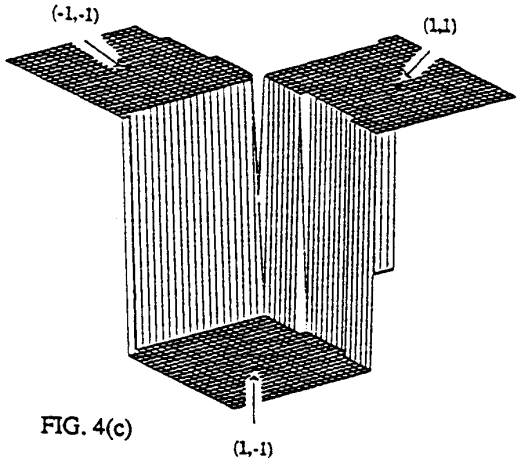

FIG. 4(c)

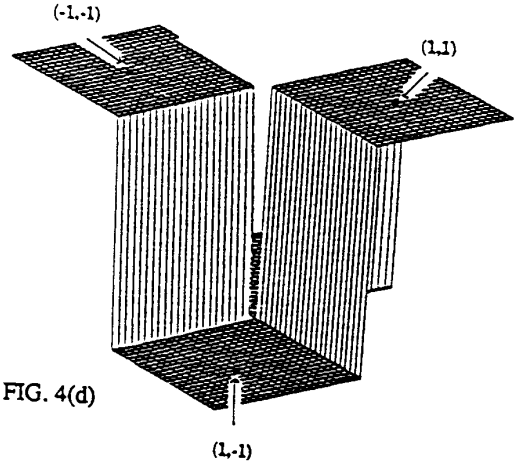

FIG. 4(d)

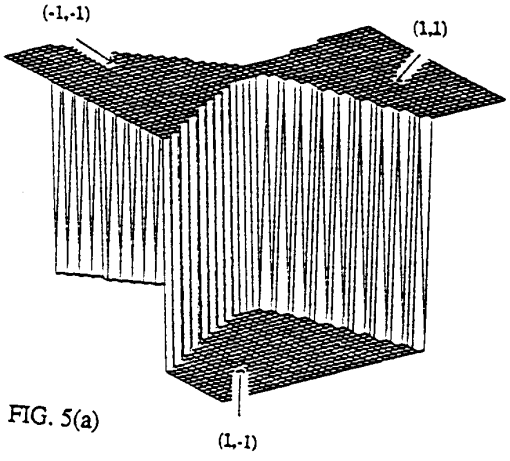

FIG. 5(a)

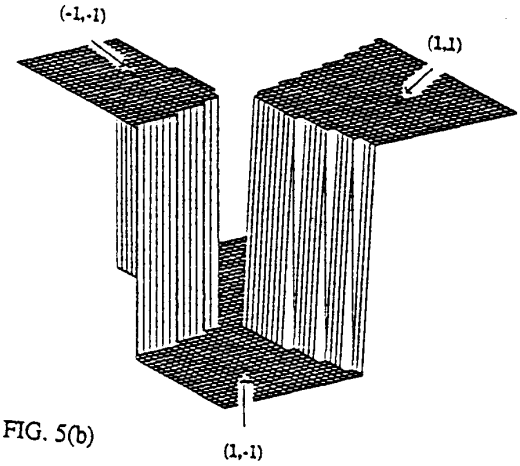

FIG. 5(b)

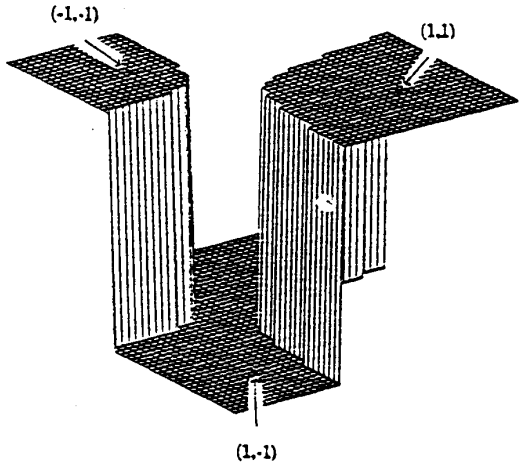

FIG. 6

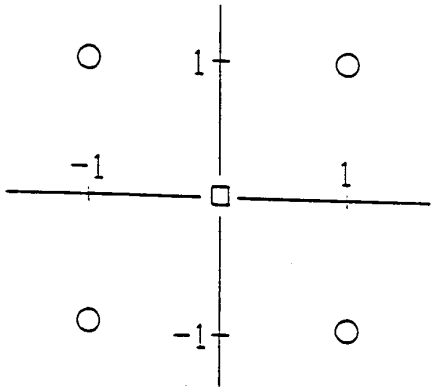

FIG. 7

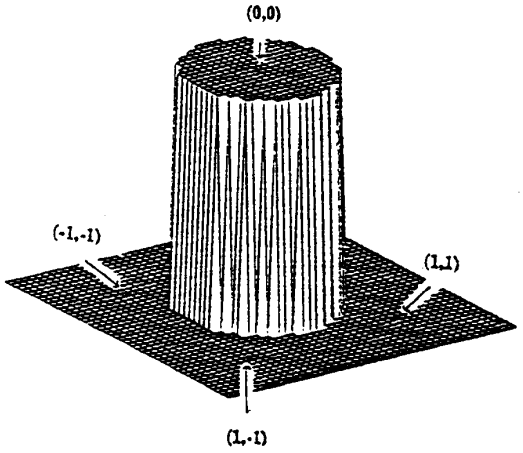

FIG. 8(a)

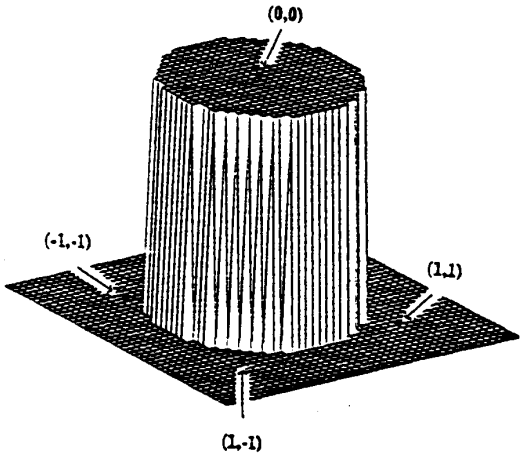

FIG. 8(b)