INPUT DATA TRANSFORMATION FOR BETTER PATTERN CLASSIFICATIONS WITH FEWER NEURONS

Yasuhiro Ota
Bogdan Wilamowski
Electrical Engineering Department
University of Wyoming
Laramie, WY 82071, USA

Abstract:

Ordinary discriminant functions for pattern separations are normally linear. Neural networks with one-layer architecture can classify only linearly separable patterns, and thus multilayer neural networks are required for separation of nonlinearly separable patterns. In this paper, an improved formulation of discriminant functions with fewer neurons is proposed. This is accomplished by introducing an additional dimension to a set of input patterns.

I. Introduction:

A pattern is the quantitative description of an object, phenomenon, or event. A classification of patterns can be spatial or temporal. Examples of the former case are pictures, video images, and characters. Examples of the latter case include speech signals, seismograms, and electrocardiograms, which normally involve ordered sequences of data appearing in time. The goal of pattern classification is to assign a physical object, phenomenon, or event to one of the prespecified classes. The mechanism of pattern recognition (classification) in the human brain seems to be almost impossible to reveal it. However, an artificial intelligence classifying system consists of an input transducer which provides the input pattern data to the feature extractor [1]. Typically, inputs to the feature extractor are sets of data vectors that belong to a certain category.

Several designs have been presented in the past for classifying patterns using n-dimensional discriminant functions. The efficient classifiers, in general, are described by discriminant functions that are nonlinear functions of input patterns [2][3]. As was described by Marvin Minsky and Seymour Papert one-layer neural networks have very limited ability for pattern classifications [4]. They can classify only linearly separable patterns; therefore, multilayer neural networks are required for separation of nonlinearly separable patterns. This paper discusses how to reduce the number of neurons with an effective nonlinear pattern classification. The formulation of the input data transformation method is described in Section II, and the simulation of a proposed network design is shown in Section III. Section IV concludes the design and gives suggestions of possible future studies applicable to this design.

II. Structure of Pattern Classifications

First, the assumption is made that both a set of n-dimensional input patterns $\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_P\}$ and the desired classification for each input pattern $\{\mathbf{d}_1,\mathbf{d}_2,\cdots,\mathbf{d}_P\}$ are known. The size P of the pattern set is finite and usually much larger than the dimensionality n of the pattern space. In many practical cases, it is assumed that P is much larger than the number of categories (classes) R. The goal is to classify input patterns into R categories. For given input patterns $\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_P\}$ with R categories, each category of input patterns normally has a center of gravity, and it can be found by

$$\mathbf{x}_{CG_R}=\frac{\mathbf{x}_{R_1}+\mathbf{x}_{R_2}+\cdots+\mathbf{x}_{R_k}}{k} \tag{1}$$

where the subscript CG_R stands for the center of gravity for the R-th category with k input data in that category. Once the centers of gravity for each category are defined, the radii r_R of circles (or spheres) that enclose all the input data that belong to a certain category can be found from the following equation:

$$r_R=\left|\mathbf{x}_{R_k}-\mathbf{x}_{CG_R}\right|_{\max} \tag{2}$$

One more parameter, D_{MAX}, must be defined in order to transform n-dimensional input data into $(n+1)$-dimensions. The maximum distance of an input point from the origin, D_{MAX}, is used to scale all the input data in transforming them into new $(n+1)$-dimensional input arrays. Finally, the series of the following equations is utilized for the transformation of input patterns.

$$\begin{aligned}
z_1 &= \cos\alpha_1 \\
z_2 &= \sin\alpha_1\cos\alpha_2 \\
z_3 &= \sin\alpha_1\sin\alpha_2\cos\alpha_3 \\
\vdots \quad & \quad \vdots \\
z_n &= \sin\alpha_1\sin\alpha_2\cdots\sin\alpha_{n-1}\cos\alpha_n \\
z_{n+1} &= \sin\alpha_1\sin\alpha_2\cdots\sin\alpha_{n-1}\sin\alpha_n
\end{aligned} \tag{3}$$

where z_i $(i=1,2,\cdots,n+1)$ are the new $(n+1)$-dimensional transformed input data space and the arguments α_j $(j=1,2,\cdots,n)$ are defined by

$$\alpha_j=\left(\frac{x_j}{D_{MAX}}\right)\pi \tag{4}$$

Notice that $(n+1)$-dimensional input data are mapped such that

$$\sum_{i=1}^{n+1} z_i^2=1 \tag{5}$$

By employing relations (3) and (4) all the n-dimensional input data including the centers of gravity of all the R categories can be transformed. In order to find discriminant functions for the separation of patterns, equations of decision surfaces (separation planes) are required. Notice that the $(n+1)$-dimensional data at the center of gravity represent the normal vector of a separation plane; i.e.,

$$\hat{\mathbf{n}}_R = \left[z_{CG_{R_1}} \; z_{CG_{R_2}} \; \cdots \; z_{CG_{R_{n+1}}} \right] \tag{6}$$

Once some point which lies on this separation plane is known, the equation of the plane can be established. The point which lies on the boundary, r_R, of the original n-dimensional pattern should be mapped onto the edge of the boundary in the $(n+1)$-dimensional pattern, and hence this point, $\mathbf{z}_{EDGE_R}$, shoucld be used in formulating the equations of the discriminant functions. Thus, the discriminant functions can be given by

$$\hat{\mathbf{n}}_R^T \left(\mathbf{z} - \mathbf{z}_{EDGE_R} \right) = 0 \tag{7}$$

The above transformation can be used, not only for the condition with linear separability of patterns, but also for the case of linearly nonseparable patterns. The analytical weights for the neuron being activated for the $R\text{-}th$ category can then be given as

$$\mathbf{W}_R = \left[z_{CG_{R_1}} \; z_{CG_{R_2}} \; \cdots \; z_{CG_{R_{n+1}}} \quad -\sum_{i=1}^{n+1} z_{CG_{R_i}} z_{EDGE_{R_i}} \right] \tag{8}$$

III. Simulation Results

The following simple example will allow the reader to gain better insight into the discussion of the pattern classification issue. Now consider looking at two-dimensional patterns (n=1) with two categories (R=2). Initially, seventeen patterns were assigned in two-dimensional pattern space according to their membership in sets as follows:

$$\text{Class 1 } (\mathbf{d}_1 = +1)\text{:} \quad \left\{ \begin{bmatrix}5\\5\end{bmatrix}, \begin{bmatrix}3\\4\end{bmatrix}, \begin{bmatrix}5\\4\end{bmatrix}, \begin{bmatrix}6\\5\end{bmatrix}, \begin{bmatrix}4\\3\end{bmatrix}, \begin{bmatrix}4\\4\end{bmatrix}, \begin{bmatrix}7\\5\end{bmatrix} \right\}$$

$$\text{Class 2 } (\mathbf{d}_2 = -1)\text{:} \quad \left\{ \begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}9\\3\end{bmatrix}, \begin{bmatrix}1\\10\end{bmatrix}, \begin{bmatrix}5\\0\end{bmatrix}, \begin{bmatrix}8\\8\end{bmatrix}, \begin{bmatrix}2\\8\end{bmatrix}, \begin{bmatrix}5\\10\end{bmatrix}, \begin{bmatrix}10\\4\end{bmatrix}, \begin{bmatrix}12\\9\end{bmatrix} \right\}$$

In order to classify the given patterns into two categories with the ordinary decision lines, discriminant functions, at least four decision lines and a two-layer neural network are necessary as shown in Figure 1. On the other hand, only one neuron is necessary to perform the same function if the proposed design is employed since it is possible to separate the two categories with one circle as shown in Figure 2. Figure 3 illustrates the given input data after they have been transformed into three-dimensional patterns.

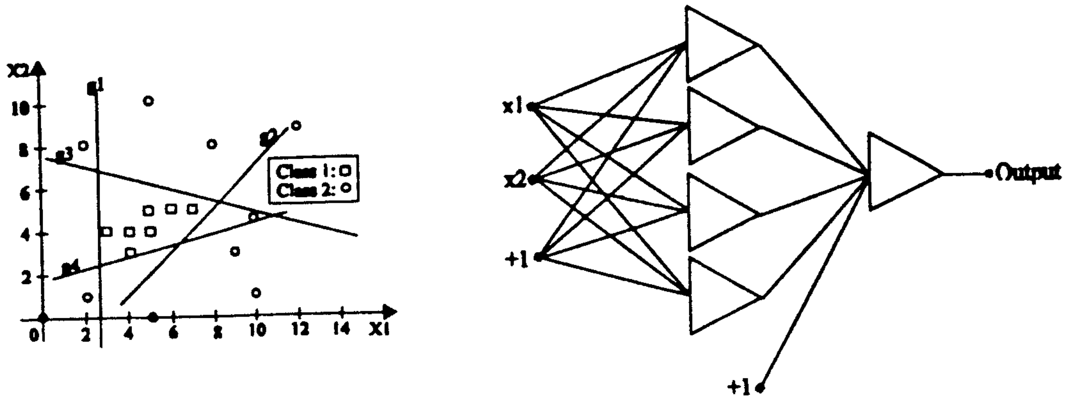

Figure 1. Two-Dimensional Input Patterns with Ordinary Discriminant Functions and the Two-Layer Neural Network.

Figure 2. Two-Dimensional Input Patterns with the Improved Model and the Single-Neuron Network.

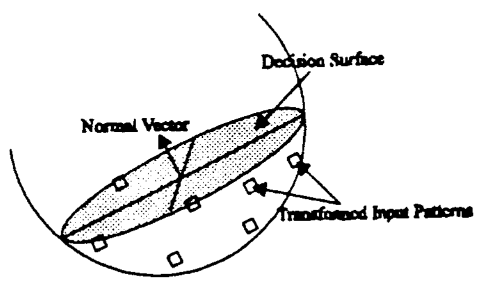

Figure 3. Input Patterns After the Transformation.

For the given single-neuron network, the analytical weight vector can be found by utilizing relation (8). This vector is given as

$$W_{1_{analytical}} = [\ 0.5305 \quad 0.5180 \quad 0.6710 \quad -0.8008\ ]$$

The popular error back-propagation training algorithm [5][6] (delta training for a single-layer network) can also be utilized to compute the optimal weight vector. The performance of the neural network will then be compared with both the analytical and the delta training weights. Initial weights for training are randomly chosen as in the normal procedure, and the total output error does converge towards zero as shown in Figure 4. After the training of the network with 500 iterations, the delta-trained weight vector is found to be

$$W_{1_{\Delta}} = [\ 0.4448\ \ 0.5034\ \ 0.7408\ \ -0.7890\ ]$$

Testing of pattern classifications is achieved with the two weight vectors listed above, and the results are tabulated in Figure 5. Figure 6 illustrates the mesh plots of the actual input-output nonlinear mappings in the original pattern space.

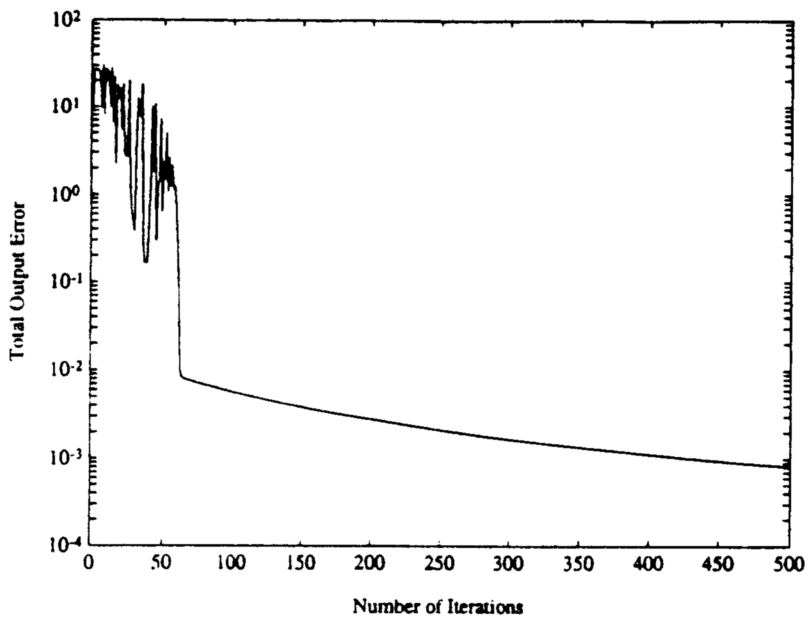

Figure 4. Total Output Error of the Neural Network with the Delta Training.

Test Input Pattern	Desired Output	Analytical Method		Delta Training	
		Output	% Error	Output	% Error
[4 5]	1	0.9226	7.74	0.9141	8.59
[12 11]	-1	-1.0000	0.00	-1.0000	0.00
[0 10]	-1	-0.9847	1.53	-0.9959	0.04
[9 1]	-1	-0.9963	0.37	-0.9927	0.73
[5 6]	1	0.8857	11.43	0.9923	7.78

Figure 5. Test Results of the Neural Network.

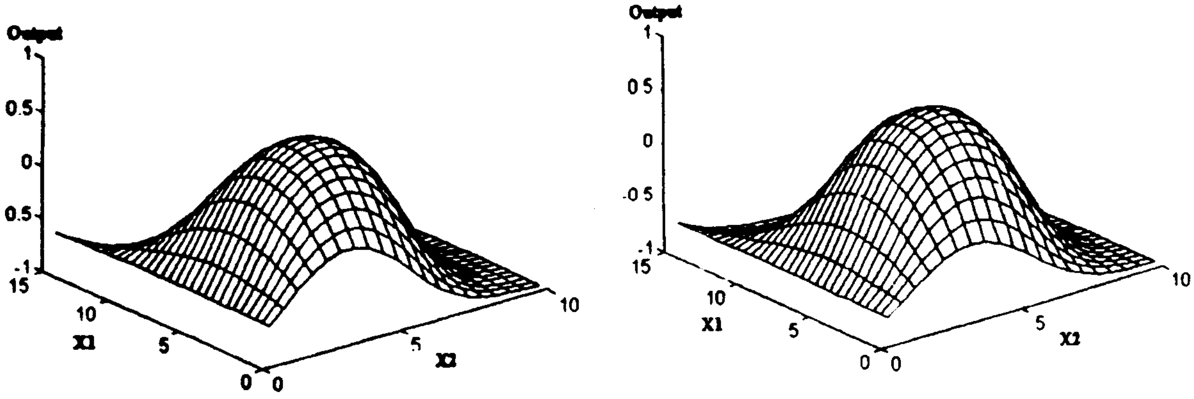

Figure 6. Actual Output Transfer Characteristics: (a) Using the Analytical Weight Vector and (b) Using the Weight Vector with the Delta Training.

As can be seen from this simulation, the proposed design of input data transformation for pattern classifications works well and is superior to the ordinary pattern classification designs.

IV. Conclusions

An improved technique of input data transformation for effective pattern classification with a minimum number of neurons has been presented in this paper. The simulation results clearly demonstrated that the number of necessary neurons could be effectively minimized with accurate classifications of input patterns by introducing an additional dimension (freedom) for the input patterns. The simulation revealed that even with a simple linearly nonseparable example the number of neurons required was reduced by using the improved method. In fact, the number of necessary neurons for the example case used was reduced by four.

Some possible further studies include testing of this design to higher dimensions with more complicated patterns that have a greater number of classifying categories although this input transformation technique can be virtually applied to any pattern classification problems of any dimensions..

References

[1] J. M. Zurada, Introduction to Artificial Neural Systems., West Publishing Company, St. Paul, MN., 1992

[2] H. C. Andrews, Introduction to Mathematical Techniques in Pattern Recognition., Wiley Interscience, New York, NY., 1972.

[3] B. Widrow and M. A. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Medaline, and Backpropagation," *IEEE Proceedings*, vol. 78, no. 9, pp. 1415-1442, Sept. 1990.

[4] M. Minsky and S. Papert, Perceptrons., MIT Press., Cambridge, MA., 1969.

[5] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks Are Universal Approximations," *Neural Networks*, vol. 2, pp. 359-366, 1989.

[6] E. D. Karin, "A Simple Procedure for Pruning Back-Propagation Trained Neural Networks," *IEEE Trans. on Neural Networks*, vol. 1, no. 2, pp. 239-242, 1990.