Application of Neural Computing in Pharmaceutical Product Development

Ajaz S. Hussain,[1,2] Xuanqiang Yu,[1] and Robert D. Johnson[1]

Received December 28, 1990; accepted May 15, 1991

Neural computing technology is capable of solving problems involving complex pattern recognition. This technology is applied here to pharmaceutical product development. The most commonly used computational algorithm, the delta back-propagation network, was utilized to recognize the complex relationship between the formulation variables and the *in vitro* drug release parameters for a hydrophilic matrix capsule system. This new computational technique was also compared with the response surface methodology (RSM). Artificial neural network (ANN) analysis was able to predict the response values for a series of validation experiments more precisely than RSM. ANN may offer an alternative to RSM because it allows for the development of a system that can incorporate literature and experimental data to solve common problems in the pharmaceutical industry.

KEY WORDS: neural computing; artificial neural networks; pharmaceutical formulation; response surface methodology; controlled release; capsules.

INTRODUCTION

The pharmaceutical product development process involves the optimization of formulation and process variables. Since this process is normally characterized by multiple objectives, quantitative prediction of the system behavior from basic physical and chemical principles is often difficult. The response surface methodology (RSM) has proven to be a useful tool for handling such problems (1). RSM allows the formulator to approximate the true system behavior as a function of the formulation and process variables and to determine the apparent optimum conditions. The utility of RSM in pharmaceutical product development has been demonstrated by several workers (2–8).

Recent advances in the areas of computer science, neuroscience, and applied mathematics has resulted in the development of artificial neural networks (ANNs). ANNs can identify and learn correlative patterns between input and output data pairs. Once trained, they may be used to forecast outputs from new sets of input conditions. The use of ANNs does not require prior understanding of the underlying process or phenomena under study (9). These features make ANNs well suited for solving problems in the product development area.

The purpose of this study was to illustrate an ANN approach for a pharmaceutical technology application and to compare it with RSM. The pharmaceutical problem chosen for this study involved the design of a controlled-release hydrophilic matrix capsule containing blends of anionic and nonionic cellulose ether polymers. The *in vitro* drug release profile is a function of the various polymer characteristics and their relative amounts in the matrix. Mechanistic analysis of this system is presented elsewhere (10).

NEURAL COMPUTING

Neural computing technology utilizes the basic principles of the human brain to solve pattern recognition types of problems. The ANN is the basic mathematical model for this computational system (11–17). This network is a series of highly interconnected layers of processing elements. These elements represent a mathematical analogue of biological neurons. The connections between each individual processing element and its counterparts in other layers represent the architecture of the network. In general, a typical ANN must have one input layer and one output layer and may contain one or more hidden layers.

Neural computations occur in a three-step sequence consisting of the learning, recall, and generalization phases. During the learning phase, the ANN is presented with a series of input/output data pairs (the training data set) whereby the ANN attempts to learn the inherent relationships in the data set. Once the learning phase is completed, the trained ANN can operate in the recall phase where responses are generated from the input data used in the training set. In the generalization phase, this ANN is used to generate responses from new input data.

Of the many possible neural architectures, the back-propagation network is the most commonly used (9). This network is a system of fully interconnected layers of processing elements, as shown in Fig. 1. The input data are distributed to the processing elements in the input layer. The result from each processing element in the input layer is sent to the processing elements in the hidden layer. The hidden layer processes these signals and generates output signals that are transmitted to the nodes of the output layer. The processing results from the output layer becomes the output of the entire network.

The function of the processing elements is to receive, modify, and propagate signals. A typical processing element is shown in Fig. 2. It is capable of receiving signals from the other elements in the preceding layer or directly from input data, summing the signals, transforming the sum, and sending the results to the other elements in subsequent layers. There are a number of transform functions currently being used in neural network analysis. The most commonly used transform function in back-propagation neural network analysis is the sigmoidal function:

$$Y_i = \frac{1}{1 + e^{-x}} \quad (1)$$

During the learning phase, the information that is passed from one processing element to another is contained within a set of weights. These weights, which are adjustable during successive computational sweeps through the network, are

[1] The Division of Pharmaceutics and Drug Delivery Systems, College of Pharmacy, University of Cincinnati-Medical Center, Cincinnati, Ohio 45267-0004.

[2] To whom correspondence should be addressed.

0724-8741/91/1000-1248$06.50/0

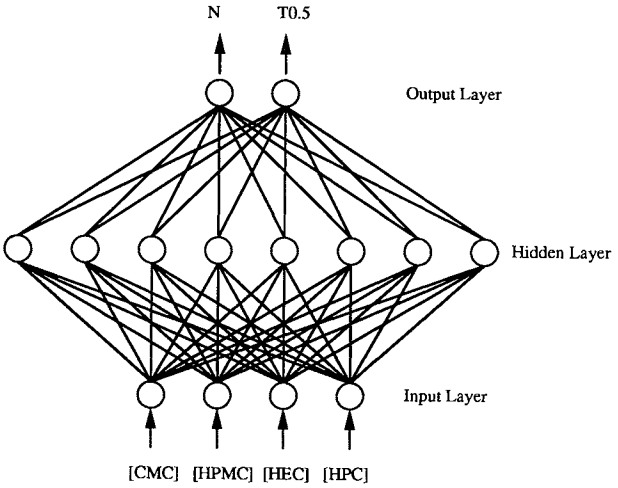

Fig. 1. Artificial neural network architecture for the problem studied.

determined by minimizing the sum of the squares of the deviation between the experimental and network output values. The minimization procedure involves two steps. The input data presented to node i from node j of the preceding layer is first transformed by the following equation:

$$I_i = \sum_j w_{ji} O_j \tag{2}$$

where w_{ji} is the adjustable weight connecting element i to element j, O_j is the output from the previous element, and I_i is the resultant input to element i. In the input layer, w_{ji} is equal to 1 and O_j corresponds to the sigmoidally transformed input data. The output from element i is transformed using the sigmoidal transform function given by Eq. (1), where x becomes I_i and Y_i becomes O_i. The most commonly used learning rule in the back-propagation algorithm is the delta rule, where the weights are adjusted by the following equation:

$$w_{ji}^{n+1} = w_{ji}^{n}\,\beta(\delta_i O_j) + \alpha(w_{ji}^{n} - w_{ji}^{n-1}) \tag{3}$$

where α is the momentum factor, β is the learning rate, n is the iteration number, and δ is the error signal from the previous processing element. The momentum factor is used to accelerate the learning process by incorporating some information from the previous computational sweep. The learning rate is a multiplication factor which is used to determine the magnitude of successive weight changes. Too large a learning rate may cause such drastic weight changes that the network is unable to learn effectively, whereas too small a learning rate slows down the learning process. The error

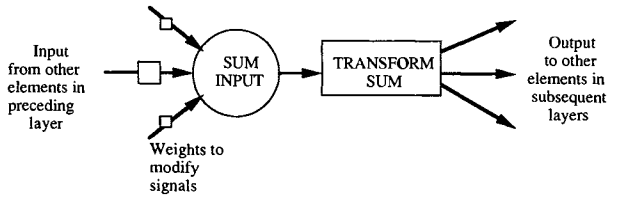

Fig. 2. Schematic of a typical processing element.

signal, δ, represents the deviation between the output of the element and its target value. The error signal for the process element i of the output layer is given by

$$\delta_i = (T_i - O_i) f'(I_i) \tag{4}$$

and that for the processing element j in the hidden layer by

$$\delta_i = f'(I_i) \sum_k (\delta_k W_{ik}) \tag{5}$$

where T_i is the target value, $f'(I_i)$ is the first derivative of the transfer function with respect to I_i, δ_k is the error signal of element k in the hidden layer, and w_{ik} is the weight between node i and node k. Once the network has been trained, it can be used for both recall and generalization.

MATERIALS AND METHODS

Formulation and Drug Release Studies

The hydrophilic polymers used in the capsule matrix were sodium carboxymethyl cellulose (CMC; DS 0.7, DP 3200, Scientific Polymer Product, Inc., Ontario, NY 14519) and hydroxypropylmethyl cellulose (HPMC; Methocell K4M, Dow Chemical Co., Midland, MI 48674). Hydroxypropyl cellulose (HPC; Klucel HF) and hydroxyethyl cellulose (HEC; Natrosol HHX) were obtained from Aqualon Co., Wilmington, DE 19850.

The effects of polymer blending on the *in vitro* drug release profiles were studied by using the four-component simplex centroid mixture design shown in Fig. 3 (21). The drug-to-polymer ratio was kept constant at 1:3. Hence, the mixture design is based only on the polymer weight fractions in the matrix. For all matrices, the weight fraction of each polymer in the matrix must adhere to the following constraint:

$$\sum_{i=1}^{4} \Theta_i = 1.0 \tag{6}$$

where Θ_i is the polymer mass fraction. The powders were mixed by passing through a No. 80 sieve 7–10 times. Approximately 275 mg of the mixture was filled in a size 2 hard capsule (Eli Lilly and Co., Indianapolis, IN 46285). The *in vitro* release studies were performed in a paddle apparatus (Vanderkamp 600, Van Kel Industries, Edison, NJ) at 50 rpm. Distilled water at 37°C was the dissolution medium. A stainless-steel screen mesh was placed above the capsule to keep it immersed in the dissolution medium. Chlorpheniramine maleate (Sigma Chemical Co., St. Louis, MO 63178) was used as the model compound. Dissolution samples were assayed by UV spectrophotometry (Varian DMS 100 UV spectrophotometer, Walnut Creek, CA) at 261 nm.

Response Variables

The response variables used to characterize the release profile were the release exponent N and the dissolution half-time $T_{0.5}$ in hours. Both parameters were estimated by nonlinear regression analysis (18) according to the following equation (19):

Table I. Experimental Design and Response Values

Formulation No.	CMC	HPMC	HEC	HPC	Exponent N, mean (SD)	$T_{0.5}$, mean (SD)
1	1	0	0	0	1.58 (0.04)	2.99 (0.08)
2	0	1	0	0	0.67 (0.02)	2.05 (0.06)
3	0	0	1	0	0.67 (0.03)	3.06 (0.19)
4	0	0	0	1	0.55 (0.02)	3.58 (0.19)
5	½	½	0	0	1.09 (0.07)	6.31 (0.75)
6	½	0	½	0	0.99 (0.03)	7.79 (0.85)
7	½	0	0	½	1.16 (0.02)	5.09 (0.14)
8	0	½	½	0	0.87 (0.02)	2.82 (0.10)
9	0	½	0	½	0.68 (0.02)	3.50 (0.20)
10	0	0	½	½	0.68 (0.02)	3.50 (0.20)
11	⅓	⅓	⅓	0	0.90 (0.01)	7.03 (0.22)
12	⅓	⅓	0	⅓	1.00 (0.01)	5.69 (0.19)
13	⅓	0	⅓	⅓	0.88 (0.02)	7.89 (0.29)
14	0	⅓	⅓	⅓	0.72 (0.02)	2.69 (0.14)
15	¼	¼	¼	¼	0.80 (0.02)	7.51 (0.78)

$$\frac{M_t}{M_\infty} = kt^N \tag{7}$$

where t is the release time, M_t/M_∞ is the fraction of the drug released at time t, k is the proportionality rate constant, and N is the release exponent. The dissolution half-times were calculated by

$$T_{0.5} = \sqrt[N]{\frac{0.5}{k}} \tag{8}$$

Three replicates were generated for each formulation. The resulting experimental data, summarized in Table I, were used to train the ANN and to estimate the coefficients in the RSM regression models. The eight validation formulations are shown in Table II. These formulations consisted of all four polymers and were used to evaluate the predictive capabilities of ANN and RSM models.

Artificial Neural Network Analysis

The neural network simulator, Nets, was used in this study (20). This program requires an ANSI-C environment and was run on an IBM compatible personal computer (BSR 386SX/16). The ANN architecture was constructed by setting the number of nodes in the input, output, and hidden layers. For the system studied, four formulations variables corresponding to different levels of HEC, HPC, HPMC, and CMC, and the two response variables, N and $T_{0.5}$ constitute the input and output layers, respectively. The hidden layer consisted of eight nodes. A learning rate of 0.25 and a momentum factor of 0.9 were used. To train the ANN, all input/output data were scaled such that all data points were within the 0.1 to 0.9 range to conform with the Nets software format. Other commercial programs, such as the Neural Works Professional II (Neural Ware Inc., Swickley, PA), automatically scale the input/output data and also allow for the selection of alternative transform functions and learning rules. The predicted values were converted back to the original scale before comparing the results with RSM.

Response Surface Analysis

The experimental design points in the simplex space occur at the four permutations of the (1,0,0,0) pure components, the six permutations of the (½,½,0,0) binary mixtures, the four permutations of the (⅓,⅓,⅓,0) ternary mixtures, and the overall centroid (¼,¼,¼,¼) (Fig. 3). The polynomial used to approximate the response surface is the special quartic equation:

$$R = \sum_{i=1}^{4} \lambda_i \theta_i + \sum\sum_{i<j}^{4} \lambda_{ij} \theta_i \theta_j + \sum\sum\sum_{i<j<k}^{4} \lambda_{ijk} \theta_i \theta_j \theta_k$$

Table II. Validation Experiments

No.	CMC	HPMC	HEC	HPC	Expt. N	ANN [N][a]	RSM [N]	Expt. $T_{0.5}$	ANN [$T_{0.5}$]	RSM [$T_{0.5}$]
1	0.50	0.21	0.21	0.08	1.08 (0.02)[b]	1.11	1.00	5.73 (0.40)	6.36	7.89
2	0.50	0.08	0.21	0.21	1.14 (0.03)	1.12	1.01	6.25 (0.60)	6.21	8.05
3	0.50	0.17	0.17	0.17	1.09 (0.02)	1.14	1.02	5.72 (0.53)	6.00	8.04
4	0.50	0.21	0.08	0.21	1.22 (0.04)	1.18	1.08	5.41 (0.59)	5.44	7.14
5	0.75	0.08	0.08	0.08	1.31 (0.04)	1.39	1.30	4.38 (0.43)	4.10	5.83
6	0.75	0.05	0.10	0.10	1.34 (0.04)	1.39	1.29	4.51 (0.52)	4.06	6.06
7	0.75	0.10	0.05	0.10	1.39 (0.06)	1.41	1.32	4.03 (0.59)	3.90	5.71
8	0.75	0.10	0.10	0.05	1.30 (0.03)	1.36	1.28	439. (0.39)	4.39	6.15

[a] Brackets indicate predicted response.
[b] Standard deviations in parentheses.

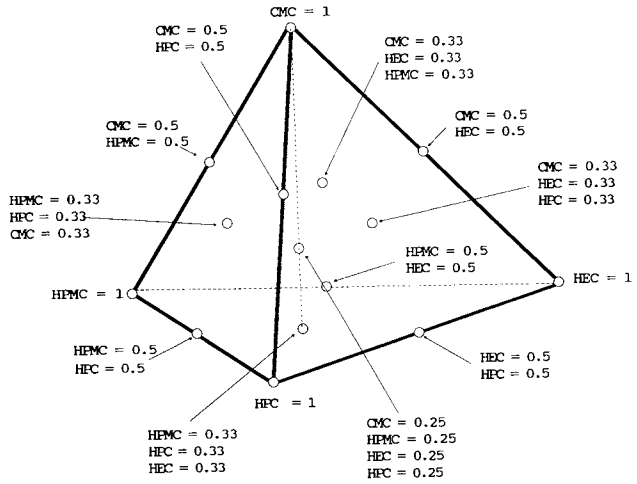

Fig. 3. Graphical representation of the four-component simplex mixture experimental design.

$$+ \lambda_{ijkl}\theta_i\theta_j\theta_k\theta_l \quad (9)$$

where R is the predicted response, Θ_i, Θ_{ij}, Θ_{ijk}, and Θ_{ijkl} are the polymer mass fractions, and λ_i, λ_{ij}, λ_{ijk}, and λ_{ijkl} are the RSM coefficients. The coefficients were estimated with the ECHIP program (22).

RESULTS AND DISCUSSION

The *in vitro* release data were well characterized by Eq. (6) (r^2: 0.975–0.999). Typical release profiles are shown in Fig. 4.

The ANN architecture depends on the problem under study. In this example, the input layer contained four nodes (four formulation variables) and the output layer contained two nodes (two response variables). The number of nodes in the hidden layer were selected by training the network with 4, 6, 8, 10, 12, and 14 nodes. A network containing eight hidden layer nodes gave the lowest residual sum of squared error for the training set.

The RSM regression coefficients for both N and $T_{0.5}$ and their respective correlation coefficients are listed in Table III. The response values from the validation experiments were predicted and compared with the experimental values and are summarized as the residual plots shown in Figs. 5 and 6. The mean sum of squared residuals (MSSRs) for the validation set (mean data) were 0.002 and 0.007 for N and 0.10 and 2.97 for $T_{0.5}$ by ANN and RSM, respectively.

Table III. Summary of Regression Results for RSM

Coefficient	N	$T_{0.5}$
λ_1	1.58	2.99
λ_2	0.67	2.05
λ_3	0.67	3.06
λ_4	0.55	3.58
λ_{12}	−0.16*	15.2
λ_{13}	−0.87	19.0
λ_{14}	0.37	7.2
λ_{23}	0.79	1.04*
λ_{24}	0.28	1.5*
λ_{34}	0.27	−0.05*
λ_{123}	−1.31	11.1*
λ_{124}	0.32*	4.41*
λ_{134}	−0.83*	47.8
λ_{234}	−1.42	−5.69*
λ_{1234}	−16.1	242
r^2	0.989	0.976

* Not significant at $P = 0.05$ level.

The formulations used in the validation set were chosen to reflect a high degree of system complexity within the simplex space. These formulations occupy regions that were not effectively mapped by the mixture design. Reevaluating the regression coefficients for N and $T_{0.5}$ by adding formulations 3 and 5 from Table II to formulations of the original mixture design reduced the MSSR from 2.97 to 1.50 for the remaining six validation points for $T_{0.5}$. However, no appreciable improvement was observed for the MSSR for N. These two formulations were selected because they lie along the axis perpendicular to the HPC, HPMC, and HEC plane in the direction of increasing CMC. Therefore, the modified train-

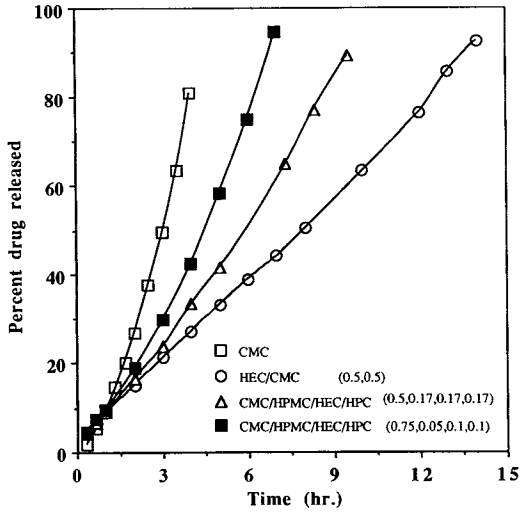

Fig. 4. Typical *in vitro* drug release profiles.

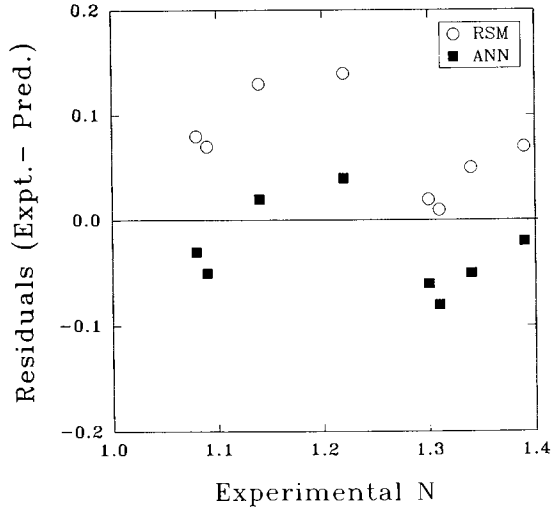

Fig. 5. Residual plot for N in the validation set (mean experimental − predicted values).

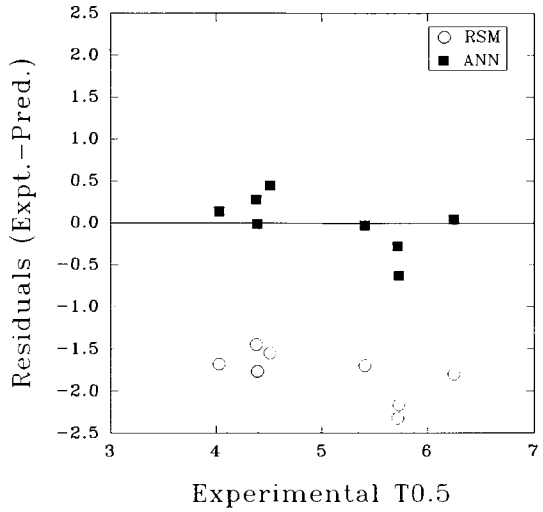

Fig. 6. Residual plot for $T_{0.5}$ in the validation set (mean experimental − predicted values).

ing set contained three interior points at three different CMC levels (0.25, 0.5, and 0.75). Even with this modification for RSM analysis, the ANN still gave better predictions than RSM for the validation set.

RSM requires the specification of a polynomial function to be regressed. The number of terms in the polynomial is limited to the number of distinct experimental design points. In some cases, the response variable may be highly nonlinear or discontinuous; therefore, use of a polynomial approximation may not provide an adequate description of the response surface; whereas, ANNs have demonstrated an ability to handle such problems (9). In addition, historic or literature data may also be used for ANN training.

One distinct advantage of RSM analysis is that the response surface is described by a continuous function. A visual representation of the surface is easily mapped as a contour or three-dimensional (3-D) plot. Also, the relationship of the rate of change of the response variable with a design variable is easily achieved by plotting the first derivative of the response variable as a function of the design variable. Because ANNs use an internal model, extensive simulations are required to generate contour or 3-D plots. Also, since the ANN output is a discrete set of points, numerical differentiation of the output data points is required to evaluate the rate of change of the response variable with its respective design variables. A distinct disadvantage of the RSM technique occurs when the system under study requires a large number of input and output variables. Selection of an appropriate polynomial equation can be extremely cumbersome because each response variable requires its own individual polynomial equation. Also, a large number of input variables may lead to a polynomial with a vast number of coefficients. While this may not present a computational problem for computer efficient estimation, subsequent evaluation can be tedious. In this situation, some investigators may prefer the internal model of the ANN analysis. We believe that ANN is a powerful tool, with potential for application in pharmaceutical product development and other areas of pharmacy.

REFERENCES

1. G. E. Box and N. R. Draper. *Empirical Model-Building and Response Surfaces,* John Wiley, New York, 1986.
2. D. E. Fonner, Jr., R. Buck, and G. S. Banker. Mathematical optimization techniques in drug product design and process analysis. *J. Pharm. Sci.* 59:1587–1596 (1970).
3. J. B. Schwartz, J. R. Flamholz, and R. H. Press. Computer optimization of pharmaceutical formulations. I. General procedure. *J. Pharm. Sci.* 62:1165–1170 (1973).
4. P. L. Gound. Optimization methods for the development of dosage forms. *Int. J. Pharm. Tech. Prod. Manuf.* 5:19–24 (1984).
5. M. R. Harris, J. B. Schwartz, and J. W. McGinity. Optimization of a slow-release formulation containing sodium sulfathiazole and a montmorillonite clay. *Drug Dev. Ind. Pharm.* 11:1089–1110 (1985).
6. R. M. Franz, J. A. Sytsma, B. P. Smith, and L. J. Lucisano. *In vitro* evaluation of a mixed polymeric sustained release using response surface methodology. *J. Control. Release* 5:159–172 (1987).
7. A. D. Johnson, V. L. Anderson, and G. E. Peck. A statistical approach for the development of an oral controlled release matrix tablet. *Pharm. Res.* 7:1092–1097 (1990).
8. M. J. Jozwiakowski, D. M. Jones, and R. M. Franz. Characterization of a hot-melt fluid bed coating process for fine granules. *Pharm. Res.* 7:1119–1126 (1990).
9. P. Bhagat. An introduction to neural nets. *Chem. Eng. Prog.* 86:55–61 (1990).
10. A. S. Hussain, R. Johnson, P. Shivanand, and O. Sprockel. Zero-order drug release from hydrophilic matrix capsule. *Pharm. Res.* 7:S159 (1990).
11. M. J. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* 5:115–133 (1943).
12. D. O. Hebb. *The Organization of Behavior,* John Wiley & Sons, New York, 1949.
13. R. Rosenblatt. *Principles of Neurodynamics,* Spartan Books, Washington, DC, 1962.
14. J. J. Hopfield and D. W. Tank. Computing with neural circuits: A model. *Science* 233:625–633 (1986).
15. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. J. Maclellan (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1. Foundations,* MIT Press, Cambridge, MA, 1986.
16. T. Kohonen. *Self-Organization and Associative Memory,* 2nd ed. Springer-Verlag, Berlin, 1988.
17. S. Grossberg. *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control,* Reidel, Amsterdam, 1982.
18. *SAS: NLIN Program,* SAS Institute Inc., Cary, NC.
19. R. W. Korsmeyer, R. Gurny, E. Doelker, P. Buri, and N. A. Peppas. Mechanism of solute release from process hydrophilic polymers. *Int. J. Pharm.* 15:25–35 (1983).
20. P. T. Baffes. Nets. *COSMIC,* Athens, GA 30602, 1989.
21. J. A. Cornell. *Experiments with Mixtures,* John Wiley & Sons, New York, 1981.
22. *ECHIP,* Echip Inc., Hockessin, DE.