APPLICATION OF NEURAL COMPUTING IN PHARMACEUTICAL PRODUCT DEVELOPMENT: COMPUTER AIDED FORMULATION DESIGN

Ajaz S. Hussain [1,3], Padmaja Shivanand [1] and Robert D. Johnson [1,2]

[1]Division of Pharmaceutics and Drug Delivery Systems
College of Pharmacy, University of Cincinnati
Cincinnati, Ohio 45267-0004.
[2] Present address:
Rhone-Poulenc Rorer Central Research
Collegeville, PA 19426
[3] To whom correspondence should be addressed.

ABSTRACT

Most pharmaceutical products are complex systems designed to meet several compendial or other performance standards simultaneously. Ideal or 'optimum' product composition and the manufacturing process variables are generally established after extensive experimentation. Artificial Neural Networks are pattern recognition tools that allow the development of 'expert' systems without having to write computer programs. With this technology it may be possible to develop formulation 'expert' systems to predict the formulation composition and the manufacturing process conditions necessary to achieve the desired performance standards. This report introduces the concept of a formulation expert system to predict the *in vitro* drug release profile from hydrophilic matrix tablets. Formulation expert systems or Computer Aided Formulation Design has the potential to reduce the time and cost of the product development process.

INTRODUCTION

The process of pharmaceutical product development generally begins after the desired pharmacologic activity of a new chemical entity has been established. The selection of an appropriate dosage form or delivery system for the drug depends not only on its therapeutic use but also on its physico-chemical and biopharmaceutical / pharmacokinetic properties. Most drug delivery systems have to meet complex, and often conflicting (for example hardness and dissolution rate of a tablet) performance criteria. The multivariable and complex nature of dosage forms hinders the development of quantitative relationships between formulation variables and product performance. In absence of such relationships the formulator has no choice but to explore the terrains of formulation variables through extensive experimentation. Over the last four decades significant progress has been made in replacing the 'art' in formulation development with sound scientific principles. In addition, the use of statistical experimental designs and optimization techniques have been adopted. However, the formulator's ability to predict the formulation variables (and their levels), that would be necessary to achieve the selected performance criteria for a (new) drug is limited. It is widely accepted that with experience, formulators develop the 'art' or expertise in formulation design that allows them to guess starting variable levels close to the 'optimum' levels. The formulation 'art' is the ability to recognize patterns of product performance and formulation variables.

The formulation 'art' may be made easily accessible by; (i) developing rule based (if-then rules) expert systems, or (ii) by using pattern recognition tools such as artificial neural networks (ANN) (1). In the former case a (human) formulation "expert" defines the decision making process by developing an algorithm of "if--then" rules. In the latter approach empirical models are developed from historical data (previous formulation experience).

Artificial neural networks (ANN) are a set of computational paradigms designed to mimic the functionality of the human brain. Although these models are not as sophisticated and lack the computational power of the brain, several impressive applications of this technology have been demonstrated. These applications fall under two major categories: (i) 'expert' system development, and (ii) pattern-recognition and/or mapping. In the last three years several applications of ANN's have been reported in the fields of chemistry and chemical engineering. Recently, several applications in pharmacy have also been proposed (2-6). Several excellent reviews and books are now available on this subject and the reader is encouraged to consult these for a more in-depth treatment of this topic (7-10). The purpose of this report is to explore potential applications of ANN's in pharmaceutical product development. The delta backpropagation training algorithm is the most widely applied ANN paradigm, therefore, is the focus of this report.

Overview of ANN Technology

An artificial neural network is a mathematical model that emulates a biological neural network. It consists of a number of processing elements (PE) or artificial neurons, interconnected and arranged in different ways to form a network. PE's are capable of receiving input-signals, processing these and producing output-signals. These signals are transferred (in one direction) by the network connections (synapses). A weight (synaptic strength) is associated with each connection. The arrangement of PE's in a network is referred to as the network topology or architecture.

Most frequently used ANN's possess multi-layer topology (Figure 1). The 'input' layer consists of PE's that receive inputs for the network and the 'output' layer produces the network's output . A number of layers may be arranged between the input and output layers. These intermediate or hidden layers perform the 'mapping' operations. Generally all PE's in the input layers are connected to the PE's in the 'hidden' layers which in turn are connected to output layer PE's. PE's within each layer are usually not connected to each other. A 'bias' PE is usually included in the input layer, with a constant input value of 1 and is connected to the PE's in the hidden and output layers.

The ANN may be a 'feedforward' or a 'feedback' network. In the 'feedforward' networks, information flows in the direction of the input to hidden to output layers only. 'Feedback' or 'Recurrent' networks may have connections that allow information to return from the output to hidden and/or the input layers. In 'auto-associative' (in contrast to 'hetero-associative') networks, input and output are identical, and these networks are generally used for data compression or as an error 'filter'.

Learning/Training

Learning is the process by which weights are adjusted to allow mapping of input - output relationships. This is accomplished by a learning algorithm. More than 100 different algorithms have been proposed. The learning procedure may be 'supervised' or 'unsupervised'. In 'supervised' learning, the network is presented with a set of inputs and the desired outputs, and in 'unsupervised' learning, only the input data set is presented to the network which then learns to categorize (cluster) the input patterns. Learning procedure is generally a variation of one of the following:

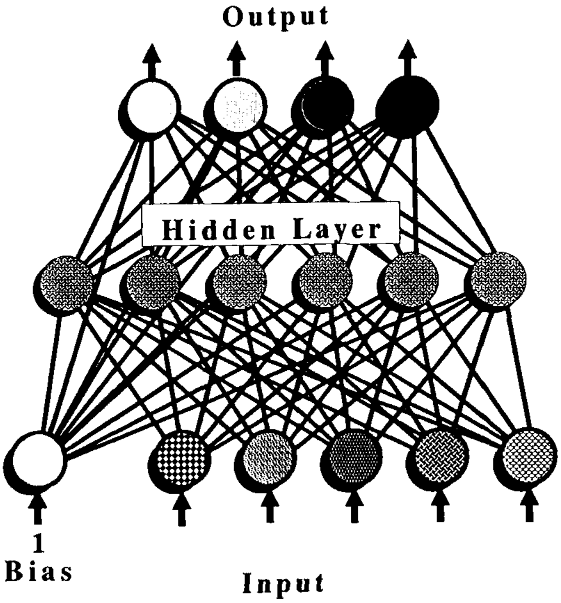

FIGURE 1
Typical ANN architecture (feed forward network)

(i) *Hebbian learning:* a connection weight in the input path of a PE is incremented if both the input and the desired output is high;
(ii) *Delta rule:* is based on reducing the error between the actual output of a PE and its desired output by modifying incoming connection weights; and
(iii) *Competitive learning* : where PE's compete among themselves and the one which yields the strongest response to a given input modifies itself.

After training, the connection weights become the memory units. Certain memories are associative, in that, if the trained ANN is presented with a partial input vector it will choose the closest match in memory to that input, and generate an output which corresponds to the full input. The nature of ANN memory leads to reasonable ANN response when presented with incomplete, noisy, or previously unseen (within its domain) input. This property is referred to as "generalization".

The Delta Backpropagation Network (DBN)

The backpropagation algorithm is one of the most widely used training algorithm for ANN. It appears to have been developed, independently, by a number of scientists (11-13). It is a supervised training algorithm that uses the delta learning rule to train multilayer feedforward ANN's. The name 'backpropagation' is derived from the process of propagating the error information backward from the output nodes to the hidden nodes. This paradigm is similar to the statistical non-linear regression analysis and could be considered as a statistical modeling tool. The training procedure is as follows:

The available data (input/output) is divided into two groups: the training and the test sets. The number of PE's in the input and output layers is defined by the problem. The number of hidden layers and the number of PE's have to be determined for the specific problem. The number of PE's in the hidden layer is related to the complexity of the problem. Too few PE's will result in poor mapping and too many PE's will slow down training and may lead to the problem of 'over-fitting' or 'memorization', which usually results in poor generalization. Many investigators have adopted techniques such as the Akaike's Information Criterion or the statistical F-test to estimate the optimum number of PE's (14).

Training begins with a random set of weights (usually in -0.1 to 0.1 range) and proceeds iteratively. Each iteration is called an epoch. In each epoch, weights are adjusted in the direction that results in reduced delta error (the difference between the current outputs and the target or desired outputs). For each example in the training set there is a forward pass to estimate the network output, followed by a backward pass to determine the adjustments in the weights to minimize the error. Similar to nonlinear regression analysis, a 'gradient decent' method is used to minimize the mean squared error. Generally the training examples are presented in a random fashion and the weights are updated after one (or a fraction of) epoch. A brief summary of the mathematical operations performed by the PE is given in the Appendix.

The DBN has been widely used in engineering, basic and applied sciences and in business fields. It is a tool that supplements the existing statistical and symbolic language based programmed expert systems. It has been shown to be superior to multiple-linear or polynomial regression analysis for analysis of nonlinear functions and has the advantage of being free from the assumptions of (normal) distribution, linear superposition and orthogonal functions (7). However, it is poor for situations requiring precise calculations or linear problems. Recently, several modifications have been proposed to overcome these deficiencies (9, 15,16).

Pharmaceutical Applications

A summary of pharmaceutical and relevant chemical applications is provided below to demonstrate the potential applications of ANN's. Readers interested in other chemical applications of ANN's are referred to the work of Zupan and Gasteiger (10).

a) *Determining spectra/structure relationships.* The problems of establishing reliable correlations between different types of spectra (IR, ^{1}H-NMR, ^{13}C-NMR, etc.) and the chemical structure or its fragments have been attempted with encouraging results (17-19).

b) *Prediction of the secondary and tertiary structure of proteins* from the sequence of amino acids (20-21).

c) *Quantitative structure-activity/property relationships (QSAR).* The 'pattern recognition' and 'feature extraction' ability of the neural network has successfully been utilized in drug design and development. Several QSAR studies utilizing ANN have been reported in literature (22-25).

d) *Pharmaceutical product development / optimization.* The potential use of ANN in product formulation development has been demonstrated in a study involving the design of a controlled release hydrophilic matrix capsule containing blends of anionic and nonionic cellulose ether polymers (4). Additional reports comparing ANN with the statistical response surface methodology have also been presented (26-28).

e) *Pharmacokinetics.* Utility of ANN for prediction of pharmacokinetic parameters of drugs in humans (from a combination of animal data and chemical structure) (6), pharmacokinetic-pharmacodynamic modeling (5) and prediction of pharmacokinetic parameters or blood levels from clinical data has also been described (3, 29).

Computer Aided Formulation Design (CAFD)

A potential application of the ANN technology is in the development of formulation 'expert' system or 'Computer Aided Formulation Design' concept (30). The purpose of CAFD is to: (1) predict the formulation/process conditions that will allow the resulting product to meet the desired performance criteria, (2) allow the formulator to carry out simulation studies, (3) to serve as "institutional" memory so as to allow a new formulator to develop expertise in the selection of excipient(s) and the use of machinery for various unit operations, and (4) to reduce the time and the cost of the product development process.

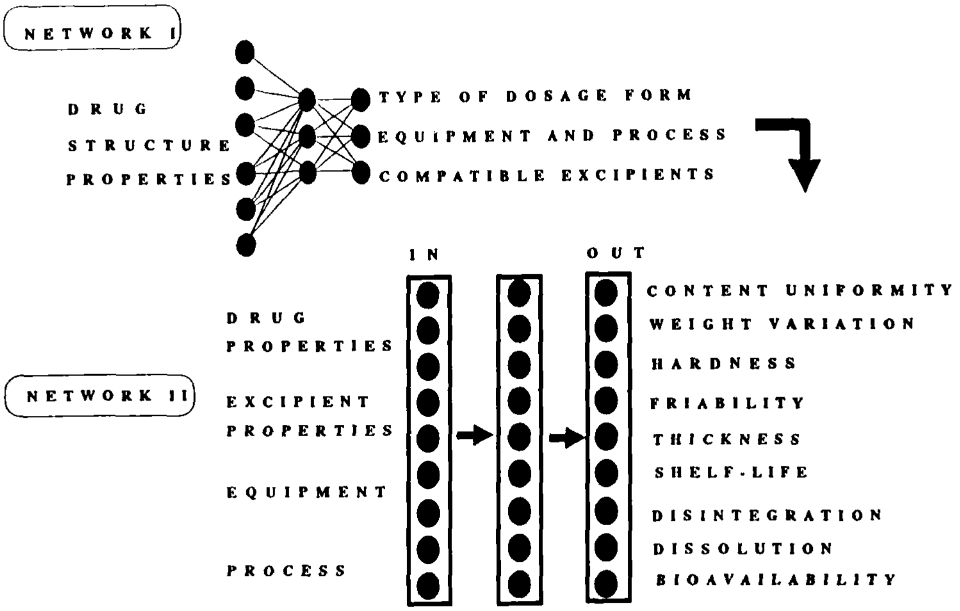

FIGURE 2
Hypothetical Unit-Operations based ANN structure

Use of ANN's for CAFD assumes the availability of information from previous developmental projects. This historical data may be combined with the existing knowledge to develop an ANN capable of predicting the performance of products containing new drug candidates from drug and excipient properties and formulation / process conditions. If similar dosage forms are being developed for new drugs then the ANN structure would be rather simple (Figure 1) consisting of input PE's to bring information regarding the drug, excipient(s) and the process in the network and the output layer consisting of PE's to predict the product performance. It is also conceivable that this approach could be extended to allow the development of new dosage forms (variable process and excipient conditions) by combining smaller ANN's that posses the information regarding each applicable unit-operation and functionality/property of excipient(s) and the drugs. A hypothetical 'Unit-Operations' based ANN is shown in Figure 2.

The objective of this study was to evaluate the feasibility of the proposed CAFD approach using a sustained release hydrophilic matrix tablet as the model formulation.

MATERIALS AND METHODS

Sodium salts of diclofenac, naproxen and salicylic acid; HCl salts of diphenhydramine, phenylpropanolamine, tetrahydroaminoacridine, propranolol; acetaminophen, caffeine, theophylline and quinidine sulfate (Sigma Chemicals, St. Louis, MO) were formulated into tablets using hydroxy propyl cellulose (Klucel®, Aqualon Co.). Three different polymer

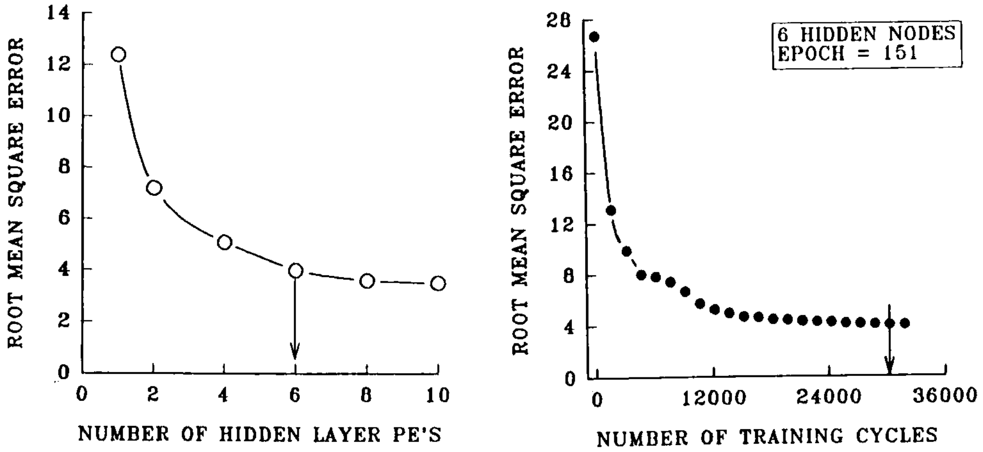

FIGURE 3

Selection of number of hidden nodes and training cycles based on Root Mean Square Error

viscosity grades, characterized by their hydration times, namely EXF (hydration time=15 sec), GF (hydration time=50 sec) and HXF (hydration time=100 sec) were studied. Klucel®-GF was passed through a 100 mesh sieve to obtain a particle size distribution similar to the EXF and HXF grades. Sieved Klucel®-GF was used for this study and in this report will be referred to as GXF grade.

Process variables: Binary mixtures of drug and polymer were directly compressed using a single punch machine with standard biconvex tooling (Dia=11 mm) to a weight of 350 mg and hardness of 10-15 kp

Formulation variables: Three different drug to polymer ratios (D/P) were investigated: 1:3; 1:1; and 3:1.

Determination of release profile: The *in vitro* release profile was obtained in deionized water at 37°C using the USP basket apparatus at a stirring speed of 100 rpm. Samples were withdrawn at hourly intervals and analyzed by UV spectrophotometry at their respective λ_{max}.

Determination of intrinsic dissolution rates : Fifty mg of the drug was compressed at 2000 lbs using 6 mm flat faced punches on a hydraulic press. The 6 mm tablet was placed in the center of the lower punch in a 12 mm punch and die arrangement and 300 mg of a hydrophobic, non swellable polymer Cellulose Acetate Butyrate 500 (Scientific Polymer Products, Ontario, NY) was placed over the tablet. The upper punch was inserted and the tablet recompressed at 800 lbs. The compressed tablet was affixed to an aluminum petriplate with the exposed surface facing outward. The tablet assembly was inserted into the dissolution flask with the exposed surface facing upwards. The dissolution of the compressed tablet was studied in a Hanson dissolution apparatus (Hanson Research Co., Northridge, CA) using deionized water as the media maintained at 37°C at a stirring speed of 100 rpm. Samples were withdrawn using the Hanson automatic sampler.

ANN system development : A nonlinear feed forward neural network, based on the extended delta bar delta (EDBD) learning rule was used to recognize the relationships between the drug, formulation properties and the *in vitro* drug release profile. The EDBD algorithm is an

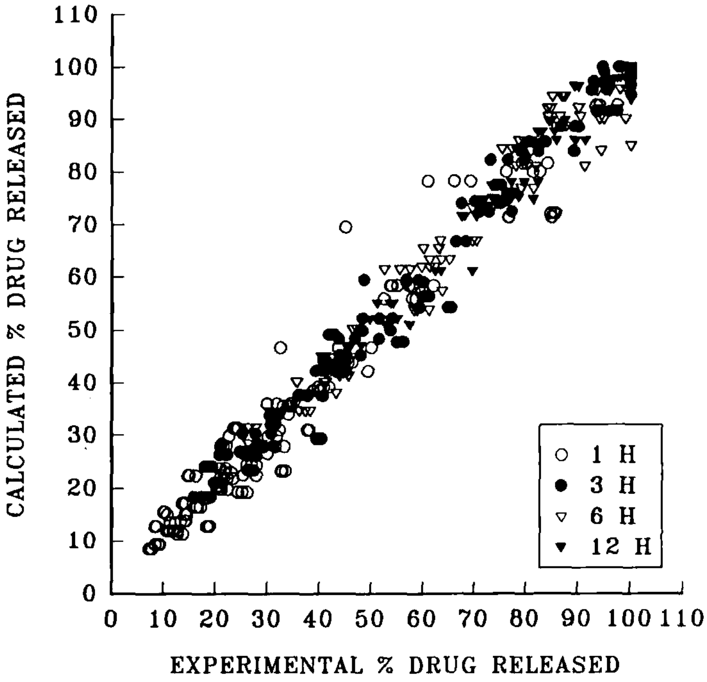

FIGURE 4
ANN training results

enhanced DBN algorithm which decreases the learning time by using the momentum (see Appendix) in a heuristic fashion (31). The network consisted of six input nodes, four output nodes and six hidden nodes. The input nodes used to define the drug were its molecular weight, intrinsic dissolution rate, pKa and the salt type, while the other two input nodes were the drug to polymer ratio and the hydration rate of the polymer. The network output nodes were the % drug released at 1, 3, 6 and 12 hours (defined as the 'release window'). The network was trained for 200 times the epoch (number of training examples), using six hidden nodes. The selection of the number of hidden nodes and the number of training cycles were based on minimizing the root mean square error as depicted in Figure 3. For this study, tablets were compressed by a direct compression method, however, for applications involving other process or manufacturing conditions (eg. wet granulation), relevant variables may be incorporated as the network input nodes.

RESULTS AND DISCUSSION

Figure 4 depicts the ANN training results generated using the entire data set. The developed network was successful in learning the existing relationship between the input variables and the output variables ($r > 0.97$; root mean square error about 5 %). The developed network was validated by evaluating its ability to predict the release profile of drugs not used in the training set and for formulations differing in D/P ratio. The validation step was performed by the 'one-out' method, i.e., removing the relevant data from the training set for a

TABLE 1: RMS values for training (first number) and test (second number) sets

Drug; (D/P)	EXF	GXF	HXF
Diclofenac Sodium (1:1)	4.6, 10.3	5.1, 2.7	3.9, 11.7
Naproxen Sodium (1:1)	1.2, 3.0	5.0, 5.9	4.3, 5.4
Sodium salicylate (1:1)	1.7, 1.5	1.6, 4.4	3.6, 3.3
Quinidine Sulfate (1:1)	2.6, 11.2	1.6, 13.2	2.2, 30.4
Caffeine (1:1)	2.1, 6.0	7.0, 9.2	2.8, 4.7
Theophylline (1:1)	1.0, 1.9	2.4, 3.1	3.0, 5.2
Theophylline (1:3)	2.5, 4.2	1.7, 5.2	2.6, 2.2
Theophylline (3:1)	1.3, 6.6	3.1, 11.6	4.3, 6.0
Acetaminophen (1:1)	1.8, 3.0	2.6, 12.9	2.8, 9.3
Acetaminophen (1:3)	3.6, 12.0	3.8, 4.5	2.9, 3.2
Acetaminophen (3:1)	12.4, 11.7	5.3, 4.3	2.3, 4.9
Diphenhydramine HCl (1:1)	6.7, 3.7	5.6, 2.6	6.6, 2.4
Propranolol HCl (1:1)	2.1, 2.8	4.0, 2.2	4.7, 6.4
Tetrahydroaminoacridine HCl (1:1)	3.0, 5.7	8.4, 10.3	1.6, 6.1
Phenylpropanolamine HCl (1:1)	6.8, 6.8	3.0, 4.9	1.3, 5.4
Phenylpropanolamine HCl (1:3)	5.4, 23.6	6.4, 9.7	3.3, 20.1
Phenylpropanolamine HCl (3:1)	1.8, 2.3	2.9, 6.2	2.0, 7.1

particular drug or a D/P ratio and retraining the network. The retrained network was then used to predict the release profile for the removed drug or formulation. This was repeated until all the drugs were removed once from the network. Table 1 summarizes the prediction error (RMS) for the training and validation procedures. The network was able to predict the *in vitro* release profile of all the sodium salts, HCl salts, acetaminophen, caffeine and theophylline (drug to polymer ratio, 1:1). The prediction error (RMS) for the formulations were found to be acceptable (see Table 1). The network was also able to generalize, with a reasonable prediction error, the release profile for theophylline and acetaminophen formulations at different drug to polymer ratios. Figure 5 is a representative comparison of the predicted and observed *in vitro* release profile for theophylline.

However, the prediction error (RMS) for quinidine sulfate and phenylpropanolamine HCl at a D/P ratio of 1:3 (EXF and HXF) were high (see Table 1). Quinidine sulfate was the only divalent example in the training set, and therefore when used as a test drug, the predictions were poor (outside the 'domain'). The reasons for the poor prediction of the phenylpropanolamine HCl formulation are not clear.

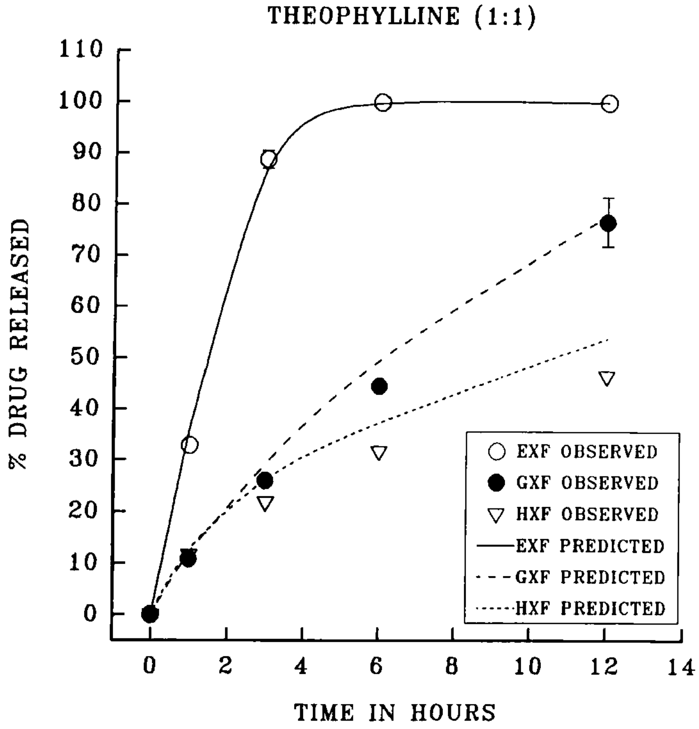

FIGURE 5
In vitro release profile prediction for theophylline

Figure 6 is a plot of the intrinsic dissolution rate vs. % drug released at 1 hour from EXF polymer matrices. The plot indicates three distinct data sets: the sodium salts, the HCl salts and neutral compounds. As discussed above, quinidine sulfate does not fit this pattern. Similar groups were also observed in the plots for the other polymers. These results indicate that the developed network was capable of recognizing such underlying patterns.

The selected formulation examples and the test conditions are obviously too simple to be considered as actual dosage form. However, they serve to illustrate the potential advantages and limitations of this approach. The developed ANN was able to differentiate between the salt types (for example; Na ion is known to interfere with the hydration of the cellulose polymer (32) and this effect appears to be more pronounced than with HCl or nonionic drugs), polymer grades and the drug to polymer ratios and successfully predicted the release profile of most drugs when they were used in the test set. When predictions outside its domain were attempted, its performance was poor. Obviously additional drugs (different salts to represent the Hofmeister and lyotropic series) would have to be included in the training set and additional formulation variables (lubricants, diluents), process conditions (granulation) and performance tests (release in simulated gastric and intestinal fluids, dissolution apparatus conditions, *in vivo* blood levels, etc.) to make this a useful 'expert' system. The relative ease with which ANN systems can be developed to handle complex relationships suggests that this technology will allow us to develop and exploit the CAFD approach to product development.

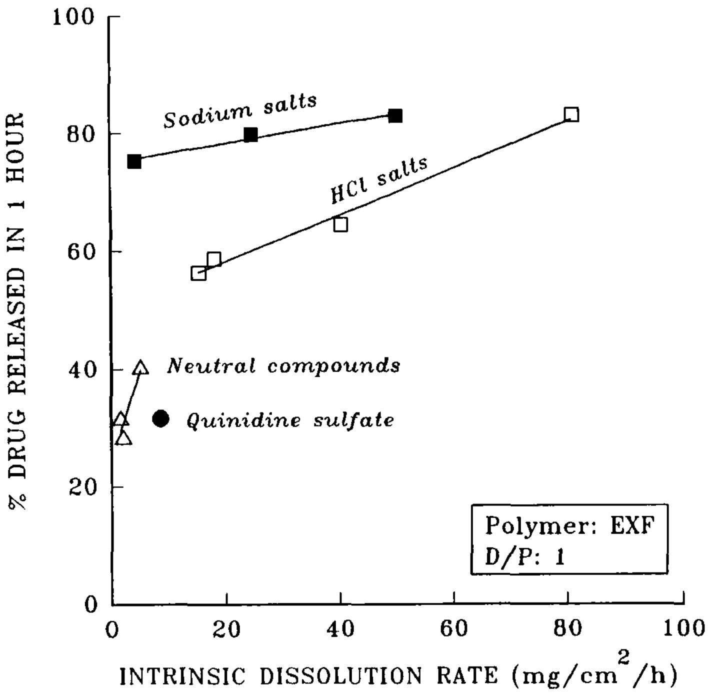

FIGURE 6
Relationship between the intrinsic dissolution rate of the drugs and their release from Klucel® EXF matrices.

CONCLUSIONS

The multi-variable nature of the product development process (and the traditional trial-and-error experimentation approach) offers considerable challenge for extracting useful 'knowledge' from the generated data. ANN's offer a way to improve knowledge extraction from historic data, and with proper selection of a training data set will allow development of formulation 'expert' systems that could serve as 'institutional memory' for predicting formulation and process conditions .

Unlike statistical methods, ANN's are independent of the type of distribution, linear superposition and orthogonal functions. However, application of statistical principles to ANN modeling maximize 'generalization' ability and minimize 'over-fitting' or 'memorization'. In future it is most likely that these two techniques will merge to overcome the deficiencies of both techniques. Compared to the "Artificial Intelligence" based expert systems ANN's allow development of 'expert' systems from data. In some situations (such as not enough data) the two approaches could be combined. Another advantage of ANN technology is that it serves as an approach to generate hypotheses which may be tested by other scientific methods. The 'black box' nature of ANN's can be reduced by development of 'Hierarchial' networks based on the existing scientific knowledge of the system and using ANNs as simulation tools to probe the nature of the input/output relationships. The rapid growth of ANN technology and its potential application suggests that it may play a significant role in pharmaceutical problem solving.

APPENDIX

Each PE performs two simple mathematical operations. The first operation is summation of products of weights and inputs (or outputs from PE's in the preceding layer) as described by eq. 1.

$$I_j^{[s]} = \sum_{i=1}^{n} \left(w_{ji}^{[s]} \cdot O_i^{[s-1]} \right) \qquad (1)$$

This sum is then transformed by a suitable nonlinear transform function to complete the second operation (eq. 2).

$$O_j^{[s]} = f\left(I_j^{[s]}\right) \qquad (2)$$

where, $O_i^{[s-1]}$ is the output of PE i from the preceding layer s-1, w_{ji} is the connection weight between PE i of layer s-1 and PE j of layer s, $I_j^{[s]}$ is the weighted sum of PE j in layer s, and $O_j^{[s]}$ represents the transform of the weighted sum. The two most common transform functions used are the (generic) "sigmoid" (eq. 3) and the hyperbolic tangent function (eq. 4).

$$f_{SG}\left(I_j^{[s]}\right) = \frac{1}{1 - e^{-I_j}} \qquad (3)$$

$$f_{HT}\left(I_j^{[s]}\right) = \frac{e^{I_j} - e^{-I_j}}{e^{I_j} + e^{-I_j}} \qquad (4)$$

During the forward pass, input data is propagated through the network using eq. (1) and (2) for each PE in conjunction with either eq. (3) or eq. (4) to generate the network output ($O_k^{[o]}$). During the reverse pass the connection weights between the hidden and output layer are modified by the delta rule (eq. 5).

$$\Delta w_{kj} = -\alpha\left(\frac{\partial E}{\partial w_{kj}^{[o]}}\right) \qquad (5)$$

where, α is the learning rate coefficient, E is the global error function for the entire network (scaled error), and $w_{kj}^{[o]}$ is the connection weight between PE j in the hidden layer (h) and PE k in the output layer (o). By application of the chain rule eq. (5) is transformed to:

$$\frac{\partial E}{\partial w_{kj}^{[o]}} = \left(\frac{\partial E}{\partial I_k^{[o]}}\right) \cdot \left(\frac{\partial I_k^{[o]}}{\partial w_{kj}^{[o]}}\right) \qquad (6)$$

and,

$$\frac{\partial E}{\partial w_{kj}^{[o]}} = \frac{\partial E}{\partial I_k^{[o]}} \cdot O_j^{[h]} \qquad (7)$$

where, $O_j^{[h]}$ is the output of PE j from hidden layer h.

There are two error functions; the global (network) error (eq. 8) and the scaled local errors associated with each individual PE .

$$E = 0.5 \cdot \left(\sum_{k=1}^{n} \left(T_k - O_k^{[o]} \right)^2 \right) \tag{8}$$

where T_k represents the desired/target output of PE k (output layer). The scaled local error ($e_k^{[o]}$) can be expressed as a function of the global error by :

$$e_k^{[o]} = -\frac{\partial E}{\partial I_k^{[o]}} \tag{9}$$

Application of the chain rule to eq. (9) results in:

$$e_k^{[o]} = -\left(\frac{\partial E}{\partial O_k^{[o]}} \right) \cdot \left(\frac{\partial O_k^{[o]}}{\partial I_k^{[o]}} \right) \tag{11}$$

and the partial derivatives, eq. (10), can be expressed as

$$\frac{\partial E}{\partial O_k^{[o]}} = \sum_{k=1}^{n} \left(T_k - O_k^{[o]} \right) \tag{10}$$

and

$$\frac{\partial O_k^{[o]}}{\partial I_k^{[o]}} = f'\left(I_k^{[o]} \right) \tag{12}$$

where, $f'(I_k^{[o]})$ is the first derivative of the nonlinear transfer function. Combining eq.. (10), (11), and (12);

$$e_k^{[o]} = f'\left(I_k^{[o]} \right) \cdot \left(\sum_{k=1}^{n} \left(T_k - O_k^{[o]} \right) \right) \tag{13}$$

The output layer weight change increment ($\Delta w_{kj}^{[o]}$) is obtained by combining eq. (5), (8), (9), and (13) to gives

$$\Delta w_{kj}^{[o]} = -\alpha \cdot f'\left(I_k^{[o]} \right) \cdot \left(\sum_{k=1}^{n} \left(T_k - O_k^{[o]} \right) \right) \cdot O_j^{[h]} \tag{14}$$

it follows logically, that

$$w_{kj}^{[o]}(n+1) = \Delta w_{kj}^{[o]} + w_{kj}^{[o]}(n) \tag{15}$$

where, $w_{kj}^{[o]}(n+1)$ and $w_{kj}^{[o]}(n)$ are the numerical values of the weights after n+1 and n iterationS,respectively. Now that the weights between the hidden layer and output layer PE's have been adjusted, the weights between the input and hidden layers need to be adjusted. However, there are no target values to directly calculate the scaled local errors for the hidden layer PE's. Therefore, the scaled local errors for the hidden layer PE's are approximated by back-propagating the scaled local errors for the output layer PE's. The scaled local errors associated with the hidden layer PE's, as in the case of the output layer PE's, are given as:

$$e_j^{[h]} = \frac{\partial E}{\partial I_j^{[h]}} \tag{16}$$

where, $e_j^{[h]}$ is the scaled local error of PE j in the hidden layer and $I_j^{[h]}$ is the weighted sum of the product of the scaled input data and the input layer/hidden layer connection weights. Similar to the mathematical manipulations for adjusting the output layer/hidden layer connection weights, the final form of the weight adjustment equation is obtained by repeated application of the chain rule.

A major problem with the standard delta rule is the selection of the magnitude of the learning rate coefficient. Large learning rate coefficients can prevent the network from achieving an optimum solution by giving rise to weight changes that are too large in magnitude. Conversely, too small of a learning rate coefficient leads to slow learning. A significant enhancement to the delta rule was the introduction of a momentum term to help stabilize and speed up the learning process. Inclusion of the momentum term modifies the delta rule such that a portion of the previous weight change is included in the current weight change to give

$$\Delta w_{ji}^{[s]} = \alpha \cdot e_j^{[s]} \cdot O_i^{[s-1]} + \beta \cdot \Delta w_{ji}^{[s]} \tag{17}$$

where β is the momentum coefficient.

Continued research in ANN is yielding new and improved algorithms such as the Radial Basis Functions Networks and Genetic algorithms. In addition procedures are being developed to optimize the network architecture and to combine ANNs with Fuzzy Logic.

ACKNOWLEDGEMENTS

Financial support to explore neural network application for formulation design was provided by the Aqualon Company (Wilmington, DE.). To be presented at the Pharmaceutical Product Development Symposium, Gifu Pharmaceutical University, September 1993, Gifu, Japan.

REFERENCES

1. E. Turban, " Expert Systems and Applied Artificial Intelligence". Macmillan Publishing Company, New York, 1992.
2. T. Aoyama, Y. Suzuki, and H. Ichikawa, Chem. Pharm. Bull., 37, 2558 (1989).
3. R. Shadmehr and D. Z. D'Argenio, Neural Computation., 2, 216 (1990).
4. A. S. Hussain, X. Yu and R. D. Johnson, Pharm Res., 8, 1248 (1991).
5. P. Veng-Pedersen and N. Modi, J. Pharmacokin. Biopharm., 20, 397 (1992).
6. A. S. Hussain, R. D. Johnson, N. Vachharajani and W. A. Ritschel : Pharm Res., 10, 466 (1993).

7. R. Hecht-Nielsen, " Neurocomputing". Addison-Wesley Publishing Company, Massachusetts, 1989.

8. M. Caudill, " AI EXPERT-Neural Network Primer". Miller Freeman Publications, San Fransisco, CA, 1990.

9. M. Smith, " Neural Networks for Statistical Modeling". Van Nostrand Reinhold, New York, 1993.

10. J. Zupan and J. Gasteiger, Analytica Chimica Acta., 248., 1 (1991).

11. P. Werbos, "Beyond Regression: New tools for prediction and analysis in the behavioral sciences". Doctoral Dissertation, Appl. Math., Harvard University, 1974.

12. D. B. Parker, Proc. of the Int. Conf. on Neural Networks, II, 593 (1987).

13. D. E. Rumelhart and J. L. McClelland., "Parallel Distributed Processing: Explorations in the Microstructure of Cognition I & II" MIT Press, Cambridge, MA, 1986.

14. M. A. Kramer, AIChE J., 37., 233 (1991).

15. T. R. Holcomb and M. Morari, Computers Chem Eng., 16., 393 (1992).

16. M. L. Mavrovouniotis and S. Chang, Computers Chem Eng, 16., 347 (1992).

17. B. Curry and D. E. Rumelhart, Tetrahedron Comp. Methodol., 4., 163 (1991).

18. J. U. Thomsen and B. Meyer, J. Magn. Res., 84., 212 (1989).

19. B. J. Withoff, S. P. Levine and S. A. Sterling, Anal. Chem., 62., 2709 (1990).

20. N. Qian and T. J. Sejnowski, J. Mol. Biol., 202., 865 (1988).

21. H. Bohr, et. al., FEBS Lett., 261., 43 (1990).

22. T. Aoyama, Y. Suzuki and H. Ichikawa, J. Med. Chem., 33., 2583 (1990).

23. D. W. Elrod and G. M. Maggiora, J. Chem. Inf. Comput. Sci., 30., 477 (1990).

24. N. Bodor, A. Harget and M-J. Huang, J. Am. Chem. Soc., 113., 9480 (1991).

25. A. S. Hussain, A. Kane, X. Yu. and R. Bronaugh, Pharm. Res. 9: S-191 (1992).

26. A. S. Hussain, R. D. Johnson and P. Shivanand, Pharm. Res. 9: S-165 (1992).

27. A. S. Hussain and T. D. Reynolds, The 53 International Congress of Pharmaceutical Sciences (FIP '93), Tokyo, 1993.

28. T. Gill and J. Shutt, Scientific Computation and Automation Magazine, September 1992.

29. D. P. Healy and A. S. Hussain. The 53 International Congress of Pharmaceutical Sciences (FIP '93), Tokyo, 1993.

30. A. S. Hussain, P. Shivanand and R. D. Johnson, Pharm. Res. 9: S-147 (1992).

31. A.A. Minai and Williams, K.D, International Joint Conference on Neural Networks. 1: 676 (1990)

32. Lapidus, H. and Lordi, N.G, J. Pharm. Sci. 55: 840 (1966)