An Alternative Approach Based on Artificial Neural Networks to Study Controlled Drug Release

MARCUS A.A. REIS, RUBÉN D. SINISTERRA, JADSON C. BELCHIOR

Departamento de Química—ICEx, Universidade Federal de Minas Gerais, Pampulha, (31.270-901) Belo Horizonte, MG—Brazil

Received 7 February 2003; revised 20 June 2003; accepted 19 August 2003

ABSTRACT: An alternative methodology based on artificial neural networks is proposed to be a complementary tool to other conventional methods to study controlled drug release. Two systems are used to test the approach; namely, hydrocortisone in a biodegradable matrix and rhodium (II) butyrate complexes in a bioceramic matrix. Two well-established mathematical models are used to simulate different release profiles as a function of fundamental properties; namely, diffusion coefficient (D), saturation solubility (C_s), drug loading (A), and the height of the device (h). The models were tested, and the results show that these fundamental properties can be predicted after learning the experimental or model data for controlled drug release systems. The neural network results obtained after the learning stage can be considered to quantitatively predict ideal experimental conditions. Overall, the proposed methodology was shown to be efficient for ideal experiments, with a relative average error of $<1\%$ in both tests. This approach can be useful for the experimental analysis to simulate and design efficient controlled drug-release systems. J Pharm Sci 93:418–430, 2004

Keywords: neural networks; controlled release; mathematical models

INTRODUCTION

Considerable attention has been focused on the development of new methodologies to study controlled-release systems mainly because of their great impact on modern therapeutics. Bioceramics or biodegradable polymers are used as matrices to deliver a wide range of drugs in an efficient way. However, analysis may be complicated by the occurrence of multicomponent transport processes, different kinds of matrices, composition, device geometries, drug loading, saturation solubility in the matrix, diffusion, swelling, polymer dissolution, and erosion,[1] and hence theoretical approaches are helpful in determining ideal parameters e.g. diffusion coefficients.[2] These quantities can be, in principle, predicted by theoretical models to reduce the number of necessary experiments. Thus, the models can be considered as useful strategies to simulate the release profile for controlled-release systems. However, as pointed out by Siepmann and Peppas,[3] mathematical approaches covering all the possible chemical and physical processes are not yet available.

Nevertheless, efficient models have been proposed, for example, by Fu et al.[2] and Higuchi,[4] to circumvent complex problems in drug delivery systems. The advantage of using these mathematical models is to allow the experimentalists to determine the drug release for tablets with a specific size, polymer, and unknown diffusion coefficient. In general, most of these models for drug delivery systems are based on the solution of Fick's second law of diffusion. However, these conventional approaches, as pointed out by Fu et al.,[2] have some limitations because all data points are treated with equal weight. Furthermore, the least-squares method may not be the

Correspondence to: Jadson C. Belchior (Telephone: 5531 3499 5775; Fax: 5531 3499 5700;
E-mail: jadson@carbono.qui.ufmg.br)

Journal of Pharmaceutical Sciences, Vol. 93, 418–430 (2004)

most appropriate to optimize the nonlinear problem, and generally Kummer's transformations[2] are used to improve the optimization. In general, as the number of parameters dependence increases, the solution based on least-squares fitting becomes more complex. Accordingly, alternative approaches should be studied that combine the advantages of conventional methods already established with those of other techniques, such as artificial neural networks (ANNs). The latter methodology has been applied, for example, to study metallic complexation[5] and to simulate the drug release profile in drug delivery systems.[6–15]

As discussed by Chen et al.,[13] a significant difference between the neural network model and least-squares methods is that ANNs can generalize the relationship between independent and dependent variables without specific mathematical functions. Nevertheless, both methodologies can be considered as complementary tools to tackle drug delivery systems. In fact, it has been shown in the literature that the ANN (multilayer perceptron) is a universal multivariate function approximator[16–18] that has a much better performance than classical polynomial adjusting methods in higher dimensions and for noisy input data.[19]

Thus, the aim of the present study was to analyze the application of ANNs as a complementary tool to study drug release. To address this problem we chose the correlation between physical and chemical properties; namely the diffusion coefficient (D), the saturation solubility (C_s), drug loading (A), the height of the device (h), and the drug fraction released. The fraction release of the hydrocortisone system, in which the release profile was determined up to 120 days, was selected as a model to test the approach.[2] In addition, the mathematical models proposed by Fu et al.[2] and Higuchi[4] were used in the actual study to simulate more experimental data.

To test the utility of the ANN model at an early time release, results for 5 and 10% Rhodium (II) butyrate antitumor agent loaded in bioceramics were taken from the literature.[20] Moreover, two new experiments, with 7.5 and 12.5% Rhodium (II) butyrate were carried out to have more experimental data.

THEORETICAL

Hydrocortisone Model System

Considering hydrocortisone as a model test, Fu et al.[2] developed a mathematical model based on the solution of the time-dependent diffusion equation, which will be detailed later. The model was originally applied to study the drug–polymer sustained-release pellets of different geometries ranging from a flat disk to a cylindrical rod. The cumulative release of hydrocortisone from three different matrices (i.e., polycarpolactone, ethylene-vinyl, and PVA pellets) was compared both experimentally and theoretically. The release of hydrocortisone from 40 mg of polycarpolactone was 62.9% in 126 days compared with the theoretical estimate of 63.9%. Similarly, using 40 mg of hydrocortisone PVA terpolymer pellets, the release values after 90 days were 9.85 and 9.65% for the experimental and theoretical results, respectively. Through a least-squares fit and the model proposed by Fu et al.,[2] the diffusion coefficient of the cumulative release of hydrocortisone in polycarpolactone ($D = 1.58 \times 10^{-10}$ cm^2 s^{-1}), ethylene-vinyl acetate ($D = 1.18 \times 10^{-11}$ cm^2 s^{-1}), and PVA terpolymer ($D = 4.31 \times 10^{-12}$ cm^2 s^{-1}) was calculated.

Theoretical Model

The diffusion of a drug from a pellet to its surrounding medium can be described by the time-dependent diffusion equation[2,21]

$$\frac{\partial C}{\partial t} = D\left[\frac{\partial^2 C}{\partial r^2} + \frac{1}{r}\frac{\partial C}{\partial r} + \frac{\partial^2 C}{\partial z^2}\right] \tag{1}$$

where D is the diffusion coefficient, C is the concentration at any time, and r and z are defined within the limits $0 \le r \le a$ and $-l \le z \le l$, respectively. The quantities a and l are, respectively, the radius and the length of the pellet. The solution of this partial differential equation can be written as

$$C = C_0 \psi(z) \phi(r, a) \tag{2}$$

where $\psi(z)$ and $\phi(r, a)$ are given by

$$\psi(z) = \frac{4}{\pi}\sum_{n=0}^{\infty}\frac{(-1)^n}{(2n+1)}\exp\left[-D(2n+1)^2\pi^2 t/4l^2\right] \times \cos\frac{(2n+1)\pi}{2l}z \tag{3}$$

and

$$\phi(r, a) = \frac{2}{a}\sum_{m=1}^{\infty}\exp\left(-D\alpha_m^2 t\right)\frac{J_0(r\alpha_m)}{\alpha_m J_1(\alpha\alpha_m)} \tag{4}$$

where $J_0(x)$ and $J_1(x)$ are Bessel functions of order zero and one, respectively. The summation in eq. 4

will depend on the number of terms needed to obtain convergence. To predict the drug fraction released as a function of time one can write[2]

$$F = \frac{M(t)}{M(\infty)} = 1 - \frac{8}{l^2\alpha^2}\sum_{m=1}^{\infty}\frac{\exp(-D\alpha_m^2 t)}{\alpha_m^2}\sum_{n=0}^{\infty}\frac{\exp(-D\beta_n^2 t)}{\beta_n^2} \quad (5)$$

where $\beta = (2n+1)\pi/(2l)$ and α_m are the roots of Bessel functions. In our previous work, seven terms were enough to provide satisfactory convergence.[20] In the case of an infinite slab, eq. (4) can be approximated to[2,21]

$$F = 1 - 8\sum_{n=0}^{\infty}\frac{\exp(-D(2n+1)^2\pi^2 t/4l^2)}{(2n+1)^2\pi^2} \quad (6)$$

where F defines the drug fraction released.

The numerical solution of eq. 5 for the particular case of fitting the diffusion coefficient can be written as[2]

$$D_{k+1} = D_k + \left[J_k^T J_k\right]^{-1} J_k^T (y - f_k) \quad (7)$$

where

$$J_k^T J_k = \sum_{i=1}^{N}\left[\frac{\partial f(t_j)}{\partial D}\right]_k^2 \quad (8)$$

and

$$J_k^T (y - f_k) = \sum_{j=1}^{N}\left[\frac{\partial f(t_j)}{\partial D}(y_j - f(t_j))\right]_k \quad (9)$$

where y and f_k define the experimental and theoretical drug fraction released, respectively. The Jacobian (J) matrix contains the derivatives of f with respect to the parameter D. As will be discussed later, this Jacobian matrix is conceptually quite different when compared with the solution based on the neural network approach. Details of diffusion models can be found elsewhere.[2,21,22]

The drug controlled release is dependent on several parameters, as previously pointed out, and the analysis is, in principle, done with many experiments. In many cases these experiments are expensive and the results are not accurate enough to achieve the correct values to define an ideal controlled release, although this strategy is generally used for experimental validation.

The alternative mathematical model proposed by Higuchi[4] to analyze the fraction of the hydrocortisone system released is written as

$$F = \left(\frac{8DC_s(A - C_s/2)t}{A^2h^2}\right)^{1/2} \quad (10)$$

where D is the diffusion coefficient, C_s is the saturation solubility of the drug in the matrix, A is the drug load, and h is the height of the device. Using eq. 10 and the parameters used by Fu et al.,[2] which were fitted to an experimental result of hydrocortisone, one can simulate other experiments by changing the variables in this equation. These simulations will be considered as the experimental results in the present work.

Neural Network Methodology

In general, to deal with the analysis of experimental data, various cuts in observed variables are made to simplify the desired feature studies. Ideally, an automated optimal choice is desired for describing the whole set of experimental data. In principle, ANNs can deal with this type of problem. The neural network is generally built with one input layer, one or more intermediary layers, and one output layer. The number of neurons in the input and output layers are defined according to the number of experimental or calculated data used to train the network. The neurons in each layer are connected to each other through weights that are fitted to the data distribution, that is, these weights are fitted to minimize the exact answer (experimental data or model calculations) against the neural network result. This is done, therefore, in a slightly different manner compared with the solution based on eq. 7, as will be seen later.

Among many ANN methodologies,[23–25] one popular approach is known as the backpropagation method or feed-forward multilayer perceptrons.[23,24] Formally, one can define the error between the observed variables (y_j) and the the experimental or model data (out_j^l) as

$$\varepsilon^l = \sum_{j=1}^{n}\left(y_j - \text{out}_j^l\right)^2 \quad (11)$$

where (out_j^l) is a function of the weights, n defines the total amount of observed data, and l specifies the answer in the actual layer. In the ANN model, the method for determining the smallest error is by minimizing the error in terms of the

weights of the network rather than in terms of the parameters (as is the case of least-squares method; see eq. 7, and it is formally written as

$$\frac{d\varepsilon^l}{dw_j} = 0 \equiv \frac{\partial \varepsilon^l}{\partial \text{out}_j^l}\frac{\partial \text{out}_j^l}{\partial w_j^l} = 0 \qquad (12)$$

The output of the neuron is calculated as

$$\text{out}_j^l = f\left(\sum_{i=1}^{m} w_{ji}^l \text{out}_i^{l-1}\right) \qquad (13)$$

where m is equal to the number of neurons in the previous layer $(l-1)$, f is chosen as a transfer function (e.g. the sigmoidal function), and w_{ji} are the weights that connect the neurons. The neuron activation function can be written as

$$f(z) = \frac{1 - e^{-\alpha z}}{1 + e^{-\alpha z}} \qquad (14)$$

In the limit as $\alpha \to \infty$, the slope approaches infinity, and $f(z)$ behaves like a threshold function. In general, with the use of a backpropagation network, the learning procedure occurs when each input pattern is applied to the input units and then propagated forward. The pattern of activation arriving at the output layer is then compared with the correct (associated) output pattern to calculate an error signal with eq. 11. The error signal for each such target output pattern is then backpropagated from the outputs to the inputs to appropriately adjust the weights in each layer of the network. These weights can be determined in a very efficient way, which will be discussed later. In general, the weights are corrected as[5,24]

$$\Delta W_{ji}^l = \eta \delta_j^l \text{out}_i^{l-1} + \mu \Delta W_{ji}^{l(\text{previous})} \qquad (15)$$

where ΔW_{ji}^l represents the correction to the weight between the jth element in the lth layer and ith element in the previous layer. The quantity out_i^{l-1} contains the output result of the $l-1$ layer. The parameters η and μ represent the learning rate and the momentum constant, respectively. These quantities determine the rate of convergence during the training procedure in which they are dynamically adjusted to obtain the best rate of convergence during the training procedure in which they are dynamically adjusted to obtain the best rate of convergence. The errors determined during the training stage are given by[5,24]

$$\delta_j^{\text{last}} = \left(y_j - \text{out}_j^{\text{last}}\right)\text{out}_j^{\text{last}}\left(1 - \text{out}_j^{\text{last}}\right) \qquad (16)$$

and

$$\delta_j^l = \left(\sum_{k=1}^{r} \delta_k^{l+1} W_{kj}^{l+1}\right)\text{out}_j^l\left(1 - \text{out}_j^l\right) \qquad (17)$$

where y_j is the output target, which is compared with the output ANN results (out_j^l) of the lth layer. In this work, the neuron behavior was calculated with the sigmoid function for the intermediary layer and a linear function for the output layer.[23,24]

To correlate the input and output data, one needs to optimize the weights and bias by minimization techniques. There are several methods for minimizing functions, such as Newton's method and the Steepest Descent method, and many others for least-squares procedures. The robust method proposed by Levenberg and implemented by Marquardt,[26] also known as the Gauss-Newton method,[23] was used in this work. It is based on both the Steepest Descent and Newton's methods and works through the dynamical adjustment of these two approaches. Its advantage is that it is much faster in the way it finds the minimum. On the other hand, it requires large memory and large storage space, which is proportional to the size of the training set and the number of neurons. However, this methodology provides an efficient procedure for finding a minimum of a given function which can be used for nonlinear models as is the case of the present problem. Details of the Levenberg–Marquardt algorithm can be found elsewhere,[27] and algorithms of this technique are also described by Press et al.[28] According to the Levenberg–Marquardt method (LMM), the updated matrix of weights is calculated as[23]

$$\mathbf{W}_{n+1} = \mathbf{W}_n - (\mathbf{H} + \beta \mathbf{I})^{-1} \nabla \varepsilon^l(\mathbf{W}_n) \qquad (18)$$

where $\mathbf{H}$ is the Hessian matrix, $\mathbf{I}$ is the identity matrix, and β is a variable parameter, which usually starts as $\beta = 0.01$. This parameter is changed during the minimization search according to the estimation of the local error. The most difficult task when the LMM is used can be attributed to the calculation of $\mathbf{H}$. The calculation of Hessian is written as

$$\mathbf{H} = \mathbf{J}^T \mathbf{J} \qquad (19)$$

where $\mathbf{J}$ is the Jacobian matrix and is given by

$$\mathbf{J} = \frac{\partial \varepsilon^l}{\partial \text{out}_j^l} \qquad (20)$$

where ε^l is given by eq. 11. This approximation for solving the Hessian matrix will avoid computation of second derivatives, which simplifies the calculations. Substituting the aforementioned approach into eq. 18, one obtains

$$\mathbf{W}_{n+1} = \mathbf{W}_n - \left[(\mathbf{J}^T(\mathbf{W}_n)\mathbf{J}(\mathbf{W}_n) + \beta_n\mathbf{I}\right]^{-1} \times \mathbf{J}^T(\mathbf{W}_n)\varepsilon^l(\mathbf{W}_n) \quad (21)$$

Equation 21 will approach the pure Newton's method if $\beta \to 0$ or the Steepest descent method when $\beta \to \infty$. It is interesting to note the similarity between the neural network update (eq. 21) and the solution based on least-squares method (eq. 7). Conceptually, these methodologies are different because in the latter case one needs to define the functional form (eq. 5 or 6), and in the ANN case, it is not necessary. However, in both cases the solution is based on the Jacobian matrix, and this is the basis of minimization problems. Therefore, conceptually, these methodologies are quite different in terms of with which parameters the jacobian matrix is constructed, but the numerical solution can be considered equivalent from the computational point of view. In the neural network model, after the training stage, one ends up with the following

$$\begin{bmatrix} D \\ C_s \\ A \\ h \end{bmatrix} = (\mathbf{w}_{kj}) \tan h\left[(\mathbf{w}_{ji})\binom{F}{t} + (\mathbf{b}_j)\right] + (\mathbf{b}_k) \quad (22)$$

where $\mathbf{w}_{ji}$ are the synaptic weights to correlate the input data defined by the F and t values, $\mathbf{b}_j$ is a vector which defines the bias terms in the intermediary layer, $\mathbf{b}_k$ defines the bias vector in the output layer, and $\mathbf{w}_{kj}$ corresponds to the synaptic weights of the output layer.

Despite the observed different concepts in the ANN and conventional methods, the neural network solution, eq. 22, is a linear function. In addition, for conditions where there exist a dependence between other parameters, one can recognize that the least-squares method becomes more complex from a mathematical point of view. For the particular case of the diffusion coefficient there is, for example, a dependence[29] of porosity (ε), tortuosity (τ), and molecular weight (M_w)[i.e., $D \to D(\varepsilon, \tau, M_w)$]. Broadly speaking, a general function of the type $F \to F(D, C_s, A, t)$ can be proposed. Moreover, each parameter in F can be a function of other variables, and here we considered a generic dependence; that is, $D \to D(r, s, v)$, $C_s \to C_s(r, s, v)$ and $A \to A(r, s, v)$, where r, s, and v define other dependencies. For this particular case, the solution based on least-squares method can be determined by constructing the Jacobian matrix, which is written as

$$J = \begin{bmatrix} \frac{\partial D}{\partial r} & \frac{\partial D}{\partial s} & \frac{\partial D}{\partial v} \\ \frac{\partial C_s}{\partial r} & \frac{\partial C_s}{\partial s} & \frac{\partial C_s}{\partial v} \\ \frac{\partial A}{\partial r} & \frac{\partial A}{\partial s} & \frac{\partial A}{\partial v} \end{bmatrix} \quad (23)$$

The addition of new variables in the solution based on neural network will be a matter of including more neurons in the input/output layers and perhaps more neurons in the intermediary layer. Therefore, it is only a matter of adding more neurons and, hence, the Jacobian in the case of the ANN method will be formally the same but with a dimension slightly higher. From eqs. 11, 12, and 20 one can write the following Jacobian:

$$J = \begin{bmatrix} \frac{\partial \varepsilon_{11}}{\partial w_1} & \frac{\partial \varepsilon_{11}}{\partial w_2} & \cdots & \frac{\partial \varepsilon_{11}}{\partial w_m} \\ \frac{\partial \varepsilon_{21}}{\partial w_1} & \frac{\partial \varepsilon_{21}}{\partial w_2} & \cdots & \frac{\partial \varepsilon_{21}}{\partial w_m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial \varepsilon_{j1}}{\partial w_1} & \frac{\partial \varepsilon_{j1}}{\partial w_2} & \cdots & \frac{\partial \varepsilon_{j1}}{\partial w_m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial \varepsilon_{1i}}{\partial w_1} & \frac{\partial \varepsilon_{1i}}{\partial w_2} & \cdots & \frac{\partial \varepsilon_{1i}}{\partial w_m} \\ \frac{\partial \varepsilon_{2i}}{\partial w_1} & \frac{\partial \varepsilon_{2i}}{\partial w_2} & \cdots & \frac{\partial \varepsilon_{2i}}{\partial w_m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial \varepsilon_{ji}}{\partial w_1} & \frac{\partial \varepsilon_{ji}}{\partial w_2} & \cdots & \frac{\partial \varepsilon_{ji}}{\partial w_m} \end{bmatrix} \quad (24)$$

As can be observed, the least-squares method needs the jacobian matrix constructed in terms of parameters eqs. 8 and 23, whereas in the ANN, such a matrix is built in terms of the weights (eq. 24). This comparison shows that in the former case, one minimizes the parameters, which defines the problem, and in the ANN case, the minimization goes through the weights. The methods can be used complementarily to tackle complex problems, such as drug release systems, when there are several variable dependences.

A flowchart describing the neural network algorithm used in this work is shown in Figure 1. As can be seen, initially one provides the input and output data, the number of neurons, and the number of layers, and that the initial weights and bias are chosen randomly. In our algorithm, we used the random procedure proposed by Park

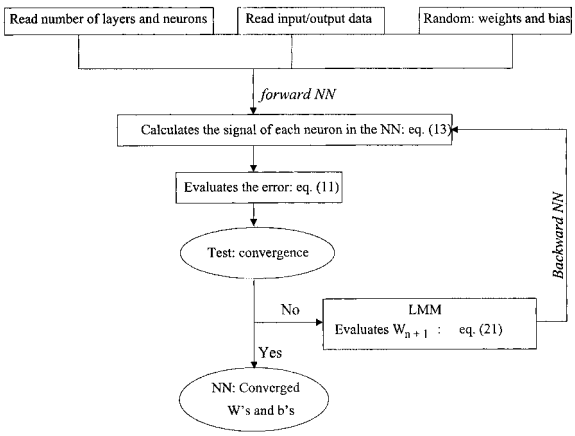

Figure 1. Schematic flowchart of the neural network used in this work.

et al.[29] to generate the initial conditions for the weights and bias.

A preliminary analysis of the converged data by the neural network was performed considering eight runs with different initial seeds. Each run was performed five times and, from these runs, we selected the best initial seed that provided the lowest error in the network. This seed was used in all calculations.

The weights in the multilayer perceptron methodology are corrected backwards and the time demand of the whole algorithm is mostly due to the calculation of the Jacobian matrix (eq. 20) in the LMM. After convergence, one obtains the W and b values that best correlate the input and output spaces. This algorithm was written in Fortran 77 and implemented on a 500 au workstation Digital with 256 Megabytes of memory.

RESULTS AND DISCUSSION

The theoretical models tested through the hydrocortisone system showed quantitative agreement with the drug fraction released from experiments. The experiments of Fu et al.[2] were considered to validate the model proposed to study dispersed-type systems.

To use the experimental data of hydrocortisone reported by Fu et al. we used his model (eq. 5) to determine the diffusion coefficient (5.58×10^{-10} cm^2 s^{-1}), which was slightly different from their result (1.58×10^{-10} cm^2 s^{-1}). Hence, we decided to use our result to analyze the drug release through the ANN. Therefore, considering $D = 4.82 \times 10^{-5}$ cm^2 days^{-1} for the particular case of the polycarpolactone matrix, one can simulate an experiment with almost real experimental conditions. The drug loading (A) was also calculated based on the data from ref. 2; that is, for a flat disk used in ref. 2, the authors considered 40 mg of the drug loading and a shape with a volume equal 0.300 cm^3 (disk of 1.5 cm diameter and 0.17 cm thick). These data provide a drug loading of 133.1 mg cm^{-3}. The same procedure used to determine the D parameter was applied to calculate the saturation solubility, C_s, as 40 mg cm^{-3}. These data can approximately reproduce the hydrocortisone drug release profile.[2]

Considering these data and eq. 10, other experiments can be simulated also using the hydrocortisone as a model, to test the ANN procedure. The parameters used to simulate three different cumulative fraction releases and the parameters used in ref. 2 to fit the hydrocortisone experimental release are shown in Table 1. As observed, the diffusion coefficient parameter is changed ~100 times between 4.82×10^{-5} and 0.042×10^{-5} cm^2 days^{-1}. As analyzed by Fu et al., four different matrices (polycarpolactone, ethylene vinyl acetate, PVA terpolymer, and silicone rubber) were used with ranges also within this magnitude. Similarly, the other two parameters (C_s and A) were modified 15 and 4 times, respectively. The height of the matrix was taken also from Fu et al.,[2] and it does not change much because the condition design of the device.

Each set of parameters given in Table 1 was substituted in eq. 10 and the release profile was calculated as a function of time, as shown in Figure 2. These four cumulative releases were assumed to be the experimental results. Each curve in this figure was calculated at 20 points to be used to train the neural network. Therefore, can one use these data to learn the behavior of the hydrocortisone cumulative fraction release. The question is whether one can predict other releases

Table 1. Hydrocortisone Parameters for a Release of ~100 Days[a]

$D \times 10^5$, cm^2 Days^{-1}	C_s, mg cm^{-3}	A, mg cm^{-3}	h, cm
4.82	40.0	133.1	0.170
2.41	29.3	100.5	0.168
0.803	9.3	54.5	0.166
0.042	2.7	33.3	0.164

[a]Experimental data (first line) were taken from ref. 2, and the others were proposed to be used in eq. (10).

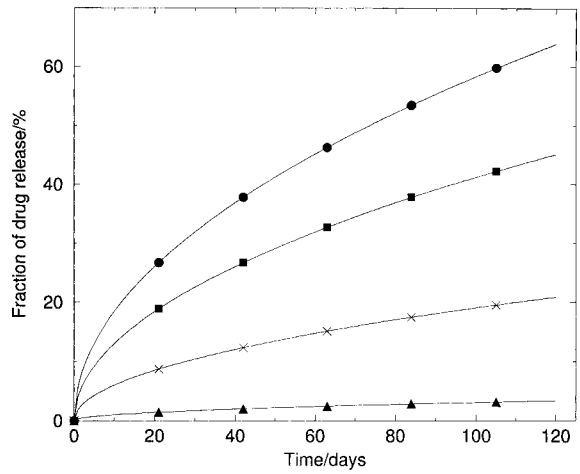

Figure 2. Four different drug fraction released used during the training procedure. Symbols are: black dots ($D = 4.82 \times 10^{-5}$ cm^2 days^{-1}, $C_s = 40.0$ mg cm^{-3}, $A = 133.1$ mg cm^{-3} and $h = 0.170$ cm), black squares ($D = 2.41 \times 10^{-5}$ cm^2 days^{-1}, $C_s = 29.3$ mg cm^{-3}, $A = 100.5$ mg cm^{-3} and $h = 0.168$ cm), X ($D = 0.803 \times 10^{-5}$ cm^2 days^{-1}, $C_s = 9.3$ mg cm^{-3}, $A = 54.5$ mg cm^{-3} and $h = 0.166$ cm), and black triangles ($D = 0.042 \times 10^{-7}$ cm^2 h^{-1}, $C_s = 2.7$ mg cm^{-3}, $A = 33.3$ mg cm^{-3} and $h = 0.164$ cm).

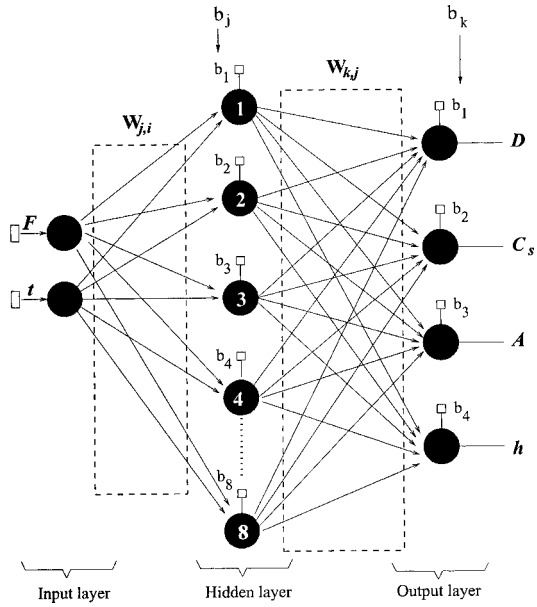

Figure 3. The artificial neural network. Input data are F and t (time) and output data are the physical-chemical parameters: D, C_s, A and h. There is one intermediary layer with 8 neurons. Each neuron in the intermediary and output layers has one bias.

after the learning stage. To answer this question we propose to use ANNs.

The ANN used to analyze the controlled release problem is shown in Figure 3. There are two input data (F and t), one intermediary layer, and one output layer that define the desired answer in the network. In the present case, these responses are D, C_s, A, and h. Therefore, one can observe that a two-dimensional input space is correlated with a hypersurface in the output network. One advantage of using the neural network is that its complexity increases linearly[30] rather than in a cubic way, as is the case of standard nonlinear methods. Accordingly, the neural network method has the advantage to tackle very complicated problems. For example, Wu et al.[31–33] used such an approach to classify DNA sequences in a very large database. This type of complexity perhaps cannot be studied by procedures based on standard nonlinear methodologies. However, the neural network has limitations, and one problem is how to define the initial conditions and the number of neurons in the intermediary layer.

In general, the network results may be unconverged before the ideal number of layers and neurons is determined. On the other hand, adding more neurons one can expect an overfitting in the neural network. Trial and error procedures can be used to solve this problem or other techniques such as genetic algorithms[34,35] to define better initial conditions and hence better neural network structures.[36]

The ideal neural network structure was proposed after several tests. First, the maximum number of epochs (number of iterations) was defined. Usually, the LMM does not use many epochs to find a reasonable convergence, whereas the backpropagation method requires larger values.[37] The number of epochs was initially fixed to 500 and the parameter β in eq. 18 was set equal to 0.0001, slightly smaller than usual (0.01). As described in the previous section, this parameter is dynamically adjusted to improve convergence. In our tests, the results showed that 3 or 4 neurons in the intermediary layer were not enough to provide a satisfactory convergence. Similarly 9 and 10 neurons in this layer can probably cause an overfitting in the ANN, although converged results could be found. As can be observed in Figure 4, the lowest error was achieved when 8 neurons were chosen. Therefore, the ANN structure was fixed with 2 neurons in the input layer, 8 neurons in one intermediary layer, and 4 neurons in the output layer. With this ANN configuration one can, in principle, assume that there is no overfitting in our calculations.

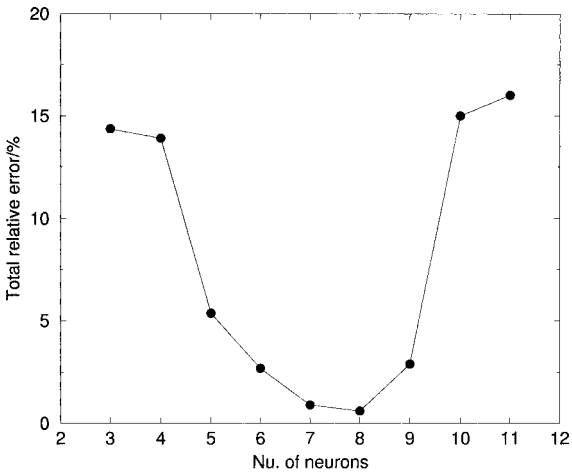

Figure 4. Total relative error of all parameters (D, C_s, A and h) as a function of number of neurons.

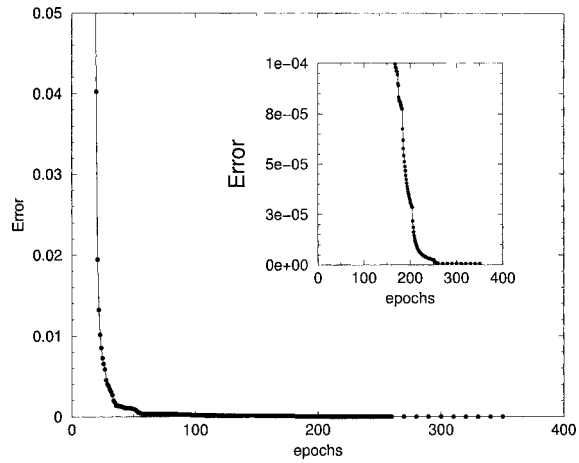

Figure 5. Relative square error (eq. 11) as a function of epochs for the ANN shown in Figure 3. Details are shown near the plateau region.

Hydrocortisone Drug Release Example

The cumulative profiles shown in Figure 2 can be used to define a region from the lowest release (~3%) to the highest release (~60%) in which an unknown release can be determined. The question now is: What are the ideal experimental parameters (D, C_s, A, and h) that could provide an efficient release within this range (3–60%). To answer this question we selected a region between 25 and 35% for the drug fraction released. Of course, a different region could be selected within the range of the lowest and highest releases shown in Figure 2.

The number of neurons in the intermediary layer was tested (Fig. 4), and 8 neurons were enough to provide a satisfactory average error in the parameters (D, C_s, A, and h) during the training stage. This result means that the ANN reached a minimum region where the square error value stabilized. One should point out that a lower minimum could be, in principle, obtained by adding more neurons or layers in the network, but this could cause overfitting as discussed above.

The ANN convergence is analyzed through the minimization of the error function (eq. 11). In this work the training error was checked and a square error of $\sim 10^{-6}$ was assumed to be a satisfactory outcome for the ANN training. This error was enough to produce an average error of ~0.1% for predicting the output physical parameters (D, C_s, A, and h) through the ANN calculation in the training stage. The error minimization is shown in Figure 5. As can be seen, the error drastically diminished with very few epochs, which is expected when the LMM is used. However, the training continues until a plateau is observed, which defines a nonsensitive region in the error surface. In the actual case, 300 epochs were enough to reach this region with the square error of $\sim 10^{-6}$, although the calculations were carried on further (i.e., up to 350 epochs). This situation can be assumed as a possible convergence condition and then the neural network training was stopped. However, depending on the initialization of bias and weights, one can obtain this plateau without convergence, but this needs to be tested.

After the convergence of the training data, all bias and weights are assumed to contain all the informations concerning the learning stage. One can argue that these network parameters do not have any physical nature of the problem and this is known.[30] Nevertheless, these bias and weights are able to construct the correlation between the nonlinear input/output data; that is, the fraction released and the properties of an unknown experimental parameters (D, C_s, A, and h) that can be mapped out. This implies another question for which standard procedures may not give an answer. What is the correct information about an ideal experiment to provide a drug fraction released in a specified condition or region? To answer this question, one can use eq. 22, defined by the neural network after the training procedure.

For testing validation one can simulate an ideal drug fraction released or one can provide a real experiment within the training region. As previously defined, we are using a region for an unknown drug released within the range of 25–35%. The dots in Figure 6 show this release in

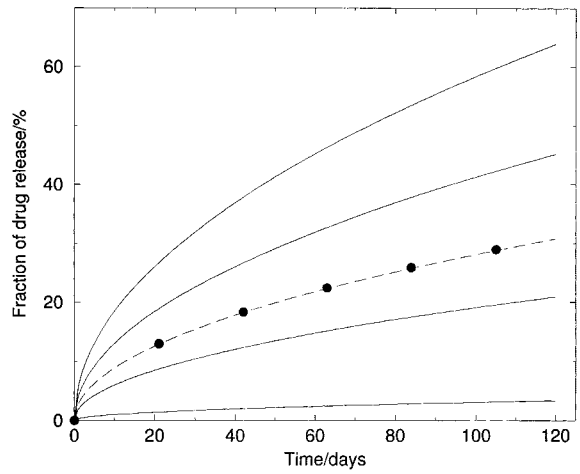

Figure 6. The predicted data (black dots) compared with the exact release (long-dashed line) for the ideal parameters ($D_{ideal} = 1.35$ cm^2 days^{-1}, $C_{s_{ideal}} = 16.2$ mg cm^{-3}, $A_{ideal} = 70.0$ mg cm^{-3} and $h_{ideal} = 0.167$ cm). The individual relative error is 0.9% (D), 0.4% (C_s), 0.7% (A) and 0.2% (h). The other releases are those shown in Figure 2 and correspond to the parameters given in Table 1.

which one does not know the physical parameters (D, C_s, A, and h). Such an unknown experiment was proposed with a fraction released of 30%. This release profile was generated by considering $D_{ideal} = 1.35$ cm^2 days^{-1}, $C_{s_{ideal}} = 16.2$ mg cm^{-3}, $A_{ideal} = 70.0$ mg cm^{-3}, and $h_{ideal} = 0.167$ cm. The question one now ask is: Can the ANN predict these parameters by using the weights and bias (eq. 22), which should contain the information of those cumulative fraction released experiments (Fig. 2)?

The neural network results calculated using eq. 22 are quantitative with respect to the ideal data previously proposed as ideal values of D, C_s, A, and h. The relative average error taken into account the latter parameters is <1%, and this was calculated according to

$$E_i = \left| \frac{P_{cal} - P_{exp}}{P_{exp}} \right| \times 100 \qquad (25)$$

where E_i is the relative error of each property i, P_{cal} defines the calculated property (D, C_s, A, or h) and P_{exp} is the corresponding property assumed as the experimental result. Individually, the diffusion coefficient (D) showed the highest relative error (0.9%), whereas the other parameters showed relatively smaller errors; namely, $E_{Cs} = 0.4\%$, $E_A = 0.7\%$, and $E_h = 0.2\%$. Although one observes a larger error for one individual parameter (D), the average is relatively smaller due to the lowest error obtained for the parameter h (0.2%).

The previous analysis was performed considering the experimental release as the highest profile. Another test was performed also considering four releases but with one simulation assumed to be higher than the experimental profile of ~60% of hydrocortisone profile and the two others were below. As compared by Cardinal et al.[21] the theoretical model (eq. 10) and the interpolated experimental data of Fu et al.[2] showed quantitative agreement for a cumulative release of ~60% (stars in Fig. 7) after ~120 days.[2,21] Their fitted parameters (D, C_s, A, and h) compared with the experimental results correspond to those given in Table 1 (first line) and also reproduced in Table 2 (last line), but the other parameters shown in Table 2 were simulated to be used by the ANN for the training stage. The same neural network structure was used in this part of the calculations; that is, 2 input data (F and t), 8 neurons in the intermediary layer, and 4 output neurons (D, C_s, A, and h). After the training stage, the parameters D, C_s, A, and h were calculated with eq. 22 and the corresponding release was calculated with eq. 10. As one observes in Figure 7, there is no difference between the neural network results (long-dashed line) and the experimental data (stars) for ~60% of release after ~100 days; that is, the error was <1%.

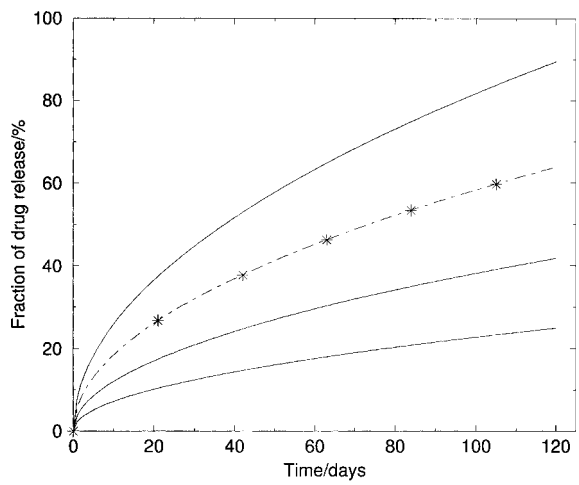

Figure 7. Experimental data (long-dashed line) for the hydrocortisone system (taken from ref. 2) and symbols (stars) are the neural network results with the parameters $D = 4.84 \times 10^{-5}$ cm^2 days^{-1}, $C_s = 39.6$ mg cm^{-3}, $A = 133.3$ mg cm^{-3} and $h = 0.1699$ cm (see Table 2 for the experimental data). The other releases were simulated using eq. (10) and the parameters given in Table 2 which were used to train the ANN.

Table 2. Prediction of Experimental Results from Reference 2, Using Three Simulated Data to Train the Neural Network

Simulated Data	$D \times 10^5$, cm^2 Days^{-1}	C_s, mg cm^{-3}	A, mg cm^{-3}	h, cm
1	1.12	19.0	115.0	0.162
2	2.34	31.0	127.0	0.166
3	8.64	45.0	147.0	0.173
Experiment	4.82	40.0	133.1	0.170

Therefore, this analysis showed that the present methodology can also predict experimental results with high accuracy. In addition, the tests showed that the prediction can be obtained either for lower or higher drug release profiles. Accordingly, the experimentalist can take the neural network information and using the parameters such as D, C_s, A, and h to perform an efficient experiment for controlled drug release.

Rhodium (II) Butyrate Drug Release Example

Finally, the present methodology was also tested with different experimental data. Although the model proposed by Fu et al. is derived for biodegradable polymers, our analysis for the study of the burst effect showed accurate results.[20] Therefore, we performed two new experiments with 7.5 and 12.5% of rhodium (II) butyrate loading in hydroxyapatite matrix, and the data from ref.[20] for 5 and 10% drug loading were used to test the present approach. The drug loading (A) was calculated considering a flat disk of radius and thickness of 0.5 and 0.1 cm, respectively, and these data provide a volume of the device of 0.08 cm^3. These experimental data were fitted to smooth data, and the results are shown in Figure 8. The molded pallets were disks of 0.5 cm radius and 0.1 cm thickness (h), and therefore, for this particular analysis, one has fixed the h parameter. For these drug contents and geometry, the drug loading (A) is determined. From eqs. 6 and 10, the parameters D and C_s can be calculated, and the data are summarized in Table 3.

The release profile of 10% (triangles) shown in Figure 8 was selected to test the neural network prediction and therefore, for cross validation, this release profile was not used to train the network. The other three sets (5, 7.5, and 12.5%) were used to train the network with the same architecture previously used for the hydrocortisone case but with one less neuron in the output layer because the height of the device, parameter h, was fixed for the Rh(II) butyrate complex. The neural network results provided the diffusion coefficient $D = 6.454 \times 10^{-6}$ cm^2 h^{-1}, $C_s = 123.3$ mg cm^{-3}, and $A = 127.0$ mg cm^{-3}. The comparison with the drug loading ($A = 127.4$ mg cm^{-3}) from ref.[20] gives an error of 0.3%. The two other parameters, D (6.5×10^{-6} cm^2 h^{-1})[18] and C_s (124.0 mg cm^{-3}), were calculated with eqs. 5 and 9, and the comparison with the corresponding data obtained by the ANN shows errors of 0.6 and 0.7%, respectively. Overall, the average error is <1%, as previously obtained for the hydrocortisone system.

As can be observed, the neural network has provided quantitative agreement with the real parameters either using a biodegradable polymer (polycarpolactone)–hydrocortisone case or using a bioceramic (hydroxyapatite)–Rh(II) complex case. In addition, the present methodology showed satisfactory results for early time release (Rh(II) complex) or for late time release (hydrocortisone

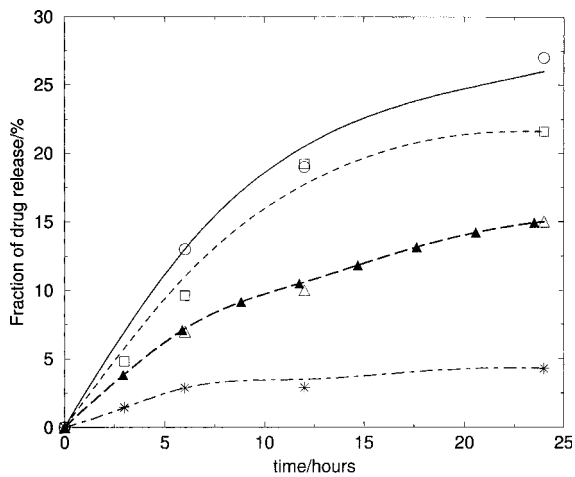

Figure 8. Rhodium (II) butyrate release profiles for four different drug loading. The open symbols are circles (5%), squares (7.5%), triangles (10%) and stars (12.5%). The lines are fitted data and the black triangles with the thick long-dashed line are results calculated with parameters determined by the neural network: $D = 6.45 \times 10^{-6}$ cm^2 h^{-1}, $C_s = 123.3$ mg cm^{-3}, and $A = 127.0$ mg cm^{-3}.

Table 3. Physical–Chemical Parameters for the Release Profile of Rhodium (II) Butyrate Complex

$D \times 10^5$, $cm^2\ h^{-1}$ [a]	C_s, mg cm^{-3} [b]	A, mg cm^{-3} [c]
2.1[d]	114	63.7
1.8	118	95.5
0.65[d]	124	127.4
0.069	127	159.2

[a] Diffusion coefficient.
[b] Saturation solubility.
[c] Drug loading, calculated with a volume equal to 0.08 cm^3 for 5, 7.5, 10, and 12.5%.
[d] Reported in ref. 20.

system). These results show that the early time or late time models[21] are included when the experiments are treated using the neural network method. Therefore, the proposed methodology can be used to help experimental studies to reduce further experiments. This analysis could be solved, in principle, by least-squares methodologies, but as previous discussed, the solution based on neural networks does not need to define mathematical functions. However, on the application of the ANN, the Higuchi and Fu et al.'s models were necessary to calculate other parameters, such as the diffusion coefficient and the saturation solubility. Nevertheless, the well-established conventional methods with the neural network approach can be complementary and hence can provide useful and more efficient strategies for studying controlled-release systems. Moreover, if experiments could provide, for example, the diffusion coefficients, one could use the data directly with the neural network method to determine ideal experimental conditions. As observed, the ANN methodology can learn with other data, whereas, in principle, least-squares method needs a functional form to be fitted. Thus, although ANNs do not have physical meaning as the well-established models, there are advantages; for example, the approach can learn the behavior of other experiments to be used to predict a desirable release profile in which the physical parameters are unknown.

CONCLUSIONS

The present study has dealt with the analysis of an alternative model based on ANNs to be a complementary strategy to study controlled drug release. As demonstrated in this work, the methodology of neural networks may be useful to the experimentalists to determine ideal experimental parameters to provide an efficient controlled drug release. However, the approach cannot be used without the experimental data and, in many cases, without the use of conventional mathematical models to simulate more experimental data when these data are not available.

In particular, the studies have focused on the analysis of controlled drug release, taking into account physical–chemical properties, namely the diffusion coefficient (D), the saturation solubility (C_s), the drug loading (A), and the height of the device (h). Considering the hydrocortisone system as a model, the procedure was quantitative enough to provide the ideal properties for an efficient release of ~30% for up to 100 days, with a relative average error of <1%.

The other test using 5, 7.5, 10, and 12.5% of drug loading of rhodium (II) butyrate complexes also showed accurate results for predicting the physical parameters D, C_s, and A, with a relative error also <1%. These tests showed that the methodology can be used either for biodegradable polymers or bioceramic matrices. In addition, the neural network provided efficient results for early time (Rh (II) complex) and late time (hydrocortisone case) releases. Accordingly, the present study showed a useful strategy to tackle drug delivery systems. Furthermore, as pointed out in ref. 38, neural networks are able to reconstruct patterns from noisy samples.

In conclusion, the number of necessary experiments can be minimized and, hence, the costs are significantly reduced.[7] Based on the results presented here, we believe that this methodology can be extended, and further studies are currently being undertaken using real experimental data to be tested against the ANN results to determine the physical and chemical parameters D, C_s, A, and h for an efficient controlled drug release system. However, it is important to point out that this ANN methodology is proposed to be a complementary approach to those methodologies already established, such as least-squares methods. Through the use of both methods, one can further provide better tools with a high level of confidence to help experimental work.

ACKNOWLEDGMENTS

We would like to thank the unknown referees whose useful comments have made our manuscript clearer. This work was supported by CNPq and FAPEMIG—Brazil.

REFERENCES

1. Siepmann J, Peppas NA. 2001. Modeling of drug release from delivery systems based on hydroxypropyl methylcellulose (HPMC). Adv Drug Deliv Rev 48:139–157.
2. Fu JC, Hagemeir C, Moyer DL, Ng EW. 1976. A unified mathematical model for diffusion from drug-polymer composite tables. J Biomed Mater Res 10:743–758.
3. Siepmann J, Streubel A, Peppas NA. 2002. Understanding and predicting drug delivery from hydrophilic matrix tablets using the "sequential layer" model. Pharm Res 19:306–314.
4. Higuchi T. 1961. Rate of release of medicaments from ointment bases containing drugs in suspension. J Pharm Sci 50:874–875.
5. de Viterbo VD, Belchior JC. 2001. Artificial neural network for studing metallic complexes. J Comput Chem 22:1691–1701.
6. Wajima T, Fukumura K, Yano Y, Oguma T. 2002. Prediction of human clearance from animal data and molecular structural parameters using multivariate regression analysis. J Pharm Sci 91:2489–2499.
7. Wu PC, Obata Y, Fujikama M, Li CJ, Higashiyama K, Takayama K. 2001. Simultaneous optimization based on artificial neural networks in ketoprofen hydrogel formula containing *O*-ethyl-3-butylcyclohexanol as percutaneous absorption enhancer. J Pharm Sci 90:1004–1014.
8. Takayama K, Morva A, Fujikama M, Hattori Y, Obata Y, Nagai T. 2000. Formula optimization of theophylline controlled-release tablet based on artificial neural networks. J Controlled Release 68:175–186.
9. Hussain AS, Shivanand P, Johnson RD. 1994. Application of neural network computing in pharmaceutical product devepment—computer-aided formulation design. Drug Dev Ind Pharm 20: 1739–1752.
10. Prank K, Kloppstech M, Brabant G. 1997. Neural networks in the analysis of episodic growth hormone release. Human Reprod Update 3:215–234.
11. Peh KK, Lim CP, Quek SS, Khoh KH. 2000. Use of artificial neurla networks to predict drug dissolution profiles and evaluation of network performance using similarity factor. Pharm Res 17:1384–1388.
12. Richardson CJ, Barlow DJ. 1996. Neural network computer simulation of medical aerosols. J Pharm Pharmacol 48:581–591.
13. Chen YX, McCall TW, Baichwal AR, Meyer MC. 1999. The applications of an artificial neural network and pharmacokinetic simulations in the design of controlled-release dosage forms. J Controlled Release 59:33–41.
14. Ibric S, Jovanović M, Djuric Z, Parojcić J, Solomun L. 2002. The application of generalized regression neural network in the modeling and optimization of aspirin extended release tablets with Eudragit (R) RS PO as matrix substance. J Controlled Release 82:213–222.
15. Takahara J, Takayama K, Nagai T. 1997. Multi-objective simultaneous optimization technique based on an artificial neural network in sustained release formulations. J Controlled Release 49: 11–20.
16. Saarinen S, Bramley R, Cybenko G. 1993. III-Conditioning in neural network training problems. SIAM J Sc Comput 14:693–714.
17. Hornik K, Stinchcombe M, White H. 1989. Multilayer feedforward networks are universal approximators. Neural Networks 2:359–366.
18. Geva S, Sitte J. 1992. A construtive method for multivariate function approximation by multilayer perceptrons. IEEE Trans. Neural Networks 3: 621–624.
19. Bishop C. 1995. Neural Networks for Pattern Recognition. Oxford, UK: Oxford Press.
20. Burgos AE, Belchior JC, Sinisterra RD. 2002. Controlled release of rhodium (II) carboxylates and their association complexes with cyclodextrins from hydroxyapatite matrix. Biomaterials 23:2519–2526.
21. Cardinal JR. 1984. Matrix Systems. In: Langer RS, Wise DL, editors. Medical applications of controlled release. Florida: Boca Raton, FL: CRC Press, Vol. I, Chapter 2.
22. Narasimhan B, Langer R. 1997. Zero-order release of micro- and macromolecules from polymeric devices: The role of the burst effect. J Controlled Release 47:13–20.
23. Haykin S. 1999. Neural networks—A comprehensive foundation, Second Edition. New Jersey: Prentice Hall.
24. Gasteiger J, Zupan J. 1993. Neural networks for chemists. Weinheim: VCH, Chapter 8.
25. Bose NK, Liang P. 1996. Neural network fundamentals with graphs, algorithms and applications. New York: McGraw-Hill.
26. Marquardt DW. 1963. An algorithm for least-squares estimation of nonlinear parameters. J Soc Ind Appl Math 11:431–441.
27. Bevington PR. 1969. Data reduction and error for the physical sciences. New York: McGraw-Hill, Chapter 11.
28. Press WH, Teukolsky SA, Veterling WT, Flannery BP. 1984. Numerical recipes in Fortran: The art of scientific computing. New York: Cambridge University Press; Chapter 14.
29. Park SK, Miller KW. 1988. Random number generators—Good ones are hard to find. Commun ACM 31:1192–1201.
30. Braga AP, Braga JP, Belchior JC. 1997. Artificial neural network applied for predicting rainbow trajectories in atomic and molecular classical collisions. J Chem Phys 107:9954–9959.

31. Wu C, Whitson G, McLarty J, Ermongkoncha A, Chang TC. 1992. Protein classification artificial neural network system. Protein Sci 1:667–677.
32. Wu C, Berry M, Shivakumar S, McLarty J. 1995. Neural network for full-scale protein-sequence classification—Sequence enconding with singular-value decomposition. Mach Learn 21:177–193.
33. Wu CH. 1998. A neural network system with n-gram term weighting method for molecular sequence classification and motif identification. U.S. Patent 5,845,049.
34. Arifovic J, Gencay R. 2001. Using genetic algorithms to select architecture of a feedforward artificial neural network. Physica A 289:574–594.
35. Richards E, Bessant C, Saini S. 2002. Optimisation of a neural network model for calibration of voltammetric data. Chem Int Lab Sys 61:35–49.
36. Udelhoven T, Schutt B. 2000. Capability of feedforward neural networks for a chemical evaluation of sediments with diffuse reflectance spectroscopy. Chem Int Lab Sys 51:9–22.
37. Gupta A, Lam SM. 1998. Weight decay backpropagation for noisy data. Neural Networks 11:1127–1137.
38. Agatonovic-Kustrin S, Beresford R. 2000. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. J Pharm Biol Anal 22:717–727.