Available online at www.sciencedirect.com
ScienceDirect
International Journal of Pharmaceutics 327 (2006) 126–138
INTERNATIONAL JOURNAL OF PHARMACEUTICS
www.elsevier.com/locate/ijpharm

Performance comparison of neural network training algorithms in modeling of bimodal drug delivery

A. Ghaffari [a], H. Abdollahi [b,*], M.R. Khoshayand [a], I. Soltani Bozchalooi [c], A. Dadgar [a], M. Rafiee-Tehrani [a]

[a] *School of Pharmacy, Tehran University of Medical Sciences, Tehran, Iran*
[b] *Institute for Advanced Studies in Basic Sciences, Zanjan, Iran*
[c] *Department of Mechanical Engineering, University of Ottawa, Ottawa, Ontario, Canada*

Received 29 January 2006; received in revised form 4 July 2006; accepted 25 July 2006
Available online 4 August 2006

Abstract

The major aim of this study was to model the effect of two causal factors, i.e. coating weight gain and amount of pectin–chitosan in the coating solution on the *in vitro* release profile of theophylline for bimodal drug delivery. Artificial neural network (ANN) as a multilayer perceptron feedforward network was incorporated for developing a predictive model of the formulations. Five different training algorithms belonging to three classes: gradient descent, quasi-Newton (Levenberg–Marquardt, LM) and genetic algorithm (GA) were used to train ANN containing a single hidden layer of four nodes. The next objective of the current study was to compare the performance of aforementioned algorithms with regard to predicting ability. The ANNs were trained with those algorithms using the available experimental data as the training set. The divergence of the RMSE between the output and target values of test set was monitored and used as a criterion to stop training. Two versions of gradient descent backpropagation algorithms, i.e. incremental backpropagation (IBP) and batch backpropagation (BBP) outperformed the others. No significant differences were found between the predictive abilities of IBP and BBP, although, the convergence speed of BBP is three- to four-fold higher than IBP. Although, both gradient descent backpropagation and LM methodologies gave comparable results for the data modeling, training of ANNs with genetic algorithm was erratic. The precision of predictive ability was measured for each training algorithm and their performances were in the order of: IBP, BBP > LM > QP (quick propagation) > GA. According to BBP–ANN implementation, an increase in coating levels and a decrease in the amount of pectin–chitosan generally retarded the drug release. Moreover, the latter causal factor namely the amount of pectin–chitosan played slightly more dominant role in determination of the dissolution profiles.

Keywords: Artificial neural network (ANN); Bimodal drug delivery; Genetic algorithm (GA); Gradient descent algorithm; Levenberg–Marquardt (LM) algorithm; Modeling; Training algorithm

1. Introduction

The application of artificial neural networks (ANNs) in the field of pharmaceutical development and optimization of the dosage forms has become a topic of discussion in the pharmaceutical literature (Kesavan and Peck, 1996; Takahara et al., 1997; Takayama et al., 1999; Chen et al., 1999; Wu et al., 2000). Compared with classical modeling techniques, such as response surface methodology (RSM), ANNs show superiority as a modeling technique for data sets showing non-linear relationships, and thus for both data fitting and prediction abilities (Bourquin et al., 1997a, 1998a,b).

ANN is a learning system based on a computational technique that can simulate the neurological processing ability of the human brain and can be applied to quantify a non-linear relationship between causal factors and pharmaceutical responses by means of iterative training of data obtained from a designed experiment (Achanta et al., 1995).

Bimodal drug release profile, where the release is slow in the initial stages of gastrointestinal tract (GIT) and increases to a faster rate at some later stages may be of the significant therapeutic benefit (Maggi and Conte, 1997). In disease states such as

* Corresponding author at: Department of Chemistry, Institute for Advanced Studies in Basic Sciences (IASBS), P.O. Box 45195-159, Zanjan, Iran. Tel.: +98 2414153122; fax: +98 2414153232.
E-mail address: abd@iasbs.ac.ir (H. Abdollahi).

0378-5173/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.ijpharm.2006.07.056

nocturnal asthma, increased drug release rates may help to prevent the exacerbation of symptoms caused by circadian rhythms (Lemmer, 1991). Bimodal release profiles can be utilized so that the drug release gets slower in a region within the GIT where absorption is good, e.g. the small intestine and increases lower down the GIT, e.g. the colon where drug absorption may be poor. The overall effect being to maintain therapeutic blood drug levels throughout (Macleod et al., 1999).

Natural or modified polysaccharides such as gelatin, dextran, chondroitin sulphate, calcium pectinate, pectin and chitosan have been used as potential carriers for the peroral delivery of drugs to the colon as they are safe, biodegradable and widely available (Ashford et al., 1994).

Among these polymers, the use of pectin and chitosan has shown particular promise, as they are able to form polyelectrolyte complex (PEC). Complexation of pectin with chitosan in the form of PEC allows control over drug release while maintaining the ability of the pectin to be degraded by colonic bacteria; thus, potentially achieving bimodal delivery (Macleod et al., 1999).

In our earlier studies (Ghaffari et al., 2006a), the effects of the ratio of pectin–chitosan complex to Eudragit® RS and total polymeric coating weight gain on the drug release from the coated pellets as well as the drug release mechanism were investigated.

The main objective of the present study was to model the effect of two variables, i.e. the amount of pectin–chitosan complex in the coating solution and the coating weight gain on the *in vitro* release profile of theophylline from coated pellets using ANN methodology and afterwards, predicting performance of various algorithms were compared. Less attention has been paid to the effect of training algorithm on the predictive ability of the resultant pharmaceutical model. Murtoniemi et al. (1993) have evaluated standard backpropagation (Freeman and Skappura, 1991), stochastic backpropagation, quick propagation (Fahlman, 1988) and Weigend weight eliminator (Weigend et al., 1992). More recently, Plumb and co-workers compared three ANN programs and four classes of training algorithm in order to optimize the predictive ability of ANN models (Plumb et al., 2005). However, neither of above-mentioned researchers have used genetic algorithm as a training method. Here, five training algorithms belonging to three classes have been evaluated: gradient descent algorithm, quasi-Newton algorithm (Levenberg–Marquardt) and genetic algorithm.

1.1. ANNs in general

Bourquin et al. (1997b) and Agatonovic-Kustrin and Beresford (2000) described the basic theories of ANN modeling as the following paragraphs.

An ANN is a biologically inspired computational model formed from several of single units, artificial neurons, connected with coefficients (weights) which constitute the neural structure. They are also known as processing elements (PE) as they process information. Each PE has weighted inputs, transfer function and one output. PE is essentially an equation which balances inputs and outputs. There are many types of neural networks designed by now and new ones are invented every week but all can be described by the transfer functions of their neurons, by the training or learning algorithm (rule), and by the connection formula.

A single-layer neuron is not able to learn and generalize the complex problems. The multilayer perceptron (MLP) overcomes the limitation of the single-layer perceptron by the addition of one or more hidden layer(s) (Fig. 1). The MLP has been proven to be a universal approximator (Cybenko, 1989). In Fig. 1, a feedforward multilayer perceptron network was presented. The arriving signals, called inputs, multiplied by the connection weights (adjusted) are first summed (combined) and then passed through a transfer function to produce the output for that neuron. The activation (transfer) function acts on the weighted sum of the neuron's inputs and the most commonly used transfer function is the sigmoid (logistic) function.

The way that the neurons are connected to each other has a significant impact on the operation of the ANN (connection formula). There are two main connection formulas (types): feedback (recurrent) and feedforward connection. Feedback is one

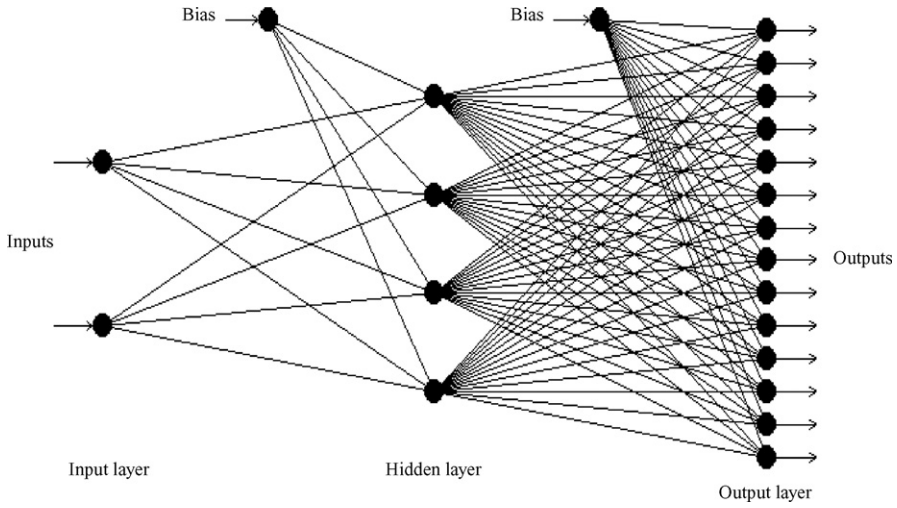

Fig. 1. Schematic representation of a multilayer perceptron feedforward network consisting of two inputs, one hidden layer with four neurons and 14 outputs.

type of connection where the output of one layer routes back to the input of a previous layer, or to same layer. Feedforward does not have a connection back from the output to the input neurons.

There are many different learning rules (algorithms) but the most often used is the Delta-rule or backpropagation (BP) rule. A neural network is trained to map a set of input data by iterative adjustment of the weights. Information from inputs is fed forward through the network to optimize the weights between neurons. Optimization of the weights is made by backward propagation of the error during training or learning phase. The ANN reads the input and output values in the training data set and changes the value of the weighted links to reduce the difference between the predicted and target (observed) values. The error in prediction is minimized across many training cycles (iteration or epoch) until network reaches specified level of accuracy. A complete round of forward–backward passes and weight adjustments using all input–output pairs in the data set is called an epoch or iteration. If a network is left to train for too long, however, it will be overtrained and will lose the ability to generalize.

The number of PEs per layer, as well as the number of layers, greatly influences the prediction abilities of the MLP. Too few of them hinder the learning process, and too many of them can depress the generalizing abilities of the MLP through overfitting or memorization of the training data set.

In this study, we focused on the learning situation known as supervised learning, in which a set of input/output data patterns is available. Thus, the ANN has to be trained to produce the desired output according to the examples.

In order to perform a supervised training we need a way of evaluating the ANN output error between the actual and the expected output. A popular measure is the mean squared error (MSE) or root mean squared error (RMSE):

$$\mathrm{MSE} = \frac{\sum (y_i - o_i)^2}{n} \quad (1)$$

$$\mathrm{RMSE} = (\mathrm{MSE})^{1/2} \quad (2)$$

where y_i is the predicted value, o_i the observed value, and n is the number of data set.

As previously mentioned, there are numerous algorithms, which can be used for training ANNs. The following algorithms were used in this study to train ANNs:

(1) Gradient descent backpropagation algorithm: this algorithm is one of the most popular training algorithms in the domain of neural networks. It works by measuring the output error, calculating the gradient of this error, and adjusting the ANN weights (and biases) in the descending gradient direction. Hence, this method is a gradient descent local search procedure (Rumelhart et al., 1986). This algorithm includes different versions such as: (1a) standard or incremental backpropagation (IBP): the network weights are updated after presenting each pattern from the learning data set, rather than once per iteration (Freeman and Skappura, 1991; CPC-X Software, 2004); (1b) batch backpropagation (BBP): the network weights update takes place once per iteration, while all learning data pattern are processed through the network (Hagan et al., 1996; CPC-X Software, 2004); (1c) quick propagation (QP): QP is a heuristic modification of the backpropagation algorithm. It is proved much faster than IBP for many problems. QP is also defined as: mixed learning heuristics without momentum, learning rate optimized during training (Fahlman, 1988; CPC-X Software, 2004).

(2) Levenberg–Marquardt backpropagation: the Levenberg–Marquardt (LM) algorithm approximates to the Newton method and has been also used for ANN training. The Newton method approximates the error of the network with a second order expression, which contrasts to the former category which follows a first order expression. LM is popular in the ANN domain (even it is considered as the first approach for an unseen MLP training task) (Hagan and Menhaj, 1994).

(3) Genetic algorithm: genetic algorithm (GA) is a stochastic general search method. It proceeds in an iterative manner by generating new populations of individuals from the old ones. This algorithm applies stochastic operators such as selection, crossover, and mutation on an initially random population in order to compute a new population (Holland, 1975). The search features of the GA contrast with those of the gradient descent and LM in that it is not trajectory-driven, but population-driven. The GA is expected to avoid local optima frequently by promoting exploration of the search space, in opposition to the exploitative trend usually allocated to local search algorithms like gradient descent or LM.

2. Materials

Theophylline anhydrous was supplied by BASF (Aktiengesellschaft, Germany), Eudragit® RS and Eudragit® L100-55 were kindly provided by Röhm Pharma GmbH (Darmstadt, Germany). Microcrystalline cellulose (Avicel® PH 101) was supplied by FMC Corp. (Philadelphia, USA). Pectin HM (High Methoxylated) from citrus fruit and chitosan (low molecular weight, 84.7% deacetylated, viscosity of 1% (w/v) solution in acetic acid 1% (v/v), 34 mPas) were purchased from Sigma–Aldrich (Dorset, UK). Pectinex® Ultra SP-L (pectinolytic enzymes, extracted from *Aspergillus Niger* with an activity of 26,000 pg/ml at pH 3.5) was donated kindly by Novo Nordisk Ferment (Dittingen, Switzerland). The other materials were of pharmaceutical or analytical grades, and used as received.

3. Methods

3.1. Preparation of polyelectrolyte complex (PEC) and coating solution

Chitosan was dissolved in 0.1 M acetic acid and pectin was dissolved in distilled water. The chitosan solution was added to the pectin aqueous solution slowly. After all of the chitosan was added, the pH of the mixture was adjusted to 5.4 and allowed to react 1 h under mechanical stirring. After mixing, it was filtered and the filter cake (PEC) was washed with

0.1 M acetic acid to remove free chitosan and also washed with warm water until the filtrate became neutral. After being dried at 50–60 °C, the PEC was dissolved in formic acid (10%, v/v). Then, appropriate amounts of the solution of pectin–chitosan complex (pectin/chitosan ratio: 2/1) were added to the aqueous dispersion of Eudragit® RS30D. Eudragit® RS30D was previously mixed for 30 min with triethyl citrate (TEC) as plasticizer (10% (w/w), related to the solid polymer content of Eudragit® RS30D). Eudragit® RS is a slightly cationic hydrophilic polymethacrylate which requires the addition of 10–20% of plasticizers in order to reduce the minimum film forming temperature (MFFT) below 20 °C.

3.2. Preparation and coating of theophylline pellets

The preparation and coating of theophylline pellets were discussed in details in our previous work (Ghaffari et al., 2006a). Briefly, theophylline pellets were prepared by the extrusion-spheronisation method (laboratory unit, Gabler GmbH, Germany). A 100-g batch of theophylline anhydrous and microcrystalline cellulose (Avicel® PH 101), in a weight ratio of 6:4 was mixed, kneaded and loaded into the extrusion (extruder type E-35, Gabler GmbH). The housing of the extruder was cooled with water at 18 °C. The extrudates were immediately transferred to a spheroniser (spheroniser type R-250, Gabler GmbH) equipped with a crosshatch plate. Extrudates prepared with above procedure were spheronised for 10 min at 1000 rpm. Pellets were collected and dried in an oven at 50 °C for 24 h after which sieve analysis was done and the fraction of 710–840 μm was separated for coating. In order to prevent the batch-to-batch variability of the pellets from affecting the different batches of coated pellets, the sieve-cuts of 710–840 μm from several batches of theophylline pellets were pooled together and blended. The pellets for coating were then taken out from this bulk. For coating, a 100-g batch of theophylline pellets were coated with a combination of Eudragit® RS and pectin–chitosan complex at different coating level in a fluidized-bed coating apparatus (Uni-Glatt, Glatt GmbH, Germany) equipped with a Wurster column (model formulations are mentioned in Section 3.4). After coating, the coated pellets were gently fluidized for about 5 min after which they were cured at 60 °C for 24 h.

3.3. Determination of drug content (assay) and release

Drug contents of the uncoated and coated pellets were determined by HPLC analysis (US Pharmacopeia, 2005). The coating level of each formulation of coated pellets was calculated taking into account the decrease of drug loading of the coated pellets in comparison with the initial drug loading of the uncoated ones (Amighi and Moës, 1996).

It was observed that isolated films of pectin–chitosan complex were dissolved easily after approximately 10 min in acidic medium, while they were slowly swelled and showed no dissolution in the other two phosphate buffer media (Ghaffari et al., 2006a). Therefore, it was assumed that pectin–chitosan complex in the mixed ternary blends (pectin/chitosan/Eudragit® RS) may be dissolved and leached from the mixed-film coatings. Hence, for bimodal drug delivery, an additional outer enteric-coating was necessary to prevent the drug release from coated pellets in the stomach, since the pectin–chitosan complexes incorporated in the film coating dissolve easily under acidic condition. So, after capsule filling of coated pellets, capsules (hard gelatin capsules no. 2) were enteric-coated by the alcoholic solution of Eudragit® L100-55 polymer in a conventional coating pan (100 mg of Eudragit® L100-55 per each capsule).

Drug release studies were conducted using the USP basket method (apparatus I) at 100 rpm and 900 ml of dissolution fluid at 37 ± 0.5 °C. Three capsules filled with each formulation were tested individually in 0.1 M HCl (pH 1.5) for the first 2 h, phosphate buffer solution pH 7.4 for the second 3 h and phosphate buffer solution pH 6.0 containing pectinolytic enzymes (4 ml/l) for the last 5 h. These media were chosen to mimic the conditions pertaining to the stomach, small intestine and colon, respectively.

3.4. Study design

As it was reported the classical experimental designs are inappropriate for ANN modeling (Plumb et al., 2002); therefore, a two-factor, three-level experimental design with additional points (i.e. selected randomly around the values of variable levels) was used in this study. The studied causal factors (independent variables) were the coating level amount (X_1) and the amount of pectin–chitosan in coating solution (X_2). Both X_1 and X_2 were expressed as grams for a 100-g batch of uncoated pellets. The dependent variables or responses were percentage of drug release at 130, 150, 180, 210, 240, 300, 310, 330, 360, 390, 420, 480, 540 and 600 min. Thus, as model formulation, 17 kinds of coated pellets were prepared according to the following custom-designed experiments (Table 1).

Table 1
Design layout of two-factor, three-level experimental study with additional points

Formulation number	Variable levels in coded form		Variable factors	
	X_1	X_2	X_1	X_2
F_1	–	–	4.3	0.2
F_2	0	0	4.0	0.4
F_3	**0**	**1**	**4.0**	**0.7**
F_4	0	2	4.0	1.0
F_5	1	0	6.5	0.4
F_6	1	1	6.5	0.7
F_7	1	2	6.5	1.0
F_8	–	–	6.4	1.3
F_9	2	0	8.0	0.4
F_{10}	2	1	8.0	0.7
F_{11}	2	2	8.0	1.0
F_{12}	–	–	7.4	1.5
F_{13}	–	–	**6.5**	**0.5**
F_{14}	–	–	6.4	0.9
F_{15}	–	–	7.8	0.6
F_{16}	–	–	**4.0**	**0.0**
F_{17}	–	–	6.5	0.0

Bold characters are selected three formulations for testing constructed ANN models; coded values: 0, low level; 1, intermediate level; 2, high level.

3.5. Computer program

Commercially available NeuralPower, professional version 2.5, (CPC-X Software, 2004), was used throughout the study with a Pentium® 4 personal computer. This software is a Windows®-based package, which supports several types of training algorithms along with a powerful, easy to use ANN package. NeuralPower operates via a graphical user interface (GUI) that enables the user to load the training and test sets, design the network architecture, select the training algorithm and generate the individual models for each output variable in a single operation. Scaling of the data to the range 0–1, necessary to prevent saturation of the transfer function, is performed automatically within NeuralPower.

Two causal factors corresponding to different amounts of coating level (X_1) and pectin–chitosan in coating solution (X_2) were used as the inputs to the network. Percentage of drug release at different dissolution time points were used as the expected output of the network (14 nodes). The number of hidden layers is difficult to decide, but typically, no more than one hidden layer is used in a network (Hush and Horne, 1993). One hidden layer was utilized in the current study. The reason we chose one hidden layer is that we did not observe a significant difference in generalization or improvement in performance by increasing the number of hidden layers to two or three. In the current study, four hidden nodes were chosen. To select the number of hidden nodes, we started with 1 hidden node and we gradually increased the number of nodes until a network of minimum RMSE for test data set was attained. Further increase in hidden nodes produced higher error. A MLP feedforward network, consisting of one hidden layer with four neurons, was built with 2 inputs and 14 outputs (often denoted as a 2:4:14 network). The architecture of MLP network used in this study was depicted in Fig. 1. The sigmoidal function was used as the transfer (activation) function for the hidden and output layer nodes. In the hidden layer and output layer, the PE sums its weighted inputs from the previous layer and then applies the sigmoidal function to compute its output to the following layer according to the below equations:

$$Y_q = \sum W_{pq} X_p \tag{3}$$

$$f(y_q) = \frac{1}{1 + \exp(-\alpha y_q)} \tag{4}$$

where W_{pq} is the weight connecting node q of the current layer to node p of the previous layer and X_p is the output value of the previous layer. $f(y_q)$ is the activation function of node q in the current layer. Alpha (α) is called slope and a parameter relating to the shape of the sigmoidal function. Non-linearity of the sigmoidal function is strengthened with an increase in α. In our study, this parameter was set to $\alpha = 1$.

Besides, two other parameters should be defined for the backpropagation ANN model training: learning rate and momentum coefficient. The learning rate is an adjustable factor that controls the speed of the learning process. With a faster learning rate, the ANN model will learn faster. However, if the learning rate is too high, the oscillations of weight changes can impede the convergence of the error surface, and may lead to overshooting a near-optimal weight factor. In contrast, if the learning rate is too slow, the ANN model may be caught in a local error minimum instead of the global minimum. Momentum coefficient is used in weight updating for backpropagation ANN to avoid local minima and to reduce oscillation of weight change. To obtain faster learning without oscillation, the weight change is related to the previous weight change to provide a smooth effect. The momentum coefficient determines the proportion of the last weight change that is added into the new weight change. The following simplified relationship presented by Erb (1993) points out the effects of these two parameters on the weight adjustment:

$$\text{new weight change} = \text{learning rate} \times \text{error} + \text{momentum coefficient} \times \text{last weight change} \tag{5}$$

In NeuralPower software, the default settings for learning rate and momentum coefficient are 0.15 and 0.8; 0.1 and 0.4, for IBP and BBP, respectively. The default learning rate for QP is 0.8 and momentum coefficient is not employed in QP mode. With both incremental and batch backpropagation, the learning rate and momentum coefficient are kept constant throughout training. But, quick propagation is an adaptive (variable) learning rate which will attempt to keep the learning step size as large as possible while keeping learning stable. The learning rate of QP is made responsive to the complexity of the local error surface. It should be mentioned here that 0.8 is the starting or initial value for the learning rate. This procedure may increase the learning rate, but only to the extent that the network can learn without large error increases. Thus, a near-optimal learning rate is obtained for local terrain. When a larger learning rate could result in stable learning, the learning rate is increased. When the learning rate is too high to guarantee a decrease in error, it decreases until stable learning resumes.

At the start of the training run, both weights and biases were initialized with random values typically in the range of −1 to 1. Each algorithm was run five times separately for ANN training with random initialization of weights.

To evaluate an ANN, we split the data set into two training and test subsets. A set of outputs and causal factors (formulation variables) was used as tutorial data (training runs F_1, F_2, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}, F_{11}, F_{12}, F_{14}, F_{15}, F_{17}) and fed into the computer (Table 1). Five algorithms were used for ANNs training. In order to limit overtraining, formulations of F_3, F_{13} and F_{16} were used as test data set (bold characters in Table 1). Two formulation variables (X_1 and X_2) were used as the ANN input, and the percentage of drug release at 14 different time points as the output. The training set was used to train the network, and the test set was used to determine the level of generalization produced by the training set and to monitor overtraining the network, each with corresponding RMSE. Neural learning is considered successful only if the system can perform well on test data on which the system has not been trained. Therefore, emphasis is on the capabilities of a network to generalize from input training samples, not to memorize them. The experimental test data in this study were chosen so that each test point was within the same

range in the factor space as the experimental points used to train networks (interpolation). The NeuralPower program was set to automatically stop training when the RMSE of test set prediction achieved a minimum. This feature of NeuralPower software, i.e. early stopped training, increases the probability of finding the global minimum RMSE of the test set. The test RMSE is the error in predicting the percentage of drug release data for the three test formulations that were excluded from the 17-formulation data set. In other words, convergence of training (termination of training) was determined when the network showed a minimum RMSE in predicting the responses in the test data set. Convergence is the process of searching a set of weight factors for the ANN model so that the prediction errors can be reduced to a minimum (Sun et al., 2003).

4. Results and discussion

Release profiles of theophylline from all formulations are shown in Figs. 2–4. A wide variation among the release profiles of formulations could be observed, indicating that they were greatly affected by changes made in the amounts of causal factors. The two separate phases (i.e. bimodal) in dissolution profiles of some formulations (e.g. F_1, F_2, F_{11} and F_{12}) can be seen in those figures.

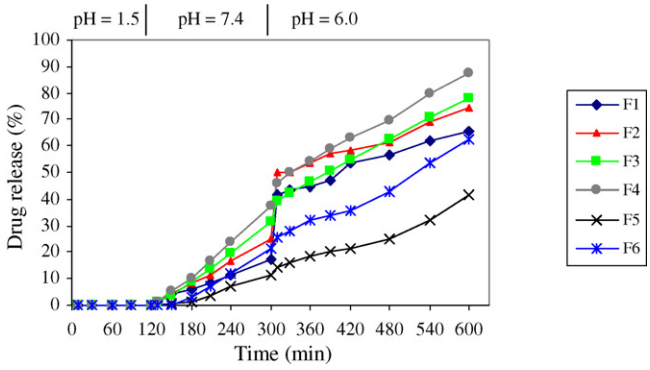

Fig. 2. Cumulative percent of drug release (the mean of three determinations) *vs.* time profiles for enteric-coated F_1–F_6 formulations. The vertical lines denote the time points for the changes in dissolution media.

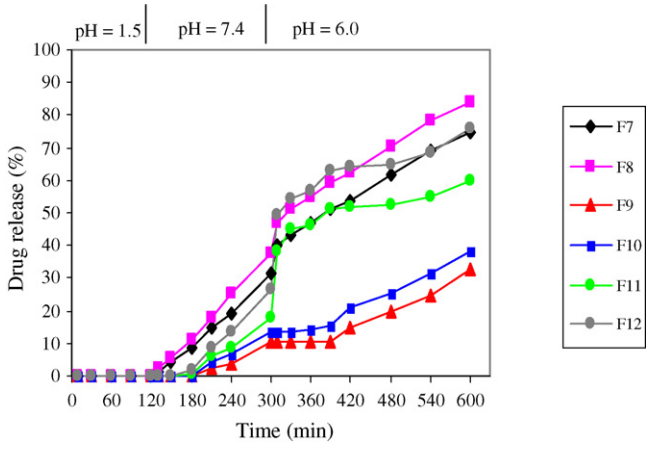

Fig. 3. Cumulative percent of drug release (the mean of three determinations) *vs.* time profiles for enteric-coated F_7–F_{12} formulations. The vertical lines denote the time points for the changes in dissolution media.

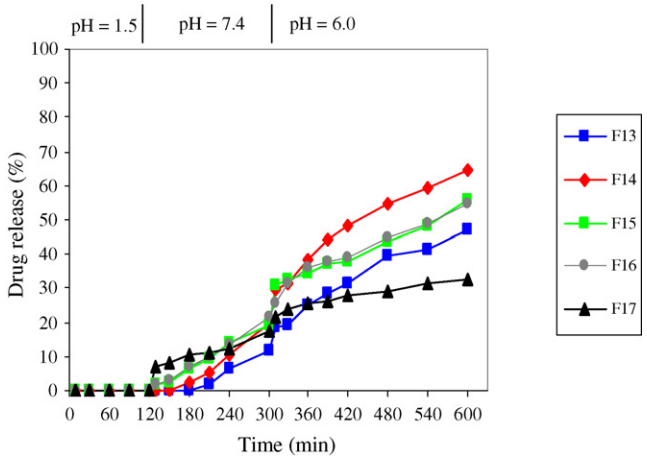

Fig. 4. Cumulative percent of drug release (the mean of three determinations) *vs.* time profiles for enteric-coated F_{13}–F_{17} formulations. The vertical lines denote the time points for the changes in dissolution media.

In the earlier study (Ghaffari et al., 2006b), possible explanations for the mechanism(s) of bimodal drug release and/or initial burst release at the third phase, i.e. 300–600 min were discussed. For F_1, F_2, F_{11} and F_{12}, there are considerable burst drug releases at the beginning of third phase. During the second phase of drug release profile, i.e. 120–300 min, the aqueous medium penetrates through the pellets' coating (particularly via pectin–chitosan portion) and starts dissolving the core drug. Therefore, inside of the pellet becomes richer of drug molecules and its concentration gradient gets higher than the external environment. Consequently, due to the positive gradient, the drug molecules are transported from the inside to the outside of the coated pellet. After phase two, due to the exposure of the mixed-film to the pectinolytic enzymes, pectin is degraded and leached from the coating and therefore a sudden drug release (burst release) and a bimodal release pattern could be observed. For bimodal drug delivery, a balance should exist between the coating thickness and pectin–chitosan amount. Either an inadequate amount of pectin–chitosan in the coating or a high coating level (thickness) may be responsible for low or no burst drug release. In other words, at a high coating level, a film barrier predominately governs drug release while all pectin–chitosan aqueous channels are blocked.

4.1. ANN model training with gradient descent algorithm

There are different training algorithms implemented in NeuralPower software and their performances and features were studied as follows: at first, gradient descent backpropagation algorithms in three versions were used to train neural networks.

As previously mentioned, the default values for learning rate and momentum coefficient in NeuralPower are 0.15 and 0.8; 0.1 and 0.4, for IBP and BBP, respectively. It was observed that if learning rate is greater than 0.15 for IBP, the oscillation of weight change hinder the convergence and lead to overshooting a near-optimal solution. Thus, we left two aforementioned train-

ing parameters (learning rate and momentum coefficient) as the default settings.

4.1.1. Performance criteria

There are various methods available for evaluating model performance, including numerical indicators (Table 2) and graphical representations (Figs. 5 and 6). In this study, the authors used some numerical measures as follow: the epoch that gave the lowest RMSE for the test set was selected as the optimum epoch at stopping point. The comparison between the three gradient descent versions is also made by considering the CPU time elapsed at the end of training. Table 2 represents the statistical measures and performance indices for each training algorithm.

Precision and bias were evaluated through RMSE for test set and mean prediction error, respectively. The mean prediction error (MPE) or bias is calculated as Eq. (6):

$$\mathrm{MPE} = \frac{\sum(y_i - o_i)}{n} \tag{6}$$

where y_i is the predicted value, o_i the observed value, and n is the number of data set. Statistical measure of the test RMSE for each mode of training algorithm was compared using a paired t-test. A p value < 0.05 was considered to be statistically significant.

The generalization (prediction) ability of the ANN models was verified by three test data sets. As can be observed in Table 2, better average precision of prediction (test RMSE) were seen with IBP and BBP as compared with QP (3.2688 and 3.2194 *versus* 5.4988). Moreover, IBP and BBP showed a smaller mean prediction error (bias) than that of QP (2.0864 and 2.0770 *versus* 3.4567). As described previously, the training was terminated when the minimum RMSE on the test data set was attained. The number of training epochs and time elapsed for total epochs reflected when the training was terminated, differed drastically among gradient descent modes (691 epochs and 14.8 s for IBP; 223 epochs and 4.2 s for BBP; 202 epochs and 3.6 s for QP, respectively). IBP and BBP outperformed QP in terms of prediction and generalization ability. In other words, IBP and BBP appeared to be less biased and more precise, as compared with QP. No significant difference was found in the predictive ability of IBP and BBP, although, the convergence speed of BBP is three- to four-fold higher than that of IBP. The CPU time required by the BBP algorithm is only 28.38% of that required by the IBP algorithm.

Fig. 5 illustrates the relationship between the ANN predicted (calculated) and observed value (drug percent release) for all three test data set at 14 dissolution times, implemented with gradient descent algorithms. Fig. 5a and b shows, respectively the model with higher (BBP, test RMSE = 3.0668) and lower (QP, test RMSE = 6.5918) prediction ability among all independent runs of three training versions. Lower R^2 value ($R^2 = 0.9344$) was obtained for QP indicating that the prediction ability of this algorithm was not as good as BBP. BBP (Fig. 5a) was considered to provide the higher overall predictive performance ($R^2 = 0.9815$). Although R^2 for QP is lower than BBP, it is still considered as a good fit. Generally, for evaluation of the robustness of a modeling technique, R^2 of test data set should be computed and

Table 2
Statistical measures and comparison between performance index of five training algorithms

Performance index	Training algorithms				
	IBP	BBP	QP	LM	GA
Average RMSE for test set	3.2688 (3.0876, 3.4501)[a]	3.2194[b] (3.0194, 3.4193)	5.4988[c] (4.7284, 6.2692)	4.0998[c] (3.3045, 4.8951)	7.9123[c] (6.8016, 9.0230)
Average RMSE for training set	5.1901	5.3488	4.4814	2.6069	5.3275
Average mean prediction error	2.0864	2.0770	3.4567	2.6893	5.9821
Average number of epochs at the end of training	691	223	202	79	1200
Average CPU time elapsed at the end of training (s)	14.8	4.2	3.6	6	25

$n = 5$ (ANN training with each algorithm was run 5 times).
[a] Data in parentheses represent the 95% confidence interval of the average.
[b] Not significantly different from IBP, $p > 0.05$.
[c] Significantly different from IBP, $p < 0.05$.

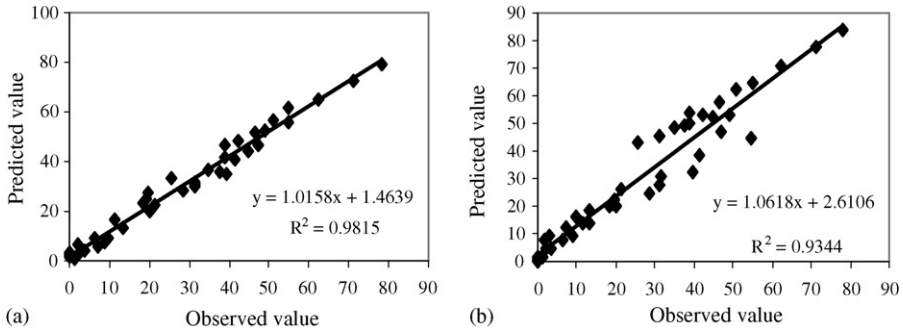

Fig. 5. Scatter plots of observed *vs*. ANN predicted drug percent release from two modes of gradient descent training approach: (a) higher prediction ability (BBP, test RMSE = 3.0668) and (b) lower prediction ability (QP, test RMSE = 6.5918). Lines represent the linear regression of observed and predicted value.

$R^2 > 0.9$ can be regarded as a good overall fit (Bourquin et al., 1998c).

4.2. ANN model training with Levenberg–Marquardt algorithm

Generally, LM is one of the most popular tools for non-linear minimum mean squares problems. Because of the properties of fast convergence and stability, the method has been employed in many modeling problems (Sakamoto et al., 2005).

Levenberg–Marquardt (LM) is another type of backpropagation training algorithm (Demuth and Beale, 2002). Neural networks with the same above-mentioned architectures trained by the LM were judged to show the overall predictive ability over the test set. Here, for LM-trained ANNs, formerly stated performance indices were determined. Average RMSE for both of the training set and test set were obtained 2.6069 and 4.0998, respectively. Average number of epochs and CPU time elapsed at the end of training, respectively were obtained 79 and 6 s. It is worth mentioning that the LM algorithm did not perform as well as IBP and BBP on this problem (Table 2). LM algorithm achieves lower precision in terms of predictive performance when compared with gradient descent algorithms (with the exception of QP). An interesting observation is that LM with the lower RMSE value for the training set does not result in better precision of test set prediction as compared with IBP and BBP.

As can be observed in Fig. 6a, the observed *versus* predicted values using LM–ANN methodology results an $R^2 = 0.9517$.

4.3. ANN model training with genetic algorithm (GA)

Genetic algorithm is a kind of search and an optimized algorithm that have been produced from stimulated biologic heredities and long evolutionary processes of creatures. It simulates the mechanism of "survival competition", reproduction, mating, and dissociation in natural selection and natural heredity procedures. Each possible solution to problems is taken as an individual among population, and each individual is coded as character string; each individual is evaluated in response to predefined objective functions. Three of its elemental operators are selection, crossing, and mutagenesis (Genshe and Xinhai, 1994). Its main features are as follows: (1) GA is to acquire the optimal solution or quasi-optimal ones through a generational search rather than a one-point search (gradient descent and LM); (2) GA is capable of global optimum searching; (3) GA is a parallel process to population change, and provides intrinsic parallelism (Fang et al., 2005).

4.3.1. Genetic operations definition

In the current work, initial population was chosen randomly. The population size indicates the number of solutions in the workspace. Larger size produces more accurate solution and the global optimum is likely to be found, but computational time is longer correspondingly. In general, the size of the population is between 20 and 100. It was set in 20 in our GA implementation.

4.3.1.1. Selection or reproduction. It is a process in which individuals are copied into the next population (generation)

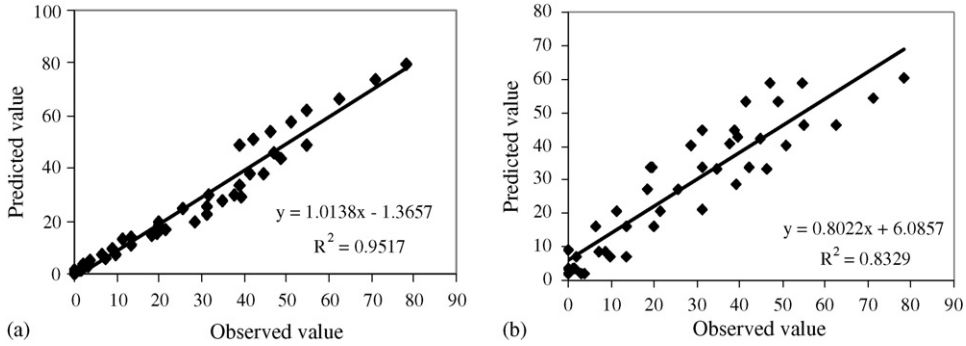

Fig. 6. Scatter plots of observed *vs*. ANN predicted drug percent released from: (a) Levenberg–Marquardt (LM) training algorithm and (b) genetic algorithm (GA). Lines represent the linear regression of observed and predicted value.

according to their fitness, i.e. the "better" individuals survive to reproduce and the less highly fit individuals "die". Fitness of an individual is determined by the objective function which is to be optimized. There are several methods for determining how effectively a given individual "competes" in the reproduction process. In this paper, *absolute top mate selection* is used. In this type of selection, the first parent is selected by the fittest individual (chromosome) of all iteration and the second parent is selected randomly (CPC-X Software, 2004). Besides, the individual smallest fitness value in the next generation is replaced by the individual highest fitness value in the previous one so that the optimal individual is not destroyed during the evolutionary process.

4.3.1.2. Crossover. As mentioned above, GA evolves a population of individuals according to the process of natural selection. During this process, genetic operators create new (child) individuals from highly fit old (parent) individuals. Crossover (also referred to as recombination) is one of the genetic operators and is a key to the power of the genetic algorithm (Holland, 1975). In other words, the optimum in the existing population can be found during the above operation but the individual that is different from the previous one cannot be produced. Then crossover can do this by swapping characters according to some probability to create new individuals so that "better" ones can be produced. The probability determines the frequency of crossover. The higher the frequency, the faster the optimum speed reaches, but is probable that convergence occurs before the optimum is reached. In general, the probability is between 0.4 and 0.9. In our proposed algorithm, *intermediate crossover* is done which is a kind of linear combination of the two parents (CPC-X Software, 2004). Besides, the crossover rate (i.e. the probability that two chromosomes will swap their genes/characters) was set in 0.8.

4.3.1.3. Mutation. The reproduction and the crossover can find the optimum in the existing character arrays, but fail to produce new characters to find the best solution because of premature convergence. The mutation changes the characters in an individual with a small probability between 0.001 and 0.4. Mutation brings in new possibilities for improvements and ensures that some important information is produced during crossover and that reproduction should not be lost. Here, the mutation rate (the probability that one or more of the individual's genes/characters will be changed) was fixed as 0.1.

In this study, GA is employed as one of the learning algorithms in order to find the optimal weights of the network. In summary, *absolute top mate selection*, *intermediate crossover*, and mutation in a fixed generation processing environment were used for GA implementation. Numerous runs were made with crossover rates ranging from 0.5 to 0.8, population sizes ranging from 20 to 100 and mutation rates ranging from 0.05 to 0.3. The best training result was obtained as setting mentioned values for those genetic operators. Termination of training was determined by the test set, and the evolutionary impetus is provided by the training set. For all case studies, at each time slot, the average of the results obtained after five independent runs.

Again, for GA-trained ANNs, aforementioned performance indices were applied. Average RMSE for both training set and test set were obtained 5.3275 and 7.9123, respectively. Average number of epochs and CPU time at stop training, respectively were obtained 1200 and 25 s. The use of the number of epochs as a measure can be misleading because of the difficulties in defining the epoch for both evolutionary methods (GA) and backpropagation methods (gradient descent and LM). Statistical measures and comparison between performance index of five training algorithms, i.e. IBP, BBP, QP, LM and GA were represented in Table 2.

The plot of ANN-trained with GA was highly scattered with $R^2 = 0.8329$ demonstrating that the model was not predictive (Fig. 6b). The comparison of the correlation coefficients (R^2, see Figs. 5 and 6) show a clear superiority of the backpropagation algorithms compared to the GA method.

ANN-trained models could also be assessed according to the scheme proposed by Plumb et al. (2002). In this procedure, the output values predicted by the ANN for the test data set were plotted against the corresponding observed values (Figs. 5 and 6). Linear regression analysis of the resulted agreement plots was then performed to calculate the gradient (m), intercept (c) and goodness of fit (R^2). Plumb and co-workers proposed that these regression parameters provide a simple, intuitive measure of model performance. For a well-trained model, m, c and R^2 ($|m, c, R^2|$) for the test data set should approximate 1, 0 and 1, respectively. The agreement vectors, i.e. $|m, c, R^2|$ of |1.016, 1.464, 0.9815|, |1.062, 2.611, 0.9344|, |1.014, −1.366, 0.9517| and |0.802, 6.086, 0.8329| were obtained for BBP, QP, LM and GA, respectively.

Indeed, in the GA-based training, the evolution of connection weights has been introduced into ANNs. The aim is to find a near-optimal set of connection weights for an ANN with previously mentioned architecture using GA. GA always searches for a globally optimal solution, while backpropagation methods can only find a local optimum in a neighborhood of the initial solution. Hence, GA training has slower convergence rate as compared with gradient descent and LM. It can be observed that GA training is far less precise and efficient when compared with IBP, BBP and LM algorithms in this problem. In general, GA has a good search accuracy (it approaches the globally optimal solution irrespective of the diverse starting conditions) and a poor search precision (it often fails in finding the very optimum solution), while the traditional local modeling techniques have a good search precision and a poor search accuracy (Leardi, 2001).

Several authors (Sutton, 1986; Whitley et al., 1990) have reported that backpropagation has drawbacks due to its use of gradient descent while it often gets trapped in a local minimum of the error function. It is incapable of finding a global minimum if the error function is multimodal and/or nondifferentiable. Conversely, in this study, no such limitation was found and the gradient descent backpropagation algorithms, especially IBP and BBP method were converged to optimum very well.

Fig. 7 depicts the comparative evaluation of three algorithms (BBP, LM and GA) implemented for training ANN in terms of neural network predicted (calculated) and observed drug release for F_3 formulation at 14 dissolution times. As shown in Fig. 7,

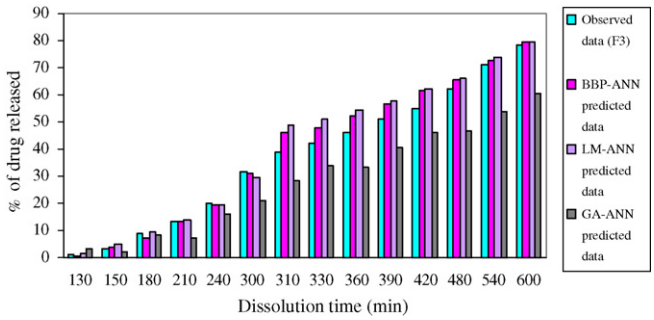

Fig. 7. Comparison among various neural network training algorithms in terms of neural network predicted and observed drug release for F_3 formulation at 14 dissolution times.

in comparisons of 14 observed and predicted dissolution rates using the BBP, LM and GA training algorithms, the overall prediction abilities of those were in the order of: BBP > LM > GA.

4.4. Response surface and contour plots

Response surface plots of the effect of coating level amount (X_1) and the amount of pectin–chitosan (X_2) on the percentage of drug release at 130, 150, 180, 300, 310 and 420 min of dissolution testing, predicted using BBP–ANN, are presented in Fig. 8a–f and corresponding contour plots in Fig. 9a–f.

According to Fig. 8, the non-linear relationships between two causal factors and percentage of drug release were represented well with response surfaces of the model based on BBP–ANN. Unexpectedly, increasing the amount of pectin–chitosan (X_2) and decreasing the amount of coating level (X_1) resulted in a decrease of drug release at 130 min. However, the effect of X_2 on the release response is more predominant than X_1. At dissolution time of 150 min, an increase of X_1 resulted in a decrease of drug release. Influence of X_2 on the response surface seems to be negligible at this time. After 150 min, response surfaces generated by the BBP-trained ANN exhibited relatively plane surfaces (i.e. parallel contours) for all responses and with decreasing X_1 and increasing X_2, the percentage of drug release was increased. The above observations can be explained as follows.

The pectin/chitosan/Eudragit® RS mixed-film is a swellable film. It was found that the higher amount of pectin–chitosan in the mixed-film, the more the drug release (Ghaffari et al., 2006b). This effect was due to the swelling-hydration ability of pectin–chitosan moiety in the mixed-film. It can be presumed that, at the beginning of dissolution and prior to 180 min, there was no steady-state condition in the polymeric film; hence, the influences of two causal factors on the response were unreliable. As it was expected, hereafter, drug release rates could be predicted according to the level of pectin–chitosan and coating weight gain. An increase in coating levels and decrease in the amount of pectin–chitosan generally retarded the drug release rate.

4.5. Analysis of relative importance of causal factors

One of the features of NeuralPower software is the analysis of the relative importance for each causal factor (input). Causal relative importance refers to the amount of change in the out-

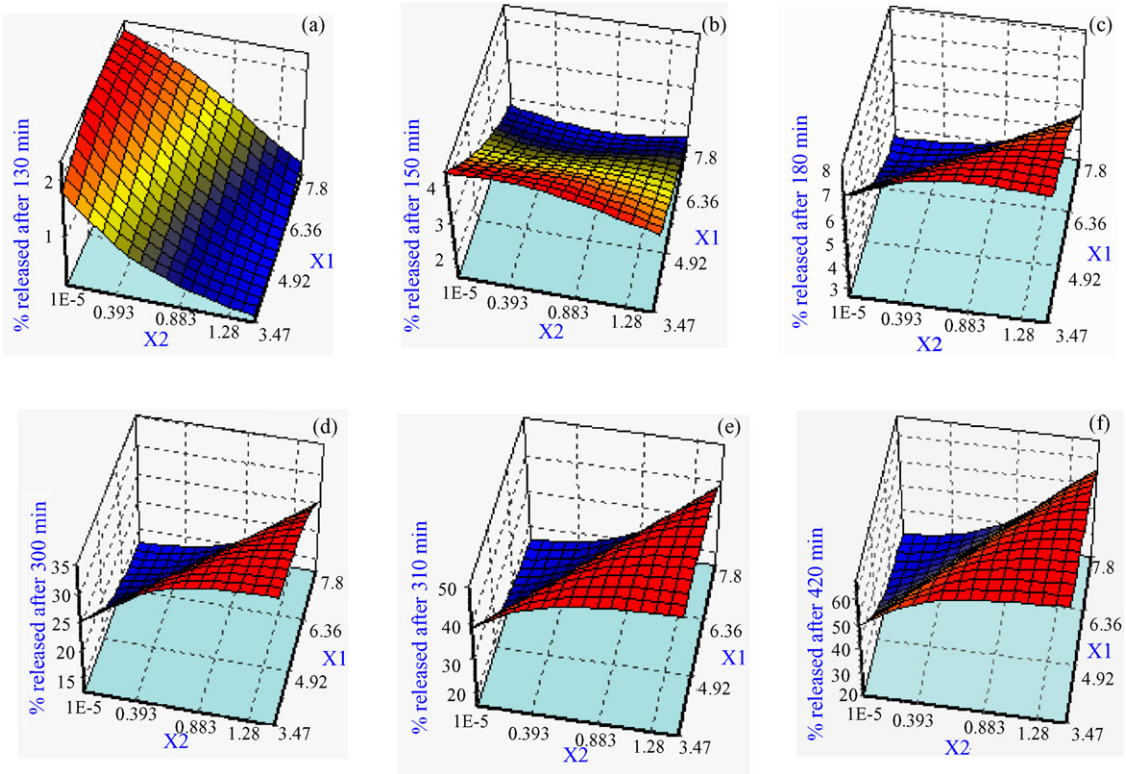

Fig. 8. Response surface plots of coating level amount (X_1) and amount of pectin–chitosan (X_2) on the percentage of drug release after: (a) 130 min, (b) 150 min, (c) 180 min, (d) 300 min, (e) 310 min and (f) 420 min; predicted using BBP–ANN.

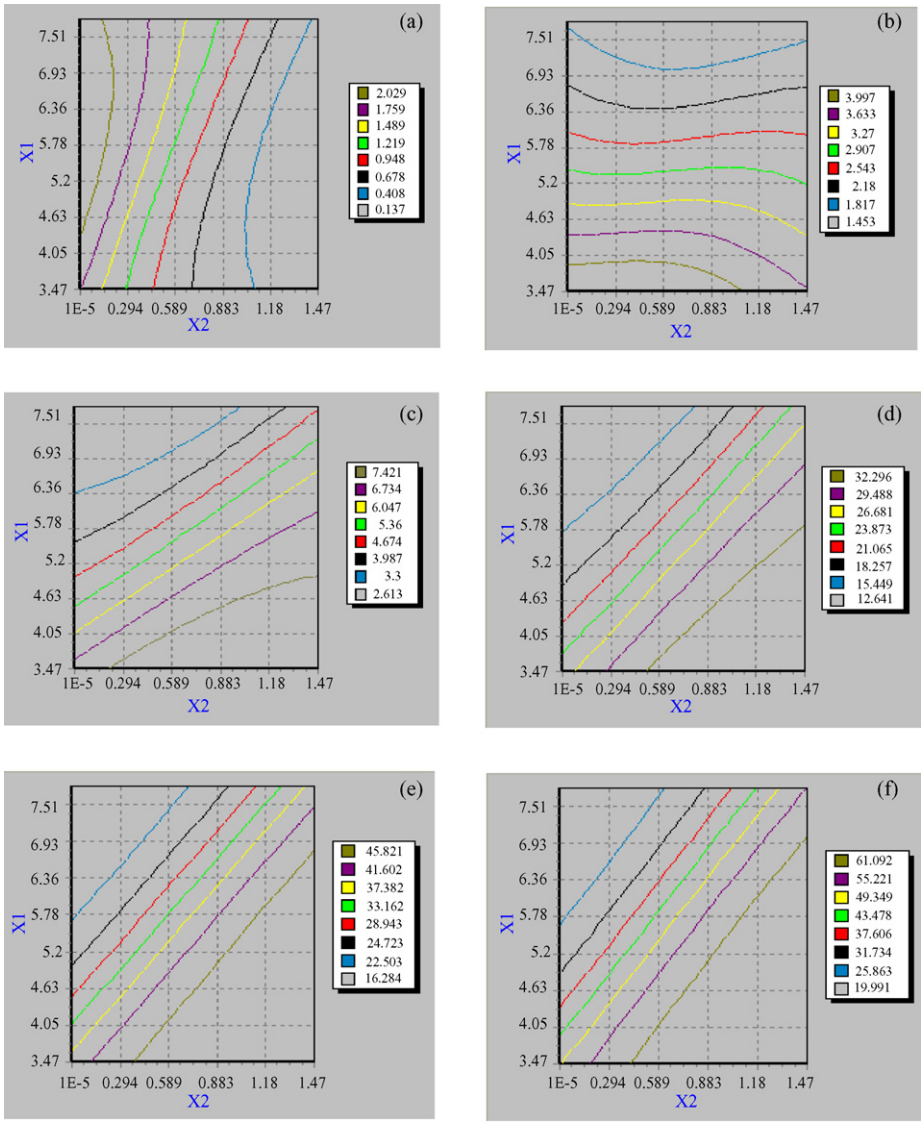

Fig. 9. Contour plots of coating level amount (X_1) and amount of pectin–chitosan (X_2) on the percentage of drug release after: (a) 130 min, (b) 150 min, (c) 180 min, (d) 300 min, (e) 310 min and (f) 420 min; predicted using BBP–ANN.

put corresponding to a given change in an input variable (Sarle, 2000). According to BBP–ANN implementation for our problem, the relative importances were obtained 49.15% and 50.85% for the X_1 and X_2, respectively.

5. Conclusions

At first, key properties of ANN modeling techniques and basic concepts of this methodology were introduced. This methodology is historically based on the attempt to model the way a biological brain processes data.

In the present study, the effects of two causal factors (i.e. amount of pectin–chitosan complex in the coating solution and coating weight gain) on the *in vitro* release profile of theophylline from coated pellets were designed. Five algorithms for ANN training belonging to three broad classes have been evaluated: gradient descent, quasi-Newton algorithm (Levenberg–Marquardt) and genetic algorithm. The degree of generalization or the precision of predictive ability was measured for each training algorithm and their predictive abilities were in the order of: IBP, BBP > LM > QP > GA. In summary, gradient descent backpropagation algorithm in particular incremental and batch backpropagation (IBP, BBP) can be used as the best training algorithms for the modeling and prediction of *in vitro* drug release profiles.

According to BBP–ANN implementation, an increase in coating levels and a decrease in the amount of pectin–chitosan generally retarded the drug release. Moreover, the latter causal factor (i.e. amount of pectin–chitosan) played slightly more dominant role in determination of the dissolution profiles.

Several authors (Sutton, 1986; Whitley et al., 1990) have reported that gradient descent backpropagation has drawbacks due to the possibility of getting trapped in a local minimum of the error function. Some researchers (Yao, 1999; Cortez et al., 2002) have also proposed using evolutionary approach (GA) instead of backpropagation for ANN training. In this study, no

such limitation was found and based on our experience; it seems that gradient descent has a good capability in data modeling. It should be noted that although GA is often useful in a robust evaluation of the best region of a solution space; it is inefficient and ineffective in fine-tuning the local search within our problem's region. The impact of this inability to fine-tune could possibly be limited by integrating a gradient-based algorithm in the late stages of training, thus taking advantage of one algorithm's strength to compensate for the other's weaknesses (hybridization of GA and backpropagation). Furthermore, training by GA often requires relatively long computation time (due to slow convergence). It is possible to reduce the computation time by combining the efforts of local search and global search. This hybrid-training approach can be considered for further work.

Finally, the results of this study indicate that the appropriate selection of training algorithm is essential for successful data modeling by ANN. According to the results, ANNs can be used as a powerful tool in pharmaceutical product formulation, as well as other areas in the pharmaceutical industry, so that the development tasks can be performed rapidly and efficiently with an increase of productivity, consistency and quality.

Acknowledgements

The authors wish to thank Dr. H. Shadnia and Ms. Y. Saadatnejadi, for many helpful discussions pertaining to the work reported in our paper. We are also grateful to Ms. F. Balkani for her assistance in preparation of our paper.

References

Achanta, A.S., Kowalski, J.G., Rhodes, C.T., 1995. Artificial neural networks: implications for pharmaceutical sciences. Drug Dev. Ind. Pharm. 21, 119–155.

Agatonovic-Kustrin, S., Beresford, R., 2000. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. J. Pharm. Biomed. Anal. 22, 717–727.

Amighi, K., Moës, A.J., 1996. Influence of plasticizer concentration and storage conditions on the drug release rate from Eudragit® RS30D film-coated sustained-release theophylline pellets. Eur. J. Pharm. Biopharm. 42, 29–35.

Ashford, M., Fell, J.T., Attwood, D., Sharma, H., Woodhead, P., 1994. Studies on pectin formulations for colonic drug delivery. J. Control. Release 30, 225–232.

Bourquin, J., Schmidli, H., Hoogevest, P., Leuenberger, H., 1997a. Application of artificial neural networks (ANNs) in the development of solid dosage forms. Pharm. Dev. Technol. 2, 111–121.

Bourquin, J., Schmidli, H., Hoogevest, P.V., Leuenberger, H., 1997b. Basic concepts of artificial neural networks (ANN) modeling in the application to pharmaceutical development. Pharm. Dev. Tech. 2, 95–109.

Bourquin, J., Schmidli, H., Hoogevest, P., Leuenberger, H., 1998a. Advantages of artificial neural networks (ANNs) as alternative modeling technique for data sets showing non-linear relationships using data from a galenical study on a solid dosage form. Eur. J. Pharm. Sci. 7, 5–16.

Bourquin, J., Schmidli, H., Hoogevest, P., Leuenberger, H., 1998b. Pitfalls of artificial neural networks (ANNs) modeling technique for data sets containing outlier measurements using a study on mixture properties of a direct compressed dosage form. Eur. J. Pharm. Sci. 7, 17–28.

Bourquin, J., Schmidli, H., Hoogevest, P.V., Leuenberger, H., 1998c. Comparison of artificial neural networks (ANN) with classical modeling techniques using different experimental designs and data from a galenical study on a solid dosage form. Eur. J. Pharm. Sci. 6, 287–300.

Chen, Y., McCall, T.W., Baichwal, A.R., Meyer, M.C., 1999. The application of an artificial neural network and pharmacokinetic simulations in the design of controlled-release dosage forms. J. Control. Release 59, 33–41.

Cortez, P., Rocha, M., Neves, J., 2002. A Lamarckian approach for neural network training. Neural Process. Lett. 15, 105–116.

CPC-X Software, 2004. NeuralPower User Guide. Regsoft Inc, http://www.geocities.com/neuralpower.

Cybenko, G., 1989. Approximation by superposition of a sigmoidal function. Math. Control Signals Syst. 2, 303–314.

Demuth, H., Beale, M., 2002. Neural Network Toolbox for Use with Mathlab®, User's Guide, Version 4. Mathworks Inc.

Erb, R.J., 1993. Introduction to backpropagation neural network computation. Pharm. Res. 10, 165–170.

Fahlman, S.E., 1988. Faster-learning variations on back-propagation: an empirical study. In: Proceedings of the 1988 Connectionist Models Summer School.

Fang, J., Wang, S., Zhang, C., 2005. Application of genetic algorithm (GA) trained artificial neural network to identify tomatoes with physiological diseases. Nat. Sci. 3, 52–58.

Freeman, J.A., Skappura, D.M., 1991. Neural Networks Algorithms, Applications and Programming Techniques. Addison-Wesley, Houston, pp. 12–105.

Genshe, C., Xinhai, Z., 1994. Study and development of genetic algorithm. Inform. Control 23, 215–221.

Ghaffari, A., Oskoui, M., Helali, K., Bayati, K., Rafiee-Tehrani, M., 2006a. Pectin/chitosan/Eudragit® RS mixed-film coating for bimodal drug delivery from theophylline pellets: preparation and evaluation. Acta Pharm. 56, 299–310.

Ghaffari, A., Avadi, M.R., Moghimi, H.R., Oskoui, M., Rafiee-Tehrani, M., 2006. Mechanistic investigation of drug release from theophylline pellets coated by films containing pectin, chitosan and Eudragit® RS. J. Pharm. Pharmaceut. Sci., submitted for publication.

Hagan, M.T., Demuth, H.B., Beale, M.H., 1996. Neural Network Design. PWS, Boston.

Hagan, M.T., Menhaj, M.B., 1994. Training feedforward networks with the Marquardt algorithm. IEEE Trans. Neural Netw. 5, 989–993.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Michigan.

Hush, D., Horne, B.G., 1993. Progress in supervised neural networks. IEEE Signal Process. Mag. 10, 8–39.

Kesavan, J.G., Peck, G.E., 1996. Pharmaceutical granulation and tablet formulation using neural networks. Pharm. Dev. Technol. 1, 391–404.

Leardi, R., 2001. Genetic algorithms in chemometrics and chemistry: a review. J. Chemometrics 15, 559–569.

Lemmer, B., 1991. Circadian rhythms and drug delivery. J. Control. Release 16, 63–74.

Macleod, G.S., Fell, J.T., Collett, J.H., 1999. An in vitro investigation into the potential for bimodal drug release from pectin/chitosan/HPMC-coated tablets. Int. J. Pharm. 188, 11–18.

Maggi, L., Conte, U., 1997. New tablet design for the bimodal release of drugs. In: Proceedings of the 16th Pharm. Tech. Conf., vol. 2, pp. 38–45.

Murtoniemi, E., Merkku, P., Yliruusi, J., 1993. Comparison of four different neural network training algorithms in modeling the fluidized bed granulation process. Lab. Microcomput. 12, 69–76.

Plumb, A.P., Rowe, R.C., York, P., Brown, M., 2005. Optimisation of the predictive ability of artificial neural network (ANN) models: a comparison of three ANN programs and four classes of training algorithm. Eur. J. Pharm. Sci. 25, 395–405.

Plumb, A.P., Rowe, R.C., York, P., Doherty, C., 2002. The effect of experimental design on the modeling of a tablet coating formulation using artificial neural networks. Eur. J. Pharm. Sci. 16, 281–288.

Rumelhart, D., Hinton, G., Williams, R., 1986. Learning representations by backpropagation errors. Nature 323, 533–536.

Sakamoto, H., Matsumoto, K., Kuwahara, A., Hayami, Y., 2005. Acceleration and stabilization techniques for the Levenberg–Marquardt method. IEICE Trans. Fund. E88-A, 1971–1978.

Sarle, W., 2000. How to Measure the Importance of Inputs. SAS Institute, NC, ftp://ftp.sas.com/pub/neural/FAQ2.html#A_import.

Sun, Y., Peng, Y., Chen, Y., Shukla, A.J., 2003. Application of artificial neural networks in the design of controlled release drug delivery systems. Adv. Drug Del. Rev. 55, 1201–1215.

Sutton, R.S., 1986. Two problems with backpropagation and other steepest-descent learning procedures for networks. In: Proceedings of the Eighth Annual Conference of the Cognitive Science Society, Hillsdale, NJ, pp. 823–831.

Takahara, J., Takayama, K., Nagai, T., 1997. Multi-objective simultaneous optimization technique based on an artificial neural network in sustained release formulations. J. Control. Release 49, 11–20.

Takayama, K., Fujikawa, M., Nagai, T., 1999. Artificial neural networks as a novel method to optimize pharmaceutical formulations. Pharm. Res. 16, 1–6.

US Pharmacopeia 28, 2005. US Pharmacopeial Convention. US Pharmacopeia 28, Rockville, MD, pp. 1896–1899.

Weigend, A.D., Rumelhart, D.E., Huberman, B.A., 1992. Generalization by Weigend-elimination with application to forecasting. Adv. Neural Inf. Proc. Syst. 3, 875–882.

Whitley, D., Starkweather, T., Bogart, C., 1990. Genetic algorithms and neural networks: optimizing connections and connectivity. Parallel Comput. 14, 347–361.

Wu, T., Pan, W., Chen, J., Zhang, R., 2000. Formulation optimization technique based on artificial neural network in salbutamol sulfate osmotic pump tablets. Drug Dev. Ind. Pharm. 26, 211–215.

Yao, X., 1999. Evolving artificial neural networks. Proc. IEEE 87, 1423–1447.