

Review

## Artificial Neural Networks in Evaluation and Optimization of Modified Release Solid Dosage Forms

Svetlana Ibrić \*, Jelena Djuriš, Jelena Parojčić and Zorica Djurić

Department of Pharmaceutical Technology, Faculty of Pharmacy, Belgrade University, Vojvode Stepe 450, Belgrade 11221, Serbia; E-Mails: jelena.parojic@pharmacy.bg.ac.rs (J.D.); pjelena@gmail.com (J.P.); zorica@pharmacy.bg.ac.rs (Z.D.)

\* Author to whom correspondence should be addressed; E-Mail: svetlana.ibric@pharmacy.bg.ac.rs; Tel.: +381-11-395-1363; Fax: +381-11-397-2840.

Received: 23 July 2012; in revised form: 24 September 2012 / Accepted: 5 October 2012 /

Published: 18 October 2012

---

**Abstract:** Implementation of the Quality by Design (QbD) approach in pharmaceutical development has compelled researchers in the pharmaceutical industry to employ Design of Experiments (DoE) as a statistical tool, in product development. Among all DoE techniques, response surface methodology (RSM) is the one most frequently used. Progress of computer science has had an impact on pharmaceutical development as well. Simultaneous with the implementation of statistical methods, machine learning tools took an important place in drug formulation. Twenty years ago, the first papers describing application of artificial neural networks in optimization of modified release products appeared. Since then, a lot of work has been done towards implementation of new techniques, especially Artificial Neural Networks (ANN) in modeling of production, drug release and drug stability of modified release solid dosage forms. The aim of this paper is to review artificial neural networks in evaluation and optimization of modified release solid dosage forms.

**Keywords:** artificial neural networks; modified release; pharmaceutical development

---

### 1. Introduction

The recent regulations from federal agencies, to apply the Quality by Design approach in pharmaceutical development, have compelled researchers in pharmaceutical industry to employ design

of experiments (DoE) during product development. The benefits of DoE in product and process optimization are evident; it is the most convenient way to link the drug and excipient attributes and the process parameters to the Critical Quality Attributes (CQAs). Furthermore, its usage is necessary in defining and controlling Design Space [1]. Design Space is the multidimensional combination and interaction of input variables (e.g., material attributes) and process parameters that have been demonstrated to provide assurance of quality.

In the 1980s the use of DoE, especially factorial design, was generalized in the development of solid dosage forms. Statistical tools allowed the determination of critical process parameters of complex processes, as well as improvement and optimization of formulations [2].

Among diversity of DoE techniques, Response Surface Methodology (RSM) is the most frequently used optimization technique [2], which allows using polynomial equation(s) to link monitored responses to inputs:

$$y = f(x_1, x_2, \dots, x_q) + e \quad (1)$$

where  $y$  is response and  $x_1, x_2, \dots, x_q$  are inputs.

This equation presents a mathematical description of a relationship between inputs and output. The advantage of this empirical model is that one can easily, without knowing true (*i.e.*, mechanistic) link between inputs and outputs, describe their relationship. However, since the relationship between inputs and outputs is often more complicated, the observed model sometimes results in poor estimation of optimal conditions [2]. Drawbacks of this statistical approach were overcome in the 1990s, when, with the development of computer science, researchers started to apply machine learning.

Machine learning is a branch of artificial intelligence. It is a scientific discipline involved in design and development of algorithms that allow computers to evolve behaviors based on empirical data. Machines “learn” to automatically recognize complex patterns, to distinguish between exemplars based on their different patterns, and to make intelligent decisions. There are several machine learning disciplines, such as: decision tree learning, association rule learning, artificial neural networks, genetic programming, support vector machines, clustering, Bayesian networks, *etc.* [3]. Among these, artificial neural networks are the most frequently applied in pharmaceutical development [4].

In order to define and control the design space, the most effective approach is to complement DoE with some of the more advanced machine learning techniques, such as ANNs. DoE allows easy set-up of levels and boundaries of the materials attributes and processing parameters that are influencing CQAs. But, the linkage between the input(s) and the output(s) often requires non-linear modeling techniques that are advanced in comparison to DoE approaches. DoE models are polynomial, whereas machine learning offers a diverse variety of classification and modeling techniques.

## 2. Artificial Neural Networks

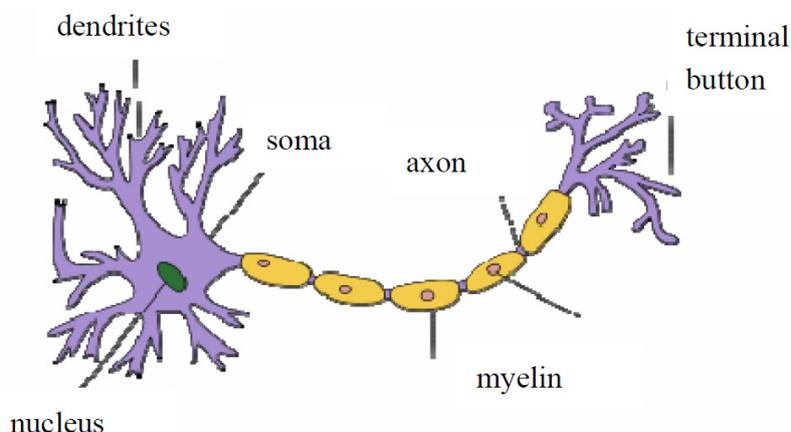
Artificial neural networks (ANNs) are computer programs, which recognize patterns in a given collection of data and produce model for that data. It emulates brain in two respects: (1) knowledge is acquired by the network through a learning process (trial and error); and (2) interneuron connection strengths (*i.e.*, synaptic weights), are used to store the knowledge [4,5].

Because neuroscience is still full of questions and because there are many levels of abstraction and many ways to take inspiration from the brain, there is no single definition of what an artificial network is. Most would agree that it involves a network of single processing elements which can exhibit complex behavior determined by the connections between the processing elements and element parameters [4,5].

Since design of artificial networks has been inspired by the structure of the human brain, it is pertinent to review/compare the function of biological neural network *versus* artificial ones.

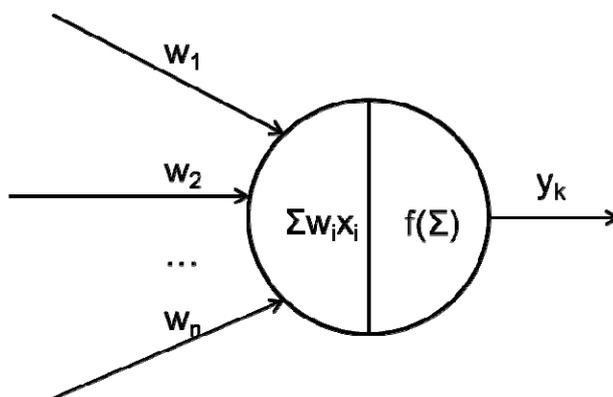
The fundamental processing unit of the human brain is the neuron (Figure 1). The neurons are linked together by dendrites, which serve to deliver messages to the neuron. Each neuron has an output channel, known as an axon by which signals can be transmitted unchanged or altered by synapses.

**Figure 1.** Schematic representation of biological neuron.



In the artificial neural network the logic processing unit is the neuron (“neurode”, “processing element” (PE) or “unit”), which takes one or more inputs and produces an output. At each neuron every input has an associate weight ( $w_1, \dots, w_n$ , Figure 2) that defines relative importance of each input connected to the neuron. Set of inputs are multiplied with sets of weights; those set of inputs are summed together, and fed into a function to produce an output ( $y_k$ , Figure 2).

**Figure 2.** Schematic representation of artificial neuron.

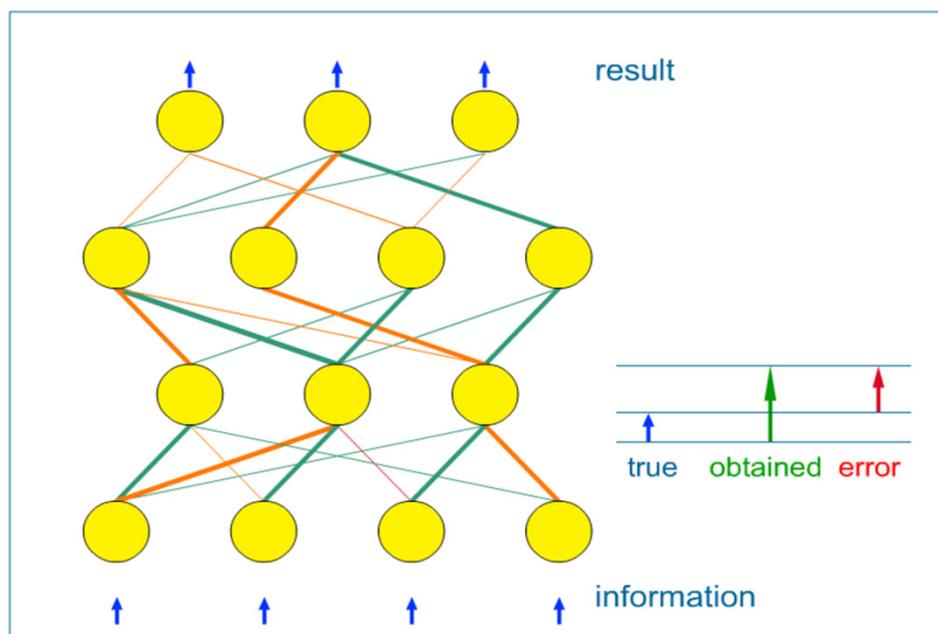


Every ANN is defined with architecture (number of layers, in each layer number of neurons, way in which neurons are connected, inter-neuron connections) and learning algorithm [4].

### 2.1. What Happens between Neurons in a Network?

In Figure 3, the simplest ANN is presented, which is a feed-forward neural network. A set of system inputs are transferred to the other neurons through synapses. The synapses store parameters called “weights” that manipulate the data in the calculations. Therefore, the set of inputs are multiplied with a set of adjustable weights. Weights are fed into the set of processing elements which then produce the set of outputs. The set of outputs are fed in another layer of weights, following another layer of processing elements, till the last processing layer, which produces output of the neural network.

**Figure 3.** Schematic representation of feed-forward neural network.



ANN is defined by the following parameters: (1) interconnection pattern between different layers of neurons; (2) the learning process for updating the weights of the interconnections; and (3) activation function that convert a neurons input to its output activation. Activation functions that are used in ANNs are sigmoidal, piecewise linear, step, Gaussian, exponential, logistic sigmoidal, *etc.* [3].

Training of the network is in fact the “learning process”. It could be divided into:

- (1) *supervised learning*: when network weights are adjusted using the known data, split into input-output pairs. Supervised learning is the most commonly used training method for ANNs and will be further explained in this paper;
- (2) *unsupervised learning*: it refers to the problem of trying to find hidden structure in unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution;
- (3) *reinforcement learning*: which differs from standard *supervised learning* in that correct input/output pairs are never presented.

Training of an ANN is actually adjustment of weight values in order to obtain the best nonlinear relationship between parameters used as inputs and outputs of the network. At the beginning of the training process, weights between the neurons have random values. During the training phase

input/output data pairs are presented to the network, and the network searches for the input-output relationships. One cycle of input-output presentation to the network is called an iteration (epoch). Process of the network training can be thought of as search for the optimal weights values that successfully convert inputs to outputs through sometimes numerous iterations. This process is often called convergence [5]. During the process of the weights adjustment (*i.e.*, network training), some of the interconnections are strengthened and some are weakened, so that a neural network will output a more correct answer [4]. Once the optimal set of weight values is found, the network training stops and squared error between the actual (training) output values and the ones predicted by the network is minimal at this point. At the beginning of the training process, data is divided in two subsets: training and test. Training data is used to search for the optimal weight values, whereas by using test data the network checks its predictive ability internally. The sum of squared errors (SSE) for the training and test subsets can be calculated using the following equation [6]:

$$SSE = \frac{1}{N} \sum_{p=1}^N \sum_{i=1}^M (t_{pi} - o_{pi})^2 \quad (2)$$

where  $o_{pi}$  is the output predicted by the network of  $i$ th output node from  $p$ th sample;  $t_{pi}$  is the training (actual) output of the  $i$ th output node from  $p$ th sample;  $N$  is the number of the training samples and  $M$  is the number of the output nodes.

In the process of the network training, the progress of change of error for both training and test data sets are evaluated simultaneously. In the case of training data set, *SSE* decreases indefinitely with increasing number of hidden nodes or training iterations (epochs) [5]. Initially, *SSE* for the training set decreases rapidly due to learning, whereas its subsequent slower decrease is attributed to memorization or overfitting (if the number of training cycles or hidden nodes is too large). In the case of test data set, *SSE* decreases initially but subsequently increases due to memorization and overfitting of the ANN model [5]. It is therefore recommended to stop the training once the test error starts to increase and to select the number of hidden nodes when the test error is minimal [5]. Steps in the supervised network training and usage are presented in Table 1.

**Table 1.** Steps in the supervised network training and usage.

<i>Training of the network</i>	
•	Data is presented to the network
•	Network computes an output
•	Network output is compared to desired output
•	Network weights are modified to reduce error
<i>Usage of the network</i>	
•	Present new, unseen data to the network
•	Network computes an output based on its training

It is important to emphasize that ANNs work as “black boxes”. It is not possible to see the mathematical relationship between inputs and the output.

## 2.2. Basic Topologies (Architectures) of Artificial Neural Networks

Generally, ANNs could be divided into static ANNs and dynamic ANNs. Differences between static and dynamic networks are in the way in which signals are transmitted through the network.

In order to obtain relevant and reliable results by using ANN models, it is recommended that the number of experimental runs is 10 times greater in comparison to the number of inputs; or, if this is not feasible, at least two to three times the number of inputs [5]. It is, therefore, practical to first conduct screening experimental design, in order to select the most significant input variables that influence output properties since it leads to reduction of the number of the ANN inputs (and number of the examples needed for the network training at the same time).

### 2.2.1. Static Networks

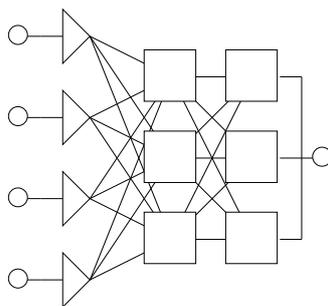
There are numerous types of static neural networks. The most frequently used static neural network in pharmaceutical development is the multilayer perceptron network.

Multilayer perceptron (MLP) is the simplest neural network architecture, that is a *feedforward artificial neural network* model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one (Figure 4). Except for the input neurons, each neuron has a nonlinear activation function. MLP utilizes a *supervised learning* technique called the *backpropagation* algorithm [7,8]. For an MLP neural network, the relationship between the inputs and outputs can be represented as follows:

$$y_i = f_o \left[ \sum_{o=1}^{N_o} b_o + \sum_{h=1}^{N_h} w_{ho} \cdot f_h \left( b_h + \sum_{i=0}^{N_i} w_{ih} x_i \right) \right] \quad (3)$$

where  $x_i$  and  $y_i$  are networks primary input and output;  $w_{ih}$  and  $w_{ho}$  ( $i = 1, 2, \dots, N_i$ ;  $o = 1, 2, \dots, N_o$ ) are the weights of the connections between the input and hidden units, and between the hidden and the output units, respectively.  $b_h$  and  $b_o$  are biases of hidden units and output units, and  $f_h(\cdot)$  and  $f_o(\cdot)$  are hidden and output functions respectively [9]. Biases are neurons that always have constant values and serve to allow certain shifting in the predicted outputs (*i.e.*, they can be thought of as function intercepts).

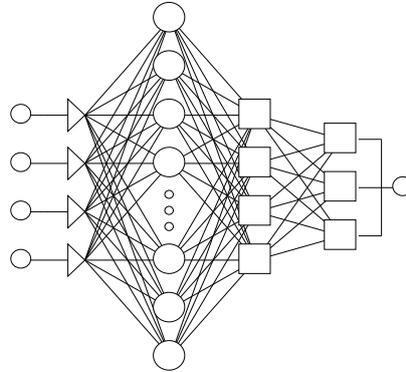
**Figure 4.** Example of Multilayer perceptron (MLP) network architecture with four layers (first layer with four neurons, second and third layers with three neurons and fourth layer with one neuron).



Generalized regression neural network (GRNN) was first introduced by Specht [10]. It models the function directly from the data. It was recognized that GRNNs have been derived from statistical

method of function approximation [11]. GRNNs always have exactly four layers: input layer, a layer of so-called radial centers, a layer of regression units and output layer (Figure 5).

**Figure 5.** Example of generalized regression neural network (GRNN) architecture.



Radial layer is used to perform clustering on the known, training data. Clustering algorithms usually used for training of this layer are sub-sampling, K-means or Kohonen [12]. The number of clusters, *i.e.*, radial units (neurons) can be optimized and usually it corresponds to the number of the training samples. Regression layer has one unit (neuron) more than the output layer and their activation function is linear. One extra unit (in comparison to the output layer) is used for calculation of the probability density whereas remaining units are used for calculation of outputs. In the output layer a specialized function is applied where the calculated outputs of the regression layer are divided by the probability density [12].

Fundamental equation of GRNN can be presented as:

$$\hat{y} = \frac{\sum_{i=1}^n y_i \exp(-D(x, x_i))}{\sum_{i=1}^n \exp(-D(x, x_i))} \quad (4)$$

where the distance  $D$  between the training sample and the point of prediction, is used as a measure of how well the each training sample can represents the position of prediction, using the standard deviation or the smoothness parameter,  $\sigma$  [10].

$$D(x, x_i) = \sum_{j=1}^p \left( \frac{x_j - x_{j,i}}{\sigma_j} \right)^2 \quad (5)$$

The advantage of the GRNN is that its training is much faster compared to MLPs.

### 2.2.2. Dynamic Neural Networks

Dynamic (or recurrent) networks have the ability to store data and elaborate it in time. That is the advantage of dynamic neural networks (compared to the static networks). Every input is analyzed as a function of the previous input. The network remembers past inputs; therefore, the current output is integration of past inputs and current response of the system [13].

Connections between the neurons can be set up to have a memory which is very important for dynamic networks. The order of the memory says how many time steps the signal will be delayed. Treatment of dynamic data requires this kind of temporal dependencies of signal channeling. Network topology (architecture), together with the control system for time delay of the signal, forms a complete system. Correction of weights in the dynamic neural network is somewhat more complicated, in comparison to static neural networks. It is possible to use the technique called backpropagation through time where the backpropagated signal is buffered and reversed which enables getting forward and backpropagated signals synchronized in time [13].

Gamma memory is a specific short memory recurrent structure which preserves temporal information about the system. Distinctive feature of gamma memory is the number of taps-number of signal delays. For a given number of taps memory remembers previous system states and integrates them with current ones. From the point of view of signal transmittance Gamma memory can be seen as recursive low-pass filter (each output gives a more filtered version of the original signal) which acts as an infinite response filter. It is ideal for adaptive systems since its interpolation weight  $\mu$  can be adapted using the usual algorithms. An important class of memory filters is based on the (non-dispersive) delay operator. Gamma memory is actually a combination of Tapped-Delay-Line (TDL) and a simple feedback neuron. So, the signals  $g_k(t)$  at the taps  $k$  in time  $t$  of the Gamma memory are convulsions of the impulse response  $g_k$  tap  $k$  [13].

A  $K$ -th order tapped delay line, the memory structure for Time Delay Neural Network (TDNN) [14], can be regarded as a single-input- $K$ -output filter with impulse responses  $g_k(t) = d(t - k)$ .

Depending on the way that signals are sent back in the neural network, partial or full recurrence exists. Partial recurrence occurs, for example, when recurrent connections are sent from the hidden layer back to itself. In the fully recurrent networks the final output (of the network) is sent back into the network. If connections between the neurons are set up to have a memory then the order of the memory says how many time steps the signal will be delayed (default value is one time step, *i.e.*,  $-1$ ). Over time, the network stores long term memory structure in its feedback (recurrent) and regular connections, whose weights are adjusted during the training [15].

Elman neural network (ENN) is considered as a special kind of feed-forward networks that has additional memory neurons and local feedback [16]; therefore it is a simple dynamic neural network. Recurrent links are used to provide network with a dynamic memory when hidden unit patterns are fed back to themselves [17]. Activation functions of neurons in the hidden layer are fed back to copied layer at every time step to provide an additional input in conjunction with other input features. This recurrence gives the network dynamic properties, which make it possible for the network to have internal memory [18].

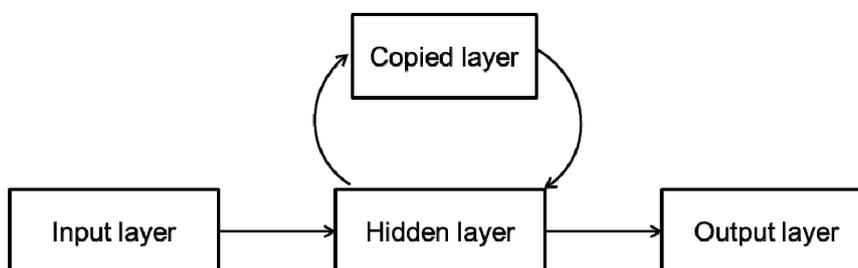
Essential characteristic of Elman neural network (Figure 6) is the fact that a copied (context) layer exists that serves as the temporary memory of the hidden layer outputs. Once the signal is transferred from the input layer to the hidden layer, it is elaborated and sent both to output and copied layer. During the first input-output cycle, copied layer is empty, *i.e.*, its outputs are equal to 0. Then, when the activation function in the hidden layer is applied for the first time its outcomes are sent to output and copied layer. During the second training cycle, the hidden layer receives inputs from both input and copied layer which are then combined and analyzed simultaneously. The process is repeated for the next training cycles so that each time the copied layer contains values of the outputs of hidden

layer from the previous training cycle. This type of signal recurrence is recognized as a one-step time delay [9]. The algorithm of ENN can be represented as follows:

$$y_k = f_o \left[ \sum_{o=1}^{N_o} b_o + \sum_{h=1}^{N_h} w_{ho} \cdot f_h \left( b_h + \sum_{i=0}^{N_i} w_{ih} x_i + \sum_{j=0}^{N_h} w_{jh} a_j(k-1) \right) \right] \quad (6)$$

where  $x_k$  and  $y_k$  are networks primary input and output;  $w_{ih}$ ,  $w_{jh}$  and  $w_{ho}$  ( $i = 1, 2, \dots, N_i$ ;  $j, h = 1, 2, \dots, N_h$ ;  $o = 1, 2, \dots, N_o$ ) are the weights of the connections between the input and hidden units, between the copied and the hidden units and between the hidden and the output units, respectively.  $b_h$  and  $b_o$  are biases of hidden units and output units, and  $f_h(\cdot)$  and  $f_o(\cdot)$  are hidden and output functions respectively. The weights of the copied layer determine the impact of signal recurrence on the networks output.

**Figure 6.** Topology of Elman neural network.



### 2.3. Application of ANNs in Formulation and Evaluation of Modified Release Dosage Forms

A pioneer work in the field was a paper published by Hussein *et al.* in 1991. They modeled the *in vitro* release characteristics of drugs in matrices prepared from various hydrophilic polymers [19,20]. Neural networks with a single hidden layer were applied to predict drug release from matrix tablets. Predicted drug release profiles were comparable with those generated through the use of statistical analysis [19].

Chen and coworkers [21] studied the influence of seven formulation variables and three tablet variables (moisture, particle size and hardness) on dissolution-time profiles at ten sampling times. Twenty two formulations were used for the network training. The optimized ANN model was constructed by selecting the optimal number of hidden layers and neuron number in each layer. Optimized ANN model was used for prediction of formulations based on two desired target dissolution profiles, as well as two desired bioavailability profiles.

In modeling of a roller-compaction process, Turkoglu *et al.* [22] varied four formulation and process variables through 42 experiments. Results were analyzed using ANN with genetic algorithms. When a set of validation experiments was analyzed, genetic algorithm predictions of tablet characteristics were much better than the ANN.

Optimization of theophylline release from controlled release tablets was performed according to desirable drug release profiles, based on the plasma concentration profiles of theophylline predicted by the pharmacokinetic analysis in humans [23]. Experimentally observed drug release profiles from 16 formulations were used for ANN training. Simultaneous optimization was performed by minimizing the generalized distance between the predicted values of each response and the desirable one that was optimized individually. Similar study was performed by Peh *et al.* [24] with optimization

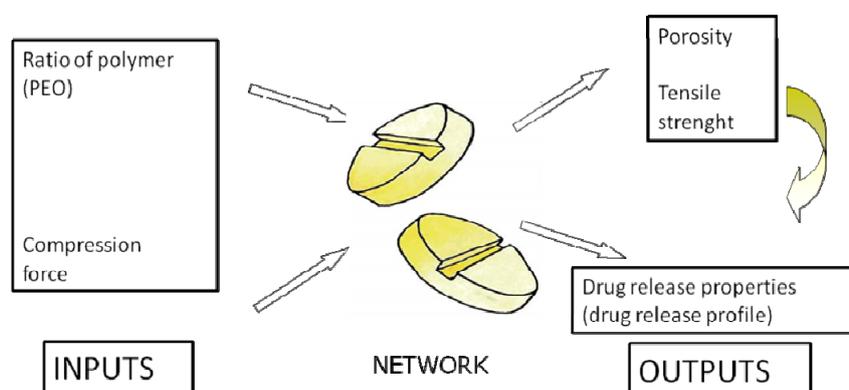
of glyceryl monostearate matrix tablets. MLP neural network was trained and the closeness between the predicted and the reference dissolution profiles was investigated using similarity factor ( $f_2$ ).

In a number of papers, ability of ANNs to predict controlled release profiles of a drug was compared with the response surface methodology [25,26]. In general, formulations were optimized using both RSM and ANNs. Afterwards, predicted profiles were compared with experimentally observed. Differences between predicted and experimentally observed results were discussed.

Generalized regression neural network was applied in optimization of formulation of aspirin matrix tablets with Eudragit RS and Eudragit L, as matrix forming materials [12,27]. The percent of Eudragit and tablet hardness were inputs of the network, varied according to the central composite design, while percent of aspirin released at 4 sampling time points, as well as release exponent of Korsmayer–Peppas equation were outputs of the network. In another study, the same authors applied GRNN network in predicting drug stability in matrix tablets [28]. Optimization of formulation with respect to drug release profile, as well prediction of drug stability showed abilities of GRNN network in formulation of controlled release preparations. It seems that this type of network has ability to optimize controlled release formulations better than MLP network. Also, training with GRNN network is much faster than with other network types [12].

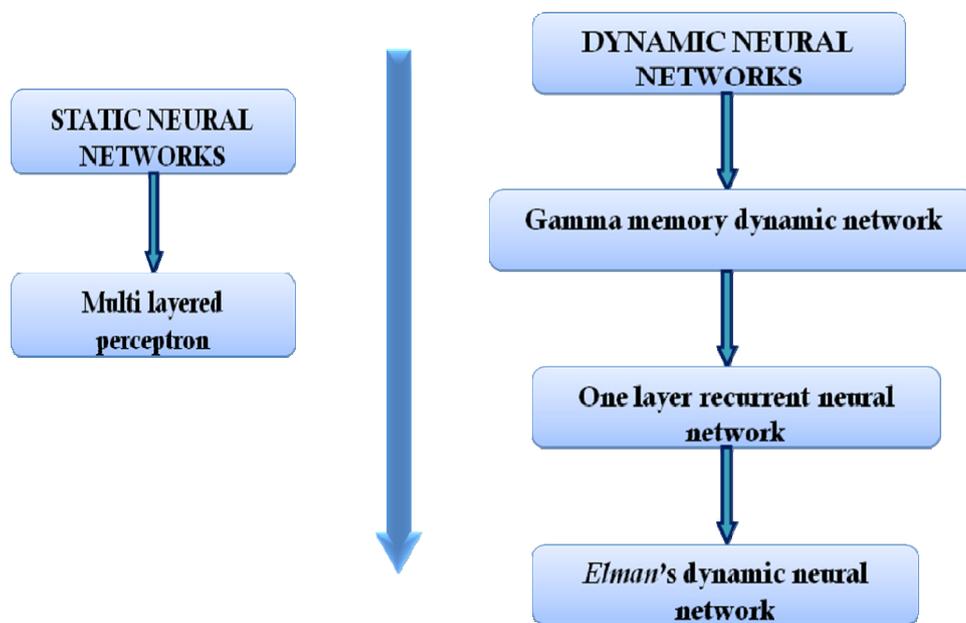
Dynamic neural networks were for the first time applied in the prediction of drug release profiles from controlled release formulation in 2002 [18]. Drug release profiles were treated as time series curves in which information contained in one time point affects further predictions. Elman recurrent network was applied. Since in dynamic neural networks data is stored and elaborated in time, it is expected that it could be useful in time dependent processes, like drug release prediction or drug stability issues. Therefore, our research group have successfully applied dynamic neural networks in establishing design space for formulation of matrix tablets and compared those results with static networks [13,29,30]. Following case study presents our unpublished data and could be useful for the readers as a guide in selection and development of appropriate network for the problem studied. The aim of the study was to develop an ANN model for optimization of matrix tablets mechanical properties and drug release profile as well. Inputs were ratio of matrix forming material (Polyox WSR 1105 and Polyox WSR Coagulant) and compression force, while mechanical properties (porosity and the tensile strength) of tablets, as well as drug release profiles were outputs of the network (Figure 7).

**Figure 7.** Schematic representation of inputs and outputs for the optimization of matrix tablets [30].



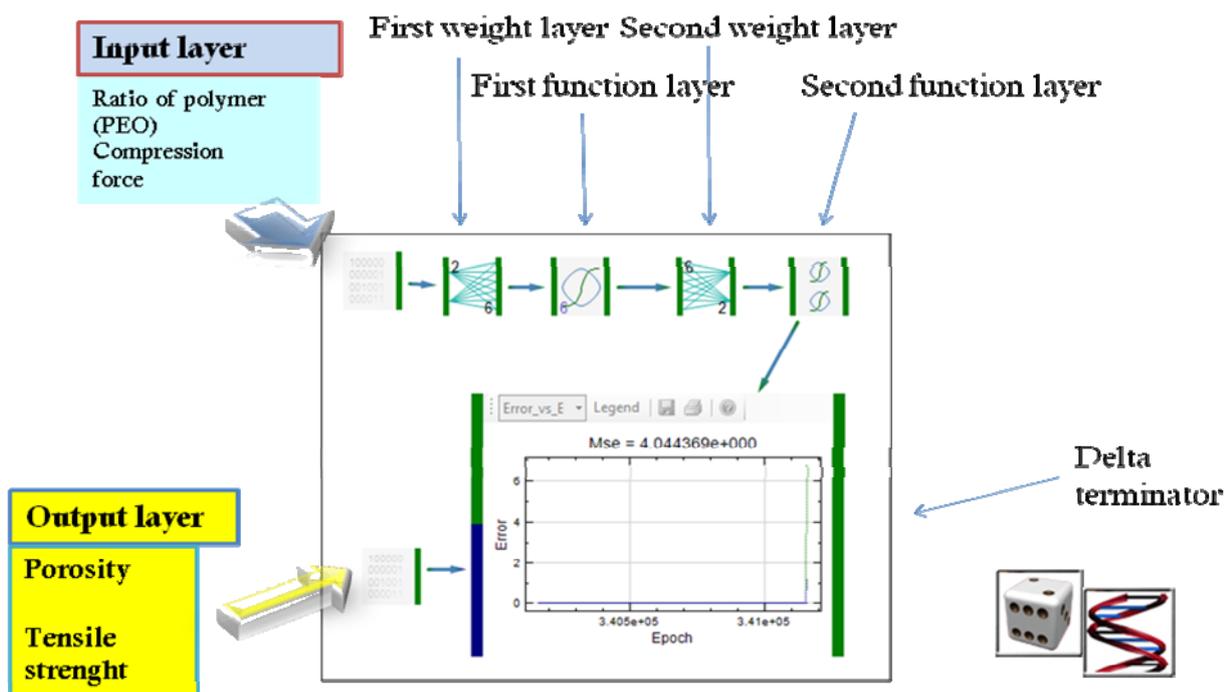
Eleven formulations were prepared (nine, according to full  $3^2$  factorial design for network training and two for network testing). Network topology was chosen according to the flow diagram presented in Figure 8.

**Figure 8.** Flow diagram for the network architecture selection [30].



Firstly, MLP network was chosen, as the most frequently used static neural network. The structure of MLP network used in the study is presented in Figure 9.

**Figure 9.** Schematic representation of MLP network.



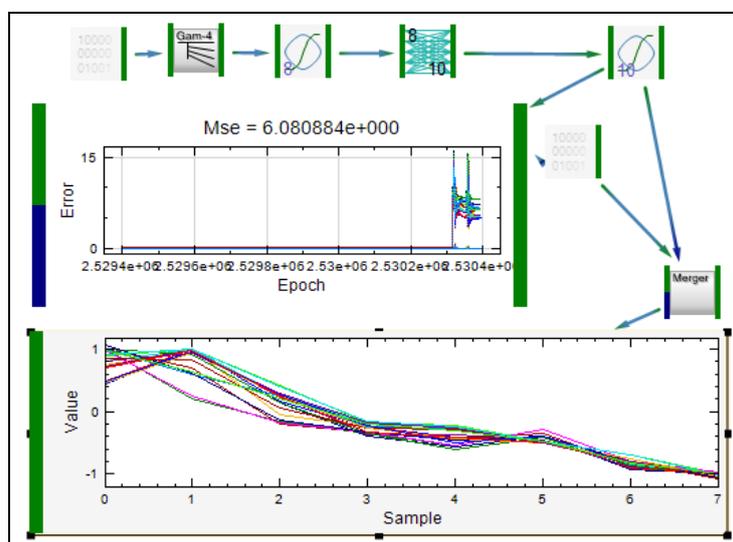
When the trained MLP network was tested using the test formulations with respect to mechanical properties, the model developed by MLP network enables prediction of porosity and tensile strength for test formulation, on the basis of known compression force and weight ratio of PEO polymer used (Table 2).

**Table 2.** Observed *versus* MLP predicted values for test formulations.

Formulation	Observed values Porosity (%) and tensile strength (MPa)	Predicted values Porosity (%) and tensile strength (MPa) ( $r^2 = 0.9982$ )
Test 1	19.55 ± 0.49	20.34 ± 0.78
	1.304 ± 0.042	1.313 ± 0.155
Test 2	17.55 ± 0.55	17.33 ± 0.78
	1.661 ± 0.035	1.539 ± 0.155

Afterwards, the same MLP network was used for training, when the outputs were drug release profiles. Percents of drug released in predetermined time intervals were outputs of the network. Inputs of the network were the same as in previous case. After the training session, the established network was not able to predict drug release profiles for test formulations. The calculated values for similarity factor indicated that profiles are not similar ( $f_2$  55.37 and 48.22 for test 1 and test 2 formulation, respectively). The next step was to change the network type and to evaluate the drug release prediction using dynamic neural networks. Gamma memory one layer dynamic neural network was applied (GMDNN) (Figure 10). Gamma memory (GM) used had 4 taps.

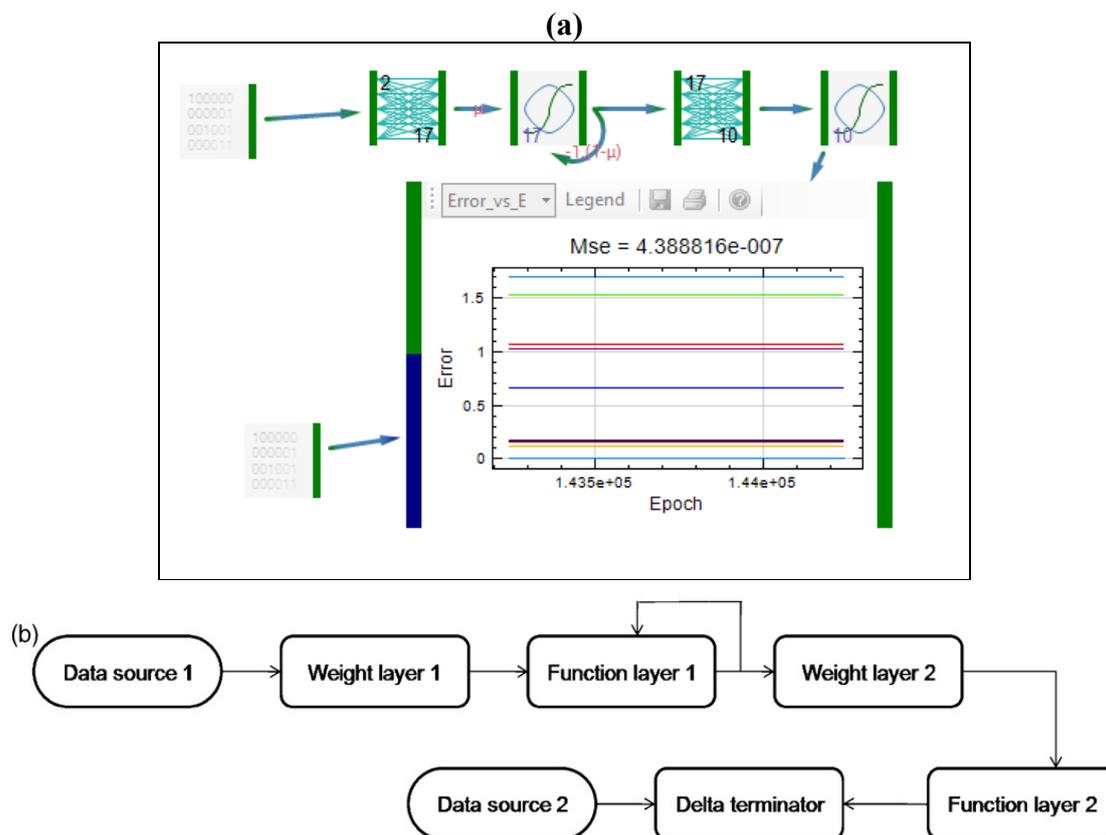
**Figure 10.** Schematic representation of gamma memory dynamic neural network (GMDNN).



From GM signal is transferred to the first function layer, then to the weight layer and subsequently to the second function layer. Delta terminator compares two signals: one comes from the data source and represents real, observed outputs; whereas the second signal comes from the second function layer and represents outputs predicted by the dynamic system. The number of inputs and outputs in weight and function layers, as well as tapped delay line, were optimized using Monte Carlo simulations. As in the case of model developed by MLP, the obtained model does not predict drug release successfully.

GM predicted shifting of profiles towards higher percentage of drug released after second hour of study, which was not confirmed experimentally. Topology of the recurrent one layer dynamic neural network (OLDNN), which was next used in the training, is presented in Figure 11.

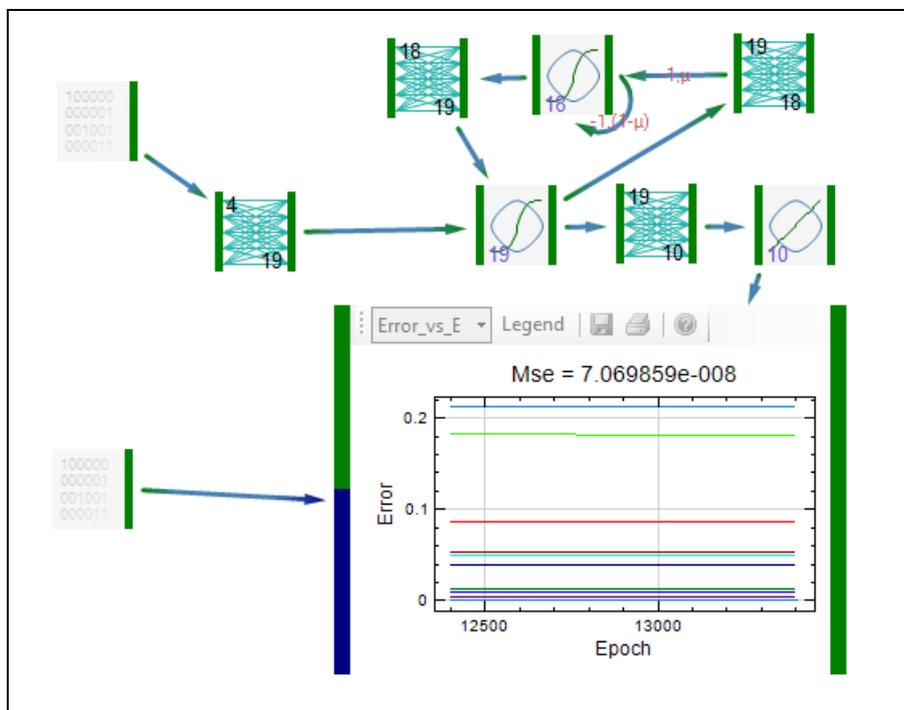
**Figure 11.** Representation of recurrent one layer dynamic neural network (OLDNN): (a) Peltarion<sup>®</sup> software layout and (b) network's schematic representation.



Most of the networks elements were the same as in previously described GMDNN. From the first data source inputs are transferred to the first weight layer. The first weight layer has 2 inputs and 17 outputs. The number of its outputs was optimized using Monte Carlo simulations. OLDNN is a partially recurrent network since the recurrent connections are sent from the hidden layer back to itself (first function layer). In the fully recurrent networks the output is sent back into the network. From the first, recurrent function layer, signal goes to the second weight layer that has 17 inputs and 10 outputs. Then the signal goes to the second functional layer, which is an ordinary function layer. Once again, the network was trained and afterwards tested. The model obtained using OLDNN enabled adequate prediction of drug release profiles, with similarity factors for test formulations 64.01 and 77.54, respectively. Even though results obtained using this network were better, there were still shifting values of predicted drug release profiles compared to experimentally observed values at later time points. Therefore, in the next study [29], the starting point of experiments was changed: since it was demonstrated that mechanical properties of the tablets have impact on drug release, porosity and the tensile strengths were selected to be inputs, together with ratio of the polymer and compression force. Elman dynamic neural network was applied, with topology shown in Figure 12. This type of network

was also applied in modeling of drug release from lipid matrix tablets [29]. After the training session, prediction of drug release from test formulations was better in comparison to OLDNN application.

**Figure 12.** Representation of Elman dynamic neural network.



In Table 3, studies on formulation, characterization and optimization of modified release formulations are listed, with respect to inputs/outputs selected, type of the network used, and the authors and year of performed studies.

**Table 3.** Formulation, characterization and optimization of modified release solid dosage forms are listed, with respect to inputs/outputs selected, type of the network used, and the authors(year) of performed studies.

Formulation, characterization and optimization of modified release formulation		
Inputs/outputs/aim	Network type	Authors, year
Design of controlled release formulations. Varying formulation variables were used as inputs and <i>in vitro</i> cumulative percentages of drug released were used as ANN outputs.	MLP	Chen, 1999 [21]
Optimization of diclofenac sodium sustained release matrix tablets. Trained model was used to predict release profiles and to optimize the formulation composition.	MLP	Zupancic Bozic, 1997 [31]
Design of extended release aspirin tablets. The amount of Eudragit® RS PO/Eudragit® L-polymer and compression pressure were selected as inputs, whereas <i>in vitro</i> dissolution profiles, release order and release constant from the Korsmayer Peppas equation were selected as output parameters.	GRNN	Ibrić, 2002, 2003, 2007 [12,27,28]

Table 3. Cont.

Formulation, characterization and optimization of modified release formulation		
Inputs/outputs/aim	Network type	Authors, year
Prediction of drug dissolution profiles. Inputs for the network training were the matrix forming agents' ratio, the time point of the measurement of percent dissolved, and the difference between the release rate of the preceding two time points of the predicted profile. <i>In vitro</i> dissolution profiles were used as network outputs.	MLP	Peh, 2000 [24]
Investigation of controlled drug release. Drug fraction and time were used as network inputs and <i>in vitro</i> dissolution profiles as outputs.	MLP	Reis, 2004 [32]
Prediction of dissolution profiles for matrix controlled release theophylline pellet preparation. Inputs for the network training were the matrix forming agents' ratio, and the time point of the measurement of percent dissolved; <i>in vitro</i> dissolution profiles were used as outputs.	EDNN	Goh, 2002 [18]
Modeling of diclofenac sodium release from Carbopol 71G matrix tablets. Polymer and binder content were inputs, while <i>in vitro</i> dissolution profiles were outputs	MLP	Ivic, 2010a [26]
Modeling of diclofenac sodium release from polyethylene oxide matrix tablets. Polymer weight ratio and compression force were used as inputs, whereas <i>in vitro</i> dissolution profiles were used as networks outputs. Dissolution profiles were treated as time series using dynamic neural networks.	MLP, GMDNN, OLDNN	Petrović, 2009 [13]
Drug release control and system understanding of sucrose esters matrix tablets. Networks inputs were HLB values of sucrose esters (SEs), SEs concentration, tablet volume, tablet porosity and tablet tensile strength. <i>In vitro</i> dissolution profiles and parameters indicative of burst release mean dissolution time and release exponent were used as outputs.	MLP	Chansanroj, 2011 [33]
A number of unique ANN configurations are presented, that have been evaluated for their ability to determine an IVIVC from different formulations of the same product. <i>In vitro</i> dissolution data were used as inputs and associated outputs were pharmacokinetic time points from nine patients enrolled in a crossover study.	MLP, GRNN, RNN	Dowell, 1999 [34]
Development of level A <i>in vitro</i> – <i>in vivo</i> correlation. Inputs for the network training were <i>in vitro</i> dissolution samples whereas <i>in vivo</i> dissolution profiles calculated by numerical deconvolution for each volunteer individually were used as outputs.	GRNN	Parojčić, 2007 [35]

Table 3. Cont.

Formulation, characterization and optimization of modified release formulation		
Inputs/outputs/aim	Network type	Authors, year
Prediction of relative lung bioavailability and clinical effect of salbutamol when delivered to healthy volunteers and asthmatic patients from dry powder inhalers (DPIs). Training of the ANN network was performed using <i>in vitro</i> aerodynamic characteristics of the formulation and demographic data of volunteers/patients as input parameters, whilst <i>in vivo</i> data (urinary excretion of the drug and its metabolite) were networks outputs.	MLP	De Matas, 2008 [36]
Prediction of kinetics of doxorubicin release from sulfopropyl dextran ion-exchange microspheres. Three independent variables, drug loading level, concentration of NaCl and CaCl <sub>2</sub> in the release medium were used as the ANN inputs and the fractional release of doxorubicin at four different time points as the outputs.	MLP, HNN	Li, 2005 [37]
Prediction of drug release profiles in transdermal iontophoresis. Neural networks inputs were the process conditions of pH, ionic strength and current, as well as the time point. The output was the predicted permeation rate of the drug (diclofenac sodium).	RBFNN	Lim, 2003 [38]
Optimization of drug release from compressed multi unit particle system (MUPS) using generalized regression neural network (GRNN)	GRNN	Ivic, 2010b [39]

Abbreviations: MLP, Multilayered Perceptron; GRNN, Generalized Regression Neural Network; RBFNN, Radial Basis Function Neural Network; EDNN, Elman Dynamic Neural Network; GMDNN, Gamma memory Dynamic Neural Network; OLDNN, One Layer Dynamic Neural Network; HNN, Hierarchical Neural Network.

The mutual idea of presented studies could be described as follows: they attempted to create a relationship between formulation and/or process variables (**INPUTS**) in production of modified release preparation (regardless of whether they were matrix tablets or multiple unit systems) and drug release profiles or dissolution rate constants (**OUTPUTS**). In the training session, prediction capability of the trained networks was tested comparing predicted and experimentally observed drug release profiles, using the similarity factor as parameter and/or correlation coefficient.

To date, there are many commercial, as well as freely available computer programs that can be used for ANN modeling. It should be pointed out that modeling of the same data through different software will give a little bit different results. An interesting study has been published by Chen and coworkers [40]. They compared four commercially ANN software programs to predict the *in vitro* dissolution profiles of controlled release tablets. The ANNs developed from the four software programs were validated by predicting the *in vitro* dissolution time-profiles of each of the 19 formulations, which were excluded from the training process. Although the same data set was used, the optimum ANN architectures generated from the four software programs was different.

Nevertheless, it was concluded that the four programs provided reasonable predictions of *in vitro* dissolution profiles for the data set employed in this study.

### 3. Future Prospects

So far, a lot of work has been done concerning application of ANNs in optimizing drug release profiles from controlled release formulation. As neural computing is progressing, current trends and recently published articles [29,33,41–43] indicate that the future state in this area will include integration of ANNs with other machine learning techniques, such as fuzzy logic, genetic programming, decision trees, self-organizing maps, *etc.* Decision tree methodology, a nonparametric statistical technique, used for classification problems, could be successfully used in characterization of controlled release problems, as was reported. The main constraint of this methodology is that it requires a large data set, because inductive generalization given in the form of a decision tree is dependent on a sufficient amount of data [29].

There have been very few attempts to use genetic programming in the optimization of pharmaceutical formulations [42–44]. Published results indicate that symbolic regression via genetic programming is a promising technique; in comparison to ANN models, it provides parametric equations, which can be more easily interpreted, thus having a greater potential for understanding of the mechanisms of the modeled process [43].

However, ANN models are well established and could be used in implementation of Quality by Design concept, *i.e.*, understanding of Design Space and Quality Risk Management for modified release formulations.

### Acknowledgments

This work was supported by the project TR34007, funded by Ministry of Education and Science, Republic of Serbia.

### Conflict of Interest

The authors declare no conflict of interest.

### References

1. Snee, R.D. Quality by design-four years and three myths later. *Pharm. Proc.* **2009**, *2*, 14–16.
2. Lewis, G.A.; Mathieu, D.; Luu, P.T. *Pharmaceutical Experimental Design*; Marcel Dekker: New York, NY, USA, 1999.
3. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin, Germany, 2006.
4. Agatonovic-Kustrin, S.; Beresford, R. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *J. Pharm. Biomed. Anal.* **2000**, *22*, 717–727.
5. Sun, Y.; Peng, Y.; Chen, Y.; Shukla, A. Application of artificial neural networks in the design of controlled release drug delivery systems. *Adv. Drug Deliv. Rev.* **2003**, *55*, 1201–1215.

6. Basheer, I.A.; Hajmeer, M. Artificial neural networks: Fundamentals, computing, design and application. *J. Microbiol. Methods* **2000**, *43*, 3–31.
7. Rosenblatt, F. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*; Spartan Books: Washington DC, USA, 1961.
8. Rumelhart, D.E.; Geoffrey, E.H.; Williams, R.J. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*; MIT Press: Massachusetts, UK, 1986.
9. Zhang, J.; Man, K.F. Time series prediction using recurrent neural network in multi-dimension embedding phase space. *IEEE Syst. Man Cybern. Soc.* **1998**, *2*, 11–14.
10. Specht, D.F. A general regression neural network. *IEEE Trans. Neural Netw.* **1991**, *2*, 568–576.
11. Patterson, D. *Artificial Neural Networks*; Prentice Hall: Singapore, 1996.
12. Ibrić, S.; Jovanović, M.; Djurić, Z.; Parojčić, J.; Solomun, L. The application of generalized regression neural network in the modeling and optimization of aspirin extended release tablets with Eudragit RS PO as matrix substance. *J. Control. Release* **2002**, *82*, 213–222.
13. Petrović, J.; Ibrić, S.; Betz, G.; Parojčić, J.; Đurić, Z. Application of dynamic neural networks in the modeling of drug release from polyethylene oxide matrix tablets. *Eur. J. Pharm. Sci.* **2009**, *38*, 172–180.
14. Waibel, A.; Hanazawa, T.; Hinton, G.; Shikano, K.; Land, K. Phoneme recognition using time-delay neural networks. *IEEE Trans. ASSP* **1989**, *37*, 328–339.
15. Samarasinghe, S. *Neural Networks for Applied Sciences and Engineering*; Auerbach Publications: New York, NY, USA, 2006.
16. Köker, R. Design and performance of an intelligent predictive controller for a six-degree-of-freedom robot using the Elman network. *Inf. Sci.* **2006**, *12*, 1781–1799.
17. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *2*, 179–211.
18. Goh, W.Y.; Lim, C.P.; Peh, K.K.; Subari, K. Application of a recurrent neural network to prediction of drug dissolution profiles. *Neural Comput. Appl.* **2002**, *10*, 311–317.
19. Hussain, A.S.; Yu, X.; Johnson, R.D. Application of neural computing in pharmaceutical product development. *Pharm. Res.* **1991**, *8*, 1248–1252.
20. Hussain, A.S.; Shivanand, P.; Johnson, R.D. Application of neural computing in pharmaceutical product development: Computer aided formulation design. *Drug Dev. Ind. Pharm.* **1994**, *20*, 1739–1752.
21. Chen, Y.; McCall, T.W.; Baichwal, A.R.; Meyer, M.C. The application of an artificial neural network and pharmacokinetic simulations in the design of controlled-release dosage forms. *J. Control. Release* **1999**, *59*, 33–41.
22. Turkoglu, M.; Aydin, I.; Murray, M.; Sakr, A. Modeling of a roller-compaction process using neural networks and genetic algorithms. *Eur. J. Pharm. Biopharm.* **1999**, *48*, 239–245.
23. Takayama, K.; Morva, A.; Fujikawa, M.; Hattori, Y.; Obata, Y.; Nagai, T. Formula optimization of theophylline controlled-release tablet based on artificial neural networks. *J. Control. Release* **2000**, *68*, 175–186.
24. Peh, K.K.; Lim, C.P.; Quek, S.S.; Khoh, K.H. Use of artificial neural networks to predict drug dissolution profiles and evaluation of network performance using similarity factor. *Pharm. Res.* **2000**, *17*, 1384–1389.

25. Vaithiyalingam, S.; Khan, M.A. Optimization and characterization of controlled release multi-particulate beads formulated with a customized cellulose acetate butyrate dispersion. *Int. J. Pharm.* **2002**, *234*, 179–193.
26. Ivić, B.; Ibrić, S.; Cvetković, N.; Petrović, A.; Trajković, S.; Djurić, Z. Application of design of experiments and multilayer perceptrons neural network in the optimization of diclofenac sodium extended release tablets with Carbopol® 71G. *Chem. Pharm. Bull.* **2010**, *58*, 947–949.
27. Ibrić, S.; Jovanović, M.; Djurić, Z.; Parojčić, J.; Petrović, S.D.; Solomun, L.J.; Stupar, B. Artificial neural networks in the modeling and optimization of aspirin extended release tablets with Eudragit L 100 as matrix substance. *AAPS Pharm. Sci. Tech.* **2003**, *4*, 62–70.
28. Ibrić, S.; Jovanović, M.; Djurić, Z.; Parojčić, J.; Solomun, L.; Lučić, B. Generalized regression neural networks in prediction of drug stability. *J. Pharm. Pharmacol.* **2007**, *59*, 745–750.
29. Petrović, J.; Ibrić, S.; Betz, G.; Djurić, Z. Optimization of matrix tablets controlled drug release using Elman dynamic neural networks and decision trees. *Int. J. Pharm.* **2012**, *428*, 57–67.
30. Djuriš, J. Characterization of controlled release hydrophilic and lipid matrix tablets using in silico methods, Ph.D. dissertation, Belgrade University, Belgrade, Serbia, June 2010.
31. Zupancic Bozic, D.; Vrečer, F.; Kozjek, F. Optimization of diclofenac sodium dissolution from sustained release formulations using an artificial neural network. *Eur. J. Pharm. Sci.* **1997**, *5*, 163–169.
32. Reis, M.A.A.; Sinisterra, R.D.; Belchior, J.C. An alternative approach based on artificial neural networks to study controlled drug release. *J. Pharm. Sci.* **2004**, *93*, 418–430.
33. Chansanroj, K.; Petrović, J.; Ibrić, S.; Betz, G. Drug release control and system understanding of sucrose esters matrix tablets by artificial neural networks. *Eur. J. Pharm. Sci.* **2011**, *44*, 321–331.
34. Dowell, J.A.; Hussain, A.; Devane, J.; Young, D. Artificial neural networks applied to the *in vitro*–*in vivo* correlation of an extended-release formulations: Initial trials and experience. *J. Pharm. Sci.* **1999**, *88*, 154–160.
35. Parojčić, J.; Ibrić, S.; Đurić, Z.; Jovanović, M.; Corrigan, O.I. An investigation into the usefulness of generalized regression neural network analysis in the development of level A *in vitro*–*in vivo* correlation. *Eur. J. Pharm. Sci.* **2007**, *30*, 264–272.
36. De Matas, M.; Shao, Q.; Richardson, C.H.; Chrystyn, H. Evaluation of *in vitro* *in vivo* correlations for dry powder inhaler delivery using artificial neural networks. *Eur. J. Pharm. Sci.* **2008**, *33*, 80–90.
37. Li, Y.; Rauth, A.M.; Wu, X.Y. Prediction of kinetics of doxorubicin release from sulfopropyl dextran ion-exchange microspheres using artificial neural networks. *Eur. J. Pharm. Sci.* **2005**, *24*, 401–410.
38. Lim, C.P.; Quek, S.S.; Peh, K.K. Prediction of drug release profiles using an intelligent learning system: An experimental study in transdermal iontophoresis. *J. Pharm. Biomed. Anal.* **2003**, *31*, 159–168.
39. Ivić, B.; Ibrić, S.; Betz, G.; Djurić, Z. Optimization of drug release from compressed multi unit particle system (MUPS) using generalized regression neural network (GRNN). *Arch. Pharm. Res.* **2010**, *33*, 103–113.

40. Chen, Y.; Jiao, T.; McCall, T.W.; Baichwal, A.R.; Meyer, M.C. Comparison of four artificial neural network software programs used to predict the *in vitro* dissolution of controlled-release tablets. *Pharm. Dev. Tech.* **2002**, *7*, 373–379.
41. Tan, C.; Degim, I.T. Development of sustained release formulation of an antithrombotic drug and application of fuzzy logic. *Pharm. Dev. Tech.* **2012**, *17*, 242–250.
42. Barmpalexis, P.; Kachrimanis, K.; Tsakonas, A.; Georgarakis, E. Symbolic regression via genetic programming in the optimization of a controlled release pharmaceutical formulation. *Chemom. Intell. Lab. Syst.* **2011**, *107*, 75–82.
43. Barmpalexis, P.; Kachrimanis, K.; Georgarakis, E. Solid dispersions in the development of a nimodipine floating tablet formulation and optimization by artificial neural networks and genetic programming. *Eur. J. Pharm. Biopharm.* **2011**, *77*, 122–131.
44. Colbourn, E.A.; Roskilly, S.J.; Rowe, R.C.; York, P. Modelling formulations using gene expression programming-A comparative analysis with artificial neural networks. *Eur. J. Pharm. Sci.* **2011**, *44*, 366–374.

© 2012 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).