Artificial Neural Networks Applied to the In Vitro–In Vivo Correlation of an Extended-Release Formulation: Initial Trials and Experience

JAMES A. DOWELL,[†,‡] AJAZ HUSSAIN,[§] JOHN DEVANE,[†,||] AND DAVID YOUNG*[,†,‡]

Contribution from *In Vitro–In Vivo Relationship Cooperative Working Group, Pharmacokinetics-Biopharmaceutics Laboratory, Department of Pharmaceutical Sciences, University of Maryland at Baltimore, Baltimore, Maryland 21201, Food and Drug Administration, Rockville, Maryland 20855, and Elan Corporation plc, Athlone, Ireland.*

Received April 15, 1997. Final revised manuscript received March 9, 1998.
Accepted for publication September 8, 1998.

Abstract □ Artificial neural networks applied to in vitro–in vivo correlations (ANN–IVIVC) have the potential to be a reliable predictive tool that overcomes some of the difficulties associated with classical regression methods, principally, that of providing an a priori specification of the regression equation structure. 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. Configuration variables included a combination of architectural structures, learning algorithms, and input–output association structures. The initial training set consisted of two formulations and included the dissolution from each of the six cells in the dissolution bath as inputs, with associated outputs consisting of 1512 pharmacokinetic time points from nine patients enrolled in a crossover study. A third formulation IVIVC data set was used for predictive validation. Using these data, a total of 29 ANN configurations were evaluated. The ANN structures included the traditional feed forward, recurrent, jump connections, and general regression neural networks, with input–output association types consisting of the direct mapping of the dissolution profiles to the pharmacokinetic observations, mapping the individual dissolution points to the individual observations, and using a "memorative" input–output association. The ANNs were evaluated on the basis of their predictive performance, which was excellent for some of these ANN models. This work provides a basic foundation for ANN–IVIVC modeling and is the basis for continued modeling with other desirable inputs, such as formulation variables and subject demographics.

Introduction

It is often desirable to determine a good correlation between the in vitro dissolution data and the in vivo pharmacokinetics. This modeled relationship can then be used in product development or in establishing dissolution specifications. Many of the previous examples of defining an in vitro−in vivo correlation (IVIVC) in drug studies follow simple linear models, relating a parameter or a time point descriptive of the dissolution to a parameter or a time point descriptive of the pharmacokinetic absorption.[1−3] However, often a model is unsuccessful in completely describing the IVIVC and sometimes no relationship can be determined. The number of possible variables, the model being unable to account for some physiological rate determining process, and the possible amount of variability intrinsic to the parameters of these modeled relationships are some examples of these difficulties.[4−6]

It is an aim of the IVIVR Cooperative Working Group to extend the development of IVIVC using newer modeling tools, such as those in the field of artificial intelligence. The self-organizational properties of these methods and their ability to incorporate a large number of possible variables and relationships without a predefined model structure encourage the evaluation of artificial neural networks (ANN) in determining an IVIVC.

The term ANN refers to a group of algorithms used for pattern recognition and data modeling. As its name implies, ANN systems are loosely based on neural physiology, using the concept of a highly interconnected system of parallel processing units. It is not the intent of this paper to intensively cover ANN methodology; a review of the development of ANN can be found elsewhere,[7] as well as a complete description of the theory involved and tested in this research.[8−10]

The application of neural network concepts is relatively new to the field of pharmacokinetics and pharmacodynamics. An introduction to ANN as applied to the field of pharmacokinetics was given by Erb, who described the common back-propagation learning algorithm[11] and demonstrated its ability to be used as a bayesian classifier using simulated data.[12] The application of real pharmacokinetic data for the task of learning interspecies scaling, using different input−output data formats and neural network configurations, has been described by Hussain et al.[13] They also described the problem of the lack of a structured set of rules or guidelines in determining network configuration variables, such as the number of hidden nodes, the necessary number of training iterations, and the proper data format. Application of an ANN to predict drug behavior based on patient demographics and patient factors, and a comparison to a more traditional approach has also been described.[14] The implementation of ANNs in pharmacodynamics to predict the central nervous system activity due to the drug alfentanil has been examined.[15] ANNs have also been successfully implemented in problems very similar to IVIVC, such as product development[16] and quantitative structure−pharmacokinetic relationships,[17] again using a common back-propagation approach to ANN learning.

It is our eventual aim to develop a methodical approach to ANN−IVIVC, and the intent of this paper and current research is to show the feasibility of ANN−IVIVC by presenting the results from some common ANN configurations and data formats, using a relatively small set of IVIVC data for training and prediction. The results from a newer ANN, which does not use the back-propagation iterative learning paradigm, are also presented. It is not the intent of this paper to rigorously compare this method

* Author to whom correspondence should be addressed at: GloboMax LLC 7250 Parkway Dr., Suite 430 Hanover, MD 21076. Telephone: 410-712-9500. FAX: 410-712-0737. E-mail: youngd@globomax.com.
[†] In Vitro−In Vivo Relationship Cooperative Working Group.
[‡] Pharmacokinetics−Biopharmaceutics Laboratory.
[§] Food and Drug Administration.
[||] Elan Corporation plc.

10.1021/js970148p CCC: $18.00
Published on Web 11/19/1998

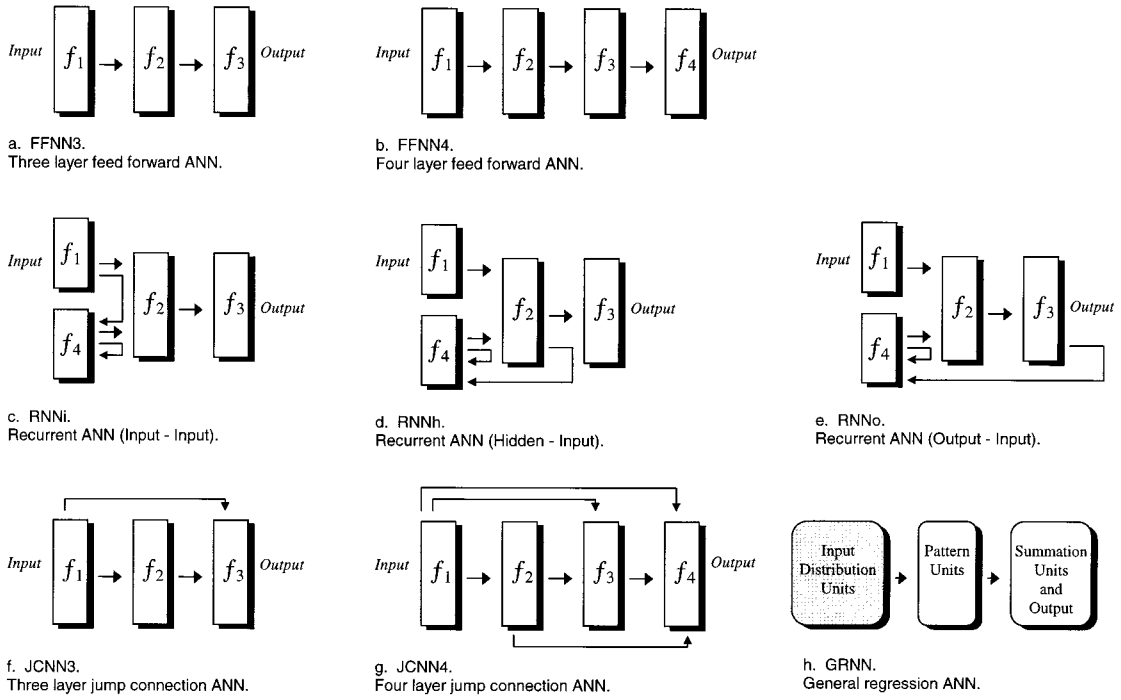

Figure 1—Block diagrams of the ANN architectures used in the study. Almost all of the architectures employ some type of back-propagation learning. The general regression neural network uses a statistical technique known as kernel regression.

with other available IVIVC methods, suggest this approach as appropriate for different situations, nor suggest that this method is a simpler means for correlating in vitro–in vivo data. Such comparisons and analyses, which are ongoing within our group, require numerous types of data and experimental design.

Methods

Description of Terms—There is a great deal of diversity in terminology within the ANN literature, and it is necessary to define some important terms that will be used throughout this paper. These defintions are not intended to be a glossary of terms in the area of artificial intelligence programming, but rather a necessary beginning in establishing a common foundation. Beginning with the format of data, each correlation between a set of input variable(s) and a set of output variable(s) is defined as an *input–output association* and all of the input–output associations collectively form a *pattern file.* ANNs learn using a pattern file known as a *training pattern file,* in which the individual input–output associations from this file are presented either randomly or in a defined order. A *validation pattern file,* which does not use data involved in training, is applied to the trained ANN to test the predictive capabilities of the trained ANN. *Memorization* of an ANN is a common problem referring to when an ANN has become over-trained and is mimicking the training pattern file. This is a situation where the ANN does not have the ability to predict well using inputs other than those found in the training pattern file. To avoid memorization and to establish a criteria to stop training, a *test pattern file* is constructed using associations from the training pattern file and applied periodically to the ANN during training. The input–output associations from the test pattern file are not used in the training, but as a measure of memorization and as a criteria to terminate training. The basic functional element of a neural network is defined as a *node,* which possesses a certain type of *transfer function.* The connections between nodes carry a weighting term, which is the element of an ANN that is continually adjusted during training. Nodes are arranged in layers: *input layer, hidden layer(s),* and *output layer.* It is necessary to define these terms prior to describing any methodology or discussing the results to prevent any confusion due to a lack of formalized vocabulary in artificial intelligence research.

Trial Data and Software—The data set reported in this study included in vitro inputs (% dissolved) and in vivo outputs (plasma concentrations). Inputs in the training pattern files consisted of the dissolution values from two extended-release formulations with seven dissolution time points each, at which six tablets were tested per formulation. Each formulation was administered to nine individuals in a crossover trial. Corresponding ANN outputs consisted of the drug plasma concentrations, sampled at 15 time points following oral tablet administration. A third extended release formulation, with the same experimental setup and part of the same crossover study, was used as a validation set. Success of the ANNs was based on the prediction of the validation profile.

The complete data set, consisting of the three separate formulations and correlated kinetics, was chosen because the pharmacokinetics for this drug was known to follow a "flip–flop" first-order absorption model, where the absorption of this drug was relatively slow compared with its elimination. This situation, as a trial for ANN–IVIVC, gave reasonable assurance that the dissolution kinetics could be considered a variable influential throughout the pharmacokinetic profile.

All ANN training and application were performed using Ward Systems' software package, NeuroShell 2.[18]

Choice of Neural Network Configurations—Part of the aim of these studies and the focus of this paper was to determine the best network configuration for this relatively small set of in vitro–in vivo data using a systematic approach. This determination was performed by initially selecting a set of common network architectures. Network configuration variables that are the focus here included the type of input–output association, network architecture, and for some architectures, the number of hidden layers.

Network Architectures—These trials involved four basic types of ANN architectures contained within the NeuroShell 2 software: traditional feed forward neural networks, recurrent neural networks, jump connection neural networks, and generalized regression neural networks. Diagrams of these network structures, with the nodes represented collectively as functional blocks, are shown in Figure 1. Including the type of network architecture and the number of hidden layers, we have tested a total of eight types of network architectures. A summary of each ANN architecture is given below and shown in Table 1.

Two of these ANN architectures are the common three and four layer feed forward neural networks (FFNN3 and FFNN4 shown in Figures 1a and 1b, respectively), which have one and two hidden layers, respectively. To give the network functional flexibility, a linear function ($f_1(x) = x$) was used for the nodes in the input

Table 1—Summary of the Eight Types of ANN Architectures Tested

type	architecture	data presentation	test set	node configuration (input-hidden-output)
FFNN3	feed forward 3 layers	random	≈10% randomly selected	linear–logistic–logistic
FFNN4	feed forward 4 layers	random	≈10% randomly selected	linear–logistic–logistic–logistic
RNNi	recurrent input–input	rotational	individual subject with single formulation with all 6 dissolution sets	linear–logistic–logistic
RNNh	recurrent hidden–input	rotational	individual subject with single formulation with all 6 dissolution sets	linear–logistic–logistic
RNNo	recurrent output–input	rotational	individual subject with single formulation with all 6 dissolution sets	linear–logistic–logistic
JCNN3	jump connections 3 layers	random	≈10% randomly selected	linear–logistic–logistic
JCNN4	jump connections 4 layers	random	≈10% randomly selected	linear–logistic–logistic–logistic
GRNN	general regression neural network	N/A	≈10% randomly selected to determine smoothing	N/A

Table 2—Pattern Files Constructed from the Different Input–Output Association Types[a]

#	association type	association input(s)	association output(s)	pattern structure	#associations/ formulation
1	functional	(7) dissolution set$_j$ (t_{DISS1}:t_{DISS7})	(15) PK$_i$ (t_{PK1}:t_{PK15})	subject$_{1-9}$ (dissolution set$_{1-6}$)	54
2	time series	(8) t_{PK}, dissolution set$_j$ (t_{DISS1}:t_{DISS7})	(1) PK$_i$ (t_{PK})	subject$_{1-9}$ (dissolution set$_{1-6}$($t_{PK\ 1-15}$))	810
3	time series	(2) $t_{DISS|PK}$, dissolution set$_j$ (t_{DISS})	(1) PK$_i$ (t_{PK}) only those outputs where $t_{PK} = t_{DISS}$	subject$_{1-9}$ (dissolution set$_{1-6}$($t_{DISS|PK\ 1-7}$))	378
4	time series memorative association	(1–8) t_{PK}, dissolution set$_j$ (if $t_{DISS} < t_{PK}$)	(1) PK$_i$ (t_{PK})	subject$_{1-9}$ (dissolution set$_{1-6}$($t_{PK\ 1-15}$))	810

[a] PK = Pharmacokinetic observations (in vivo); DISS = % dissolved (in vitro); i = subject number; j = tablet number; t_{PK} = pharmacokinetic time point; t_{DISS} = dissolution time point.

layer and a logistic function ($f_2(x)$, $f_3(x)$, $f_4(x) = 1/(1 + \exp(-x))$) was used for each node in the hidden and output layers.

The recurrent networks defined as RNNi, RNNh, and RNNo (Figures 1c, 1d, and 1e, respectively) had recurrent connections to the input, hidden, and output layers, respectively. Recurrent architectures do not have the input to output feed forward design; their recurrent design allows for a "long-term memory". This type of structure has the ability to learn sequences of input−output associations. Therefore, the setup of input−output data sequence presented in training and prediction becomes very important to these networks and their order must be considered, such as when the data is presented as a time series. The transfer function of each node in the hidden and output layers was set as a logistic function, whereas the input layer nodes were set to a linear function. The fourth layer can be called the network's "long-term memory", and has no node functionality. It contains the contents of the connected layer as it was in the previous training. These types of networks have been shown to work well with time series data that depend on history.[19]

The following two network architectures are a type of ANN known as jump connections. In this type of back-propagation network, every layer is connected in a feed forward manner. Three and four layer jump connection ANN architectures, designated JCNN3 and JCNN4 (Figures 1f and 1g), respectively, were used because they may be possible alternatives to the traditional feed forward structures, and were given the same node functions as FFNN3 and FFNN4.

Unlike the other ANNs, a general regression neural network (GRNN) is not an ANN that uses the back-propagation learning paradigm. The GRNN is an ANN system that involves a statistical technique known as kernel regression and requires the data to only be iterated once through the network during training.[10] Training of the GRNN was done using two options available in the NeuroShell 2 software. Patterns were compared based on their differences in distance using the vanilla or Euclidean distance metric, and smoothing was performed by using a genetic algorithm that selectively breeds a solution to the problem using a fitness function as a measure of survival. The mean squared error of the outputs in the test pattern file is used as the fitness function in the software. The NeuroShell 2 software manual contains descriptions of this terminology and these options with suggestions for their implementation.[18]

In all but the GRNN, training was performed by the software using a back-propagation scheme. Back-propagation is the most widely used learning algorithm employed in training neural networks. In its simplest form, it should be very familiar to those involved in data fitting and regression because it is an iterative gradient descent procedure that minimizes the error.[20]

Network Training Criteria—In all nonrecurrent networks, 10% of the training pattern file was randomly selected and placed aside as a test pattern file during training. Recurrent neural networks, however, rely on previous history, which required the data to be presented as a time series across input−output associations. This presentation also included the application of any test or validation pattern file. For this reason, the test pattern file applied to any recurrent architecture consisted of the associations from the ninth subject/second formulation, kept in time sequence.

The test pattern file was not included in the training, but was used as a periodic measure of the network's ability to successfully predict while being trained. The test pattern file was applied to the network after every 200 input−output associations (training events), using the NET-PERFECT feature in the NeuroShell 2 software. The prediction of the outputs in the test pattern file was used as a stop criterion. In each case, the network was directed to stop training after 20000 training events following a minimum error, and the weights corresponding to that minimum were saved as the trained ANN.

Associations and Pattern Files—Four different types of pattern files constructed from the same data were selected for evaluation. The pattern files, named ASSOCIATION 1 through 4, were unique because of different formatting of the input−output association. Each training pattern file had corresponding validation and test pattern files constructed with the same type of input−output association. A diagram of each type of input−output association is shown in Figure 2, indicating the structure of the relationship as well as using subscripts to show how the data are formatted across associations to create a pattern file. General descriptions of the input−output associations are given next, and a summary of the constructed pattern files is shown in Table 2.

ASSOCIATION 1—Initially the data were presented as the functional relationship shown in Figure 2a, with an input−output association that used all of the pharmacokinetic concentration values from an individual as an output set associated with an input set that consisted of the dissolution profile from an individual tablet. The pattern file then contained each pharmacokinetic observation set associated with each of the six tablet dissolution profiles. The dissolution mean was not used in this type of pattern file, or in either of the other three pattern files.

ASSOCIATION 2—Like ASSOCIATION 1, the input−output associations in this pattern file included the complete kinetic set of dissolution values for each tested tablet, but each was associated with a single respective pharmacokinetic output. Collectively, the input−output association lines of the pattern file formed a pharmacokinetic time sequence. The pharmacokinetic time point

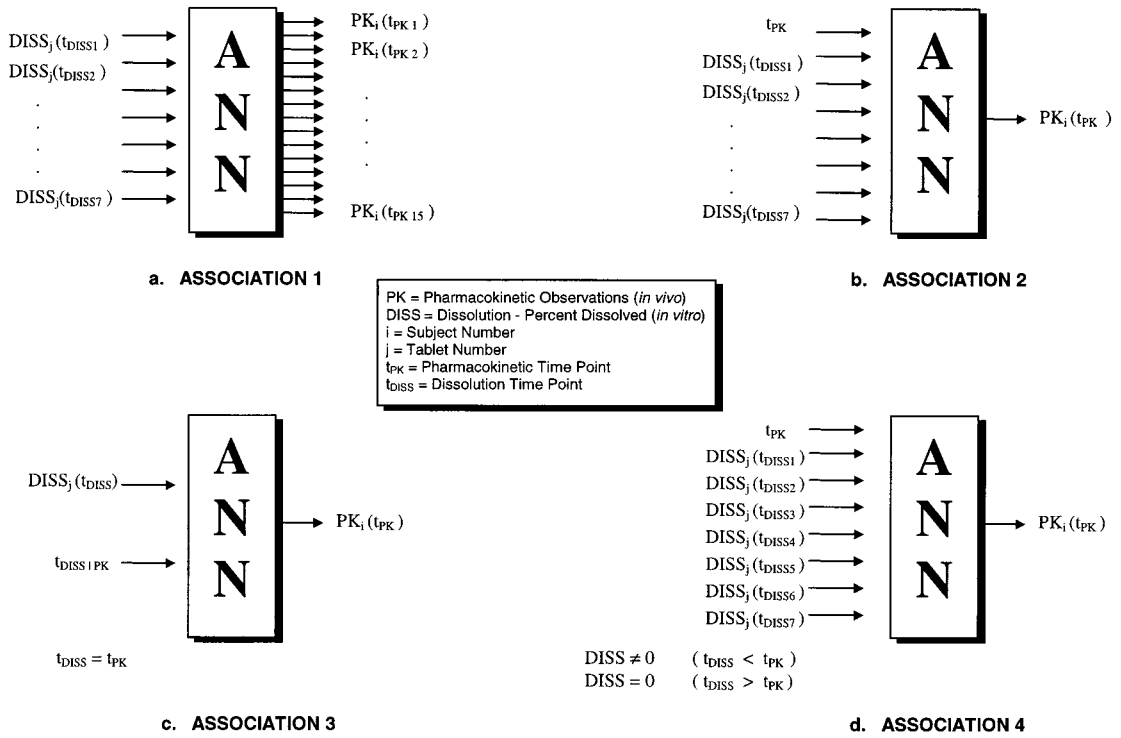

Figure 2—Diagrams of the input–output associations used in pattern files ASSOCIATION 1 through 4. The subscripts i and j are used to show the number of associations across association types.

was also included as an input. A diagram of this relationship is shown in Figure 2b.

ASSOCIATION 3—The input−output associations of this pattern file, shown in Figure 2c, consisted of each in vitro value as an input associated with each in vivo output. Pharmacokinetic observations with no directly associated dissolution observations were not used in the training. The time of the observation was also added as an input.

ASSOCIATION 4—This pattern file, shown in Figure 2d, attempted to unite some of the more desirable features of the previous pattern files that included presenting the entire dissolution profile per tested tablet as inputs (ASSOCIATIONS 1 and 2), presenting the data as a time sequence (ASSOCIATIONS 1, 2, and 3), and utilizing all of the in vitro data (ASSOCIATIONS 1, 2, and 4). Pattern file ASSOCIATION 4 was a sequential time series and included previous dissolution values as inputs. This type of pattern file can be termed a *memorative association* and was a type of time progressive synthesis neural network configuration described by Veng-Pedersen.[15] The output consisted of the pharmacokinetic concentration value, whereas the inputs were the pharmacokinetic time point and all the dissolution values that preceded that point in time. Dissolution values that occurred after that pharmacokinetic time point were set to zero in the pattern file and were interpreted as null inputs by the software.

Results and Discussion

A total of 29 network configurations, which included the eight different types of ANN architectures and four types of input−output associations, were tested. The three recurrent architectures were not used with ASSOCIATION 1, because this type of relationship did not have a sequential format across associations.

Each network was trained as described in the methodology, and the weights of each of the 29 trained ANNs were saved. Inputs from the training and validation pattern files were applied to the trained networks and these respective ANN outputs were compared to the actual observations. Shown in Table 3 are the results for both the training and validation pattern files for each ANN configuration, consisting of the correlation coefficient (R^2), mean prediction error (MPE), and mean absolute error (MAE). These values are defined as

$$R^2 = \frac{\sum(y - \hat{y})^2}{\sum(y - \bar{y})^2}$$

$$\text{MPE} = \frac{1}{N}\sum(\hat{y} - y)$$

$$\text{MAE} = \frac{1}{N}\sum|y - \hat{y}|$$

where y = actual observation, $\hat{y}$ = ANN prediction, $\bar{y}$ = average observation, and N = number of observations. Also shown is the ratio of R^2 between the predictions and training pattern files, as an indicator of possible network memorization.

Results from these trials, as summarized in Table 3, reflect the success of each ANN configuration with this particular set of IVIVC data. These results are measured by the precision and bias of the outputs from the training and validation pattern files. The ANNs attempted to determine a mean concentration curve based on the information contained in the dissolution kinetics, and in some configurations, attempted to account for the variability in the pharmacokinetics due to variability in the dissolution kinetics.

More than half of these ANN configurations could be considered successful in predicting the pharmacokinetic data from the dissolution kinetics. The better network architectures for this IVIVC data set seem to be the feed forward and the generalized regression architectures, based on their ability to give good model predictions with all four pattern files. The more successful pattern files included formatting the data as a functional relationship (ASSOCIATION 1) and as a memorative pattern file (ASSOCIATION 4). An example of a model prediction from one of these network configurations is shown in Figure 3 and Figure 4. In this example, the ASSOCIATION 4

Table 3—Statistical Results for the 29 Network Configurations Applied to ANN–IVIVC

network	training set R^2	training set MPE	training set MAE	validation set R^2	validation set MPE	validation set MAE	R^2 ratio (prediction/ training)
Association 1							
FFNN3	0.878	–0.229	3.110	0.803	–1.431	3.992	0.915
FFNN4	0.880	–0.196	3.109	0.790	–1.435	4.089	0.897
RNNi	N/A	N/A	N/A	N/A	N/A	N/A	N/A
RNNh	N/A	N/A	N/A	N/A	N/A	N/A	N/A
RNNo	N/A	N/A	N/A	N/A	N/A	N/A	N/A
JCNN3	0.875	0.454	3.207	0.819	–0.415	3.957	0.937
JCNN4	0.872	0.574	3.238	0.815	–0.526	4.007	0.935
GRNN	0.877	–0.001	3.180	0.800	–1.302	3.997	0.912
Association 2							
FFNN3	0.136	–1.371	9.486	0.142	–2.180	9.588	1.041
FFNN4	0.865	0.493	3.316	0.792	–0.910	4.179	0.915
RNNi	0.732	3.642	5.080	0.742	3.061	5.422	1.013
RNNh	0.728	4.351	5.101	0.692	3.731	5.749	0.950
RNNo	0.741	4.273	5.268	0.763	2.312	5.336	1.029
JCNN3	0.101	–2.234	9.520	0.114	–3.062	9.612	1.134
JCNN4	0.078	–1.365	9.653	0.149	–1.396	9.592	1.910
GRNN	0.878	–0.035	3.184	0.788	–1.357	4.178	0.897
Association 3							
FFNN3	0.749	–0.825	4.184	0.608	–2.931	5.352	0.812
FFNN4	0.752	–0.514	4.167	0.620	–2.623	5.296	0.824
RNNi	0.539	4.583	6.237	0.594	2.642	6.088	1.102
RNNh	0.420	6.089	7.171	0.559	3.493	6.381	1.333
RNNo	0.478	4.347	6.668	0.578	1.404	6.093	1.210
JCNN3	0.745	–0.073	4.267	0.614	–2.259	5.327	0.825
JCNN4	0.746	–0.235	4.238	0.617	–2.391	5.265	0.827
GRNN	0.769	0.031	3.998	0.634	–2.110	5.092	0.825
Association 4							
FFNN3	0.846	–0.076	3.540	0.771	–1.573	4.465	0.911
FFNN4	0.854	–0.538	3.389	0.770	–1.683	4.280	0.901
RNNi	0.658	4.650	5.626	0.571	5.540	7.196	0.868
RNNh	0.696	0.744	5.187	0.580	5.174	7.135	0.834
RNNo	0.596	1.252	6.111	0.628	3.404	6.303	1.054
JCNN3	0.841	–0.192	3.607	0.789	–0.798	4.230	0.938
JCNN4	0.856	–0.286	3.357	0.787	–1.618	4.098	0.919
GRNN	0.859	–0.015	3.303	0.773	–0.749	4.295	0.900

pattern file was used to train the GRNN. Following training, the dissolution values from the training pattern files were used as inputs to predict the pharmacokinetic data. Comparisons of the actual observations with these ANN outputs are shown in Figure 3. The dissolution values from the validation pattern file were then presented to this trained ANN, interpolating the pharmacokinetic predictions shown in a comparison with the actual pharmacokinetic observations in Figure 4. These figures are representative of those network configurations that did relatively well, with a MPE close to zero for both training and validation sets, an R^2 of >0.85 for the training set, and an R^2 of >0.77 for the validation set (R^2 ratio >0.9).

The common and relatively simple FFNN architectures (Figures 1a and 1b) worked well with this data set based on the predictions of the validation pattern file outputs, especially when the data were presented as a functional relationship (ASSOCIATION 1). For some IVIVC data, however, these types of architectures may not work as well as time series predictors. Some IVIVC tend to be nonlinear, requiring the ANN to incorporate past history. The feed forward structure cannot incorporate history, but this may be accounted for if the data is arranged as a memorative association (ASSOCIATION 4), which also proved successful with this IVIVC data. An interesting example of network performance as a function of configuration variables is seen in comparing the ASSOCIATION 2–FFNN3 trial with the ASSOCIATION 2–FFNN4 trial,

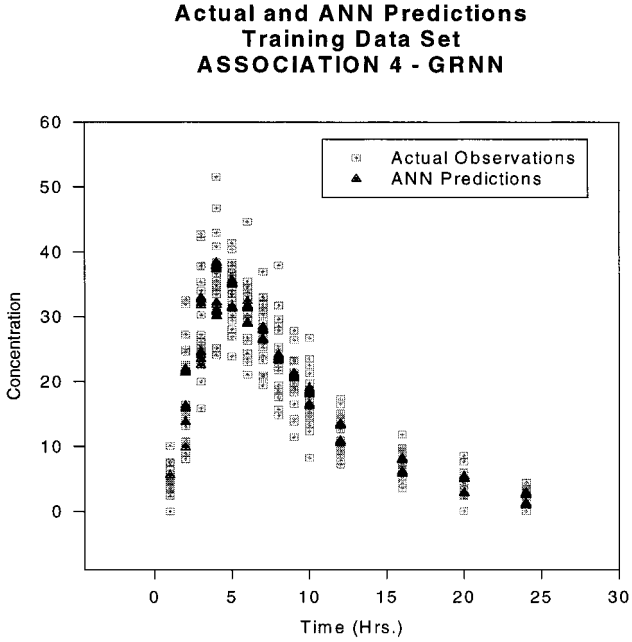

Figure 3—Actual pharmacokinetic observations from the training data set are compared with ANN pharmacokinetic predictions using in vitro inputs from the training data set. The GRNN was trained with the training pattern file ASSOCIATION 4.

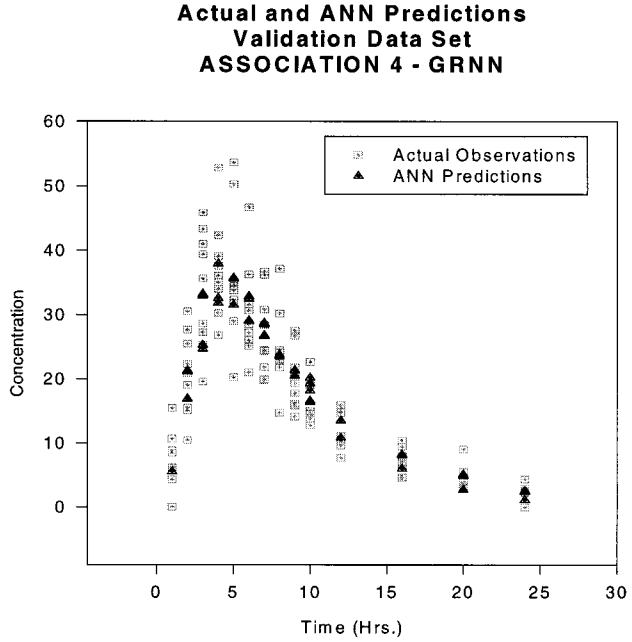

Figure 4—Actual pharmacokinetic observations from the validation data set are compared with ANN pharmacokinetic predictions using in vitro inputs from the validation data set. The GRNN was trained with the training pattern file ASSOCIATION 4.

where the additional hidden layer improved prediction dramatically. Practical ANN experience has shown that a majority of problems can be solved with a three-layered design, and that a four-layered ANN may be prone to fall into a local minima.[21] However, with this IVIVC data set formatted as the ASSOCIATION 2 pattern file, a four-layered feed forward structure predicted well, whereas the three-layered ANN failed to converge on a solution.

The JCNN architectures (Figures 1f and 1g), which are structurally very similar to the FFNN architectures, also compared well. The lack of any significant improvement in describing these data, however, suggests that the additional jump connections were not necessary.

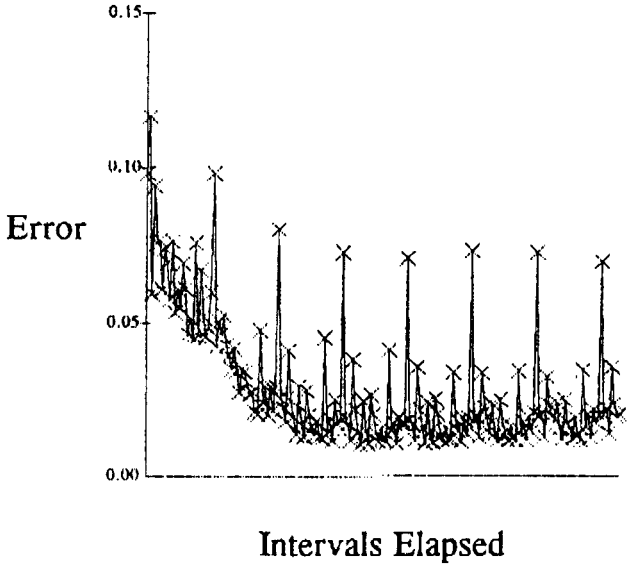

Figure 5—Error between actual observations and ANN predictions of the test pattern file during training of the recurrent network RNNo with ASSOCIATION 3 training pattern file. Each point (x) represents the error when the test pattern file is tested during network learning. The test pattern file was applied once after every 200 input–output associations of the training pattern file.

A thorough description of the theoretical development and implementation of the GRNN is too involved for this discussion, but can be referenced in the work by Specht.[10] This network structure requires only a single iteration to provide estimates of continuous variables and has a distinct advantage in that it can converge to a linear or nonlinear regression surface, even with relatively little data. This property as well as its success in these trials with all of the different types of pattern files encourage its use in this type of research.

A problem in implementing the recurrent network structures, as reflected in the results presented here, was the determination of a stopping criteria. As with all back-propagation network training performed in these trials, the test pattern file was applied periodically throughout training and was a measure used to indicate the completion of training before the onset of memorization. Because of the importance in keeping a sequential structure with recurrent networks, rather than the test set randomly constructed from the training data, the ninth subject receiving the second formulation was used as the selected test pattern file. This procedure proved to be a good measure that prevented memorization, but biased the trained network in the favor of this test set. A typical plot of the error as a function of the training intervals for a recurrent network is shown in Figure 5. Once a minimum is found, network training oscillates across subjects until training is stopped, the time of which is based on the number of iterations following a minimum in the test set. A trained network based on a minimum corresponding to the test pattern file is saved, biasing the results to the data used as the test set. So, although the recurrent ANN structures produced fair results, these results were biased to the selected test pattern file. Better results may be expected if a better "average" or unbiased test pattern can be found, the intersubject variability can be better described using additional inputs, or another method to protect against network memorization can be found.

In most ANN structures, the number of inputs and outputs dictate the number of input and output nodes, respectively. Hence, more inputs and outputs lead to a more complicated network structure. Although relatively successful here, the ASSOCIATION 1 pattern file had a total of 15 outputs, which must be considered in the evaluation of these types of input−output associations for ANN−IVIVC. When the level of complexity of the structure increases, the likelihood of obtaining a good solution decreases. As this research progresses, it is expected that the number of independent variables, or inputs, will be increased, adding to the complexity of the models and making input−output associations, like that found in ASSOCIATION 1, undesirable. The memorative type of input−output association used to construct the ASSOCIATION 4 pattern files worked well with all eight ANN architectures. This type of pattern file had the advantage of being a generalized format with a single output, which also allows the network to incorporate relationships from the previous inputs. A disadvantage of this pattern file is its inability to use information from dissolution values with a time of dissolution greater than the corresponding time of the pharmacokinetic observation. This pattern file may not predict well in data sets where the dissolution lags behind the pharmacokinetics.

The number of network configurations can be immense when considering some of the variables examined here, such as network architecture, data formats, and number of hidden layers, and considering some of the other possible network configuration variables not addressed; for examples, number of hidden nodes, additional network structures, learning algorithms, and the different types of node transfer functions. The number of hidden nodes in these trials were set to the software defaults, which were allowed to be conservatively large, because the periodic application of the test pattern file helps to prevent memorization. Some of the more common network architectures were examined in this study, but there were many more that may prove to be as good or better in ANN−IVIVC. Other possible structures that may prove to be applicable are the newer multilayer networks that include a lag in the data between the dependent and independent variables.[9] The node transfer functions were limited to the linear and logistic transfer functions in this study, but many other different types of functions, such as a limit, competitive, hyperbolic tangent, sine, or Gaussian may be used.

Conclusion

We have evaluated a number of possible network configurations, many of which successfully predicted a mean in vivo plasma concentration profile using dissolution kinetics. This work has demonstrated the feasibility of ANN−IVIVC by showing a number of potential ANN configurations that can be considered successful with this data set, but has illustrated a need for a methodical approach in applying ANN to problems. Additional input variables, including subject demographics, dissolution method variables, and formulation variables, are currently being introduced to attempt to account for the nonrandom error associated in the relationship. The ANN−IVIVC has the potential to establish complex relationships and may also possess the ability to interpolate pharmacokinetic parameters and profiles given formulation specifications. Also, algorithms and software currently exist to reverse map from the plasma concentration curve to the dissolution profile, possibly forecasting a range of dissolution profiles that will provide bioequivalent formulations.

References and Notes

1. Sullivan, T. J.; Sakmar, E.; Wagner, J. G. Comparative bioavailability: a new type of *in vitro-in vivo* correlation exemplified by prednisone. *J. Pharmacokinet. Biopharm.* **1976**, *4(2)*, 173−181.

2. Graffner, C.; Nicklasson, M.; J.-E. Lindgren, Correlations between in vitro dissolution rate and bioavailability of alaproclate tablets. *J. Pharmacokinet. Biopharm*. **1984**, *12(4)*, 367−380.
3. Caramella, C.; Ferrari, F.; Bonferoni, M. C.; Sangalli, M. E.; De Bernardi Di Valserra, M.; Feletti, F.; F., Galmozzi, M. R. In vitro/in vivo correlation of prolonged release dosage forms containing diltiazem HCl. *Biopharm. Drug Dispos.* **1993**, *14(2)*, 143−160.
4. Wood, R. W.; Martis, L.; Gillum, A. W.; Roseman, T. J.; Lin, L.; L.; Bernardo, P. In vitro dissolution and in vivo bioavailability of commercial levothyroxine sodium tablets in the hypothyroid dog model. *J. Pharm. Sci.* **1990**, *79(2)*, 124−127.
5. Barr, W. H.; Zola, E. M.; Candler, E. L.; S−M. Hwang, Tendolkar, A. V.; Shamburek, R.; Parker, B.; Hilty, M. D. Differential absorption of amoxicillin from the human small and large intestine. *Clin. Pharmacol. Ther.* **1994**, *56(3)*, 279−285.
6. Levy, G.; Hollister, L. E. Inter- and intra subject variations in drug absorption kinetics. *J. Pharm. Sci.* **1964**, *53(12)*, 1446−1452.
7. Anderson, J. A. *An introduction to neural networks*; The MIT Press: Cambridge, MA, 1995.
8. Gallant, S. I. *Neural Network Learning and Expert Systems*; The MIT Press, Cambridge, MA, 1993.
9. Hagan, M. T.; Demuth, H. B.; Beale, M. *Neural Network Design*; PWS Publishing: Boston, 1996.
10. Specht, D. F. A general regression neural network. *IEEE Trans. Neural Networks* **1991**, *2(6)*, 568−576.
11. Erb, R. Introduction to back-propagation neural network computation. *Pharm. Res.* **1993**, *10(1)*, 165−170.
12. Erb, R. The back-propagation neural network − A bayesian classifier. Introduction and applicability to pharmacokinetics. *Clin. Pharmacokinet.* **1995**, *29(2)*, 69−79.
13. Hussain, A. S.; Johnson, R. D.; Vachharajani, N. N.; Ritschel, W. A. Feasibility of developing a neural network for prediction of human pharmacokinetic parameters from animal data. *Pharm. Res.* **1993**, *10(3)*, 466−469.
14. Brier, M. E.; Zurada, J. M.; Aronoff, G. R. Neural network predicted peak and trough gentamicin concentrations. *Pharm. Res.*, **1995**, *12(3)*, 406−412.
15. Veng-Pedersen, P.; Modi, N. B. Application of neural networks to pharmacodynamics. *J. Pharm. Sci.*, **1993**, *82(9)*, 918−926.
16. Hussain, A. S.; Yu, X. Q.; Johnson, R. D. Application of neural computing in pharmaceutical product development. *Pharm. Res.* **1991**, *8(10)*, 1248−1252.
17. Gobburu, J. V. S.; Shelver, W. H. Quantitative structure-pharmacokinetic relationships (QPSR) of beta blockers derived using neural networks. *J. Pharm. Sci.* **1995**, *84(7)*, 862−865.
18. *NeuroShell 2 Manual Third Edition*; Ward Systems Group, Inc., Executive Park West, 5 Hillcrest Drive, Frederick, MD 21702, 1995.
19. Elman, J. L. Finding structure in time. *Cognitive Science* **1990**, *14(1)*, 179−211.
20. Haykin, S. *Neural networks: a comprehensive foundation*; Macmillan: New York, 1994.
21. De Villiers, J.; Barnard, E. Back-propagation neural nets with one and two hidden layers. *IEEE Trans. Neural Networks* **1992**, *4(1)*, 136−141.

Acknowledgments

The work reported in this manuscript and ongoing research in ANN-IVIVC is supported by the Elan Corporation, plc. as part of their overall sponsorship of the IVIVR Cooperative Working Group. The views expressed by Ajaz S. Hussain are his own and may not reflect the policies of the FDA. Figures are reproduced with the permission of Plenum Publishing from the book entitled *In Vitro-In Vivo Correlations*.

JS970148P