

SVETLANA IBRIĆ
 ZORICA DJURIĆ
 JELENA PAROJČIĆ
 JELENA PETROVIĆ

Institute of Pharmaceutical Techno-
 logy, Faculty of Pharmacy,
 Belgrade, Serbia

REVIEW PAPER

UDC 615.4:004.032.26

DOI: 10.2298/CICEQ09042271

ARTIFICIAL INTELLIGENCE IN PHARMA- CEUTICAL PRODUCT FORMULATION: NEURAL COMPUTING*

The properties of a formulation are determined not only by the ratios in which the ingredients are combined but also by the processing conditions. Although the relationships between the ingredient levels, processing conditions, and product performance may be known anecdotally, they can rarely be quantified. In the past, formulators tended to use statistical techniques to model their formulations, relying on response surfaces to provide a mechanism for optimization. However, the optimization by such methods can be misleading, especially if the formulation is complex. More recent advances in mathematics and computer science have led to the development of alternative modeling and data mining techniques which work with a wider range of data sources: neural networks (an attempt to mimic the processing of the human brain); genetic algorithms (an attempt to mimic the evolutionary process by which biological systems self-organize and adapt), and fuzzy logic (an attempt to mimic the ability of the human brain to draw conclusions and generate responses based on incomplete or imprecise information). In this review the current technology will be examined, as well as its application in pharmaceutical formulation and processing. The challenges, benefits and future possibilities of neural computing will be discussed.

Key words: artificial neural networks; pharmaceutical formulation; genetic algorithms; fuzzy logic; optimization.

The development of a commercial product, whether it is a relatively simple formulation (e.g. a capsule, tablet or oral liquid) or a controlled release formulation (e.g. an implant), is always a time-consuming and complicated process. Generally an initial formulation consisting of one or more drugs mixed with various ingredients (excipients) is prepared, and, as development progresses, the choice of these and their levels, as well as the manufacturing process, are changed and optimized as a result of intensive, time-consuming experimentation. These iterations, in turn, result in the generation of large amounts of data, the processing and understanding of which is challenging. In reality, the formulator has to work in a design space that is multi-dimensional and virtually impos-

sible to conceptualize. To date, statistics has been used as one approach to this problem. This method has the advantage of generating clearly expressed models with associated confidence factors. However, for more than three or four inputs, statistical approaches rapidly become unwieldy, so that the formulator is tempted to oversimplify the problem (for example, restricting a study to three input variables) in order to model it. Statistics also often require the assumption of a functional form (for example, linearity) in order to generate a model and such assumptions can be inappropriate for complex tasks like formulation.

In recent years, it has been shown that neural networks can provide an alternative approach [1]. Neural networks are mathematical constructs that are capable of “learning” relationships within data, with no prior knowledge required from the user. The neural network makes no assumptions about the functional form of the relationships; it simply generates and assesses a range of models to determine one that will best fit the experimental data provided to it. As such, increasingly, artificial neural networks (often referred

Corresponding author: S. Ibrić, Institute of Pharmaceutical Technology, Faculty of Pharmacy, Vojvode Stepe 450, 11221 Belgrade, Serbia.

E-mail: ibric@pharmacy.bg.ac.rs

Paper received: 1 July, 2009

Paper accepted: 15 July, 2009

*A part of this study was presented as a section lecture at the 8th Symposium „Novel Technologies and Economics Development“, University of Niš, Faculty of Technology, Leskovac, October 21–24, 2009.

to as ANNs) are used to model a complex behavior in problems like pharmaceuticals formulation and processing.

The models generated by neural networks allow “what if” possibilities to be investigated easily. However, their capabilities are enhanced substantially by combining them with other technologies. For example, using genetic algorithms for optimization, together with neural networks models, has proved to be exceptionally powerful when the formulator must develop a formulation to meet stringent, often conflicting, objectives. The objectives for the optimization can easily and intuitively be defined using another artificial intelligence technology, fuzzy logic. Fuzzy logic has proved to be especially valuable when conflicting properties (for example, hard tablets that disintegrate quickly) are desired. More recently, efforts have been made to integrate the technologies even more tightly, creating new methodologies like neurofuzzy logic, which combines the ability of neural networks to “learn” from data, with fuzzy logic’s capacity to express complex concepts in a simple fashion. These techniques are capable of “mining” the information directly from data, presenting it in the form of easy to understand, actionable rules that can guide the formulator’s future work.

NEURAL NETWORKS

Like humans, neural networks learn directly from input data. The learning algorithms take two forms. Unsupervised learning, where the network is presented with input data and learns to recognize patterns in the data, is useful for organizing amounts of data into a smaller number of clusters. For supervised learning, which is analogous to “teaching” the network, the network is presented with a series of matching input and output examples, and it learns the relationships connecting the inputs to the outputs. Supervised learning has proved most useful for the formulation, where the goal is to determine cause-and-effect links between inputs (ingredients and processing conditions) and outputs (measured properties).

The basic component of the neural network is the neuron, a simple mathematical processing unit that takes one or more inputs and produces an output. For each neuron, every input has an associated weight that defines its relative importance, and the neuron simply computes the weighted sum of all the outputs and calculates an output. This is then modified by means of a transformation function (sometimes called a transfer or activation function) before being forwarded to another neuron. This simple pro-

cessing unit is known as a perceptron, a feed-forward system in which the transfer of data is in the forward direction, from inputs to outputs, only. A neural network consists of many neurons organized into a structure called the network architecture. Although there are many possible network architectures, one of the most popular and successful is the multilayer perceptron (MLP) network. This consists of identical neurons all interconnected and organized in layers, with those in one layer connected to those in the next layer so that the outputs in one layer become the inputs in the subsequent layer. Data flow into the network via the input layer, pass through one or more hidden layers, and finally exit via the output layer (Fig.1). In theory, any number of hidden layers may be added, but in practice multiple layers are necessary only for those applications with extensive nonlinear behavior, and they result in extended computation time. It is generally accepted that the performance of a well-designed MLP model is comparable with that achieved by classic statistical techniques.

Unlike conventional computer programs, which are explicitly programmed, supervised neural networks are “trained” with previous examples. The network is presented with example data, and the weights of inputs feeding into each neuron are adjusted iteratively until the output for a specific network is close to the desired output. The method used to adjust the weights is generally called back propagation, because the size of the error is fed back into the calculation for the weight changes. There are a number of possible back propagation algorithms, most with adjustable parameters designed to increase the rate and degree of convergence between the calculated and the desired (actual) outputs. Although training can be a relatively slow process, especially if there are large amounts of data, once trained, neural networks are inherently fast in execution.

FUZZY LOGIC

Conventional logic demands that a proposition is either true or false. This maps onto a conventional set theory, so that a hypothesis lies either in the “true” set, or lies wholly outside it. That is, the membership function in the “true” set is either 1 (the hypothesis is true) or 0 (the hypothesis lies outside the “true” set, and is false). In real life, though, these black-and-white concepts may be of little utility. An oft-cited example is the definition of a comfortable room temperature. If a temperature of 20 °C is defined as “comfortable”, conventional logic would dictate that 19 or 21 °C, which lie outside this set, are “uncomfortable”). A very

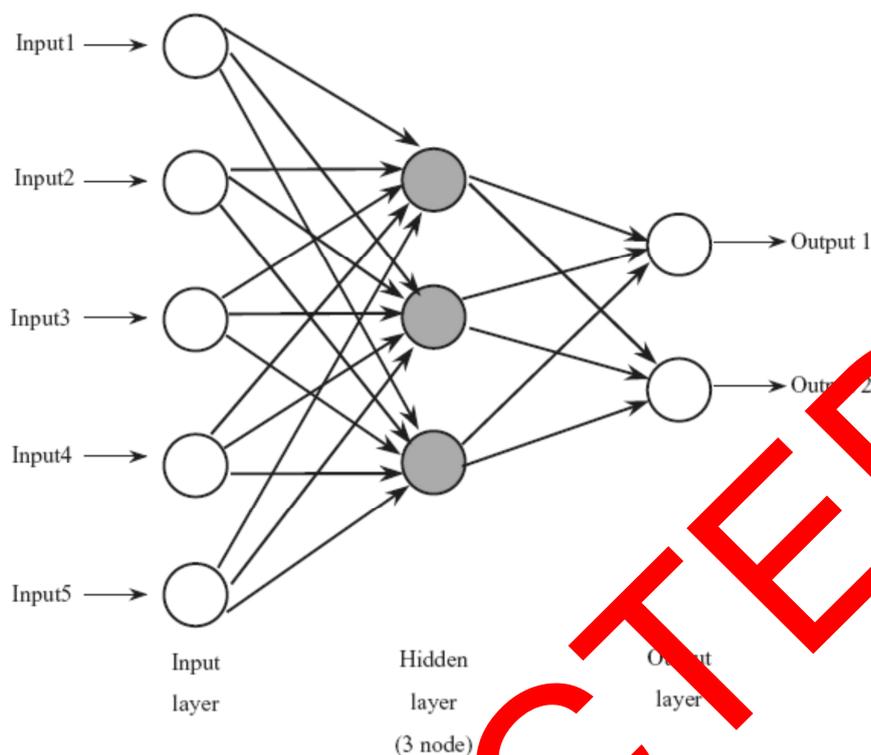


Figure 1. Diagram of a multilayer perceptron with one hidden layer.

complex set of rules would be required to define “comfortable” using conventional logic. Fuzzy logic allows more intuitive statements and descriptors to be used, so is more akin to the linguistic format adopted by formulators. Fuzzy logic is based on the concept of fuzzy sets introduced in the 1960s by Lotfi Zadeh [1]. For fuzzy sets, membership functions are not restricted to be 0 or 1, but can take any continuous value between these limits. In the context of comfortable room temperature, for example, a temperature of 17 °C might have a membership of 0.4 in the “hot” set and 0.6 in the “cold” set, as illustrated in Figure 2.

Fuzzy logic can be especially useful in describing target properties for optimizations. For example, the formulator might be seeking a tablet disintegration time of 300 s, *i.e.* any value less than 300 s has a desirability of 1 (*i.e.* 100%). But a tablet which disintegrates in 310 s is not entirely undesirable (as crisp logic would insist), and instead might be assigned a desirability value of 0.9.

Fuzzy logic is also widely used in the area of process control, because it allows rules to be expressed in a simple linguistic form IF (A) THEN (B) with an associated confidence function that is related to the

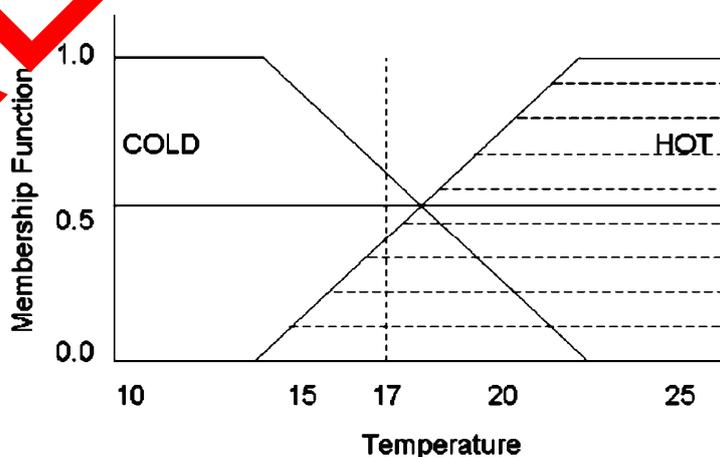


Figure 2. Fuzzy logic representation of room temperature.

set membership. To understand how this is used for the process control, consider a simple example of a fan heater governed by 4 rules, summarized in Figure 3. These rules map onto the four fuzzy sets COLD, COOL, WARM and HOT. So, for example, if the room temperature is 18 °C, then by Rule 2 the fan speed is medium, with truth level 0.7, and by Rule 3 the fan speed is low, with truth level 0.3. The process of defuzzification then allows the correct and smooth adjustment of the fan speed.

NEUROFUZZY LOGIC

Because fuzzy logic allows objectives to be expressed in simple terms, it complements neural network modelling. In the case of neurofuzzy logic, as the name suggests, the fuzzy logic is tightly coupled with a neural network.

Neurofuzzy logic combines the ability of neural networks to learn from data with fuzzy logic's ability to express complex concepts intuitively. This creates a degree of transparency for the otherwise "black box" neural network models, leading to the term "grey box modelling" being applied for these methods. Neurofuzzy logic has proved to be exceptionally suited to data mining, since it not only can develop good models from data, but it also has the capability of expressing these as linguistic IF...THEN rules.

The neurofuzzy architecture is in essence a neural network with two additional layers for fuzzification of inputs and defuzzification of outputs [1]. The modeling capabilities of neurofuzzy systems depend on the number, shape and distribution of the fuzzy membership input functions. In the simplest case,

only two, LOW and HIGH would suffice. In some cases, it is appropriate to add more; for example, a problem showing a quadratic dependency would require at least LOW, MEDIUM and HIGH in order that it be properly represented. Where data are scarce, relatively few membership functions should be used. As the number and complexity of the inputs increases, the rules become more complicated, and this can make them difficult to understand.

EVOLUTIONARY COMPUTING

Evolutionary computing is a general term that describes computational processes in which solutions evolve, using rules of inheritance, recombination (or cross-over), mutation and selection. The particular subset of this, evolutionary algorithms, has found the application in the formulation research.

GENETIC ALGORITHMS

Genetic algorithms were pioneered in the 1970s by John Holland [4]. They provide a search technique which is particularly suited to optimization; a trial population is assumed, and this evolves in an iterative process. During this process, an initial population of solutions is generated, and the fitness of each member of the population is assessed. The fittest solutions then become the "parents" of the next generation. Allowing some recombination and mutation introduces a further degree of novelty into the population so that the genetic algorithm is more likely to find a global optimum solution. It is this ability to find the global optimum in a complex design space which renders

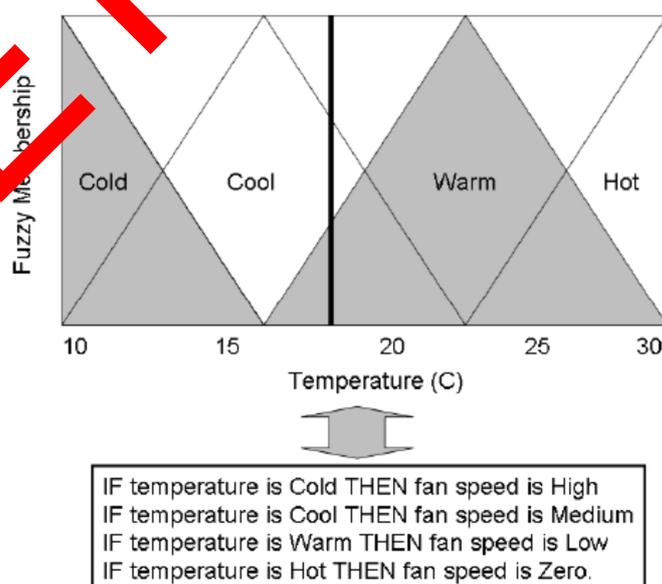


Figure 3. Fuzzy logic for process control of room temperature.

genetic algorithms so useful, especially when compared with more directed searches like conjugate gradient and steepest descent methods.

One requirement for genetic algorithms is that a criterion of “fitness” can be defined. This can vary from problem to problem. For multi-dimensional optimization, it has proved useful to define an objective function which is a weighted sum of the desirability of each of the properties. The use of weights in the sum allows some properties to assume more importance than others, and the fittest solutions are those that best meet the overall objectives.

In defining the desired values of the properties, fuzzy logic (discussed previously) provides a useful framework. Two typical desirability functions are shown in Figure 4. One is for the case where disintegration time of a tablet is most desirable below 240 s, and completely undesirable above 360 s. The second is

for the case where the disintegration time should lie between 240 and 360 s, becoming progressively less desirable as it moves farther away from this region.

Genetic programming, generally regarded as a subset of genetic algorithms, is the most recent of the techniques reviewed here, having been widely popularized only in the 1990s primarily by Koza [5]. As yet, it has had a limited use in pharmaceutical formulation, but it shows great promise since it has the learning capabilities similar to that of neural networks but the transparency associated with a single forward mathematical expression. In genetic programming, each solution is a “tree”, in which each tree node has an operator function and each terminal node is an operand. These trees provide an alternative way of representing equations; Figure 5 shows one such example. An initial population of solutions is assumed, and as with other evolutionary methods, the fitness of

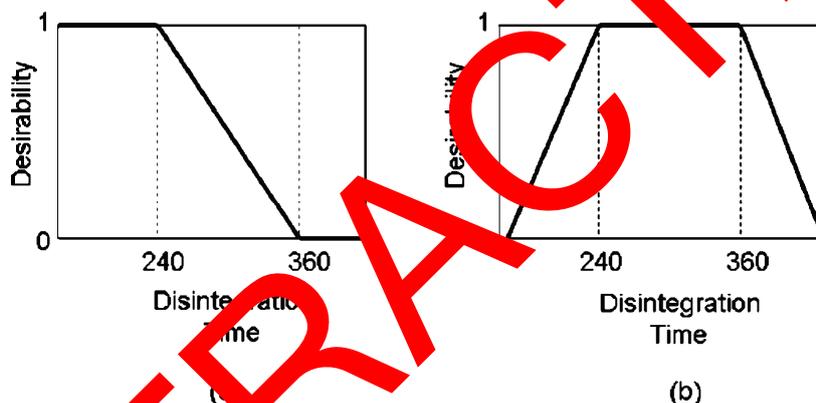


Figure 4. Desirability functions for tablet disintegration time (a) below 240 s and (b) between 240 and 360 s.

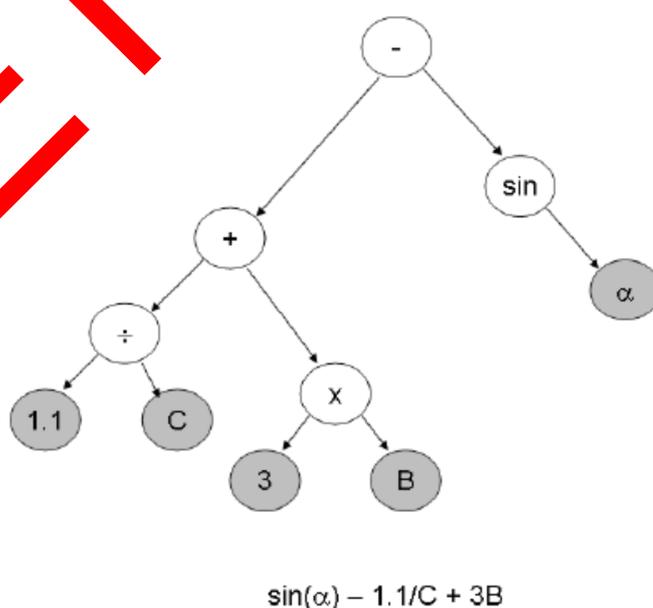


Figure 5. Genetic programming tree representation of equation.

each member is assessed. The population then evolves allowing crossover (whereby parts of trees are swapped) and mutation; these operations are illustrated in Figure 6. The evolution is biased so that the fittest solutions are emphasized in successive generations, leading to increased improvement in the fit of the model to the training data.

In the same way as for other genetic algorithms, a criterion of fitness needs to be defined. The simplest criterion would simply minimize the mean-squared error between the calculated and actual values, but this could result in an overly complex, and potentially over-fitted, model. Therefore, it is often appropriate to use a model assessment criterion (such as structural risk minimization) to penalize those solutions whose added complexity does not return significant new knowledge.

Genetic programming currently suffers from the disadvantage that it is time consuming, and its application is less well understood in the formulation domain than are neural networks. Nonetheless they are attractive possibilities for future work, because they can produce “transparent” models.

INTEGRATED SOFTWARE

The technologies described above are well suited to data mining and modelling, but in their raw form require a degree of expertise. To be truly useful to product formulators, the technologies described above need to be integrated into packages that use sensible default values for the parameters, and that incorporate all of the essential tools. For example, to develop a package aimed at producing optimized formulations, the modelling capability of neural networks combines well with the optimization provided by genetic algorithms. As discussed above, fuzzy logic complements this by providing a useful framework for defining the objectives for the optimization in a clear and intuitive way.

In addition to the specific neural and evolutionary technologies, it is useful also to integrate some basic statistics (both for examining the data and for assessing the quality of the models, using ANOVA Analysis of Variance statistics), to provide a visualization capability. Such integrated packages are now available commercially and are proving useful in the pharmaceutical industry. An early exemplar of this is CAD/Chem [1], more recently, since CAD/Chem is no

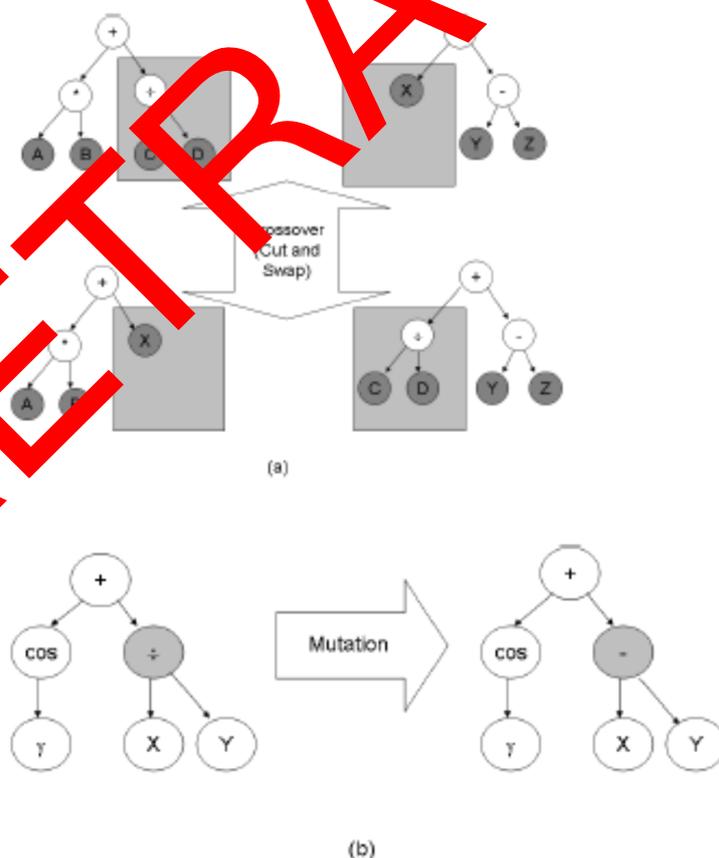


Figure 6. Crossover (a) and mutation (b) for genetic programming trees.

longer available, INForm from Intelligensys has been developed. A data mining package based on neuro-fuzzy logic, FormRules, is also commercially available, with integrated visualization as well as statistical techniques to assess the quality of models.

APPLICATIONS

Over the past fifteen years the technology described above has been used extensively to model and optimize formulations from simple to very complex. It has also been used to a lesser degree in pharmaceutical processing.

Oral formulations - immediate release

The earliest study in this field was reported by Turkoglu *et al.* [7] who modeled a direct compression tablet formulation containing hydrochlorothiazide in order to maximize tablet strength and select the best lubricant. In another study, Kesavan and Peck [8] modeled a tablet formulation of caffeine in order to relate both formulation (diluent type and concentration, binder concentration) and processing variables (type of granulator, method of addition of binder) with granule and tablet properties (friability, hardness, and disintegration time). Both these investigations showed that neural networks performed better than conventional statistical methods. Subsequently, the data of Kesavan and Peck were reanalyzed using a combination of neural networks and genetic algorithms [9]. This showed that the optimum formulation depends both on the relative importance placed on the output properties and on the constraints applied both to the levels of the ingredients and to the processing variables. Many optimum formulations could be produced, depending on the “trade-offs” that could be accepted for different aspects of product performance. The same data have been studied using neurofuzzy computing [10]. Useful rules were automatically generated, highlighting the most important factors for each property and their interdependencies. As expected, the rules for friability were the inverse of those for tablet hardness, while the rules for disintegration time involved the diluent itself, the binder concentration, the method of addition of binder (wet or dry) and the method of granulation.

In a series of papers, Bourquin *et al.* [11-14] highlighted the advantages and disadvantages of neural networks in formulating the immediate release tablets. The data generated by these authors have been used by Plumb *et al.* [15] to compare three different neural network programs and four classes of training algorithms in terms of capability of generating predictive models. They found that the most predict-

tive models from each neural network varied with respect to the optimum network architecture and training algorithm. No significant differences were found in the predictive ability of these models.

Recently, the same data have been analysed by Shao *et al.* [16] who compared neurofuzzy techniques with neural networks. They found that neurofuzzy methods developed models that were almost as good as neural networks (as determined using analysis of variance statistics). However, the neurofuzzy results had the advantage that they also generated rules that were presented in a simple understandable format.

Rocksloh *et al.* [17] used neural networks successfully to optimize the crushing strength and disintegration time of a high-dose plant extract tablet. In another study using both neural networks and genetic algorithms, Doi *et al.* [18] showed the advantages of combining these technologies in the formulation of antacid tablets. Chen *et al.* [19] used neural networks to predict the drug content and hardness of intact tablets of ophylline mixed with microcrystalline cellulose from their near-infrared spectra. The model proved better than a statistical model generated with the same data. The superiority of neural network models over statistical models has also been found by Sathe and Vernekar [20] this time for predicting the dissolution of 28 diltiazem immediate release tablet formulations. In a recent paper, Lindberg and Colbourn [21] working in Sweden and the UK used neural networks, genetic algorithms and neurofuzzy to analyze historical data from three different immediate release formulations. In all cases, the models generated performed satisfactorily in producing tablets with specific desired properties.

Apart from immediate release tablet formulations, neural networks have also been applied to modelling the immediate release capsule formulations [22], rapidly disintegrating or dissolving tablets [23] and a novel oral microemulsion formulation of rifamycin and isoniazid for the treatment of children during the continuation phase of tuberculosis [24]. Solid dispersion formulations of ketoprofen have recently been modelled using both neural networks and neurofuzzy with good predictability [25]. The study has also been extended by the addition of a microemulsion formulation [26].

In a detailed evaluation of both neural networks and guided evolutionary simulated annealing for the modeling and optimization of a tablet coating formulation, Plumb *et al.* [27,28] concluded that, for conflicting properties such as crack propagation and film opacity that displayed highly curved responses with respect to the formulation inputs (*e.g.* pigment particle

size, pigment concentration and film thickness), classical experimental designs to map the experimental space are inappropriate for neural network modeling. However, if a pseudo-random design was used, it was possible to model and optimize the film coating, predicting formulations that were either crack resistant or that were fully opaque. Similar formulations have also been studied using neurofuzzy computing [29], where rules were generated relating both the opacity and crack resistance to the input variables. In line with expectations, the technique discovered for itself that for maximum opacity the films needed to be thick with a high pigment concentration and a small pigment size.

Controlled release oral formulations

Chen *et al.* [30] used an artificial neural network (ANN) and pharmacokinetic simulations in the design of controlled-release formulations. Seven formulation variables and three other tablet variables (moisture, particle size and hardness) for 22 tablet formulations of a model drug were used as the ANN model inputs. *In vitro* cumulative percentage of drug released at 10 different sampling time points were used as outputs. The ANN model was developed and trained from the input and the output data sets using CAD/Chem software. The trained ANN model was used to predict optimal formulation compositions based on two desired *in vitro* dissolution-time profiles and two desired *in vivo* release profiles. The authors assumed that the dissolution is the rate-limiting step in the *in vivo* absorption of the drug that the fraction of drug absorbed *in vivo* is linearly related to the *in vitro* dissolution of the drug. Three out of four predicted formulations showed very good agreement between the ANN predicted and the observed *in vitro* release profiles based on difference factor, f_1 , and similarity factor, f_2 .

Zupančić Borić *et al.* [31] developed an ANN model to optimize diclofenac sodium sustained release matrix tablets. Formulation variables including concentrations of acetyl alcohol, polyvinylpyrrolidone K 30 and magnesium stearate, and sampling time were chosen as inputs. Twelve hidden nodes were included in the hidden layer. The percentage of the drug released at each sampling time point was used as the output. A trained ANN model was employed to predict release profile and optimize the formulation composition based on the percentage of the drug released.

Tokayama *et al.* [32] developed a simultaneous optimization technique in which the ANN model was used to optimize controlled release theophylline tablets prepared with controse, the mixture of hydroxy-

propylmethyl cellulose with lactose and cornstarch. The release profiles of theophylline were characterized as the sum of the fast and slow release fractions. To build the ANN model, the amounts of controse, cornstarch and compression pressure were selected as causal factors; the initial weight of theophylline, the rate constant in the fast release fraction and the release rate constant in the slow release fraction were chosen as response variables. The results predicted by the trained ANN model agreed well with the observed values. Assuming that the release rate of theophylline in the GI tract is equal to the absorption rate, the rate constant in the fast release fraction and the release rate constant in the slow release fraction were used as absorption rate constants. The plasma concentrations profiles were simulated based on the simulated plasma concentration profiles. The optimization of the controlled release theophylline tablets was performed by a generalized distanced function method using the optimal release parameters.

A generalized regression neural network (GRNN) was used in the design of extended-release aspirin tablets by Ibrić *et al.* [33,34]. Ten aspirin matrix tablet model formulations were prepared with Eudragit RS PO [33]. The amount of Eudragit RS PO and compression pressure were selected as causal factors. *In vitro* dissolution-time profiles at four different sampling time points, as well as coefficients n (release order) and $\log k$ (release constant) from the Peppas equation were estimated as release parameters. A set of release parameters and causal factors were used for training. The optimized GRNN model was used to predict the formulation and process factors for the optimized formulations, which would give the desired *in vitro* drug release profiles. The two optimized formulations were then prepared and tested *in vitro*. The comparison between the GRNN predicted and observed *in vitro* profiles, and estimated coefficients indicated that there is no difference between the predicted and experimentally observed drug release profiles for the two tested formulations based on the difference factor, f_1 and similarity factor, f_2 . The same authors applied GRNN for predicting the drug stability, and *in vitro-in vivo* correlation [35,36].

In coated tablets, the controlling mechanism for drug release is generally the film applied to the tablet, although in some circumstances the release may be controlled in addition by the tablet core formulation. In a detailed study of 125 formulations for small tablets prepared from a model drug embedded in a hydrophilic matrix and then coated with an enteric polymer, Leane *et al.* [37] were able to apply various input feature selection algorithms, including genetic algo-

rithms, to evaluate the relative importance of the input variables. They then used a neural network to model subsets of the data, with the less significant inputs eliminated. As expected, the elimination of the less significant inputs results in more generalized predictive models. In another study, Wu *et al.* [38] used a neural network to model the formulation of salbutamol sulfate osmotic pump tablets, using the amount of hydroxypropyl methyl cellulose and polyethylene glycol present in the cellulose acetate coating, in addition to the coating weight, as control factors. Using the model, the authors predicted the release parameters for 1000 formulations, from which they selected an optimum with the desired release pattern.

In pelleted or multi-particulate formulations, the drug release mechanism can be controlled either by using a rate controlling matrix or by the use of films. The pellets are either produced by using extrusion and spheronization or by layering onto sugar cores. In some cases, the pellets may be tableted. In others, they are packed into hard gelatin capsules. Peh *et al.* [39,40] applied both multi-layer perceptrons and recurrent neural networks [41] to model successfully the release of theophylline from a matrix controlled release pellet formulation prepared using extrusion and spheronization. In another study on pellets, this time prepared using the layering technique followed by polymer film coating, Vaithiyalingam *et al.* [42] compared the modelling and optimization abilities of simplex and neural network procedures. They concluded that from very limited data sets, simplex optimization was more appropriate although neural networks were “a valuable and predictive tool”. In a following study, Vaithiyalingam and Khan [43] compared a response surface methodology and neural networks for modelling and optimizing the effect of the process and formulation variables on the release profile of verapamil hydrochloride. In each case, the observed drug release profile of the optimized formulation was close to that predicted from the model. In another study, fluidized bed manufactured, enteric-coated, omeprazole pellets compressed into tablets were analyzed using neural networks [44]. From the model, the authors were able to predict a positive correlation between the tablet strength and the concentration of the microcrystalline cellulose used as a compression aid. However, the degradation of the omeprazole in such media was also dependent on the microcrystalline cellulose concentration.

A bimodal drug delivery system consisting of pellets coated with pectin and chitosan has recently been modeled using neural networks with five different training algorithms [45]. The authors concluded that

those networks trained using gradient descent back-propagation algorithms outperformed the others. The textural properties of a novel pellet formulation capable of extensive gelation and swelling in biological fluids, coined “gelisphere” by its inventor, have been modeled by Pillay and Danckwerts [46] using both statistics and neural networks. The authors concluded that neural networks were a more reliable data predictor in the design of their system.

BENEFITS AND ISSUES

Although there is a great deal of interest in neural computing, the quantified information on the benefits has been harder to find. From the applications described above, benefits that could be included:

- effective use of incomplete data sets,
- rapid analysis of data,
- ability to accommodate more data and retrain the network (retrain the model),
- effective exploration of the total design space, irrespective of complexity,
- ability to accommodate constraints and preferences and
- ability to generate understandable rules.

A survey [47] on the use of 93 neural computing applications in 75 UK companies covering all business sectors, the major benefits identified were the improved quality, improved response times, and increased productivity. Eighty-four percent of users were satisfied or very satisfied with their systems with only three percent expressing dissatisfaction. Business benefits, specifically for the domain of product formulation (albeit for nonpharmaceuticals), have been given as [48]:

- enhancement of product quality and performance at low cost,
- shorter time to market,
- development of new products,
- improved customer response,
- improved confidence and
- improved competitive edge.

As this new technology moves from the realm of academe into practical application, there are also issues regarding the implementation of neural computing. Users in the previously cited study were asked to identify where they had experienced problems. Thirty-nine percent had found problems related to software and lack of development skills; this will be reduced as commercial packages come into wider use and there is less need for bespoke in-house systems with their high programming and maintenance burden.

However, even when commercial packages are used, there are a number of features that should be present before neural computing can be used to advantage. The problem must be numeric in nature, and reasonable quantities of data should be available to train an adequate model. The greatest benefits are achieved for multidimensional problems where it is difficult to express any analytic model and difficult to abstract the rules by any other mechanism than neural computing. It helps if the problem is of practical importance, is part of the organization's essential activity, and meets a real business need. Pharmaceutical formulation meets these criteria well, and neural computing can be expected to provide significant benefits in industry in the future.

REFERENCES

- [1] R. C. Rowe, E. A. Colbourn, *Pharm. Tech. Eur.* **9** (1996) 46-55
- [2] L. A. Zadeh, *Fuzzy sets. Information and Control* **8** (1965) 338-353
- [3] K. M. Bossley, *Neurofuzzy Modelling Approaches in System Identification*, PhD Thesis, University of Southampton, UK, 1997
- [4] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, USA (1975)
- [5] J. R. Koza, *Genetic Programming - On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992
- [6] W. H. Verduin, *Better Products Faster - A Practical Guide to Knowledge-Based Systems for Manufacturing*, Professional Publishers, USA, 1997
- [7] M. Turkoglu, R. Ozarslan, A. Akkr, *Eur. J. Pharm. Biopharm.* **41** (1995) 315-322
- [8] J. G. Kesavan, G. E. Beck, *Pharm. Dev. Technol.* **1** (1996) 391-404
- [9] E. A. Colbourn, R. C. Rowe, *Pharm. Tech. Eur.* **8** (1996) 46-55
- [10] R. C. Rowe, E. A. Colbourn, *Pharm Tech Eur.* **12** (2000) 24-27
- [11] J. Bourquin, H. Schmidli, P. van Hoogevest, H. Leuenberger, *Pharm. Dev. Technol.* **2** (1997) 111-121
- [12] J. Bourquin, H. Schmidli, P. van Hoogevest, H. Leuenberger, *Eur. J. Pharm. Sci.* **6** (1998) 287-300
- [13] J. Bourquin, H. Schmidli, P. van Hoogevest, H. Leuenberger, *Eur. J. Pharm. Sci.* **7** (1998) 5-16
- [14] J. Bourquin, H. Schmidli, P. van Hoogevest, H. Leuenberger, *Eur. J. Pharm. Sci.* **7** (1998) 17-28
- [15] A.P. Plumb, R.C. Rowe, P. York, M. Brown, *Eur. J. Pharm. Sci.* **25** (2005) 395-405
- [16] Q. Shao, R. C. Rowe, P. York, *Eur. J. Pharm. Sci.* **28** (2006) 394-404
- [17] K. Rocksloh, F.R: Rapp, S. Abu Abed, W. Mueller, M. Reher, G. Gauglitz, P. C. Schmidt, *Drug Dev. Ind. Pharm.* **25** (1999) 1015-1025
- [18] Q. M. Do, G. V. Dang, N. Q. Le, *Tap Chi Duoc Hoc* **6** (2000) 16-19
- [19] U. Chen, S. Thosor, R. Forbess, M. Kemper, R. L. Rubinovitz, A. J. Shukla, *Drug Dev. Ind. Pharm.* **27** (2001) 623-631
- [20] P. M. Sathe, J. Venitz, *Drug Dev. Ind. Pharm.* **29** (2003) 349-355
- [21] N. O. Lindberg, E. A., Colbourn, *Pharm. Tech. Eur.* **16** (2004) 35-39
- [22] R. Zolotariou, J. Anwar, *J. Pharm. Pharmacol.* **50** (1998) 190
- [23] H. Sunada, Y. Bi, *Powder Technol.* **122** (2002) 188-198
- [24] S. Agatonovic-Kustrin, B. D. Glass, M.H. Wisch, R.G. Alany, *Pharm. Res.* **20** (2003) 1760-1765
- [25] A. Mendyk, R. Jachowicz, *Expert Sys. Appl.* **28** (2005) 285-294
- [26] A. Mendyk, R. Jachowicz, *Expert Sys. Appl.* **32** (2007) 1124-1131
- [27] A. P. Plumb, R. C. Rowe, P. York, C. Doherty, *Eur. J. Pharm. Sci.* **16** (2002) 281-288
- [28] A. P. Plumb, R. C. Rowe, P. York, C. Doherty, *Eur. J. Pharm. Sci.* **18** (2003) 259-266
- [29] R. C. Rowe, C. G. Dadgar, *Pharm. Sci. Technol. Today* **2** (1999) 455-497
- [30] Y. Chen, W. McCall, V. Baichwal, M.C. Meyer, *J. Control. Release* **59** (1999) 33-41
- [31] D. Zupancic Bork, F. Vrecar, F. Kozjek, *Eur. J. Pharm. Sci.* **5** (1997) 163-170
- [32] K. Takayama, A. Morva, M. Fujikawa, Y. Hattori, Y. Obata, T. Nagai, *J. Control. Release* **68** (2000) 175-186
- [33] S. Ibric, M. Jovanovic, Z. Djuric, J. Parojcic, Lj. Solomun, *J. Control. Release* **82** (2002) 213-222
- [34] S. Ibric, M. Jovanovic, Z. Djuric, J. Parojcic, S.D. Petrovic, Lj. Solomun, B. Stupar, *Pharm. Sci. Tech.* **4** (2003) 62-70
- [35] S. Ibric, M. Jovanovic, Z. Đurić, J. Parojčić, Lj. Solomun, B. Lučić, *J. Pharm. Pharmacol.* **59** (2007) 745-750
- [36] J. Parojčić, S. Ibric, Z. Djurić, M. Jovanovic, O. I. Corrigan, *Eur. J. Pharm. Sci.* **30** (2007) 264-272
- [37] M. M. Leane, I. Cumming, O. Corrigan, *Pharm. Sci. Technol.* **4** (2003) 218-229
- [38] T. Wu, W. Pao, J. Chen, R. Shang, *Drug Dev. Ind. Pharm.* **26** (2000) 211-215
- [39] K.K. Peh, C.P. Lim, S.S. Qwek, K.H. Khoti, *Pharm. Res.* **17** (2000) 1386-1398
- [40] S.S. Qwek, C.P. Lim, K.K. Peh, *Int. J. Comp. Intel. Appl.* **1** (2001) 187-292
- [41] W.Y. Goh, C.P. Lim, K.K. Peh, K. Subari, *Neural Comput. Appl.* **10** (2002) 311-317
- [42] S. Vaithiyalingam, A.A. Zaghloul, I.K. Reddes, M.A. Khan, *Pharm. Tech.* **24** (2000) 60-72
- [43] S. Vaithiyalingam, M.A. Khan, *Int. J. Pharm.* **234** (2002) 179-193
- [44] M. Turkoglu, H. Varol, M. Celikok, *Eur. J. Pharm. Biopharm.* **57** (2004) 277-286
- [45] A. Ghaffari, H. Abdollahi, M.R. Khoshayand, I. Soltani Bozchalooi, A. Dadgar, M. Rafiee-Tehrani, *Int. J. Pharm.* **327** (2006) 126-138
- [46] V. Pillay, M.P. Danckwerts, *J. Pharm. Sci.* **91** (2002) 2559-2570
- [47] C. Rees, *Neural Computing—Learning Solutions—User Survey*. Department of Trade and Industry, London, UK, 1996.
- [48] R. C. Rowe, R. J. Roberts, *Intelligent software for product formulation*, Taylor and Francis, London, 1998.