We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Decision Support Systems for Pharmaceutical Formulation Development Based on Artificial Neural Networks

Aleksander Mendyk and Renata Jachowicz
Dept. of Pharmaceutical Technology and Biopharmaceutics Jagiellonian University
Medical College, Medyczna 9 Str, 30-688 Kraków,
Poland

1. Introduction

Once discovered and established as therapeutic agent, the drug substance is used for pharmacotherapy of various diseases. The drug substance itself has unique properties, which in certain cases do not allow for effective therapy. This is the area, where pharmaceutical technology allows to improve drug substance original characteristics by optimization of pharmaceutical formulation. The latter is a complicated process involving many variables concerning formulation qualitative and quantitative composition as well as technology parameters. This chapter will be dedicated to the computer systems based on artificial neural networks allowing for guided pharmaceutical formulation optimization.

2. Artificial neural networks (ANN) foundations

The artificial neural networks (ANNs) are non-linear, information-processing systems designed in a manner similar to the biological neural structures, which is expressed in the structural and the functional composition of ANNs. The latter is based on so-called connectionist model of neural systems. It assumes that topology and electrophysiology of synapses (connections) in the brain or other biological neural systems are the key factors of neural systems ability to process information (Hertz et al. 1991; Wikipedia, 2009c, Żurada 1992).

One of the several definitions of ANNs is that they are dispersed knowledge processing systems built from so-called "nodes" hierarchically organized into the layers. This definition does not implement the most important feature of ANNs which is their ability to learn on the available data. Thus, ANNs are representatives of Computational Intelligence paradigm in contrast to classical Artificial Intelligence systems, where all the knowledge of the system must be implemented from the scratch by the programmer.

Typical ANN of the most common Multi Layer Perceptron type (MLP) is built on four main elements (Fig. 1):

1. input layer
2. hidden layer(s)
3. output layer
4. connections (weights)

Each layer consists of few "nodes" which in fact are artificial neurons connected between layers via "weights" – artificial synapses. The information flow is unidirectional from the input to the output.
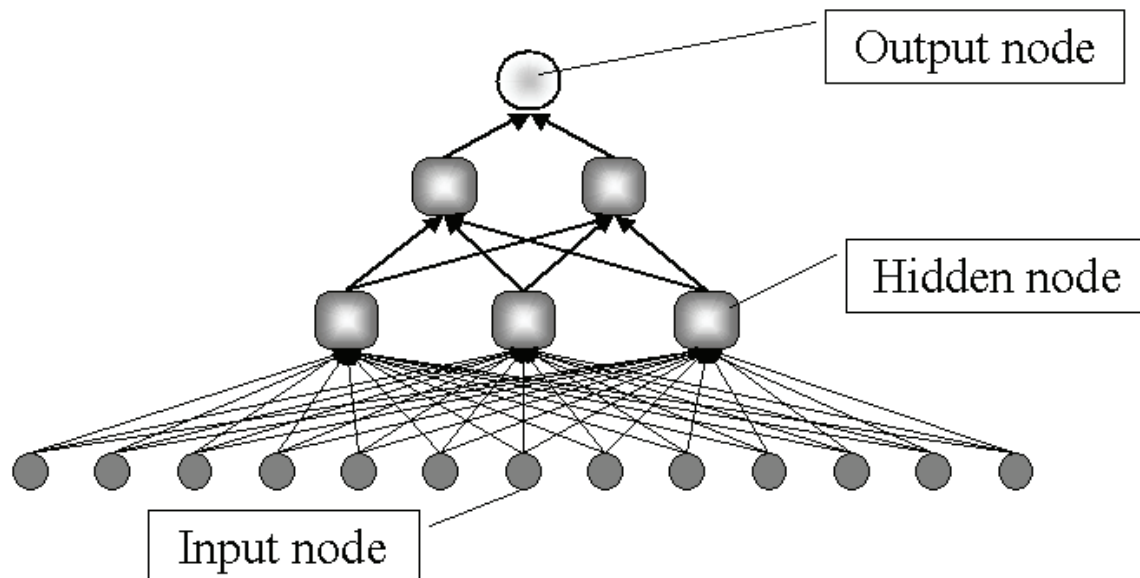

Fig. 1. Typical structure of MLP ANN.

MLP ANN works in two phases:
1.   training
2.   testing

The training phase is based on the iterative presentations of the available data patterns in order to teach ANN to perform designated task. Since MLP ANNs are supervised training systems, they have to be presented with data on the input and output as well. This allows for adjusting weights values in such a manner that ANN becomes competent in the designated task. Adjusting of the weights is performed automatically with use of special algorithm designed for this purpose. One of the most common training algorithms for ANNs is back propagation (BP), where the teaching signal is the difference between current output and the desired one and is propagated backwards from the output layer to the input layer in order to modify weights values (Fig. 2). The whole procedure is automatic and once started does not require any intervention from the user.

According to the connectionist model of the neural systems, ANNs topology is the most important factor influencing their modeling abilities. The topology of ANNs, called also architecture, is expressed in terms of number of layers and nodes in each layer. However, it is not the nodes themselves but number, signs and values of connections between the particular nodes, which encode the knowledge of the system. Since all the BP procedure is automatic, user does not have to put any assumptions about a model shape a priori to the system, thus ANNs represent empirical modeling approach. Automatic training procedure and model identification by ANNs are the most commonly known advantages of these systems. Another advantage is their superior ability to identify non-linear systems. It is because ANNs are usually built on non-linear activation functions, therefore being non-linear systems themselves. Next distinguishing feature of ANNs is their relative ease of dealing with large number of data cases and features. However, so-called curse of
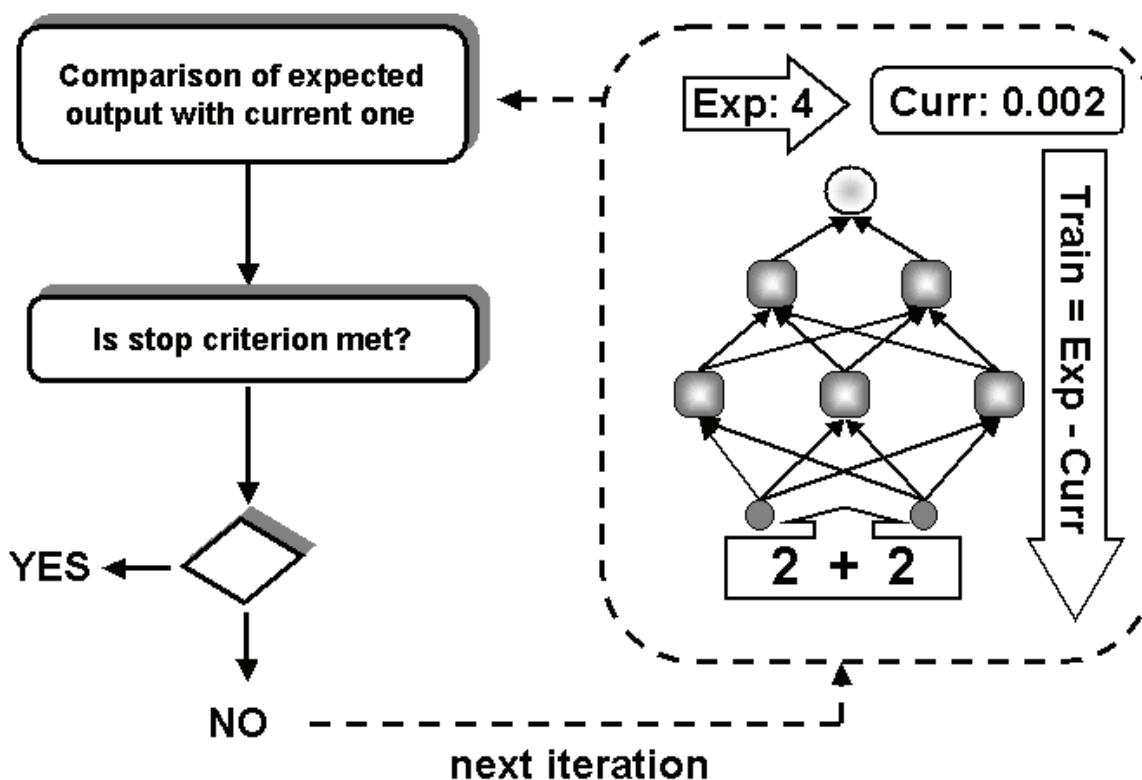
Fig. 2. Scheme of the back propagation algorithm.

dimensionality is also applicable to the ANNs, nevertheless it is less pronounced than for classical statistical systems. Moreover, ANNs are able to decide on inputs importance, thus providing sensitivity analysis feature, which is a way to reduce unnecessary inputs. It improves system performance but also provides knowledge about analyzed problem derived from ANNs behavior. Therefore, ANNs are also used as data mining tools allowing for automated knowledge extraction.

All the features of ANNs described above, allow using them as generic, empirical modeling tools in vast areas of science and technology:

- economy
- engineering
- chemistry
- neurobiology
- medicine and pharmacy

Although, it is impossible to present all applications of neural networks, there might be named major areas of their usage:

- signal processing (noise reduction, compression)
- pattern recognition and features extraction (handwriting, facial recognition, medical imaging, fraud detection)
- forecasting (financial, medical, environmental).
- data mining

Pharmaceutical applications of ANNs are still far from being routine, however ANNs are gradually coming into the focus in different pharmacy areas: pharmacokinetics (Brier & Aronoff, 1996; Brier & Żurada, 1995; Chow et al., 1997; Gobburu & Chen, 1996; Veng-Pedersen & Modi, 1992), drug discovery and structure-activity relationships (Huuskonen. et al, 1997; Polański, 2003; Taskinen & Yliruusi, 2003), pharmacoeconomics and epidemiology (Polak & Mendyk, 2004; Kolarzyk et al, 2006), in vitro in vivo correlation (Dowell et al., 1999) and pharmaceutical technology (Behzadia et al. 2009; Hussain et al., 1991; Bourquin et al., 1998a, 1998b, 1998c; Chen et al., 1999; Gašperlin et al., 2000; Kandimalla et al., 1999; Mendyk & Jachowicz, 2005, 2006, 2007; Rocksloh et al., 1999; Takahara et al., 1997; Takayama et al., 2003; Türkoğlu et al., 1995).

3. Empirical modeling as decision support systems (DSS):

3.1 General remarks

Decision support systems (DSS) are usually computer information processing tools that support decision-making activities in the field of particular interest (Wikipedia, 2009c). As computer tools, they are generally understood as an extension of commonly known expert systems – the systems derived from artificial intelligence field (AI). The expert systems' definition "enhancement" allows, among other differences, to use "black box" models in contrast to the classical hard AI systems, where the system behavior is algorithmic, thus understandable on the every level of its action. DSS exploit every available techniques of data processing in the benefit of accuracy of decision making support. This includes ANNs as well, which will be advertised in this chapter as very suitable tools for DSS in the pharmaceutical technology.

Every DSS has to include basic set of elements:

a. knowledge base
b. model or so-called inference machine
c. user interface (Hand et al., 2001)

A knowledge base is usually consisting a set of all available information gathered in the strictest organizational way that is possible to achieve. This includes data-formatting and preprocessing in order to make it easier to be processed by any numerical analysis tools to be employed in the future. It is a very tedious and complicated task and in the same time is crucial to the future system accuracy.

The knowledge sources might be categorized into two main classes:

a. empirical results
b. theoretical background

If available, both sources might be combined in the benefit of the DSS. In pharmaceutical technology there is a lot of strong physicochemical background, which allows for describing pharmaceutical formulations in terms of their components properties. However, pharmaceutical formulations are very complicated structures, where many factors play, sometimes not very well defined, role in their behavior. Complexity of the pharmaceutical formulations, including their preparation technology, make them very difficult to classical analytical description. Hundreds of well defined physicochemical factors are becoming well defined description only, without practical meaning for prospective decision support. Regarding this it is noteworthy, that so far in pharmaceutical technology empirical knowledge plays still most important role in particular problem description. It is that's why in this field, when numerical analysis of the data is employed, empirical modeling becomes

the tool of the choice to create appropriate model (the inference machine). It allows to create the model based on the data only, without a priori assumptions and therefore without a need of a priori knowledge. The model is created based on the data only, which reflects current state of knowledge about the problem. With lack of the well established theories present, partially verified hypotheses or theories from different fields could be even misleading, therefore the model based on the data only has the advantage of lack of bias. Typical examples of empirical modeling tools are ANNs, which became very handy tools for empirical modeling implementation. Specifically, ANNs can work in two main modes:

a. predictive modeling
b. data mining

As it would be shown below, both modes are complementary to each other, which is another example of smooth and effective work of ANNs.

The user interface is a final part of DSS to be prepared and is strictly dependent on the particular problem specifics.

Complete algorithm of DSS preparation with emphasis on ANNs use could be described as follows:

1. Definition of the model function
2. Preparation of the knowledge database
    a. data acquisition
    b. data preprocessing
        - definition of input and output vector
        - scaling, normalization, noise addition, classes balancing
    c. splitting original dataset to two nonequal datasets according to k-fold cross-validation scheme
3. Construction of inference engine as ANN model
    a. ANN training and search for optimal (or suboptimal) architecture
    b. validation by k-fold cross-validation scheme
    c. sensitivity analysis and input vector reduction if applicable
    d. preparation of the higher order models – expert committees (ensembles)
4. User interface preparation

The above scheme depicts main steps to be performed in order to create DSS with use of ANNs. After preparatory phase including points 1 & 2, the modeling procedures have to be employed (p. 3). ANNs are used as tools to model relationships of interest in particular problem. This is usually done by creation of the predictive models designed to answer the question what would be the action of the new component introduction or modification of qualitative/qualitative composition. This would help to decide whether to use or not the composition tested in silico in the prospective laboratory experiments. The search for the most promising formulations-candidates could be realized in the most simplistic way as a combinatorial approach where there are set boundary conditions (i.e. the set of available excipients) and criteria of optimal formulation acceptance (Fig. 3). In case of the DSS total failure, i.e. all predictions were falsified by laboratory experiments, it is possible to enter interactive mode, (Fig. 3 dotted line) where the results of final (unsuccessful) laboratory experiments are added to the initial database and used for subsequent modeling procedure. Re-training of the neural models is usually much easier than the original step of optimal ANN model search, thus the interactive mode could be of choice when very little information is available at the beginning of the analysis.
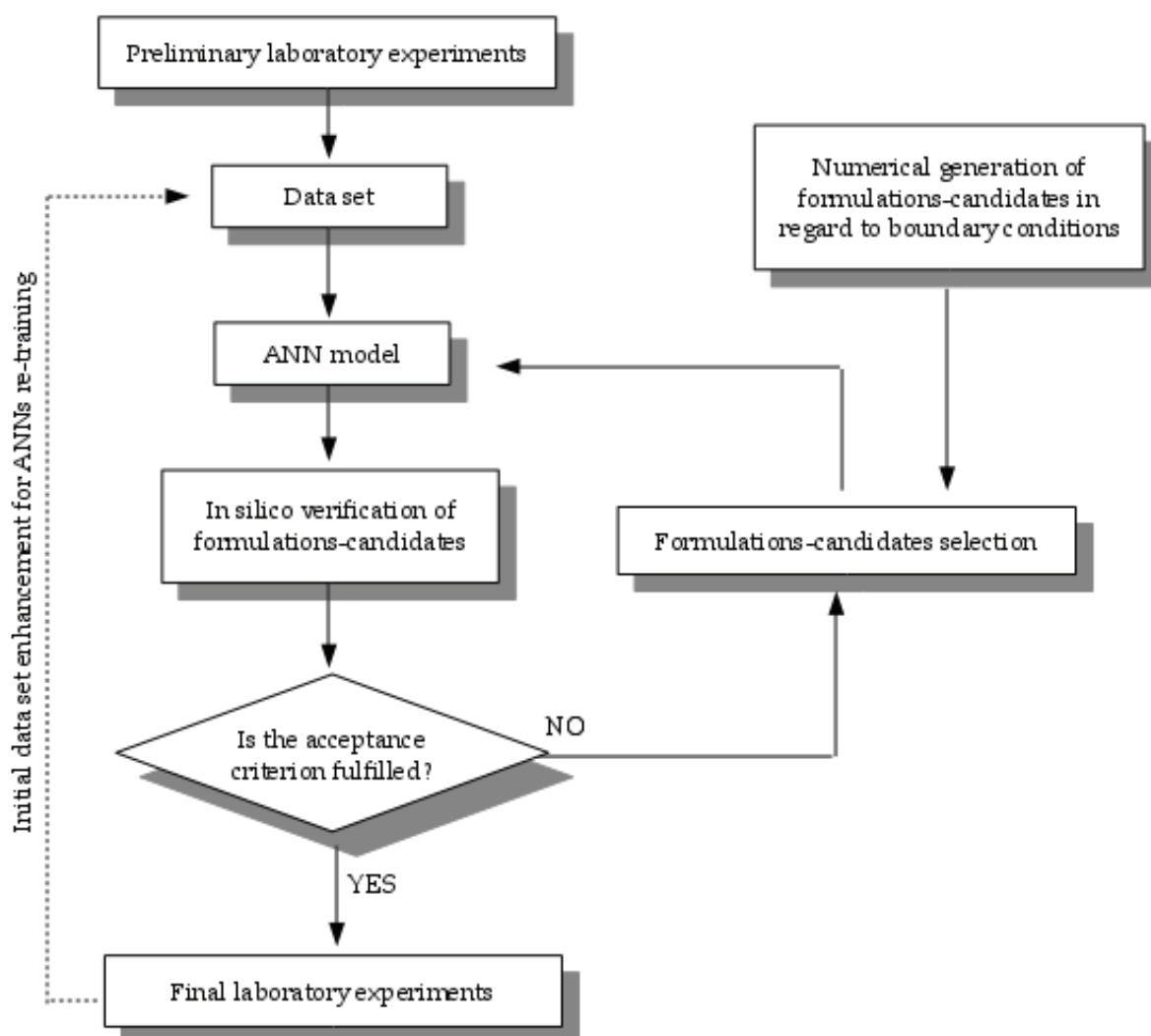
Fig. 3. The algorithm of ANN used as a tool for computed-aided formulation procedure

The use of ANNs in the predictive models function supports the decision based on the "black box" model. This means that no decision explanation and justification is available from the system. Such an approach is acceptable in the DSS, however it could be sometimes unsatisfactory for the user. Therefore, ANNs could be also used in the data mining function in order to provide an insight into the data and some means to formulate hypotheses about the analyzed problem.

ANNs unique features allow them to perform following operations in the data mining approach:
a. select crucial variables for the problem
b. extract logical rules (neuro-fuzzy systems)
c. provide response surfaces for a single input variable or their set

The latter is especially interesting as it allows to switch from "black box" modeling to classical statistical analysis when the problem dimensions reduction was carried out to the sufficient level (i.e. less than 10 input variables). Therefore, it could be created an ordinary mathematical equation quantifying analyzed relationship. Selection of the crucial variables and logical rules extraction form neuro-fuzzy systems are another ANNs powerful features,

which would be described further in this chapter. At this moment it is worthy to present only an interesting feature of ANNs employed as data mining tools. In order to obtain the most reliable results it is necessary to find the most competent ANN model. Since ANNs are empirical "black box" models, it is natural that their competence is assessed as the ability to solve unknown cases. This is nothing else but generalization error assessment, which is performed by predictive modeling. Based on the above statements it could be concluded, that data mining procedures include predictive modeling as well. This could be demonstrated by the analysis of the crucial variables iterative procedure with use of ANNs (Fig. 4).
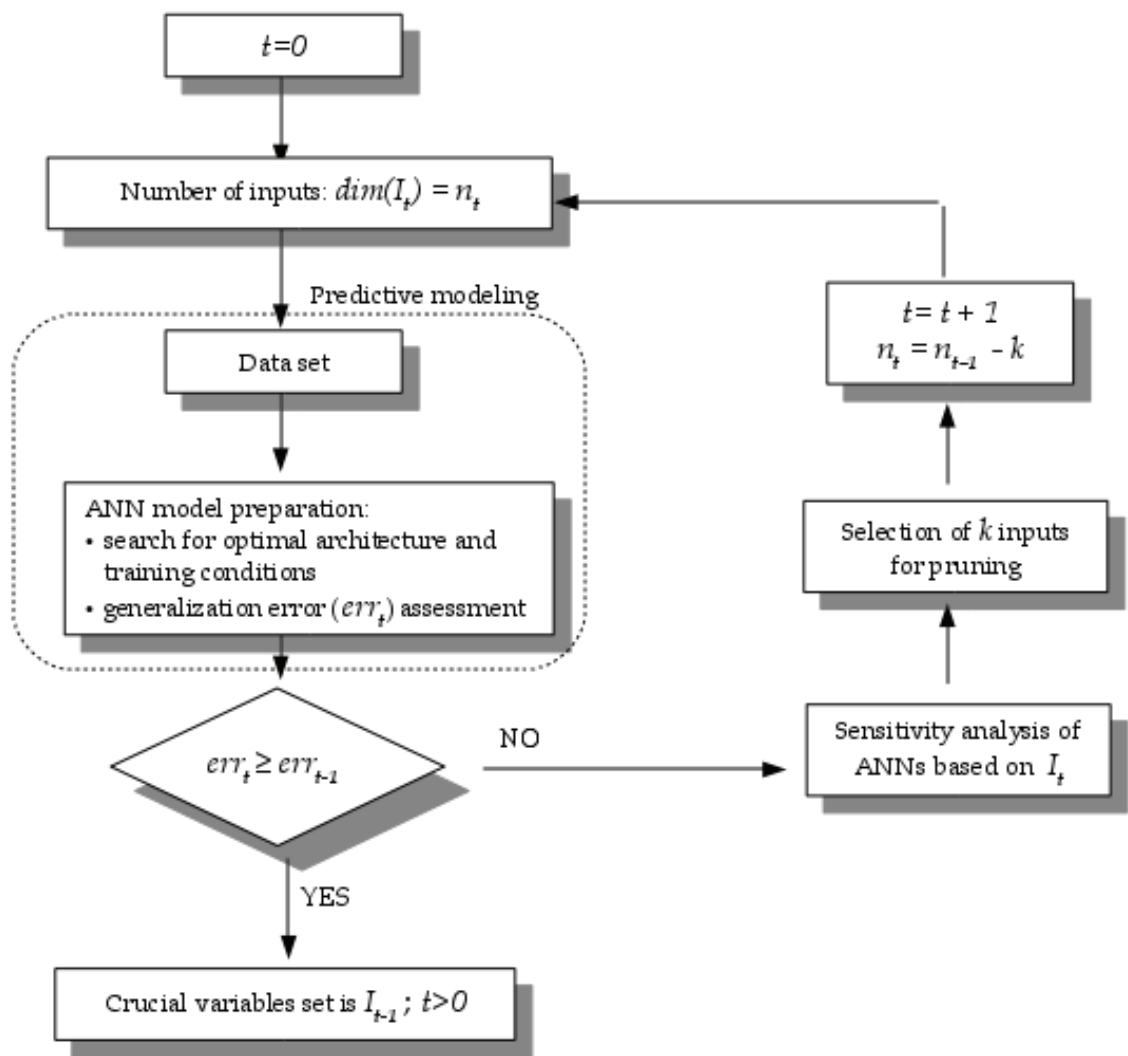

Fig. 4. The algorithm of inputs reduction with use of sensitivity analysis. t – time step; I – inputs vector; n – number of inputs; k – number of inputs for pruning; err – generalization error;

The algorithm presented in Fig. 4 allows the smallest number of input variables estimation with regard to the ANN model predictive competence. In other words, the final model is the most general of the best predictive models. This allows to decide, which variables are absolutely necessary to provide competent model, and which could be excluded without performance loss. This results in the very valuable information about the character of the analyzed problem and in the same time an inference machine for DSS is provided.

4. Predictive modeling

Predictive modeling is focused on the generalization abilities of the system, which is usually commonly understood as the extrapolation beyond available database. It is the most difficult task to be performed during the DSS construction.

4.1 Data preparation and preprocessing

Since ANNs are numerical analysis tools they require numerical representation of the whole data available for the problem. This statement is not as trivial as it seems, when the real life data, i.e. pharmaceutical technology, are at the focus. It's challenging to develop numerical representation of pharmaceutical formulation qualitative composition or its preparation technology. So far there is no universal solution of this problem, therefore several methods are used to deal with this task. Among them two main groups of numerical representations could be named:

a.   topological
b.   physical

In the topological representation input vector is usually binary and the presence of particular formulation compound is denoted by position of its non-zero element. The same could be adapted for formulation technology or other abstract information. The advantage of this approach is its simplicity. One of the disadvantages is a large number of inputs causing problems with high dimensionality of created model. Even if ANNs are working relatively well with multidimensional problems, it should be avoided if possible. More serious drawback of topological encoding is its lack of physical meaning as it is used as completely abstract and subjective design (Fig. 5). Therefore, it could be possible that by use of different encoding scheme (i.e. shifted arbitrary positions of particular components), there would be achieved different modeling results.
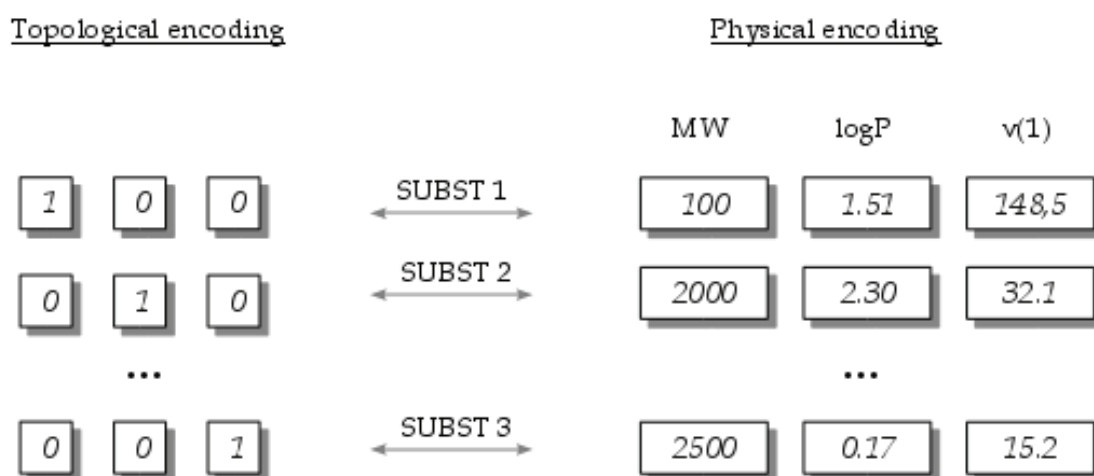

Fig. 5. A comparison between topological and physical representation of pharmaceutical formulations. SUBST – chemical substance, MW – molecular weight, logP – water/oil partition coefficient, v(1) – connectivity index.

The most important disadvantage is that ANN model is restricted only to the established set of substances available at the beginning of the modeling procedure, therefore it has no

generalization abilities in terms of qualitative composition. Of course it could be possible to add some additional "dummy" inputs for unknown substances, however regarding previous remarks about arbitrary design of inputs topology without physical meaning, it could be achieved only prediction for some "unknown" substance but not for a specified, particular structure. This is the main reason why topological encoding is treated as the last resort. In contrast, physical encoding has no such drawbacks. It is based on available characteristics of particular excipient (i.e. molecular weight, melting point) or technological process (i.e. compression force). It looks straightforward and perfect approach. Unfortunately, there is one but major drawback of physical encoding – availability of ready-to-use information. Various manufacturers provide different sets of features of their products. Moreover, various substances cannot be characterized in the same manner due to their native character as i.e. being in solid or liquid state. Unification of substances description is required when ANN model has to be built on all available examples. The more data examples, the more competent is the model, thus it is advisable to include every information describing analyzed problem. This is however contradictory with above described problems with unified knowledge representation of the chemical substances. An effective solution could be application of chemical informatics tools, which generally are computer programs able to compute chemical substances properties (so-called molecular descriptors) based on their molecular structure. Chemical informatics has long history and many different applications (Agrafiotis et al., 2007). It is beyond the scope of this chapter to provide complete description of this vast discipline. In pharmaceutical applications, cheminformatics is mostly known at the very early stage of active pharmaceutical ingredient (API) search regarding its desired pharmacological activity. QSAR methods are now routinely applied as tools reducing laboratory experiments number in order to find new promising API, which could become valuable drug in the future. Prediction of toxicological properties of drugs is also at the scope. Cheminformatics is not so popular yet in pharmaceutical technology, however currently it is drawing more attention due to its advantages:

a. unified description of all substances
b. vast number of molecular descriptors counted in thousands
c. prediction of real physical properties (i.e. logP, logD, pKa, etc)

There are disadvantages of cheminformatics use as well:

a. requirements of high computational power for *ab initio* modeling
b. accuracy of physical parameters prediction
c. restrictions of maximum atoms numbers in the analyzed molecule

Unified numerical description of substances is the result of algorithms, on which cheminformatics software is based, thus all molecules are processed in the same reproducible manner. This is crucial for maintaining methodology of ANN model preparation. The large number of molecular descriptors available allows to choose the most representative ones for analyzed problem, which is the most important in data mining procedures, but improves predictability of the model as well. Moreover, in predictive modeling molecular descriptors could be treated as a numerical representation of the molecule without the need of complete understanding of their physical meaning. In fact many of the molecular descriptors are nothing else like numerical representation of 2D (sometimes 3D) structure of analyzed molecule with regard to number of atoms, its geometry, topology and other constitutional features involved. Since the procedure of computations is algorithmic, it allows to use molecular descriptors empirically, based on the

ANN selection of what is the most suitable to achieve maximum predictability of the model. Combining this approach with large number of molecular descriptors available, results in the powerful tool for creating numerical representation of pharmaceutical formulations. Specifically, in predictive modeling the accuracy of physical parameters prediction by cheminformatics software is not an issue as long as ANN model is used as a "black box" in the DSS and the same software is used to encode all substances in the database. The cheminformatics software will be commented in the next section of this chapter.

Overcoming all the problems with pharmaceutical formulation encoding results in the database or so-called "knowledge base" – a source of knowledge for ANN model. In order to be used effectively, the database must be preprocessed. First and obligatory preprocessing procedure is scaling according to the ANNs activation functions domains. Usually the scaling is performed in range (-1;1) but other ranges are also applied, like i.e. (0;1). The latter is sometimes realized as normalization procedure, however more frequently linear scaling is carried out.

4.2 ANNs training

ANNs need to be trained on the data in order to create competent model. Training of ANNs is a serious task and it is impossible to cover all aspects of this issue in this chapter. Following there will be described only the issues, which in authors' opinnon are the most relevant to the neural modeling for DSS. Generally, training of ANNs requires several issues to be solved:

a. software and hardware environment
b. training algorithm and scheme
c. topology of ANN (architecture)
d. error measure and model accuracy criterion

Since for the software and hardware environment there will be dedicated further section of this chapter, it is only worthy noting in this place that there is plenty of software available either as free of charge or as commercial packages. The next issue is the subject of many research ongoing, as the universal and perfect ANNs training algorithm does not exist. This is confusing especially when the ANNs simulator provides many algorithms of the choice. Regarding applications of ANNs in pharmacy, the most common and robust ANNs training algorithms could be named as follows:

a. backpropagation with modifications
b. conjugated gradient and scaled conjugated gradient
c. Kalman filter and its extensions
d. genetic algorithms and particle swarm optimization

The above chosen algorithms are mostly associated with so-called supervised learning, where the knowledge base consists of known outputs associated with the inputs. This type of learning is the most suitable for building ANN-based DSS in pharmaceutical technology. Authors are using software with backpropagation (BP) learning algorithm including *momentum*, *delta-bar-delta* and *jog-of-weights* modifications. Backpropagation is a very old and therefore well-established algorithm, which is relatively slow-converging comparing to the newest ones, however is very robust and versatile: i.e. it is suitable for neuro-fuzzy systems as well. The above and BP mathematical simplicity makes it a good choice for implementation in DSS preparation with ANNs. BP with momentum modification has two parameters (learning rate and momentum coefficient), which are chosen arbitrary by user. However, *delta-bar-delta* and *extended delta-bar-delta* modifications allow ANN to modify

these parameters during the training process – this improves learning dramatically. *Jog-of-weights* technique is a stochastic search of optimal solution, which is carried-out by simple addition of noise to the ANN weights values when no more training improvement is found during previously set number of iterations. Setting the architecture of ANN is another difficult task, which affects the model performance. Unfortunately, there is no algorithmic solution here. It is usually realized by trial and error experiments carried-out with large number of architectures-candidates in order to select the best one for particular problem. Some improvement is promised by use of hybrid ANNs systems with genetic algorithms (GA). In this evolutionary approach GA is responsible for ANNs architecture adjustment and ANN itself is trained by BP. However, there are still contradictory opinions about suitability of such hybrid systems. In order to decide, which architecture is the most suitable for becoming the core of DSS, it is necessary to apply some quality criterion. Predictive performance is in this case the most applicable criterion expressed as generalization error. The most commonly known method to measure ANNs generalization is k-fold *cross-validation*, where "k" is integer number in the range (0; ∞). The procedure is designed to assess generalization error on the whole available data set. The latter is divided into the two non-equal data-sets: the larger one as training data set and the smaller one as validation (test) set. The ANN is trained on the larger data set and after the training phase the validation set is presented – the error encountered on this set is the generalization error. After that, the validation set is returned to the training set and the new pair of training-validation sets is created, however no previously chosen validation data is included in the new validation set. Again, the ANN is trained on the training set and validated on the smaller one. This algorithm is repeated with respect to the "k" value. The most common "k" value is 10 and each time 10% of original database is excluded from the database to become validation set. After 10 iterations for each architecture the generalization error is assessed for the whole original database (10 x 10% = 100%). Although computationally expensive, this procedure is a standard when the database is small, which is almost an omnipresent situation in real-life examples. A modification of this procedure is *leave-one-out*, where "k" value is equal to the data records number, thus in the validation set there is always only one data record. This is even more computationally expensive, yet from the statistical point of view it provides the most unbiased estimation of ANNs generalization abilities. There are several error measures applicable to express the generalization error of ANNs. Among them, dependig on the analyzed problem type, the most commonly applied are:

a.   linear correlation coefficient (R) of predicted vs. observed values
b.   *mean squared error* (MSE) or *root mean squared error* (RMSE)
c.   classification rate or other classification measures (specificity, sensitivity, etc.)
d.   problem-specific measures, i.e.: *similarity factor* (f2) for drug dissolution tests (FDA, 2000)

Each of the error measures allows generalization error quantification, yet it is not absolute – there is no modeling success criteria available. This means that no error measure allows to prove mathematically, that on its specific level the model is competent and reliable. This situation is not only the domain of ANNs. There are present some rules of thumb that beyond some borderline value the model is acceptable. An example of such rule is correlation coefficient where the value over 0.95 is usually acceptable as the indication of good linear correlation between variables, however some authors are more restrictive and demand the value to be over 0.99. Therefore, every generalization error estimation should be regarded with care and related to the problem analyzed.

After the search phase of ANNs best architecture there is provided the ranking of ANNs generalization abilities. The best architecture of ANN is chosen as the final DSS inference machine. However, to improve performance of the model there are built so-called ensemble ANNs consisting of several neural models, which outputs are combined to provide final system output (Maqsood, 2004). The outputs combination is the key factor of ensemble performance. There are many methods for outputs combination, namely:

a. simple average
b. weighted average
c. non-linear regression
d. ANN of second order

The latter method with second order ANN is used very rarely due to the computational burden, yet seems very interesting as the method of non-linear estimation of each ensemble element influence on the final output of the system.

4.3 Modeling example

Preparation of ANN model for DSS in pharmaceutical technology could be illustrated by the example of neural modeling for optimization of so-called solid dispersions systems. Solid dispersions are usually defined as systems consisting of a poorly soluble drug and at least one carrier characterized by good water solubility. The purpose to formulate solid dispersions is to increase water solubility of poorly soluble drugs and in consequence to improve drugs pharmaceutical and biological availability. Unfortunately, there is no clear theory how to adjust quantitative and qualitative compositions of solid dispersions in order to achieve drug solubility enhancement. This could be the domain to DSS – to help in the right choice of the carrier and drug/carrier ratio in order to improve particular drug solubility in water. The neural model was constructed to predict dissolution profile of various drugs, in regard to the solid dispersion (SD) quantitative and qualitative composition as well as SD preparation technology. There were 17 inputs and one output of ANN. The inputs encoded following parameters in physical encoding system:

a. SDs' compositions
b. dissolution test conditions

There was also abstract classification of the methods of SDs preparation added to the input vector as well as the single input expressing the time-point after which the amount of dissolved drug was to be predicted by ANN and presented at the single output. The number of data records was around 3000. Totally, there were around 6 000 ANNs trained and tested in this experiment. The best ANN architecture derived generalization error RMSE = 14,2 vs. maximum output value 100. It was complex ANN with 4 hidden layers and hyperbolic tangent activation function. By introduction of ANNs ensemble with 10 ANNs included and simple average of their outputs, it was possible to achieve generalization error RMSE = 13.4.

The whole neural system was tested as DSS on the following possible scenario: what would be optimal ratio of papaverine (spasmolytic drug) and Macrogol (water-soluble polymer) in SD in order to achieve designated papaverine dissolution profile? This is a typical task to solve in pharmaceutical technology, where the formulation is a tool for modification of the drug course of action. The data were derived from publications, therefore the papaverine's dissolution profiles from various SDs were known and presented to DSS as a task to solve. The above mentioned data was of course unknown to ANNs, which means that the data was not included in the training data set. The system was working according to the
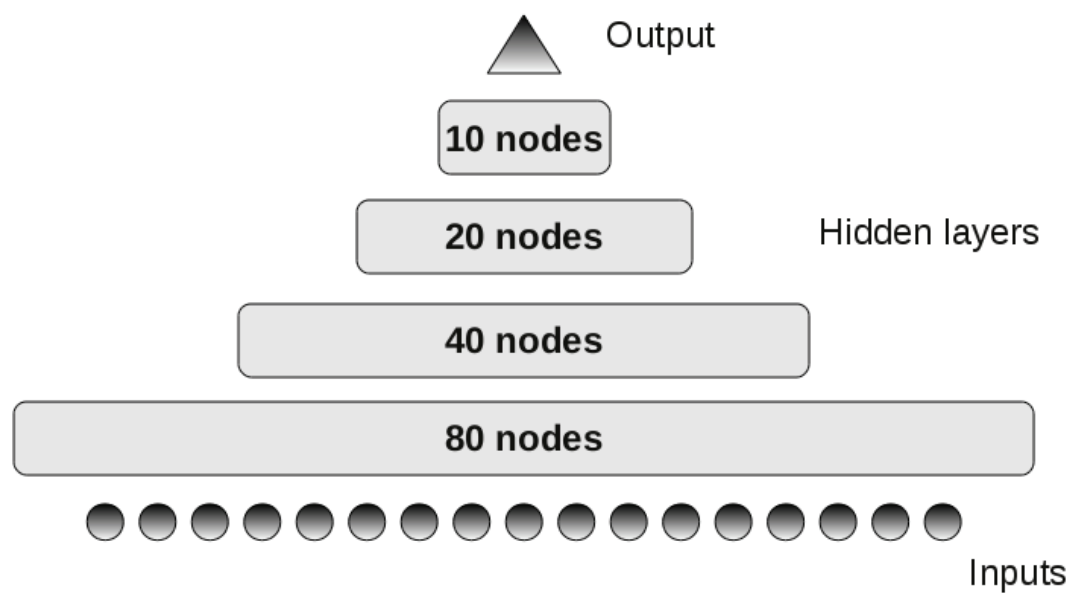
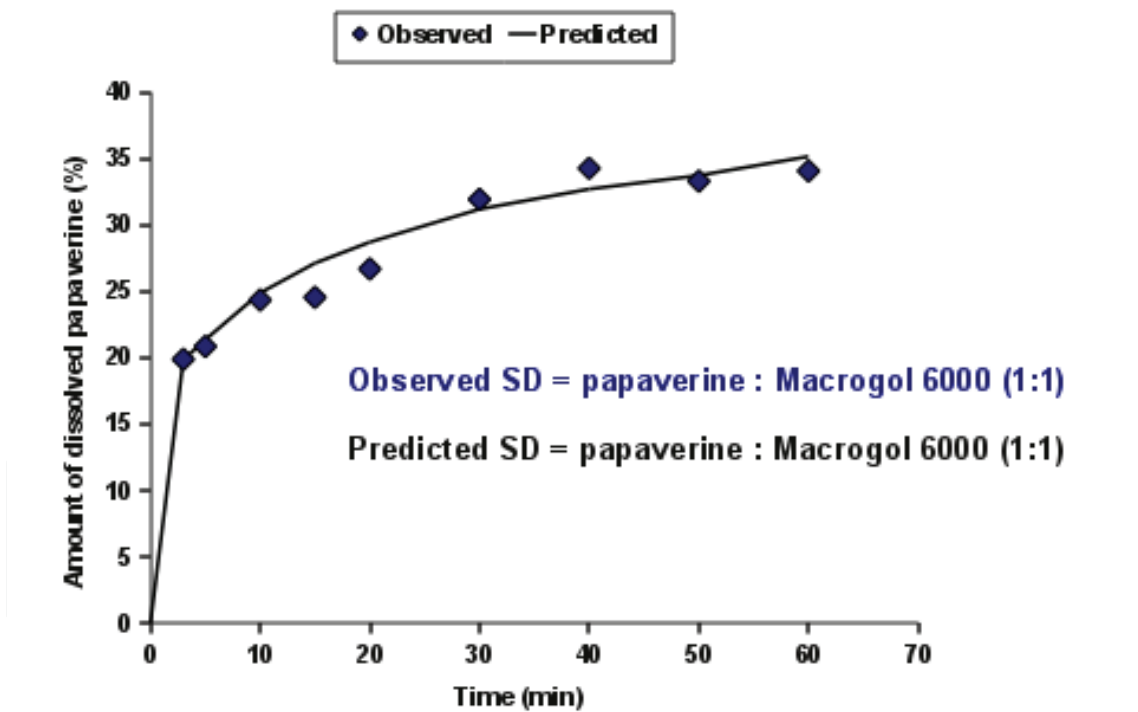Fig. 6. Best ANN architecture for prediction of drugs dissolution from SDs.

Fig. 7. Appropriate prediction of SD papaverine : Macrogol 6000 1:1 ratio. Prediction error RMSE = 1.3.

algorithm described previously (Fig. 3) wit boundaries selected for qualitative and quantitative composition. Iterative procedure based on the presentation of around 2 000 formulation-candidates with papaverine dissolution profiles as the acceptance criterion. There were 8 profiles presented to the system. As a result in 6 cases qualitative and quantitative compositions of SDs were predicted by the system accurately (Fig. 7). This

meant that DSS recommended the same SD composition to achieve particular drug dissolution profile, which was in fact a true source of this profile described in the publication. In conclusion, it was confirmed that DSS based on the ANN could be competent and useful in assisting in the pharmaceutical formulation optimization according to the specified criteria.

5. Data mining

Data mining is a process of knowledge extraction from the database usually associated with discovery of hidden patterns in the data (Wikipedia, 2009b). Empirical modeling with ANNs is one of the standard tools applied in the data mining.

5.1 Sensitivity analysis

Sensitivity analysis is regarded as one of the data mining tools. As a result of this procedure the ranking of relative importance of inputs over the output is provided. It allows to select crucial variables set (Fig. 4). Detailed review of crucial variables characteristics leads to the deeper insight into the analyzed problem. The ranking created by ANNs is the result of observation of data made by machine learning system of empirical modeling. It is quite common, that machine observes data in a different manner than human, and thus the results of such observations are also different. That is exactly what is expected from ANNs at this moment – the unbiased observation of the data conceiving the results, which might be sometimes even contradictory with so-called "common knowledge". These contradictions, or at least unexpected outcomes, are supposed to direct researchers' reasoning to other paths, which could be successful in preparation of the optimal pharmaceutical formulation, when conventional approach fails.

There are many methods of a sensitivity analysis, but two of them are worth mentioning here, since they are commonly used for ANNs. First method is based on the simple assumption that inputs importance could be measured by ANN prediction error changes when particular input is excluded from ANN. The procedure is usually carried out by setting value of input of interest to "0" and assessment of prediction error on the data test set. The bigger error increase, the more important is the selected input. An advantage of this method is its simplicity and versatility – it could be used to every modeling system, not only ANNs. However, this method has some major drawbacks. The most important is that the outcome depends on the data test set used. This makes the procedure difficult to be reproducible. Another issue is the fact that sometimes the "0" value of the variable denotes some information to the system, therefore it creates confusion when all values of particular variable are set to "0". Last but not least is the fact that this method works on the ANN model in its non-natural state, when one of the inputs is in fact nonfunctional. The error increase is the reflection of how badly ANN was destructed by pruning one input. The criticism here is also augmented by unidimensional type of analysis performed. In contrast, second method is much more complicated mathematically but in the same time more sophisticated. Żurada (Żurada et al., 1997) developed method for pruning redundant features based on the analysis of derivative of outputs over ANN inputs (Eq.1).

$$S_{ki} = \frac{\delta y_k}{\delta x_i} \tag{1}$$

where:

S_{ki} – sensitivity of k-th output over i-th input

y – output

x – input

k/i – output/input indexes

The derivatives are computed according to the chain rule through the whole ANN for every training pattern. It results in the matrix, which after additional processing provides ranking of inputs. This procedure is reproducible as it works on the training dataset by default. ANN is not altered in any way – it is processed after the training phase in its natural, the most competent state. There is also one drawback of this method – so far it has been developed for MLP ANNs only.

In order to decide, which inputs to prune there must be applied some criterion of how to find a cut-off point in the inputs ranking. Unfortunately, regardless of the method used for ranking creation, there is no universal method of decision where would be the borderline. Usually, the cut-off point is chosen at the largest difference between sensitivity values of adjacent variables in the ranking – this is the borderline between pruned and remaining variables (Fig. 8).
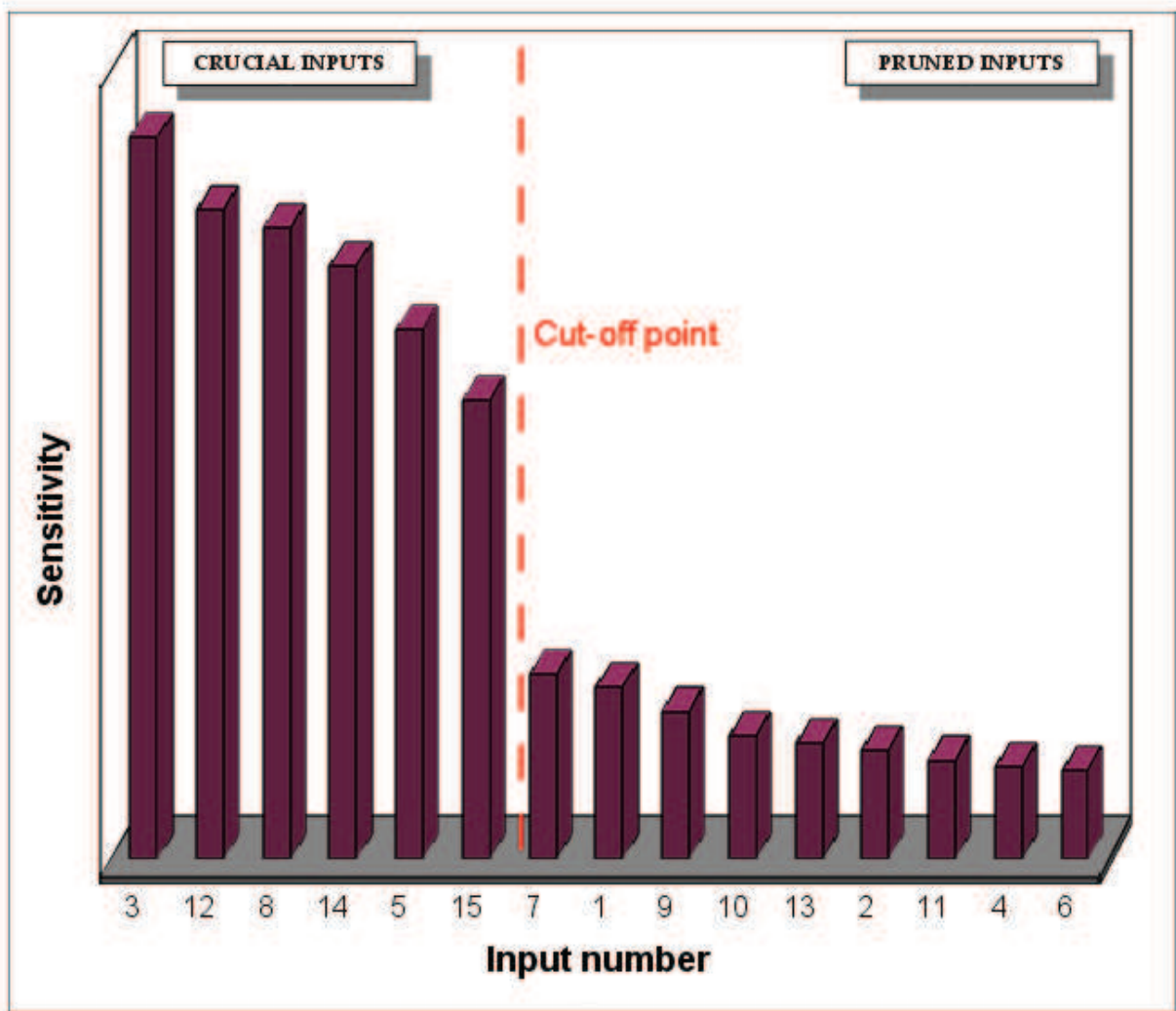

Fig. 8. Sensitivity analysis example with cut-off point selection.

5.2 Fuzzy logic and neuro-fuzzy systems

Fuzzy logic was defined in 1965 when Lotfi Zadeh proposed theory of fuzzy sets. In summary, fuzzy reasoning is based on the probabilistic approach, where every value could be expressed as probability of being a member of some values sets. This is another type of commonly known reasoning based on the classical, crisp numbers. In the simple example a value 0.1 could be a member of set "0" but in the same time be a member of set "1". Probabilities of the memberships to particular sets are designated by so-called membership functions.

Fuzzy reasoning could be encoded in rules tables (Eq. 2).

$$\text{IF } a = A \text{ AND } b = B \text{ AND} \dots z = Z \text{ THEN } y = Y \tag{2}$$

The above example of simple logical rule could be extended in terms of number of variables and rules as well. Moreover, fuzzy reasoning allows to introduce so-called linguistic variables produced by human experts as non-numerical description of their professional experience expressed in qualitative terms like: "high", "low", "moderate", etc. However, for the improvement of DSS construction it is important to mention hybrid neuro-fuzzy systems: ANNs coupled with fuzzy logic. The neuro-fuzzy system exploits both approaches advantages, namely fuzzy rule-based problem description with self-learning empirical modeling abilities of ANNs. This creates powerful data analysis tool, which is able to observe presented data and to provide self-generated logical rules (Mansa et al. 2008). The latter could be easy decoded to the human-readable form like presented in Eq. 2. In the simplest Mamdani model (Yager & Filev, 1994) neuro-fuzzy system consists of only one hidden layer with specially augmented nodes representing "IF" part of the logical rule. Thus, the number of nodes determines the number of rules – their adjustment might be
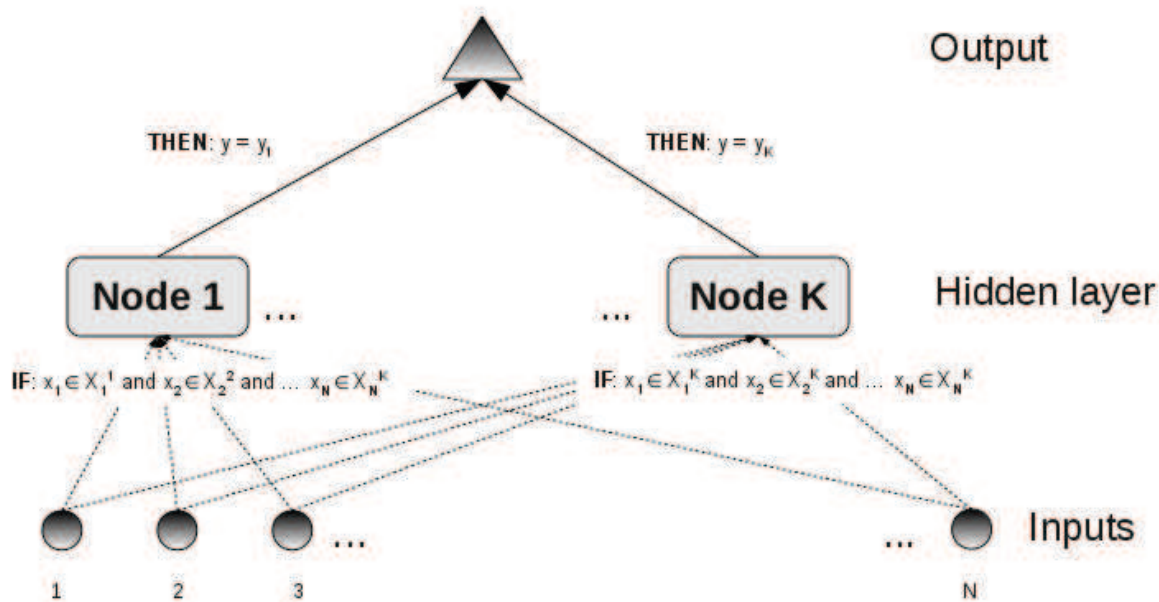

Fig. 9. A simplified scheme of neuro-fuzzy system of Mamdani multiple input single output (MISO) type; x – input, y – output, N – number of inputs, K – number of hidden units, capital letters – membership functions, small letters – crisp numbers.

made manually or automatically by specific algorithms. The outcome of the rule (THEN) is encoded in the synaptic weight connecting particular hidden node with the output node. The whole system could be trained with classical, well-established BP algorithm.

As for every tool, there are also drawbacks of the neuro-fuzzy systems. They are not so versatile like MLP ANNs. This means that not all the problems could be covered by neuro-fuzzy systems, since in fact they are classification-based tools. Their approximating abilities are far below MLP ANNs. In personal experience of authors, neuro-fuzzy systems provide sometimes contradictory or "dummy" logical rules, which from the professional, pharmaceutical point of view are useless and have to be reviewed with utmost care and criticism. In complex problems, like i.e. in pharmaceutical technology, the number of hidden nodes tends to become large, thus making logical rules harder for direct human interpretation. All the above criticism refers to the simplest Mamdani neuro-fuzzy systems. Perhaps the use of Takagi-Sugeno models or more sophisticated architectures optimization algorithms would solve abovementioned problems. This would be the task for the future research. The last, empirical remark about neuro-fuzzy systems would be in favor of their use as members of ensemble ANNs. It was observed several times that when neuro-fuzzy system was added, it improved ensemble performance significantly. This was found even when neuro-fuzzy system was far less competent than several MLPs in the ranking of ANNs generalization abilities. A working hypothesis is that coupling MLP with neuro-fuzzy system allows to exploit both tools different approaches for data analysis. However, for now it is too early research phase to conclude this hypothesis.

5.1 Modeling example

An example of successful sensitivity analysis would be the research about possible mechanisms of drugs release from solid dosage forms. The objective of this study was to identify the mechanisms of model drugs release from hydrodynamically balanced systems (HBS). HBS are prepared in a form of capsule filled with drug substance and mixture of polymers.

Ketoprofen (KT), a poorly soluble non-steroidal anti-inflammatory drug was chosen as a model active substance. Several polymers were used as matrices alone or in binary mixtures: cellulose derivatives (hypromelose), carrageens and alginates. ANNs models were constructed to predict drug release profile from HBS formulations based on their quantitative and qualitative composition. For qualitative composition encoding cheminformatics software was used in order to provide appropriate numerical representation. An initial number of input variables was around 2700. It was the result of cheminformatics encoding of HBS matrices. Data mining methodology was based on the crucial variables set analysis. Search for crucial variables set was performed according to the algorithm depicted in Fig. 4. However, classical sensitivity analysis method was altered due to difficulties with finding significant differences in the ranking of input variables, which made difficult to establish cut-off point. The altered procedure was "context-based" search for the minimum number of variables within original ranking of variables provided by sensitivity analysis. The final choice of variables was performed according to the information about chemical descriptors class, where only one representative of each class was chosen as crucial variable. Numerical experiments with comparison of generalization error between models based on the original and altered variables choice procedure confirmed that application of context based search is beneficial to the model performance

(Fig. 10). In result, it was possible to achieve substantial reduction from 2700 to 8 inputs finally. Final ANNs model confirmed its performance with generalization RMSE = 5.93. The successful generalization examples for unknown formulations were found (Fig. 10). Analysis of 8 inputs meaning allowed to formulate hypothesis about importance of the polymer geometry to the drug release profile.
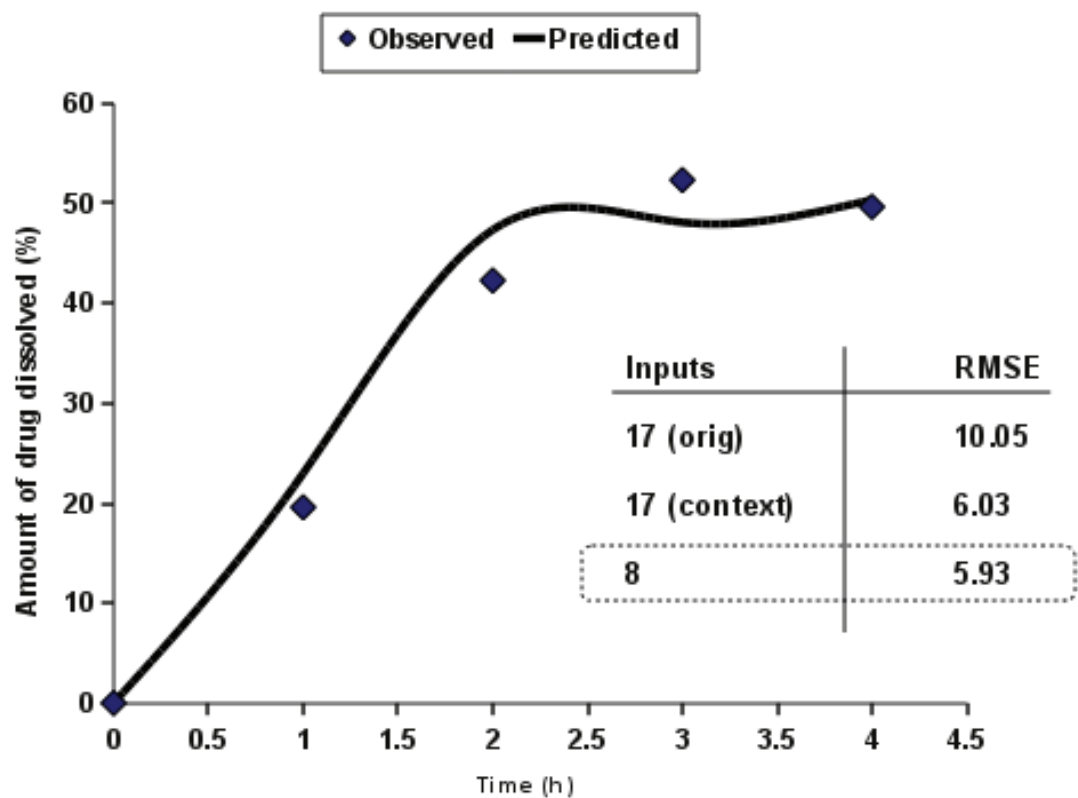

Fig. 10. Graph: results of prediction of HBS formulation with carrageen. A comparison between various ANNs with inputs selected by original sensitivity analysis (orig) and altered procedure (context).

6. Software and hardware requirements for DSS with ANNs

6.1 Software
Software environment is crucial for every IT project development. Apart from data processing software like spreadsheets and word processors for documentation preparation, the most important software for DSS preparation with ANNs is ANNs simulator. The term "simulator" is used because there are specialized hardware realizations of ANNs available even as PCI extension cards for PC computers, not mentioning specialized neurocomputers. Hardware ANNs have one advantage over software simulators: they perform parallel computations exploiting this ANNs feature. However, these specialized solutions are very expensive and regarding fast increase of computational power of PC computers, the use of software ANNs simulators seems to be justified. During last 20 years ANNs became so popular that to name all ANNs resources available is impossible for now. Therefore, let us present some examples based on the authors' experience with this type of software. There are several well established commercial packages available:

- NeuralWorks - Professional II/PLUS
- Matlab Neural Networks Toolbox
- statistics software: SPSS, Statistica
- NeuroSolutions

There is also a lot of free software for Windows and Linux/Unix/ MacOS:

- Stuttgart Neural Network Simulator (SNNS)
- GENESIS
- Emergent (former PDP++)
- WEKA

An important issue, when the software choice is to be decided, would be the work mode. If it would be only for data mining, then usually less computational power is required than for the predictive modeling. However, when strictly following previously described algorithm of inputs reduction (Fig. 4) then computational power requirements are high. It was roughly estimated before, that predictive modeling requires usually thousands of ANNs to be trained and tested in order to find the most optimal solution. The task of ANNs training is computationally expensive, therefore it is realized with use of distributed computing on so-called "grids" or "server farms", where several computers are working simultaneously and processing different ANNs. It is the simplest parallelization system, which is in the same time very effective when using ANNs. However, it requires as many licenses of the software as there will be the number of parallel processes running out simultaneously. Regarding the commercial packages, it becomes very expensive to buy separate licenses for each of running processes. Moreover, most of the commercial software is dedicated to MS Windows environment. The simulators are usually standalone packages with point-and-click GUI, without batch mode option. On the contrary, free software is at no cost with as many running instances as it is needed. Many of free packages are built for console mode, thus the batch processing mode is the default option. This is especially characteristic for Open Source software released under various versions of GPL (Gnu Public License). Authors are working with in-house ANN simulator written in Pascal and compiled with use of FreePascal and Lazarus. All computers are working under control of various Linux distributions and there is also developed in-house software for automatic control and distribution of computational tasks. In conclusion, it is worthy to consider Open Source software solutions and Linux environment for ANNs models preparation for DSS, because of a good cost-effectiveness ratio, availability of software and its stability.

Apart from ANNs simulator, cheminformatics software was mentioned as an important element of DSS preparation for pharmaceutical technology. It is a very similar situation in this field like in ANNs – there is plenty of the software available with even more Open Source or Free Software present (Linux4chemistry).

Commercial packages:

- Gaussian
- Gamess-UK
- Sybyl
- Dragon
- Molecular Modeling Pro

Open Source/Free packages:

- Gamess-US
- MarvinBeans (free for academia and non-profit activities)

- RDKit
- ABINIT
- AMMP
- Gromacs
- MOPAC

There is even a special Linux Live CD distribution dedicated to cheminformatics: Vigyaan.

6.2 Hardware

ANNs foundations were noted early 50's of the last century. After some disappointment in their abilities they were forgotten for some time, but 80's was the time of ANNs renaissance. It happened partially because of rapid growth of the computational power of PCs. Internet revolution and development of distributed computing was another factor of increasing interest in the neural modeling. Today, CPUs manufacturers developed new strategy of computational power increase and provide multicore CPUs for desktop computers. It allows for real multi-tasking in the work of modern computers. In order to build the mini-grid, all the infrastructure needed is a set of workstations, some LAN cables and switches. Coupled with Open Source software it provides low-cost, effective tool for ANNs development. There is no means to estimate minimum number of the workstations required. Regarding ANNs, an obvious truth is that the more computers available, the better. A very subjective estimation would be that a good start for the hardware environment is 10 workstations, each one based on 4-core CPU. The system is scalable. An enhancement of such structure with new workstations, even of different type, is very easy and does not generate additional costs beyond hardware price, assuming Open Source software use. In conclusion, building ANNs-based DSS is much easier and cheaper now, when there are present such interesting trends in the PC computers development.

7. References

Agrafiotis D.A. Bandyopadhyay D., Wegner J, & van Vlijmen H., (2007) Recent Advances in Chemoinformatics, *J. Chem. Inf. Model.*, Vol. 47, No 4, pp 1279–1293, ISSN: 1549-9596

Behzadia S.S., Prakasvudhisarnb C., Klockerc J., Wolschannc P. & Viernsteina H., (2009) Comparison between two types of Artificial Neural Networks used for validation of pharmaceutical processes, *Powder Technology*, Vol 195, No 2, 150-157, ISSN: 0032-5910.

Bourquin J., Shmidli H., van Hoogevest P. & Leuenberger H. (1998 a) Comparison of artificial neural networks (ANN) with classical modeling techniques using different experimental designs and data from a galenical study on a solid dosage form, *Eur. J. Pharm.Sci.*, 1998, Vol. 6, No 4, 287-300, ISSN: 0928-0987.

Bourquin J., Shmidli H., van Hoogevest P. & Leuenberger H. (1998 b) Advantages of Artificial Neural Networks (ANNs) as alternative modeling technique for data sets showing non-linear relationship using data from a galenical study on a solid dosage form. *Eur. J. Pharm. Sci.*, Vol. 7, No 1, 5-16, ISSN: 0928-0987.

Bourquin J., Shmidli H., van Hoogevest P. & Leuenberger H. (1998 c) Pitfalls of artificial neural networks (ANN) modeling technique for data sets containing outlier measurements using a study on mixture properties of a direct compressed dosage form. *Eur. J. Pharm.Sci.*, Vol. 7, No 1, 17-28, ISSN: 0928-0987.

Brier M. E. & Aronoff G. R. (1996), Application of artificial neural networks to clinical pharmacology. *Int. Jour. Clin. Pharm. Ther.*, Vol. 34, No 510-514, ISSN: 0174-4879.

Brier M. E. & Smith B. P. (1996), Statistical Approach to Neural Network Model Building for Gentamycin Peak Predictions., *J. Pharm. Sci.*, Vol. 85, No 1, 65-69, ISSN: 0022-3549.

Brier M. E. & Żurada J. M. (1995), Neural Network Predicted Peak and Trough Gentamicin Concentrations. *Pharm. Res.*, Vol. 12, No 3, 406-412, ISSN: 0724-8741.

Chen Y., McCall T.W., Baichwal A.R. & Meyer M.C. (1999), The application of an artificial neural network and pharmacokinetic simulations in the design of controlled-release dosage form., *J Contr. Release*, Vol. 59, No 1, 33-41, ISSN: 0168-3659.

Chow H-H., Tolle K.M., Roe D.J., Elsberry V. & Chen H. (1997), Application of Neural Networks to Population Pharmacokinetic Data Analysis. *J. Pharm. Sci.*, Vol. 86, No 7, 840-845, ISSN: 0022-3549.

Dowell J., Hussain A., Devane J. & Young D. (1999), Artificial Neural Networks Applied to the In Vitro - In Vivo Correlation of an Extended-Release Formulation: Initial Trials and Experience. *J. Pharm. Sci.*, Vol. 88, No 1, 154-160, ISSN: 0022-3549.

FDA (2000), *Guidance for industry. Waiver of In Vivo Bioavailability and Bioequivalence Studies for Immediate-Release Solid Oral Dosage Forms Based on Biopharmaceutics Classification System*, U.S. Department of Health and Human Services, Food and Drug Administration, Center for Drug Evaluation and Research (CDER), USA.

Gašperlin, M., Tušar, L., Tušar, M., Šmid-Korbar, J., Zupan, J. & Kristl, J. (2000) Viscosity prediction of lipophilic semisolid emulsion systems by neural network modeling. *Int. J. Pharm.*, Vol. 196, No 1, 37-50, ISSN: 0378-5173.

Gobburu V.S. & Chen E.P. (1996), Artificial Neural Networks As a Novel Approach to Integrated Pharmacokinetic - Pharmacodynamic Analysis. *J. Pharm. Sci.* Vol. 85, No 5, 505-510, ISSN: 0022-3549.

Hand D., Mannila H. & Smyth P. (2001), Principles of Data Mining, MIT Press, ISBN: 0-262-08290-X, USA.

Hertz J.; Krogh A. & Palmer R., (1991). *Introduction to the Theory of Neural Computation,* Addison-Wesley, ISBN-10: 0201515601, USA.

Hussain A.S., Yu X. & Johnson R.D. (1991) Application of Neural Computing in Pharmaceutical Product Development. *Pharm. Res.*, Vol. 8, No 10, 1248-1252, ISSN: 0724-8741.

Huuskonen J., Salo M. & Taskinen J. (1997), Neural Network Modeling for Estimation of the Aqueous Solubility of Structurally Related Drugs. *J. Pharm. Sci.*, Vol. 86, No 4, 450-454, ISSN: 0022-3549.

Kandimalla K. K., Kanikkannan N. & Singh M. (1999), Optimization of a vehicle mixture for the transdermal delivery of melatonin using artificial neural networks and response surface method. *J. Contr. Release.* Vol. 61, No 1-2, 71-82, ISSN: 0168-3659.

Kolarzyk E, Stepniewski M, Mendyk A, Kitlinski M & Pietrzycka A. (2006), The usefulness of artificial neural networks in the evaluation of pulmonary efficiency and antioxidant capacity of welders. *Int J Hyg Environ Health*, Vol. 209, No 4, 385-392, ISSN: 1438-4639 .

Linux4chemistry, http://www.redbrick.dcu.ie/~noel/linux4chemistry/

Mansa R.F., Bridson R.H., Greenwood R.W., Barker H. & Seville J.P.K., (2008), Using intelligent software to predict the effects of formulation and processing parameters on roller compaction. *Powder Technology*, Vol. 181, No 2, 217-225, ISSN: 0032-5910

Maqsood I., Khan M. R. & Abraham A. (2004), An ensemble of neural networks for weather forecasting. *Neural Comput & Applic,* Vol. 13, No 2, 112–122, ISSN 0941-0643

Mendyk A. & Jachowicz R. (2006), ME_expert - a Neural Decision Support System as a Tool in the Formulation of Microemulsions. *Biocybernetics and Biomedical Engineering*, Vol. 26, No 4, 25-32, ISSN: 0208-5216.

Mendyk A. & Jachowicz R. (2007), Unified methodology of neural analysis in decision support systems built for pharmaceutical technology. *Expert Systems with Applications*, Vol. 32, No 4, 1124–1131, ISSN: 0957-4174.

Mendyk A. & Jachowicz R., (2005) Neural network as a decision support system in the development of pharmaceutical formulation – focus on solid dispersions *Expert Systems With Applications*, Vol. 28, No 2, 285-294, ISSN: 0957-4174..

Polak S. & Mendyk A. (2004) Artificial Intelligence Technology as a Tool for Initial GDM Screening. *Expert Systems with Applications*, Vol. 26, No 4, 455-460, ISSN: 0957-4174.

Polański J. (2003), Self-organizing neural networks for pharmacophore mapping, *Adv. Drug Delivery Rev.*, Vol. 55, No 9, 1149-1162, ISSN: 0169-409X.

Rocksloh K., Rapp F.R., Abed Abu S., Müller W., Reher M., Gauglitz G. & Schmidt P.C. (1999) Optimization of Crushing Strength and Disintegration Time of a High-Dose Plant Extract Tablet by Neural Networks. *Drug Dev Ind Pharm*, Vol. 25, No 9, 1015-1025, ISSN 0363-9045.

Takahara J., Takayama K. & Nagai T. (1997), Multi–objective optimization technique based on an artificial neural network in sustained release formulations. *J. Control. Release*, Vol. 49, No 1, 11-20, ISSN: 0168-3659.

Takayama K., Fujikawa M., Obata Y. & Morishita M. (2003) Neural network based optimization of drug formulations. *Adv. Drug Delivery Rev.* Vol. 55, No 5, 1217-1231, ISSN: 0169-409X

Taskinen J. & Yliruusi J. (2003) Prediction of physicochemical properties based on neural network modelling. *Adv. Drug Delivery Rev.,* Vol. 55, No 5, 1163-1183, ISSN: 0169-409X.

Türkoğlu M., Özarslan R. & Sakr A. (1995) Artificial Neural Network Analysis of a Direct Compression Tabletting Study, *Eur. J .Pharm. Biopharm.*, Vol. 41, No 5, 315-322, ISSN: 0939-6411.

Veng-Pedersen & P. Modi N.B. (1992), Neural Networks in Pharmacodynamic Modeling. Is Current Modeling Practice of Complex Kinetic Systems at a Dead End? *J. Pharm. Biopharm.*, Vol. 20, No 4, 397-412, ISSN: 1567-567X.

Wikipedia (2009 a), http://en.wikipedia.org/wiki/Artificial_neural_network

Wikipedia (2009 b), http://en.wikipedia.org/wiki/Data_mining

Wikipedia (2009 c), http://en.wikipedia.org/wiki/Decision_support_system

Yager, R.R. & Filev, D.P., (1994), *Essentials of fuzzy modeling and control*. John Wiley & Sons, Inc., USA

Żurada J.M. (1992). *Introduction to Artificial Neural Systems*, West Publishing Company, ISBN-10: 053495460X, USA.

Żurada J.M., Malinowski A. & Usui S. (1997) Perturbation Method for Deleting Redundant Inputs of Perceptron Networks. *Neurocomputing*, Vol. 14, No 5, 177-193, ISSN: 0925-2312.
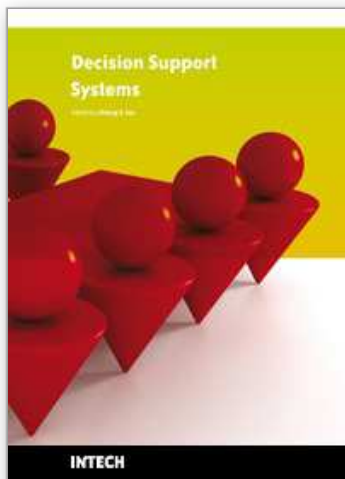
Decision Support Systems

Edited by Chiang S. Jao

Decision support systems (DSS) have evolved over the past four decades from theoretical concepts into real world computerized applications. DSS architecture contains three key components: knowledge base, computerized model, and user interface. DSS simulate cognitive decision-making functions of humans based on artificial intelligence methodologies (including expert systems, data mining, machine learning, connectionism, logistical reasoning, etc.) in order to perform decision support functions. The applications of DSS cover many domains, ranging from aviation monitoring, transportation safety, clinical diagnosis, weather forecast, business management to internet search strategy. By combining knowledge bases with inference rules, DSS are able to provide suggestions to end users to improve decisions and outcomes. This book is written as a textbook so that it can be used in formal courses examining decision support systems. It may be used by both undergraduate and graduate students from diverse computer-related fields. It will also be of value to established professionals as a text for self-study or for reference.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds