Pharmacological Reports
2012, 64, 761–781
ISSN 1734-1140

Review

Prediction of enzyme activity with neural network models based on electronic and geometrical features of substrates

Maciej Szaleniec

Joint Laboratory of Biotechnology and Enzyme Catalysis, Institute of Catalysis and Surface Chemistry, Polish Academy of Science, Niezapominajek 8, PL 30-239 Kraków, Poland

Correspondence: Maciej Szaleniec, e-mail: ncszalen@cyfronet.pl

Abstract:
Background: Artificial Neural Networks (ANNs) are introduced as robust and versatile tools in quantitative structure-activity relationship (QSAR) modeling. Their application to the modeling of enzyme reactivity is discussed, along with methodological issues. Methods of input variable selection, optimization of network internal structure, data set division and model validation are discussed. The application of ANNs in the modeling of enzyme activity over the last 20 years is briefly recounted.
Methods: The discussed methodology is exemplified by the case of ethylbenzene dehydrogenase (EBDH). Intelligent Problem Solver and genetic algorithms are applied for input vector selection, whereas k-means clustering is used to partition the data into training and test cases.
Results: The obtained models exhibit high correlation between the predicted and experimental values ($R^2 > 0.9$). Sensitivity analyses and study of the response curves are used as tools for the physicochemical interpretation of the models in terms of the EBDH reaction mechanism.
Conclusions: Neural networks are shown to be a versatile tool for the construction of robust QSAR models that can be applied to a range of aspects important in drug design and the prediction of biological activity.

Key words:
artificial neural network, multi-layer perceptron, genetic algorithm, ethylbenzene dehydrogenase, biological activity

Introduction

In recent years, the construction of predictive models assessing the biological activity of various chemical compounds has gained deserved respect, not only among medical chemists but also among biologists, pharmacists and synthetic chemists who appreciate the powerful explanatory and predictive powers of such models in solving even very complex problems. *In silico* drug candidate screening for potential activity or toxicity is now a standard procedure used in most drug development projects. The foundations for this field of science were laid by Corwin Hansch [20, 21], who had the evident but still ingenious idea that the structure of a compound should be connected with its chemical, biological or pharmacological activity and that it can be described with statistical methods. This idea led to the development of the huge and robust field of structure-activity analysis, which finds application not only in drug development but also in many other fields, such as the following:

i) synthetic organic chemistry (to understand reaction mechanisms),

ii) chromatography (to predict chromatographic retention times),

iii) physical chemistry (to predict various important physicochemical properties, such as log P, solubility, refractivity etc.) and

iv) catalysis and biocatalysts (in which models are built to find novel substrates and investigate factors determining catalysts reactivity).

By no means should these applications be separated from one another, as these fields are closely related. For example, quantitative structure-property relationships (QSPRs), developed for the prediction of log P in physical chemistry, have the utmost importance in the preparation of many models for drug design, whereas models developed by chemists working in biocatalysis can frequently be transferred into pharmacology for drugs targeting not receptors but specific enzymes of interest.

Traditionally, quantitative structure-activity relationships (QSARs) are constructed with linear models that link a dependent variable in the form of a logarithm of kinetic rate constant (log k), or logarithm of equilibrium constant (log K) with a linear combination of various independent variables that describe properties of the investigated compounds. Such an approach has various merits, among which the simplicity of interpretation appears to be the strongest. However, this approach is typically limited to the linear relationships between a dependent variable and independent predictors. This limitation is sometimes circumvented by the introduction of binary or quadratic interactions and splines [14], but experience reveals that such an approach is rarely practical, due to an increase in the predictor number in the case of the cross-terms and a decrease in the model generalization capabilities in the case of splines. As long as the investigated relationships are indeed highly linear, the standard QSAR approach is mostly valid and can have generalization powers even outside the descriptor values range used for model creation. Such a situation, however, is rare, as pure linear relationships are typically found in very simple or isolated systems in which no complex interactions take place. Of course, one can hardly expect that real biological problem can be narrowed down to such idealized conditions and still be fully described. This situation leaves basically two options: i) the construction of linear models that are valid locally and are used only for the assessment of the activity of similar congeners or ii) the application of more advanced statistical tools that can handle non-linear relations and still be trained with a reasonable number of cases. The latter solution leads to the tool that forms the topic of this paper – artificial neural networks.

Artificial neural networks

Features

Artificial neural networks (ANNs) are modular informatics soft computing methods that can be used to automate data processing [45]. They were designed as an artificial computer analog of biological nervous systems and, to some extent, possess the same wonderful characteristics. Although they are frequently regarded as the mysterious offspring of artificial intelligence from science fiction novels, it is better to treat them as a slightly more advanced form of mathematical equations linking a handful of inputs with a predicted dependent variable. Nevertheless, neural networks surpass standard linear or even non-linear models in multiple characteristics [62]:

– ANNs optimize the way they solve the problem during the iterative training procedure – as a result, there is no need to predefine the structure of equations. The neural models are built from modular processing elements with standard, predefined mathematical characteristics (i.e., the functions used for data processing). The mathematical nature of a neural model is naturally dependent on the structure of the network and the type of selected activation and transfer functions. However, many studies concerning the behavior of neural models showed a lack of unequivocal relationships between the performance of the obtained ANN, its structure and the mathematical functions utilized by its processing elements. It has been shown that almost identical predictions can be obtained from models with markedly different structures and mathematical characteristics. Therefore, although the ANN user is typically expected to define a network structure and the type of the neurons to be used in the modeling, such selection does not equate with the choice of the particular mathematical formulas to be used in classical modeling. The selection of the ap-

propriate functions to be utilized by the processing elements is typically conducted with automatic or heuristic methods or proposed by the authors of the ANN modeling application and is not typically of great concern to the ANN user.

– ANNs learn using examples that are presented to the networks in iterations (called epochs) and minimize the error of their predictions. In contrast to many non-linear optimization protocols, this process does not require any *a priori* parameters to reach a minimum, as the initial parameters (weights) are assigned randomly. Advanced training algorithms are very efficient in the minimization of the prediction error and are able to circumvent the problem of local minima. That efficiency does not imply, however, that the problem is solved automatically, regardless of the structure and characteristics of the chosen neural model. The selection of appropriate training algorithms and their performance parameters (such as, for example, the learning rate η used in back propagation, quick propagation and delta-bar-delta, learning momentum, or Gaussian noise) frequently has an important influence on a model's training time and its final quality (see Training section).

– ANNs comprise informatics neurons that collect inputs and process them with non-linear functions. This procedure automatically eliminates the need for the introduction of cross-terms or quadratic terms [65]. The mutual interactions, if necessary, will emerge in the network structure. Moreover, non-linearity is essentially a structural ANN feature, which allows more complex modeling (i.e., the creation of a much more complex response surface that is possible in the case of linear models).

– ANNs exhibit great versatility in the way the dependent variable is predicted, i.e., they can work in either a regression or a classification regime. In the former case, a continuous dependent variable is predicted, whereas in the latter, it is provided in the form of non-continuous categorical classifiers (for example 0 – non-active, 1 – active, 2 – very active). Moreover, as the non-linearity is built in, the relationship of the predictors to the predicted dependent variable does not have to be continuous. For example, it is possible to model enzyme substrate conversion rates together with inhibitors, although there is no common scale for both groups.

– ANNs typically exhibit higher generalization capabilities than linear models. As such, they are predestined to obtain global relationships, in contrast to the local ones characteristic for the linear models.

All of these advantages are so powerful that one can wonder why neural networks have not superseded traditional linear approaches in structure-activity research. The answer is very simple – artificial neural networks have an important drawback that limits their application. In most cases, the robustness of neural models comes at a cost – the algorithms are too complex to be analyzed by their users. As a result, there is no straightforward method for the physicochemical interpretation of such models. Thus, there is a simple tradeoff between prediction quality and interpretability. As neural networks are better prepared to model complex nature, they are much harder to understand than simple linear models. Moreover, as complex problems can be frequently solved in more than one way, it is very common to obtain a range of good ANN models that utilize different descriptors. As a result, there is no ultimate answer for a given problem, which yields additional uncertainty in the interpretation.

Nevertheless, as ANNs are models that offer theoretical representations of the studied problem, in principle it is possible to study them and gain insight into the described process.

Structure of neural networks

There are numerous different types of ANNs. A description of these types is beyond the scope of this paper and therefore, if readers are interested in this specific subject, they should refer to an appropriate textbook on neural science [8, 12, 41]. As the feed-forward Multi-Layer Perceptrons (MLPs) are most frequently utilized in QSAR research, they are also used as an example in this paper. Their short description is provided below.

MLPs are built from artificial neurons (Fig. 1), frequently referred to as processing elements (PEs). Each neuron collects signals from other neurons (through artificial synapses) and assigns a different importance to each signal. This assignment is typically achieved by the linear combination of each signal x_i multiplied by an adjustable weight constant w_i plus a constant θ used as a neuronal activity threshold (the aggregation or transfer function ζ).

The result of the aggregation function ζ is subjected to further mathematical processing by the so-called activation function $\sigma(\zeta)$, which is responsible for the in-

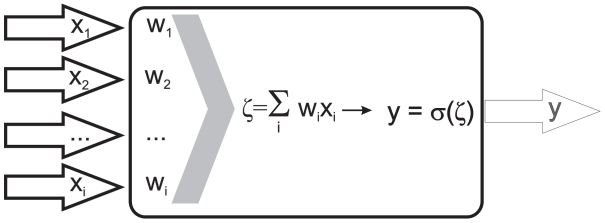


Fig. 1. The schematic of the artificial neuron. ζ – aggregation function, σ – activation function, x_i – input signal, w_i - weight, θ – bias, y – output signal

troduction of non-linearity into the model. Although there is a range of different types of activation functions, the most frequently used are logistic or continuous log-sigmoid functions that allow the smooth adjustment of the neuron activation level to the given set of inputs. The result of the activation function *y* is further transmitted along an artificial connection (axon) to become an input signal for other PEs.

As mentioned above, the most common neural architecture utilized in QSAR research is feed-forward MLP (Fig. 2). In such a network, the neurons are aligned in three (rarely more) layers, and signal is transferred in only one direction, from the input neurons toward the output neurons. In the input layer, the PEs are responsible for the introduction of descriptors

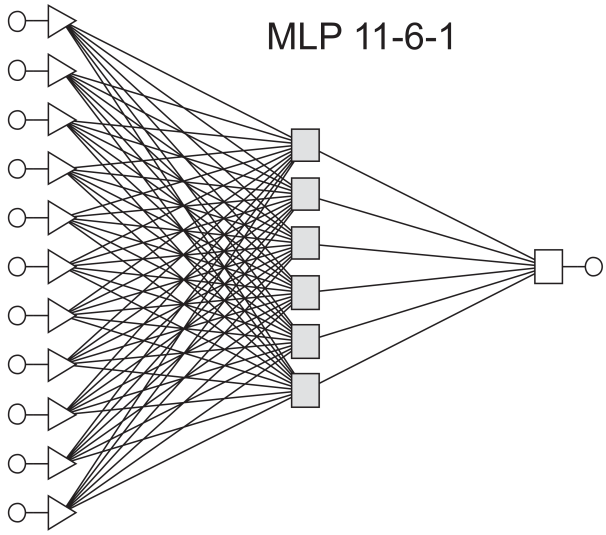


Fig. 2. A schematic representation of an MLP neural network. Triangles – input neurons, gray rectangles – hidden neurons, white rectangle – output neuron, circle – input and output numerical variables. MLP 11-6-1 – 11 input neurons, 6 hidden neurons, 1 output neuron

into the model, and their only task is to normalize the value of the parameter and transfer it to each neuron in the next ('hidden') layer. The hidden layer is the most important part of the network, performing, in fact, most of the 'thinking'. Each PE of this layer collects the signals from all of the input neurons and produces various levels of activation as an output. The final signals are collected in the output neuron(s), which produces the predicted value as its output.

Training

ANN training methods can be divided into two general groups: supervised and unsupervised. In both cases, the ANN is mapping input data into its structure, but during supervised learning, a dependent variable is also provided. As a result, the ANN is trained to associate characteristics of the input vectors with a particular value or class (dependent variable). In contrast, in the case of unsupervised learning, no dependent variable is provided, and the resulting map of cases is based on the intrinsic similarities of the input data. The most common unsupervised networks are Kohonen networks (Self Organizing Maps) [31], which perform mapping of an N-dimensional input space onto a 2D map of Kohonen neuron. Such an approach is excellent for the investigation of complex data structures.

In structure-activity relationship (SAR) research, ANNs with supervised learning are most frequently encountered. Therefore, the description of this training method is the most relevant for the topic of that paper.

The mean difference between the predicted values of the dependent variable and the experimental values determines the error of prediction. The main aim of the training algorithm is to decrease this error. This reduction is achieved by so-called 'back-propagation' algorithms that are based on the value and gradient of the error. These algorithms adjust the weights in the network's PEs in a previous layer (thus propagating an error signal 'back') in a step-by-step manner, aiming at the minimization of the final error. As a result, the ANN learns by examples what prediction should be made for a particular set of inputs and adjusts its mathematical composition in such a way as to be the most efficient. There are a number of different algorithms available, but the back propagation algorithm was the first one. Its discovery was a cornerstone of the application of ANNs in all fields of science [12].

The price paid for the greater (non-linear) modeling power of neural networks is that, although it is possible to adjust a network to lower its error, one can never be sure that the error could not be still lower. In terms of optimization, this limitation means that one can determine the minimum of the network error but cannot know whether it is the absolute (global) minimum or simply a local one.

To determine minimum error, the concept of the error surface is applied. Each of the N weights and biases of the network (i.e., the free parameters of the model) is taken to be a dimension in space. The N + 1 dimension is the network error. For any possible configuration of weights, the error for all of the training data can be calculated and next can be plotted in the N + 1 dimension space, forming an error hypersurface. The objective of network training is to find the lowest point in this hypersurface. Although the literature describing the learning methods depicts error surfaces as smooth and regular, the neural network error surfaces for real tasks are actually very complex and are characterized by a number of unhelpful features, such as local minima, flat spots and plateaus, saddle-points and long, narrow ravines. As it is not possible to analytically determine where the global minimum of the error surface is, the training of neural network is essentially an exploration of the error surface. From an initially random configuration of weights and thresholds, the training algorithms incrementally search for the global minimum. Typically, the gradient of the error surface is calculated at the current point and used to make a downhill move. The extent of the weight correction is controlled by the learning rate parameter, η (also called the learning constant). The value of η has an influence on the rate of the model training – for learning rates that are too small, the final (optimal) set of weights will be approached at a very slow pace. In contrast, values of the parameter that are too high may result in jumping over the best descent to the global minimum of the error surface, which may result in the instability of the training procedure. In most cases of ANN application, the character of the error surface is unknown, and the selection of the learning rate value has to be completely arbitrary. It is only possible to tune its value as a result of ex-post analysis of the training procedure. Fortunately, only the back-propagation algorithm requires a fixed a priori determination of the η value. Most of the modern algorithms (such as quick propagation, quasi-Newton-based algorithms or conjunct gradient algorithms) apply the adaptive learning rate, which results in incremental convergence along the searching direction [69]. To attain more efficient ANN training, additional tricks are sometimes applied, such as momentum (which maintains the gradient-driven direction of optimization and allows passing over local minima) or Gaussian noise (added to the training input values, also aimed at jumping out of the local minima). Eventually, the algorithm stops in a low point, which hopefully is the global minimum.

Of course, ANNs are prone to the same over-fitting problems as any other model. The origin of the problem is the so-called over-training of the network. One of the possible descriptions of over-training is that the network learns the gross structure first and then the fine structure that is generated by noise [65] or that it accommodates too much to the structure of the learning data. The problem is handled by splitting the database into training and test sets [60]. The error of the test set is monitored during the ANN optimization procedure but is not used for ANN training. If the error of the test set starts to rise while the error of the training set is still decreasing, the training of the ANN is halted to avoid over-training (which can be compared with 'learning by heart' by the student).

As, to a certain extent, the test set indirectly determines the training process (the decision when to stop), it is also advisable to design yet another validation subset, which is used as an ultimate robustness cross-check. If the error obtained for the validation and test sets is in the similar range, one can be fairly sure that the model is not over-fitted and retains high generalization powers. Moreover, although the error of the training set is, in most cases, lower than those of the test and validation sets, it should also be in a similar range. If the training error is many orders of magnitude lower than that obtained for validation and test sets, it is a sure indication of over-fitting. In other words, the neural model is too well accommodated to describing the training cases and likely will not have high prediction and generalization capabilities. In such a case, the training of the model should be repeated and/or the initial neural architecture modified to avoid over-training.

Data partitioning

As was mentioned above, before training starts, the available data set has to be partitioned into at least

two subsets (the training and test subsets) [11]. The main problem in the selection of the validation and test cases is their representativeness to the whole studied population. In most QSAR problems, each compound is, to a certain extent, a unique case with distinct chemical features. Removal of such a compound from the data set excludes part of the important knowledge from which neural model learns how to solve the problem.

The test set is supposed to be representative for the whole data set, as it controls the endpoint of the training. If the test cases are too easy for the model to predict, the error might be lower than for more difficult cases. As a result, such an easy test set will allow over-fitting effect to take place unnoticed. In contrast, if the test set is composed from compounds that have features that do not occur in the training set, the model might not be able to extrapolate its knowledge to predict their activity correctly. This case yields a low training error and a high test error, which should result in the rejection of the model. Finally, the validation set should also be representative, but it may contain several 'difficult' cases. As the validation set is not involved in the training, one can use it to assess the generalization capabilities of the network.

The most common method used is random partitioning of the cases into three subsets (in which a 2:1:1 ratio is very common). Such an approach is correct as long as the data set is numerous, and it is reasonable to assume that randomized choice will guarantee the representativeness of all subsets [59]. It is worthwhile to warn potential ANN users that the subset partitioning should be performed only once before the optimization starts. Many programs have the option of shuffling the test and training cases between attempts at ANN optimization to diminish the chance of an uneven random selection of the data sets. However, with the ever increasing power of computers, it is possible, in a very short time, to train thousands of models. When the cases are randomly parted into test and training subsets and only the best models are selected from the tested group, it is possible to 'randomly optimize' the easiest possible test and validation sets. In such a case, the model can have extremely low errors but virtually no generalization capabilities being over-fitted to all cases. Approximately two years ago, such a problem was encountered in our own research and was carefully analyzed and discussed [57]. Therefore, it is best to avoid random partitioning and to use chemical sense in the selection of the test and the validation representatives or to apply partitioning combined with cluster analysis. Cluster-based partitioning uses all collected descriptors and dependent variables to divide compounds into subgroups. The cases are selected from these groups based on random choice, the distance from the cluster center or any other feature. It is also sensible to include fingerprint descriptors among the variables used in partitioning. The fingerprint descriptors allow more objective chemical (i.e., structure-based) selection of the test and validation cases. Such an approach was used, for example, in the papers of Andrea et al. and Polley et al. [2, 42], and in the example of our research provided below. Recently, a custom protocol allowing the easy partitioning of data sets based on cluster analysis was introduced into Accelrys Discovery Studio 2.5.

Selection of the input vector

In most experimental studies, the QSAR researcher is confronted with a limited number of experimental cases that can be collected and an almost limitless number of possible descriptors that can be used in the model construction. Sometimes, it is very difficult to judge which of the variables available in many QSAR programs will be best suited for the modeling. That situation typically leaves us with the problem of an undetermined data set, in which the number of tested compounds is far lower than the number of descriptors. In the past, there were numerous strategies developed that handled that problem (also referred to as the 'dimensionality problem'). For standard linear regression, forward and backward stepwise selection algorithms were used that test the correlation of the particular descriptor with the dependent variable. Such an approach, however, does not take into account any interactions occurring between the descriptors and is vulnerable to the so-called co-linearity problem. Finally, such algorithms are limited to the selection of linearly correlated descriptors.

Another strategy is based on principal component analysis (PCA), which forms a linear combination of a number of descriptors and selects only those components that have the greatest influence on the determination of variance of the dependent variable. Such a strategy is also routinely used in PLS protocols employed in 3D Comparative Molecular Field Analysis (COMFA)

QSAR. Although PCA is a very effective strategy in decreasing the number of dimensions of the input vector, the obtained PCA components do not have straightforward physicochemical interpretations, which renders them less valuable for explanatory models.

ANNs are also subject to the dimensionality problem [64]. The selection of the most important variables facilitates the development of a robust model. There are a number of approaches that can be used:

1. **Removal of redundant and low-variance variables**. In the first place, the manual removal of descriptors that have a pairwise correlation coefficient greater than 0.9 should be performed. In such a case, one of two redundant descriptors is removed, as it does not introduce novel information and typically will not improve the robustness of the model. However, there are several ANN researchers who suggest that, as long as both descriptors originate from different sources, they should be left in place due to possible synergetic effects they can enforce on the model. Moreover, it is advisable to remove descriptors that do not introduce valuable information into the data set. Low variance of a variable is typically an indication of such a situation. Such an approach was used, for example, by Kauffman and Jurs as a first step in their 'objective feature selection' protocol [27].

2. **Brute force' experimental algorithms.** As long as the training data set is relatively small in number (and typically in the case of QSAR applications, the data set does not exceed 100 compounds), one can attempt an experimental data selection approach. Due to the high performance of modern computers it is possible to build thousands of ANNs with different input vectors within an hour's time. If a uniform training algorithm is applied and a selection is based on the value of the test set error, such an approach allows the selection of the most robust input vector. An example of such an algorithm, Intelligent Problem Solver (IPS), can be found in the Statistica Neural Networks 7.0. Examples of applications of that strategy can be found in our previous papers [57, 59, 61].

3. **Genetic algorithm – linear models**. Another technique that is extremely efficient applies a genetic algorithm for the selection of the input vector [44]. Genetic algorithms utilize evolutionary principles to evolve an equation that is best adapted to the 'environment' (i.e., to the problem to be solved). The evolutionary pressure is introduced by a fitness function (based on the correctness of the prediction of the dependent variable) that determines the probability of a given specimen to mate and produce offspring equations. The optimization of the initially randomly distributed characteristics (i.e., input variables) is assured by the cross-over between the fittest individuals and by accidental mutations (insertion or deletion of individual feature-variables). However, in the case of neural networks, GA optimization becomes somewhat trickier than in the case of traditional linear models. It is important to consider the type of fitness function that is applied to select the best equation. In most cases, the genetic algorithms are intended to develop multiple regression models, and the fitness function assesses how well the obtained linear model predicts the desired dependent variable. Such algorithms favor the selection of variables that have a linear correlation with the predicted value and do not have to be necessarily the best suited for the construction of a non-linear neural network. If the problem can be solved well in a linear regime, the introduction of more flexible non-linear networks will not improve the performance [13]. However, if the problem is not linear, it is advisable to apply as many non-linear features in the GA as possible. These features can be binary or quadratic interactions as well as splines (inspired by the MARS algorithm of Friedman [14, 51]). Although splines allow interpolative models to be built, achieving response surfaces of considerable complexity from simple building blocks, it should be understood that these models no longer hold direct physicochemical interpretations, and as such, should be used with caution.

4. **Genetic algorithm – neural models.** Another approach to input selection is the use of genetic algorithms in conjunction with neural networks (so-called Genetic Neural Networks – GNN). The difference from the approach described above is that, in a GNN population, an evolutionary population comprises neural networks instead of linear equations. As a result, the fitness function assesses the prediction quality of a particular neural network instead of a linear equation. Other than this aspect, the process is basically similar. It involves a cross-over of the parent models and breeding offspring models with characteristics of both parents (in this case, parts of the input vector and sometimes structural features of the parent models). Moreover, a mutation possibility is allowed that introduces or eliminates certain of the input descriptors. Such an approach was carefully analyzed by So and Karplus in their series of papers [47–49] and in the study of Yasri and Hartsough [68]. The main advantage of such an approach is that the fitness of

the model has already been assessed within a non-linear neural architecture. Therefore, there is no danger that crucial non-linear relationships will be lost during the optimization protocol. Moreover, the algorithm delivers a number of well-suited models that solve the same problem, sometimes in different manners. This feature allows a more robust and error-proof prediction strategy to be employed and provides more insight into the studied problem. However, as usual, there is a significant cost of such an approach. Each network must be iteratively trained before its prediction is ready to be assessed by the fitness function. In the case of a reasonable population of models (a few hundred) and a sensible number of generations that ensures the stability of the selected features (at least 100, but in many cases 1000+), the optimization of the models becomes very time-consuming. This consideration may explain the relatively low popularity of this technique. However, as the desktop computers become faster and faster, this problem should soon be alleviated. One way to avoid this handicap is to use Bayesian Neural Networks (also called Generalized Regression Neural Networks – GRNNs), which are famed for their instant training. This method allows a quick optimization of the input vector in a non-linear regime but introduces a need to train new MLP neural models based on the result of the optimization protocol.

5. **Post-processing sensitivity analysis.** Finally, one can resort to a manual elimination of variables based on a sensitivity analysis. In this method, the initial network is trained, and the influence of a particular input variable is assessed. The assessment is based on the ratio of error calculated for models with and without a particular input variable. If the prediction of ANN improves after the removal of a particular descriptor, the error ratio assumes values below 1 and suggests that the variable can be safely removed without compromising the ANN prediction capabilities. Every time the input vector is modified, the neural network must be retrained and the sensitivity analysis performed again. Moreover, one must keep in mind that the decreasing number of inputs combined with the constant size of the hidden layer increases the danger of over-fitting (see the next section). Therefore, it is best to adjust the number of hidden neurons as well. Although this protocol requires more experience from its user, it has been found to be very effective in our research and has been applied frequently as a final touch after the application of automatic optimization protocols [59].

Regardless of the method used for pruning the initial descriptor number, it is always important to keep the number of independent variables in check to avoid a chance correlation effect [66]. Jurs et al. [34] suggest keeping the ratio of the number of descriptors to the training set compounds below 0.6 as a useful precaution. Moreover, one can apply a scrabbling technique to test whether a chance correlation is not present in the final model. This method randomly assigns the dependent variable to the training cases and constructs the model. As, under such conditions, no real structure-activity can exist, a significant deterioration of the prediction should occur.

Determination of the size of the hidden layer

For a Multiple-Layer Perceptron, the correct determination of the input layer has the utmost importance. However, it is also very important to properly adjust the capacity of the hidden layer. If there are too many neurons, the network will be prone to over-fitting. The system will tend to accommodate too much to the training data, exhibiting a loss of generalization capabilities. Such a behavior can be spotted during manual network training as a rapid increase in the test error curve. Sometimes, however, this issue can occur in a slightly more subtle form, when the training error is decreasing while the test error is not, keeping in a constant horizontal trend (Fig. 3) [56, 61]. In such a situation, it is worthwhile to decrease the number of hidden neurons and repeat the training. It is advisable to optimize the size of the network to the minimal number of hidden neurons that still provides sufficient computational capacity to ensure a good model performance. Many authors tend to assess model robustness as a function of the number of hidden neurons [2, 7, 18]. Others utilize automated protocols (such as IPS or GA) for the optimization of the hidden layer [18, 59, 68]. Automated protocols are present in most of the software packages devoted to ANN, such as Statistica Neural Network (IPS) or NeuroSolution.

Generally speaking, the smaller model the better, as it requires fewer examples to be trained. This feature, in turn, allows a more profound testing approach, with larger testing and validation sets. It should be kept in mind, however, that there are no

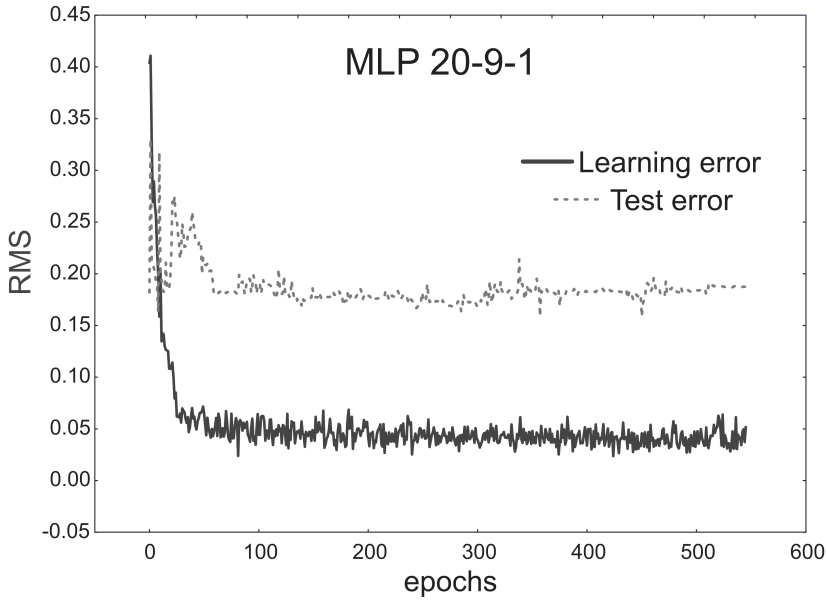


Fig. 3. An example of ANN training with too high a capacity. The parallel trend of the test error indicates overfitting [57, 62]

analytical rules describing the maximal size of the network for a given number of training cases, as different relationships may exhibit various degrees of linear dependence. The more complicated the dependence, the more cases are required for a proper training. However, several authors provide a 'rule of thumb' introduced by Andrea et al. [2, 50] linking the total number of adjustable parameters with the number of cases. For a three-layer MLP, the number of adjustable parameters P is calculated from the following formula:

$$P = (I + 1)H + (H + 1)O$$

where I, H and O are the numbers of input, hidden and output neurons, respectively, and

$$\rho = \text{no. of training cases}/P$$

Adrea et al. suggest that ρ, the ratio of input training data to P, should be in the range of 1.8 to 2.2. Models with a ρ value close to 1 are prone to overfitting, whereas those with a ρ above 3 are unable to extract all of the important features and have poor prediction and generalization capabilities. However, our own experience in this matter suggests that the optimal range of ρ is problem-dependent. Nevertheless, every user of ANNs should be aware of all of the possible pitfalls connected with the above-mentioned problems and should apply rigorous cross-validation schemes.

Types of descriptors and their encoding

Almost all of the types of numerical descriptors used in QSAR can be fed into the neural network. Continuous variables, such as log P or molecular weight, are introduced into the network by single input neurons. The same situation occurs for categorical parameters that assume only two binary values. However, the descriptors that assume three or more non-continuous values require more attention. One can encode any feature with numerical codes (for example, attribute 0 to the substituent *ortho* position in the phenyl ring, 1 to *meta* and 2 to *para*), but it should be kept in mind that a neural network will try to find a connection between the introduced numeric values (in this case, the values of codes) and the predicted parameter. As there are no grounds to attribute a higher numerical value to *para*-substituted compounds and there is no logical justification favoring on such encoding over another, such a presentation of descriptors delivers false information into the model. To avoid this problem, 'one-of-N' encoding is used, which utilizes one neuron per code of the descriptors. In the case of substituent position in the phenyl ring, it must be encoded by 3 binary-value neurons (each able to assume a value of 0 or 1) that describe the presence or absence of the substituent in the *ortho*, *meta* or *para* position, respectively. This approach renders fingerprint descriptors not very useful for ANN, as their sheer length necessitates the

formation of huge input layers, which in turn requires a huge training data set.

In addition to the points discussed above, the selection of the input descriptors has the same requirements as in every QSAR: descriptors relevant to the studied problem should be included, describing electronic, thermodynamic, hydrophobic, steric, structural and topological characteristics of the investigated compounds.

Applications of neural networks in enzyme studies

The application of neural networks in drug design has caught the interest of many authors. This topic was reviewed by Winkler [67], Duch et al. [10] and, most recently, Michielan and Moro [36]. However, the topic of this paper, the application of neural networks for the prediction of enzymatic activity, appears to be not so popular. It should be mentioned that this topic can be approached from two perspectives. Certain authors are more concerned with the prediction of the enzymes' reaction rates from an engineering point of view. Therefore, these authors focus on identifying relationships between experimental variables, such as temperature, spinning rate of the reactor, and concentrations of various reagents and biocatalysts, with the observed conversion rate, product yield or fermentation process [1, 32, 46]. As these studies do not have any connection with SAR, they are beyond the scope of this paper.

The second group of papers, a literature concerning applications of ANNs in enzyme reactivity prediction, can be further divided into two subsets. Certain papers focus on the development of methodologies and utilize a handful of benchmark data sets that allow model comparisons. The other group of papers is problem-oriented and utilizes developed methodologies for the description of specific biological phenomena.

To our best knowledge, one of the earliest applications of neural networks for the prediction of enzyme biological activity as a function of chemical descriptors should be attributed to Andrea and Kalayeh [2]. Their paper used the benchmarking data set – diaminodihydrotriazines inhibitors of dihydrofolate reductase (DHFR) and utilized back-propagation feed-forward MLP models. This pioneering work was soon followed by So and Richards [50], who, in turn, studied the inhibition of DHFR by 2,4-diamino-5-(substituted-benzyl)pyrimidines. Both papers demonstrated that ANNs outscore traditional linear QSAR approaches. Both model data sets of DHFR inhibitors were also used by Hirst et al. to compare the robustness of the multivariate linear models, non-linear ANNs and inductive logic programming (ILP) methods [22, 23]. Their work suggested that the ANN and ILP methods did not significantly surpass linear models due to the over-fitting and data splitting problems. The topic of the potential superiority of ANNs over traditional linear models was also discussed in the paper of Lučić et al. [33]. These authors advocate the construction of non-linear regression models over neural networks or neural network ensembles (i.e., groups of NN models averaging their predictions). The DHFR pyrimidine derivatives data set and the HIV-1 reverse transcriptase derivatives of 1-[2-hydroxyethoxy)methyl]-6-(phenylthio)thymine (HEPT) inhibitors were also used by Chiu and So to construct ANN models based on Hansch substituent constants (π, MR, F and R) derived from QSPR models [9]. Such an approach enhanced the interpretability of the obtained models and delivered another proof of the higher predictive powers of ANNs in comparison to linear MLR models.

A handful of other successful applications of neural networks to SAR modeling should be mentioned. Novic et al. applied a counter-propagation neural network (CP-ANN) to model the inhibitory effect of 105 flavonoid derivatives on $p56^{lek}$ protein tyrosine kinase [39], whereas Jalali-Heravi et al. used MLP ANNs to study the inhibitory effect of HEPT on HIV-1 reverse transcriptase [25] and investigated possible inhibitors of heparanase [24]. Both papers argue against superior ANN performance in comparison to that of linear models, which has been attributed to the partially non-linear characteristics of inhibitory action. The inhibition of carbonic anhydrase was studied by Mattoni and Jurs [34] with both regression and classification approaches. Approximately a 20–30% improvement of the prediction quality in the case of neural networks with respect to MLR models was observed. Nevertheless, the paper by Zernov et al. argues that supported vector machine (SVM)-based models are of a better quality than ANNs in the estimation of the activity of carbonic anhydrase II inhibitors [70]. Another paper by Kauffman and Jurs modeled the activity of selective cyclooxygenase-2 inhibitors with ensembles of ANNs [27]. These authors claim that the applica-

tion of a committee of nonlinear neural networks has a specifically beneficial effect on the quality of the prediction in cases in which a high degree of uncertainty and variance may be present in the data sets. An interesting example of a similar approach by Bakken and Jurs has been applied to the modeling of the inhibition of human type 1 5α-reductase by 6-azasteroids [3]. The authors constructed multiple linear regression, PCA, PLS and ANN models predicting the pK_i values of inhibitors of 5α-reductase and compounds' selectivity toward 3β-hydroxy-Δ^5-steroid dehydrogenase/3-keto-Δ^5-steroid isomerase. Again, neural network models appeared to have higher predictive powers than their linear counterparts. An interesting and original application of ANNs can be found in the paper of Funar-Timofei et al., who developed MLR and ANN models of enantioselectivity in the reaction of oxirane ring-opening catalyzed by epoxide hydrolases [15]. The application of neural models significantly improved the prediction relative to that of the linear models.

Apart from the most common MLPs, there are a number of different neural architectures that have been successfully applied to studies on enzyme activity. In addition to the already-mentioned CP-ANNs, Bayesian regularized neural network (BRANN [7]) models were used with success by Polley et al. [42] and Gonzalez et al. [18] to predict the potency of thiol and non-thiol peptidomimetic inhibitors of farnesyltransferase and the inhibitory effects of flavonoid derivatives on aldose reductase [13]. The authors note that the application of the Bayesian rule to the ANN automatically regularizes the training process, avoiding the pitfalls of over-fitting or overly complex solutions. This benefit, according to Burden [7], is due to the nature of the Bayesian rule, which automatically panelizes complex models. As a result, authors frequently do not use the test sets, utilizing all available compounds to train the networks. Another application of neural networks with similar architecture can be found in the paper of Niwa [38]. The author used probabilistic neural networks (PNNs) to classify with high efficiency 799 compounds into seven different categories exhibiting different biological activities (such as histamine H_3 antagonists, carbonic anhydrase II inhibitors, HIV-1 protease inhibitors, 5 HT_{2A} antagonists, tyrosine kinase inhibitors, ACE inhibitors, and progesterone antagonists). The obtained model is clearly meant to be used for HTS screening purposes (and is thus barely within the scope of this paper). A good experience with PNN models was also reported by Mosier and Jurs [37], who applied that methodology to the modeling of inhibition of soluble epoxide hydrolase set. The same data base was also modeled by McElroy and Jurs [35] with either regression (MLR or ANN) or classification (k-nearest neighbor (kNN), linear discriminant analysis (LDA) or radial basis function neural network (RBFNN)) approaches. Their research demonstrated that non-linear ANN models are better in the regression problems but that kNN classifiers exhibit better performance that ANNs using radial basis functions.

Finally, the input of our research should also be mentioned here. A handful of models have been derived that predicted the biological activity of ethylbenzene dehydrogenase, an enzyme that catalyzes the oxidation of alkylaromatic hydrocarbons to secondary alcohols. In our research, both regression and classification MLP networks [59] were used, as were Bayesian-type general regression neural networks (GRNNs) [56, 57, 61], which were based on descriptors derived from DFT calculations and simple topological indices. Our approach differed from that presented above. It was possible to construct a model that predicted the biological activity of both inhibitors and substrates. Moreover, our studies were more focused on gaining insight into the mechanism of the biocatalytic process than the development of robust QSAR modeling methods. Nevertheless, the discussed issues concerned the random cases partitioning problem, resulting in over-training, and the function of the selected ANN architecture [57].

It appears that the presentation of an example of enzyme activity modeling with ANNs might be beneficial to the reader. Therefore, a description of a simple case is presented below, illustrating the problem of enzyme reactivity model construction.

Practical example

Modeling problem

A molybdoenzyme, ethylbenzene dehydrogenase (EBDH), is a key biocatalyst of the anaerobic metabolism in the denitrifying bacterium *Azoarcus sp.* EbN1. This enzyme catalyzes the oxygen-independent, stereospecific hy-

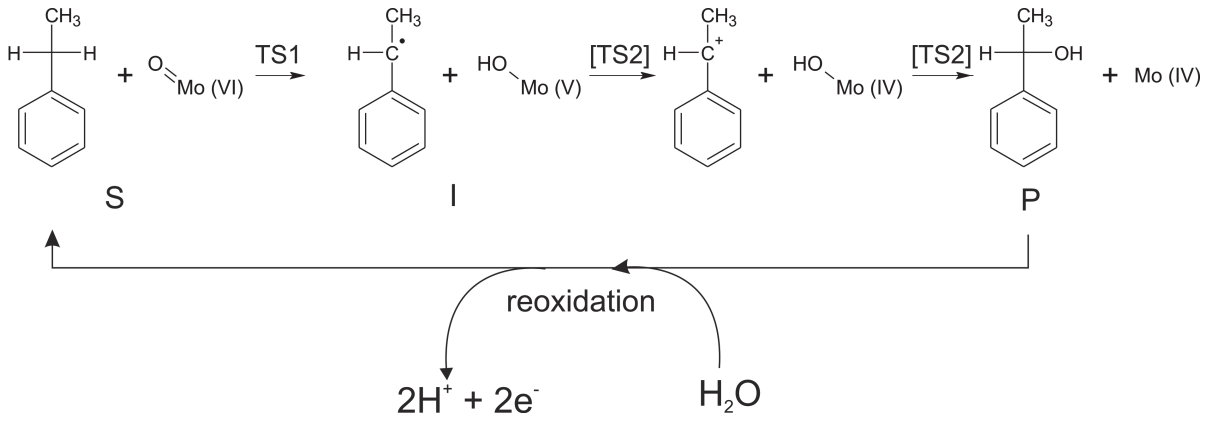


Fig. 4. The reaction scheme of ethylbenzene oxidation by EBDH. The reaction pathway involves two one-electron steps: homolytic C-H activation (TS1) leading to radical intermediate I and an electron transfer coupled to OH-rebound: [TS2] indicates that both processes take place in a concerted manner

droxylation of ethylbenzene to (*S*)-1-phenylethanol. It represents the first known example of the direct anaerobic oxidation of a non-activated hydrocarbon [26, 30]. EBDH promises potential applications in the chemical and pharmaceutical industries, as the enzyme is enantioselective [53] and reacts with a wide spectrum of substrates [54].

According to our recent quantum chemical modeling studies, the mechanism of EBDH [52, 55] consists of two steps (Fig. 4). First, the C-H bond of the substrate is cleaved in a homolytic way, and the hydrogen atom is transferred to the molybdenum cofactor, forming an OH group and the radical hydrocarbon intermediate. Then, OH is rebound to the radical hydrocarbon, and in the transition state associated with that step, a second electron is transferred to the molybdenum cofactor. As a result, in the vicinity of the transition state, a carbocation is formed. The completion of the OH transfer results in the formation of the product alcohol. After the reoxidation of the enzyme by an external electron acceptor, EBDH is ready for the next reaction cycle.

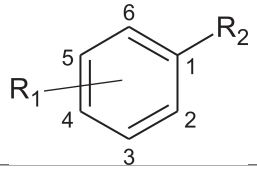
Our kinetic studies revealed that EBDH catalyzes the oxidation of ethyl and propyl derivatives of aromatic and heteroaromatic compounds. Recently, its activity with compounds containing cycloalkyl chains fused with aromatic rings (derivatives of indane and tetrahydronaphthalene) has been discovered, as well as activity with allyl- and propylbenzene. Moreover, EBDH is inhibited by a range of methyl and ethyl aromatic and heteroaromatic compounds [54, 59]. It also exhibits inhibition by its products (i.e., compounds with ethanol substituents) and non-reactive structural analogues of ethylbenzene (e.g., aromatic compounds with methoxy groups). Therefore, as the enzyme has potential application in the fine chemical industry, finding a reliable screening tool predicting its activity *in silico* is very appealing. Ideally, such a prediction model should be able to discern inhibitors and weak substrates from those which are converted with a high rate (and therefore can be oxidized on the commercial scale). It is also interesting to utilize insights gained from the QSAR studies in understanding the reaction mechanism.

Methods

Activity measurement

The specific activity of 46 compounds (Tab. 1) was assessed according to the procedure established previously [54] and related to the enzyme activity with the native substrate, ethylbenzene (rSA 100%). The activity of most compounds was fitted with a non-linear Michaelis-Menten model. In certain cases, it was necessary to fit the experimental result with a Michaelis-Menten equation with substrate inhibition. The biological activity of inhibitors (i.e., compounds that were not converted but exhibited various types of inhibitory effects on ethylbenzene) was always treated as 0.

Tab. 1. Substrates and inhibitors (n = 46) of EBDH. R_1 stands for substituents or modifications of the aromatic ring system (e.g., 1-naphthalene indicates that a naphthalene ring is present instead of a phenyl ring and that R_2 is in the 1- position), and R_2 describes structural features of the substituent that is oxidized to alcohol in the methine position. $-CH_2CH_2CH_2-$ like symbols depict cycloalkyl chains of indane, tetrahydronaphthalene or coumaran; rSA indicates specific activity related to EBDH activity with ethylbenzene in standard assay condition



No.	R_1	R_2	rSA [%]	rSA model 1	rSA model 2
	Training				
1	2-CH_2CH_3	CH_2CH_3	2.62	–21.4	-6.4
2	4-CH_2CH_3	CH_2CH_3	64.27	63.2	43.3
3	4-$CHOHCH_3$	CH_2CH_3	2.68	–1.7	1.2
4	1-naphthalene	CH_2CH_3	0	1.1	-20.5
5	1H-indene	CH_2CH_3	242	241.6	246.0
6	2-OCH_3	CH_2CH_3	1.1	–7.2	14.7
7	2-naphthalene	CH_3	9.3	12.8	7.4
8	2-SH	CH_2CH_3	0	14.0	–14.0
9	2-OH	CH_2CH_3	56.14	56.4	9.2
10	2-py	CH_2CH_3	0	7.5	–2.4
11	2-pyrrole	CH_2CH_3	234.63	233.9	277.0
12	2-CH_3	CH_2CH_3	3.8	11.7	–10.4
13	2-furan	CH_2CH_3	133.92	132.8	147.3
14	2-thiophene	CH_3	0	3.8	-2.3
15	3-NH_2	CH_2CH_3	25.45	34.6	–11.0
16	3-py	CH_2CH_3	3.72	6.1	3.5
17	H	CH_2CHCH	282.6	286.1	244.4
18	3-COCH3	CH_2CH_3	3.47	1.1	–21.4
19	4-NH2	CH_2CH_3	134	132.8	97.8
20	4-OCH3	CH_2CH_3	313.27	314.9	273.9
21	4-COOH	CH_2CH_3	0	0.7	–24.7
22	4-Ph	CH_2CH_3	29.65	29.4	5.1
23	4-OH	CH_2CH_3	259.01	253.1	241.8
24	4-py	CH_2CH_3	0	–26.1	–1.3
25	2,4-di-OH	CH_2CH_3	362.69	373.0	302.3
26	4-F	CH_2CH_3	6.4	6.3	-2.4
27	p-OCH3	$-CH_2CH_2CH_2-$	122.12	122.3	93.2
28	naphthalene	1,8-CH_2-CH_2-	0	2.3	–16.3
29	H	$-CH_2-CH_2-O-$	80	73.8	86.2
30	H	CH_2CH_3	100	96.1	60.4
31	H	$-CH_2CH_2CH_2-$	80.34	86.1	72.3
32	H	$CH_2CH_2CH_3$	14	16.1	33.8
33	H	CH_3	0	–8.3	–17.9
34	H	$CHOHCH_3$	0	–3.6	–7.7
	Test				
35	2-NH_2	CH_2CH_3	94.53	76.2	58.0
36	2-thiophene	CH_2CH_3	242.92	217.8	173.1
37	2-furan	CH_3	0	5.8	15.2
38	3-CH_3	CH_2CH_3	10	9.1	40.7
39	4-CH2OH	CH_2CH_3	7.4	24.5	34.4
40	4-CH3	CH_2CH_3	28	15.7	52.9
41	4-OH	$CH_2CH_2CH_3$	180	206.8	124.4
42	p-NH2	$-CH_2CH_2CH_2-$	33	12.7	47.6
	Validation				
43	2-Br	CH_2CH_3	0	–11.5	–21.3
44	3-OH	CH_2CH_3	24.31	12.3	34.4
45	H	$CH_2CH{=}CH_2$	281.5	226.4	258.8
46	H	CH_2 $CH_2CH_2CH_2-$	4.45	–9.8	86.

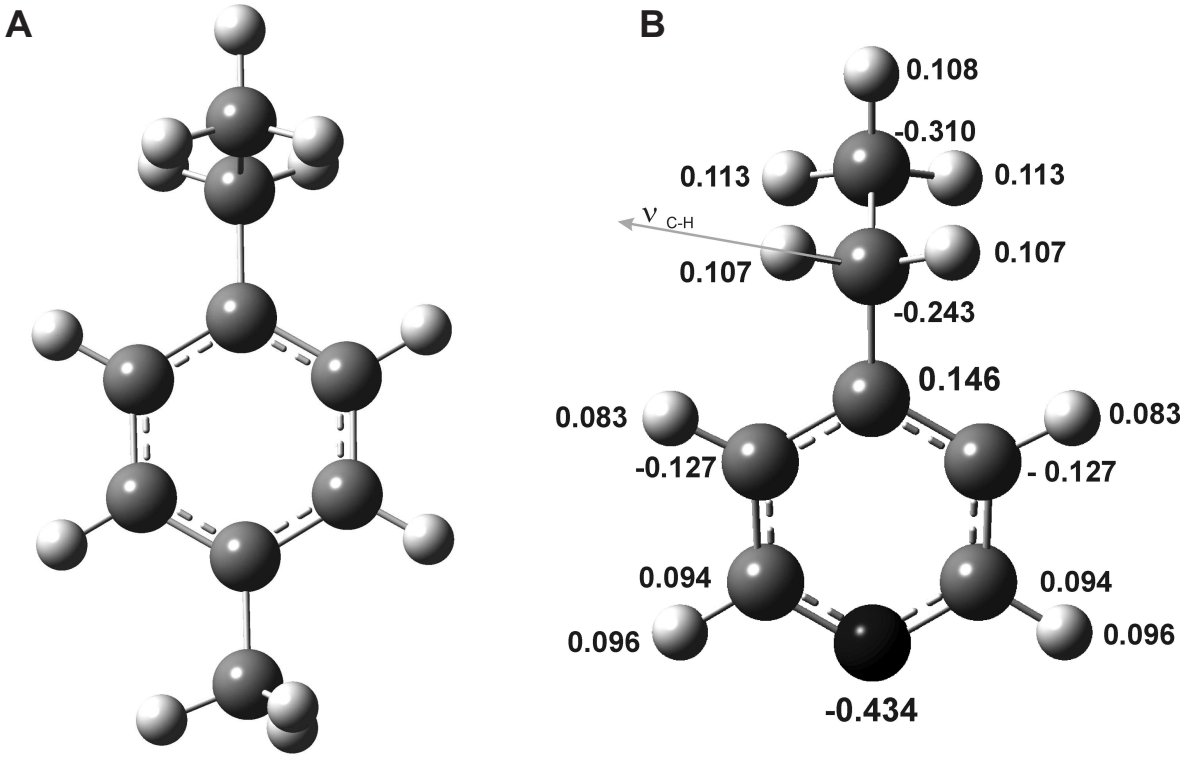


Fig. 5. (**A**) An example of topologic encoding: number of substituents: 2; number of heavy atoms in the longest substituent: 2; number of heavy atoms in all substituents: 3; localization of substituents (*para*, *meta*, and *ortho*) in the aromatic ring in relation to the ethyl group: (1,0,0); (**B**) Mulliken charge analysis of 4-ethylpyridine (highest charge: 0.146; and lowest charge: –0.434) and the vibration mode of C-H bond stretching [60]

Calculation of descriptors

Descriptors were calculated in Gaussian 09 [17] according to the procedure provided by Szaleniec et al. [59]. For each compound, the following quantum-chemical parameters were computed: the partial α-carbon charge, the highest and the lowest atomic charges, the difference between the highest (positive) and the lowest (negative) charge in both Mulliken and NBO analyses, the ν_sC-H symmetrical stretching frequency, the H-NMR shift of substituted hydrogen, the C-NMR shift of α-carbon, the dipole moment μ, the frontier orbital energies, i.e., E_{LUMO} (as an approximation of electron affinity) and E_{HOMO} (as an approximate measure of ionization potential) [43] and GAP – the energy difference between E_{LUMO} and E_{HOMO} (as a measure of absolute hardness) [40] (Fig. 5B). The substituents' influence on the relative changes in the Gibbs free enthalpy of radical and carbocation formation ($\Delta\Delta G^{radical}$ and $\Delta\Delta G^{carbocation}$) was calculated according to previously described methods [58]. Moreover, a computational alternative to Hammet σ, an electrostatic potential at the nuclei, was calculated according to the procedure described by Galabov et al. [16]. A compound's bulkiness was described by the molecular weight (MW), the total energy of the molecule (SCF) and the zero point energy (ZPE) (derived from DFT calculations), as well as the molecular surface area (Mol_Surf), molecular fractional polar surface area (Pol_Mol_Surf) and molecular refractivity (MR) (calculated in the Accerlys Discovery Studio 2.5). Moreover, the logP and logP_MR (hydrophobicity) and the Kier and Hall molecular connectivity indices χ^0, χ^1, and χ^2 [28, 29] as well as the Balaban [4] and Zagreb indices [5] (topology), were also calculated in Accerlys Discovery Studio 2.5. The topology descriptors were supplemented with the lengths of substituents, their mutual locations (*ortho*, *meta*, *para*), the number of heavy atoms in all substituents and the number of heavy atoms in the longest substituent, taking the structure of ethylbenzene as a reference core (see Fig. 5A for an example).

Selection of variables

The input vector was optimized with three of the methods described above:

– removal of redundant variables;

– genetic algorithm optimization in the linear regime (performed in Accelrys Material Studio 4.4);

– brute force experimental optimization by Intelligent Problem Solver (IPS in Statistica 7.0, Statistica Neural Networks package).

The initial removal of the redundant variables was based on the correlation matrix. This process resulted in the elimination of χ^1 and χ^2, topological indexes that correlated with χ^0 with $R > 0.95$, as well as logP_MR, which was highly co-linear with logP. These descriptors were not considered in further optimization.

The input vector optimized by GA was set to predict log rSA only for the substrate, as there are no linear relationships of descriptors with both substrates and inhibitor activities. Moreover, the logarithmic scale of rSA makes the pursuit relation more linear, thus rendering the linear GA method more appropriate for the selection of variables for the ANN.

The GA population comprised 250 equations with 5, 8 and 10 terms that were optimized through 2,500 generations. The 15 most frequently used descriptors that occurred in all three optimizations were collected together (reducing the number of descriptors from 34 to 13) and subjected to a subsequent IPS optimization of the neural network architecture (reducing the number of descriptors from 13 to 11). To compare the robustness of the variable selection protocols, the IPS was set to optimize the initial population of variables with rSA as a dependent variable. Therefore, both inhibitors and substrates could be used as cases in the optimization protocol. IPS optimized both the input vector and the number of hidden neurons. In each IPS optimization run, 2500 networks were constructed, and the 50 best models were retained. The selection of models was conducted based on the lowest learning and testing error and on the correlation of the predicted values with the experimental data.

Data partitioning

The selection of validation and test cases was achieved with k-means neighbor clustering. A total of 12 clusters were selected, based on all of the calculated descriptors additionally supplemented with atom-based extended-connectivity fingerprints (ECFC_6, ECFC_8 and ECFC_12). The hold-out subgroup (26%) comprised 12 compounds that were randomly selected from each cluster that contained more than one representative. From this subset, 4 additional cases were removed to form a validation set that was used for external cross-validation of the model. The representativeness of the selected test and validation subsets was verified with PCA analysis on the 3D graph of the first three principal components. The selected points were determined to be randomly scattered among the rest of the points and were not outliers of the main cloud of the training cases.

Results

Two regression models were obtained, one that was generated by IPS from GA-optimized input variables (model 1) and another that was obtained using only the IPS brute force optimization protocol (model 2).

The obtained GA-ANN model had 11 input neurons, 6 hidden neurons and one logistic output neuron (MLP 11-6-1). This model was trained with 100 epochs of back-propagation algorithm, followed by 20 epochs of conjunct gradient and 155 epochs of conjunct gradient with momentum. It exhibited comparable errors in all subsets (see Tab. 2) and a relatively low mean absolute prediction error (8.56%).

The IPS protocol not preceded by GA optimization produced an overall worse population of models (lower prediction correlations). It contained either very complex models that could not be accepted due to the small number of training cases or very simple ones that exhibited fundamental errors even among the training cases. However, it was possible to manually modify one of the moderately sized models, which, after optimization of the hidden layer, yielded satisfactory results. Model 2 is an MLP 12-7-1 neural network that exhibits slightly higher test and validation mean absolute prediction errors than model 1. It was trained with 50 epochs of quick-propagation algorithm with added Gaussian noise and 1428 epochs of BFGS quasi-Newton-based algorithm [6] with momentum. Nevertheless, both models achieve reasonably accurate prediction results (see Tab. 1). The analysis of scatter plots (Fig. 6) shows that both models are able to discern between inhibitors and substrates of the enzyme. They are also able to predict fairly well

Tab. 2. Statistical summary of the obtained models

Model	R^2	Test R^2	Validation R^2	Mean pred. error	Learning error	Test error	Validation error
Model 1 MLP 11-6-1	0.984	0.957	0.996	8.56%	0.0174	0.0396	0.0654
Model 2 MLP 12-7-1	0.929	0.953	0.874	23.33%	0.0545	0.0856	0.0970

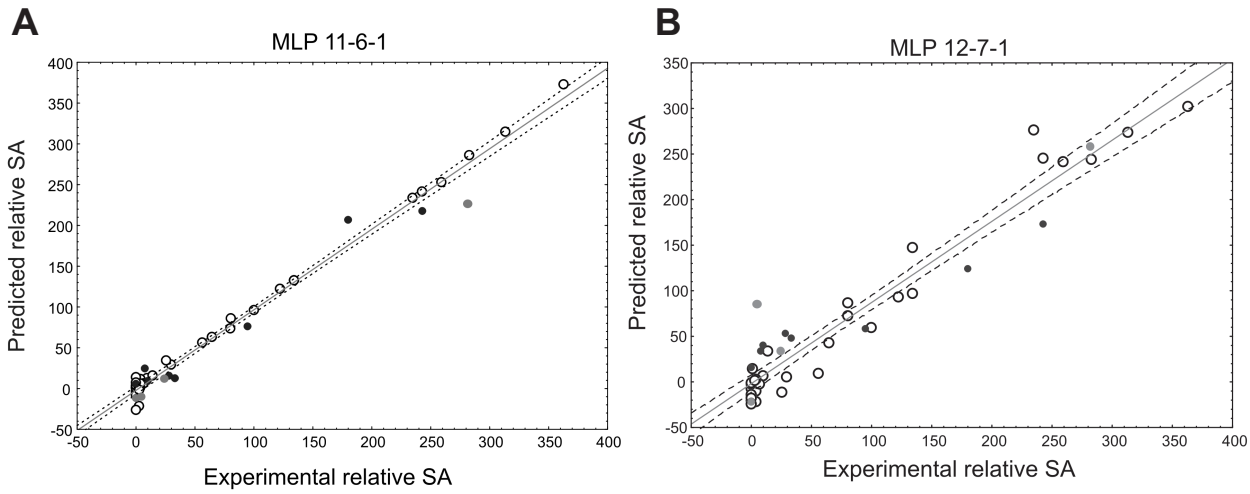


Fig. 6. Scatter plots of the experimental relative specific activity (rSA) and the predicted rSA for (**A**) model 1 MLP 11-6-1 network (R^2 = 0.986).and (**B**) model 2 MLP 12-7-1 (R^2 = 0.929). The full circles depict test and validation cases. Empty circles – training cases; black circles – test cases; gray circles – validation cases; the dashed line depicts the 95% confidence interval of the model

the extent of activity for the test and validation cases, although model 1 exhibits markedly better performance. It should be underlined here that the obtained models are far more complex than advised by the rule derived by Andrea et al. However, rigorous model cross-validation with test and validation cases demonstrates their high generalization capabilities. This finding again proves that the optimal ρ interval is a problem-dependent issue.

Interpretation

Very often, an understanding of the factors influencing enzyme catalysis is more important than even the best prediction tool. The interpretation of ANN models, although not so straightforward as in the case of MLR, is possible by two means: i) sensitivity analysis and ii) response curves and surfaces. Sensitivity analysis (see Tab. 3) provides a ranking of the variables used in the model [19]. However, as our example has shown, ANNs are able to achieve similar prediction levels by different approaches. Therefore, it is advisable to perform a sensitivity analysis on a group of models and draw conclusions from a repetitive occurrence of the important variables.

In our example, both models suggest that occupation of the *para* position is one of the most important factors, as is a substituent's ability to stabilize the carbocation form. Additionally, the energy of the HOMO orbital and $\Delta\Delta G^{radical}$ are ranked comparably high.

More information can be gained from the response curves that depict the dependent variable (here rSA) as a function of an individual selected descriptor

Tab. 3. The results of sensitivity analysis. The ranks provide the descriptor's order of importance in the model

Model 1		Model 2	
Descriptor	Rank	Descriptor	Rank
$\Delta\Delta G^{carbocation}$	1	para	1
para	2	$\Delta\Delta G^{carbocation}$	2
ZPE	3	$\Delta\Delta G^{radical}$	3
E_{HOMO}	4	ortho	4
$\Delta\Delta G^{radical}$	5	E_{HOMO}	5
χ^0	6	GAP	6
EPN	7	νC-H	7
q^{min} NBO	8	Number of heavy atoms in the longest substituent	8
Δq NBO	9	E_{LUMO}	9
C-NMR	10	Pol_Mol_Surf	10
Mull. q^{meth}	11	q^{min} NBO	11
		EPN	12

(typically holding the remainder of the input at the average value, although various other approaches have also been used [2]). As a result, one obtains a 2D slice through the multidimensional response hypersurface. In our case, one can analyze the influence of $\Delta\Delta G^{carbocation}$ and $\Delta\Delta G^{radical}$ on the rSA. Both models provide non-linear relationships between rSA, and both descriptors predict high values of rSA for negative (high stabilization) values of $\Delta\Delta G^{carbocation}$ and $\Delta\Delta G^{radical}$ (Fig. 7A, B, E, F). The influence of a steric effect can be analyzed in the same way for ZPE in the case of model 1 (Fig. 7C) and the 'number of heavy atoms in the longest substituent' descriptor in the case of model 2 (Fig. 7G). In both cases, low activity is predicted for high values of the descriptors, which indicates a negative steric effect on the specific activity. The influence of hydrophobicity can be analyzed by following the response of the difference between the maximum and minimum partial charges (Δq NBO – Fig. 7D) or the fractional polar surface (Fig. 7H). In both cases, the obtained relationships indicate that non-polar, ethylbenzene-like compounds will be of a higher activity. It should also be noted that, as these two variables are less important than the top-ranked one (e.g., $\Delta\Delta G^{carbocation}$), changes in their values exert a smaller effect on rSA.

One should be aware however, that the analysis of response curves might sometimes be misleading, as the 2D projection of ANN behavior is far from a real situation. In a real compound, one can hardly change $\Delta\Delta G^{carbocation}$ without changing $\Delta\Delta G^{radical}$. A glimpse of the complexity of the true response hypersurface can be gained through an analysis of the 3D response surfaces (Fig. 8), which shows that simple conclusions drawn from response curves might not be entirely true for the whole range of values. The figure predicts a high activity for compounds with a high stabilization of the carbocation and low stabilization of the radical, which appears chemically counter-intuitive. Therefore, it is always advisable to study ANN models with real chemical probes – i.e., design *in silico* a new compound with the desired characteristics and use the obtained model for prediction of the activity.

Nevertheless, all of these findings appear to be in accordance with our recent results on the EBDH reaction mechanism. Both the radical (TS1) and the carbocation (TS2) barriers are of similar height and therefore, influence the observed kinetics. Moreover, the enzyme has evolved to host non-polar hydrocarbons, such as ethylbenzene and propylbenzene, which explains why less polar compounds are favored in the reactivity. However, the electronic stabilization of the transition states introduced by heteroatoms, especially in a resonance-conjunct *para* position, appears to overcome the polarity problems. This situation results in a specific activity higher than that of ethylbenzene for compounds such as 4-ethylphenol or 4-ethylresorcine.

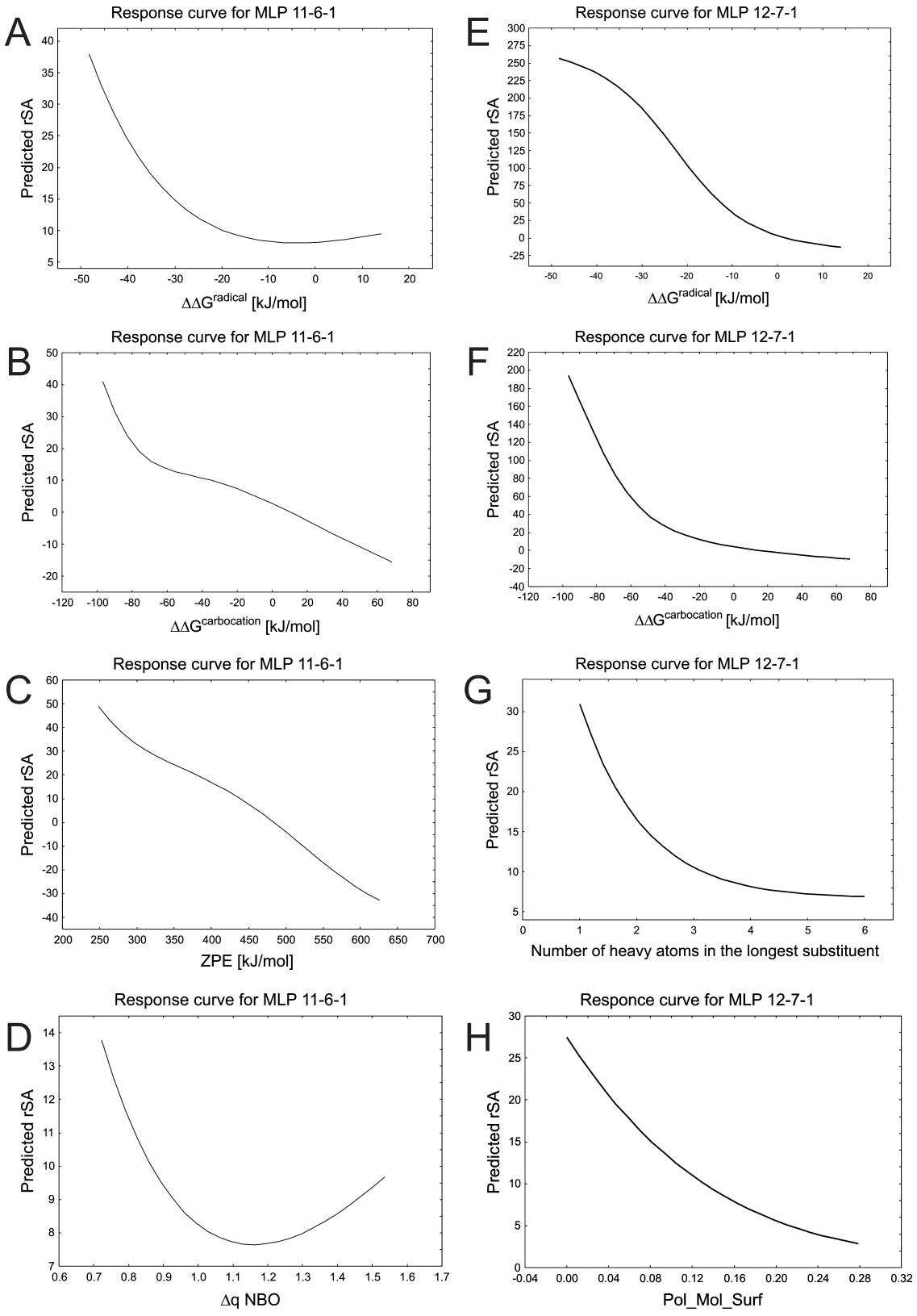


Fig. 7. The response curve for MLP 11-6-1 relative Specific Activity as a function of (**A**) $\Delta\Delta G^{radical}$, (**B**) $\Delta\Delta G^{carbocation}$, (**C**) ZPE and (**D**) DNBO; and for MLP 12-7-1 (**E**) $\Delta\Delta G^{radical}$, (**F**) $\Delta\Delta G^{carbocation}$, (**G**) number of heavy atoms in the longest substituent, and **H**) molecular fractional polar surface area

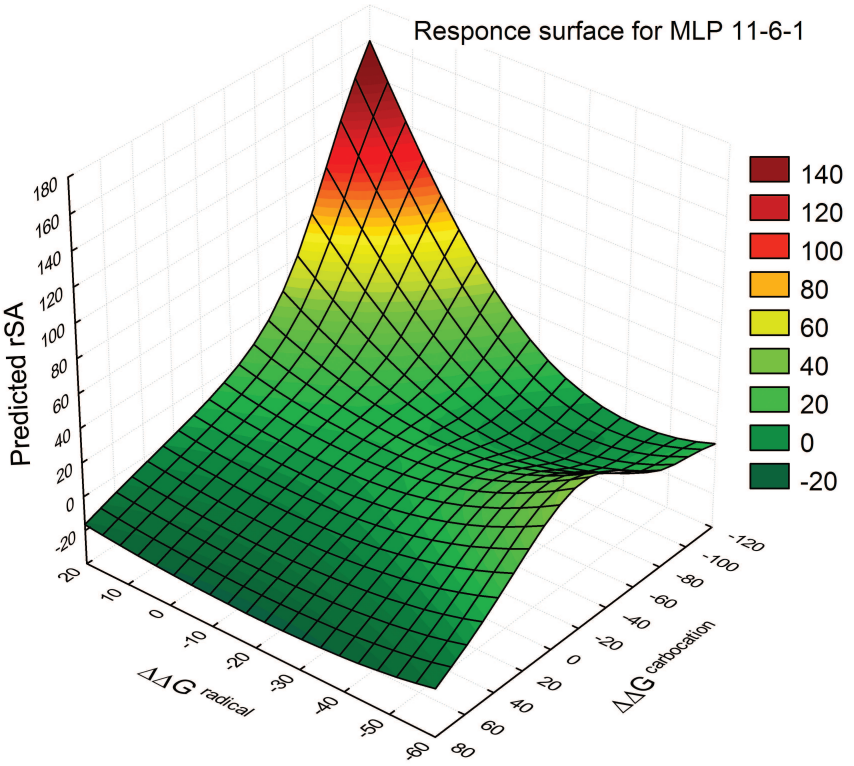


Fig. 8. The response surface obtained for MLP 11-6-1 with $\Delta\Delta G^{radical}$ and $\Delta\Delta G^{carbocation}$ as independent variables

Conclusion

Neural networks are extremely powerful modeling tools. Each year, more and more studies are published that utilize ANNs in various fields of science [63]. It is surprising that neural networks receive relatively little attention in the research on enzyme activity. However, they are increasingly applied in drug design, as huge databases are abundant in that field of science.

Acknowledgments:
The author acknowledges the financial support of the Polish Ministry of Science and Higher Education under the grant N N204 269038 and the computational grant KBN/SGI2800/PAN/037/2003. The author thanks Katalin Nadassy from Accelrys for her support in the development of the cluster-based data-splitting protocol.

References:

1. Abdul Rahman MB, Chaibakhsh N, Basri M, Salleh AB, Abdul Rahman RN: Application of artificial neural network for yield prediction of lipase-catalyzed synthesis of dioctyl adipate. Appl Biochem Biotechnol, 2009, 158, 722–735.
2. Andrea TA, Kalayeh H: Applications of neural networks in quantitative structure-activity relationships of dihydrofolate reductase inhibitors. J Med Chem, 1991, 34, 2824–2836.
3. Bakken GA, Jurs PC: QSARs for 6-azasteroids as inhibitors of human type 1 5α-reductase: prediction of binding affinity and selectivity relative to 3-BHSD. J Chem Inf Comput Sci, 2001, 41, 1255–1265.
4. Balaban AT, Ivanciuc O: FORTRAN77 computer program for calculating the topological index J for molecules containing heteroatoms. In: Studies in Physical and Theoretical Chemistry, Ed. Graovac A. Elsevier, Amsterdam,p. 193–211, 1989.
5. Bonchev D: Information theoretic indices for characterization of chemical structures. Chemometrics Series, Vol. 5, Ed. Bawden DD, Research Studies Press Ltd., New York, 1983.
6. Broyden CG: The convergence of a class of double-rank minimization algorithms 1. General considerations. IMA J Appl Math, 1970, 6, 76–90.
7. Burden FR: Robust QSAR models using bayesian regularized neural networks. J Med Chem, 1999, 42, 3183–3187.
8. Cartwright H: Using Artificial Intelligence in Chemistry and Biology. A Practical Guide. CRC Press, Taylor & Francis Group, London, 2008.
9. Chiu T-L, So S-S: Development of neural network QSPR models for hansch substituent constants. 2. Applications in QSAR studies of HIV-1 reverse transcriptase and dihydrofolate reductase inhibitors. J Chem Inf Comput Sci, 2003, 44, 154–160.

10. Duch W, Swaminathan K, Meller J: Artificial intelligence approaches for rational drug design and discovery. Curr Pharm Des 2007, 13, 1497–1508.
11. Dudek-Dyduch E, Tadeusiewicz R, Horzyk A: Neural network adaptation process effectiveness dependent of constant training data availability. Neurocomputing, 2009, 72, 3138–3149.
12. Fausett L, Fundamentals of Neural Networks: architecture, algorithms, and applications. Prentice-Hall Inc., Englewood Cliffs, New Jersay., 1994.
13. Fernandez M, Caballero J, Helguera AM, Castro EA, Gonzalez MP: Quantitative structure-activity relationship to predict differential inhibition of aldose reductase by flavonoid compounds. Bioorg Med Chem, 2005, 13, 3269–3277.
14. Friedman JH: Multivariate adaptive regression splines. Ann Statist 1991, 19, 1–67.
15. Funar-Timofei S, Suzuki T, Paier JA, Steinreiber A, Faber K, Fabian WM: Quantitative structure-activity relationships for the enantioselectivity of oxirane ring-opening catalyzed by epoxide hydrolases. J Chem Inf Comput Sci, 2003, 43, 934–940.
16. Galabov B, Cheshmedzhieva D, Ilieva S, Hadjieva B: Computational study of the reactivity of n-phenylacetamides in the alkaline hydrolysis reaction. J Phys Chem A, 2004, 108, 11457–11462.
17. Gaussian 09 Revision A.01, Frisch MJ, Trucks GW, Schlegel HB, Scuseria GE, Robb MA, Cheeseman JR et al.: Gaussian, Inc., Wallingford CT, 2009.
18. González MP, Caballero J, Tundidor-Camba A, Helguera AM, Fernández M: Modeling of farnesyltransferase inhibition by some thiol and non-thiol peptidomimetic inhibitors using genetic neural networks and RDF approaches. Bioorg Med Chem, 2006, 14, 200–213.
19. Guha R, Jurs PC: Interpreting computational neural network QSAR models: a measure of descriptor importance. J Chem Inf Model, 2005, 45, 800–806.
20. Hansch C: Quantitative approach to biochemical structure-activity relationships. Acc Chem Res, 1969, 2, 232–239.
21. Hansch C, Maloney PP, Fujita T, Muir RM: Correlation of biological activity of phenoxyacetic acids with Hammett substitution constants and partition coefficients. Nature, 1962, 194, 178–180.
22. Hirst JD, King RD, Sternberg MJE: Quantitative structure-activity relationships by neural networks and inductive logic programming. I. The inhibition of dihydrofolate reductase by pyrimidines. J Comput Aided Mol Des, 1994, 8, 405–420.
23. Hirst JD, King RD, Sternberg MJE: Quantitative structure-activity relationships by neural networks and inductive logic programming. II. The inhibition of dihydrofolate reductase by triazines. J Comput Aided Mol Des, 1994, 8, 421–432.
24. Jalali-Heravi M, Asadollahi-Baboli M, Shahbazikhah P: QSAR study of heparanase inhibitors activity using artificial neural networks and Levenberg-Marquardt algorithm. Eur J Med Chem, 2008, 43, 548–556.
25. Jalali-Heravi M, Parastar F: Use of artificial neural networks in a QSAR study of anti-HIV Activity for a large group of HEPT derivatives. J Chem Inf Comput Sci, 2000, 40, 147–154.
26. Johnson HA, Pelletier DA, Spormann AM: Isolation and characterization of anaerobic ethylbenzene dehydrogenase, a novel Mo-Fe-S enzyme. J Bacteriol, 2001, 183, 4536–4542.
27. Kauffman GW, Jurs PC: QSAR and k-nearest neighbor classification analysis of selective cyclooxygenase-2 inhibitors using topologically-based numerical descriptors. J Chem Inf Comput Sci, 2001, 41, 1553–1560.
28. Kier LB, Hall LH, Molecular connectivity indices in chemistry and drug research. Ed. de Stevens G, Academic Press, New York, 1976.
29. Kier LB, Hall LH: Molecular connectivity in structure-activity analysis, Chemometrics Series, Ed. Bawden DD, Vol. 9., Research Studies Press Ltd., New York, 1985.
30. Kniemeyer O, Heider J: Ethylbenzene dehydrogenase, a novel hydrocarbon-oxidizing molybdenum/iron-sulfur/heme enzyme. J Biol Chem, 2001, 276, 21381–21386.
31. Kohonen T: Self-organized formation of topologically correct feature maps. Biol Cybern, 1982, 43, 59–69.
32. Linko S, Zhu Y-H, Linko P: Applying neural networks as software sensors for enzyme engineering. Trends Biotechnol, 1999, 17, 155–162.
33. Lučić B, Nadramija D, Bašic I, Trinajstić N: Toward generating simpler QSAR models: nonlinear multivariate regression versus several neural network ensembles and some related methods. J Chem Inf Comput Sci, 2003, 43, 1094–1102.
34. Mattioni BE, Jurs PC: Development of quantitative structure-activity relationship and classification models for a set of carbonic anhydrase inhibitors. J Chem Inf Comput Sci, 2002, 42, 94–102.
35. McElroy NR, Jurs PC, Morisseau C, Hammock BD: QSAR and classification of murine and human soluble epoxide hydrolase inhibition by urea-like compounds. J Med Chem, 2003, 46, 1066–1080.
36. Michielan L, Moro S: Pharmaceutical perspectives of nonlinear QSAR strategies. J Chem Inf Model, 2010, 50, 961–978.
37. Mosier PD, Jurs PC: QSAR/QSPR studies using probabilistic neural networks and generalized regression neural networks. J Chem Inf Comput Sci, 2002, 42, 1460–1470.
38. Niwa T: Prediction of biological targets using probabilistic neural networks and atom-type descriptors. J Med Chem, 2004, 47, 2645–2650.
39. Novič M, Nikolovska-Coleska Z, Solmajer T: Quantitative structure-activity relationship of flavonoid p56lck protein tyrosine kinase inhibitors. A neural network approach. J Chem Inf Comput Sci, 1997, 37, 990–998.
40. Orzel L, Kania A, Rutkowska-Zbik D, Susz A, Stochel G, Fiedor L: Structural and electronic effects in the metalation of porphyrinoids. Theory and experiment. Inorg Chem, 2010, 49, 7362–7371.
41. Patterson D: Artificial Neural Networks, Prentice Hall, Singapore, 1996.
42. Polley MJ, Winkler DA, Burden FR: Broad-based quantitative structure-activity relationship modeling of potency and selectivity of farnesyltransferase inhibitors

using a Bayesian Regularized Neural Network. J Med Chem, 2004, 47, 6230–6238.
43. Rodakiewicz-Nowak J, Nowak P, Rutkowska-Zbik D, Ptaszek M, Michalski O, Mynarczuk G, Eilmes J: Spectral and electrochemical characterization of dibenzotetraaza[14]annulenes. Supramol Chem, 2005, 17, 643–647.
44. Rogers D, Hopfinger AJ: Application of genetic function approximation to quantitative structure-activity relationships and quantitative structure-property relationships. J Chem Inf Comput Sci, 1994, 34, 854–866.
45. Rutkowski L, Scherer R, Tadeusiewicz R, Zadeh LA, Zurda JM: Eds. Artificial Intelligence and Soft Computing. Springer-Verlag: Berlin, 2010.
46. Silva J, Costa Neto E, Adriano W, Ferreira A, Gonçalves L: Use of neural networks in the mathematical modelling of the enzymic synthesis of amoxicillin catalysed by penicillin G acylase immobilized in chitosan. World J Microbiol Biotechnol, 2008, 24, 1761–1767.
47. So S-S, Karplus M: Evolutionary optimization in quantitative structure-activity relationship: An application of genetic neural networks. J Med Chem, 1996, 39, 1521–1530.
48. So S-S, Karplus M: Genetic neural networks for quantitative structure-activity relationships: Improvements and application of benzodiazepine affinity for benzodiazepine/$GABA_A$ receptors. J Med Chem, 1996, 39, 5246–5256.
49. So S-S, Karplus M: Three-dimensional quantitative structure-activity relationships from molecular similarity matrices and genetic neural networks. 1. Method and validations. J Med Chem, 1997, 40, 4347–4359.
50. So SS, Richards WG: Application of neural networks: quantitative structure-activity relationships of the derivatives of 2,4-diamino-5-(substituted-benzyl)pyrimidines as DHFR inhibitors. J Med Chem, 1992, 35, 3201–3207.
51. Sutherland JJ, O'Brien LA, Weaver DF: Spline-fitting with a genetic algorithm: a method for developing classification structure-activity relationships. J Chem Inf Comput Sci, 2003, 43, 1906–1915.
52. Szaleniec M, Borowski T, Schuhle K, Witko M, Heider J: Ab inito modeling of ethylbenzene dehydrogenase reaction mechanism. J Am Chem Soc, 2010, 132, 6014–6024.
53. Szaleniec M, Dudzik A, Pawul M, Kozik B: Quantitative structure enantioselective retention relationship for high-performance liquid chromatography chiral separation of 1-phenylethanol derivatives. J Chromatogr A, 2009, 1216, 6224–6235.
54. Szaleniec M, Hagel C, Menke M, Nowak P, Witko M, Heider J: Kinetics and mechanism of oxygen-independent hydrocarbon hydroxylation by ethylbenzene dehydrogenase. Biochemistry, 2007, 46, 7637–7646.
55. Szaleniec M, Salwiński A, Borowski T, Heider J, Witko M: Quantum chemical modeling studies of ethylbenzene dehydrogenase activity. Int J Quantum Chem, 2012, 112, 1990–1999.
56. Szaleniec M, Tadeusiewicz R, Skoczowski A: Optimization of neural models based on the example of assessment of biological activity of chemical compounds. Comput Methods Mater Sci, 2006, 6, 65–80.
57. Szaleniec M, Tadeusiewicz R, Witko M: How to select an optimal neural model of chemical reactivity? Neurocomputing, 2008, 72, 241–256.
58. Szaleniec M, Witko M, Heider J: Quantum chemical modelling of the C–H cleavage mechanism in oxidation of ethylbenzene and its derivates by ethylbenzene dehydrogenase. J Mol Catal A Chem, 2008, 286, 128–136.
59. Szaleniec M, Witko M, Tadeusiewicz R, Goclon J: Application of artificial neural networks and DFT-based parameters for prediction of reaction kinetics of ethylbenzene dehydrogenase. J Comput Aided Mol Des, 2006, 20, 145–157.
60. Tadeusiewicz R: Neural Networks (Polish). Akademicka Oficyna Wydawnicza, Warszawa, 1993.
61. Tadeusiewicz R: Using Neural Models for Evaluation of Biological Activity of Selected Chemical Compounds, in Applications of Computational Intelligence in Biology, Current Trends and Open Problems, Studies in Computational Intelligence. Springer-Verlag: Berlin – Heidelberg – New York, 2008, 135–159.
62. Tadeusiewicz R: Neural network as a tool for medical signals filtering, diagnosis aid, therapy assistance and forecasting improving, in image processing, biosignals processing, modelling and simulation, biomechanics. IFMBE Proceedings, Eds. Dössel O, Schlegel WC, Springer Verlag, Berlin, Heidelberg, New York, 2009, 1532–1534.
63. Tadeusiewicz R: New trends in neurocybernetics. Comput Methods Mater Sci, 2010, 10, 1–7.
64. Tadeusiewicz R, Grabska-Chrząstowska J, Polcik H: Attempt of neural modelling of castings crystallization control process. Comput Methods Mater Sci, 2008, 8, 58–69.
65. Tetko IV, Livingstone DJ, Luik AI: Neural network studies. 1. Comparison of overfitting and overtraining. J Chem Inf Comput Sci, 1995, 35, 826–833.
66. Topliss JG, Edwards RP: Chance factors in studies of quantitative structure-activity relationships. J Med Chem, 1979, 22, 1238–1244.
67. Winkler D: Neural networks as robust tools in drug lead discovery and development. Mol Biotechnol, 2004, 27, 139–167.
68. Yasri A, Hartsough D: Toward an optimal procedure for variable selection and QSAR model building. J Chem Inf Comput Sci, 2001, 41, 1218–1227.
69. Yu XH, Chen GA, Cheng SX: Dynamic learning rate optimization of the backpropagation algorithm. IEEE Trans Neural Netw, 1995, 6, 669–677.
70. Zernov VV, Balakin KV, Ivaschenko AA, Savchuk NP, Pletnev IV: Drug discovery using support vector machines. The case studies of drug-likeness, agrochemical-likeness, and enzyme inhibition predictions. J Chem Inf Comput Sci, 2003, 43, 2048–2056.

Received: February 7, 2012; **in the revised form:** April 2, 2012; **accepted:** April 16, 2012.