# Chapter 9

# Interpreting Computational Neural Network QSAR Models: A Detailed Interpretation of the Weights and Biases

## 9.1 Introduction

As we have seen in the preceding chapters, interpretability plays an important role in the QAR modeling process. The statistical and machine learning literature provide a wide variety of modeling methods to choose from, ranging from linear regression models to more complex techniques such as neural networks and random forests. The modeling techniques differ in a number of ways such as complexity, flexibility, accuracy and speed. A very important aspect of these models is interpretability. In the absence of an interpretation, the model can be used only for predictive purposes. This implies that structure-property information encoded in the model is not further utilized. In many cases, such as high throughput screens, such usage of the model is sufficient. But when models are developed with the aim of providing input to structure based drug design, more detailed information than just predicted values must be extracted from the model. That is, one would like to know what structure-property trends have been captured by the model. In other words, we would like to understand how the model correlates the input descriptors to the predicted activity. Furthermore, some measure of interpretability is needed to provide a sense of confidence regarding the soundness of the model, and it would provide evidence to support the use of a particular model in a given instance.

The degree of interpretability of QSAR models varies, depending on the modeling technique. In some cases, such as linear regression models, interpretation is relatively simple and can be carried out using a PLS technique developed by Stanton[1] and described in Chapter 3. In this case, the interpretation is detailed in the sense that one can extract information about how individual descriptors correlate to the predicted property. A

This work was published as Guha, R.; Jurs, P.C., "Interpreting Computational Neural Network QSAR Models: A Detailed Interpretation of the Weights and Biases", *J. Chem. Inf. Model.*, **2005**, *ASAP*.

number of applications of this technique have been reported.[1–3] In other models, the interpretation is not as detailed. This is the case for random forest[4] models. For these types of models interpretation is restricted to a summary of descriptor importance.[5] This summary ranks descriptors in order of importance to predictive ability. Thus, one does not get a detailed view of how the descriptors contribute to the predicted property.

The high predictive ability and flexibility of CNN models have made them very attractive to QSAR modelers. However the lack of interpretability has led to the general characterization of CNN models as *black boxes*. A number of attempts to extract information regarding the internal working of CNN models have been described. In some cases these methods are in the form of rule extractions.[6–8] These methods can be heuristic[9–11] in nature or analytical.[12] A number of these methods are focussed on specific types of neural networks.[8,9,13] Chastrette et al.[13] describe a method for interpreting a CNN model describing structure-musk odor relationships. Their approach was limited to a measure of contribution of the descriptors to the predicted value. Hervás et al.[14] describe a method interpretation that is focussed on a pruning algorithm. As a result the method is not applicable to CNN models developed using alternative algorithms.

The analysis of descriptor contributions is an approach that has been followed. Some of these approaches, such as that described by Chastrette et al.[13] provide only a broad view of which descriptors are important. Other approaches, however, have been devised that allow for a measure of correlation between input descriptors and the network output. An example is the method described by Mak et al.,[15] in which a form for the relative contribution of input neurons to the output value is developed. The relative values are then divided to obtain a measure of contribution to each hidden layer neuron. The result of this approach is that the contributions of the input neurons can be divided into negative or positive contributions. Chapter 8 described a method to obtain a measure of descriptor importance for neural network models based on a sensitivity analysis[16] of a trained network. Though similar in intent to the methods described by Chastrette et al. and Mak et al., the method provides an easily visualization method to understand which descriptors play the main role in the models predictive ability. However, the method also shares the main shortcoming with other approaches to measure descriptor importance (or contributions) in that it provides a very broad view and is not capable of describing in detail, the nature of the correlation between a given descriptor and the network output.

In this chapter we describe a method to interpret a CNN model in a detailed manner by considering the final, optimized weights and biases. As a result of this approach, the method is generalizable to different types of CNN algorithms that result in a set of weights and biases. Currently the method is restricted to the interpretation of 3-layer, feed-forward networks, though extension to more hidden layers is possible. The methodology is similar in concept to the PLS technique in that it interprets the weight matrix in a manner analogous to the interpretation of the X-weights in the PLS analysis. The method also shares certain characteristics with the method described by Mak et al. The next section describes the methodology in detail.

## 9.2 Methodology

The detailed CNN interpretation methodology was developed by attempting to mimic the procedure used for the interpretation of linear models using partial least squares. Though we have described the PLS methodology in detail in Chapter 3, we provide a short summary below.

The descriptors for the linear model are used to build a PLS model using a leave-one-out cross-validation method. The PLS model consists of a number of latent variables (components) which are linear combinations of the original descriptors. The number of components is equal to the number of input descriptors (assuming no overfitting has occurred). The results of the PLS analysis are summarized by two tables. The first table tabulates the cumulative variance and $Q^2$ values for each component. In many cases the first few components explain a large portion of the total variance (70% - 90%). As a result, the remaining components can be ignored. The second table lists the X-weights for each component. These are the weights used to linearly combine each input descriptor in a given component. Analysis of these weights allows one to understand how significantly, and in which direction, a given descriptor is correlated to the value predicted by that component. Finally, using plots of X-scores (projections of the observations along the rotated axes) versus Y-scores (that portion of the observed Y that is explained by that component) one can focus on the correlations between structural features and property for specific molecules.

### 9.2.1 Preliminaries

The CNN interpretation method is based on the assumption that the hidden layer neurons are analogous to the latent variables in a PLS model. Clearly, this is not a one-to-one correspondence due to the sigmoidal transfer function employed for each neuron in the CNN. By considering the weights connecting the input descriptors to a specific hidden layer neuron, we can then interpret how each descriptor correlates to the output of that hidden layer neuron. Finally, by defining the contribution of each hidden layer neuron to the output value of the network, we can determine which hidden layer neurons are the most significant and which ones can be ignored. The problem of interpreting a CNN model involves understanding how the output value of the network varies with the input values. This in turn is dependent on how the weights and biases modify the input values as they pass through the layers on the network. First, we present a brief analysis of how the input values will, in general, relate to the output value. We restrict ourselves to a 3-layer, fully-connected, feed-forward network.

The output value of a CNN for a given set of input values is obtained via a sigmoidal transfer function. Thus we can write the output value, $O$, as

$$O = \frac{1}{1 + \exp(-X)} \tag{9.1}$$

where $X$ is the sum of weighted outputs from the hidden layer neurons. Denoting the output of each hidden layer neuron by $x_j^H,\ 1 \leq j \leq n_H$, where $n_H$ is the number of hidden layer neurons, and the weight between each hidden layer neuron and the output neuron as $w_j^H,\ 1 \leq j \leq n_H$, we can write $X$ as,

$$X = \sum_{j=1}^{n_H} w_j^H x_j^H$$

The above equation does not include a bias term and we provide a justification for ignoring the bias term below. Eq. 9.1 can be rewritten as

$$\begin{aligned} O &= \frac{1}{1 + \exp(-\sum_{j=1}^{n_H} w_j^H x_j^H)} \\ \frac{1}{O} &\sim \exp(-\sum_{j=1}^{n_H} w_j^H x_j^H) \\ O &\sim \exp\left(w_1^H x_1^H + \cdots + w_{n_H}^H x_{n_H}^H\right) \end{aligned} \tag{9.2}$$

where we drop the constant term since it does not affect the general trend between the output value and exponential term. From Eq. 9.2 we can see that $O$ is a monotonic increasing function of the individual components, $w_j^h x_j^h$, of the argument. Keeping in mind that the output from each hidden neuron will be a positive number, Eq. 9.2 indicates that, if a certain hidden neuron has a large weight between itself and the output neuron, then the output from that hidden neuron will dominate the sum. This allows us to *order* the hidden neurons based on their contribution to the output value. Furthermore the sign of the weights indicate how the hidden neuron will affect the output value. Negative weights will correlate to smaller values of the output value and vice versa for positive weights.

### 9.2.2 Combining Weights

The above discussion applies to connections between the hidden layer and output layer. However it is clear that the same reasoning can be applied to the connections between the input and hidden layers. Thus, one way to consider the effect of the weights is to realise that the weights are cumulative. We denote the weights between the input layer neuron $j$ and the hidden layer neuron $i$ as

$$w_{ij},\ 1 \leq i \leq n_I \text{ and } 1 \leq j \leq n_H$$

where $n_I$ is the number of input layer neurons (i.e., descriptors). Now, let us consider the value of the first input descriptor (for a specific observation). As this value goes from the first input neuron to the first hidden layer neuron, it will be affected by the weight, $w_{11}$. The value from the first hidden neuron is passed to the output neuron and is affected by the weight $w_1^H$. Thus we can qualitatively say, that, as the input value passes from the input layer to the output layer, it is affected by a weight denoted by $w_{11}w_1^H$. This is because a large positive value of $w_{11}$ would cause the output of the first hidden neuron to be positively correlated with the first input descriptor. If $w_1^H$ is also a large positive weight then the final output value would be positively correlated with the output value of the first hidden neuron and thus would also be positively correlated with the value of the first input neuron. Thus, we can consider the network as consisting of a connection between the first input neuron and the output neuron weighted by an effective weight equal to $w_{11}w_1^H$.

Similarly, for the same input value passing through the second hidden neuron and then to the output neuron, we can write the corresponding effective weight as $w_{12}w_2^H$.

In general the effective weight between the $i^{\text{th}}$ input neuron and the output neuron, via the $j^{\text{th}}$ hidden layer neuron will be $w_{ij}w_j^H$. Clearly the effective weights are gross simplifications, since we neglect the intermediate summations (over all neurons in a layer) and transfer functions. However as shown in the previous section we can see that the output value of the transfer function is a monotonic increasing function of the product of the weights and neuron outputs. More importantly, our main interest is in the *sign* of the effective weight, rather than its absolute value. The absolute value of the weights between the hidden layer neurons and the output neuron might be one indication of which hidden neuron is more important that another in terms of contribution to the final output value. However as pointed out above, the sign of the weights indicates the trend of the output value. Thus for example if the weights $w_{11}$ and $w_1^H$ are both positive we can expect that input values flowing down that path will show a positive correlation with the output value. If $w_{11}$ and $w_1^H$ are positive and negative respectively, one would expect that the net effect would be a negative correlation between the input values and output values.

### 9.2.3 Interpreting Effective weights

We can now consider two possible ways to use the effective weights to interpret the behavior of the CNN. From the preceding discussion we can write the effective weights in tabular form as shown in Table 9.1, where H1, H2 and H3 represent the first, second and third hidden neurons and I1, I2, I3 and I4 represent the input neurons. The first step in interpreting the effective weight matrix is to decide the order of the hidden layer neurons, in terms of their contributions to the output value of the net. We discuss hidden layer neuron contributions in detail below and for now we assume that the order of importance of the hidden layer neurons is given by H1 > H2 > H3. Thus, the first hidden neuron is the main contributor to the output value. Next we consider the first column. If the value in a given row is higher than the others it implies that the corresponding input neuron will contribute more to the hidden layer neurons. Since we have already ordered the hidden neurons in terms of their contribution to the output, this means that we can say (indirectly) which input neuron is contributing more to the output. Furthermore, the sign of the element will indicate whether high values of that input neuron correspond to high or low values of the output value. The approach is similar to the PLS interpretation scheme, especially if we consider the hidden layer neurons to be a transformed (via the transfer function) set of latent variables.

### 9.2.4 The Bias Term

In the preceding discussion we have ignored the role of the bias term when considering effective weights. We now provide a justification for this approach. The input to a given hidden neuron consists of the weighted outputs of the input neurons plus the bias term. As a result, the effective weights for the input neurons, as they pass through a given hidden neuron, should consider the bias term. However, if we consider the effective weights for individual input neurons, we must partition the bias term between these input neurons. The simplest way of partitioning the bias term would be to simply divide the bias term evenly between the input neurons. As a result, for $n_I$ input neurons the effective weights between them and the $j^{\text{th}}$ hidden neuron would include $b_j/n_I$ where $b_j$ is the bias term for that hidden neuron. The net result is that for the $j^{\text{th}}$ hidden neuron, the effect of the bias term would be the same for all input neurons connected to it. As a result, it is equivalent to ignoring the bias term when considering effective weights. Clearly this is based on the assumption that the bias for a given hidden neuron can be equipartitioned between the input neurons. A priori, there is no reason for choosing an alternative partitioning scheme.

A more rigorous approach is to consider that fact that a bias term is effectively an intercept term. If the hidden neurons contained linear transfer functions, the bias term is precisely an intercept term. The inputs to a hidden neurons form a $p$-dimensional space and the result of the activation function for a hidden neuron is to draw a hyperplane through the input space. One side of this hyperplane represents the *off* output and the other side represents the *on* output. This description also holds for sigmoidal activation functions in which case the two sides of the hyperplane would correspond to the extreme ends of the functions domain. The region close to the hyperplane would correspond to the intermediate values of the activation function. In the absence of the bias term this hyperplane passes through the origin of the input space. However, when bias terms are included, they merely translate the hyperplanes from the origin. That is, they do not change the form of the hyperplane. In effect, the bias term is a constant. Now, one property of neural networks (more specifically, multi layer perceptrons) is universal function approximation.[17] For this to be true, the bias term must be included. However, it has been shown by Hornik[18] that a sufficient condition for this property to be true in the absence of bias terms is that the derivatives of the activation function must be non-zero at the origin. For a sigmoidal activation function, this implies that the bias term can simply be a constant value as opposed to a trainable weight. Clearly, if the

bias term can be considered as a constant (i.e., training is not required), this implies that it would not affect the interpretation of the optimized weights. Thus, viewing the bias terms in the context of partitioning or in the context of the universal approximation property indicates that development of the effective weights without including the bias terms is a valid approach.

### 9.2.5 Ranking Hidden Neurons

An important component of the interpretation method is the ranking of the hidden neurons, which is necessary as all hidden neurons will not contribute to the output value equally. The contributions of the input neurons to the output value, via those hidden neurons with lower contributions, will be diminished. A number of methods to determine the relative contribution of input neurons have been described in the literature. These methods can be applied to the case of the hidden layer. For example, the method described by Garson[19] calculates a measure of the relative contribution of the $i^{\text{th}}$ input neuron to the $k^{\text{th}}$ output neuron and places more stress on the connections between the hidden and output layers. Yoon et al[20] extended this approach but still focussed on contributions of input descriptors to the output via the hidden layer neurons. A common feature of both approaches is their empirical nature. That is, the final contribution values are obtained by using the original training set.

Our first attempt at designing a ranking method followed the approach of Garson. In this case we defined the squared relative contribution of the $j^{\text{th}}$ hidden neuron for the $k^{\text{th}}$ example in the training set to be

$$SRC_{kj} = \frac{\left({}^{k}x_j^H w_j^H\right)^2}{\sum_{j=1}^{n_H}\left({}^{k}x_j^H w_j^H\right)^2 + b^2} \tag{9.3}$$

where $x_j^H$ and $w_j^H$ are the output and weight to the output neuron for of the $j^{\text{th}}$ hidden neuron respectively, $b$ is the bias term for the output neuron and $n_H$ is the number of hidden layer neurons. The superscript $k$ indicates that the output of the $j^{\text{th}}$ neuron is for the $k^{\text{th}}$ example. The final value of the squared relative contribution for the $j^{\text{th}}$ hidden neuron was given by

$$SRC_j = \frac{1}{n}\sum_{k=1}^{n} SRC_{kj} \tag{9.4}$$

where $n$ is the number of examples in the training set. However, interpretations developed from the above ranking were not consistent with the interpretations obtained from the linear models.

An alternative approach that we considered was not empirical in nature. That is, it did not directly utilize the dataset used to build the model. In this approach we considered the fact that the contributions of a given hidden layer neuron depends not only on the nature of its contribution to the output neuron, but also on the nature of the contributions to the hidden neuron from the preceding input layer. This is implicitly considered in the empirical approach described. In this approach we considered the overall contribution of a hidden neuron to the output by taking into account all the effective weights associated with this hidden neuron. That is, the contribution of the $j^{\text{th}}$ hidden neuron was initially defined as

$$CV_j = \frac{1}{n_I} \sum_{i=1}^{n_I} w_{ij} w_j^H \tag{9.5}$$

where $n_I$ is the number of input neurons, $w_{ij}$ is the weight between the $i^{\text{th}}$ input neuron and $j^{\text{th}}$ hidden neuron and $w_j^H$ is the weight between the $j^{\text{th}}$ hidden neuron and the output neuron. The above equation simply represents the column means of the effective weight matrix. The resultant values are signed and the absolute value of these contribution values can be used to rank the hidden neurons. However, to make the relative contributions of the hidden neurons clearer we considered the values obtained from Eq. 9.5 as squared contribution values defined as

$$SCV_j = \frac{CV_j^2}{\sum_{j=1}^{n_H} CV_j^2} \tag{9.6}$$

The result of this transformation is that the $SCV_j$ values sum to 1.0. Consequently, the $SCV_j$ values provide a clearer view of the contributions of the hidden neurons and allow us to possibly ignore hidden neurons that have very small values of $SCV_j$.

One aspect of this approach is that we do not take into account the bias terms. Clearly, this approach is not utilizing all the information present within the neural network. There are two reasons why the bias term should be taken into account when ranking hidden neurons. First, most reported measures of contribution are empirical in nature and thus implicitly take into account the bias terms. Second, since we are focussing on the contribution made by a given hidden neuron, we need to consider all

the effects acting through the hidden neuron. Since the bias terms corresponding to the hidden layer can be considered as weights coming form an extra (constant) input neuron, the effective weights being summed in Eq. 9.5, should include an extra term corresponding to the bias term for that hidden neuron. Thus if we denote the bias term for the $j^{\text{th}}$ hidden neuron as $b_j$ then Eq. 9.5 can be rewritten as

$$CV_j = \frac{1}{n_I + 1}\left(\sum_{i=1}^{n_I} w_{ij} w_j^H + b_j w_j^H\right) \tag{9.7}$$

Using this equation, values for $SCV_j$ can be calculated using Eq. 9.6

### 9.2.6 Validation

To ensure that the methodology provides valid interpretations we compared the results of the method to interpretations developed for linear models. For a given QSAR problem, the descriptor subsets that lead to the best linear model are generally not the same as those that lead to the best CNN model. However, comparing interpretations of CNN and linear models with different descriptors would lead to a less accurate comparison. Furthermore, one would expect that given the same descriptors, both CNN and linear models should capture the structure-property trends present in the dataset in a similar fashion. If the interpretation of these trends in the CNN model does not match those developed using the linear model, the discrepancy would indicate that the CNN interpretation methodology is flawed. As a result, we developed the CNN models using the same subset of descriptors that were present in the corresponding linear models. The CNN models built using these descriptors are not necessarily the best (in terms of training set and prediction set performance). However, this work focusses on the extraction of structure-property trends in a human understandable format rather than investigating the predictive power of the CNN models. In this respect we feel that the comparison of the CNN interpretations to those made for linear models using the same set of descriptors is a valid procedure.

The linear models were developed using the ADAPT methodology, described in Chapter 3. This involved the use of a simulated annealing[21,22] algorithm to search for good descriptor subsets. The models were then interpreted using the PLS analysis technique described above. As mentioned, the CNN models used the same descriptors that were present in their corresponding linear models. For each dataset a number of CNN models with different architectures (i.e., different numbers of hidden neurons) were built.

The architectures were limited by considering a rule of thumb that indicates that the total number of parameters (weights and biases) should not be more than half the size of the training set. The final architecture for each CNN model was chosen by considering the cost function defined by Eq. 2.6. The architecture that gave the lowest value of the cost function was chosen as the best CNN model for that dataset.

## 9.3 Datasets

We considered three datasets. The first dataset consisted of a 147 member subset of the DIPPR boiling point dataset studied by Goll et al.[23] The dependent variable for this dataset ranged from 145.1 K to 653.1 K. The original work reported linear and nonlinear models. However no interpretations of these models were provided. We consider the linear model for this dataset that was described in Chapter 8. Though that chapter also reported a CNN model, we develop a new CNN model so that we would be able to obtain a more direct comparison between the final interpretations as described above.

The second dataset consisted of 97 compounds studied by Stanton et al.[24] These compounds were studied for their ability to cross the blood-brain barrier and the modeled property was the logarithm of the blood-brain partition coefficient ($\log(BB)$). The dependent variable in this dataset ranged from $-2.00$ to 1.44 log units. The previously published work reported a linear model and an associated PLS based interpretation.

The final dataset consisted of 136 compounds studied by Patel et al.[25] The work considered the skin permeability of the 136 compounds. The dependent variable for this dataset was the logarithm of the permeability coefficient, $\log(K_p)$ and ranged from $-5.03$ to $-0.85$ log units. Though the paper reported a set of linear models, we developed a new linear model using a variety of descriptors including hydrophobic surface area descriptors.[24,26] A PLS analysis of this model is also presented for comparison to the interpretation of the corresponding CNN model.

## 9.4 Results

For each dataset we present a summary of the linear model and the associated PLS interpretation. We then describe the neural network model built using the dataset and descriptors from the linear models, and present the corresponding interpretation. For all models the descriptors are summarized in Table 9.2.

### 9.4.1 DIPPR Dataset

The statistics of the linear model are summarized in Table 9.3. The descriptors are shown in Table 9.2. The $R^2$ and RMSE values were 0.98 and 9.98, respectively. The $F$-statistics (on 7 and 139 degrees of freedom) was 1001 which is greater than the critical value of 2.78 ($\alpha = 0.05$ level). The model is thus statistically valid. Tables 9.4 and 9.5 summarize the results of the PLS analysis for the linear model. The $Q^2$ column in Table 9.4 indicates that the first two components explain approximately 95% of the structure-property relationship (SPR) encoded by the model. As a result, the bulk of the linear interpretation is provided by these components. If we now look at the column for the first component in Table 9.5 we see that the most important descriptors are MW (molecular weight) and V4P-5 ($4^{\text{th}}$ order valence path connectivity index). Both these descriptors characterize molecular size and it is evident that larger values of these descriptors correlate to higher values of the boiling point. Considering the second component we see that the most important descriptors are now RSHM and PNSA-3. The former characterizes hydrogen bonding ability and the latter is a measure of the charge weighted partial negative surface area. The negative sign for PNSA-3 indicates that molecules with smaller values of the descriptor (i.e., having smaller charge weighted partial negative surface area) should have lower boiling points. On the other hand, the positive sign for RSHM indicates that molecules with better hydrogen bonding ability should have higher boiling points, which is in accord with experimental observations. In summary, the linear model encodes two main SPR's. The first trend is dispersion forces, in which atomic contributions to these forces are individually weak but for larger molecules the greater number of interactions leads to a larger attractive force. The other main trend is the attractive forces mainly due to hydrogen bonding. Clearly, this description of the SPR is not new or novel. However now that we know what type of descriptors contribute to the SPR and the nature of the correlations we can compare these observations with those obtained from the CNN model.

The CNN model that was developed for this dataset had a 7–4–1 architecure. As described above, the descriptors for the CNN model were the same as those used in the linear model. The statistics for the training, cross-validation and prediction sets are shown in Table 9.6. The effective weight matrix for this model is shown in Table 9.7. The columns correspond to the hidden neurons and are ordered by the SCV values described previously, and they are shown in the last row of the table. The SCV values indicate that the first and third hidden neurons are the most important, whereas the second

and fourth hidden neurons play a lesser role. The use of the SCV values in choosing which hidden neurons to concentrate on is analogous to the use of the $Q^2$ value to focus on components in the PLS approach. Considering the first column in Table 9.7 we see that the most weighted descriptors are V4P-4, RSHM and MW. All three descriptors have positive weights indicating a that these descriptors are positively correlated with boiling point. When we consider the second column (the third hidden neuron) we see that the two of the three most important descriptors are the same as in the first hidden neuron. Since these have the same signs as before, we may ignore them and consider the most important descriptor not already considered. This descriptor is PNSA-3, and it is negatively correlated to the boiling point. It is clear that the types of descriptors, as well as their correlations, that play the main roles in the SPR encoded by the CNN model are the same as described by the linear model. The main difference is that the relative importance of the descriptors, over the hidden neurons being considered, are different from that described in the linear model. For example, the linear model indicates that PNSA-3 plays a very important role in the SPR, whereas the CNN accords it a less significant role. On the other hand, the hydrogen bonding ability described by RSHM plays a very important role in the CNN, making up for the absence of the charged surface area descriptor. Similarly, the relative importance of the V4P-5 and MW descriptors are swapped in the two interpretations, but since both characterize size, the main SPR trends for the dataset are explained in a similar fashion by both models. These differences are not unexpected, since the CNN correlates the descriptors in a nonlinear fashion. Thus it is expected that the relative roles played by each descriptor in the nonlinear relationship will be different when compared to the linear model. The point to note is, that, though MW is relegated to a less important role, the main SPR trends extracted from the CNN model by this interpretation technique are identical to those present in the linear model.

### 9.4.2 BBB Dataset

The linear model and associated PLS interpretation are described in the original work.[24] However, we summarize the statistical results of the original model in Table 9.8. The $R^2$ for the model was 0.78 and the $F$-statistic was 80.7 (on 4 and 92 degrees of freedom) which was greater than the critical value of 2.47 ($\alpha = 0.05$). The results of the PLS analysis are presented in Tables 9.9 and 9.10. From Table 9.9 we see that the first two components explain 76% of the total variance in the observed property thus allowing us to ignore the remaining components. From Table 9.10 we see that in

the first component the most weighted descriptors are PNHS-3 (a hydrophobic surface area descriptor), NDB (count of double bonds) and WNSA-3 (a charged partial surface area descriptor, characterizing the partial negative surface area). Clearly, larger values of PNHS-3 and WNSA-3 will correlate to larger values of the property whereas lower values of NDB will correlate to larger values of the property. In component 2 we see that WNSA-3 is opposite in sign. This indicates that the second component makes up for over-predictions made by the first component. Similar reasoning can be applied to the weight for V4P-5 in the second component. Some large hydrophobic molecules are under-estimated by component 1. In component 2 however, the positive weight for V4P-5 (which is a measure of branching and thus size) indicates that larger molecules will have higher penetration ability. Therefore the second component makes up for under-estimation of large hydrophobic molecules by the first one. In brief, the SPR trends captured by the linear model indicate that smaller hydrophobic molecules will better penetrate the BBB compared to larger hydrophilic molecules. These trends are discussed in more detail in the original work.[24]

The CNN model developed for this dataset had a 4–4–1 architecture. The statistics for this model are presented in Table 9.6 and the effective weight matrix is shown in Table 9.11. The SCV values for the hidden neurons are shown in the last row of Table 9.11. They indicate that the first and second hidden neurons contribute to the bulk of the SPR encoded by the model. If we consider the weights for the first hidden neuron (first column) we see, in general, the same correlations as described in the linear model. Both PNHS-3 and WNSA-3 are positively correlated with the predicted property and NDB is negatively correlated. However the difference we see here is that V4P-5 is one of the most important descriptors and is positively correlated with the predicted property. On the other hand PNHS-3 plays a much smaller role in this model than in the linear model. If we consider the second hidden neuron (second column), we see that the weight for V4P-5 is now lower and that for NDB has increased. One can consider this as the CNN attempting to downplay the increased size effects described by V4P-5. When we consider the fourth hidden neuron we see that the V4P-5 now has a negative weight and thus serves to balance the over-estimation of the property for larger molecules made by the first two hidden neurons. Overall, we see that the main trends described by the CNN model indicate that fewer double bonds and more hydrophobicity lead to higher ability to penetrate the BBB, though it does appear that the model focuses on a positive correlation between size and $\log(BB)$ values via the V4P-5 descriptor. This is quite similar

to the conclusions obtained from the PLS interpretation of the linear model. Some differences are present - mainly in the context of size (described by V4P-5) and the relative importance of the descriptors for a given hidden neuron. As described above, this is not surprising given that the nonlinear relationship between the descriptors generated by the CNN is significantly different from the linear relationship described by the original regression model. However, the fact that the description of the main SPR trends encoded within the CNN model compare well with those of the linear model, serve to confirm our assumption that both models should encode similar trends as well as the validity of this interpretation technique to extract these trends. Furthermore, the structure-property trends extracted from both types of models by the repspective interpretation techniques are consistent with physical descriptions of the factors that are believed to affect the transport of drugs across the blood brain barrier[27,28]

### 9.4.3 Skin Permeability Dataset

This dataset was originally studied by Patel et al.[25] where they developed a linear regression model using 158 compounds. However, owing to the presence of outliers, the final models were built using 143 compounds. We considered the original 158 compounds and chose a 136 member subset to work with. The linear model we developed for this dataset is summarized in Table 9.12. The $R^2$ value for this model was 0.84 and the $F$-statistic was 97.5 on 7 and 128 degrees of freedom which was greater than the critical value of 2.08 ($\alpha = 0.05$) indicating that the model was statistically valid. We then developed a PLS interpretation of this linear model and the results of the PLS model are summarized in Tables 9.13 and 9.14. The $Q^2$ values in Table 9.13 indicate that the first three components describe the bulk of the SPR. If we now consider Table 9.14 we see that the most important descriptors in the first component are MOLC-9, FPSA-2 and RNHS. MOLC-9 represents Balabans J topological index which is derived from the distance connectivity matrix and characterizes molecular branching. Smaller values of this descriptor indicate smaller or more linear compounds. The FPSA-2 descriptor characterizes the relative partial positive surface area. The negative weight for this descriptor indicates that molecules with smaller partial positive surface areas will be more active. This descriptor characterizes molecules like 2,4,6-trichlorophenol whose molecular surface has a large number of partial negative charged regions. Finally the RNHS descriptor characterizes the hydrophilic surface area. This descriptor serves to balance the effects of FPSA-2 and this can be seen when comparing the activities for

2,4,6-trichlorophenol and 3,4-dimethylphenol (Table 9.1). It is clear that both are of similar size and both have similar activities. However, if FPSA-2 were acting alone, the high value of this descriptor for 3,4-dimethylphenol (due to the partial positive charges on the methyl groups) would lead to a significantly lower activity. However, RNHS indicates that not all small hydrophobic molecules will have negatively charged atoms. Thus the first components indicates that smaller and more hydrophobic molecules will exhibit higher activities.

If we now consider the second component we see that the most weighted descriptors are PPHS, SA and WPHS-3. PPHS measures the total hydrophobic surface area and WPHS-3 is defined as the surface weighted hydrophobic surface area. The positive weights on these descriptors indicate that a larger hydrophobic surface area is correlated positively with activity. SA, which measures molecular surface area, is positively weighted indicating larger molecules are more active. The role of this component is to account for some larger molecules observed to be moderately active (Table 9.2). Essentially, the larger size of these molecules leads to a larger hydrophobic surface area which enhances permeability. The component thus corrects for the under-estimation of the larger molecules by component 1 (such as fentanyl and sulfentanil) by taking into account their higher hydrophobicity and also corrects for the over-estimation of smaller molecules (such as methanol and urea) by component 1 by taking into account their lower hydrophobicity.

The third component mainly corrects for the over-estimation of hexachlorobutadiene and hexachloroethane by component 2 due to emphasis on the hydrophobic surface area. This is corrected for by component 3 by the negative weights for FPSA-2 and WPHS-3.

Thus the main conclusion that can be drawn from the linear model is that molecular size and hydrophobicity are two key characteristics that appear to explain the observed skin permeability of these compounds. This is consistent with the conclusions of Patel et al.[25] and also with the general understanding regarding the mechanism of skin permeation.

The CNN model developed for this dataset had a 7–5–1 architecture and the statistics are reported in Table 9.6. The effective weight matrix is shown in Table 9.15. In the case of this model, we see that the SCV value for the $5^{th}$ hidden neuron is nearly six times larger than that for the next most important neuron. If we consider the most important hidden neuron (5) we see that the most weighted descriptors are FPSA-2, NN (the number of nitrogens) and PPHS. The signs of these effective weights are the

same as described by the PLS analysis of the linear model. That is, these descriptors have the same effect on the output of the model in both the linear and nonlinear cases. Thus, the most important hidden neuron indicates that molecules with smaller polar surface area and larger hydrophobic surface area will exhibit higher activity. Moving onto the next most important hidden neuron, we see that the most weighted descriptors are SA, PPHS, MOLC-9. It is clear that this hidden neuron focuses on size effects. However, the negative weight for surface area indicates that larger molecules will be more active. This is a valid conclusion since the dataset does indeed have some larger molecules which are moderately active. This conclusion is further justified by the fact that a larger molecule would have a correspondingly larger hydrophobic surface area, which as the positive weight for PPHS indicates, will lead to higher activity. At the same time, all large molecules do not exhibit high activities. Thus the effect of the SA descriptor is balanced by the positive weight for the MOLC-9 descriptor. Since larger values of MOLC-9 correlate to smaller molecules, the effect of the MOLC-9 descriptor balances the SA descriptor, ensuring that this hidden neuron does not predict all large molecules as active.

In the next most important hidden neuron (4) we see that the most weighted descriptors are RNHS, FPSA-2, PPHS and MOLC-9. In this hidden neuron, MOLC-9 describes the effect of size and indicates that smaller molecules will exhibit higher activity. However, the positive weight FPSA-2 indicates that molecules with larger partially positive charged surface area will exhibit higher activity. When we also consider the negative weight for PPHS (indicating more active molecules should have lower hydrophobic surface area) we see that this neuron focuses mainly on smaller, more polar molecules. This trend is reinforced to some extent by the negative weight for RNHS. RNHS describes both hydrophilic and partial negatively charged regions regions of a molecule. Due to the design of the descriptor, the negative sign on this descriptor indicates that molecules with smaller partial negatively charged surface area and more hydrophilic atoms will exhibit relatively higher activities. At the same time if we consider the SCV value for this hidden neuron we see that it is just 2% of the SCV for the most important hidden neuron. One would thus expect that this neuron would not provide very detailed information regarding the SPR encoded in the model. Similar reasoning can be applied to the last two columns of Table 9.15.

The interpretation of the CNN model described here matches quite closely with that of the linear model. The main difference is in the ordering of the important trends. As described before, this is not surprising due to the nonlinear encoding of the structure

property relationships by the CNN model. However, though the above description is quite detailed and by allows us to look at descriptor values for individual observations and understand why they are predicted to display greater or less skin permeation, a visual approach to understanding the effects of each hidden neuron, analogous to score plots[1] in the PLS interpretation scheme, would be useful. One approach to this problem is to generate plots using the effective weights.

### 9.4.4 Score Plots

As described above, the use of the effective weights *linearizes* the neural network. In effect, the network is transformed into a set of connections between input descriptors and the output neuron, ignoring nonlinear transfer functions. The *pseudo-network* can be used to generate a set of score values for each hidden neuron. For the $k^{\text{th}}$ member of the dataset, the score for that member using the $j^{\text{th}}$ hidden neuron can be defined as

$$score_{kj} = \sum_{i=1}^{n_I} w_{ij} w_j^H x_{ki} \tag{9.8}$$

where $w_{ij}w_j^H$ is simply the effective weight for the $i^{\text{th}}$ input neuron (see Table 9.1), $n_I$ is the number of input descriptors and $x_{ki}$ is the value of the $i^{\text{th}}$ descriptor for the $k^{\text{th}}$ member of the dataset. The result of Eq. 9.8 is that for each hidden neuron, a set of scores are obtained for the dataset. Clearly, these are not meant to quantitatively model the observed properties well. However, our interest in lies in the qualitative behavior of the scores. That is, we expect that if a compound has a high observed activity, its score value should be high and vice versa for compounds with low observed activity. Thus a plot of the scores for a given hidden neuron versus the observed property should lie along the 1:1. Points lying in the lower right quadrant would represent compounds over-estimated by the hidden neuron and points lying the in the upper left quadrant would represent compounds under-estimated by the hidden neuron.

We tested this approach by creating score plots for the three most important hidden neurons for the CNN model developed for the skin permeability dataset. These are shown in Figs. 9.5, 9.6 and 9.7. Considering the plot for the $5^{\text{th}}$ hidden neuron we see that the plot does exhibit the behavior we expect. Compounds **21**, **43** and **75** are predicted as active and **42**, **46** and **135** are predicted as inactive. The structures for these compounds are shown in Fig. 9.3. As described previously, active compounds will be characterized by smaller size and increased hydrophobicity. As Fig. 9.3 shows,

active compounds do indeed have a hydrophobic benzene ring. In contrast the inactive compounds are generally larger and, more significantly, have a number of polar regions. An interesting case is urea (compound **135**). This is a very small compound, but it is dominated by polar groups, responsible for hydrogen bonding. The fact that the neuron predicts this compound correctly as inactive is due to the fact that this neuron mainly stresses the FPSA-2 and NN descriptors. As seen from Table 9.15, the negative weights indicate that a larger number of polar groups would inhibit activity. As a result, though compound **135** is small, the polar effects outweigh the size effect. Fig. 9.5 also indicates that compounds **81**, **114**, **69** and **77** are all mispredicted. The first two are over-estimated and the last two are under-estimated.

If we now consider the score plot for the 2$^{nd}$ hidden neuron in Fig. 9.6, we see the four mispredicted compounds, mentioned above, are now more correctly predicted. However **81** does appear to be over-estimated. Apart from these cases, the majority of the compounds do not appear to be well predicted. We believe that this can be explained by the very low contribution value of this hidden neuron compare to the 5$^{th}$ hidden neuron. Due to the very low SCV value for this neuron, we believe that it does not have significant explanatory power. The structures of the active and inactive compounds are compared in Fig. 9.4. As described above, the main focus of the 2$^{nd}$ hidden neuron is to account for compounds which are relatively larger but also moderately active. As can be seen from the structures, though compounds are **78**, **81** and **114** are significantly larger than the active compounds in the preceding component, they are indeed moderately active. Correspondingly, this hidden neuron is also able to account for the low activity of a number of small compounds (**72** and **77**). Though this hidden neuron mispredicts a number of compounds, the majority have already correctly predicted in the preceding hidden neuron. A number of them, such as **87**, are corrected by the next most important hidden neuron.

Considering the score plot for the 4$^{th}$ hidden neuron we see that, though it does correctly predict a number of compounds as active, it performs poorly on inactive compounds. Once again, we believe that the low contribution value (1% of the SCV of the most important hidden neuron) indicates that it will not have significant explanatory power. However, it does correct for the misprediction of **87** by the 2$^{nd}$ hidden neuron. In addition, compound **81** is now shifted closer to the 1:1 line, correcting for the slight overestimation by the preceding hidden neuron. Score plots for the remaining hidden neurons can be similarly analysed, though we observed that they did not explain any

significant trends and rather corrected for a few mispredictions by the preceding 3 hidden neurons.

The above discussion shows that the score plots derived from the effective weights help provide a more visual approach to the interpretation of a CNN model. Coupled with an analysis of the effective weight table, a CNN model can be interpreted in a very focused, compound-wise manner.

## 9.5 Discussion & Conclusions

The CNN interpretation methodology that we have presented provides a means for using CNN models both for predictive purposes as well as for understanding structure-property trends present in the dataset. The methodology is similar in concept to the PLS interpretation method for linear regression models. The analogy to the PLS method is strengthened when we consider that the hidden neurons are analogous to latent variables (and in the case of linear transfer functions, are identical). Though a number of approaches to understanding a CNN model exist in the literature, our approach provides a detailed view of the effect of the input descriptors as they act via each hidden neuron. Furthermore, previous approaches are empirical in the sense that they require the direct use of the training set to determine the importance of input or hidden neurons. The method described here avoids this by making use of the effective weights only. A justification for this approach is that the weights and biases in the final CNN model are derived from the structure-property trends present in the data. As a result the optimized weights and biases already contain the information regarding the SPR's and thus subsequent use of the training set to develop the interpretation is unnecessary. However, the training set is used to generate the hidden neuron score plots which can be used to focus on the contributions of individual hidden neurons to the overall predictive behavior of the model and understand the behavior of the hidden neurons by considering specific compounds.

The method was validated on three datasets covering physical and biological properties. Interpretations from the CNN model were compared to linear models built for these datasets (using the same descriptors that were present in the CNN model) and it can be seen that the structure property trends described by both models are in very close agreement. The main differences between the interpretations is in the importance ascribed to specific descriptors. That is, the most important descriptor in the most important latent variable in the PLS interpretation might not occupy the same position in

the CNN interpretation. This is not surprising due to the fact that the neural network combines the input descriptors nonlinearly and thus the role of the individual descriptors in the nonlinear relationship may be different from that played in a linear relationship.

Another important aspect of this study was that we considered CNN models which contained the same descriptors as the linear models. The linear models were developed using a genetic algorithm for feature selection and were thus optimal linear models. This is not the case for the corresponding neural network models. This is due to the fact that, in general, when a CNN routine is linked to the genetic algorithm, the optimal descriptor subsets differ from the case where the objective function for the genetic algorithm is a linear regression function. As a result, in the examples we considered, the CNN models were not necessarily optimal and hence the interpretations may differ to some extent when optimal models are built for the datasets. However structure property trends are a feature of the data rather than the model describing the data. Thus even if optimal descriptor subsets are considered, it is expected that these descriptors will capture the structure property trends present in the dataset, albeit with greater accuracy. Hence, it is expected that interpretations from the optimal CNN models will not differ significantly from those described here.

However, there is one aspect that should be considered when interpreting CNN models using this method. The definition of effective weights ignores the effect of the nonlinear transfer function for each neuron. In effect, the effective weights linearize the model. As a result the interpretation does not provide a full description of the nonlinear relationships between structural features and the property. That is, some information regarding the encoded SPR is lost. We feel that the tradeoff between interpretability and information loss is justified due to the simple nature of method. To fully describe the nonlinear encoding of an SPR would essentially require that the CNN model be analyzed to generate a functional form corresponding to the encoded SPR. The neural network literature describes a number of approaches to rule extraction in the form of if-then rules.[8–11,29] as well as some instances of analytical rule extraction.[12,30,31] As mentioned previously, most of the previous approaches to the interpretation of neural networks or extraction of rules from neural networks are focused on specific types of neural network algorithms. In addition, a number of the rule extraction methods described in the literature are carried out by analysing the neural network with the help of a genetic algorithm[29,32] or by decision trees,[11] adding an extra layer of complexity to the methodology. The method described here is quite general as it requires only the optimized weights and biases from the network. The only current restriction on the method

is that the neural network must have a single hidden layer. However, the methodology described in this chapter can be extended to the case of multiple hidden layers though the complexity of the treatment will correspondingly increase.

The interpretation method described in this work expands the role of CNN models in the QSAR modeling field. The black box reputation of CNN models has led to their main usage as predictive tools with no explanation of the structure-property trends that are encoded within the model. We believe that this interpretation method will allow for a detailed understanding of the structure-property trends encoded in CNN models allowing them to be used for both predictive and design purposes.

Table 9.1. Tabular representation of effective weights for a hypothetical 4–3–1 CNN model. I1, I2, I3 and I4 represent the four input neurons (descriptors). $w_{ij}$ represents the weight for the connection between the $i^{\text{th}}$ input neuron and the $j^{\text{th}}$ hidden neuron. $w_j^H$ represents the weight between the $j^{\text{th}}$ hidden neuron and the output neuron. For this example $i$ ranges from 1 to 4 and $j$ ranges from 1 to 3

| | Hidden Neuron | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| I1 | $w_{11}w_1^H$ | $w_{12}w_2^H$ | $w_{13}w_3^H$ |
| I2 | $w_{21}w_1^H$ | $w_{22}w_2^H$ | $w_{23}w_3^H$ |
| I3 | $w_{31}w_1^H$ | $w_{32}w_2^H$ | $w_{33}w_3^H$ |
| I4 | $w_{41}w_1^H$ | $w_{42}w_2^H$ | $w_{43}w_3^H$ |

Table 9.2: A glossary of the descriptors used in this study

| Descriptor code | Meaning | Reference |
|---|---|---|
| DPHS | The difference between the hydrophobic and hydrophilic surface area | 24, 26 |
| FPSA-2 | Charge weighted partial positive surface area divided by the total surface area | 33 |
| MOLC-9 | Balaban J topological index | 34, 35 |
| MW | Molecular weight | |
| NDB | Number of double bonds | |
| NN | Number of nitrogens | |
| PNHS-3 | Atomic constant weighted hydrophilic surface area | 24, 26 |
| PPHS | Total molecular hydrophobic surface area | 24, 26 |
| SA | Surface are of the molecule | |
| S4PC-12 | $4^{\text{th}}$ order simple path cluster molecular connectivity index | 36–38 |
| V4P-5 | $4^{\text{th}}$ order valence path molecular connectivity index | 36–38 |
| WNSA-3 | Difference between the partial negative surface area and the sum of the surface area on negative parts of molecule multiplied by the total molecular surface area | 33 |
| WPHS-3 | Surface weighted hydrophobic surface area | 24, 26 |
| WTPT-2 | Molecular ID divided by the total number of atoms | 39 |
| RNHS | Product of the surface area for the most negative atom and the most hydrophilic atom constant divided by the sum of the hydrophilic constants | 24, 26 |
| RSHM | Fraction of the total molecular surface area associated with hydrogen bond acceptor groups | 33 |

Table 9.3. Summary of the linear regression model developed for the DIPPR dataset

| | Estimate | Std. Error | $t$ |
|---|---|---|---|
| (Intercept) | −215.09 | 29.45 | −7.30 |
| PNSA-3 | −3.56 | 0.21 | −16.90 |
| RSHM | 608.07 | 21.30 | 28.55 |
| V4P-5 | 19.57 | 3.30 | 5.92 |
| S4PC-12 | 12.08 | 1.57 | 7.69 |
| MW | 0.57 | 0.061 | 9.42 |
| WTPT-2 | 236.10 | 16.57 | 14.25 |
| DPHS | 0.19 | 0.02 | 7.07 |

Table 9.4. Summary of the PLS analysis based on the linear regression model developed for the DIPPR dataset

| Components | Error SS | $R^2$ | PRESS | $Q^2$ |
|---|---|---|---|---|
| 1 | 94868.50 | 0.86 | 99647.60 | 0.85 |
| 2 | 26221.60 | 0.96 | 29046.70 | 0.95 |
| 3 | 16614.80 | 0.97 | 19303.30 | 0.97 |
| 4 | 14670.80 | 0.97 | 17027.60 | 0.97 |
| 5 | 14032.50 | 0.97 | 16281.30 | 0.97 |
| 6 | 13775.90 | 0.98 | 15870.60 | 0.97 |
| 7 | 13570.90 | 0.98 | 15653.00 | 0.97 |

Table 9.5. The X-weights for the PLS components from the PLS analysis summarized in Table 9.4

| | Component | | | | | | |
|---|---|---|---|---|---|---|---|
| Descriptor | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| PNSA-3 | -0.30 | -0.42 | 0.20 | -0.25 | 0.25 | -0.73 | -0.12 |
| RSHM | 0.19 | 0.77 | 0.34 | -0.03 | 0.22 | -0.37 | 0.20 |
| V4P-5 | 0.48 | -0.15 | -0.07 | -0.66 | -0.36 | -0.09 | 0.38 |
| S4PC-12 | 0.28 | -0.07 | -0.57 | 0.53 | -0.03 | -0.46 | 0.26 |
| MW | 0.49 | -0.085 | 0.36 | 0.24 | -0.39 | -0.17 | -0.60 |
| WTPT-2 | 0.48 | -0.05 | -0.26 | -0.22 | 0.70 | 0.13 | -0.35 |
| DPHS | 0.26 | -0.41 | 0.54 | 0.32 | 0.29 | 0.18 | 0.48 |

Table 9.6. Summary of the architectures and statistics for the CNN models developed for the datasets considered in this study. In all cases, the input descriptors were the same as those used in the corresponding linear models

| | | RMSE | | | $R^2$ | | |
|---|---|---|---|---|---|---|---|
| Dataset | Architecture | TSET | CVSET | PSET | TSET | CVSET | PSET |
| DIPPR | 7–4–1 | 15.21 | 38.51 | 15.07 | 0.91 | 0.45 | 0.94 |
| BBB | 4–4–1 | 0.25 | 0.38 | 0.47 | 0.88 | 0.88 | 0.74 |
| Skin | 7–5–1 | 0.23 | 0.27 | 0.31 | 0.94 | 0.93 | 0.91 |

Table 9.7. The effective weight matrix for the 7–4–1 CNN model developed for the DIPPR dataset. The columns (hidden neurons) are ordered by the the squared contribution values (SCV) shown in the last row. Note that the SRC value for the bias term is not considered during the ranking

| | Hidden Neuron | | | |
|---|---|---|---|---|
| | 1 | 3 | 2 | 4 |
| PNSA-3 | −1.80 | −6.57 | 0.39 | −1.43 |
| RSHM | 4.03 | 6.15 | 1.50 | 1.01 |
| V4P-5 | 9.45 | 2.15 | 3.24 | 0.60 |
| S4PC-12 | 3.36 | 2.73 | 1.99 | 0.56 |
| MW-16 | 3.94 | 8.42 | 1.94 | 0.76 |
| WTPT-2 | 1.71 | 2.61 | 1.17 | −0.13 |
| DPHS | 0.66 | 0.44 | 0.33 | 1.65 |
| SCV | 0.52 | 0.33 | 0.13 | 0.01 |

Table 9.8. Summary of the linear regression model developed for the BBB dataset

| | Estimate | Std. Error | $t$ |
|---|---|---|---|
| (Intercept) | 0.53 | 0.07 | 7.28 |
| WNSA-3 | 0.04 | 0.01 | 6.24 |
| V4P-5 | 0.24 | 0.03 | 7.13 |
| NDB | -0.13 | 0.03 | -5.05 |
| PNHS-3 | 0.03 | 0.00 | 6.93 |

Table 9.9. Summary of the PLS analysis based on the linear regression model developed for the BBB dataset

| Components | Error SS | $R^2$ | PRESS | $Q^2$ |
|---|---|---|---|---|
| 1 | 22.40 | 0.62 | 23.80 | 0.59 |
| 2 | 13.90 | 0.76 | 15.40 | 0.74 |
| 3 | 13.00 | 0.78 | 14.80 | 0.75 |
| 4 | 13.00 | 0.78 | 14.70 | 0.75 |

Table 9.10. The X-weights for the PLS components from the PLS analysis summarized in Table 9.9

| | Component | | | |
|---|---|---|---|---|
| Descriptor | 1 | 2 | 3 | 4 |
| WNSA-3 | 0.54 | -0.13 | 0.79 | 0.28 |
| V4P-5 | -0.09 | 0.97 | 0.17 | 0.12 |
| NDB | -0.57 | -0.08 | 0.58 | -0.58 |
| PNHS-3 | 0.62 | 0.17 | -0.12 | -0.76 |

Table 9.11. The effective weight matrix for the 4–4–1 CNN model developed for the BBB dataset. The columns are ordered by the squared contribution values for the hidden neurons, shown in the last row

| | Hidden Neuron | | | |
|---|---|---|---|---|
| | 1 | 2 | 4 | 3 |
| WNSA-3 | 52.41 | 29.30 | −19.64 | 2.26 |
| V4P-5 | 37.65 | 22.14 | −3.51 | −13.99 |
| NDB | −10.50 | −16.85 | −5.02 | 22.16 |
| PNHS-3 | 11.46 | 6.59 | −2.72 | 8.36 |
| SCV | 0.74 | 0.16 | 0.08 | 0.03 |

Table 9.12. Summary of the linear regression model developed for the skin permeability dataset

| | Estimate | Std. Error | $t$ |
|---|---|---|---|
| (Intercept) | -5.47 | 0.24 | -22.94 |
| SA | 0.00 | 0.00 | 6.92 |
| FPSA-2 | -2.38 | 0.17 | -14.12 |
| NN | -0.28 | 0.05 | -6.05 |
| MOLC-9 | 0.50 | 0.07 | 7.19 |
| PPHS | 0.009 | 0.0007 | 13.47 |
| WPHS-3 | -0.02 | 0.00 | -5.41 |
| RNHS | 0.05 | 0.00 | 7.48 |

Table 9.13. Summary of the PLS analysis based on the linear regression model developed for the skin permeability dataset

| Components | Error SS | $R^2$ | PRESS | $Q^2$ |
|---|---|---|---|---|
| 1 | 68.16 | 0.44 | 73.40 | 0.40 |
| 2 | 41.24 | 0.66 | 44.79 | 0.64 |
| 3 | 24.22 | 0.80 | 28.64 | 0.77 |
| 4 | 19.79 | 0.84 | 23.21 | 0.81 |
| 5 | 19.40 | 0.84 | 22.21 | 0.82 |
| 6 | 19.39 | 0.84 | 22.23 | 0.82 |
| 7 | 19.39 | 0.84 | 22.20 | 0.82 |

Table 9.14. The X-weights for the PLS components from the PLS analysis summarized in Table 9.13

| Descriptor | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| SA | −0.08 | 0.52 | 0.20 | −0.31 | −0.29 | −0.71 | −0.07 |
| FPSA-2 | −0.52 | 0.14 | −0.48 | −0.38 | −0.16 | 0.20 | 0.52 |
| NN | −0.36 | −0.03 | 0.07 | 0.45 | −0.74 | 0.18 | −0.27 |
| MOLC-9 | 0.61 | 0.11 | −0.32 | 0.36 | −0.33 | −0.16 | 0.50 |
| PPHS | 0.03 | 0.69 | 0.45 | 0.17 | 0.13 | 0.48 | 0.23 |
| WPHS-3 | 0.09 | 0.48 | −0.65 | 0.10 | 0.16 | 0.10 | −0.55 |
| RNHS | 0.46 | −0.04 | 0.07 | −0.63 | −0.42 | 0.41 | −0.21 |

Table 9.15. The effective weight matrix for the 7–5–1 CNN model developed for the skin permeability dataset. The columns are ordered by the squared contribution values for the hidden neurons, shown in the last row

| | Hidden Neuron | | | | |
|---|---|---|---|---|---|
| | 5 | 2 | 4 | 3 | 1 |
| SA | −44.17 | 67.34 | 8.33 | 8.18 | 5.96 |
| FPSA-2 | −156.82 | −10.72 | 20.85 | −13.07 | −92.47 |
| NN | −97.81 | 2.22 | −6.65 | 1.71 | −12.70 |
| MOLC-9 | −28.85 | 17.79 | 15.40 | −11.36 | −1.20 |
| PPHS | 106.55 | 31.30 | −16.76 | −13.99 | 34.55 |
| WPHS-3 | −11.36 | −14.31 | −2.31 | −10.01 | 54.16 |
| RNHS | 20.16 | −5.89 | −49.57 | 23.88 | 27.09 |
| SCV | 0.85 | 0.13 | 0.02 | 0.01 | 0.00 |

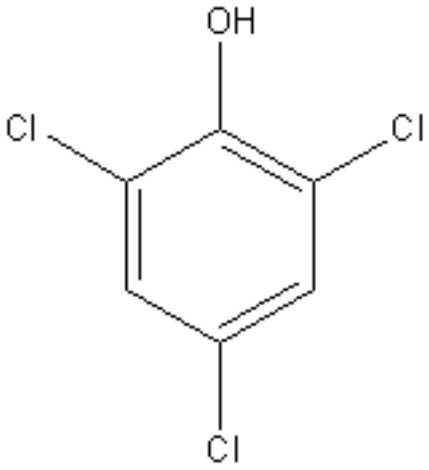
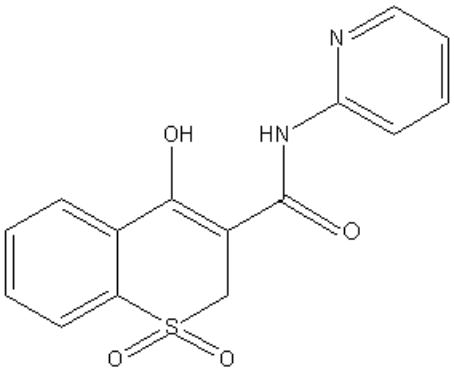
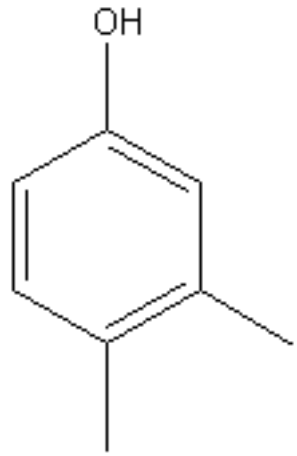
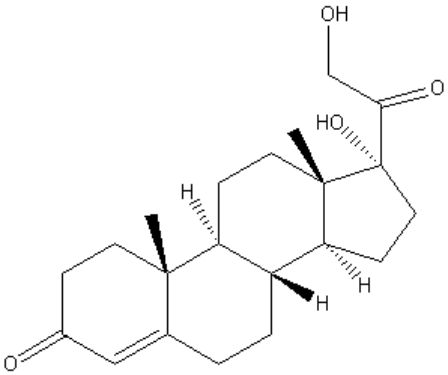
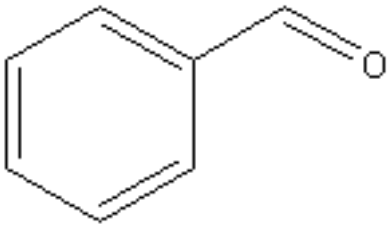
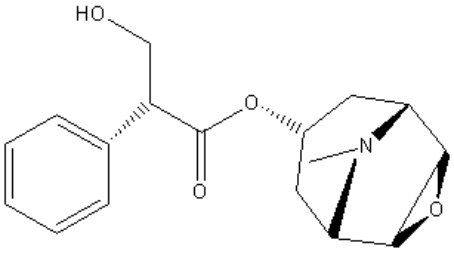
| High Permeability | Low Permeability |
|---|---|
|  |  |
| **3** (-1.23) | **121** (-1.38) |
|  |  |
| **18** (-1.44) | **54** (-4.13) |
|  |  |
| **43** (-0.85) | **129** (-4.30) |

Fig. 9.1. A comparison of compounds exhibiting high and low skin permeability to illustrate the SPR encoded by component 1. The bold number is the serial number and the measured permeability coefficient is displayed in parentheses.

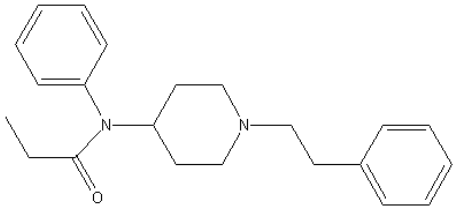
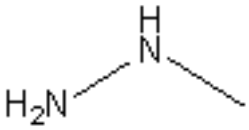
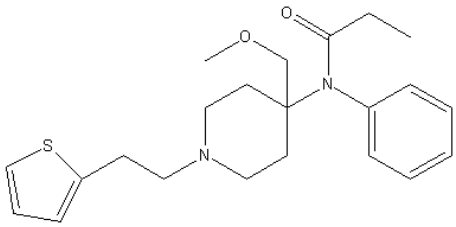
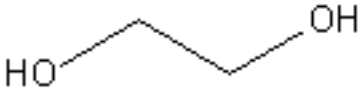
| High Permeability | Low Permeability |
|---|---|
|  |  |
| **82** (-2.25) | **101** (-3.75) |
|  |  |
| **130** (-2.26) | **77** (-4.01) |

Fig. 9.2. A comparison of compounds with high and low skin permeability, predicted by the second PLS component. The bold number is the serial number and the measured permeability coefficient is displayed in parentheses.

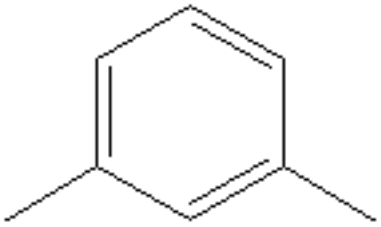
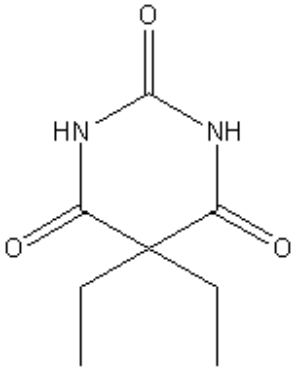
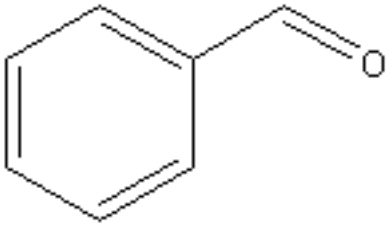
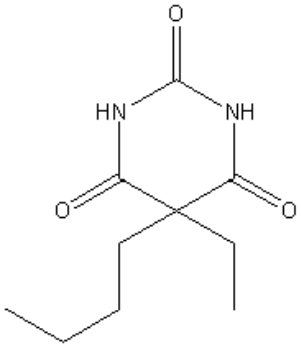
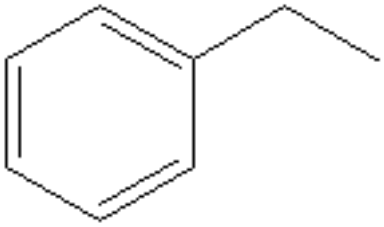
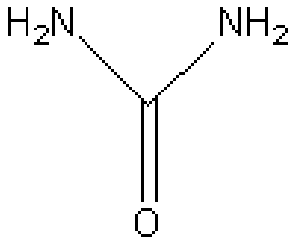
| Moderate Permeability | Low Permeability |
|---|---|
|  |  |
| **21** (-1.10) | **42** (-3.95) |
|  |  |
| **43** (-0.85) | **46** (-3.72) |
|  |  |
| **75** (-1.15) | **135** (-3.83) |

Fig. 9.3. A comparison of structures illustrating compounds with high and low skin permeability, predicted by the $5^{th}$ hidden neuron. The bold number is the serial number and the number in brackets is the measured permeability coefficient

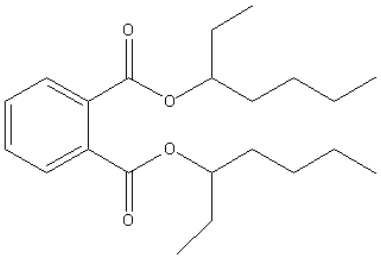
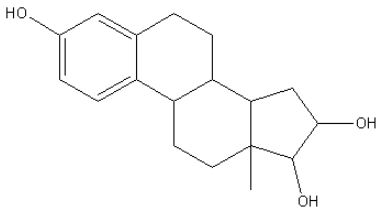
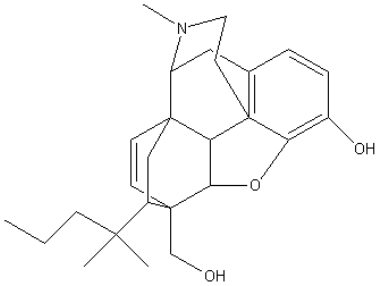
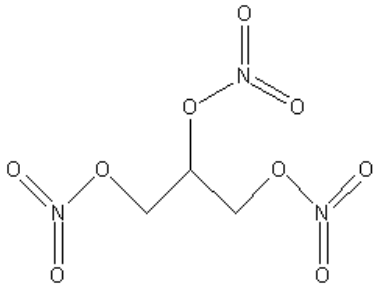
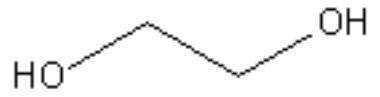
| Moderate Permeability | Low Permeability |
|---|---|
|  |  |
| **78** (-1.52) | **69** (-4.40) |
|  |  |
| **81** (-2.44) | **72** (-4.02) |
|  |  |
| **114** (-1.96) | **77** (-4.07) |

Fig. 9.4. A comparison of structures illustrating compounds with moderate and low skin permeability predicted by the 2nd hidden neuron. The bold number is the serial number and the number in brackets is the measured permeability coefficient

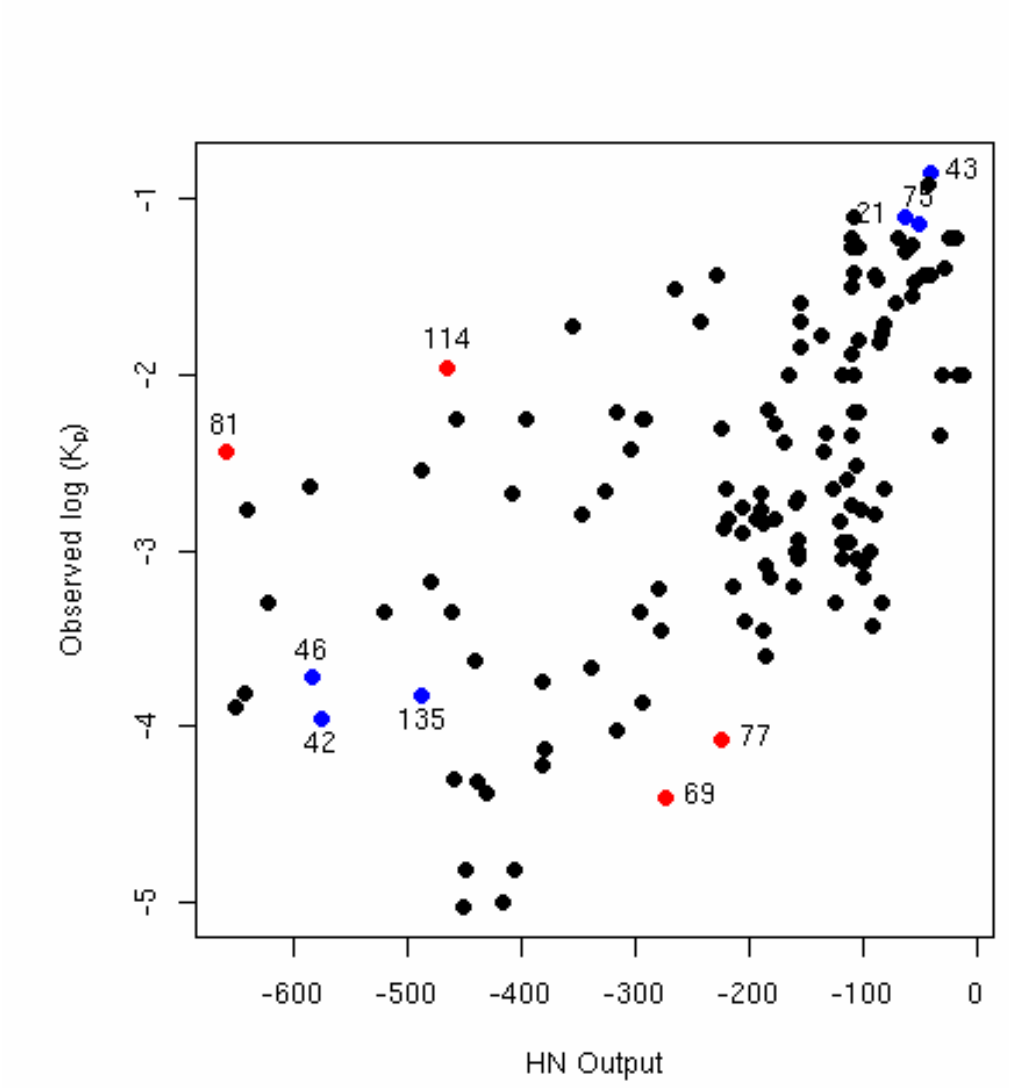



Fig. 9.5. The score plot for the $5^{th}$ hidden neuron. Points marked in red are examples of mispredicted molecules. Points colored blue are examples of well predicted molecules.

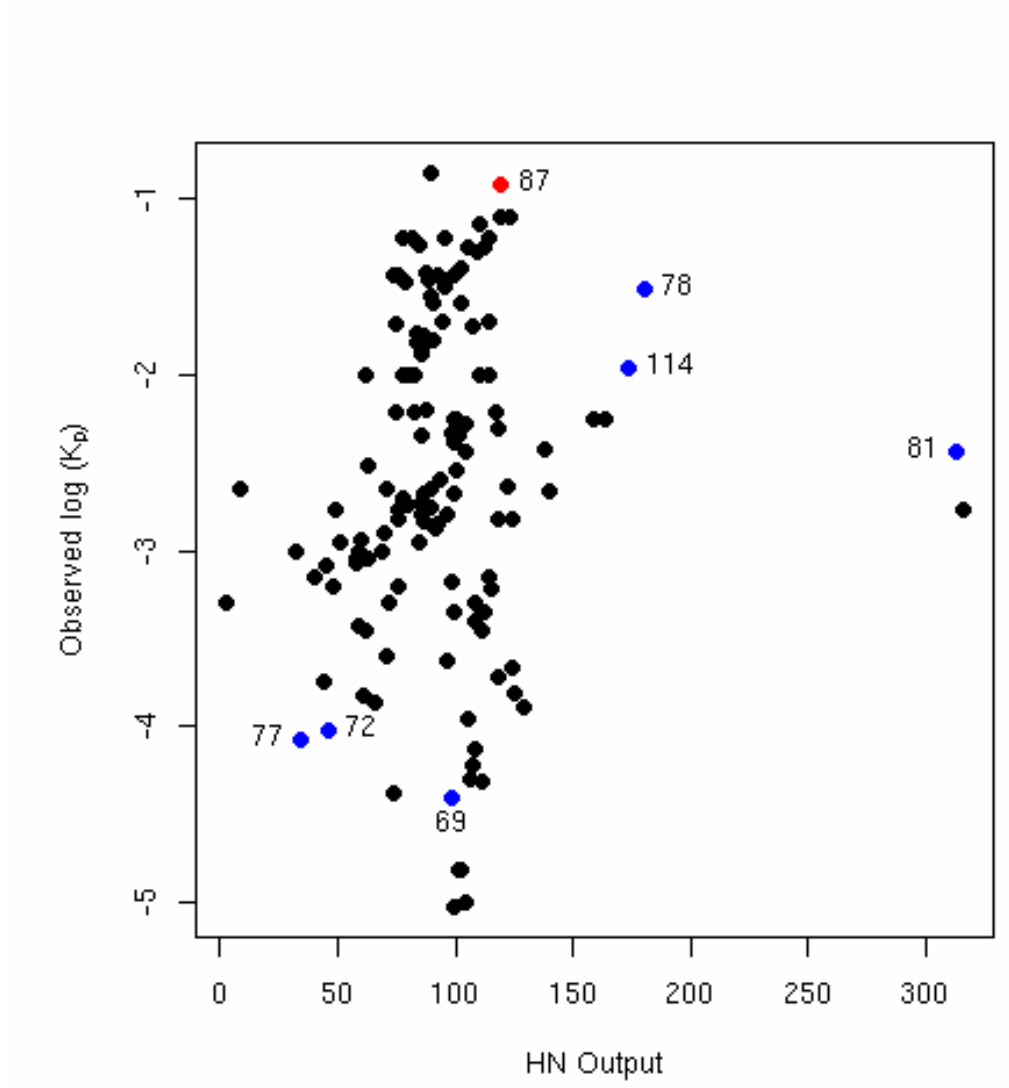



Fig. 9.6. The score plot for the 2nd hidden neuron.Points marked in red are examples of mispredicted molecules. Points colored blue are examples of well predicted molecules.

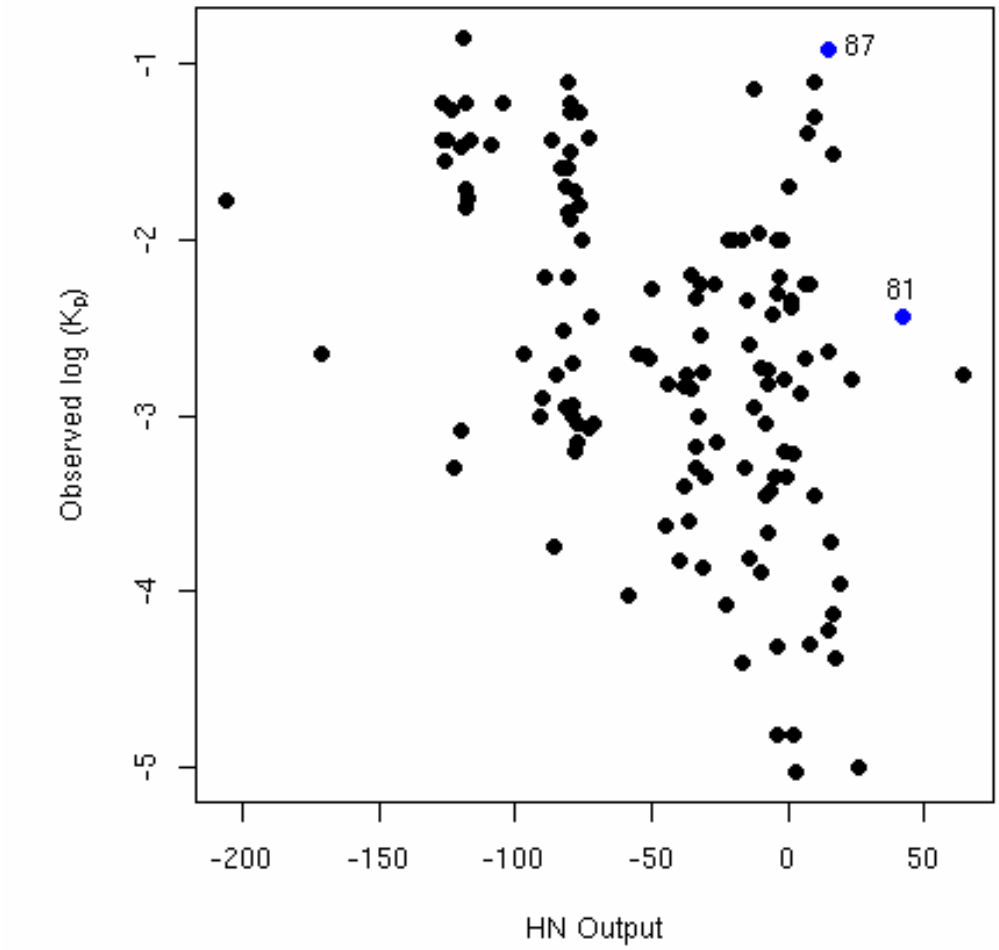



Fig. 9.7. The score plot for the $4^{th}$ hidden neuron. Points marked in red are examples of mispredicted molecules.

## References


[1] Stanton, D. On the Physical Interpretation of QSAR Models. *J. Chem. Inf. Comput Sci.* **2003,** *43,* 1423–1433.

[2] Guha, R.; Jurs, P. C. The Development of Linear, Ensemble and Non-linear Models for the Prediction and Interpretation of the Biological Activity of a Set of PDGFR Inhibitors. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 2179–2189.

[3] Guha, R.; Jurs, P. C. The Development of QSAR Models to Predict and Interpret the Biological Activity of Artemisinin Analogues. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1440–1449.

[4] Breiman, L.; Friedman, J.; Olshen, R.; Stone, C. *Classification and Regresion Trees;* CRC Press: Boca Raton, FL, 1984.

[5] Breiman, L. Random Forests. *Machine Learning* **2001,** *45,* 5–32.

[6] Castro, J.; Mantas, C.; Benitez, J. Interpretation of Artificial Neural Networks by Means of Fuzzy Rules. *IEEE Trans. Neural Networks* **2002,** *13,* 101–116.

[7] Limin, F. Rule Generation from Neural Networks. *IEEE Trans. Systems, Man and Cybernetics* **1994,** *24,* 1114–1124.

[8] Bologna, G. Rule Extraction from Linear Combinations of DIMLP Neural Networks. In *Proceedings of the Sixth Brazilian Symposium on Neural Networks*; IEEE: New York, NY, 2000.

[9] Yao, S.; Wei, C.; He, Z. Evolving Fuzzy Neural Networks for Extracting Rules. In *Fuzzy Systems, Proceedings of the Fifth IEEE International Conference on*, Vol. 1; IEEE: New York, NY, 1996.

[10] Tickle, A.; Golea, M.; Hayward, R.; Diederich, J. The Truth is in There: Current Issues in Extracting Rules from Trained Feedforward Artificial Neural Networks. In *Neural Networks, International Conference on*, Vol. 4; IEEE: New York, NY, 1997.

[11] Sato, M.; Tsukimoto, H. Rule Extraction from Neural Networks Via Decision Tree Induction. In *Neural Networks, International Joint Conference on*, Vol. 3; IEEE: New York, NY, 2001.

[12] Gupta, A.; Park, S.; Lam, S. Generalized Analytic Rule Extraction for Feedforward Neural Networks. *IEEE Transactions on Knowledge and Data Engineering* **1999,** *11,* 985–991.

[13] Chastrette, M.; Zakarya, D.; Peyraud, J. Structure-Musk Odor Relationships for Tetralins and Indans Using Neural Networks (On the Contribution of Descriptors to the Classification). *Eur. J. Med. Chem.* **1994,** *29,* 343–348.

[14] Hervas, C.; Silva, M.; Serrano, J. M.; Orejuela, E. Heuristic Extraction of Rules in Pruned Artificial Neural Network Models Used for Quantifying Highly Overlapping Chromatographic Peaks. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1576–1584.

[15] Mak, B.; Blanning, R. An Empirical Measure of Element Contribution in Neural Networks. *IEEE Trans. Systems, Man and Cybernetics C* **1998,** *28,* 561–564.

[16] 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.

[17] Haykin, S. *Neural Networks;* Peason Education: Singapore, 2$^{nd}$ ed.; 2001.

[18] Hornik, K. Some New Results on Neural Network Approximation. *Neural Networks* **1993,** *6,* 1069–1072.

[19] Garson, D. Interpreting Neural Network Connection Strengths. *AI Expert* **1991,** 47–51.

[20] Yoon, Y.; Guimaraes, T.; Swales, G. Integrating Artificial Neural Networks with Rule-Based Expert Systems. *Decision Support Sys.* **1994,** *11,* 497–507.

[21] Metropolis, N.; Rosenbluth, A.; Rosenbluth, M.; Teller, A.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953,** *21,* 1087–1092.

[22] Sutter, J.; Dixon, S.; Jurs, P. Automated Descriptor Selection for Quantitative Structure-Activity Relationships Using Generalized Simulated Annealing. *J. Chem. Inf. Comput. Sci.* **1995,** *35,* 77–84.

[23] Goll, E.; Jurs, P. Prediction of the Normal Boiling Points of Organic Compounds from Molecular Structures with A Computational Neural Network Model. *J. Chem. Inf. Comput. Sci.* **1999,** *39,* 974–983.

[24] Stanton, D.; Mattioni, B. E.; Knittel, J.; Jurs, P. Development and Use of Hydrophobic Surface Area (HSA) Descriptors for Computer Assisted Quantitative Structure-Activity and Structure-Property Relationships. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1010–1023.

[25] Patel, H.; Berge, W. t.; Cronin, M. Quantitative Structure-Activity Relationships (QSARs) for the Prediction of Skin Permeation of Exogenous Chemicals.. *Chemosphere* **2002,** *48,* 603–613.

[26] Mattioni, B. E. *The Development of Quantitative Structure-Activity Relationship Models for Physical Property and Biological Activity Prediction of Organic Compounds,* PhD thesis, Pennsylvania State University, 2003.

[27] Audus, K.; Chikhale, P.; Miller, D.; Thompson, S.; Borchardt, R. Brain Uptake of Drugs: The Influence of Chemical and Biological Factors. *Adv. Drug Res.* **1992,** *23,* 1–64.

[28] Gratten, J.; Abraham, M.; Bradbury, M.; Chadha, H. Molecular Factors Influencing Drug Transfer Across the Blood Brain Barrier. *J. Pharm. Pharmacol.* **1997,** *49,* 1211–1216.

[29] Ishibuchi, H.; Nii, M.; Tanaka, K. Fuzzy-Arithmetic-Based Approach for Extracting Positive and Negative Linguistic Rules from Trained Neural Networks. In *Fuzzy Systems, Proceedings of the IEEE International Conference on*, Vol. 3; IEEE: New York, NY, 1999.

[30] Chen, P.; Mills, J. Modeling of Neural Networks in Feedback Systems Using Describing Functions. In *Neural Networks, International Conference on*, Vol. 2; IEEE: New York, NY, 1997.

[31] Siu, K.-Y.; Roychowdhury, V.; Kailath, T. Rational Approximation, Harmonic Analysis and Neural Networks. In *Neural Networks, International Joint Conference on*, Vol. 1; IEEE: New York, NY, 1992.

[32] Fu, X.; Wang, L. Rule Extraction By Genetic Algorithms Based on a Simplified RBF Neural Network. In *Evolutionary Computation, Proceedings of the 2001 Congress on*, Vol. 2; IEEE: New York, NY, 2001.

[33] Stanton, D.; Jurs, P. Development and Use of Charged Partial Surface Area Structural Descriptors in Computer Assissted Quantitative Structure Property Relationship Studies. *Anal. Chem.* **1990,** *62,* 2323–2329.

[34] Balaban, A. Higly Discriminating Distance Based Topological Index. *Chem. Phys. Lett.* **1982,** *89,* 399–404.

[35] Kier, L.; Hall, L. *Molecular Connectivity in Chemistry & Drug Research;* Academic Press: New York, 1976.

[36] Kier, L.; Hall, L.; Murray, W. Molecular Connectivity I: Relationship to Local Anesthesia.. *J. Pharm. Sci.* **1975,** *64,*.

[37] Kier, L.; Hall, L. *Molecular Connectivity in Structure Activity Analysis.;* John Wiley & Sons: Hertfordshire, England, 1986.

[38] Kier, L.; Hall, L. Molecular Connectivity VII: Specific Treatment to Heteroatoms. *J. Pharm. Sci.* **1976,** *65,* 1806–1809.

[39] Randic, M. On Molecular Idenitification Numbers. *J. Chem. Inf. Comput. Sci.* **1984,** *24,* 164–175.