

Interpreting Computational Neural Network Quantitative Structure–Activity Relationship Models: A Detailed Interpretation of the Weights and Biases

Rajarshi Guha,[†] David T. Stanton,[‡] and Peter C. Jurs^{*,†}

Chemistry Department, Penn State University, University Park, Pennsylvania 16802, and Procter & Gamble, Miami Valley Laboratories, Cincinnati, Ohio 45252

Received April 6, 2005

In this work, we present a methodology to interpret the weights and biases of a computational neural network (CNN) quantitative structure–activity relationship model. The methodology allows one to understand how an input descriptor is correlated to the predicted output by the network. The method consists of two parts. First, the nonlinear transform for a given neuron is linearized. This allows us to determine how a given neuron affects the downstream output. Next, a ranking scheme for neurons in a layer is developed. This allows us to develop interpretations of a CNN model similar in manner to the partial least squares (PLS) interpretation method for linear models described by Stanton (Stanton, D. T. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1423–1433). The method is tested on three datasets covering both physical and biological properties. The results of this interpretation method correspond well to PLS interpretations for linear models using the same descriptors as the CNN models, and they are consistent with the generally accepted physical interpretations for these properties.

INTRODUCTION

The development of predictive statistical models is one of the fundamental tasks for a quantitative structure–activity relationship (QSAR) modeler. The statistical and machine learning literature provides a wide variety of methods to choose from. These include simple techniques such as linear regression models as well as 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, one 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 partial least squares (PLS)

technique described by Stanton.¹ In this case, the interpretation is detailed in the sense that one can determine the specific role a descriptor plays in the model and how structural feature changes correlate to differences in the property being modeled for specific compounds. A 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 forest⁴ models. Random forest models are based on the properties of an ensemble of recursive partitioning models.⁵ The interpretation of a model based on recursive partitioning is straightforward and comprehensive, resulting in a set of rules explaining the predicted activity or classification of the observations.⁶ As a result, such models can provide a detailed interpretation of the structure–activity trends in the dataset. However, the random forest approach considers collections of trees obtained by the recursive partitioning method. Each tree is grown to its full height, and no pruning⁵ is carried out. Thus, the individual trees in a random forest model are not necessarily optimal. Furthermore, each individual tree is developed using a random subset of descriptors from the descriptor pool. The net result of these two features is that, though the rules for each individual tree can be easily extracted, a coherent interpretation of the rules of all the trees comprising the random forest model is not necessarily possible, or even useful. However, methods have been described to provide some measure of interpretability. A global approach described by Breiman et al.⁵ is to develop a measure of descriptor importance for the whole forest. This summary ranks the input 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. Other methods focus on the extraction of representative trees or groups of trees from a forest, for modeling purposes. For

* Corresponding author e-mail: pcj@psu.edu; tel.: 814-865-3739; fax: 814-865-3314.

[†] Penn State University.

[‡] Procter & Gamble.

example, Hawkins and Musser⁷ analyzed a forest to understand which variables tend to occur together in individual trees. Other methods define tree metrics and attempt to visualize clusters of trees using a multidimensional scaling procedure.^{8,9} Thus, these approaches do not always consider the original forest as the model. In addition, a common feature of all these approaches is that they focus on specific trees (or groups of trees) to interpret the entire random forest and, thus, do not necessarily take into account all the information encoded within the forest. Thus, compared to the interpretability of linear regression models, that of random forests is not as detailed. Finally, in the case of computational neural network (CNN) models, interpretability, in general, is lacking.

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.^{10–12} These methods can be heuristic^{13–15} in nature or analytical.¹⁶ However, a number of these methods are focused on specific types of neural networks.^{12,13,17} Chastrette et al.¹⁷ describe a method for interpreting a CNN model describing structure–musk-odor relationships. However, their approach was limited to a measure of contribution of the descriptors to the predicted value. Hervás et al.¹⁸ describe a method interpretation that is focused 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.,¹⁷ 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 and Blanning¹⁹ 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.

In this paper, we describe a method to interpret a CNN model 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 three-layer, feed-forward, fully connected 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 and Blanning.¹⁹ The next section describes the methodology in detail.

METHODOLOGY

The CNN interpretation methodology was developed by attempting to mimic the procedure used for the interpretation

of linear models using PLS. The PLS method is summarized as follows.

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) that 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 properties for specific molecules.

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 three-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

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

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 $w_j^{H,H}$, $1 \leq j \leq n_H$, we can write X as

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

The above equation does not include a bias term, and we

provide a justification for ignoring the bias term below. Equation 1 can be rewritten as

$$O = \frac{1}{1 + \exp(-\sum_{j=1}^{n_H} w_j^H x_j^H)}$$

$$\frac{1}{O} \approx \exp(-\sum_{j=1}^{n_H} w_j^H x_j^H)$$

$$O \approx \exp(w_1^H x_1^H + \dots + w_{n_H}^H x_{n_H}^H) \quad (2)$$

where we drop the constant term because it does not affect the general trend between the output value and the exponential term. From eq 2, we can see 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 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 on the basis of 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.

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 realize that the weights are cumulative. We denote the weights between the input layer neuron j and the hidden layer neuron i as w_{ij} , where $1 \leq i \leq n_I$ 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 th input neuron and the output neuron, via the j th hidden layer neuron, will be $w_{ij}w_j^H$. Clearly, the effective weights are gross simplifications, because we neglect the intermediate summations (over all neurons in a layer) and also the transfer functions. However, as shown in the previous section, we can see that the output value of the transfer function is a monotonic

Table 1. Tabular Representation of the Effective Weights for a Hypothetical 4–3–1 CNN Model^a

	hidden neuron		
	H1	H2	H3
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	$w_{41}w_1^H$	$w_{42}w_2^H$	$w_{43}w_3^H$

^a I1, I2, I3, and I4 represent the four input neurons (descriptors). w_{ij} represents the weight for the connection between the i th input neuron and the j th hidden neuron. w_j^H represents the weight between the j th hidden neuron and the output neuron. For this example, i ranges from 1 to 4 and j ranges from 1 to 3.

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 than 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 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, we expect that the net effect would be a negative correlation between the input and output values.

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 1, where H1, H2, and H3 represent the first, second, and third hidden neurons, respectively, 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 neuron. Because 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.

The Bias Term. In the preceding discussion, we ignored the role of the bias term when we constructed the effective weights. We now provide a justification for this decision. The input to each hidden neuron consists of the weighted outputs of all 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 th hidden neuron would include b/n_I , where b_j is the bias term for that hidden neuron. The net result is that for the j 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 the fact that a bias term is effectively an intercept term. If the hidden neurons contained linear activation functions, the bias term is precisely an intercept term. The inputs to a hidden neuron forms 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 the bias term is included, it merely translates the hyperplane from the origin. That is, it does not change the form of the hyperplane. In effect, the bias term is a constant. Now, one property of neural networks (more specifically, multilayer perceptrons) is universal function approximation.²⁰ For this to be true, the bias term must be included. However, it has been shown by Hornik²¹ 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 nonzero 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.

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 Garson²² calculates a measure of the relative contribution of the i th input neuron to the k th output neuron and places more stress on the connections between the hidden and output layers. Yoon et al.²³ extended this approach but still focused on contributions

of input descriptors to the output via the hidden layer neurons. A common feature of these 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 th hidden neuron for the k th example in the training set to be

$$\text{SRC}_{kj} = \frac{(x_{kj}^H w_j^H)^2}{\sum_{j=1}^{n_H} (x_{kj}^H w_j^H)^2 + b^2}$$

where x_{kj}^H and w_j^H are the output and weight to the output neuron of the j th hidden (for the k th example) neuron, respectively, b is the bias term for the output neuron, and n_H is the number of hidden layer neurons. The final value of the squared relative contribution for the j th hidden neuron was given by

$$\text{SRC}_j = \frac{1}{n} \sum_{k=1}^n \text{SRC}_{kj}$$

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 depend 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 th hidden neuron was initially defined as

$$\text{CV}_j = \frac{1}{n_I} \sum_{i=1}^{n_I} w_{ij} w_j^H \quad (3)$$

where n_I is the number of input neurons, w_{ij} is the weight between the i th input neuron and the j th hidden neuron, and w_j^H is the weight between the j 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 values of these contributions 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 3 as squared contribution values (SCVs) defined as

$$\text{SCV}_j = \frac{\text{CV}_j^2}{\sum_{j=1}^{n_H} \text{CV}_j^2} \quad (4)$$

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, because we are focusing on the contribution made by a given hidden neuron, we need to consider all the effects acting through the hidden neuron. Because the bias terms corresponding to the hidden layer can be considered as weights coming from an extra (constant) input neuron, the effective weights being summed in eq 3 should include an extra term corresponding to the bias term for that hidden neuron. Thus, if we denote the bias term for the j th hidden neuron as b_j , then eq 3 can be rewritten as

$$CV_j = \frac{1}{n_I + 1} \left(\sum_{i=1}^{n_I} w_{ij} w_j^H + b_j w_j^H \right) \quad (5)$$

Using this equation, values for SCV_j can be calculated using eq 4.

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 the CNN and linear models should capture the structure–property trends present in a 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 is meant to focus 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. This involved the use of a simulated annealing^{24,25} algorithm to search for good descriptor subsets. The objective function for the algorithm was a linear regression algorithm. The algorithm has an option to reject models having a t statistic less than 4.0. However, this sometimes leads to poorer models and, hence, was not used in this study. The search algorithm found a number of possible models. The final model was then selected on the basis of R^2 and root-mean-square error (RMSE) values. The linear regression models were then interpreted using the PLS analysis

technique described above. It should be noted that though PLS models are generally used for their ability to handle a large number of predictors in terms of a reduced number of latent variables, the usage of the PLS algorithm in this context is to extract information from the original linear regression models and not to provide an alternative predictive model. 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 a cost function defined as

$$\text{Cost} = \text{TSET}_{\text{RMSE}} + 0.5|\text{TSET}_{\text{RMSE}} - \text{CVSET}_{\text{RMSE}}|$$

where $\text{TSET}_{\text{RMSE}}$ and $\text{CVSET}_{\text{RMSE}}$ represent the RMSEs for the training and cross-validation sets, respectively. The architecture that gave the lowest value of the cost function was chosen as the best CNN model for that dataset.

Datasets. We considered three datasets. The first dataset consisted of a 147-member subset of the Design Institute for Physical Property Data (DIPPR) boiling point dataset studied by Goll and Jurs.²⁶ The dependent variable (normal boiling point) for this dataset ranged from 145.1 to 653.1 K. The original work reported linear and nonlinear models. However, no interpretations of these models were provided. Our previous work on descriptor importance²⁷ reported a new linear model developed using this dataset. Although that work also reported a CNN model, we developed 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 molecules studied by Stanton et al.²⁸ These molecules were studied for their ability to cross the blood–brain barrier (BBB), and the modeled property was the logarithm of the blood–brain partition coefficient, $\log(\text{BB})$. The dependent variable in this dataset ranged from -2.00 to $+1.44$ log units. The work reported a linear model and an associated PLS-based interpretation.

The third dataset consisted of 136 molecules studied by Patel et al.²⁹ The work considered the skin permeability of the 136 compounds. The dependent variable for this dataset was the logarithm of the permeability coefficient, $\log(K_p)$, which ranged from -5.03 to -0.85 log units. Although the paper reported a set of linear models, we developed a new linear model using a variety of descriptors including hydrophobic surface area descriptors.^{28,30} A PLS analysis of this model is also presented for comparison to the interpretation of the corresponding CNN model.

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 we then present the CNN interpretation. For all models, the descriptors utilized are summarized in Table 2.

DIPPR Boiling Point Dataset. The statistics of the seven-descriptor linear model are summarized in Table 3. An

Table 2. Glossary of the Descriptors Reported

descriptor code	meaning	reference
DPHS	the difference between the hydrophobic and hydrophilic surface areas	28, 30
FPSA-2	charge-weighted partial positive surface area divided by the total surface area	37
MOLC-9	Balaban distance connectivity index J	38, 39
MW	molecular weight	
NDB	number of double bonds	
NN	number of nitrogens	
PNHS-3	atomic-constant-weighted hydrophilic surface area	28, 30
PPHS	total molecular hydrophobic surface area	28, 30
SA	surface area of the molecule	
S4PC	fourth-order simple path cluster molecular connectivity index	40–42
V4P	fourth-order valence-corrected path molecular connectivity index	40–42
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	37
WPHS-3	surface-weighted hydrophobic surface area	28, 30
WTPT-2	molecular ID divided by the total number of atoms	43
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	28, 30
RSHM	Fraction of the total molecular surface area associated with hydrogen bond acceptor groups	37

Table 3. Summary of the Linear Regression Model Developed for the DIPPR Boiling Point 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	19.57	3.30	5.92
S4PC	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 4. Summary of the PLS Analysis Based on the Linear Regression Model Developed for the DIPPR Boiling Point Dataset

components	error SS	R^2	PRESS	Q^2
1	94 868.50	0.86	99 647.60	0.85
2	26 221.60	0.96	29 046.70	0.95
3	16 614.80	0.97	19 303.30	0.97
4	14 670.80	0.97	17 027.60	0.97
5	14 032.50	0.97	16 281.30	0.97
6	13 775.90	0.98	15 870.60	0.97
7	13 570.90	0.98	15 653.00	0.97

explanation of the descriptors is given in Table 2. $R^2 = 0.98$ and RMSE = 9.98 K. The F statistic (for 7 and 139 degrees of freedom) was 1001, which is much greater than the critical value of 2.78 ($\alpha = 0.05$ level). The model is, thus, statistically valid. Tables 4 and 5 summarize the results of the PLS analysis for the linear model. The Q^2 column in

Table 5. X-Weights for the PLS Components from the PLS Analysis Summarized in Table 4

descriptor	component						
	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	0.48	−0.15	−0.07	−0.66	−0.36	−0.09	0.38
S4PC	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 6. Summary of the Architectures and Statistics for the CNN Models Developed for the Datasets Considered in This Study^a

dataset	architecture	RMSE			R^2		
		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

^a In all cases, the input descriptors were the same as those used in the corresponding linear models.

Table 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 5, we see that the most important descriptors are MW (molecular weight) and V4P (fourth order valence path connectivity index). Both of 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 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 compounds with smaller values of this 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 compounds 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 SPRs. The first trend focuses on dispersion forces, in which atomic contributions to these forces are individually weak, but for larger compounds, the greater number of interactions leads to a larger attractive force. The other main trend focuses on 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 best CNN model developed for this dataset had a 7–4–1 architecture. The seven descriptors used 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 6. The effective weight matrix for this model is shown in Table 7. The columns correspond to the hidden neurons and are ordered by the SCV values, which are shown in the last row of the table. The SCV values indicate that the first and third hidden neurons are the most

Table 7. Effective Weight Matrix for the 7–4–1 CNN Model Developed for the DIPPR Dataset^a

	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	9.45	2.15	3.24	0.60
S4PC	3.36	2.73	1.99	0.56
MW	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

^a The columns (hidden neurons) are ordered by the SCVs shown in the last row.

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 7, we see that the most heavily weighted descriptors are V4P, RSHM, and MW. All three descriptors have positive weights indicating that these descriptors are positively correlated with the boiling point. When we consider the second column (the third hidden neuron), we see that two of the three most important descriptors are the same as those in the first hidden neuron. Because these have the same signs as before, we can 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 those 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 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 and MW descriptors are swapped in the two interpretations, but because both characterize size, the main SPR trends for the dataset are explained in a similar fashion by both models. These differences are not unexpected, because the CNN correlates the descriptors to the boiling point in a nonlinear relationship. 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 and, in both cases, are consistent with chemical reasoning in the context of boiling points.

BBB Dataset. The linear model and associated PLS interpretation are described in the original work.²⁸ However, we summarize the statistical results of the original model in Table 8. $R^2 = 0.78$ and the F statistic was 80.7 (for 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 and 10. From Table 9, we see that

Table 8. Summary of the Linear Regression Model Developed for the BBB Dataset

	estimate	std. error	<i>t</i>
(intercept)	0.53	0.07	7.28
WNSA-3	0.04	0.01	6.24
V4P	0.24	0.03	7.13
NDB	−0.13	0.03	−5.05
PNHS-3	0.03	0.00	6.93

Table 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 10. X-Weights for the PLS Components from the PLS Analysis Summarized in Table 9

descriptor	component			
	1	2	3	4
WNSA-3	0.54	−0.13	0.79	0.28
V4P	−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

the first two components explain 76% of the total variance in the observed property, thus allowing us to ignore the remaining components. From Table 10, we see that in the first component, the most heavily 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). Larger values of PNHS-3 and WNSA-3 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 overpredictions made by the first component. Similar reasoning can be applied to the weight for V4P in the second component. Some large hydrophobic compounds are underestimated by component 1. In component 2, however, the positive weight for V4P (which is a measure of branching and, thus, size) indicates that larger compounds will have a higher penetration ability. Therefore, the second component makes up for the underestimation of large hydrophobic compounds by the first one. In brief, the SPR trends captured by the linear model indicate that smaller hydrophobic compounds will better penetrate the BBB compared to larger hydrophilic compounds. These trends are discussed in more detail in the original work.

The CNN model developed for this dataset had a 4–4–1 architecture. The statistics for this model are presented in Table 6, and the effective weight matrix is shown in Table 11. The SCV values for the hidden neurons are shown in the last row of Table 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 those 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 is one of

Table 11. Effective Weight Matrix for the 4–4–1 CNN Model Developed for the BBB Dataset^a

	hidden neuron			
	1	2	4	3
WNSA-3	52.41	29.30	−19.64	2.26
V4P	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

^a The columns are ordered by the SCVs for the hidden neurons, shown in the last row.

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 CNN model than in the linear model. If we consider the second hidden neuron (second column), we see that the weight for V4P 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. When we consider the fourth hidden neuron, we see that the V4P now has a negative weight and, thus, serves to balance the overestimation of the property for larger compounds 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 a 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 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) 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 descriptions of the main SPR trends encoded within the CNN model compare well with those of the linear model serves 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 respective interpretation techniques are consistent with physical descriptions of the factors that are believed to affect the transport of drugs across the blood–brain barrier.^{31,32}

Skin Permeability Dataset. This dataset was originally studied by Patel et al.,²⁹ where they developed a linear regression model using 158 compounds. However, because of 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 12. $R^2 = 0.84$ 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 developed a PLS interpretation of this linear model, and the results of the PLS model are summarized in Tables 13 and 14. The Q^2 values in Table 13 indicate that the first three components describe the bulk of the SPR. Table 14 shows that the most important descriptors in the first component are MOLC-9, FPSA-2, and RNHS. MOLC-9

Table 12. Summary of the Linear Regression Model Developed for the Skin Permeability Dataset

	estimate	std. error	t
(intercept)	−5.48	0.25	−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	0.01	0.00	13.47
WPHS-3	−0.02	0.00	−5.41
RNHS	0.05	0.01	7.48

Table 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 14. X-Weights for the PLS Components from the PLS Analysis Summarized in Table 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

represents Balaban distance connectivity index J , which is derived from the distance connectivity matrix and characterizes molecular size and 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 compounds with smaller partial positive surface areas will be more active. This descriptor characterizes molecules such as 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 15). 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 (because of the partial positive charges on the methyl groups) would lead to a significantly lower skin permeability. However, RNHS indicates that not all small hydrophobic compounds will have negatively charged atoms. Thus, the first component indicates that smaller and more hydrophobic compounds will exhibit greater permeability.

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 skin permeability. SA, which encodes molecular surface area, is positively weighted, indicating that

Table 15. Comparison of Compounds Exhibiting High and Low Skin Permeabilities to Illustrate the SPR Encoded by Component

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

Table 16. Comparison of Compounds with High and Low Skin Permeabilities, Predicted by the Second PLS Component

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

large compounds can exhibit significant permeability. The role of this component is to account for some larger hydrophobic compounds that are observed to permeate skin more readily than is predicted by the first component (Table 16). At the same time, component 2 also shows that small compounds will penetrate skin less well if they are moderately hydrophilic. The second component, thus, corrects for the underestimation of the larger compounds by component 1 (such as fentanyl and sulfentanil) by taking into account

Table 17. Effective Weight Matrix for the 7–5–1 CNN Model Developed for the Skin Permeability Dataset^a

	hidden neurons				
	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

^a The columns are ordered by the SCVs for the hidden neurons, shown in the last row.

their higher hydrophobicity and also corrects for the overestimation of smaller compounds (such as methanol and urea) by component 1 by taking into account their lower hydrophobicity.

The third component mainly corrects for the overestimation of hexachlorobutadiene and hexachloroethane by component 2 due to an 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 6. The effective weight matrix is shown in Table 17. In the case of this model, we see that the SCV value for the fifth hidden neuron is nearly six times larger than that for the next most important neuron. If we consider the most important hidden neuron, 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 those 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 compounds with a smaller polar surface area and a larger hydrophobic surface area will exhibit greater permeability. Moving on to the next most important hidden neuron, we see that the most weighted descriptors are SA, PPHS, and MOLC-9. It is clear that this hidden neuron focuses on size and branching effects. However, the negative weight for surface area indicates that larger compounds will exhibit higher permeability. This is a valid conclusion because the dataset does indeed have some larger compounds that are moderately permeable. This conclusion is further justified by the fact that a larger compound would have a correspondingly larger hydrophobic surface area, which, as the positive weight for PPHS indicates, will lead to greater permeability. At the same time, all large compounds do not exhibit high skin permeability. Thus, the effect of the SA descriptor is balanced by the positive weight for the MOLC-9 descriptor. Because larger values of MOLC-9 correlate to more branched compounds, the effect of the MOLC-9 descriptor balances the SA descriptor, ensuring that this hidden neuron does not predict that all large compounds will exhibit high permeation.

In the next most important hidden neuron (number 4), we see that the most weighted descriptors are RNHS, FPSA-2, PPHS, and MOLC-9. For this hidden neuron, MOLC-9 describes the effect of size and indicates that compounds with a higher degree of branching will exhibit greater permeability. However, the positive-weight FPSA-2 indicates that compounds with larger partially positive-charged surface area will exhibit greater permeability. 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 compounds. This trend is reinforced to some extent by the negative weight for RNHS. RNHS describes both hydrophilic and partial negatively charged regions of a compound. Because of the design of the descriptor, the negative sign on this descriptor indicates that compounds with a smaller partial negatively charged surface area and more hydrophilic atoms will exhibit relatively greater permeability. 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 17.

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 because of the nonlinear encoding of the structure property relationships by the CNN model. Although the above description is quite detailed and allows us to look at descriptor values for individual observations and understand why they are predicted to display greater or lesser skin permeation, a visual approach to understanding the effects of each hidden neuron, analogous to score plots¹ in the PLS interpretation scheme, would be useful. One approach to this problem is to generate plots using the effective weights.

Score Plots. As described above, the use of 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*th member of the dataset, the score for that member using the *j*th hidden neuron can be defined as

$$\text{score}_{kj} = \sum_{i=1}^{n_i} w_{ij} w_j^H x_{ki}$$

where $w_{ij} w_j^H$ is simply the effective weight for the *i*th input neuron (see Table 1), n_i is the number of input descriptors, and x_{ki} is the value of the *i*th descriptor for the *k*th member of the dataset. The result of eq 6 is that for each hidden neuron, a set of scores are obtained for the dataset. Clearly, these are not expected to quantitatively model the observed property well. However, our interest lies in the qualitative behavior of the scores. That is, we expect that if the observed activity of a compound is high, 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 activity should lie along the 1:1 line.

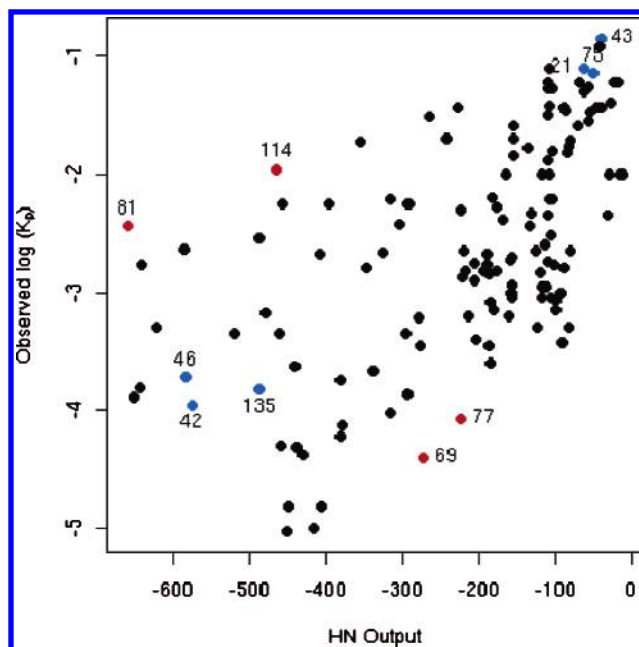


Figure 1. Score plot for the fifth hidden neuron (SCV = 0.85). Points colored red are examples of molecules mispredicted by this neuron. Points colored blue are examples of molecules well-predicted by this neuron.

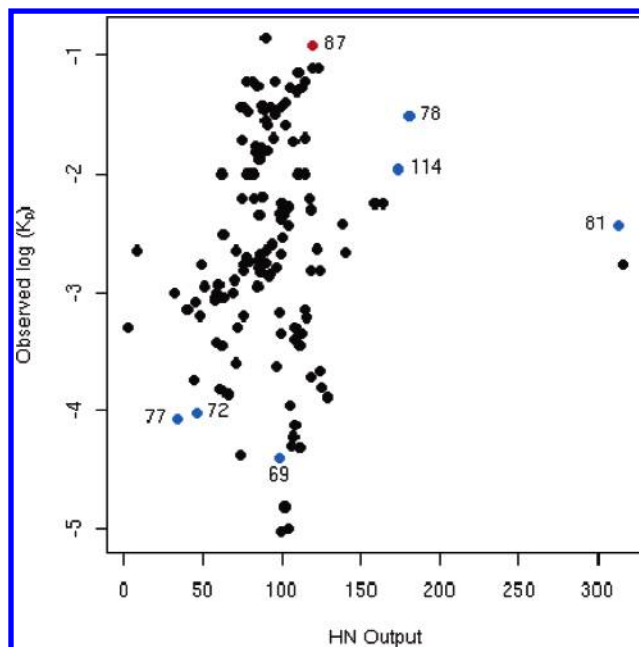


Figure 2. Score plot for the second hidden neuron (SCV = 0.13). Points colored red are examples of molecules mispredicted by this neuron. Points colored blue are examples of molecules well-predicted by this neuron.

Points lying in the lower right quadrant would represent compounds overestimated by the hidden neuron, and points lying in the upper-left quadrant would represent compounds underestimated 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 Figures 1, 2, and 3. Considering the plot for the fifth hidden neuron, we see that it does exhibit the behavior we expect. Compounds 21, 43, and 75 are correctly predicted as having high permeabilities, and 42, 46, and 135 are correctly predicted as having low permeabilities. The struc-

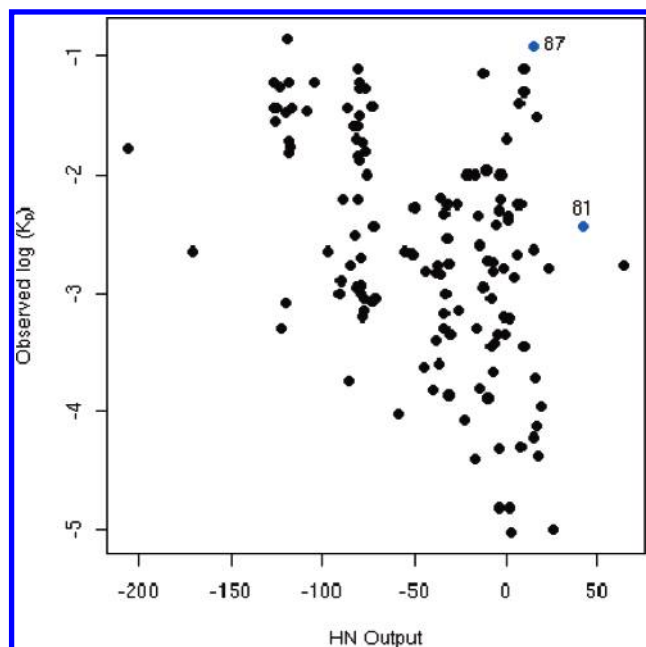
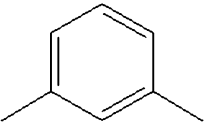
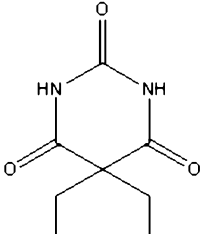
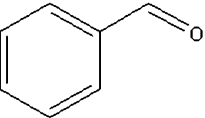
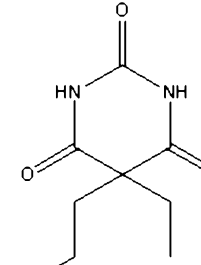
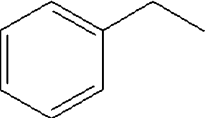
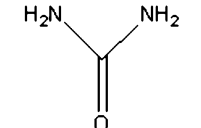


Figure 3. Score plot for the fourth hidden neuron (SCV = 0.02). Points colored blue are examples of molecules well-predicted by this neuron.

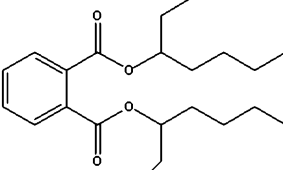
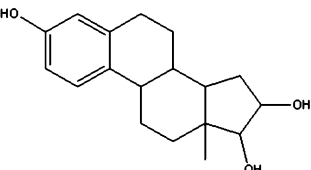
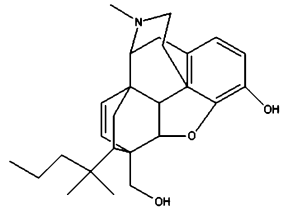
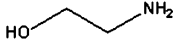
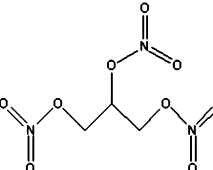
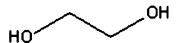
Table 18. Comparison of Structures Illustrating Compounds with High and Low Skin Permeabilities, Predicted by the Fifth Hidden Neuron^a

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

^a The bold number is the serial number, and the number in parentheses is the measured permeability coefficient.

tures for these compounds are shown in Table 18. As described previously, compounds exhibiting higher skin

Table 19. Comparison of Structures, Illustrating Compounds with Moderate and Low Skin Permeabilities, Predicted by the Second Hidden Neuron^a

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

^a The bold number is the serial number, and the number in parentheses is the measured permeability coefficient.

permeation will be characterized by smaller size and increased hydrophobicity. As Table 18 shows, these types of compounds do indeed have a hydrophobic benzene ring. In contrast, compounds with lower skin permeability are generally larger and, more significantly, have a number of polar regions. An interesting case is compound **135**. This is a very small compound, but it is dominated by polar groups. The fact that the neuron predicts this correctly as inactive is due to the fact that this neuron stresses the FPSA-2 and NN descriptors. As seen from Table 17, the negative weights indicate that a larger number of polar groups would reduce skin permeation. As a result, although compound **135** is small, the polar effects outweigh the size effect. Figure 1 also indicates that compounds **81**, **114**, **69**, and **77** are all mispredicted. The first two are overestimated and the last two are underestimated.

If we now consider the score plot for the second hidden neuron (Figure 2), we see that the four mispredicted compounds mentioned above are now more correctly predicted. However, **81** does appear to be overestimated. However, 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 compared to the fifth hidden neuron. Because of the very low SCV value for this neuron, we believe that it does not have significant explanatory power. The structures of the compounds exhibiting high and low skin permeabilities are compared in Table 19. As described above, the main focus of the second hidden neuron is to account for compounds that are relatively large but also moderately permeable. As can be seen from the structures, although compounds **78**, **81**, and **114** are significantly larger

than the permeable compounds in the preceding component, they are indeed moderately permeable. Correspondingly, this hidden neuron is also able to account for the low observed permeability of a number of small compounds (**72** and **77**). Though this hidden neuron mispredicts a number of compounds, the majority have already been 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 fourth hidden neuron, we see that, though it does correctly predict a number of compounds as having high permeability, it performs poorly on compounds with low observed permeability. 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 second 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 analyzed, though we observed that they did not explain any significant trends and, rather, corrected for a few mispredictions by the preceding three 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.

DISCUSSION

The CNN interpretation methodology 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). Although 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 SPRs, 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 including 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 with each other and are consistent with physical analyses of the factors that affect the properties modeled in this study. The main differences between the interpretations are 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 because the neural network combines the input descriptors nonlinearly, and thus, the role of the individual descriptors in the nonlinear relationship can be different from that played in a linear relationship.

Another important aspect of this study was that we considered CNN models that 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 can 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 because of the simple nature of the 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^{12–15,33} as well as some instances of analytical rule extraction.^{16,34,35} 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 analyzing the neural network with the help of a genetic algorithm^{33,36} or by decision trees,¹⁵ 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 paper can be extended to the case of multiple

hidden layers, though the complexity of the treatment will correspondingly increase.

CONCLUSIONS

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 usage mainly 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.

REFERENCES AND NOTES

- (1) Stanton, D. T. 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 nonlinear 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. Random forests. *Mach. Learning* **2001**, *45*, 5–32.
- (5) Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. *Classification and Regression Trees*; Wadsworth: Belmont, CA, 1984.
- (6) Rusinko, A. I.; Farnen, M. W.; Lambert, C. G.; Brown, P. L.; Young, S. S. Analysis of a Large Structure/Biological Activity Data Set using Recursive Partitioning. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1017–1026.
- (7) Hawkins, D. M.; Musser, B. J. One Tree or a Forest? Alternative Dendrographics Models. *Comput. Sci. Stat.* **1998**, *30*, 543–542.
- (8) Chipman, H. A.; George, E. I.; McCulloch, R. E. Making Sense of a Forest of Trees. *Comput. Sci. Stat.* **1998**, *30*, 84–92.
- (9) Urbanek, S. Exploring Statistical Forests, In *Proc. 2002 Joint Statistical Meeting*, 2002; *Mira DP*, in press.
- (10) Castro, J. L.; Mantas, C. J.; Benitez, J. M. Interpretation of artificial neural networks by means of fuzzy rules. *IEEE Trans. Neural Networks* **2002**, *13*, 101–116.
- (11) Limin, F. Rule generation from neural networks. *IEEE Trans. Syst., Man Cybernetics* **1994**, *24*, 1114–1124.
- (12) Bologna, G. Rule extraction from linear combinations of dimlp neural networks. In *Proc. Sixth Brazilian Symposium on Neural Networks*; 2000.
- (13) Yao, S.; Wei, C.; He, Z. Evolving fuzzy neural networks for extracting rules. In *Fuzzy Systems, Proc. Fifth IEEE Intl. Conf.*; 1996.
- (14) Tickle, A. B.; 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, Intl. Conf.*; 1997.
- (15) Sato, M.; Tsukimoto, H. Rule extraction from neural networks via decision tree induction. In *Neural networks, Intl. Joint Conf.*; 2001.
- (16) Gupta, A.; Park, S.; Lam, S. M. Generalized analytic rule extraction for feedforward neural networks. *IEEE Trans. Knowledge Data Eng.* **1999**, *11*, 985–991.
- (17) Chastrette, M.; Zakarya, D.; Peyraud, J. F. 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.
- (18) Hervás, 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.
- (19) Mak, B.; Blanning, R. W. An empirical measure of element contribution in neural networks. *IEEE Trans. Syst., Man Cybernetics C* **1998**, *28*, 561–564.
- (20) Haykin, S. *Neural Networks*; Prentice Hall: New York, 2001.
- (21) Hornik, K. Some new results on neural network approximation. *Neural Networks* **1993**, *6*, 1069–1072.
- (22) Garson, D. Interpreting neural network connection strengths. *AI Expert* **1991**, 47–51.
- (23) Yoon, Y.; Guimaraes, T.; Swales, G. Integrating artificial neural networks with rule-based expert systems. *Decis. Support Syst.* **1994**, *11*, 497–507.
- (24) Sutter, J. M.; Dixon, S. L.; Jurs, P. C. Automated descriptor selection for quantitative structure–activity relationships using generalized simulated annealing. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 77–84.
- (25) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M.; Teller, A. H.; Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092.
- (26) Goll, E. S.; Jurs, P. C. 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.
- (27) Guha, R.; Jurs, P. C. Interpreting Computational Neural Network QSAR Models: A Measure of Descriptor Importance. *J. Chem. Inf. Model.* **2005**, *45*, 800–806.
- (28) Stanton, D. T.; Mattioni, B. E.; Knittel, J. J.; Jurs, P. C. 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.
- (29) Patel, H.; Berge, W. T.; Cronin, M. T. D. Quantitative structure–activity relationships (QSARs) for the prediction of skin permeation of exogenous chemicals. *Chemosphere* **2002**, *48*, 603–613.
- (30) Mattioni, B. E. The development of quantitative structure–activity relationship models for physical property and biological activity prediction of organic compounds. Ph.D. Thesis, Pennsylvania State University, University Park, PA, 2003.
- (31) Gratten, J. A.; Abraham, M. H.; Bradbury, M. W.; Chadha, H. S. Molecular factors influencing drug transfer across the blood/brain barrier. *J. Pharm. Pharmacol.* **1997**, *49*, 1211–1216.
- (32) Audus, K. L.; Chikhale, P. J.; Miller, D. W.; Thompson, S. E.; Borchardt, R. T. Brain uptake of drugs: The influence of chemical and biological factors. *Adv. Drug Res.* **1992**, *23*, 1–64.
- (33) Ishibuchi, H.; Nii, M.; Tanaka, K. Fuzzy-arithmetic-based approach for extracting positive and negative linguistic rules from trained neural networks. In *Fuzzy Systems Conference Proc., IEEE International*; 1999.
- (34) Chen, P. C. Y.; Mills, J. K. Modeling of neural networks in feedback systems using describing functions. In *Neural networks, International Conference*; 1997.
- (35) Siu, K.-Y.; Roychowdhury, V.; Kailath, T. Rational approximation, harmonic analysis and neural networks. In *Neural networks, Intl. Joint Conf.*; 1992.
- (36) Fu, X.; Wang, L. Rule extraction by genetic algorithms based on a simplified RBF neural network. In *Evolutionary computation, Proc. of the 2001 Congress on*; 2001.
- (37) Stanton, D. T.; Jurs, P. C. Development and use of charged partial surface area structural descriptors in computer assisted quantitative structure property relationship studies. *Anal. Chem.* **1990**, *62*, 2323–2329.
- (38) Balaban, A. T. Highly discriminating distance based topological index. *Chem. Phys. Lett.* **1982**, *89*, 399–404.
- (39) Kier, L. B.; Hall, L. H. *Molecular Connectivity in Chemistry and Drug Research*; Academic Press: New York, 1976.
- (40) Kier, L. B.; Hall, L. H.; Murray, W. J. Molecular Connectivity I: relationship to local anesthesia. *J. Pharm. Sci.* **1975**, *64*.
- (41) Kier, L. B.; Hall, L. H. *Molecular connectivity in structure activity analysis*; John Wiley & Sons: New York, 1986.
- (42) Kier, L. B.; Hall, L. H. Molecular Connectivity VII: specific treatment to heteroatoms. *J. Pharm. Sci.* **1976**, *65*, 1806–1809.
- (43) Randic, M. On molecular identification numbers. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 164–175.

CI050110V