

Neural Models and Extracted Rules for Knowledge Discovery in Predictive Toxicology

Dennis Bahler and Brian Stone

Artificial Intelligence Laboratory
 Department of Computer Science
 North Carolina State University
 Raleigh NC 27695-8206 USA

Abstract— We are using neural networks as a tool for predicting chemically-induced carcinogenesis in rodents by training on data derived from a long series of expensive and time-consuming animal tests. Neural networks have shown to be a capable model for accomplishing this task, providing results as good or better than other approaches to the same problem. A new approach to relevant feature subset selection is presented which uses the connection weights of a trained network to assign relevance weights to the attributes; a threshold is then determined by hill climbing. Our Single Hidden Unit Method is shown to provide good results in reasonable time compared with other feature selection methods. Once a network was trained, its weight matrix was pruned in anticipation of rule extraction. Our iterative method is shown to be capable of pruning roughly three-fourths of the connections while improving accuracy. Finally, rule extraction is investigated as a means for networks to explain themselves. A brute force approach to rule extraction in which all possible inputs are listed as rules and the rules are then collapsed to M -of- N rules is shown to build a reasonably small rule set that only suffers a small drop in accuracy from the neural network. An algorithm is presented for the brute force approach which allows it to finish in reasonable time. The set of 22 M -of- N rules so derived are readable and useful for describing the knowledge learned by the network in terms that humans can understand. By applying these new tools to the field of predictive toxicology, a network is trained that is estimated to have good predictive accuracy relative to other efforts in this field. In addition, the results from feature selection and the extracted rules provide new information to predictive toxicologists that is interesting because of the new approach, provocative results, and potential for pointing the way toward new insights in the field.

I. INTRODUCTION

This paper will show how a neural network can be trained to predict chemical carcinogenesis in rodents in such a way that the results are easily comprehensible by human experts. We construct our models by training on a data set compiled since 1978 by the National Cancer Institute (NCI) and the National Toxicology Program (NTP). This data is described more fully in Section II. There is rapidly increasing current interest in predictive toxicology and in using automated learning and knowledge discovery methods in predictive toxicology [3], [5], [16], [18]. The results from applying neural networks to this problem contribute to the ongoing process of evaluating and interpreting the data collected from chemical carcinogenesis studies.

Neural networks have several inherent advantages over

other machine learning tools. First, they do not assume any underlying model. During the training process, a neural network will form its own associations strictly from the training data and is not biased by or reliant on expert knowledge. Other well-known advantages include a graceful degradation in performance when there is missing data and good performance on noisy data. Finally, the accuracy of trained neural networks has been shown to match or exceed other machine learning approaches in a wide variety of test applications [20].

Unfortunately for the purposes of knowledge discovery, neural networks also suffer several disadvantages. First, data representation can be tricky since networks require the input and present the output in a normalized numerical form. Second, there is no easy way to determine the ideal network architecture for a given application. Third, training using backpropagation is slow and may converge to local (i.e., suboptimal) minima. Fourth, without careful controls, networks may overfit the training data. Fifth, the internal structure of even relatively small networks is extremely difficult to interpret. Once trained, neural networks can provide no meaningful explanation for their classifications without further processing.

This paper will show how a neural network approach can overcome these difficulties so as to predict chemical carcinogenesis in rodents while providing useful explanations to domain specialists.

The ultimate goal of applying any machine learning tool is to construct a model that both represents the data as accurately and concisely as possible and is able to explain its results in terms that humans can easily understand. In the case of our toxicology application, for example, such a model would take as input the test results in lab animals together with characteristics of the test article (i.e., chemical compound, polymer, or mixture) itself, and make a decision as to whether or not the test article is a carcinogen.

When exploring the viability of a method for constructing a model, there are three goals or measures of performance that must be considered. The first is the level of human guidance that is required. Such guidance may take the form of parameter setting or initializing a learner with expert knowledge. In any case, the best learner would minimize the amount of human guidance required. Second, one must consider the amount of generalization the learner is

capable of. Performance must be measured not just against the training data, but also against unseen data from the same domain. The performance of a model on unseen data can be estimated using techniques of cross validation. Finally, these algorithms should be measured by the ability of the constructed models to explain the knowledge that they contain. The better a model is at explaining its results, the more useful it will be to humans. The explanation capability may be in the form of a simple list of IF-THEN rules, or it may involve a more sophisticated Query/Response based explanation of the model’s behavior. In any event, the explanation should accurately and precisely identify the reasons underlying any output the model generates in terms that can be understood by humans with knowledge of the application domain.

The remainder of this paper examines the application of neural networks to predictive toxicology in light of how closely such models can be adapted to achieve these three goals, and is organized as follows. Section II describes the training data. Section III describes the architecture of the neural networks we use. A method is shown for mapping the features in the data to the input layer and for mapping the output of the network to different classifications. A modification to an existing formula is presented for determining the optimum size of the hidden layer in a three-layer fully-connected network. Combining early stopping during training with a careful choice of hidden layer size helps to prevent the trained network from overfitting. Using an inertia term during backpropagation helps avoid convergence on local minima.

Section IV presents a new approach to automated feature selection we have found more suitable to this application than existing methods. Our approach allows the machine to identify, independent of expert knowledge, which features in the training data are most relevant to learning. It also significantly improves the predictive accuracy of neural models on the data set. Section V describes different methods for pruning unnecessary weights in a trained network. This process speeds up the performance of the network and simplifies its internal structure, thereby making rule extraction easier. Connection weight pruning also actually improves accuracy slightly in our application. Section VI presents a new method for extracting *M-of-N* rules from networks, and a set of 22 *M-of-N* rules is extracted from a network trained on the NTP data using feature selection and whose weights have been pruned. By presenting an effective method for rule extraction the status of neural networks is elevated from ‘black boxes’ to machine learning tools that are capable of explaining themselves and justifying their predictions.

We give results of our experiments in Section VII and conclude in Section VIII.

II. THE TRAINING DATA

The data used for training and testing the neural networks comes from carcinogenicity studies begun in the late 1960’s by the National Cancer Institute (NCI) and conducted since 1978 by the National Toxicology Program

(NTP) [10]. The data is classified according to level of evidence of carcinogenicity and contains bioassay results on more than 400 long-term chemical carcinogenesis studies, comprising nearly 1600 individual experiments in which a test article was administered in various doses to cohorts of test animals for a period of 104 weeks via one of five route of administration [12].

A large part of the information available on test articles is results of *in vitro* microbial assays for mutagenesis in salmonella cultures. The results of the salmonella assays are reported as one of a set of 30 possible values indicating the reaction of the salmonella strains to the test article in question [11]. This test is relative inexpensive and fast to administer, especially relative to the long-term *in vivo* rodent bioassay.

The other primary data set we used consists of the results of ninety-day subchronic studies. These are studies conducted prior to the commencement of a two-year bioassay in order to estimate chronic dose levels [10]. The results are reported as a list of histopathologic changes induced in the rodent as a result of administering the test article in question over a ninety-day period. In the data we are using, 38 different organ were affected and 72 morphologies occurred, resulting in a total of 2736 possible pathological changes that could be recorded. In practice, no more than 10 to 20 pathological changes are recorded in any one rodent.

In addition to salmonella mutagenesis assays and ninety-day subchronic studies, the data also included physical chemical parameters, structural alerts, and dosage levels for the individual experiments. Twenty different physical chemical parameters were included, among them molecular weight, logP, molecular hardness, as well as many others, all well-known to structural chemists. Structural alerts are molecular substructures that human expertise [2] has identified as potential indicators of carcinogenicity. Eighteen of these were included as a set of eighteen binary-valued attributes. Dosage levels are recorded as a single real-valued attribute.

Upon completion of a bioassay and a substantial period of subsequent peer review of the results, each sex/species grouping is assigned a level of evidence from one of five categories: clear evidence, some evidence, equivocal evidence (when the result is uncertain but not negative), no evidence, and inadequate experiment (for seriously flawed experiments). This data is reported in the NCI and NTP Technical Reports Series [10]

For this paper, no experiments labeled equivocal evidence or inadequate experiment were included. In addition, the classifications clear evidence and some evidence were combined into a single classification labeled positive. Classifications of no evidence were labeled negative. Therefore, for the purposes of this paper, the carcinogenicity data is divided into only two sets, positive and negative. This way, the learning tool was not forced to train on data for which no clear evidence is available either way. Also, the learning tool will not be forced to attempt to distinguish between those test agents which we know are positive and

those test agents which we only think are positive. Furthermore, this encoding enabled us to incorporate in the training set experiments conducted before the NTP itself began to distinguish between clear and some evidence.

Dividing the carcinogenicity data in this way resulted in 744 experiments for which a complete set of data is available, meaning that both the results of the salmonella mutagenicity assays and the ninety day subchronic studies are available. There are still occasionally missing data among the physical chemical parameters, structural alerts, and dosage levels. Sex/species, route, and strain information was also included so that the learning tool could distinguish between the type of rodent and the method of test agent administration. Of the 744 experiments, 468 (62.9%) were classified as positive and 276 (37.1%) were classified as negative.

III. NEURAL NETWORK ARCHITECTURE

A 3-layer fully connected feed-forward neural network was used as the basic model for all of the results that follow. This is the smallest network that is guaranteed to be computationally complete. Training and testing of the neural networks was done primarily using the Aspirin/MIGRAINES Neural Network Software [15] running on a SPARCstation. The learning method used throughout is standard error backpropagation.

The choices of the learning rate and the inertia are important in defining the behavior of the neural network. In general, a higher learning rate will cause the network to converge more rapidly. However, if the learning rate is too high, then the network may never converge to a stable weight matrix. Similarly, a high inertia helps to avoid convergence on local minima, although if the inertia is too high then it may slow or prevent the network from reaching a stable configuration.

The best setting for the learning rate and the inertia is different for every training set and every network. No attempt to define a formal method for choosing these values is made here. In practice, initial testing, careful observation of preliminary results, and some trial and error were necessary for setting the learning rate and the inertia correctly. For all the results that follow the learning rate was set at 0.01 and the inertia was set to 0.95.

Most neural net research recommends that training be stopped well before the network reaches 100% accuracy. This practice, called early stopping, is used to avoid overtraining the network. The most common observation is that the accuracy of the network on unseen data will rise along with the training accuracy until some maximum is reached. At that point, the training accuracy will continue to rise while the predictive accuracy falls off sharply. At this point the network stops generalizing and begins memorizing the training data.

In the case of the networks we trained on the toxicology data, the size of the hidden layer was chosen to guard against overtraining. The expected observation, therefore, is that both the training accuracy and the predictive accuracy will rise to a maximum and then level off without

dropping back down. Indeed, while training the neural network models, the training accuracy rose slowly beyond 90% while the test accuracy leveled off at about 87%. A predictive accuracy of 87% already compares favorably with some of the other machine learning approaches to modeling this and similar data sets [5]. See Section VII for detailed results.

Input Layer. The toxicology data contains eight different types of data that must be mapped to the input layer of the network: salmonella test results, route of administration, strain of animal, sex/species, physical chemical parameters, structural alerts, dosage levels, and organ morphology results. The real-valued attributes were normalized to a value between 0 and 1 and mapped to an input node. The values for most of the data were discrete and thus easily mapped to a set of binary input nodes. Results of salmonella tests, routinely performed multiple times on a single test article, were encoded as strings of pluses and minuses, with a total of 30 possible unique strings. Each string was mapped to a single binary input node, resulting in a total of 30 input nodes for the salmonella data. The route, strain, sex/species, and organ morphology data was similarly mapped to sets of binary input nodes. In the case of the organ morphology data, there were 72 different morphologies that can occur in any of 38 different organs. Therefore, 2736 input nodes were required for these results.

With the input layer defined in this manner, a total of 2815 input nodes would be required. Even with a relatively small hidden layer, this would result in a large complicated network that would be computationally expensive to train. Fortunately, most of the 2815 input nodes are not really needed. Input nodes whose input is consistently 0 in all examples in the data set may simply be removed without changing the behavior of the network. In the case of the 744 examples in the toxicology data, this condition applies to nearly 90% of the data (2527 features). Removing the 2527 non-contributing features leaves a set of 288 features and an input layer of 288 nodes. Assuming the size of the hidden layer is not too large, the resulting neural network is computationally very feasible both for training and testing.

Hidden Layer. Since the neural network being used is fully connected between three layers, the only choice involved in defining the hidden layer is the number of nodes to use. Unfortunately, the optimal choice for the size of the hidden layer is often difficult to determine. Sometimes, a good choice can be found by trial-and-error. If the hidden layer contains too many nodes the network may converge rapidly, but is vulnerable to simply memorizing the training data. This results in an underdetermined network with little predictive capability. On the other hand, if the hidden layer is too small, the network may converge slowly or not at all.

Instead of trial-and-error, it is possible to determine a good choice for the size of hidden layer by comparing the number of training examples, the size of the input layer, and the size of the output layer [7]. A small training set can cause the same problem as too many hidden nodes. The number of undetermined parameters NT is a 3-layer

network is given by $NT = J(I + 1) + K(J + 1)$, where I is the number of input nodes, J is the number of hidden nodes, and K is the number of output nodes. This equation can be solved to give

$$J = \frac{NT - K}{I + K + 1}$$

Carpenter and Hoffman give this formula for the number of hidden nodes and propose that NT be set to some number less than the number of training examples N , thus guaranteeing that the network will have fewer undetermined parameters than the number of examples. Carpenter and Hoffman substitute N/α for N and suggest a value of 1.5 for α , but this formula works only when every node in the input layer corresponds to a relevant feature in the data. In the case where some of the input nodes are mapped to random or irrelevant data, a better approach is to substitute I' for I , where I' is the number of relevant features determined by feature selection. This results in

$$J = \frac{N/\alpha - K}{I' + K + 1} \quad (1)$$

Relevant feature subset selection, discussed in Section IV, yields an average of 74 relevant features for the toxicology data. The output layer of the network is a single node. The number of training examples under 10-way cross-validation is typically 90% of the total number of examples. There are 744 examples in the training set. Solving Eq. 1 with α set to 0.5 rounds to a suggested choice of four hidden nodes, and that is the size of the hidden layer used in our subsequent experiments.

Output Layer. The neural network will output a real-valued number from 0 to 1 for each node in the output layer. Defining the output layer is simply a matter of choosing how many output nodes are required and defining a mapping from the resulting real-valued outputs to the target classification in the training data.

In the 744 examples from the toxicology data, all examples are classified as either positive or negative. Therefore, the output layer can be defined as a single node. Real-valued outputs greater than or equal to 0.5 are mapped to a positive classification. Outputs of less than 0.5 are mapped to a negative classification. During backpropagation, the target value for a positive example is 1 and the target for a negative example is 0. This mapping is used for all further results.

IV. RELEVANT FEATURE SUBSET SELECTION

In any machine learning study where a large amount of raw real-world training data is used, it is quite possible that only some of the attributes present in the data are relevant to the task of training and prediction. This is especially true with the data used in this paper, where every available attribute is used to train the neural networks regardless of what we might assume about the attribute's contribution to the study of predictive toxicology. Ideally, a machine learning algorithm could achieve maximum performance in this situation by using only those attributes which increase

predictive accuracy and ignoring the remaining attributes. Unfortunately, it is almost never possible to do this. Irrelevant attributes added to training and testing data often serve only to hamper the performance of the machine learning tool. In the neural network, irrelevant attributes interact with backpropagation and can negatively affect the bias of each neuron. The problem of how to identify and remove extra attributes which either don't contribute or contribute negatively to the prediction task is called relevant feature subset selection. The optimal choice of relevant features could result in higher prediction accuracy, smaller model size, or use of fewer input attributes.

A. Relevance

There are two tasks to be accomplished when doing feature selection. The first is defining the notion of relevance. The second is creating an algorithm capable of identifying those attributes which are irrelevant under the definition. Simple induction of a minimal structure is generally NP-hard [4], so a heuristic or some other clever approach is necessary.

A simple definition of relevance can be given in the absence of noise: a feature X_i is said to be relevant to a concept C if X_i appears in every Boolean formula that represents C , and irrelevant otherwise [1]. A better definition defines X_i to be relevant if the probability of the concept C can change when all knowledge about X_i is removed. Later [13], the notion of strong and weak relevance was proposed. A strongly relevant feature cannot be removed without definite loss of prediction accuracy. Weak relevance implies only that the feature can sometimes contribute to prediction accuracy.

In this paper, an even looser view of relevance is used. A feature is not simply called relevant or irrelevant. Instead, each feature is assigned a degree of relevance which can be any real number from zero to one. A relevance of one would indicate that the feature was indispensable. A relevance of zero means that the feature is completely irrelevant and may be removed without affecting the model in any way. Most features would lie somewhere between these two extremes.

B. Filter and Wrapper Methods

There are two basic approaches to the task of feature selection: filter methods and wrapper methods. A selection algorithm is called a filter method when it can execute independently of the learning tool. Filter methods [1], [14] generally use some sort of heuristic search over the space of possible features to determine a near-optimal feature set. The alternative is a wrapper method [13], which uses results from the learning tool itself to decide which features are relevant or irrelevant.

Existing filter methods for feature selection like FOCUS [1], Relief [14], and wrapper methods [13] were all tested with the toxicology data. None provided satisfactory results. FOCUS fails because of the presence of features which uniquely identify the examples in the data set but which do not generalize well. Relief fails to identify use-

ful subsets of weakly relevant features and cannot identify redundant subsets of features. The wrapper models are simply too computationally complex to be useful.

C. Single Hidden Unit Method

When a neural network trains on a given set of training data, it learns real-valued weights for each connection between the nodes in the network. It seems natural to use these weights to indicate the relevance of the input attributes to the network. If the network were a three-layer network with a single hidden node, the once the network were trained, the magnitude of the connection weight for a given attribute would be relative to the level of contribution that the attribute makes to the function of the network. The absolute value of the magnitude of the weight could be considered the relevance weight of the attribute. In our experiments we used a new approach to relevant feature subset selection based on these ideas. The Single Hidden Unit (SHU) method [19] uses the connection weights in a trained neural network to assign a relevance weight to each attribute. Then, a threshold τ is used to split the attributes into sets of relevant and irrelevant features depending on whether the feature’s relevance weight falls above or below the threshold. The algorithm for the Single Hidden Unit method is given in Table I.

Train a 3-layer network with a single hidden node on all available data until the network either converges or the training accuracy begins to level off.
For each attribute A_i , let the relevance weight of A_i be the magnitude of the connection weight to the single hidden node from the input node which corresponds to A_i .
Choose the threshold τ to be some number greater than the minimum relevance weight and less than the maximum relevance weight.
For each attribute A_i , if the relevance weight of A_i is less than τ then label A_i as irrelevant. Otherwise label A_i as relevant.

TABLE I
ALGORITHM FOR SHU FEATURE SELECTION

It is conceded that the accuracy of the Single Hidden Unit network will be much lower than a larger network, but the relevance or irrelevance of attributes is consistent regardless of network structure. The real difficulty in using the Single Hidden Unit method is the selection of the threshold τ . Since there is no optimal means of selecting τ , it is necessary to simply make a guess initially. Hill climbing can then be used to narrow in on the best choice for τ . A hill climbing algorithm for finding an appropriate threshold is given in Table II.

V. CONNECTION WEIGHT PRUNING

One of the primary issues involved in evaluating the neural networks has been their ability to generalize to patterns

Find the standard deviation of the relevance weights. Use one standard deviation as the initial value of τ . Let 0.1 be an initial value for the step size s .
Label any feature whose relevance weight is less than the cutoff as irrelevant. Find the cross-validate test accuracy of the network with all irrelevant features masked out.
Adjust τ by positive s and negative s and find the cross-validated test accuracy of the networks with the new sets of relevant features. If both results are less than the current predictive accuracy then we are finished. Otherwise, let τ be the threshold which resulted in an improved predictive accuracy.
If the sign of s that produced a better τ is the opposite of the previous sign of s that produced a better τ , then the direction of hill climbing has reversed. In this case, cut the size of s in half.
Go to Step 3.

TABLE II
HILL CLIMBING ALGORITHM

outside the training set. One of the measures of a network’s performance is prediction accuracy, which is estimated with the use of cross validation. Connection weight pruning is a tool which can improve the prediction accuracy of trained neural networks.

The standard approach to obtaining good generalization is to use the smallest system that will fit the data. The choice of the size of the hidden layer and relevant feature subset selection attempts to do this before the neural network begins training. However, these methods produce only an estimate of the best possible network size. The most obvious solution, therefore, would be to successively train smaller and smaller networks until the one that generalizes best on the data is found. Unfortunately, this approach is impractical because of the amount of time required to iteratively train many networks. Also, the smaller the networks become, the more sensitive they are to initial settings and the order of the training data and the more testing is required to obtain an accurate model of their capabilities.

Connection weight pruning avoids these problems by waiting until after the neural network is finished training before reducing its size. The goal of connection weight pruning is to remove connections between nodes in a previously trained network until an optimal sized network remains. Many different algorithms are available which will prune network connections, but unfortunately none are able to guarantee that the end result will be an optimal network [17].

Most of the algorithms can be divided into two broad categories. The sensitivity methods start with a trained network. The sensitivity of the error function to the removal of each weight is estimated and those weights with the least effect are removed. The penalty-term methods, on the other hand, require that the training of the network

itself be modified, in that terms are added to the objective function which reward the network for choosing efficient solutions.

Any sensitivity method can be defined by specifying two things. First, the sensitivity calculation method must be defined. Then, once each weight has a sensitivity value assigned to it, the process by which these values are used to eliminate connections must be determined.

Our experiments indicate that it is reasonable to assume that the weights in the network can be divided into groups where all the connection weights in a single group provide the same function. An ideal method for connection pruning would be able to define the groups and then prune all but one connection from each group. This is a potentially daunting task since there is no way of knowing the size or number of groups or even if the groups are all distinct. Some groups may intersect other groups. Fortunately, the same result as this ideal method can be achieved by iteratively pruning the connection weights. This paper uses an iterative method to prune the trained neural networks. The algorithm is given in Table III.

Let N_0 be the network with all connection weights unmodified.
Impose an arbitrary ordering from 1 to 296 (4×74) on the weights.
For $i = 1$ to 296 do: Let N_i be equal to N_{i-1} with connection weight w_i set to 0. Test N_i and N_{i-1} on the training examples. If the accuracy of N_i is less than the accuracy of N_{i-1} then reset the value of w_i to its original value. Otherwise leave $W - I$ set to 0.
The final neural network is equal to N_{296} .

TABLE III
CONNECTION WEIGHT PRUNING ALGORITHM

The primary benefit of this method over previous methods is that the sensitivity value of each connection weight is not measured in isolation from all the other connections. Instead, the sensitivity value of each connection is computed in order and takes into account which of the connections that preceded it were pruned.

Connection weight pruning has other advantages besides improving the predictive accuracy of the neural network. A pruned neural network has the minimum number of connections required for that network to generalize the data. Removing the unnecessary nodes and connections will significantly increase the speed at which the neural network can make classifications from input data. This allows for a broader range of potentially computationally intensive analyses to be performed on the trained and pruned networks. One of the types of analysis is rule extraction. Rule extraction benefits not only from the increased speed but also from the reduced complexity of the weight matrix which describes the neural network.

VI. RULE EXTRACTION

In Section I we listed several problems with constructing neural network models. Most of these problems have been addressed earlier in this paper. By precisizing defining how to structure and train the networks, using feature selection to remove irrelevant attributes, and pruning unnecessary weights from the trained networks, a method has been created that is capable of modeling the toxicity data and achieving an estimated prediction accuracy that favorably compares with other approaches to this application. There is only one remaining problem with neural networks that must be addressed. They are incapable of explaining their results.

Attempts have been made to provide a way to interpret the knowledge learned by a neural network. A great deal of early work focused on converting neural networks to fuzzy systems. Neural and fuzzy systems are theoretically equivalent [6]; however, the actual process of conversion from one to another is rarely straightforward. Most existing methods require the network to be extended or otherwise modified. A group of more general approaches to extracting rules does exist [8], [9], but again these methods mostly require modification to the structure or learning algorithm of the network. Upon testing existing extraction methods on our networks (see Section VII), it became clear that what was really needed is a method for rule extraction that is capable of generating a minimally sized set of readable rules strictly from the weight matrix or an arbitrarily structured trained network. Only such a method is suitable for interpreting the networks trained on our toxicology data.

Two goals for a successful rule extraction algorithm are that the extracted rules closely model the behavior of the network and that they be general and easy to understand. It is important to separate these goals.

An M -of- N rule has the following format:

If at least M of a set of N attributes and no others are present, then classify the example as positive, otherwise classify as negative.

We use a brute force method that generates a set of M -of- N rules that completely describe the behavior of a network trained on the NTP data. Conceptually, our extraction method works as follows. First, we attempt to specify the behavior of the network completely by building a large set of very specific rules. This would satisfy the goal of closely modeling the network. Once the rule set is complete and shown to be accurate, then the rules are combined to form a smaller set of more general rules that are readable. This is done to satisfy the goal of generality and understandability.

The brute force method is so named because of the way it goes about building an initial set of precise rules that completely define the behavior of the network. In principle, the rules can be constructed by simply iterating every possible combination of inputs to the network and matching those inputs with the corresponding output of the neural network on that input. Each rule applies to exactly one set of inputs. In order to completely model the neural network, only confirming rules are necessary. This is because if there exists a rule for every case when the network out-

puts a positive classification, then negative classifications are defined by the absence of rules.

In practice, the task of enumerating every possible input to the network is clearly intractable, and would result in rule sets of size roughly 10^{12} . Therefore, we proceed bottom-up, by beginning with only those examples which are present in the training data and then building M -of- N rules by adding attributes to those examples. The brute force rule extraction algorithm is given in Table IV.

Let \mathbf{R} be the set of M -of- N rules. Initially, \mathbf{R} is empty.
For each example in the training set classified as positive, do Step 3.
If the example is not already predicted positive by a rule in the rule set \mathbf{R} then make a M -of- N rule where M is equal to the number of positive attributes in the example and N includes all the positive attributes in the example. Next, do Step 4. When Step 4 is complete, add the resulting M -of- N rule to \mathbf{R} .
For each attribute not already in N , attempt to add the attribute to N by the following process. If every combination of the attribute and $M-1$ attributes from N is classified positive by the network, then add the attribute to N .

TABLE IV
BRUTE FORCE RULE EXTRACTION

The revised method for brute force rule extraction is quick and easy to implement and yet it yields a modestly sized rule set which is easy to interpret and very closely models the behavior of the neural network. Furthermore, this method for rule extraction will work regardless of the structure of the neural network or the activation function. The capability of extracting rules from a general trained neural network is essential to their use in our toxicology application.

VII. RESULTS

A. Evaluation

When measuring the performance of any model constructed using machine learning tools, it is desirable to measure both the model’s ability to classify the data it was trained on and also the model’s ability to classify unseen data from the same domain. This second measure of performance is called the predictive ability of the model and can be estimated by using the technique of cross-validation. To estimate the predictive ability of the neural net model described previously, 10-way cross-validation was used. The data was split randomly into ten sets with the single condition that the proportion of examples with a positive classification to examples with a negative classification be the same in each of the ten sets. Since there was a total of 744 examples in the data, the average training set size was 670 examples and the average test set size was 74 examples. The general outline of each step of this process is shown in Fig. 1.

Evaluation by human experts – toxicologists, structural chemists, biologists, and pathologists – is also essential to project such as this one. For several year, we have made use of regular feedback from collaborators at the National Institute of Environmental Health Sciences and scientists at the Health Effects Research Laboratory of the U.S. Environmental Protection Agency. Their reaction – positive and negative – to our models has been an invaluable guide that has saved us from countless pitfalls.

B. Feature Selection Results

The Single Hidden Unit Method was used to find a relevant subset of the 288 attributes in the training data. The step of assigning the relevance weights is very quick and efficient. Furthermore, finding an optimal value for the threshold t required only four iterations of the hill climbing algorithm. The optimal threshold was determined to be within 0.2 of the standard deviation of the relevance weights. After this threshold was used to split the 288 features into relevant and irrelevant, an average of 74 features were labeled as relevant.

10-way cross-validated results are shown in Table V. A test accuracy of 89% is considerably better than any of the existing methods we considered for relevant feature subset selection. This is especially impressive considering that the SHU feature selection algorithm is simple and efficient.

After Feature Selection ($p < 0.001$)				
		Predicted Bioassay		
		Positive	Negative	Total
Actual Bioassay	Positive	392	60	452
	Negative	20	272	292
	Total	412	332	744
Accuracy:		0.89		
Sensitivity:		0.87		
Specificity:		0.93		
+ Predictivity:		0.95		
– Predictivity:		0.82		
Corr. Coeff.:		0.78		

TABLE V
MODEL AFTER SHU FEATURE SELECTION (CROSS-VALIDATED AVERAGES)

These results demonstrate that the accuracy of neural networks trained on all 744 examples, as estimated by cross-validation, can be much improved by using Single Hidden Unit feature selection. It is also worthwhile to show explicitly that randomly generated features and features which are clearly irrelevant are indeed thrown out by this method of feature selection.

To demonstrate this, we began with the set of 288 features, all of which are used at least once in the 744 experiments. The 288 features included salmonella results, strain, sex/species, route, and sub-chronic data. To the 288 features, 10 more features were added. For the first experiment, TRUE or FALSE values were randomly assigned to

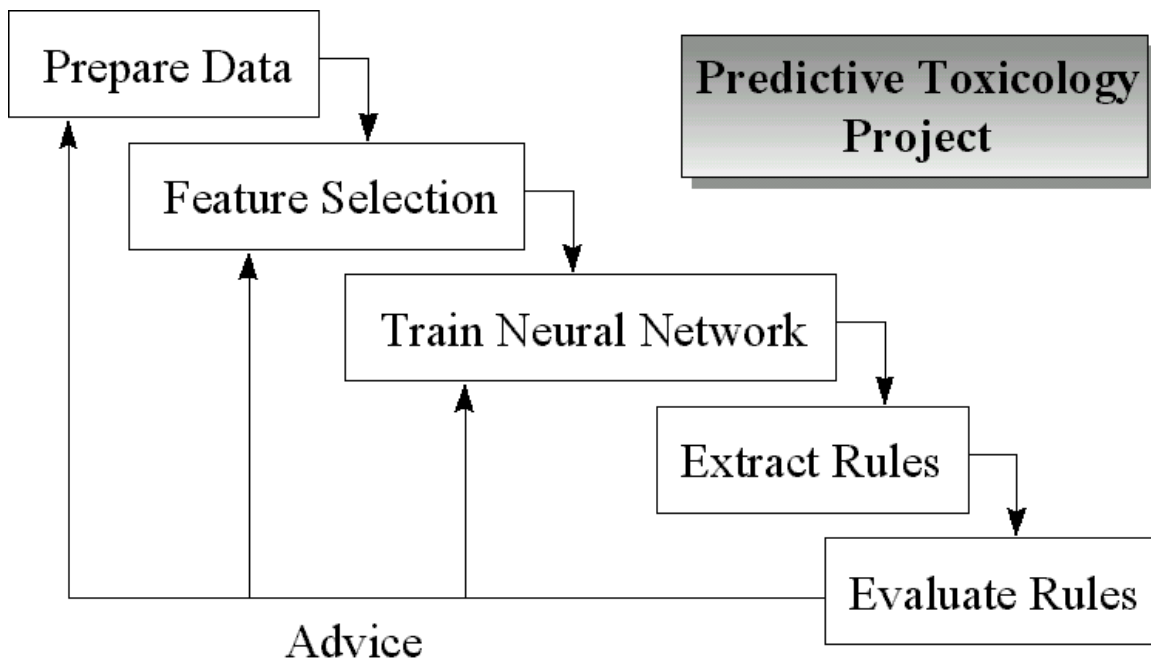


Fig. 1. Knowledge Discovery in Toxicology: Contents of One Cross-Validation Step

all occurrences of the 10 new features in the 744 examples. Since the values of the features were randomly generated, they are clearly irrelevant to the classification task. The new data was then run through the Single Hidden Unit feature selection process. The results were as expected. The relevance level of all ten randomly generated features fell well below the cutoff for relevance. The optimum cutoff only took the highest 35% of the features ordered according to relevance. All of the ten new features were in the bottom 47% of features with most of them being in the bottom 15%.

A second experiment assigned all of the ten new features a constant TRUE value throughout all 744 examples. Clearly, these features should also be labeled as irrelevant. Indeed, the Single Hidden Unit Method assigned all ten features a relevance level well below the cutoff with each feature being in the bottom 35%. The experiment in which all ten features are assigned a constant FALSE value is unnecessary since the process always begins by eliminating all features in this category.

These results demonstrate that the Single Hidden Unit feature selection method is not only improving training accuracy through hill climbing, but also eliminating individual features which are not relevant to the classification task.

C. Connection Weight Pruning Results

Results of iterative pruning are listed in Table VI. An average of 279.8 (80 of the connection weights are pruned using this method. Most of the connection weights remaining are on the third hidden node. Although the average estimated prediction accuracy rises less than 1% (not significant by McNemar's Test), considering that the network is roughly 80% smaller than it was prior to pruning, any rise in accuracy is an excellent result.

Average No. of Pruned Weights	278.8
...on Node 1	84.2
...on Node 2	78.9
...on Node 3	29.8
...on Node 4	85.9
Cross-Validated Test Accuracy	90%

TABLE VI
AVERAGE RESULTS OF CONNECTION WEIGHT PRUNING

Since the iterative method both improves the predictive accuracy of the network and significantly reduces the number of connection weights, this is the pruning method which is applied to the trained neural networks prior to weight analysis and rule extraction.

D. Rule Extraction Results

The bottom-up approach to brute force rule extraction was applied to the trained and pruned neural networks. The 10-way cross validated results are listed in Table VII.

Just over 2% of predictive accuracy was lost between the neural network and the set of M -of- N rules. The rule set itself is compact and easy to read and interpret. Only 22 rules were required to completely model the behavior of the trained neural network. Of these, two of the rules, the first and the seventh, are used the majority of the time. A few other rules are used four or more times. The bulk of the rules are used only once or twice. This would indicate that there are only a few general rules which can be defined to describe the classifications in the data. In order to achieve high accuracy, these general rules must be supplemented with many other rules to describe special cases. This re-

After Rule Extraction ($p < 0.001$)				
		Predicted Bioassay		
		Positive	Negative	Total
Actual Bioassay	Positive	389	74	463
	Negative	25	256	281
	Total	414	330	744
Accuracy:		0.87		
Sensitivity:		0.84		
Specificity:		0.91		
+ Predictivity:		0.94		
- Predictivity:		0.78		
Corr. Coeff.:		0.73		

TABLE VII
EXTRACTED RULE MODEL (CROSS-VALIDATED AVERAGES)

quirement makes perfect sense given the nature and source of the data.

For completeness, a trimmed version of the complete rule set is listed in the Appendix. All of the rules are included. However, attributes on the left hand side of a rule which are never used during testing on the 744 training examples have been removed for greater readability. The rules are all positive indicators. Therefore, if a test example activates any rule then it is classified positive. If no rules are activated then the classification is negative.

Our human expert collaborators are currently evaluating these models, but this process is far from complete.

For comparison, we extracted rules from our trained networks using two methods in addition to our own: the Weighted Attribute approach [8], and the KT algorithm [9]. Results are compared to the average trained network in Table VIII. Significant accuracy was lost by both the Weighted Attribute and KT methods, and in at least one case the set of rules was unacceptably large.

Method	Cross-Validated Accuracy	No. of Rules
Neural Net	89%	—
Weighted Attribute	68%	74
KT	69%	3346
Brute Force	87%	22

TABLE VIII
COMPARISON OF RULE EXTRACTION METHODS

VIII. CONCLUSIONS

The task of using neural models in a knowledge discovery application such as the prediction of chemical carcinogenesis in rodents is by no means trivial. Neural networks have been shown to be a capable model for accomplishing this task, providing as good or better results than other approaches to the same problem. However, in order for the networks to be useful, several pre-processing and post-processing steps had to be defined.

Relevant feature subset selection was considered as a

means for removing features which did not contribute to the classification task. A new approach to feature selection was used which uses the connection weights of a trained network to assign relevance weights to the attributes.

Finally, rule extraction was investigated as a means for networks to explain themselves. The brute force approach to rule extraction built a reasonably small rule set that only suffered a small drop in accuracy from the neural network. An algorithm was presented for the brute force approach which allowed it to finish in reasonable time. The M -of- N rules are readable and useful for describing the knowledge learned by the network in terms that humans can understand.

By combining the strong inductive capability of neural networks with tools like feature selection, connection weight pruning, and rule extraction, neural networks become very powerful knowledge discovery tools. In addition to training a network that is estimated to have relatively good predictive accuracy in this field, the results from feature selection and the rules that were extracted from the network are providing new information to predictive toxicologists that has great potential for pointing the way toward new insights in the field.

REFERENCES

- [1] Almuallim, H. and T. G. Dietterich, "Learning with Many Irrelevant Features," *Proc. 9th Nat. Conf. on Art. Intell. (AAAI-91)*, MIT Press, 547-552, 1991.
- [2] Ashby, J. and R. W. Tennant 1991. Definitive Relationships among chemical structure, carcinogenicity, and mutagenicity for 301 chemicals tested by the U.S. NTP. *Mutation Research* 257: 229-306.
- [3] Bahler, D. and D. W. Bristol, "The Induction of Rules for Predicting Chemical Carcinogenesis in Rodents," in L. Hunter, J. Shavlik, and D. Searls, eds. *Intelligent Systems for Molecular Biology*, Cambridge, MA: AAAI/MIT Press, 1993.
- [4] Blum, A.L. and R.L. Rivest, "Training a 3-node Neural Network is NP-Complete," *Neural Networks* 5, 117-127, 1992.
- [5] Bristol, D. W., J.T. Wachsman, and A. Greenwell, "The NIEHS Predictive-Toxicology Evaluation Project," *Environmental Health Perspectives* 104 (Supplement 5), 1996, 1001-1010.
- [6] Buckley, J.J., Y. Hayashi, and E. Czogala, "On the equivalence of neural networks and fuzzy expert systems," *Proc. IFCNN-92*, v. 2, 691-695, 1992.
- [7] Carpenter, William C. and Margery E. Hoffman, "Training Backprop Neural Networks," *AI Expert*, 30, March 1995.
- [8] Craven, M. W., *Extracting Comprehensible Models From Trained Neural Networks*, Ph.D. Dissertation, Department of Computer Science, University of Wisconsin, 1996.
- [9] Fu, L., "Rule Generation From Neural Networks," *IEEE Trans. on Systems, Man, and Cybernetics* 24(8), August 1994.
- [10] Huff, J.E. and J.K. Haseman, "Long-Term Chemical Carcinogenesis Experiments for Identifying Potential Human Cancer Hazards: Collective Database of the National Cancer Institute and National Toxicology Program (1976-1991)," *Environmental Health Perspectives* 96, 23-31, 1991.
- [11] Huff, James, E.E. McConnell, and J.A. Moore, "The National Toxicology Program Toxicology Data Evaluation Techniques and Long-Term Carcinogenesis Studies," *Safety Evaluation of Drugs and Chemicals* 441-447, Washington DC, 1985.
- [12] Huff, J.E., E.E. McConnell, J.K. Haseman, G.A. Boorman, S.L. Eustis, B.A. Schwetz, G.N. Rao, C.W. Jameson, L.G. Hart, and D.P. Rall, "Carcinogenesis Studies: Results from 398 Experiments on 104 Chemicals from the U.S. National Toxicology Program," *Proc. Nat. Acad. Sci.* 534 1-30, New York, 1988.
- [13] John G. H., R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," *Proc. 11th Intl. Conf. on Machine Learning*, Rutgers, NJ, 1994.

- [14] Kira, K. and L. Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm," *Proc. 10th Nat. Conf. on Art. Intell. (AAAI-92)*, 129-134, 1992.
- [15] Leighton, Russell R., *The Aspirin/MIGRAINES Neural Network Software Release v.6.0*, Mitre Corporation, 1992.
- [16] Lewis, David F.V., "Comparison between Rodent Carcinogenicity Test Results of 44 Chemicals and a Number of Predictive Systems," *Regulatory Toxicology and Pharmacology*, 215-222, 1994.
- [17] Reed, Russell, "Pruning Algorithms - A Survey," *IEEE Transactions on Neural Networks* 4(5), September, 1993.
- [18] A. Srinivasan, S.H. Muggleton, R.D. King, and M.J.E. Sternberg, "The Predictive Toxicology Evaluation Challenge," *Proc. IJCAI-97*, August 1997, 4-9.
- [19] Stone, Brian, "Feature Selection and Rule Extraction for Neural Networks in the Domain of Predictive Toxicology," Master's Thesis, North Carolina State University, December 1995.
- [20] Thrun, S.B. et al., "The MONK's Problems - A Performance Comparison of Different Learning Algorithms," CS-CMU-91-197, Dept. of Computer Science, Carnegie-Mellon University, 1991.

Appendix

The following is a list of the *M-of-N* Rules generated from an artificial neural network with 74 input nodes and 4 hidden nodes trained on all 744 of our examples grouped by experiment.

1. If 1 or more of the following and not any other features:
 Salmonella: +
 SA: A
 SA: C
 SA: E
 SA: G
 SA: O
 SA: Q
 SA: S
 SA: T
 rate Ke
 Z-depth
 hyperplasia bone marrow
 hyperkeratosis forestomach
 granular casts kidney
 karyomegaly kidney
 necrosis kidney
 regeneration kidney
 hypertrophy liver
 histiocytosis lung
 erosion ulceration skin
 necrosis trachea
 hemorrhage urinary bladder
 hyperplasia urinary bladder
2. If 2 or more of the following and not any other features:
 SA: P
 SA: U
 rate Ke
3. If 2 or more of the following and not any other features:
 Salmonella: +W
 hypertrophy liver
4. If 2 or more of the following and not any other features:
 Salmonella: +-
 SA: I
5. If 2 or more of the following and not any other features:
 Salmonella: on-test
 SA: G
 degeneration olfactory nasal cavity
6. If 3 or more of the following and not any other features:
 Salmonella: +
 SA: B
 SA: C
 karyomegaly kidney
7. If 3 or more of the following and not any other features:
 Salmonella: +, SA: G, SA: Q, Clogp, PA, Z-depth, hyperkeratosis forestomach, karyomegaly kidney, necrosis kidney, hypertrophy liver, histiocytosis lung, erosion ulceration skin
8. If 4 or more of the following and not any

- other features:
 Salmonella: - - +W
 Salmonella: +-
 SA: G
 SA: O
 SA: R
 SA: T
 Z-depth
 electronegativity
 molecularhardness
 hardness2
 degeneration spinal cord
9. If 4 or more of the following and not any other features:
 SA: C
 SA: Q
 pigmentation kidney
 hypertrophy kupffer cell liver
10. If 5 or more of the following and not any other features:
 SA: R
 SA: S
 rate Ke
 electronegativity
 molecularhardness
 hardness2
 necrosis kidney
11. If 5 or more of the following and not any other features:
 SA: Q
 PA
 hyperplasia bone marrow
 atrophy spleen
 hemorrhage urinary bladder
12. If 5 or more of the following and not any other features:
 Salmonella: +
 SA: R
 rate Ke
 Clogp
 Z-depth
 electronegativity
 molecularhardness
 hardness2
 granular casts kidney
13. If 5 or more of the following and not any other features:
 Salmonella: ?+
 SA: Q
 SA: U
 PA
 electronegativity
14. If 5 or more of the following and not any other features:
 Salmonella: +
 electronegativity
 molecularhardness
 hardness2
 regeneration kidney
15. If 6 or more of the following and not any other features:
 Salmonella: +
 SA: Q
 PA
 Z-depth
 pigmentation kidney
 hypertrophy liver
16. If 6 or more of the following and not any other features:
 Salmonella: +
 SA: T
 rate Ke
 Clogp
 Elumo
 PA
 Z-depth
 electronegativity
 molecularhardness
 hardness2
 karyomegaly kidney
 necrosis kidney
 regeneration kidney
 hypertrophy liver
 necrosis trachea
 hemorrhage urinary bladder
17. If 7 or more of the following and not any other features:
 Salmonella: +
 SA: A
 SA: Q
 SA: R
 rate Ke
 PA
 Z-depth
 electronegativity
 molecularhardness
 hardness2
 logp
 hyperkeratosis forestomach
 karyomegaly kidney
 necrosis trachea
 hemorrhage urinary bladder
 hyperplasia urinary bladder
18. If 7 or more of the following and not any other features:
 SA: A
 Z-depth
 electronegativity
 molecularhardness
 hardness2
 logp
 necrosis brain
19. If 7 or more of the following and not any other features:
 Salmonella: +
 SA: Q
 rate Ke
 PA
 Z-depth
 electronegativity
 molecularhardness
 hardness2
 moleculararea
 hyperplasia transitional cell kidney
 karyomegaly kidney
 regeneration kidney
 hypertrophy liver

20. If 7 or more of the following and not any other features:

rate Ke
Clogp
Z-depth
electronegativity
molecularhardness
hardness2
molecularvolume
moleculararea
logp

21. If 8 or more of the following and not any other features:

rate Ke
Clogp
PA
Z-depth
electronegativity
molecularhardness
hardness2
moleculararea

22. If 8 or more of the following and not any other features:

Salmonella: +
SA: Q
PA
hyperplasia transitional cell kidney
necrosis kidney
pigmentation kidney
hypoplasia thyroid
hemorrhage urinary bladder
hyperplasia urinary bladder