Generating concise and accurate classification rules for breast cancer diagnosis

Rudy Setiono

School of Computing

National University of Singapore

Kent Ridge, Singapore 119260

Republic of Singapore

Email:rudys@comp.nus.edu.sg

Phone:(65) 772-6297

Fax :(65) 779-4580

Abstract

In our previous work, we have presented an algorithm that extracts classification rules from trained neural networks and discussed its application to breast cancer diagnosis. In this paper, we describe how the accuracy of the networks and the accuracy of the rules extracted from them can be improved by a simple pre-processing of the data. Data pre-processing involves selecting the relevant input attributes and removing those samples with missing attribute values. The rules generated by our neural network rule extraction algorithm are more concise and accurate than those generated by other rule generating methods reported in the literature.

Keywords. Neural network rule extraction; Wisconsin breast cancer diagnosis; Data pre-processing; Attribute selection.

1 Introduction

Neural networks have been shown to be valuable tools for pattern classification in various areas of the medical field [2, 3, 5, 6, 7, 9, 10, 12, 20, 21]. Given an input pattern, the neural network prediction is normally computed as a complex nonlinear mapping of the input attribute values. In some applications, in addition to achieving high predictive accuracy, having a set of meaningful and easy to explain classification rules is desirable. Such rules can be verified by the experts and may provide better understanding of the problem in hand. It is not surprising, therefore, that in recent years there is a proliferation of algorithms which extract rules from trained neural networks [1].

A rule extraction algorithm that we have proposed earlier is NeuroRule [13]. It is an algorithm that extracts rules from trained feedforward neural networks with a single hidden layer. Two key components of this algorithm are a network pruning method [15] and a hidden unit clustering algorithm. An effective pruning algorithm removes the redundant connections and units from the network. A robust clustering algorithm clusters or discretizes the hidden unit activation values of the input patterns into a small number of clusters. The rules are extracted in two steps. The network outputs are first described as classification rules in terms of the clustered hidden unit activation values. Each cluster of hidden unit activation values is then explained as rules that involve the input attributes. By merging the two sets of rules, the algorithm obtains a set of rules that explains the network outputs in terms of the input attributes of the data. A more concise set of rules can be thus expected from a network with fewer connections and fewer clusters of hidden unit activations.

An application of NeuroRule to the Wisconsin breast cancer diagnosis (WBCD) problem has been reported in our previous work [14]. Since then, there have been a number of papers that introduce new methods for rule generation and their application to WBCD. Among the methods presented in these papers are a combined fuzzy-genetic approach [11] and three neural network rule extraction algorithms [18, 19]. A common feature of these more recent works is that their authors claim their methods can achieve better performance than NeuroRule in terms of rule simplicity and rule accuracy.

In this paper we propose how the performance of NeuroRule for WBCD can be improved by data pre-processing. Two steps are involved in the data pre-processing. The first step is to check samples with missing attributes values. Of the 699 samples in the WBCD data set, 16 samples have one missing attribute value each. Before neural network training starts, those samples with missing attribute values are simply discarded. This is also done by other researchers [11, 19] before they applied their methods. The second step of data pre-processing is to select the most relevant attributes of the data for classification. Attribute selection is done by neural networks with one hidden unit. The advantage of attribute selection is that it reduces the computation time required to train and prune the the neural networks for rule generation.

In Section 2 of this paper we provide a brief outline of NeuroRule. The results of our experiments using NeuroRule on the pre-processed WBCD data set are presented in Section 3. Sample rules that are extracted from three different pruned networks are presented here. In this section, we also compare our results with those from other rule generating methods. Finally, we conclude in Section 4.

2 Rule extraction with NeuroRule

We give a brief outline of NeuroRule [13] in this section. As the feedforward neural networks are among the most common type of networks used, NeuroRule attempts to extract rules from this type of networks. The network is assumed to have an input layer, a hidden layer, and an output layer. The number of units in the input and output layers depends on the dimensionality of the data and the number of classes or groups of samples. The number of units in the hidden layer can be determined by either having many hidden units and then removing those redundant ones or by starting with one hidden unit and then adding more units as needed to achieve the minimum required accuracy. In this paper, as in our previous work, we adopt the first approach. The number of hidden units needed for classifying the samples of the WBCD data set using a single hidden layer feedforward neural networks is as few as three and as many as nine [4, 14, 19].

Network connections link units in the input layer to units in the hidden layer and units in the hidden layer to those in the output layer. There is no direct connections between units in the input layer and units in the output layer. Given this network structure, it is natural to decompose the process of rule extraction into two steps and then combine the two sets of rules. The outline of the rule extraction method is as follows:

1. Select and train a network to meet the prespecified accuracy requirement.
2. Remove the redundant connections in the network by pruning while maintaining its accuracy.
3. Discretize the hidden unit activation values of the pruned network by clustering.
4. Extract rules that describe the network outputs in terms of the discretized hidden unit activation values.
5. Generate rules that describe the discretized hidden unit activation values in terms of the network inputs.
6. Merge the two sets of rules generated in Steps 4 and 5 to obtain a set of rules that relates the inputs and outputs of the network.

In Step 1, we train the network by minimizing the augmented error function

$$F(w,v) = -\sum_{i=1}^{k}\sum_{p=1}^{C}\left(t_p^i \log S_p^i + (1-t_p^i)\log(1-S_p^i)\right) + P(w,v), \qquad (1)$$

where $P(w,v)$ is the penalty term

$$P(w,v) = \epsilon_1 \left(\sum_{j=1}^{h} \sum_{\ell=1}^{n} \frac{\beta (w_\ell^j)^2}{1+\beta (w_\ell^j)^2} + \sum_{j=1}^{h} \sum_{p=1}^{C} \frac{\beta (v_p^j)^2}{1+\beta (v_p^j)^2} \right) + \quad (2)$$

$$\epsilon_2 \left(\sum_{j=1}^{h} \sum_{\ell=1}^{n} \left(w_\ell^j\right)^2 + \sum_{j=1}^{h} \sum_{p=1}^{C} \left(v_p^j\right)^2 \right).$$

In the above equations, ϵ_1, ϵ_2 and β are positive parameters, t_p^i is the target output at unit p for the n-dimensional input sample x^i, while S_p^i is the network output for this sample

$$S_p^i = \sigma \left(\sum_{j=1}^{h} \delta \left((x^i)^T w^j \right) v_p^j \right), \quad (3)$$

where $(x^i)^T w^j$ denotes the scalar product of the input vector x^i and w^j, the weight vector for connections from the input units to hidden unit j. $\delta(.)$ is the hyperbolic tangent function, and $\sigma(.)$ is the sigmoid function. The weight of the connection from input unit ℓ to hidden unit j and the connection from hidden unit j to output unit p are denoted as w_ℓ^j and v_p^j, respectively. The number of training samples is k, the dimensionality of the data is n and the number of hidden units is h. For a binary classification problem $C = 1$ output unit is sufficient, otherwise C is set to the number of classes in the data set.

Once network training has stopped, network pruning begins. Network connections are selected for removal based on their magnitudes. After one or more connections have been removed, the network is retrained and rechecked to see if any more connections can be removed. The details of the network pruning algorithm can be found in [15].

Since hidden unit activation values have been computed as the hyperbolic tangent of the weighted inputs, their values will lie in the interval $[-1, 1]$. Before rules are extracted, the activation values are clustered. Clustering is equivalent to dividing the interval $[-1, 1]$ into several subintervals. We have developed several clustering algorithm such as Chi2 [16] and greedy clustering algorithm (GCA) [17]. The goal of applying a clustering algorithm to the hidden unit activations is to have the network outputs described by as few cluster combinations as possible. Cares are taken when clustering so that a sample that is correctly classified by the trained network will still be correctly classified after its hidden unit activation values have been clustered.

In the next section, we present the results from applying the clustering algorithm to a pruned network that have been trained on the WBCD data set. We also show the rules that are extracted from 3 different pruned networks.

3 The WBCD problem

3.1 Data set and its pre-processing

The data set [1] consists of 699 samples taken from fine needle aspirates from human breast tissue. They have been collected by Dr. W.H. Wolberg at the University of Wisconsin - Madison Hospitals. Each sample consists of nine measurements: $\mathcal{A}_1$. clump thickness, $\mathcal{A}_2$. uniformity of cell size, $\mathcal{A}_3$. uniformity of cell shape, $\mathcal{A}_4$. marginal adhesion, $\mathcal{A}_5$. single epithelial cell size, $\mathcal{A}_6$. bare nuclei, $\mathcal{A}_7$. bland chromatin, $\mathcal{A}_8$. normal nucleoli, and $\mathcal{A}_9$. mitosis. The measurements are assigned an integer value between 1 and 10, with 1 being the closest to benign and 10 the most anaplastic. Associated with each sample is its class label, which is either benign or malignant.

Since we are extracting Boolean rules, the attribute values are recoded into binary values 0 or 1 using the following scheme

$$\mathcal{A}_i = k \Leftrightarrow \begin{cases} I_{10\times(i-1)+j} = 1 & \forall \quad j = 1, 2, \ldots, k \\ I_{10\times(i-1)+j} = -1 & \forall \quad j = k+1, k+2, \ldots, 10 \end{cases}$$

Hence, in total there are 9×10 input units in the network. The number of output unit is 1, benign samples are given the target value of 0, while malignant samples are given the target value of 1. Binary coding the original data increases the dimensionality of the input data ten fold. This may increase the network training time since the number of network connections increases proportionally. However, this drawback can be more

[1] It is publicly available from http://www.ics.uci.edu/~mlearn/MLRepository

than compensated by the fact that the classification problem in the new input space can be very much simpler than in the original input space. Indeed, for the WBCD data set, after the input data are binary coded as described above, all 699 samples become *linearly separable*. There exists a hyperplane in $\mathbb{R}^{90}$ such that all the benign samples lie on one side of this hyperplane and all the malignant samples lie on the other side.

The two data pre-processing steps are the following:

1. Removal of data with missing attributes values. A small portion of the data (16 out of 699 samples) has 1 missing attribute value. We discard all these samples. This step was also applied by other authors [11, 18], hence fair comparison of our results against theirs can be made. The 683 samples (339 malignant and 444 benign) with complete attribute values are split randomly into a training set that consists of 119 malignant samples and 222 benign samples and a test set that consists of the remaining 120 malignant samples and 222 benign samples.

2. Selection of relevant input attributes. Previous works on this data set indicate that only a small subset of the input attributes is needed for classification. Our approach to reduce the dimensionality of the input data is to employ neural networks with a single hidden unit. Fifty neural networks each having 90 input units, 1 hidden unit and 1 output unit are trained and pruned. Each of these networks is trained to achieve 100% accuracy on the training data set. Network connections are removed as long as the resulting network can still achieve this perfect classification accuracy. The input units left after pruning indicate not only which of the original 9 input attributes of the data are relevant, but they also indicate the cut-off values of each of these relevant attributes. Due to the nonconvexity of the error function (1), networks that have been initialized with different random connection weights will result in different network configuration when the pruning process terminates. As a consequence, the selected input sets are not always the same. This is the reason why we trained and pruned 50

networks initialized with different weights to select the relevant input attributes. Attributes that are still present in 10 or more pruned networks are considered to be important for classification. There are 25 such attributes and they are used to train neural networks for rule generation.

3.2 Results from network training and pruning

We conducted two sets of experiments on the WBCD data training and test data sets. In the first experiment, 100 neural networks each with 3 hidden units were trained. While in the second experiment, another 100 neural networks each with 5 hidden units were trained. Each network connection was assigned a different initial random weights in the interval $[-1, 1]$. Network connections were removed by our N2P2F pruning method [15] as long as the their accuracy rates on the training data set is still 95% or higher. From each training session, we also recorded the network with the fewest connections that correctly classifies at least 98% of the training samples for rule extraction. The parameters in the penalty term (2) were set as follows: $\beta = 10, \epsilon_1 = 1, \epsilon_2 = 10^3$. The results from the experiments are summarized in Table 1.

TABLE 1 HERE

The figures in Table 1 show that by removing samples with missing values, more accurate pruned networks can be obtained. In our previous work [14], we reported average predictive accuracy rates of 92.73% and 93.78% respectively, for the networks pruned to 95% and 98% accuracy on the training samples. The corresponding figures are now 95.44% and 96.66%, thus giving an average increase of almost 3% for each network on the test samples and 1.5% on the entire 683 samples with no missing values. As a result, we are also able to extract rules with better accuracy rates. Some of the pruned networks and the rules extracted from them are presented in the next subsection.

Pruning the networks to achieve 95% classification accuracy reduces the number of network connections significantly. On average, the networks have 8 connections left. However, many of these networks seem to be over-pruned as their accuracy rates both on the training and test data sets drop. The average predictive accuracy drops by slightly more than 1% for the networks with 3 and 5 hidden units. Networks that were pruned to 98% accuracy preserve their predictive accuracy rate of about 96.7%. Is is also worth noting that in order to achieve 1% higher predictive accuracy rates, almost twice as many network connections are needed. For the 3 hidden unit networks, networks with an average number of 7.59 connections achieve 95.51% predictive accuracy. The average number of connections of networks that achieve 1.20 % higher predictive accuracy is 13.89. For the 5 hidden unit networks, an improvement of 1.24% in the predictive accuracy is obtained by the networks with an average of 14.35 connections over those with an average of 8.03 connections.

3.3 Extracted rules

We give three examples of rule sets that are extracted from the pruned networks here. The examples are chosen to illustrate the many different possibilities to describe the WBCD data set with high degree of accuracy. Neural networks trained with different initial weights will yield different final pruned networks. The rules generated from the different pruned networks vary in their complexity and generalization capability as shown by our there examples presented below.

Example 1.

FIGURE 1 HERE

A small network with only 4 input units and 2 hidden units is found to be able to classify the samples in the training and test data sets with more than 97% accuracy rates. This network is depicted in Figure 1. Of the 341 training samples, 331 are correctly classified, giving a classification accuracy rate of 97.07%. The activation values of the 331 samples (216 benign samples and 115 malignant samples) were clustered by GCA and 2 clusters were found at each of the 2 hidden units. At hidden unit H1, all activation values in the subinterval $[-1, 0)$ can be replaced by its lower bound value of -1 and all activation values in the subinterval $[0, 1]$ can be replaced by 0. At hidden unit H2, the 2 cluster values are -0.94 and 0.15. There was no training sample with hidden unit activation that is less than -0.94 at H2. The 2 hidden units left and the 2 clusters at each hidden unit imply that it is possible to represent the WBCD training data as four 2-dimensional samples and correctly distinguish between 216 benign samples and 115 malignant samples.

Let us denote $\alpha_1 = 1$ if a sample activation value at hidden unit H1 falls in the first subinterval $[-1, 0)$, and $\alpha_1 = 2$ otherwise. Similarly, let $\alpha_2 = 1$ indicate that the sample activation value at hidden unit H2 falls in in the subinterval $[-0.94, 0.15)$ and $\alpha_2 = 2$ indicate that the activation value falls in the interval $[0.15, 1]$.

The samples represented by 4 hidden unit cluster combinations can be distinguished by the very simple rule:

Rule 1a:

If $\alpha_1 = 2$ and $\alpha_2 = 1$, the benign,

else malignant.

Instead of having a default rule for malignant class, we can also have rules that describe the samples of this class explicitly:

Rule 1b:

If $\alpha_1 = 1$, then malignant,

If $\alpha_1 = 2$ and $\alpha_2 = 2$, then malignant,

else benign.

Hidden unit H1 is connected to $\mathcal{I}_{16}$ and $\mathcal{I}_{55}$, while hidden unit H2 is connected to $\mathcal{I}_4$ and $\mathcal{I}_{72}$. The next step of NeuroRule is to determine what combinations of these inputs produce the different cluster activation values. For this purpose, we make use of X2R [8]. It takes as input a set of discrete patterns with the class labels and produces the rules describing the relationship between the patterns and their class labels. For H1, it found that $\alpha_1 = 1$ iff $\mathcal{I}_{16} = 1$ or $\mathcal{I}_{55} = 1$, otherwise $\alpha_1 = 2$. For H2, $\alpha_2 = 1$ iff $\mathcal{I}_4 = \mathcal{I}_{72} = 0$, otherwise $\alpha_2 = 2$.

We replace the rule conditions in Rule set 1b to obtain the following rules in terms of the binary encoded inputs of the WBCD data set:

Rule 1c:

If $\mathcal{I}_4 = \mathcal{I}_{16} = \mathcal{I}_{55} = \mathcal{I}_{72} = 0$, then benign,

else malignant.

or the second set of rules that describes malignant samples:

Rule 1d:

If $\mathcal{I}_{16} = 1$, then malignant,

else if $\mathcal{I}_{55} = 1$, then malignant,

else if $\mathcal{I}_4 = 1$, then malignant,

else if $\mathcal{I}_{72} = 1$, then malignant,

else benign.

In terms of the original 9 attributes of the data, the corresponding rule set is either the set that describes the benign samples:

Rule 1e:

If $\mathcal{A}_1 \leq 6, \mathcal{A}_2 \leq 4, \mathcal{A}_6 \leq 5, \mathcal{A}_8 \leq 8$, then benign,

else malignant.

or a second set of rules that describes malignant samples:

Rule 1f:

If $\mathcal{A}_2 \geq 5$, then malignant,

else if $\mathcal{A}_6 \geq 6$, then malignant,

else if $\mathcal{A}_1 \geq 7$, then malignant,

else if $\mathcal{A}_8 \geq 9$, then malignant,

else benign.

The accuracy rates of the rule sets 1e and 1f on both the training and test data sets are exactly the same as the accuracy of the pruned neural network from which they were generated. These are summarized in Table 2.

TABLE 2 HERE

Example 2.

FIGURE 2 HERE

The network chosen for this example has 3 hidden units left after pruning (Figure 2). Hidden unit H1 is connected to input unit $\mathcal{I}_{16}$, hidden unit H2 to input unit $\mathcal{I}_{54}$, and hidden unit H3 to input units $\mathcal{I}_5, \mathcal{I}_{36}, \mathcal{I}_{58}$ and $\mathcal{I}_{78}$. Two clusters were found by the clustering algorithm at each of the 3 hidden units. The rule set that is extracted from this pruned network is as follows:

Rule 2a:

If $\mathcal{A}_2 \leq 4$ and $\mathcal{A}_6 \leq 2$ and $\mathcal{A}_8 \leq 2$, then benign,

else if $\mathcal{A}_1 \leq 6$ and $\mathcal{A}_1 \leq 5$ and $\mathcal{A}_2 \leq 4$ and $\mathcal{A}_8 \leq 2$, then benign,

else if $\mathcal{A}_1 \leq 5$ and $\mathcal{A}_2 \leq 4$ and $\mathcal{A}_4 \leq 4$ and $\mathcal{A}_6 \leq 2$, then benign,

else malignant.

A set of rules that describe the malignant samples instead of the benign samples can also be generated from the same pruned network:

Rule 2b:

If $\mathcal{A}_2 \geq 5$, then malignant,

else if $\mathcal{A}_6 \geq 7$, then malignant,

else if $\mathcal{A}_6 \geq 3$ and $\mathcal{A}_8 \geq 3$, then malignant,

else if $\mathcal{A}_1 \geq 6$ and $\mathcal{A}_6 \geq 3$, then malignant,

else if $\mathcal{A}_4 \geq 5$ and $\mathcal{A}_8 \geq 3$, then malignant,

else if $\mathcal{A}_1 \geq 6$ and $\mathcal{A}_8 \geq 3$, then malignant,

else benign.

The accuracy rates of Rule 2a and Rule 2b are the same, but they are slightly higher than the accuracy of the pruned network. A malignant sample in the training data set that was misclassified by the pruned network is correctly classified by the rules. Only samples that have been correctly classified by a pruned network are used to find the clusters of hidden unit activation values. The rules extracted from the network will correctly classified all these samples. However, it is possible that one or more samples that are misclassified by the network to be correctly classified by the rules as this example shows. For this example, it turns out the rules also correctly classify one benign sample in the test set that was misclassified by the network. The accuracy rates of the rules are summarized in Table 3.

TABLE 3 HERE

Example 3.

One of the pruned networks with the highest overall accuracy rate of 98.24% has only 2 hidden units and 9 connections. The rule set extracted by NeuroRule is as follows:

Rule 3:

If $\mathcal{A}_2 \leq 4$ and $\mathcal{A}_6 \leq 2$ and $\mathcal{A}_8 \leq 2$, then benign,

else if $\mathcal{A}_2 \leq 4$ and $\mathcal{A}_6 \leq 2$ and $\mathcal{A}_8 \leq 8$ and $\mathcal{A}_1 \leq 6$, then benign,

else if $\mathcal{A}_1 \leq 5$ and $\mathcal{A}_4 \leq 4$ and $\mathcal{A}_6 \leq 5$ and $\mathcal{A}_8 \leq 2$, then benign,

else if $\mathcal{A}_1 = 6$ and $\mathcal{A}_2 \leq 4$ and $\mathcal{A}_6 \geq 6$ and $\mathcal{A}_8 \leq 8$, then benign,

else if $\mathcal{A}_2 \leq 4$ and $\mathcal{A}_4 \geq 5$ and $\mathcal{A}_6 \leq 5$ and $3 \leq \mathcal{A}_6 \leq 5$ and $\mathcal{A}_8 \leq 8$, then benign,

else malignant.

TABLE 4 HERE

The accuracy rates of Rule 3 are shown in Table 4. The classification accuracy of these rules is slightly higher than the accuracy of the network and the rules in Example 2. However, their predictive accuracy is lower. This example clearly illustrates that a network that fits the training data better with more parameters does not necessarily generalize better. An additional disadvantage of having a pruned network that overfits the data is that the rule set generated from it will be more complex. Compared to the rule sets in Examples 1 and 2, this rule set has more rules and more conditions per rule on average.

3.4 Comparison with other rule extraction methods

TABLE 5 HERE

Several other authors have made use of the WBCD data set to measure the performance of their rule generating algorithms. Taha and Gosh [18, 19] proposed 3 algorithms for rule extraction from neural networks. These algorithms are Binarized Input-Output Rule Extraction (BIO-RE), Partial Rule Extraction (Partial-RE) and Full Rule Extraction (Full-RE). BIO-RE is a black box rule extraction technique which does not require informations regarding the internal network structure to generate rules. The method can be applied to data with binary attributes only. A truth table is constructed from the pairs of input samples and their network outputs, logic minimization tools are then utilized to obtain the optimal binary classification rules.

Partial-RE searches for a set of incoming connections to a unit that will caused the unit to be active. To speed up the search, weight connections to each hidden unit and output unit are first sorted. Positive weights are grouped into one set and negative weights into a second set. The rule generated can be of the form: *if (input units with positive connections are active), then output unit is active* or *if (input units with negative connections are not active), then output unit is active*, or the combination of the two. Partial-RE assumes that all inputs have the same range so that the effect of these inputs on a hidden unit is determined by their weights only.

The third technique Full-RE is similar to NeuroRule, both methods decompose the rule extraction process into two steps: rules between hidden and output units and rules between input units and hidden units. The difference is that Full-RE employs linear programming and an input discretization method to find a combination of the input values that will cause a hidden unit to be active.

A rule generating method that combines genetic algorithm and fuzzy logic is proposed by Peña-Reyes and Sipper [11]. The number of rules to be generated by the method needs to be determined a priori. For WBCD, this ranges from 1 to 5. However, the number of conditions per rule is automatically found by the evolutionary algorithm. A rule condition is composed of one or more simple fuzzy expressions joined by fuzzy operators. A simple fuzzy rule condition is either *if input is Low* or *if input is High*. For each of the 9 input attributes, two parameters are needed to encode its membership function. P and d are the parameters that define the start point and the length of membership function edges, respectively. To obtain their results, the authors fixed the population size to 200 individuals. The algorithm terminates when the maximum number of generations is reached or when there is no significant increase in the fitness of successive generations. A second set of experiments was also conducted by the authors where they restricted the number of conditions per rule. To make up for less conditions per rule, they increased the number of rules up to 7.

Table 5 compares the performance of NeuroRule and the other algorithms. BIO-RE, Partial-RE and Full-RE results were reported by Taha and Gosh in [19]. Fuzzy-GAs are the results obtained by Peña-Reyes and Sipper from their fuzzy genetic approach. Fuzzy-GA results shown in Table 5 are summarized according to the number of rules since since their algorithm requires it to be fixed by the user.

From Table 5 we can see that NeuroRule generates more compact sets of rules. Rule 1e has only 1 rule with 4 conditions. The only other algorithm that can generate a rule set with only 1 rule is the one that specifically searches for this type of rules, namely Fuzzy-GA1. Fuzzy-GA1 rule however, is not as accurate as Rule 1e of NeuroRule. Fuzzy-GA1 rule has 4 conditions, but it should be noted that a fuzzy rule condition such as Low or High is actually defined by 2 parameters P and d. Hence, the number of conditions per rule of Fuzzy-GAs is actually twice as much.

The highest overall accuracy is 98.24% obtained by Rule 3 and by Fuzzy-GA4. Fuzzy-GA4 rule set has on average 5.8 fuzzy conditions per rule, while Rule 3 consists of 5 rules with 21 crisp conditions. From the figures in Table 5, we may conclude that NeuroRule and the fuzzy genetic approach produce rule sets that are comparable in accuracy, but the rule sets of the former are smaller that those of the latter.

The accuracy of the rules obtained by BIO-RE, Partial-RE and Full-RE are lower than those obtained by the NeuroRule or the fuzzy genetic approach. The number of conditions per rule of these 3 algorithms are comparable to those of NeuroRule, but more rules are generated by BIO-RE, Partial-RE and Full-RE. It is pointed out by Taha and Gosh [18, 19] that the accuracy rates of NeuroRule drop significantly when the default rule is not taken into account. They also pointed out that it may not be desirable to classify a sample by default, that is, a sample is classified as either benign or malignant because "none of the above conditions" is satisfied. A complete rule set that covers each sample in the training data set by at least one of the rules in the set can be generated by NeuroRule. Examples 1 and 2 clearly illustrate this possibility. We can simply merge Rule 1e and Rule 1f for example, to obtain rules for both benign and malignant cases. When this is done, the overall accuracy of the resulting merged rule set is still 97.36%. The total number of rules will be 5 and the number of conditions per rule is 1.6. The resulting merged rule set is more compact and the overall accuracy is more than 1% higher than Full-RE's rule set. Having a default rule is still necessary even when rules that cover all samples in the training data have been generated. Without a default rule, it is possible that a new sample cannot be classified because its attribute values do not satisfy the conditions of any of the rules.

4 Conclusion

In this paper, we present the application of our method for rule extraction from neural networks on the Wisconsin Breast Cancer Diagnosis dataset. The paper highlights the usefulness of data pre-processing before rules are extracted or neural networks are trained. The data pre-processing step entails the removal of samples with missing attribute values and the selection of a subset of the original attributes for classification. The advantages of data pre-processing include higher network predictive accuracy and faster network training. More accurate rule sets can be extracted from pruned networks that achieve high accuracy rates. Faster network training time allows us to train many networks which will provide us with different pruned network structures to choose from for rule extraction. We note that our algorithm NeuroRule can be applied to extract rules from any pruned neural networks with a single hidden layer. It can be expected that networks with higher classification accuracy will generate larger sets of rules. This, however, does not imply that the networks and the rules extracted from them necessarily have better predictive accuracy as the examples presented in this paper show.

We compared the results from our method against those of other rule extraction methods. Our method can generate more compact rules with higher overall accuracy rates. We hope that the results presented here will lead to more application of the method to other problem domains.

References

[1] R. Andrews, J. Diederich and A. Tickle, Survey and critique of techniques for extracting rules from trained artificial neural networks, *Knowledge Based Systems*, 8(6) (1995) 373–389.

[2] B. Apolloni, G. Avanzini, N. Cesa-Bianci and G. Ronchine, Diagnosis of epilepsy

via backpropagation, in: *Proc. International Joint Conf. on Neural Networks, Vol. II*, Washington D.C (1990) 517–574.

[3] W. Baxt, Use of an artificial neural network for data analysis in clinical decision making, *Neural Computation* 2 (1990) 480–489.

[4] K.P. Bennett and O.L. Mangasarian, Neural network training via linear programming, in: P.M. Pardalos ed., *Advances in Optimization and Parallel Computing,* (Elsevier Science Publishers B.V., Amsterdam, 1990) 56–67.

[5] J. Boone, G. Gross and G. Shaber, Computer aided radiologic diagnosis using neural networks, in: *Proc. Internat. Joint Conf. on Neural Networks, Vol. II*, Washington D.C (1990) 98-101.

[6] S. Cho and J.A. Reggia, Multiple disorder diagnosis with adaptive competitive neural networks, *Artificial Intelligence in Medicine* 5 (1993) 469-487.

[7] D.B. Fogel, E.C. Wasson III, E.M. Boughton and V.W. Porto, Evolving artificial neural networks for screening features from mammograms, *Artificial Intelligence in Medicine* 14(3) (1998) pp. 317

[8] X2R: A Fast Rule Generator, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, IEEE Press, New York (1995) 388–391.

[9] B. Mulsant and E. Servan-Schreiber, A connectionist approach to the diagnosis of dementia, in: *Proc. Internat. Joint Conf. on Neural Networks, Vol. I*, San Diego, CA (1990) 33-38.

[10] C.D. Nugent, J.A.C. Webb, M. McIntyre, N.D. Black and G.T.H. Wright, Computerised electrocardiology employing bi-group neural networks, *Artificial Intelligence In Medicine* 13(3) (1998) 167.

[11] Carlos Andrés Peña-Reyes and Moshe Sipper, *Artificial Intelligence in Medicine* (1999) forthcoming.

[12] E. Pesonen, M. Eskelinen and M. Juhola, Treatment of missing data values in a neural network based decision support system for acute abdominal pain, *Artificial Intelligence in Medicine* 13(3) (1998) 139–146.

[13] R. Setiono and H. Liu, Symbolic representation of neural networks, *Computer* March (1996) 71–77.

[14] R. Setiono, Extracting rules from pruned neural networks for breast cancer diagnosis, *Artificial Intelligence In Medicine* 8(1) (1996) 37–51.

[15] R. Setiono, A penalty function approach for pruning feedforward neural networks, *Neural Computation* (9)1 (1997) 185–204.

[16] H. Liu and R. Setiono, Feature selection via discretization of numeric attributes, *IEEE Trans. on Knowledge and Data Engineering* (9)4 (1997) 642–645.

[17] R. Setiono and H. Liu, Analysis of hidden representations by greedy clustering, *Connection Science* (10)1 (1998) 21–42.

[18] I. Taha and J. Gosh, Symbolic interpretation of artificial neural networks, Tech. Report TR-97-01-106, the Computer and Vision Research Center, University of Texas, Austin, 1996.

[19] I. Taha and J. Gosh, Characterization of the Wisconsin breast cancer database using a hybrid symbolic-connectionist system, Tech. Report UT-CVISS-TR-97-007, the Computer and Vision Research Center, University of Texas, Austin, 1996.

[20] C. Ulbricht, G. Dorffner, A. Lee, Neural networks for recognizing patterns in cardiotocograms, *Artificial Intelligence in Medicine* (12)3 (1998) 271.

[21] Y. Yoon R. Brobst, P. Bergstresser and L. Peterson, A desktop neural network for dermatology diagnosis, *Neural Network Comput.* (Summer 1989) 25-43.

Table captions:

Table 1 Summary of the results from 100 networks with h = 3 hidden units and 100 networks with h = 5 hidden units. The values shown are the averages and in parentheses, the standard devisions.

Table 2 The accuracy of Rules 1e and 1f.

Table 3 The accuracy of Rules 2a and 2b.

Table 4 The accuracy of Rule set 3.

Table 5 Comparison of NeuroRule and other rule extraction algorithms on WBCD.

Figure captions:

Figure 1 A pruned network with only 4 connections from the input units to the hidden units. Its classification accuracy is 97.07% and its predictive accuracy is 97.66%. Numbers shown are connection weights.

Figure 2 A pruned network that achieves 97.65% classification accuracy and 97.95% predictive accuracy.

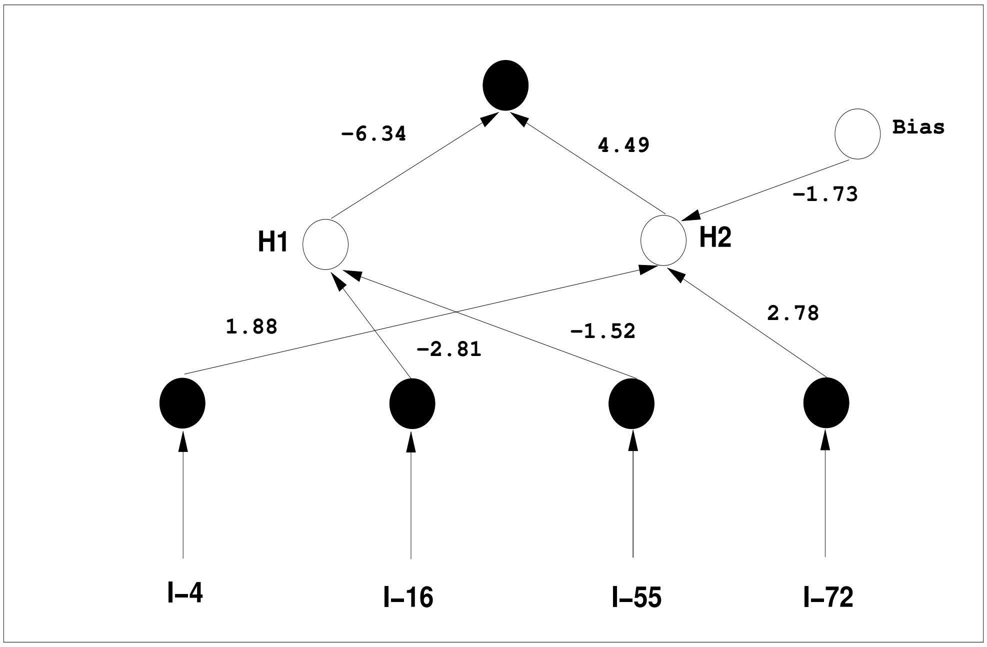
h = 3	**Before pruning:**		
	Number of connections	81 (0.00)	
	Accuracy on training data	99.36 (0.50)	
	Accuracy on test data	96.67 (0.63)	
	Overall accuracy	98.01 (0.30)	
	After pruning:	**95%**	**98%**
	Number of connections	7.59 (1.75)	13.89 (2.58)
	Accuracy on training data	95.83 (0.70)%	98.40 (0.22)%
	Accuracy on test data	95.51 (1.25)%	96.71 (0.57)%
	Overall accuracy	95.67 (0.83)%	97.56 (0.29)%
h = 5	**Before pruning:**		
	Number of connections	135 (0.00)	
	Accuracy on training data	99.50 (0.32)	
	Accuracy on test data	96.52 (0.62)	
	Overall accuracy	98.01 (0.31)	
	After pruning:	**95%**	**98%**
	Number of connections	8.03 (1.96)	14.35 (2.33)
	Accuracy on training data	95.87 (0.70)%	98.43 (0.25)%
	Accuracy on test data	95.36 (1.29)%	96.60 (0.63)%
	Overall accuracy	95.61 (0.89)%	97.52 (0.34)%

	Training set	Test set	Training and test sets
Malignant samples	115/119 96.64 %	117/120 97.50 %	232/239 97.07 %
Benign samples	216/222 97.30 %	217/222 97.75 %	433/444 97.52 %
Overall	331/341 97.07 %	334/342 97.66 %	665/683 97.36 %

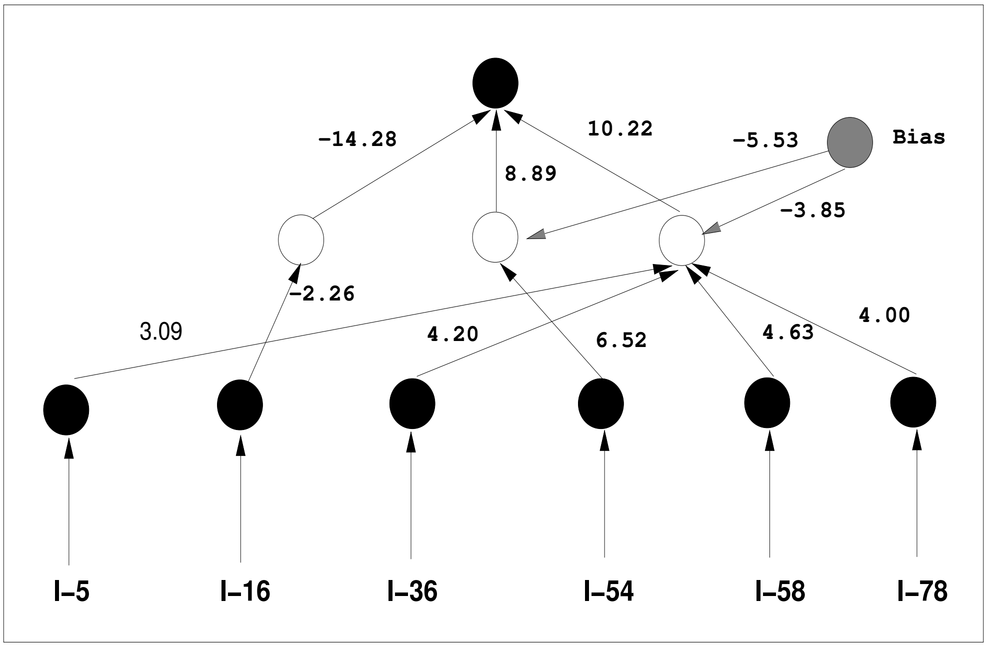
	Training set	Test set	Training and test sets
Malignant samples	118/119 99.16 %	120/120 100.00 %	238/239 99.58 %
Benign samples	216/222 97.30 %	216/222 97.30 %	432/444 97.30 %
Overall	334/341 97.95 %	336/342 98.25 %	670/683 98.10 %

	Training set	Test set	Training and test sets
Malignant samples	118/119 99.16 %	119/120 99.17 %	237/239 99.17 %
Benign samples	218/222 98.20 %	216/222 97.30 %	434/444 97.75 %
Overall	336/341 98.53 %	335/342 97.95 %	671/683 98.24 %

Method	#Rules	#conditions	#conditions per rule	Overall accuracy
NeuroRule - Rule 1e	1	4	4.0	97.36%
NeuroRule - Rule 1f	4	4	1.0	97.36%
NeuroRule - Rule 2a	3	11	3.7	98.10%
NeuroRule - Rule 2b	6	10	1.7	98.10%
NeuroRule - Rule 3	5	21	4.2	98.24%
BIO-RE	10	30	3.0	96.63%
Partial-RE	9	24	2.7	96.49%
Full-RE	5	9	1.8	96.19%
Fuzzy-GA1	1	4	4.0	97.07%
Fuzzy-GA2	2	6	3.0	97.36%
Fuzzy-GA3	3	16	5.3	97.66%
Fuzzy-GA4	4	23	5.8	98.24%
Fuzzy-GA5	5	30	6.0	97.95%
Fuzzy-GA6	6	37	6.2	98.10%
Fuzzy-GA7	7	35	5.0	97.95%



-6.34
4.49
Bias
-1.73
H1
H2
1.88
-2.81
-1.52
2.78
I-4
I-16
I-55
I-72



-14.28
10.22
-5.53
Bias
8.89
-3.85
-2.26
3.09
4.20
6.52
4.63
4.00
I–5
I–16
I–36
I–54
I–58
I–78

Table 1: Summary of the results from 100 networks with h = 3 hidden units and 100 networks with h = 5 hidden units. The values shown are the averages and in parentheses, the standard devisions.

h = 3	**Before pruning:**		
	Number of connections	81 (0.00)	
	Accuracy on training data	99.36 (0.50)	
	Accuracy on test data	96.67 (0.63)	
	Overall accuracy	98.01 (0.30)	
	After pruning:	**95%**	**98%**
	Number of connections	7.59 (1.75)	13.89 (2.58)
	Accuracy on training data	95.83 (0.70)%	98.40 (0.22)%
	Accuracy on test data	95.51 (1.25)%	96.71 (0.57)%
	Overall accuracy	95.67 (0.83)%	97.56 (0.29)%
h = 5	**Before pruning:**		
	Number of connections	135 (0.00)	
	Accuracy on training data	99.50 (0.32)	
	Accuracy on test data	96.52 (0.62)	
	Overall accuracy	98.01 (0.31)	
	After pruning:	**95%**	**98%**
	Number of connections	8.03 (1.96)	14.35 (2.33)
	Accuracy on training data	95.87 (0.70)%	98.43 (0.25)%
	Accuracy on test data	95.36 (1.29)%	96.60 (0.63)%
	Overall accuracy	95.61 (0.89)%	97.52 (0.34)%

Table 2: The accuracy of Rules 1e and 1f.

	Training set	Test set	Training and test sets
Malignant samples	115/119 96.64 %	117/120 97.50 %	232/239 97.07 %
Benign samples	216/222 97.30 %	217/222 97.75 %	433/444 97.52 %
Overall	331/341 97.07 %	334/342 97.66 %	665/683 97.36 %

Table 3: The accuracy of Rules 2a and 2b.

	Training set	Test set	Training and test sets
Malignant samples	118/119 99.16 %	120/120 100.00 %	238/239 99.58 %
Benign samples	216/222 97.30 %	216/222 97.30 %	432/444 97.30 %
Overall	334/341 97.95 %	336/342 98.25 %	670/683 98.10 %

Table 4: The accuracy of Rule set 3.

	Training set	Test set	Training and test sets
Malignant samples	118/119 99.16 %	119/120 99.17 %	237/239 99.17 %
Benign samples	218/222 98.20 %	216/222 97.30 %	434/444 97.75 %
Overall	336/341 98.53 %	335/342 97.95 %	671/683 98.24 %

Table 5: Comparison of NeuroRule and other rule extraction algorithms on WBCD.

Method	#Rules	#conditions	#conditions per rule	Overall accuracy
NeuroRule - Rule 1e	1	4	4.0	97.36%
NeuroRule - Rule 1f	4	4	1.0	97.36%
NeuroRule - Rule 2a	3	11	3.7	98.10%
NeuroRule - Rule 2b	6	10	1.7	98.10%
NeuroRule - Rule 3	5	21	4.2	98.24%
BIO-RE	10	30	3.0	96.63%
Partial-RE	9	24	2.7	96.49%
Full-RE	5	9	1.8	96.19%
Fuzzy-GA1	1	4	4.0	97.07%
Fuzzy-GA2	2	6	3.0	97.36%
Fuzzy-GA3	3	16	5.3	97.66%
Fuzzy-GA4	4	23	5.8	98.24%
Fuzzy-GA5	5	30	6.0	97.95%
Fuzzy-GA6	6	37	6.2	98.10%
Fuzzy-GA7	7	35	5.0	97.95%

Figure 1: A pruned network with only 4 connections from the input units to the hidden units. Its classification accuracy is 97.07% and its predictive accuracy is 97.66%. Numbers shown are connection weights.

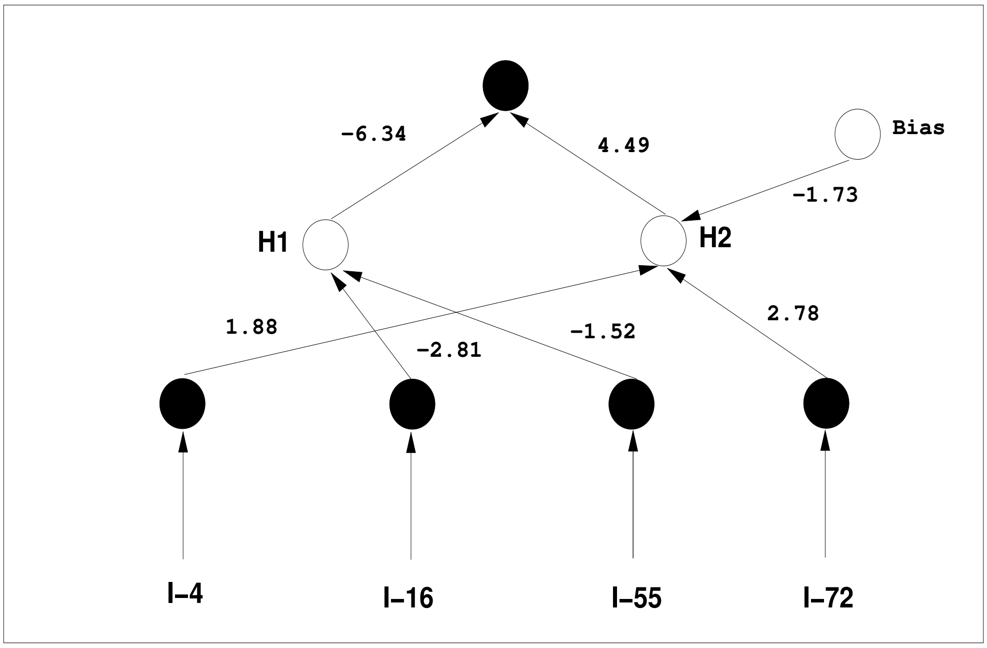


Figure 2: A pruned network that achieves 97.65% classification accuracy and 97.95% predictive accuracy.