Appears in Artificial Intelligence in Medicine, Vol. 8, No. 1, February 1996, pp. 37–51

Extracting rules from pruned neural networks for breast cancer diagnosis

Rudy Setiono

Department of Information Systems and Computer Science

National University of Singapore

Kent Ridge, Singapore 0511

Republic of Singapore

Email:rudys@iscs.nus.sg

Phone:(65) 772-6297

Fax :(65) 779-4580

Abstract

A new algorithm for neural network pruning is presented. Using this algorithm, networks with small number of connections and high accuracy rates for breast cancer diagnosis are obtained. We will then describe how rules can be extracted from a pruned network by considering only a finite number of hidden unit activation values. The accuracy of the extracted rules is as high as the accuracy of the pruned network. For the breast cancer diagnosis problem, the concise rules extracted from the network achieve an accuracy rate of more than 95 % on the training data set *and* on the test data set.

Keywords. Neural network pruning; penalty function; rule extraction; breast cancer diagnosis.

1 Introduction

Neural networks techniques have recently been applied to many medical diagnostic problems [1, 2, 4, 5, 11, 22]. Although the predictive accuracy of neural networks is often higher than that of other methods or human experts, it is generally difficult to understand *how* the network arrives at a particular conclusion due to the complexity of the network architecture. However, it is often desirable to have a set of comprehensible rules that describe under what conditions a pattern will be classified as a member of a certain class.

The complexity of a network can be reduced by removing its connections that are redundant through pruning. Many algorithms for neural network pruning have been proposed in the past few years. It has often been mentioned in the literature [6, 7, 8, 10, 17] that neural network pruning is beneficial in two ways. The first advantage is that a pruned network can achieve a higher accuracy rate on new patterns not used for training. The second advantage, which will be the focus of this paper, is that rules may be extracted from a network with a small number of connections. In this paper we describe an algorithm for pruning a standard three-layer feedforward neural network. We use a breast cancer diagnosis problem to demonstrate the effectiveness of this pruning algorithm. The pruned networks have a small number of connections and yet their accuracy rates are high for patterns in the training set and in the testing set. The small number of connections left in the pruned network allows us to extract meaningful rules that describe the classification process.

When the hyperbolic tangent function $\psi(\xi) = (e^{\xi} - e^{-\xi})/(e^{\xi} + e^{-\xi})$ is used as the hidden unit activation function, the hidden unit activation of a neural network can take any value in the interval $[-1, 1]$. We will describe in this paper, how the activation values of a hidden unit can be clustered such that only a finite and usually small number of discrete values need to be considered while at the same time maintaining the network accuracy. A small number of different discrete activation values and a small number of connections from the input units to the hidden units will yield a set of compact rules for

the problem.

This paper is organized as follows. In Section 2 we describe our neural network pruning algorithm. Our experimental results obtained from applying this algorithm on the Wisconsin Breast Cancer Diagnosis problem are presented in Section 3. In Section 4 we describe our algorithm for extracting rules from a pruned network. Rules that are extracted for the breast cancer diagnosis problem are presented here. Finally in Section 5 we conclude the paper.

2 Pruning a feedforward neural network

The neural network that we use for our experiments is the standard three layer feedforward network [14]. Connections in the network are allowed only between input units and hidden units and between hidden units and output units. We denote the weights of the connections between input units and hidden units by $w_\ell^j, \ell = 1, 2, \ldots, n, j = 1, 2, \ldots, h$ and the weights of the connections between the hidden units and the output units by $v_p^j, p = 1, 2, \ldots, o, j = 1, 2, \ldots, h$, where h is the number of hidden units, n is the dimensionality of the input patterns and o is the number of output units. The number of output units is equal to the number of classes present in the data set. Let us denote the classes in the data as $\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_o$, then each pattern x^i in class $\mathcal{C}_c$ will be given a target value which is an o-dimensional vector t^i such that $t_m^i = 0, \forall m \neq c$ and $t_c^i = 1$.

Given a set of input patterns $x^i \in \mathbb{R}^{\mathrm{n}}, \mathrm{i} = 1, 2, \ldots \mathrm{k}$, we compute the best set of weights (w, v) by minimizing the cross entropy function:

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

S_p^i is the output of the network

$$S_p^i = \sigma \left(\sum_{j=1}^{h} \delta \left((x^i)^T w^j \right) v_p^j \right), \tag{2}$$

where $(x^i)^T w^j$ denotes the scalar product of the vectors x^i and w^j, $\delta(.)$ is the hyperbolic tangent function, and $\sigma(.)$ is the sigmoid function. It has been suggested that faster

convergence can be achieved by minimizing this entropy function instead of the usual squared error function [13].

Since our goal is to find and eliminate as many unneeded network connections as possible, it is important that they are identified during the training process. A penalty term is normally added to the error function so that at the end of the training process, unnecessary connections will have very small weights. Connections with small weights may be eliminated from the networks without too much effect on the accuracy of the network. Recently, we have proposed a new penalty function for neural network pruning [15]. The penalty term is defined as

$$P(w,v) = \epsilon_1 \left(\sum_{j=1}^{h} \sum_{\ell=1}^{n} \frac{\beta (w_\ell^j)^2}{1+\beta (w_\ell^j)^2} + \sum_{j=1}^{h} \sum_{p=1}^{o} \frac{\beta (v_p^j)^2}{1+\beta (v_p^j)^2} \right) + \tag{3}$$

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

The values for the weight decay parameters $\epsilon_1, \epsilon_2 > 0$ must be chosen to reflect the relative importance of the accuracy of the network versus its complexity. More weights may be removed from the network at the cost of a decrease in its accuracy with larger values of these two parameters. They also determine the range of values where the penalty for each weight in the network is approximately equal to ϵ_1. The parameter $\beta > 0$ determines the steepness of the error function near the origin.

The derivatives of S_p^i with respect to the weights of the networks are as follows

$$\frac{\partial S_p^i}{\partial w_\ell^j} = S_p^i \times (1 - S_p^i) \times v_p^j \times x_\ell^i \times \left(1 - \delta((x^i)^T w^j)^2 \right)$$

$$\frac{\partial S_p^i}{\partial v_q^j} = \begin{cases} S_p^i \times (1 - S_p^i) \times \delta((x^i)^T w^j) & \text{if } p = q \\ 0 & \text{otherwise,} \end{cases}$$

for all $i = 1, 2, \ldots, k$, $\ell = 1, 2, \ldots, n$, $j = 1, 2 \ldots, h$, and $p, q = 1, 2, \ldots o$. For the moment, let us consider S_p^i as a function of a single variable corresponding to the connection between the ℓth input unit and the jth hidden unit. By the mean value theorem, we have that

$$S_p^i(w) = S_p^i(w_\ell^j) + \frac{\partial S_p^i(w_\ell^j + \lambda(w - w_\ell^j))}{\partial w_\ell^j} \times (w - w_\ell^j), \tag{4}$$

where $0 < \lambda < 1$. Assuming that $x_\ell^i \in [0,1]$, it follows that

$$\left| \frac{\partial S_p^i(w)}{\partial w_\ell^j} \right| \leq |v_p^j|/4, \ \forall w \in \mathbb{R}. \tag{5}$$

Combining Equations (4) and (5) and letting w be equal to zero, we obtain

$$\left| S_p^i(0) - S_p^i(w_\ell^j) \right| \leq |v_p^j w_\ell^j|/4. \tag{6}$$

The above inequality gives an upper bound on the change in the output of the network when the weight w_ℓ^j is eliminated.

Similarly, by considering S_p^i as a function of a single variable v that corresponds to the connection between the jth hidden unit and the pth output unit, we have the bound

$$\left| \frac{\partial S_p^i(v)}{\partial v_p^j} \right| \leq 1/4, \ \forall v \in \mathbb{R}. \tag{7}$$

Hence, the change in the output of the network after the weight v_p^j has been eliminated is bounded by

$$\left| S_p^i(0) - S_p^i(v_p^j) \right| \leq |v_p^j|/4. \tag{8}$$

A pattern is correctly classified if the following condition is satisfied

$$\max_p |e_p^i| = \max_p |S_p^i - t_p^i| \leq \eta_1, \tag{9}$$

where $\eta_1 \in [0, 0.5)$. Suppose now that with the original fully connected network, we have classified correctly pattern x^i. Equation (6) shows that we can set w_ℓ^j to zero without deteriorating the overall accuracy rate of the network if the product $|v^j w_\ell^j|$ is sufficiently small. If $\max_p |v_p^j w_\ell^j| \leq 4\eta_2$ and the sum $\eta_1 + \eta_2$ is less than 0.5, then

$$\begin{aligned} \left| S_p^i(0) - t_p^i \right| &\leq \left| S_p^i(0) - S_p^i(w_\ell^j) \right| + \left| S_p^i(w_\ell^j) - t_p^i \right| \\ &\leq \eta_1 + \eta_2 \\ &< 0.5. \end{aligned}$$

Hence, the network can still classify x^i correctly. Similarly, if $|v_p^j| \leq 4\eta_2$, then v_p^j can be removed from the network.

We summarize our pruning algorithm below.

Neural network pruning algorithm

1. Let η_1 and η_2 be positive scalars such that $\eta_1 + \eta_2 < 0.5$.

2. Pick a fully connected network. Train this network until a predetermined accuracy rate is achieved and for each correctly classified pattern condition (9) is satisfied. Let (w, v) be the weights of this network.

3. For each w_ℓ^j, if
$$\max_p |v_p^j \times w_\ell^j| \leq 4\eta_2, \tag{10}$$
then remove w_ℓ^j from the network

4. For each v_p^j, if
$$|v_p^j| \leq 4\eta_2, \tag{11}$$
then remove v_p^j from the network

5. If no weight satisfies condition (10) or condition (11), then remove w_ℓ^j with the smallest product $\max_p |v_p^j \times w_\ell^j|$.

6. Retrain the network. If classification rate of the network falls below an acceptable level, then stop. Otherwise, go to Step 3.

Note that in Step 3 and 4 of the algorithm, we remove all the weights that satisfy condition (10) or condition (11). This is intended to reduce the amount of retraining time. Although it can no longer be guaranteed that the retrained pruned network will give the same accuracy rate as the original network, our experiments show that many weights can be eliminated simultaneously without deteriorating the performance of the network. This is especially true when the starting fully connected network has an excessive number of redundant hidden units.

The two conditions (10) and (11) for pruning depend on the magnitude of the weights for connections between input units and hidden units and between hidden units and output units, it is imperative that during training these weights be prevented from getting too large. At the same time, small weights should be encouraged to decay rapidly to zero. We will also see in Section 4 how the magnitudes of the connections between hidden units and output units play an important role in determining the complexity of the rules that can be extracted from a pruned network. These are the reasons why we have chosen to use penalty function (3). To conclude, the function to be minimized during the training process is

$$\theta(w,v) = -\sum_{i=1}^{k}\sum_{p=1}^{o}\left(t_p^i \log S_p^i + (1-t_p^i)\log(1-S_p^i)\right) + P(w,v). \tag{12}$$

3 The Wisconsin Breast Cancer Diagnosis problem

The database for the Wisconsin Breast Cancer Diagnosis is available publicly via anonymous ftp from the University of California Irvine repository [12]. This data set has been used as the test data for several studies on pattern classification methods using linear programming techniques [3, 9, 20] and statistical techniques [21].

Each pattern in the data set has nine attributes. The nine measurements taken from fine needle aspirates from human breast tissues correspond to cytological characteristics of benign or of malignant sample. These are $\mathcal{A}_1$. 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. Each of these nine attributes of the fine needle aspirates was graded 1 to 10 at the time of sample collection, with 1 being the closest to benign and 10 the most anaplastic (more detailed description of these attributes can be found in [20]). Since the attributes are integer-valued ranging from 1 to 10, we created 10 input units for each attribute. With an additional input for the bias weight at the hidden units, we have a total of 91 input units. Let us denote these inputs as $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_{91}$. For $i = 0, 1, \ldots, 8$ the following

coding schemes for the input data is used

$$\begin{aligned} \mathcal{I}_{10\times i+j} &= 1 \iff \mathcal{A}_{i+1} \geq 11-j,\ j=1,2,\ldots,10 \\ \mathcal{I}_{10\times i+j} &= 0 \iff \mathcal{A}_{i+1} \leq 10-j,\ j=1,2,\ldots,9, \\ \mathcal{I}_{91} &= 1. \end{aligned}$$

Note that with this coding, $\mathcal{I}_{10\times j}=1$ for all $j=1,2,\ldots,9$ for all patterns with valid attribute values in $\{1,2,\ldots,10\}$.

There are a total of 699 samples in the database, of which 458 are benign samples and 241 are malignant samples. We randomly selected 229 benign samples and 121 malignant samples to form the training set and used the remaining patterns as the testing data. The number of output units is two, since the problem here is to distinguish between benign and malignant samples. The target value for all benign samples is $t^i=\{0,1\}$, and for the malignant samples the target value is $t^i=\{1,0\}$.

Two set of experiments were conducted. In the first set, fifty networks with 3 hidden units were trained. The initial weights of these networks were generated randomly in the interval $[-1,1]$. The networks were then pruned. The penalty function (12) was minimized using a variant of the quasi-Newton for unconstrained minimization to speed up convergence. This method is the BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm which have been shown to be superior to the backpropagation method for network training [16, 19]. The parameters of the penalty function $P(w,v)$ were set as follows: $\beta=100, \epsilon_1=1, \epsilon_2=10^{-3}$.

For each network, we recorded the smallest number of connections that were present in the network when the accuracy on the training data was at least 98 %. The pruning process was continued until the network accuracy dropped below 95 %. The weights of the smallest network with at least 95% accuracy rate on the training data were saved for rules extraction. The second set of experiments were similar as the first set, except that we started with 50 networks each having 5 hidden units. The results of the experiments are summarized in Table 1. We list in this table the average number of connections in

the pruned networks and their average accuracy rates on the training and testing data sets. Figures in parenthesis indicate their standard deviations.

The effectiveness of the pruning algorithm is demonstrated by the fact that there is no significant difference in the number of connections left after pruning and in the accuracy of the pruned networks on the test data set between networks with original number of hidden units equal to three or five. Note also the significant decrease in the number of connections. A fully connected network with 3 hidden units has 279 connections, while a network with 5 hidden units has 465 connections. The small number of connections of the pruned network enables us to extract rules from the network. We will describe how this can be done in the next section.

4 Extracting rules from a pruned network

Previously reported algorithm for extracting rules [18] assume that the hidden units in the networks are either maximally active or inactive. That is, they have values that are either very close to one, or very close to minus one (or zero if the sigmoid function is used). In contrast, our algorithm requires no such assumption.

Regardless of the magnitude of the weights of the network, the activation value at each hidden unit is in the interval $(-1, 1)$. While the activation values can be anywhere in this interval, it is possible to clustered them and use their discretized values without causing any deterioration in the accuracy of the network.

For ease of notation, given an input pattern x^i let us define its activation value at hidden unit j

$$\alpha^j = \delta\left((x^i)^T w^j\right),$$

for all $j = 1, 2, \ldots h$. Let $\overline{\alpha}^j$ be its discretized value such that

$$\overline{\alpha}^j = \alpha^j + \Delta^j.$$

Theorem 1 below shows that if Δ^j is sufficiently small, then the network accuracy will be preserved.

Theorem 1 *Let α^j be the activation value of input pattern x^i at hidden unit j. Suppose that x^i has been correctly classified by the network such that condition (9) is satisfied. If the activation values $\alpha^j, j = 1, 2, \ldots h$ are replaced by their corresponding discretized values $\overline{\alpha}^j$ such that*

$$\overline{\alpha}^j = \alpha^j + \Delta^j,$$

where for some $\eta_2 > 0$ and Δ^j satisfies the condition

$$|\Delta^j| \leq 4\eta_2/(h \max_p |v_p^j|), \tag{13}$$

then the error of the new predicted output $\overline{S}_p^i$ of the network for x^i is bounded by

$$\max_p |\overline{e}_p^i| = \max_p |\overline{S}_p^i - t_p^i| \leq \eta_1 + \eta_2.$$

Proof:

We have

$$\begin{aligned}
\max_p |\overline{e}_p^i| &= \max_p |\overline{S}_p^i - t_p^i| \\
&\leq \max_p \left\{ |\overline{S}_p^i - S_p^i| + |S_p^i - t_p^i| \right\} \\
&\leq \max_p \left\{ \left| \sigma\left(\sum_{j=1}^{h} (\alpha^j + \Delta^j) v_p^j \right) - \sigma\left(\sum_{j=1}^{h} \alpha^j v_p^j \right) \right| \right\} + \max_p |S_p^i - t_p^i| \\
&\leq \eta_1 + \max_p \left\{ \left| \sigma\left(\sum_{j=1}^{h} \alpha^j v_p^j \right) \left(1 - \sigma\left(\sum_{j=1}^{h} \alpha^j v_p^j \right) \right) \times \lambda \sum_{j=1}^{h} \Delta^j v_p^j \right| \right\}, \quad \lambda \in (0,1) \\
&\leq \eta_1 + \frac{1}{4} \sum_{j=1}^{h} |\Delta^j| \max_p |v_p^j| \\
&\leq \eta_1 + \eta_2.
\end{aligned}$$

The third inequality above follows from the mean value theorem, while the last inequality follows from the definition of Δ^j. Q.E.D.

Hence, as long as $\eta_1 + \eta_2 < 0.5$, pattern x^i will still be correctly classified by the network with discretized activation values. If we define

$$\tau^j = 4\eta_2/(h \max_p |v_p^j|),$$

divide the interval $[-1, 1]$ into N equal subintervals with length τ^j, and place the discretized values $\overline{\alpha}_1^j, \overline{\alpha}_2^j, \ldots, \overline{\alpha}_{\lceil N/2 \rceil}^j$ such that

$$\overline{\alpha}_m^j = -1 + (2m-1)\tau^j \quad m = 1, 2, \ldots \lceil N/2 \rceil,$$

then condition (13) of the theorem will be satisfied.

The complexity of the rules extracted from the pruned network is proportional to the number of discrete activation values at the hidden units. The result of Theorem 1 shows that the number of discrete activation values is in turn proportional to the number of hidden units left in the pruned network and the magnitude of the connections between the hidden units and the output units. It is therefore important that the network is trained such that the use of large weights in the network is discouraged and that the network is pruned such that as many redundant hidden units as possible are removed.

The number of discrete activation values $\lceil N/2 \rceil$ can be large. In may also be unwise to place the discrete values equally spaced in $[-1, 1]$, since it does not take into consideration the distribution of the input patterns. In practice, we find that only a small number of discrete activation values are needed. We have developed a heuristic clustering algorithm for discretizing the activation values of a hidden unit. The algorithm place candidates for discrete values such that the distance between them is at least ϵ. A very small ϵ will always guarantee that the network with discrete activation values will have the same accuracy as the original network with continuous activation values. The algorithm can then be run again with a larger value of ϵ to reduce the number of clusters. The complete algorithm is described below.

Hidden unit activation values clustering algorithm

1. Let $\epsilon \in (0,1)$. Let D be the number of discrete activation values in the hidden unit. Let α_1 be the activation value in the hidden unit for the first pattern in the training set. Let $H(1) = \alpha_1, count(1) = 1, sum(1) = \alpha_1$ and set $D = 1$.

2. For all patterns $i = 2, 3, \ldots k$ in the training set:

   - Let α be its activation value.
   - If there exists an index $\overline{j}$ such that

$$\begin{aligned} |\alpha - H(\overline{j})| &= \min_{j \in \{1,2,\ldots,D\}} |\alpha - H(j)| \text{ and} \\ |\alpha - H(\overline{j})| &\leq \epsilon, \end{aligned}$$

     then set $count(\overline{j}) := count(\overline{j}) + 1, sum(D) := sum(D) + \alpha$

     else $D = D + 1, H(D) = \alpha, count(D) = 1, sum(D) = \alpha$.

3. Replace H by the average of all activation values that have been clustered into this cluster:

$$H(j) := sum(j)/count(j),\ j = 1, 2 \ldots, D.$$

Once the discrete values of all hidden units have been obtained, the accuracy of the network is checked again with the activation values at the hidden units replaced by one of the discrete values. An activation value α is replaced by $H(\overline{j})$, where index $\overline{j}$ is chosen such that $\overline{j} = \text{argmin}_j|\alpha - H(j)|$. If the accuracy of the network falls below the required accuracy, then ϵ must be decreased and the algorithm is run again. For a sufficiently small ϵ, it is always possible to maintain the accuracy of the network with continuous activation values, although the resulting number of different discrete activations can be impractically large.

We shall use several pruned networks for the breast cancer diagnosis problem to illustrate how this clustering algorithm works and how rules can be extracted from networks with discrete hidden unit activation values. After pruning 100 fully connected networks having three and five hidden units, networks with different topology and different set of connections are obtained. It may seem to be disheartening to find so many different pruned networks for the same set of training data. We must note however, that since the number of patterns used for training is a mere fraction of the number of all possible input patterns, there exist many rules that can describe the training patterns with a high degree of accuracy. Some of the rules that can be extracted from the pruned networks are given in the following 3 examples.

Example 1.

The smallest of the pruned network with more than 95 % accuracy rate on the training data has only 1 hidden unit and 5 connections. The network is depicted in Figure 1. The accuracy of this network on the training data and testing data are 96.86 % and 93.98 %, respectively. In this example, only input $\mathcal{I}_{16}$ and $\mathcal{I}_{55}$ are important. Only two discrete values are needed to maintain the accuracy of the network. The values found by the clustering algorithm are -0.90 and 1.00. Of the 350 training data, 234 patterns have the first value and 116 the second value. Since the connection from the hidden unit to the first output unit is 3.7 and to the second output unit is -3.7, all patterns with activation value equals to -0.90 will have an output value $S^i = \{0.03, 0.97\}$ and those

with activation value equals to 1 will have an output value $S^i = \{0.98, 0.02\}$. Hence, 234 patterns will be predicted as benign and the rest malignant.

Given the two discrete activation values at the hidden unit, it is a simple task to extract the rule that describes how each of these values are obtained since they are determined by only two attributes. We print inputs 16 and 55 of all training patterns and their corresponding predicted output. After removing all duplicates, only 4 unique patterns are left. A very simple rule can be immediately obtained from these 4 patterns: if $\mathcal{I}_{16} = 0$ and $\mathcal{I}_{55} = 0$, then activation value = -0.90 (ie. sample is benign), otherwise activation value = 1.0 (ie. sample is malignant). Our coding scheme for the input data allows us to easily get the rule in term of the original attributes:

Rule 1:

If $\mathcal{A}_2 \leq 4$ and $\mathcal{A}_6 \leq 5$, then benign.

Else malignant.

The accuracy of this rule is summarized in Table 2.

Example 2.

In this example we illustrate how rules can be extracted even when there are more than 1 hidden units left in the pruned network. We will use a pruned network with 3 hidden units and an average number of connections, ie. 10. This network is depicted in Figure 2.

The results of the hidden unit activation values clustering algorithm on this pruned network are as follows.

1. Hidden unit 1: there are 2 discrete values: 0 and -1. Of the 350 training data, 244 patterns have the first value and 106 patterns have the second value.

2. Hidden unit 2: there are 2 discrete values: -1 and 0.45. The distribution of the training data is 268 and 82, respectively.

3. Hidden unit 3: there are 2 discrete values: 0 and 1. Of the 350 training data, 320 patterns have the first value and 30 patterns have the second value.

Since there are 2 discrete activation values at each of the three hidden units, we have a total of 8 possible outcomes at the output units. The 8 possible outputs are summarized in the Table 3.

From the entries in this table, it is clear that in order to obtain rules that classify a pattern to be a benign sample, we only need to check under what conditions will the first hidden unit have activation value equals to 0, the second equals to -1 and the third equals to 0. The first hidden unit is connected only to one input, namely input $\mathcal{I}_{67}$. Only when $\mathcal{I}_{67} = 0$, the activation value will be 0. Similarly for the third hidden unit, the activation value will also be zero if $\mathcal{I}_{71} = 0$. There are two inputs connected to the second hidden unit, the two inputs are $\mathcal{I}_4$ and $\mathcal{I}_{70}$. However, as we have noted in the previous section, input $\mathcal{I}_{70}$ will always have value equals to 1. This correspond to the

fact that attribute $\mathcal{A}_8$ has value that is greater than or equal to 1 for all input patterns. Input $\mathcal{I}_4$ having a zero value gives the second hidden unit an activation value that is equal to -1. Therefore, the rule for a pattern to be classified as a benign sample is as follows: if $\mathcal{I}_4 = 0$ and $\mathcal{I}_{67} = 0$ and $\mathcal{I}_{71} = 0$. In term of the original attributes, we have this rule:

Rule 2:

If $\mathcal{A}_1 \leq 6$ and $\mathcal{A}_7 \leq 3$ and $\mathcal{A}_8 \leq 9$, then benign.

Else malignant.

The accuracy of this rule is summarized in Table 4.

Example 3.

In this example, we will extract rules from the pruned network that has the highest accuracy on the testing set among the 100 pruned networks. This network is depicted in Figure 3. Its accuracy rates on the training data and the testing data are 97.71 % and 96.56 %, respectively.

Applying the hidden unit activation values clustering algorithm with $\epsilon = 0.4$ allows us to maintain these rates. Four discrete activation values are obtained: -1, 1, -0.33, and 0.24. There is only one hidden unit left. The weights of the connections between the hidden unit and the first and the second output unit are 5.75 and -5.75, respectively. It follows that the output of all patterns with activation value equals to -1 is $S^i = \{0, 1\}$ and the output of all patterns with activation value equals to -0.33 is $S^i = \{0.13, 0.87\}$. Hence, all these patterns will be predicted as benign samples. In order to determine which patterns have either one of these two activation values, we print inputs $\mathcal{I}_4, \mathcal{I}_{13}, \mathcal{I}_{28}, \mathcal{I}_{58}$ and $\mathcal{I}_{72}$ of all training patterns. After removing all duplicates, only two patterns with activation value equals to -1 are found. These are $\{0, 0, 0, 0, 0\}$ and $\{0, 0, 1, 0, 0\}$. There

are also only two patterns with activation values equal to -0.33, they are $\{0, 0, 0, 1, 0\}$ and $\{1, 0, 0, 0, 0\}$. Hence, the following rules can be deduced: if $\mathcal{I}_4 = \mathcal{I}_{13} = \mathcal{I}_{58} = \mathcal{I}_{72} = 0$ or if $\mathcal{I}_4 = \mathcal{I}_{13} = \mathcal{I}_{28} = \mathcal{I}_{72} = 0$ or if $\mathcal{I}_{13} = \mathcal{I}_{28} = \mathcal{I}_{58} = \mathcal{I}_{72} = 0$ then predict the input as benign sample, otherwise predict the input as malignant sample. In term of the original attributes, we have the following rules:

Rule 3:

If $\mathcal{A}_1 \leq 6$ and $\mathcal{A}_2 \leq 7$ and $\mathcal{A}_3 \leq 2$ and $\mathcal{A}_8 \leq 8$, then benign.

Else if $\mathcal{A}_1 \leq 6$ and $\mathcal{A}_2 \leq 7$ and $\mathcal{A}_6 \leq 2$ and $\mathcal{A}_8 \leq 8$, then benign.

Else if $\mathcal{A}_2 \leq 7$ and $\mathcal{A}_3 \leq 2$ and $\mathcal{A}_6 \leq 2$ and $\mathcal{A}_8 \leq 8$, then benign.

Else malignant.

The accuracy rates on the training data and testing data are summarized in Table 5.

5 Conclusion

We have described how rules can be extracted from pruned neural networks for breast cancer diagnosis. Two factors enable us to extract simple rules than classify a sample as either benign or malignant with a high degree of accuracy. The first factor is a very effective neural network pruning algorithm. Using our new penalty function, we have been able to prune networks such that only very few input units, hidden units and connections left in the networks. The second factor is an algorithm that clusters hidden unit activation values of a pruned network. This algorithm allows us to consider only a small number of different hidden unit activation values and still maintain the accuracy of the original network. For the Wisconsin Breast Cancer Diagnosis problem, we have been able to extract simple rules that achieve more than 95 % accuracy rates on both the training data and the testing data.

References

[1] 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.

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

[3] 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.

[4] 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.

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

[6] F.L. Chung and L. Lee, A node pruning algorithm for backpropagation network, *Int. Journal of Neural Systems* 3 (3) (1992) 301-314.

[7] M. Hagiwara, A simple and effective method for removal of hidden units and weights, *Neurocomputing* 6 (1994) 207-218.

[8] E.D. Karnin, A simple procedure for pruning back-propagation trained neural networks, *IEEE Trans. on Neural Networks* 1 (2) (1990) 239-242.

[9] O.L. Mangasarian, R. Setiono and W.H. Wolberg, Pattern recognition via linear programming: theory and application to medical diagnosis, in: T.F. Coleman and Y. Li. eds., *Large-scale Numerical Optimization,* (SIAM, Philadelphia, 1990) 22-30.

[10] M.C. Mozer and P. Smolensky, Skeletonization: a technique for trimming the fat from a network via relevance assessment, in: D.S. Touretzky, ed., *Advances in Neural Information Processing Systems I*, (Morgan Kaufmann, San Mateo, 1989) 107-115.

[11] 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.

[12] P.M. Murphy and D.W. Aha, UCI repository of machine learning databases. Machine-readable data repository. Irvine, CA: University of California, Department of Information and Computer Science, 1992.

[13] A. van Ooyen and B. Nienhuis, Improving the convergence of the backpropagation algorithm, *Neural Networks* 5 (1992) 465-471.

[14] D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing.* (MIT Press, Cambridge, MA, 1986).

[15] R. Setiono, A penalty function approach for pruning feedforward neural network, submitted to *Int. Journal of Neural Systems.*

[16] R. Setiono and L.C.K. Hui, Use of quasi-Newton method in a feedforward neural network construction algorithm, *IEEE Trans. on Neural Networks* 6 (1) (1995) 273-277.

[17] H.H. Thodberg, Improving generalization of neural networks through pruning, *Int. Journal of Neural Systems* 1 (4) (1991) 317-326.

[18] G.G. Towell and J.W. Shavlik, Extracting refined rules from knowledge-based neural networks, *Machine Learning* 13 (1) (1993) 71-101.

[19] R.L. Watrous, Learning algorithms for connectionist networks: applied gradient methods for nonlinear optimization, in: *Proc. of the IEEE First Internat. Conf. on Neural Networks. San Diego* (IEEE Press, New York, 1987) 619-627.

[20] W.H. Wolberg and O.L. Mangasarian, Multisurface method of pattern separation for medical diagnosis applied to breast cytology, *Proceedings of the National Academy of Sciences* 87 (1990) 9193-9196.

[21] W.H. Wolberg, M.A. Tanner and W.Y. Loh, Diagnostic schemes for fine needle aspirates of breast masses, *Analytical and Quantitative Cytology and Histology* 10 (1988) 225-228.

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

Table 1.

		95 %	**98 %**
h = 3	No. connections	10.22 (4.36)	16.54 (4.74)
	Accuracy on train data	96.88 (0.94) %	98.43 (0.30) %
	Accuracy on test data	92.76 (1.85) %	93.69 (1.46) %
h = 5	No. connections	10.70 (4.53)	17.42 (4.18)
	Accuracy on train data	96.72 (1.11) %	98.53 (0.29)%
	Accuracy on test data	92.70 (1.57) %	93.87 (1.16) %

Table 2.

	Training set	Testing set
Malignant	113/121	105/120
	93.39 %	87.50 %
Benign	226/229	223/229
	98.69 %	97.38 %
Overall	339/350	328/349
	96.86 %	93.98 %

Table 3.

Hidden unit activations			Predicted output		Classification
1	2	3	1	2	
0	-1	0	0.02	0.98	Benign
0	-1	1	0.78	0.22	Malignant
0	0.45	0	0.84	0.16	Malignant
0	0.45	1	1.00	0.00	Malignant
-1	-1	0	0.74	0.26	Malignant
-1	-1	1	1.00	0.00	Malignant
-1	0.45	0	1.00	0.00	Malignant
-1	0.45	1	1.00	0.00	Malignant

Table 4.

	Training set	Testing set
Malignant	118/121	109/120
	97.52 %	90.83 %
Benign	219/229	216/229
	95.63 %	94.32 %
Overall	337/350	325/349
	96.29 %	93.12 %

Table 5.

	Training set	Testing set
Malignant	119/121	116/120
	98.35 %	96.67 %
Benign	223/229	221/229
	97.38 %	96.51 %
Overall	342/350	337/349
	97.71 %	96.56 %

Figure 1.

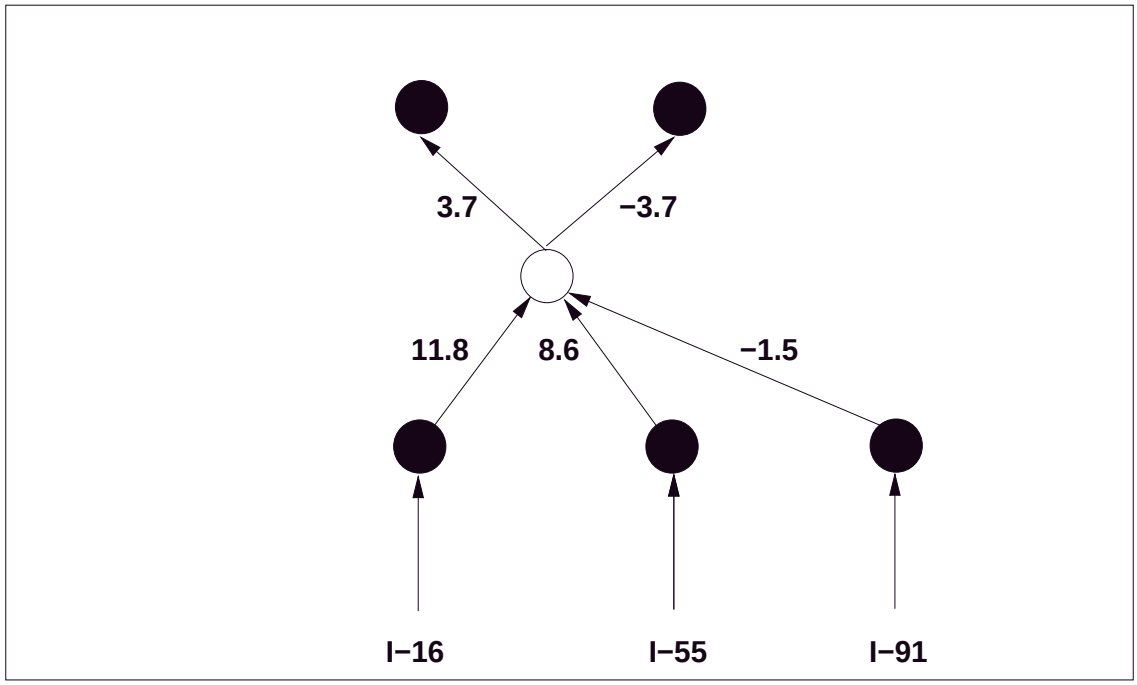


Figure 2.

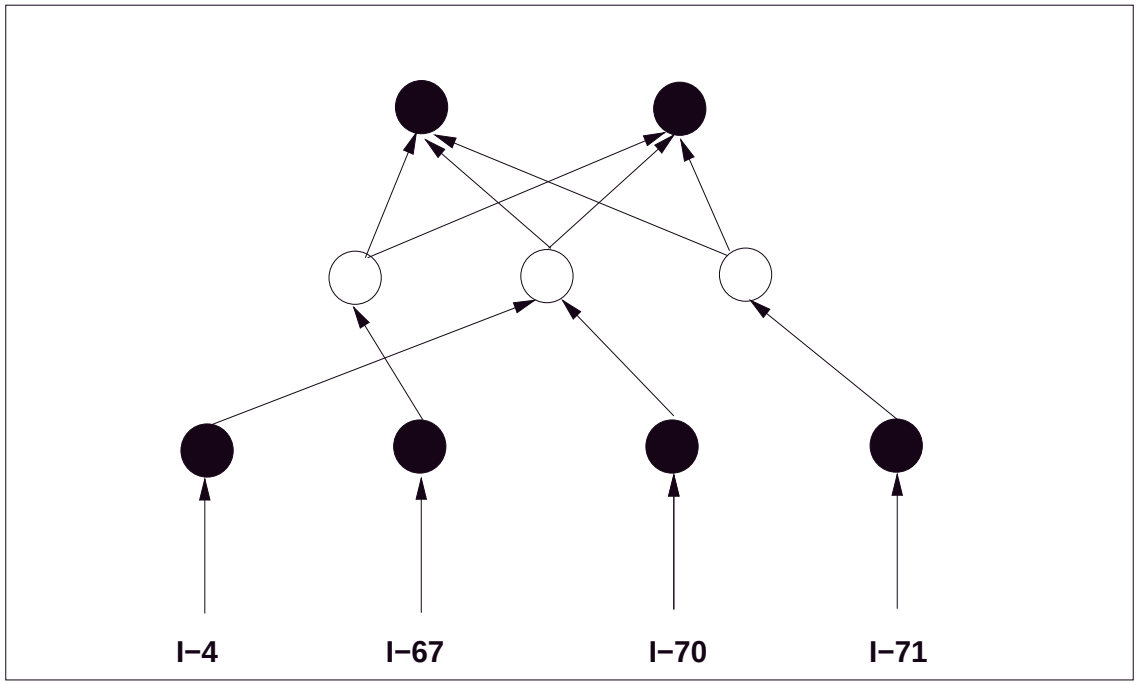
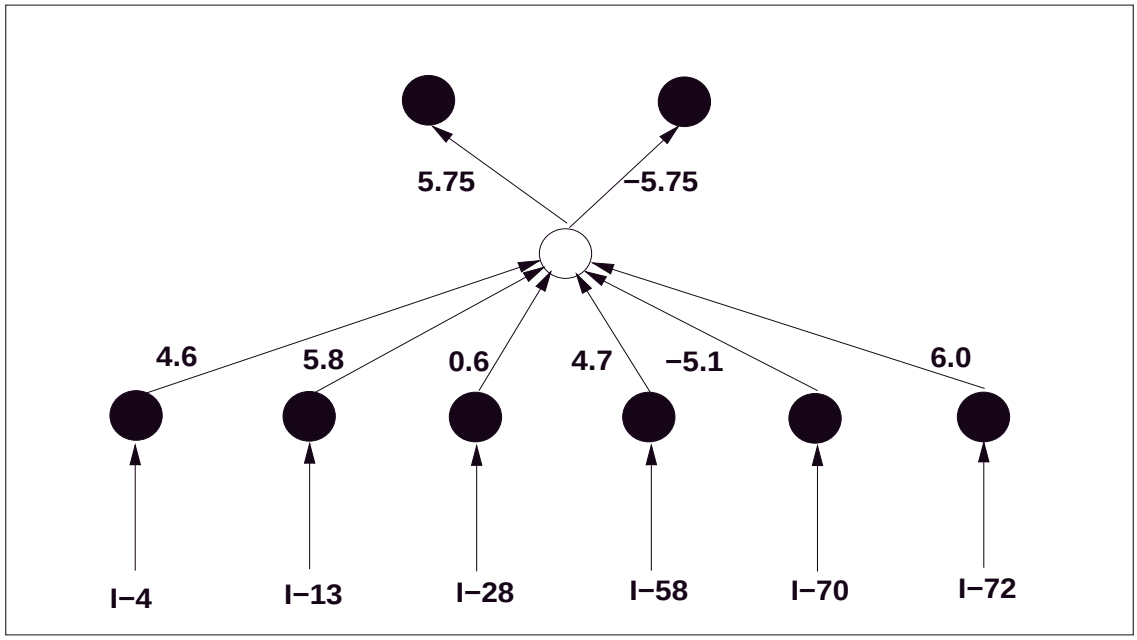


Figure 3.



List of table captions.

1. Table 1. Average number of connections and accuracy of 100 pruned networks on the training and testing data sets and their standard deviations.

2. Table 2. Accuracy of Rule 1 on the training and testing data.

3. Table 3. Predicted output of the pruned network in Figure 2.

4. Table 4. Accuracy of Rule 2 on the training and testing data.

5. Table 5. Accuracy of Rule 3 on the training and testing data.

List of figure captions.

1. Figure 1. Pruned network with only 5 connections, accuracy on training set = 96.86 %, accuracy on the testing set = 93.98 %.

2. Figure 2. Pruned network with 10 connections, accuracy on training set = 96.29 %, accuracy on the testing set = 93.12 %. Weights of the connections are not shown to reduce cluttering.

3. Figure 3. Pruned network with an accuracy rate on training set = 97.71 % and an accuracy rate on the testing set = 96.56 %.