Automated Optimization of Neural Networks in Estimating Medical Outcomes

Monique Frize¹² (Monique_Frize@carleton.ca, frize@site.uottawa.ca),
Colleen M. Ennett¹ (ennett@canada.com),
Elaine Charette² (echarette@canada.com)

¹Systems and Computer Engineering, 1125 Colonel By Drive, Carleton University, Ottawa, ON, K1S 5B6
²School of Information Technology and Engineering, 161 Louis Pasteur, University of Ottawa, Ottawa, ON, K1N 6N5

Abstract

Our artificial neural network (ANN) software requires nine parameters to be initialized when running an experiment. Tuning each parameter for optimum ANN performance, one at a time, is very time-consuming since a user must adjust each or a combination of these parameters to obtain optimal results. The objective of this work was to develop a program that automatically optimizes all of these parameters without user-supervision. The problem was approached using a “divide and conquer” technique. The ANN results obtained with the new automated network were compared with results obtained previously with the manual method. In addition, a new stopping criterion where the network monitors its own performance to choose when to stop training was introduced. The accuracy of the new ANN was similar to the previously manually-optimized networks. The network parameters’ sensitivity curves, in determining the highest correct classification rate (best accuracy), show that the momentum, learning rate, learning rate increment, and the error ratio were the most sensitive parameters; the weight-decay constant and the learning rate decrement were least sensitive on network performance. The improvements in the experimental approach allow our future experiments to be run around the clock, on several computers simultaneously, and without user-supervision.

Keywords: Artificial neural networks, stopping criterion, sensitivity, outcome prediction, optimization, automatization.

BACKGROUND

In previous work, the research group used a three-layer feed-forward backpropagation artificial neural network (ANN), employing the hyperbolic tangent transfer function, to estimate a number of medical outcomes. A three-layer network refers to the input layer, a layer of hidden nodes and an output layer. To date, the outcomes investigated by the research group are: Estimating in-hospital mortality, length of stay, and duration of artificial ventilation for adult intensive care unit (ICU) patients [1,2], mortality risk stratification for coronary artery bypass grafting surgery patients [3,4], and a variety of outcomes with neonatal ICU patients [5,6].

Quantifying the effectiveness of ANNs as pattern classifiers for estimating various medical outcomes has been the focus of much research. The behaviour of a neural network depends on the value of the network weights, of the network parameters, and the type of transfer function used [7,8]. Several attempts have been made to improve the generalization ability of feed-forward backpropagating ANNs. One technique called weight-elimination simplifies the complexity of the ANN by adding a penalty term to the standard backpropagation cost function, forcing already small weights to zero [9,10]. Previous research done by the team has shown this technique to enhance the performance of the ANNs for the databases and outcomes listed above.

METHODOLOGY

The new series of experiments were performed on a three-layer feed-forward ANN with two nodes in the hidden layer, and was trained using the backpropagation algorithm with weight-elimination added to the sum of squared errors cost function. As in previous work, the transfer function used in this network was the hyperbolic tangent.

The ANN was designed to classify duration of ventilation into two classes for an adult ICU database of postoperative patient case histories compiled at the Doctor Everett Chalmers Hospital (DECH) in Fredericton, NB, Canada. The database contained 883 surgical patients admitted to the ICU between January 1, 1991 and September 29, 1993, and was randomly separated with two-thirds going into the training set (589

cases) and the remainder becoming the test set (294 cases). The outcome class investigated for this study was whether patients required greater than eight hours of artificial ventilation, or less than or equal to eight hours of ventilation. The prevalence of patients requiring greater than eight hours of artificial ventilation was 71.1 percent for the training set, and 72.3 percent for the test set. This study did not focus on the clinical relevance of the outcome, but rather on the efficiency of the data modelling procedure employed.

Implemented using MATLAB Version 5.3, our ANN needs to initialize nine parameters at the beginning of each neural network execution. In our model, the parameters that required adjustments were: Learning rate, adaptive learning rate scale factors, weight-decay constant, adaptive weight-decay factors, weight-elimination scale factor, momentum and maximum error difference. These parameters are defined in Appendix 1.

In the past, each of these variables was adjusted manually, little by little, until an optimal response was obtained. The optimal response can be set either as a minimum overall error rate on the test set or as a maximum accuracy (the number of cases that were correctly classified out of all of the cases presented to the network), also on the test set.

When running an ANN experiment, a graph shows the classification rate of both the training and test sets. Thus, for every value of each of these parameters, the user must determine the best classification rate (highest accuracy) of the training and of the test sets from these graphs. The overall performance of an ANN depends on the values assigned to these network parameters at the beginning of each execution [11]. Since most of these parameters have a large range of possible values, finding the best combination of parameter values is time-consuming since every step requires user supervision to achieve success [2,3,12]. In fact, running a set of experiments to classify one outcome with a single database could take days or even weeks.

**Automation of the Neural Network**

A program was developed that can automatically optimize each network parameter used for estimating medical outcomes without user-supervision [12]. When developing the model, two approaches came to mind: the first, called “brute force,” meant periodically initializing the ANN using every possible combinations of values of each parameter. This method was judged to have impossibly long compilation time. For nine parameters, each with 200 possible values, and a single execution running time of three minutes, it would take $2.9 \times 10^{15}$ years to run. A much more efficient approach called “divide and conquer algorithm” was selected.

In the previous manual experiments, “optimal” values of the nine parameters were identified for the database under study here, called “default values”. Therefore, to begin the automatization process one parameter at a time, the other eight parameters were set to default values and the remaining parameter was processed with the “divide and conquer” approach. The process was repeated for each of the nine parameters and graphs were recorded for the average squared error and the classification accuracy for the outcome class selected and for the particular set of experiments.

The divide and conquer algorithm uses the minimum, maximum and midpoint values of a parameter to calculate the test set classification rate. Then, the parameter values generating the two best test set classification rates and their midpoint are used again in a recursive function until two equal classification rates are found or until no further parameter values are possible. Thus, the search space is recursively divided in half to converge onto the best values. The parameter intervals used were similar to those used previously in the manual approach.

**Stopping Criterion**

In previous experiments done by the research group, the ANN executed 3000 training epochs (where an epoch refers to a single presentation of the entire training set to the network), after which the user determined the best accuracy and at what specific epoch this was attained by examining the graphed results of accuracy and error rate [2]. In these experiments, a new stopping criterion was used to shorten the running time; this was based on the network’s performance [12]. The network monitors the accuracy and/or the error rate (depending on whether the desire is to maximize the number of correctly classified cases or to minimize the average squared error of the network) of the ANN on the training and test sets, and if the training or test set accuracy does not improve, or the
error rate stops decreasing in the following 500 epochs (a value set arbitrarily) training stops. This approach gives the ANN the decisive power to stop training after permitting a reasonable number of epochs to monitor the network’s performance and observe any improvements.

RESULTS AND DISCUSSION

Table 1 shows the initial network parameter values as selected manually in earlier work [2] and those specified by the automatization process [12]. Recall that each parameter value for the automated network was determined independently while the remaining eight parameters were set to their default values (i.e., the manually-derived values). From Table 1, we can see that most of the optimal parameter values found using the automated approach are similar to the manually-derived values. This confirms the use of the manually-chosen parameters as the default values in the automated network. In fact, only two parameters are noticeably different from their manual counterparts: weight-elimination scale factor (w_o) and momentum.

For the manual approach, the best accuracy for the test dataset attained by the ANN initialized with the parameter values found manually was 91.8 percent at 2071 epochs. When the new stopping criterion was employed and the network was initialized with the manual parameter values, an accuracy of 91.5 percent was achieved in only 860 epochs.

Figure 1 shows the graphs of the correct classification rate (accuracy) for the manual ANN and the automated network with the new stopping criterion. When the automated network was executed with the new stopping criterion and the optimal value-selection technique, the best accuracy obtained was also 91.8 percent, but with a much shorter execution time of 606 epochs. This performance was attained for the case of optimizing the weight-decay factor while the other parameters were set to their default values. Running the ANN automatically provided simulations using a larger number of possible parameter values, thus permitting improved outcomes even with the use of the new earlier stopping criterion.

Table 1: The network parameter interval values and selected initial optimal values found manually and those found by the automatization process.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Parameter interval values used</td>
<td>Optimal parameter value obtained</td>
</tr>
<tr>
<td>lr</td>
<td>[0.000005, 0.005]</td>
<td>0.001</td>
</tr>
<tr>
<td>lr_inc</td>
<td>[1, 1.005]</td>
<td>1</td>
</tr>
<tr>
<td>lr_dec</td>
<td>[1, 0.997]</td>
<td>1</td>
</tr>
<tr>
<td>λ</td>
<td>[0, 0.000001] and [0.000001, 0.0008]</td>
<td>0 to start 0.0003 after 500 epochs</td>
</tr>
<tr>
<td>λ_inc</td>
<td>[1, 1.005]</td>
<td>1</td>
</tr>
<tr>
<td>λ_dec</td>
<td>[1, 0.997]</td>
<td>1</td>
</tr>
<tr>
<td>w_o</td>
<td>[0.001, 1]</td>
<td>0.1</td>
</tr>
<tr>
<td>momentum</td>
<td>[0, 0.95]</td>
<td>0.975</td>
</tr>
<tr>
<td>[0.95, 0.99]</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>err_ratio</td>
<td>[1.001, 1.04]</td>
<td>1.01</td>
</tr>
</tbody>
</table>

1 [0, 0.000001] to start, then [0.000001, 0.0008] after 500 epochs.
2 Emphasis was focussed on [0.95, 0.99].
3 Values were chosen with up to six decimal places as in 1.

![Graph](image1)

**Figure 1:** (a) The accuracy curve for the best classification rate obtained using the manual ANN; (b) The accuracy curve for the best classification rate using the automated ANN with the new stopping criterion.

**Table 2:** The maximum accuracy obtained for each independently automatized parameter value. The full execution of this automatized ANN took 97 minutes with a data set containing 883 patients (train set: 589; test set: 294).*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Accuracy (%)</th>
<th>No. of epochs to find max test set accuracy</th>
<th>Time of execution (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training set</td>
<td>Test set</td>
<td></td>
</tr>
<tr>
<td>lr</td>
<td>92</td>
<td>91.5</td>
<td>871</td>
</tr>
<tr>
<td>lr_inc</td>
<td>91.7</td>
<td>91.5</td>
<td>860</td>
</tr>
<tr>
<td>lr_dec</td>
<td>91.7</td>
<td>91.5</td>
<td>860</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>91.7</td>
<td>91.5</td>
<td>875</td>
</tr>
<tr>
<td>(\lambda_{inc})</td>
<td>91.7</td>
<td>91.5</td>
<td>860</td>
</tr>
<tr>
<td>(\lambda_{dec})</td>
<td>91.7</td>
<td>91.5</td>
<td>860</td>
</tr>
<tr>
<td>(w_o)</td>
<td>91.7</td>
<td>91.8</td>
<td>606</td>
</tr>
<tr>
<td>momentum</td>
<td>93.2</td>
<td>91.2</td>
<td>711</td>
</tr>
<tr>
<td>err_ratio</td>
<td>91.5</td>
<td>91.5</td>
<td>907</td>
</tr>
</tbody>
</table>

**Total Time:** 97

* Manual experiment values: Training accuracy = 91.2%; Test accuracy = 91.8%; Epoch = 2071; Time = 10 min (just running the ANN once, not including weeks of tweaking).

When one considers the time it took to optimize the network for each individual parameter, the total time of execution was 97 minutes, as shown in Table 2. Recall that these experiments did not require user-supervision, and therefore can be seen as a dramatic improvement in the time required to optimize the ANN compared to the weeks of user-dependent simulations necessary to obtain the results manually. Although the individual automation has a total running time of 97 minutes, if we attempted to apply the system to the nine combinations simultaneously, it would take 97³ minutes to execute, which is still too large a task to complete at this time.

**Sensitivity of the Network Parameters**

To further improve this automation technique,
knowledge of how sensitive the network’s performance is to changes in the value of each network parameter was paramount. Several of the nine parameters in our model consistently allowed the ANN to attain a high accuracy and a low error rate regardless of the changes in the initial values. Table 1 shows that the optimal values of the learning rate increment and decrement values, as well as the weight-decay increment and decrement values are 1. This means that they have no effect on the network because these values are simply multiplied by the learning rate and weight-decay constant, respectively. Their lack of impact is also evident from the performance evaluations in Table 2 since the accuracy is not improved over the manual experiments with the new stopping criterion. This means that, for the current database under study, we could set these network parameters to constant values and focus on automatically tweaking the more sensitive parameters.

To demonstrate this sensitivity, the optimal classification rate obtained for each parameter within a range of values was plotted. This information indicated, for each parameter, the best range of values that could be used to improve the automated ANN’s outcomes. This allows one to tweak more than one parameter at a time and to initialize the insensitive values as constants for each new database and outcome classification problem. For the particular database and outcome studied here, the most sensitive parameters were: Momentum, learning rate, learning rate increment, and the error ratio. Although learning rate increment was also found to have no effect on the network’s performance (because its optimal value was 1), it was highly sensitive to changes in its value. The changes, however, resulted in worse performance than the value of 1. Due to the sensitivity of this parameter, learning rate increment should not be set as a constant for this database; instead, attempts to optimize it like the other network parameters should be made. Those with fairly low sensitivity to change were the weight-decay constant and learning rate decrement value.

CONCLUSION

These experiments demonstrated the application of a feed-forward backpropagation artificial neural network with two nodes in the hidden layer to the problem of classifying ICU patients as to whether or not they will require more than eight hours of artificial ventilation. A new stopping criterion was presented, as well as the results achieved by the automated ANN that requires no user-supervision. Also, sensitivity graphs of the adjustable network parameters identified which parameters were most sensitive and which were least sensitive for this adult ICU database.

The new stopping criterion that stops training after 500 epochs after no performance improvement can be seen reduced the required network training time without a dramatic effect on the ANN’s accuracy. This fact is demonstrated by a accuracy of 91.5 percent which was only 0.3 percent lower than the accuracy of the manually-stopped network, and a reduction of more than 1200 epochs using the new criterion.

The automated ANN reduced the execution time required to optimize the network from several weeks of user-supervised-time to just 97 minutes of unsupervised-time while still achieving the same accuracy as the manually-optimized network. The manually-derived parameter values were confirmed as optimal by the automated network. Although the number of combined parameters that can be automated using this algorithm has not yet been determined, the results show that the “divide and conquer” approach is efficient and future work will attempt to automate the selection of optimal values for a combination of parameters.

The sensitivity graphs indicated that momentum, learning rate, learning rate increment and the error ratio were highly sensitive to changes in their parameter values. The weight-decay constant and the learning rate decrement showed low sensitivity to change. The sensitivity graphs can provide a good idea of the range of the best possible values for each network parameter as a starting point when working with new databases.

Finally, the ability to run several ANN experiments around the clock will substantially enhance the progress expected for the analysis of a variety of medical databases.

APPENDIX 1

The Network Parameters

Learning rate (lr): The value of the learning rate determines the speed at which the network attains a minimum in the criterion function so long as it is small
enough to insure convergence [10]. If the learning rate is too high, it may oscillate around the global minimum, and is unable to converge.

**Learning rate increment (lr_inc):** The learning rate’s incremental value.

**Learning rate decrement (lr_dec):** The learning rate’s decrement value.

**Weight-decay constant (\(\lambda\))**: The weight-elimination constant determines how strongly the weights are penalized [9,10].

**Weight-decay constant increment (\(\lambda_{\text{inc}}\))**: The weight-decay constant’s incremental value.

**Weight-decay constant decrement (\(\lambda_{\text{dec}}\))**: The weight-decay constant’s decrement value.

**Weight-elimination scale factor (\(w_o\))**: Weight-elimination scale factor defines the sizes of “large” and “small” weights. When \(w_o\) is small, the small weights will be forced to zero resulting in fewer large weights (i.e., weight-elimination). A large \(w_o\) causes many small weights to remain and limits the size of large weights (i.e., weight-decay) [9,10].

**Momentum (momentum)**: The momentum parameter adds a proportion of the previous weight-change value to the new value, thereby giving the algorithm some “momentum” to prevent it from getting caught in local minima [11].

**Error ratio (err_ratio)**: The error ratio controls how the backpropagation makes adaptive changes in the learning rate, the weight-decay constant, and the momentum term.

**REFERENCES**


