J Clin Epidemiol Vol. 49, No. 11, pp. 1225–1231, 1996
Copyright © 1996 Elsevier Science, Inc.

ELSEVIER

0895-4356/96/$15.00
PII S0895-4356(96)00002-9

Advantages and Disadvantages of Using Artificial Neural Networks versus Logistic Regression for Predicting Medical Outcomes

Jack V. Tu

INSTITUTE FOR CLINICAL EVALUATIVE SCIENCES, NORTH YORK, ONTARIO, CANADA, DEPARTMENT OF MEDICINE, SUNNYBROOK HEALTH SCIENCE CENTRE, UNIVERSITY OF TORONTO, NORTH YORK, ONTARIO, CANADA, AND DIVISION OF HEALTH POLICY RESEARCH AND EDUCATION, HARVARD UNIVERSITY, BOSTON, MASSACHUSETTS

ABSTRACT. Artificial neural networks are algorithms that can be used to perform nonlinear statistical modeling and provide a new alternative to logistic regression, the most commonly used method for developing predictive models for dichotomous outcomes in medicine. Neural networks offer a number of advantages, including requiring less formal statistical training, ability to implicitly detect complex nonlinear relationships between dependent and independent variables, ability to detect all possible interactions between predictor variables, and the availability of multiple training algorithms. Disadvantages include its "black box" nature, greater computational burden, proneness to overfitting, and the empirical nature of model development. An overview of the features of neural networks and logistic regression is presented, and the advantages and disadvantages of using this modeling technique are discussed. J CLIN EPIDEMIOL 49;11:1225–1231, 1996.

KEY WORDS. Neural networks, logistic regression

INTRODUCTION

Clinical prediction rules can be developed using a number of techniques, including a variety of statistical methods (e.g., logistic and linear regression, discriminant analysis, and recursive partitioning [CART]), and the clinical judgment of experts [1,2]. For predicting dichotomous outcomes, logistic regression has emerged as the statistical technique of choice [3]. Artificial neural networks represent a newer technique that has emerged as a potential alternative to logistic regression analysis and other classical statistical techniques [4,5]. Neural networks are not constrained by a predefined mathematical relationship between dependent and independent variables, and have the ability to model any arbitrarily complex nonlinear relationship [6]. Developers of artificial neural network prediction models do not require formal training in statistical methodology, and models can be developed by users with a minimum of theoretical knowledge. A few studies suggest that they may offer significantly better predictive performance than traditional statistical methods for certain problems [7,8]. Thus, neural networks may represent an attractive alternative to logistic regression as a statistical modeling technique under certain circumstances.

Logistic Regression

Logistic regression is a widely used statistical modeling technique in which the probability of an outcome is related to a series of potential predictor variables by an equation of the form

$$\log [p/(1 - p)] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_i X_i \qquad (1)$$

where p is the probability of the outcome of interest, β_0 is an intercept term, $\beta_1, \ldots, \beta_i$ are the β coefficients associated with each variable, $X_1, \ldots, X_i$ are the values of the potential predictor variables, and i is a unique subscript denoting each variable [3]. The usual assumption is that these predictor variables are related in a linear manner to the log odds $\{\log[p/(1 - p)]\}$ of the outcome of interest. Variables are usually selected for inclusion in these models through some form of backward or forward stepwise regression technique, although stepwise variable selection techniques may be prone to problems [9]. Full model fits without deletion of insignificant variables may be preferred under certain circumstances [10]. Logistic regression models use as their convergence criterion the maximization of a likelihood function. β coefficients can easily be converted into the corresponding odds ratios by raising e to the coefficient if variables are represented by a single linear term or as dummy variables, and one can easily interpret the magnitude of importance of a predictor. In addition, widely accepted criteria such as the Hosmer–Lemeshow statistic have been developed for assessing the "goodness of fit" of these models [3]. Thus, logistic regression has become the technique of choice for statistical modeling in which the outcome of interest is dichotomous (e.g., mortality).

Neural Networks

Artificial neural networks are a new alternative to logistic regression, the statistical technique with which they share the most similarities. Neural networks are algorithms that are patterned after the structure of the human brain [5]. They contain a series of mathematical equations that are used to simulate biological processes such as learning and memory. Neural networks have been developed for diagnosing acute myocardial infarction, diagnosing acute pulmonary embolism, and predicting intensive care unit (ICU) resource utilization following cardiac surgery among other applications [7,11,12]. In a neural network, one has the same goal as in logistic regression modeling, predicting an outcome based on the values of some predictor variables. However, the approach used in developing the model is quite different. Although many different types of neural network

Address correspondence to: Dr. Jack V. Tu, Institute for Clinical Evaluative Sciences, G-106, 2075 Bayview Avenue, North York, Ontario, Canada, M4N 3M5.

Accepted for publication on 7 September 1995.

TABLE 1. Common terms in the field of neural networks and their equivalent in statistics

Neural networks	Statistics
Input	Independent (predictor) variable
Output	Dependent (outcome) variable, predicted value
Connection weights	Regression coefficients
Bias weight	Intercept parameter
Error	Residuals
Learning, training	Parameter estimation
Training case, pattern	Observation
Cross-entropy	Maximum likelihood estimation

training algorithms have been developed, the focus of this article is restricted to "back-propagation," the most popular technique. This article is also restricted to comparing neural networks with logistic regression, although neural networks can also be an alternative to linear regression, survival analysis, or time-series analysis [13].

Artificial neural networks were first developed several decades ago by researchers who were attempting to model the learning processes of the human brain [4,5]. However, it was only in the late 1980s with the rediscovery of the back-propagation training algorithm did widespread interest in this technique develop within the scientific community [14]. Neural networks have the ability to "learn" mathematical relationships between a series of input (independent, predictor) variables and the corresponding output (dependent, outcome) variables. This is achieved by "training" the network with a training (or derivation) data set consisting of predictor variables and the known or associated outcomes. Networks are programmed to adjust their internal weights based on the mathematical relationships identified between the inputs and outputs in a data set. Once a network has been trained, it can be used for pattern recognition or classification tasks in a separate test (or validation) data set.

Table 1 is a brief glossary showing some common terms in the field of neural networks and their equivalent in statistics. To illustrate some of the similarities and differences between neural networks and logistic regression, an example of both techniques is presented. Figure 1 is a diagram illustrating an artificial neural network that has been trained to predict the probability of a patient dying from a hypothetical disease based on only two predictor variables: the age of the patient ($x1$; range, 0 to 100) and the sex of the patient ($x2$; male = 0, female = 1). Neural network models are often represented using such diagrams. The circles in these diagrams are known as *nodes* (or units) while the lines connecting different nodes are known as *connection weights*. A typical neural network consists of a series of nodes that are arranged in three layers (input, hidden, output). The input nodes are where the values of the predictor variables (e.g., $x1$, $x2$) are presented to the network while the output node(s) represents the predicted output(s) of the network. Neural networks can have multiple outputs although networks with one output node only are considered in this article to maintain the parallel to logistic regression. The nodes in the hidden layer contain intermediate values that are calculated by the network. Each of the hidden and output nodes contains a function termed the *activation function* in neural network terminology [15,16]. The hidden nodes allow the network to model complex nonlinear relationships between the predictor variables and the outcome. Neural networks can be constructed with multiple hidden layers although there are usually no advantages to doing so.

Each node in the input layer is usually connected to each node in the hidden layer, and each node in the hidden layer is usually connected to each node in the output layer. In this example, there are two input nodes where the values of age ($x1$) and sex ($x2$) are input into the network along with a bias weight which is the equivalent of the intercept term found in a regression model. In the field of neural networks, it is common practice to scale each of the input variables to the range 0 to 1 or −1 to 1. In this example, the value of age is assumed to be scaled to the range 0 to 1 from rhe actual range of 0 to 100.

Each input node is connected to each hidden node in the hidden layer by a connection weight (e.g., −0.14, −0.65, −0.04, −1.46) as shown in Figure 1. The connection weights can be thought of as the neural network equivalent of the β coefficients in a regression model and contain the "knowledge" acquired by a neural network after it has been trained. At each hidden node, a weighted linear combination of the inputs to it is summed (including the bias weight) to determine the net input to that node and then this result is passed through an activation function, most commonly the logistic or sigmoid function. The activation of each hidden node (h1, h2) is then multiplied by a second set of connection weights (e.g., −0.8, −1.49) and added to a bias weight (e.g., −0.22). A logistic transformation of the weighted inputs to the output node is applied to determine the overall output (o1) of the network. The output of the network corresponds to the network prediction of the outcome and will range from 0 to 1 when the network output activation function is the logistic function. Many variants of this basic neural network architecture exist [16,17]. Networks may be developed using alternative activation functions in the hidden and output nodes such as hyperbolic tangent and linear functions [16,17]. Linear functions in the output node are used when using neural networks to predict continuous outcomes [17].

A logistic regression model for predicting the same outcome is shown in Figure 2. As demonstrated in this example, even a relatively simple neural network with one hidden layer requires significantly more parameters (nine connection weights) to be estimated than the simplest logistic regression model (three β coefficients) for predicting the same outcome. It is relatively easy to calculate the predicted probability of a patient dying from disease X, and to understand the relationship between age, sex, and the probability of dying in this logistic regression model. In contrast, the values of age and sex undergo three logistic transformations before the output of the neural network is determined, and it is difficult to understand the exact nature of the relationship between the values of age, sex, and the predicted outcome in the neural net.

The connection weights in an untrained neural network are initially assigned random values. In a neural network, the weights connecting two nodes are usually represented as w_{ij}, where i and j are subscripts for the two nodes being connected [16]. Estimating the optimal values of these connection weights is the major purpose behind training a neural network model. The network training algorithm is used to gradually adjust the weights in the network to minimize the difference between the predicted output of the network (o_p) and the known value of the outcome variable (t_p). This difference is known as the error of a neural network and is similar to the concept of minimizing the residuals in statistical regression. The total error (E) of a neural network is usually determined over the whole data set and may be calculated as follows:

$$E = \sum_p (t_p - o_p)^2 \qquad (2)$$

where E is the total error of the network, t_p is the desired or known

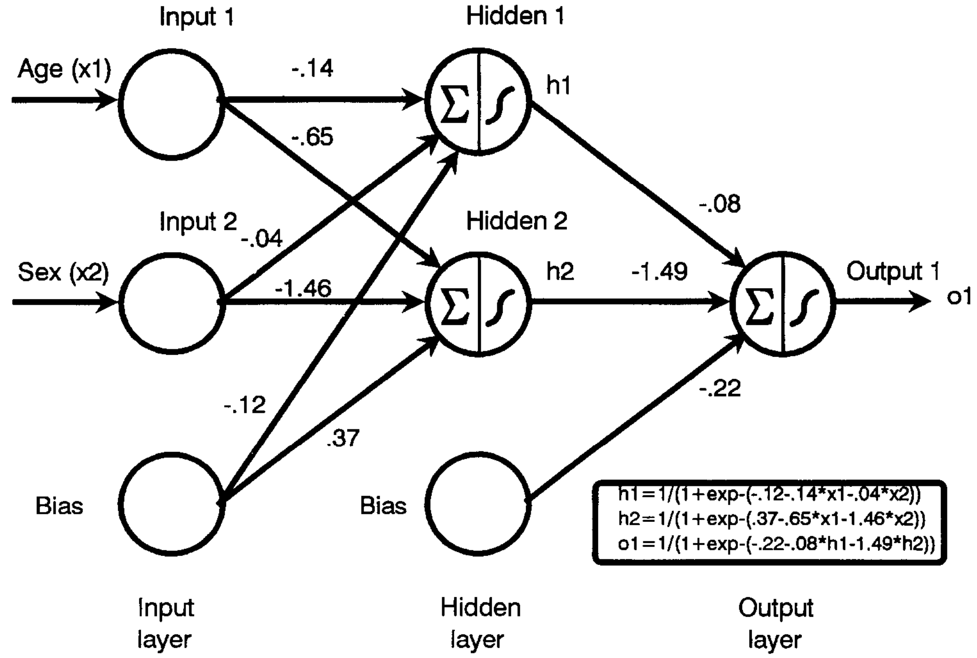

FIGURE 1. Diagram of an artificial neural network trained to predict the probability of a patient dying from a hypothetical disease on the basis of the age (*x*1) and sex (*x*2) of the patient. Each circle represents a node, while each line represents a connection weight (actual weight values are shown). The nodes of the network are arranged in three layers (input, hidden, output). A logistic activation function is used in both the hidden (h1, h2) and output nodes (o1). (h1 and h2 are the activation of hidden nodes 1 and 2; o1 is the predicted output of the network.) At each hidden and output node, a weighted linear combination of the inputs is summed and then a logistic transformation is applied.

output, o_p is the output predicted by the network, and p is an index over all observations in the data set. (In statistical regression notation, Y corresponds to t_p, and $\hat{Y}$ corresponds to o_p.) The error equation is often called the *objective function*. Variants of this error equation are commonly used, including the mean square error (MSE), which is the mean value of the total error, and the root mean square (RMS) error, which is the square root of the MSE [16–18]. Alternative objective functions such as likelihood functions may also be used and may be theoretically preferable when the outcome of interest is dichotomous [19].

A variety of learning algorithms exist for determining the optimal network weights, with the back-propagation algorithm being the one most commonly used. The exact mathematical equations in the back-propagation learning algorithm are complex and are described in detail elsewhere [14–16]. However, the algorithm may be characterized as follows. A data set is divided into a training set and one or more test sets. A training case or pattern (i.e., an observation) is selected from the training set (randomly or sequentially) and then the values of the predictor variables are used as inputs to the network to determine the predicted output of the network (e.g., o1 in Fig. 1). This predicted output is then compared with the known or desired output (e.g., 0 = live, 1 = die). If there is a difference between the predicted and the known output, an *error signal* is calculated and back-propagated through the neural network [15,16]. This signal is a mathematical factor that adjusts the value of each weight in the neural network to reduce the difference between the predicted and known output. Thus, the next time a network encounters a training case with similar input values, its prediction will be more accurate. Cases are selected from the training set and are continually presented to the neural network until the overall error has been minimized. One pass through all of the training cases is considered an *epoch* of training. The overall duration of training is often expressed in terms of the number of epochs required to reach an error minimum.

Figure 3 shows a typical relationship between the amount of error in the network and the duration of training [18]. As training proceeds, the error of the network gradually decreases until it reaches a minimum in the training set. However, the error in the test set

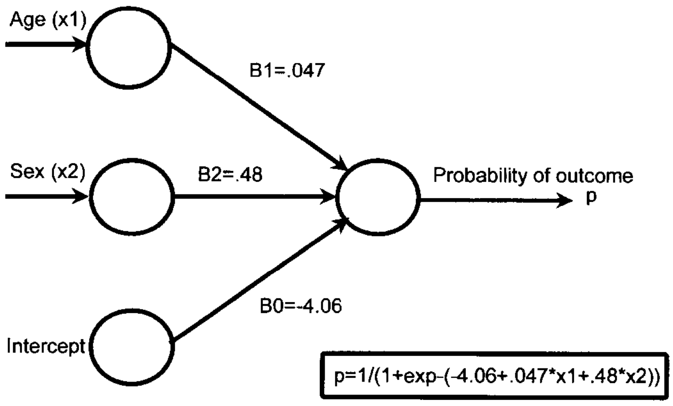

FIGURE 2. Diagram of a logistic regression model designed to predict the probability (*p*) of a patient dying from a hypothetical disease on the basis of the age (*x*1) and sex (*x*2) of the patient. The values of the β coefficients are shown.

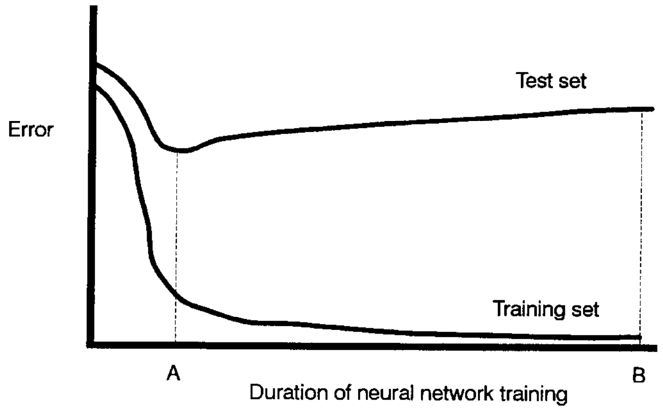

FIGURE 3. Diagram showing a typical relationship between the amount of error in the training and test data sets, and the overall duration of neural network training. (Point A represents the point at which the error is minimal in the test set, while point B represents the point at which the error is minimal in the training set.)

may decrease initially and then start to rise as the network begins to overlearn the data in the training set. It is common practice among the developers of neural network models to "cross-validate" the network on the test set periodically during training and to save the network weight configuration meeting one of two criteria: (1) the network with the minimum error in the training set (point B in Fig. 3) or (2) the network with the minimum error in the test set (point A in Fig. 3) [18]. The latter technique is often used to prevent the network from overtraining because networks are prone to overfitting. Stopping training and saving the network after a prespecified number of training epochs is also an option. A network trained using a cross-validation approach and saved at point A will likely generalize better to external test sets than a network saved at point B. Although those familiar with conventional statistical modeling might consider this to be an inappropriate technique because the network is using the test set to determine its optimal weight structure, proponents of neural networks have sometimes cited this as an advantage over conventional statistical techniques, which are not usually designed to optimize performance in the test set although they can be modified to do so [20,21]. However, for evaluating the external generalizability of a trained neural network, another independent test set should be used when this method of preventing overfitting is utilized.

The *learning rate* and *momentum* are two parameters in the back-propagation training algorithm, set by the network developer [14–16]. Typical values of the learning rate and momentum are between 0 and 1. The learning rate refers to the amount that each weight is changed after each training case, with larger learning rates indicating greater weight changes. The momentum term allows each weight change to be proportional to the magnitude of the previous change. It also allows the network to avoid local minimum in the error surface and speeds up the convergence of the network toward a global error minimum. Surprisingly, networks rarely become trapped in local minima [5,14]. Developing a network using the back-propagation training algorithm involves trying a variety of learning rates, momentum terms, and hidden nodes numbers to find the network architecture and weight configuration that performs best in either the training and/or test data sets.

Logistic regression models bear several structural similarities to neural network models. A main effects logistic regression model (with no interaction terms) is identical to a neural network model with no hidden layer (i.e., a single layer network) [19]. However, in logistic regression, the process converges when the likelihood function is maximized whereas in a neural network it is usually a least-squares error function that is minimized (although likelihood functions can be maximized, too). Spackman has shown that a single-layer back-propagation neural network developed using a maximum likelihood objective function will converge to the same solution as a main effects logistic regression model [19].

The power of a neural network over a main effects logistic regression model lies in the hidden layer of nodes. Hidden nodes are "hidden" from the input and the output of a neural network and act as data feature detectors. It is here that the network is able to learn implicitly any arbitrarily complex nonlinear relationship between the dependent and independent variables, given a sufficient number of hidden nodes [6]. Neural networks use the actual data to determine what the functional relationship should be between the dependent and independent variables rather than a prespecified relationship as occurs when one fits a logistic regression model. Of course, any desired number of interactions or nonlinear terms can also be added to logistic regression models although formal algorithms (including stopping rules) for doing so are still somewhat lacking.

DISCUSSION

This article has reviewed the features of logistic regression and neural networks, and has illustrated some of the similarities and differences between the two methods. Several investigators have compared the two methods using different data sets, with some researchers finding superior performance for the neural network and other authors finding no differences in overall predictive performance [8,22–24]. These comparisons are highly dependent on the nature of a particular data set, and one cannot conclude whether one method will be superior to the other method in a given data set without an empirical comparison. The author's experience has been that neural network models and logistic regression models usually have similar levels of predictive performance in external test data sets. As summarized in Table 2, neural networks offer both advantages and disadvantages over logistic regression for predicting medical outcomes. These issues are discussed in detail below.

Advantages of Using Neural Networks

1. Neural network models require less formal statistical training to develop: Working artificial neural network models can be developed by newcomers to neurocomputing within a relatively short time frame (i.e., days to weeks), conditional on the availability of an appropriate data set and neural network software. Any data set that can be analyzed using conventional logistic regression can also be used to develop a neural network-based prediction model. Neural networks can be trained using both continuous and categorical input and output variables, although some transformations of the data may be necessary depending on the software being used. Networks tend to work best when the data have been normalized [17]. A variety of neural network software packages are available commercially and range from simple, user-friendly, back-propagation only packages to more complex packages with multiple training algorithms [25]. Custom-programming a neural network simulator is also an option, and books have been published describing the appropriate code [26].

TABLE 2. Advantages and disadvantages of using neural networks for predicting medical outcomes

Advantages

1. Neural network models require less formal statistical training to develop
2. Neural network models can implicitly detect complex nonlinear relationships between independent and dependent variables
3. Neural network models have the ability to detect all possible interactions between predictor variables
4. Neural networks can be developed using multiple different training algorithms

Disadvantages

1. Neural networks are a "black box" and have limited ability to explicitly identify possible causal relationships
2. Neural networks models may be more difficult to use in the field
3. Neural network modeling requires greater computational resources
4. Neural network models are prone to overfitting
5. Neural network model development is empirical, and many methodological issues remain to be resolved

The developer of a neural network model needs to identify, prior to training, which variables might potentially be important predictors of an outcome. Developers need only a basic understanding of neural network model structure and the parameters that can be adjusted before being able to develop a model. An understanding of the underlying mathematical equations that comprise training algorithms such as back-propagation is not required. However, knowledge of more advanced neural network training concepts such as shrinkage, and overfitting may be beyond the scope of a newcomer and may be required for optimal model development [6,27].

In contrast, developers of logistic regression models require formal statistical training and need to understand a variety of statistical concepts, including backward and forward stepwise regression, p values, odds ratios, multicollinearity, interactions, and main effect terms. A much greater time investment may be required to learn logistic regression modeling techniques.

2. *Neural networks can implicitly detect complex nonlinear relationships between independent and dependent variables*: Neural networks have the ability to detect implicitly any complex nonlinear relationships between independent and dependent variables [5,6]. If a significant amount of nonlinearity between the predictor variables and the corresponding outcomes exists in a training data set, then the network will automatically adjust the connection weights in its structure to reflect these nonlinearities. The predictor variables in a neural network usually undergo a nonlinear transformation at each hidden node and output node (see Fig. 1), and thus a neural network can potentially model much more complex nonlinear relationships than a logistic regression model can, for which there is usually only one nonlinear logistic function. Empirical observations suggest that when complex nonlinear relationships exist in data sets, neural network models may provide a tighter model fit than conventional regression techniques [17].

Logistic regression models can also be used to model complex nonlinear relationships between independent and dependent variables. However, this requires an explicit search for these relationships by the model developer and may require complex transformations of predictor or outcome variables. Appropriate transformations may not always be available for improving model fit, and significant nonlinear relationships may go unrecognized by model developers. Other modeling techniques such as regression splines are available for modeling nonlinearities but these require moderate to advanced statistical knowledge [28].

3. *Neural networks have the ability to detect all possible interactions between predictor variables*: The hidden layer of a neural network gives it the power to detect interactions or interrelationships between all of the input variables. When important interactions exist and are not modeled explicitly in a logistic regression model, then a neural network model may be expected to predict a particular outcome better than a logistic regression model. In the neural network example discussed earlier, in which age and sex are the only predictor variables, there is only one possible interaction, and a neural network model would be unlikely to offer any predictive advantage over a logistic regression model with an age–sex interaction term. However, as the number of predictor variables in any logistic regression model increases, the number of possible interactions increases simultaneously, and model developers may find it cumbersome to test for all possible two-way interactions, let alone all possible higher-order interactions. Developing a neural network model with and without a hidden layer could potentially be used as a test to see if any major higher-order interactions exist, because if this were the case, one would expect the network with a hidden layer to perform better [29]. However, the search for higher-order interactions in a neural network implicitly or using logistic regression explicitly might lead to overfitting with both types of methods.

One of the decisions confronting the developer of a neural network model is the number of nodes to put in a hidden layer. No theory exists for predetermining the optimal number, and network developers need to try a variety of hidden node numbers to find the optimal number. Having too many hidden nodes can be likened to the situation of fitting a logistic regression model with too many interaction terms. This may lessen the ability of a network to generalize well to external validation sets.

4. *Neural networks can be developed using multiple different training algorithms*: Although in this article we have focused primarily on neural network training using the back-propagation learning algorithm, a variety of alternative neural network training algorithms are also available for use that may actually perform better, and variants of the back-propagation algorithm have also been developed [18,30]. Many different training algorithms exist, and at present the optimal training algorithm for a particular problem can only be determined empirically. Most studies within the medical domain have been conducted using the back-propagation algorithm only [7,8,11,12,22–24]. However, new improved learning algorithms are continually being developed by researchers in the field and may offer promise as improved outcome prediction tools.

Disadvantages of Using Neural Networks

1. *Neural networks are a "black box" and have limited ability to explicitly identify possible causal relationships*: Developers of statistical models often have as their primary goal causal inference. Although the presence of a statistical relationship between a predictor variable and an outcome alone does not imply causality, logistic regression models are superior to neural networks for identifying possible causal relationships. In logistic regression, a model developer is able to determine which variables are most strongly predictive of an outcome based on the magnitude of the β coefficients and the associated odds ratios. Furthermore, through a stepwise variable selection process, the model developer may be able to eliminate a number of indepen-

dent variables that are not related to a particular outcome of interest.

A neural network model is a relative "black box" in comparison to a logistic regression model [4]. The model developer sets up the training data for the network and then essentially lets the network train itself and determine which input variables are the most important. It cannot easily be determined which variables are the most important contributors to a particular output, and a neural network model may contain a number of unimportant predictor variables that the developer fails to appreciate. No well-established criteria exist for interpreting the weights in a connection weight matrix.

Investigators have been actively developing techniques to increase their understanding of the internal logic of neural networks. One of these techniques is to train the neural network with each input variable node removed one at a time and then to observe the effect on network performance [31]. Other investigators have started to develop regression-like techniques that are used to examine the connection weights of various input variables and that are used to determine which variables can be removed from a model without affecting its performance [32,33]. However, none of these techniques has achieved widespread use and offers the ease of interpretation of the odds ratios associated with the coefficients of a logistic regression model. Reducing the black box nature of neural networks will likely hasten their acceptance within the medical community.

2. *Neural network models may be more difficult to use in the field*: Neural network models may be more difficult to use in the field than logistic regression models developed using the same data set. For other users to be able to use a trained neural network model, one needs either to make a copy of the trained network software available to each end user or one needs to publish the trained connection weight matrices [34]. Although neural network connection weight matrices have occasionally been published, these weight matrices may be prohibitively large and difficult to interpret (e.g., one [i by j] matrix and one [j by k] matrix where i is the number of input nodes, j is the number of hidden nodes, and k is the number of output nodes) [16,34]. In contrast, logistic regression β coefficients can be published in scientific journals and then any end user with a handheld calculator can calculate the predicted probability of an outcome. Logistic regression models can be disseminated to a much wider audience than neural network models can. To justify their use, the onus is probably on the developers of neural network models to demonstrate better predictive performance than the corresponding logistic regression models, since they are far less transparent.

The output of a neural network model may be harder to interpret than that of a logistic regression model under certain circumstances. The logistic regression equation allows one to easily calculate the predicted probability of an outcome. Richard and Lippmann have shown that neural network model outputs can also be interpreted directly as Bayesian probabilities provided there are sufficient training data, the network is complex enough, and the training data represent true likelihood distributions and *a priori* class probabilities [35]. However, when these criteria are not fulfilled, the neural network model developer may need to calibrate the output of the network against the actual data before interpreting the output as a probability [18].

3. *Neural network modeling requires greater computational resources*: Neural network model development is a computationally intensive procedure that requires much greater computational time. With standard personal computer hardware and the back-propagation algorithm, it can often take hours to days to weeks before a network will converge to an optimum learning state with minimum error [26]. In contrast, logistic regression models of relatively large data sets can be tested within seconds on standard computers, and new models can then be evaluated. Quicker variants of the back-propagation algorithm have been developed but these may still require large amounts of training time [18,30]. Some commercial vendors have developed special network accelerator boards that can be plugged into the expansion slot of a computer to speed network training time significantly [25]. Training times can also be improved by using more powerful mainframe or minicomputers. As future advancements in computer technology occur and quicker training algorithms are developed, this limitation of neurocomputing should become less of an issue.

4. *Neural network models are prone to overfitting*: The ability of a neural network model to model interactions and nonlinearities implicitly may also be a disadvantage because it may lead to overfitting a training data set with poor performance in external test data sets. In general, overfitting may be prevented in three main ways: by limiting the number of hidden nodes, by adding a penalty term to the objective function for large weights, or by limiting the amount of training using cross-validation [18]. Although all three methods are potentially useful, limiting the amount of training using cross-validation is the most popular because it is the least computationally intensive. This process was briefly discussed earlier (see Fig. 3). In this technique, data are divided into training and test data sets. The network is trained using the training data and at predetermined training intervals (e.g., after every 1000 training cases), the error of the network is determined in both the training and the test data sets, and the connection weight configuration of the network is saved. Training continues with the error in the training set usually declining with additional training. However, the corresponding error in the test set may start to increase after a while as the network begins to "memorize" the noise in the training set. The network weight configuration with the least error in the test set is considered the best network and is used for future predictions in external validation sets. However, this technique has the disadvantage of requiring an additional test set to evaluate the true generalizability of a trained network, and a second test set may not always be available.

Limiting the number of hidden nodes may reduce the amount of overfitting but it also reduces the power of the network to model complex nonlinear relationships. Adding a penalty term to the objective training function is an advanced method of preventing overfitting and is described in detail elsewhere [18].

An alternative to the cross-validation training/test data set approach would be to use bootstrapping, in which all of the data are used both to train the network and to evaluate its performance [36]. This technique has the advantage of using all of the available data to develop a model and it has been used by some neural network researchers [37]. It may also provide a potentially more accurate assessment of the predictive ability of a network because one can create a large number of test data sets. However, most neural network packages do not have a bootstrapping algorithm available for easy out-of-data validation tests.

5. *Neural network model development is empirical, and many methodological issues remain to be solved*: Neural network modeling is a relatively new technique that has emerged from a variety of disciplines, and a number of important methodological issues remain to be resolved [6,30]. It is difficult to conclude that one has developed the best possible neural network for a particular application given the multiple training algorithms available and the empirical nature of model development. At present, model developers need to go through an empirical process of performing sensitivity analyses on training parameters such as learning rates, momentum terms, and number of hidden nodes. Few guidelines exist for predetermining

which neural network structure and training algorithms will perform best for a given problem. Furthermore, other advantages associated with the well-established technique of logistic regression have yet to be achieved by neural network models. These missing features include the ability to place confidence limits around model output predictions and parameter estimates, and how best to select variables for inclusion in a model. With additional research, many of these important methodological issues will likely be resolved.

CONCLUSIONS

This article has discussed the advantages and disadvantages of using artificial neural networks versus logistic regression as a technique for predicting dichotomous medical outcomes. It is unlikely that one method will be the technique of choice in all circumstances, and the choice as to which technique to use will depend to a large extent on the nature of a particular data set and the goals of a model developer. Neural networks may be particularly useful when the primary goal is outcome prediction and important interactions or complex nonlinearities exist in a data set, although these preferences are less clear if a regression modeler can model them using appropriate regression splines and interaction terms. Logistic regression remains the clear choice when the primary goal of model development is to look for possible causal relationships between independent and dependent variables, and a modeler wishes to easily understand the effect of predictor variables on the outcome. It is possible that some form of hybrid technique that incorporates the best features of both logistic regression and neural network modeling might lead to the best possible outcome prediction model [32]. Further research will be acquired to determine those clinical circumstances under which the advantages of neural networks over logistic regression outweigh the disadvantages in the prediction of medical outcomes.

Dr. Tu was supported by a Health Research Personnel Development Program Fellowship (04544) from the Ontario Ministry of Health. The results and conclusions are those of the author, and no official endorsement by the Ministry of Health is intended or should be inferred. This work was supported in part by a seed grant from the Center for Risk Analysis, Harvard School of Public Health.

The author thanks Dr. C. D. Naylor, Dr. D. A. Redelmeier, Dr. J. B. Willett, Dr. M. C. Weinstein, and two anonymous reviewers for their helpful comments on earlier versions of this manuscript.

References

1. Watson JH, Sox HC, Neff RK, Goldman L. Clinical prediction rules: Applications and methodological standards. **N Engl J Med** 1985; 313: 793–799.
2. Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and Regression Trees. Belmont, CA, Wadsworth International, 1984.
3. Hosmer DW, Lemeshow S. Applied Logistic Regression. New York, John Wiley & Sons; 1989.
4. Guerriere MRJ, Detsky AS. Neural networks: What are they? **Ann Intern Med** 1991; 115: 906–907.
5. Hinton GE. How neural networks learn from experience. **Sci Am** 1992; 267: 145–151.
6. White H. Learning in artificial neural networks: A statistical perspective. **Neural Comput** 1989; 1: 425–464.
7. Baxt WG. Use of an artificial neural network for the diagnosis of myocardial infarction. **Ann Intern Med** 1991; 115: 843–848.
8. Buchman TG, Kubos KL, Seidler AJ, Siegforth MJ. A comparison of statistical and connectionist models for the prediction of chronicity in a surgical intensive care unit. **Crit Care Med** 1994; 22: 750–762.
9. Derksen S, Keselman HJ. Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. **Br J Math Stat Psychol** 1992; 45: 265–282.
10. Spiegelhalter DJ. Probabilistic prediction in patient management and clinical trials. **Stat Med** 1986; 5: 421–433.
11. Patil S, Henry JW, Rubenfire M, Stein PD. Neural network in the clinical diagnosis of acute pulmonary embolism. **Chest** 1993; 104: 1685–1689.
12. Tu JV, Guerriere MRJ. Use of a neural network as a predictive instrument for length of stay in the intensive care unit following cardiac surgery. **Comput Biomed Res** 1993; 26: 220–229.
13. Liestol K, Andersen PK, Andersen U. Survival analysis and neural nets. **Stat Med** 1994; 13: 1189–1200.
14. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. **Nature** (London) 1986; 323: 533–536.
15. McClelland JL, Rumelhart DE. Training hidden units: The generalized delta rule. In: McClelland JL, Rumelhart DE, Eds. Explorations in Parallel Distributed Processing. Cambridge, MA: MIT Press: 1988: 121–160.
16. Zurada JM. Introduction to Artificial Neural Systems. St. Paul, MN: West Publishing Company, 1992.
17. Ward Systems Group, Neuroshell 2. Ward Systems Group, Frederick, MO, 1993.
18. Smith M. Neural Networks for Statistical Modeling. New York, Van Nostrand Reinhold; 1993.
19. Spackman KA. Maximum likelihood training of connectionist models: Comparison with least squares back-propagation and logistic regression. In: Proceedings of the Fifteenth Annual Symposium on Computer Applications in Medical Care. McGraw-Hill, Inc., Washington, DC, 1991, pp. 285–289.
20. Hutton LV. Using statistics to assess the performance of neural network classifiers. Johns Hopkins APL Tech Digest 1992; 13: 291–299.
21. Shao J. Linear model selection by cross-validation. **J Am Stat Assoc** 1993; 88: 486–494.
22. Tu JV. A Comparison of Neural Network and Logistic Regression Models for Predicting Length of Stay in the Intensive Care Unit following Cardiac Surgery. Masters dissertation. University of Toronto 1993.
23. Doig GS, Inman KJ, Sibbald WJ, Martin CM, Robertson JM. Modeling mortality in the intensive care unit: Comparing the performance of a back-propagation, associative-learning neural network with multivariate logistic regression. In: Proceedings of the Seventeenth Annual Symposium on Computer Applications in Medical Care. McGraw-Hill, Inc., Washington, DC, 1993, pp. 361–365.
24. Lette J, Colletti BW, Cerino M, *et al.* Artificial intelligence versus logistic regression statistical modeling to predict cardiac complications after noncardiac surgery. **Clin Cardiol** 1994; 17: 609–614.
25. Neural networks: Theory and practice. **Byte** 1989; 14: 244–245.
26. Blum A. Neural Networks in C++. An Object-Oriented Framework for Building Connectionist Systems. New York, John Wiley & Sons; 1992.
27. Astion ML, Wilding P. The application of back-propagation neural networks to problems in pathology and laboratory medicine. **Arch Pathol Lab Med** 1992; 116: 995–1001.
28. Durrleman S, Simon R. Flexible regression models with cubic splines. **Stat Med** 1989; 8: 551–561.
29. Laurentiis MD, Ravdin PM. Survival analysis of censored data: Neural network analysis detection of complex interactions between variables. **Breast Cancer Res Treat** 1994; 32: 113–118.
30. Hinton GE. Connectionist learning procedures. **Artificial Intel** 1989; 40: 185–234.
31. Baxt WG. Analysis of the clinical variables driving decision in an artificial neural network trained to identify the presence of myocardial infarction. **Ann Emerg Med** 1992; 21: 1439–1444.
32. Spackman KA. Combining logistic regression and neural networks to create predictive models. In: Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care. McGraw-Hill, Inc., Baltimore, MD, 1992; pp. 456–459.
33. Griffith JL, D'Agostino RB, Selker HP. Statistical regression techniques for the construction, interpretation and testing of computer neural networks. **Med Decis Making** 1992; 12: 343.
34. Astion ML, Wener MH, Thomas RG, Hunder GG, Bloch DA. Application of neural networks to the classification of giant cell arteritis. **Arthr Rheum** 1994; 37: 760–770.
35. Richard MD, Lippmann RP. Neural network classifiers estimate Bayesian a posteriori probabilities. **Neural Comput** 1991; 3: 461–483.
36. Efron B, Tibshirani R. An Introduction to the Bootstrap. Chapman & Hall, New York, 1993.
37. Katz S, Katz AS, Lowe N, Quijano RC. Neural net–bootstrap hybrid methods for prediction of complications in patients implanted with artificial heart valves. **J Heart Valve Dis** 1994; 3: 49–52.