DOCUMENT RESUME

ED 326 577 TM 015 942

AUTHOR Van Nelson, C.; Neff, Kathryn J.

TITLE Comparing and Contrasting Neural Net Solutions to Classical Statistical Solutions.

PUB DATE Oct 90

NOTE 12p.; Paper presented at the Annual Meeting of the Midwestern Educational Research Association (Chicago, IL, October 19, 1990).

PUB TYPE Reports - Research/Technical (143) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.

DESCRIPTORS *Admission Criteria; *Artificial Intelligence; Class Rank; College Entrance Examinations; Comparative Analysis; *Equations (Mathematics); Failure; Grade Point Average; Grades (Scholastic); *Graduate Students; Higher Education; *Mathematical Models; Regression (Statistics); Sex Differences; Success; *Undergraduate Students

IDENTIFIERS *Neural Net Models

ABSTRACT

Data from two studies in which subjects were classified as successful or unsuccessful were analyzed using neural net technology after being analyzed with a linear regression function. Data were obtained from admission records of 201 students admitted to undergraduate and 285 students admitted to graduate programs. Data included grade point averages, admission test scores, grades, gender, class rank, course instructor, and course grades. The neural net models used were the Adeline model and the Layer Back-propagation model. The neural net model makes no assumptions about the underlying distribution of the observations. In general, a neural network may be defined as a non-programmed information reduction system that develops processing abilities in response to its environment. The function of the neural network is to learn from examples. The study findings indicate that the results obtained using the neural net models are comparable with, but not the same as, those of the classical statistical approach. While the neural net technology is not a replacement for the classical techniques, it may represent a viable alternative when certain assumptions of the statistical model are grossly violated. (TJH)

COMPARING AND CONTRASTING NEURAL NET SOLUTIONS TO CLASSICAL STATISTICAL SOLUTIONS

C. Van Nelson, Ed. D.

University Computing Services

Ball State University

Kathryn J. Neff

Ph. D. Candidate

Ball State University

Paper presentation at the Annual Meeting of the
Midwestern Educational Research Association,
Chicago, Illinois. October 19, 1990.

A neural network may be defined as "a non-programmed information reduction system that develops processing abilities in response to its environment". The function of the neural network is to learn from examples. These examples "train" the network.

The structure of such a neural network consists of interconnected processing elements where each processing element has multiple weighted inputs and a single non-linear output. The weights are developed through iterative adjustment and represent the knowledge learned by the neural network. A visual model of a network appears in figure 1.

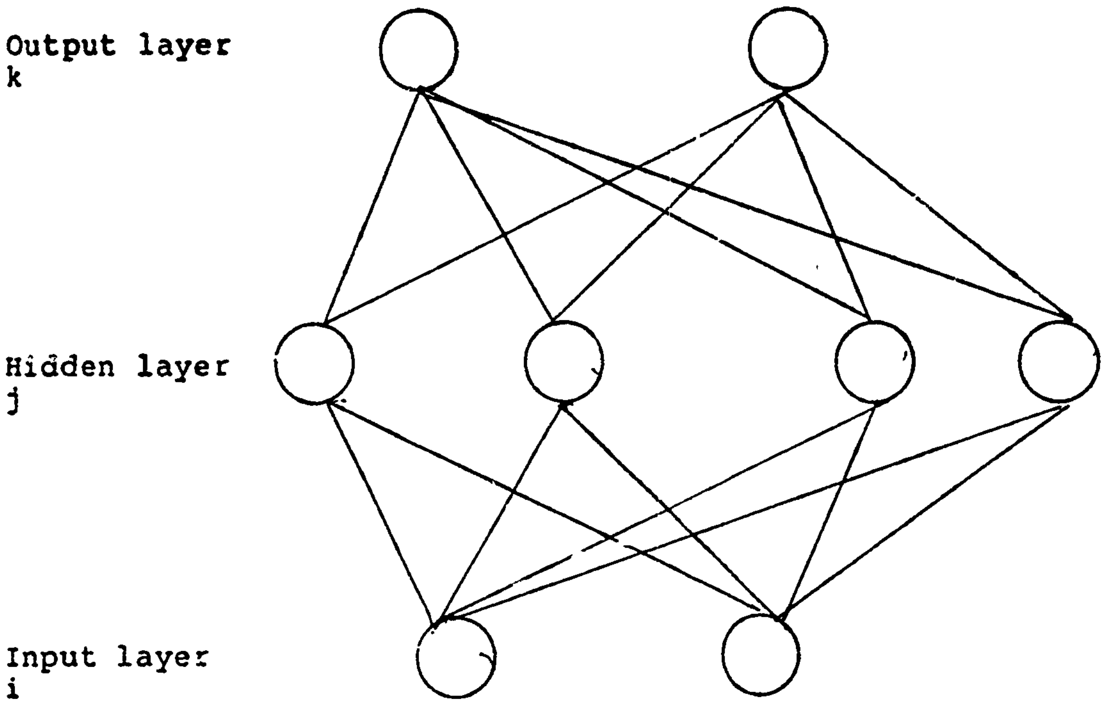

Figure 1

One use of a neural network is to classify an object as belonging to one of two or more populations. An example is that of an object hidden on the bottom of a body of water. This object may be a metal cylinder, possibly containing explosives, or it may merely be a rock. The Sonar used to explore the bottom of the body of water sends back partial information about the object. By placing known objects (rocks or metal cylinders) under the water, the neural network is trained to recognize the object from the incomplete information given by the Sonar device. Once the network is trained, the system can be used to determine whether an object detected under the water is just a rock or a metal cylinder in actual combat or surveillance duty. An important attribute of the neural network is that no assumptions are made about the distribution of the input signals received from the Sonar device.

In an educational environment, students may need to be classified where only partial information is available. For example, a decision needs to be made on admitting or not admitting a student to a particular program. Information may be available about the candidate in the form of test scores, previous grade point ratios, etc. For a sample where the outcome of success or failure is known, one calculates the coefficients of a discriminant function or a regression equation. This discriminant function or regression equation may be used to make later decisions. Determining the coefficients may be considered analogous to training the neural network. The objective of this paper is to compare the results of the classical statistical procedures in two different studies with the results obtained by using neural network techniques in these same studies.

There are many different neural network designs. Two such designs will be illustrated in this paper. The first design is the Adeline pattern recognizing control system proposed by Widrow [1] in 1963. The second design is the back propagation model. The Adeline model was selected because it represents the first application of a neural network to a real problem. This neural network is also easy to understand. The back propagation model is a more recent design and seems to show better promise for classification.

The ADELINE model, so named because it utilizes an "adaptive linear neuron was first utilized for reducing or eliminating the echo in telephone lines. A diagram of the ADELINE network is shown in figure 2.

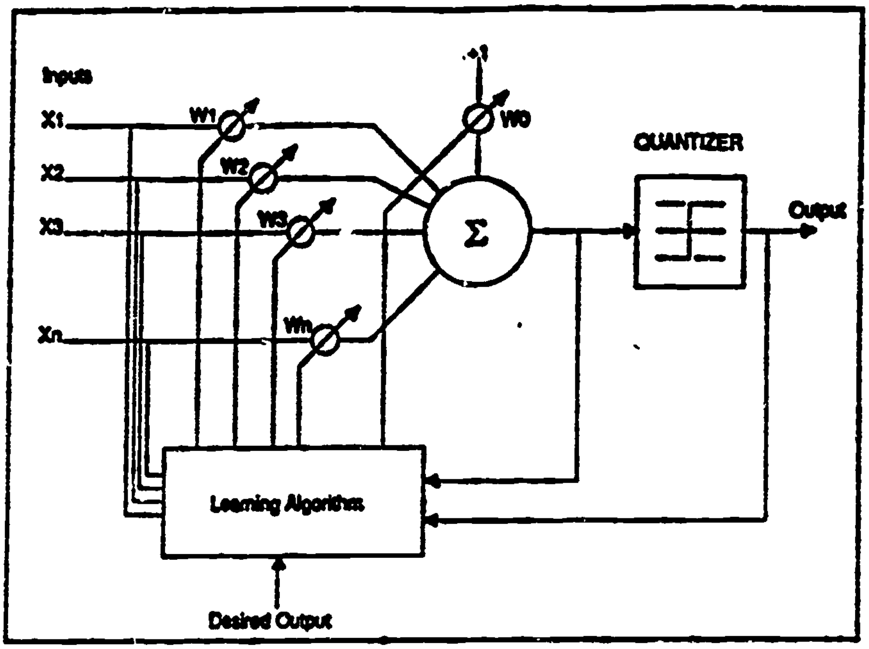
Inputs
X1
X2
X3
Xn
W1
W2
W3
Wn
+1
W0
Σ
QUANTIZER
Output
Learning Algorithm
Desired Output

Figure 2

Each input has a value of either +1 or -1. An additional constant of +1 is applied as a bias. Random values between -1 and +1 are assigned to the weights for each input and the bias. The absolute difference between the sum of the product of each input and its weight (including the bias input) is calculated. Each weight is then adjusted by the amount of the error divided by n + 1 where n + 1 is the number of weights for the n inputs plus the bias input. This process is then repeated for the next learning set (inputs with the desired output). The final output is quantized to either +1 or -1 depending on whether the sum of the products of the inputs by the weights is positive or negative. Experimental results seem to indicate that an Adeline will converage to a stable solution in five times as many learning trials as there are weights [2].

The back propagation model is more involved, but in this study produced more reasonable results. The model used in this study had three layers, an input layer, an output layer and a hidden layer. The input layer, the hidden layer, and the output layer will be referred to as the i, j, and k layers respectively. The processing function chosen is the sigmoid function which has the form $f(z) = (1 - e^{-z})^{-1}$ where z represents the vector of inputs to the neuron. This function approaches zero as z becomes negatively infinite, and approaches 1 as z goes to positive infinity. The derivative of f(z), $f'(z) = f(z) \cdot (1 - f(z))$ which indicates that the rate of change of the function is parabolic with respect to f(z). The weights are calculated to minimize the error between the desired outcomes and the actual output from the output layer. The derivation is as follows [3]:

Let the global error $E = \frac{1}{2} \sum_k (d_k - y_k)^2$, where d_k is the desired outcome and y_k is the output value produced by the neural network. These weights, w_{ij} and w_{jk} are adjusted through the sigmoid function, where $z_j = \sum_i w_{ij} \; y_i$ and $z_k = \sum_j w_{jk} \, y_j$. Using the gradient descent technique, the weights between the hidden layer and the output layer are derived as follows:

$$\Delta w_{ij} = -n\frac{\partial E}{\partial w_{ij}}$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial z_j} \cdot \frac{\partial z_j}{\partial w_{ij}}$$

$$\frac{\partial z_j}{\partial w_{ij}} = y_i$$

$$\frac{\partial E}{\partial z_j} = \sum_k \frac{\partial E}{\partial z_k} \cdot \frac{\partial z_k}{\partial z_j}$$

$$\frac{\partial E}{\partial z_j} = \frac{\partial z_k}{\partial y_j} \cdot \frac{\partial y_j}{\partial z_j}$$

$$\frac{\partial E}{\partial z_k} = -\delta_k$$

$$\frac{\partial z_k}{\partial y_j} = w_{jk}$$

$$\frac{\partial y_j}{\partial z_j} = f'(z_j)$$

So $\Delta w_{ij} = \eta(\sum_k \delta_k \, w_{jk}) \, f'(z_j) \, y_i$

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}}$$

$$\frac{\partial E}{\partial w_{jk}} = -\frac{\partial E}{\partial z_k} \cdot \frac{\partial z_k}{\partial w_{jk}}$$

$$\frac{\partial z_k}{\partial w_{jk}} = y_j$$

$$\frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial z_k}$$

$$\frac{\partial E}{\partial y_k} = d_k - y_k$$

$$\frac{\partial y_k}{\partial z_k} = f'(z_k)$$

Therefore, $\Delta w_{jk} = -\eta(d_k - y_k)f'(z_k)y_j$

Let $\delta_k = (y_k - d_k)f'(z_k)$

Then $\Delta w_{jk} = \eta\delta_k y_j$

The derivation of the weights in terms of minimizing the global error assumes that a local minimum does not exist. It should be noted that the change in the value of a weight between the input layer and the hidden layer depends upon the weights and the change in the weights between tha hidden layer and the ɔutput layer. This is the reason that this particular model is known as the back propagation model.

For the first study, data were obtained for students at a Midwestern university who had been admitted to a graduate program in business for the 1987-88 and 1988-89 academic years. The data available were the undergraduate grade point average, the graduate grade point average and score on the GMAT. Complete information was available for 285 students. The BMDP7M stepwise discriminant analysis program of the 1988 BMDP VAX/VMS [4] package was run on a VAX 11/785. Students who had a graduate grade point average of less than 3.00 on a 4.00 were classified as failing, while students with a graduate grade point average of 3.00 and above were classified as successful. The classification function for the passing group was

$$y = 25.29 * ugpa + .116 * GMAT - 70.683$$

and for the failing group was

$$y = 24.289 * ugpa + .110 * GMAT - 64.84$$

The result of the analysis was that of the 47 who were considered failing, the discriminant function correctly classified as failing 28, while 19 were misclassified as passing. Thus, 59.6% of the failures were correctly classified. Of the 238 who had graduate grade point ratios of 3.00 and above, 142 were classified as successful, while 96 were classified as failures. This represents a 59.7% accuracy on classification.

To prepare the data for the neural network analysis, the graduate grade point average was coded as -1, 1 if this value was 3.00 or greater and 1 ,-1 if the graduate grade point average was less than 3.00. The undergraduate grade point average was coded as 1, -1, -1, if the undergraduate grade point average was less than 2.5, -1, 1, -1 if this ratio was at least 2.5 but less than 2.75, and -1, -1, 1 if the ratio was at least 2.75. GMAT scores less than 450 were coded as 1, -1 while GMAT scores of 450 and above were coded as -1, 1. The value of the GMAT was selected because 450 was the value used to admit a student if the student's undergraduate grade point average was below 2.75. Since there had been a desire to lower the grade point average for admittance to 2.5, this value was selected to define the groups.

When the ADELINE model was run using NeuralWorks Professional II [2] on a personal computer with a 486 chip, all students were classified as successful. This result is not surprising, since some screening had been done on these students.

When the same data was run with the back propagation model, the match of the desired outcome by the classification of the network was about 60%. These results were judged the same as the discriminant function classification. The same coding scheme was used, except that zero was substituted for -1.

The results from this study indicated that there might be promise for the back propagation model. Therefore, a different data set was utilized for another comparison.

For the second study, data on students admitted to an undergraduate computer science program were obtained. The students were considered successful in the first course on this program if they obtained a grade of C or better, and unsuccessful if they received a D or an F. This scheme was used since faculty members differ in grading practices in awarding an A or a B, or in awarding a B or a C, but are in more agreement as to successful students (A, B, or C) and unsuccessful students (D or F). Other predictive information included the sex of the student, male or female, SAT quantitative scores, SAT verbal scores,which one of five instructors taught the computer science course, high school class percentile rank, success in the first calculus course, and success in the second calculus course. Three levels were given to the variables representing the calculus courses. One level was that the student completed the course with an A, B, or C grade. The second level was completion of the course with a D or an F. The third level was that the student had not enrolled in the course. Finally, to further examine the effect that an instructor might have, the instructor variable was coded as a different level for each of five instructors who taught the computer science course. A total of 201 students who took the first course on the computer science major during the academic year 1984-85 provided the data for this study.

The statistical technique applied was logistic regression using the BMDP LR procedure from the BMDP statistical package [4]. The SAT scores and high school class percentile rank were considered as continuous variables, while success in each of the calculus courses, instructor, and sex were coded as dummy variables. The dependent variable, success in the first computer science course was, of course, coded as zero or one, representing failure or success. The final variables remaining in the regression equation in order of the amount of variance predicted were success in the first calculus course, high school class percentile rank, and the sex of the student. Of the 201 students, complete data were available for 180. The final result of this analysis produced the following: 115 passed; 65 failed. With probability greater than .5, 123 were predicted to pass, 57 were predicted to fail. Of the 123 predicted to pass, 27 failed. Of the 57 predicted to fail, 19 passed. Thus, 22% of those predicted to pass, failed and 33% of those predicted to fail, passed. Overall, then, the analysis was accurate about 76% of the time.

In order to perform the neural network analysis, the continuous variables were recoded as follows: The SAT quantitative scores below 490 were coded as 1 0 0; SAT quantitative scores equal to or greater than 490 but less than 580 were coded as 0 1 0; SAT quantitative scores greater than or equal to 580 were coded as 0 0 1. SAT verbal scores less than 410 were coded as 1 0 0; SAT verbal scores greater than or equal to 410 but less than 480 were coded as 0 1 0; SAT verbal scores greater than or equal to 480 were coded as 0 0 1. High school class percentile rank below 66 was coded as 1 0 0; high school class percentile rank greater than or equal to 66 but less than 89 was coded as 0 1 0; high school class percentile rank greater than or equal to 89 was coded as 0 0 1. These points were selected to divide the group into the lower third for each variable, the middle third and the upper third. When data were missing, the coding was 0 0 0. Thus all 201 cases could be used even though the some variables had missing values for a particular case.

The particular neural network selected was the back propagation model. There were 22 input nodes; 2 for sex, 5 for instructor, 3 for SAT quantitative classification, 3 for SAT verbal classification, 3 for classification of high school class percentile rank, 3 for classification in the first calculus course and 3 for classification in the second calculus course. The output contained 2 nodes: one node for success in the computer science course and one node for failure in the computer science course. One hidden layer was used.

In this analysis, 130 students passed the course and 71 failed the course. From the output node representing success, 144 cases had values greater than .5. From the output node representing failure, 57 cases had values greater than .5. Therefore, the prediction was 144 successes and 57 failures. Of the 144 predicted successes, 14 failed. Therefore the neural net failed to predict success accurately 9.7% of the time. Of the 57 failures predicted, one student passed. Thus, the neural net was inaccurate in predicting failure in 1.7% of the cases. Thus the neural network had 92.5% accuracy overall in correctly classifying a student from the available information.

The results do show promise for the use of neural networks where traditional statistical models have been utilized. However, several disadvantages were noted:

1. Although continuous input can be used, the network failed to converge to a solution. This necessitated categorizing the continuous variables.

2. The training of the neural network involves considerable computer power and time. In this study, an IBM PS/2 model 70 which utilizes a 486 processor was used. The investigators were the sole users of this equipment. Some experiments in training a network took 24 hours of uninterrupted time.

3. There are no real guidelines as to which model to employ, how many nodes should be in the hidden layer, or even whether more than one hidden layer has any advantage.

[1] Widrow, Bernard and Smith, Fred W. "Pattern-Recognizing Control Systems" in Computer and Information Sciences, Tou, Julius T. and Wilcox, Richard H. editors, Spartan Books, Inc. Washington, D. C., 1964, PP. 288-317.

[2] Klimasaukas, Casimir, Guiver, John, and Pelton, Garrett NeuralWorks Professional II: User's Guide. NeuralWare, Inc. Pittsburgh, PA. 1989.

[3] Gustafson, Steven, Neural Networks: Review of Current Technology (Tutorial Notes) 5th Annual Aerospace Applications of Artificial Intelligence Conference AAAIC 1989, Dayton, Ohio, 1989.

[4] BMDP Statistical Software, Vax/Vms Version 1988.