

where l is a nonnegative real number. A popular choice is the least square problem where $l=2$. The objective is to minimize $\sum_p E^p$, where the sum is taken over the patterns in the training sample. Hence network training is an unconstrained minimization problem.

The most popular algorithm for training is called back-propagation [29]. Recently, we showed that training can be vastly improved with known nonlinear optimization algorithms [16] [36]. Indeed, for training the networks presented in this paper we used our own GRG2-based system. GRG2 is a widely distributed nonlinear programming software [19]. For details of our system, see Subramanian and Hung [36].

RESEARCH QUESTIONS

While neural networks have been used in many classification problems, in our view, no systematic evaluation has been done to answer the following questions:

- Q1: What is the appropriate neural network architecture for a classification task?
- Q2: If observations are sampled from populations, what are the appropriate sample sizes?
- Q3: How is a neural network classifier compared with the classical ones such as LDA, QDA, and nonparametric methods?

Network architecture refers to the number of layers, the number of nodes in each layer, and the number of arcs and the nodes they connect. Other network design decisions include the choice of activation functions, whether to include biases or not, and the target values for the output nodes. Lippmann [22] shows that a single-layer network is sufficient for linear separation functions, a two-layer network is sufficient for quadratic separation curves or curves which form convex regions, and a three-layer network can form any desired decision regions. Huang and Lippmann compare different architectures for problems of different characteristics and conclude that "performance is best when network structure matches the problem" [15, p. 492]. However, the choice of architecture may also depend on the classification objective. If the objective is to classify a given set of objects as well as possible, then a larger network may be desirable. On the other hand, if the network is to be used to predict the classification of unseen objects, then a larger network is not necessarily desirable.

It is reasonable to say that a larger sample is needed for a more complex problem so that the separation curve can be defined. However, a larger sample requires more effort to train the network. In addition, the question of the effect of sample size on the ability of a network to generalize has not been totally answered.

The final question is of interest to those who have a classification task to perform: what is the best method for my problem? Comparisons have been done by many people [6] [15] [22] [37]. However, most of the classification problems in these studies are either actual data or created to test the limits of neural networks. We feel that it would be useful to have a controlled deviation from the classical assumptions so that the results would be more generalizable.

In an earlier paper [37], we sought answers to the last two questions on problems with normal attributes and equal covariance matrices. We found that larger samples do lead to greater generalizability for neural networks. Neural networks

were comparable but no better than LDA in two-group two-variable problems. However, as either the number of groups or the number of variable increases, neural networks begin to show its superiority. As classification task becomes more complex, neural networks seem to perform better, as compared to the classical methods. This study extends the previous one by looking at complexity in other directions, namely, non-normality and interaction between attributes.

DESIGN OF COMPARATIVE STUDY

To find answers to the above three questions, a computer experiment was conducted. The experimental subjects are two-group two-variable classification problems. Three types of problems are considered.

P1: Variables have a bivariate normal distribution with equal variance-covariance matrices across the groups.

P2: Variables have a bivariate normal distributions but with unequal variance-covariance matrices.

P3: Variables have a bi-exponential distribution.

These three cases are chosen to provide a wide range of problem characteristics. As shown before, the optimal discriminant function for P1 is linear. It is a quadratic function for P2. The joint density function for P3 cannot be specified [20]. Therefore, the appropriate classifier may be nonparametric.

Aside from differences in distribution and variance-covariance matrix, the three types of problems are made to be as much alike as possible. For example, variables of the same group have the same means across the three cases and the same correlation. Recall that μ_1 and μ_2 are vectors of variable means in group 1 and group 2, respectively. Further, Σ_1 and Σ_2 are the respective variance-covariance matrices. In this study, these parameters are chosen as follows:

$$\mu_1 = \begin{pmatrix} 5 \\ 5 \end{pmatrix}, \mu_2 = \begin{pmatrix} 15 \\ 5 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 25.0 & 7.5 \\ 7.5 & 25.0 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 225.0 & 22.5 \\ 22.5 & 25.0 \end{pmatrix}.$$

For P1, mean vectors are as specified and the variance-covariance matrix for group 2 is set to Σ_1 . For P2 and P3, the mean vectors are the same and the variance-covariance matrices are also the same. In both matrices, the off-diagonal elements are chosen so that the correlation between variables is .3. This coefficient of correlation is arbitrary. It is large enough to create irregular overlaps among the generated patterns and it is small enough that sample covariance matrices do not become ill-conditioned.

The bi-exponential distribution belongs to the bi-gamma distribution whose joint density function cannot be specified [20]. What can be said is that each variable has an exponential distribution and the correlation between variables is known. Bi-exponential is the only non-normal bivariate distribution where random variables can be generated with controlled correlation [20]. Specifically, let x_{ij} denote variable i in group j with mean μ_{ij} , then the marginal density function is

$$f(x_{ij}) = \begin{cases} \mu_{ij}^{-1} e^{-x_{ij}/\mu_{ij}} & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Since the mean of an exponential distribution is equal to the standard deviation, it is not possible to generate exponential distributions with unequal means and equal variances. This explains the choice of the diagonal values for matrices Σ_1 and Σ_2 .

For each distribution, a test sample of 600 observations are generated using algorithms shown in Law and Kelton [20]. As mentioned before, each test set serves as the surrogate for the respective population. Training samples of various sizes are also generated from each distribution.

Examples of these three types of problems are shown in Figures 1-3. The hollow squares represent points from group 1 and the pluses represent points from group 2. Each graph shows 30 randomly generated points from each group. In P1 both groups are slanted because of the covariance between variables. In Figures 2 and 3, it can be seen that group 2 encloses group 1 because of the larger variance and covariance terms of group 2.

Sample size is the second factor in our experimental design. Three levels of training sample sizes are selected: 30 (S1), 60 (S2), and 90 (S3). These are the total observations, equally split between the two groups. Sample size 30 is reasonably small for a two-group classification problem. In a previous study [37], we found that neural classification results stabilize when sample size exceeds 90. Thirty replications are taken for each combination of sample size and problem type.

Two computer simulated experiments are set up. The first experiment is to help investigate the behavior of the neural network with respect to the effects of network architecture and training sample size. The second is to compare the performance of the neural network classifier with discriminant analysis.

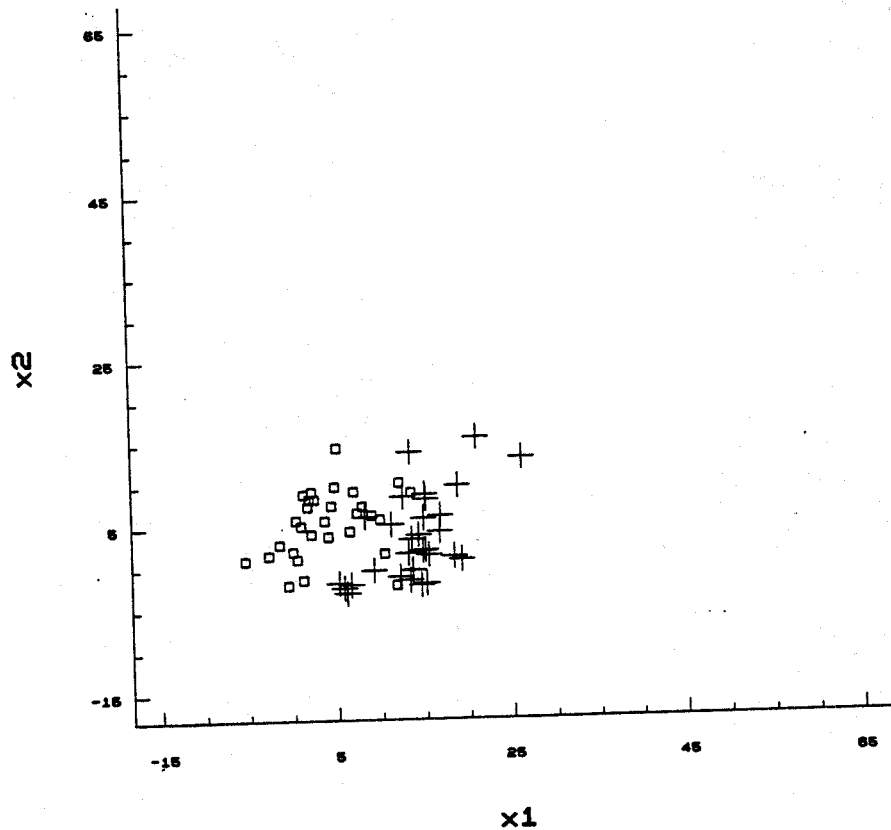
EXPERIMENT 1

Design

The objective for this experiment is to determine the appropriate architecture for neural network classifiers. The networks have three layers and are fully connected (i.e., there is an arc from each node to every node in the next higher layer). Since two-group two-variable classification problems are used as experimental subjects, each network has two input nodes and one output node. The number of hidden nodes varies. As that number increases, more arcs are introduced into the network and hence more weights can be used to find a good fit. As a result, the classification rate in the training sample will improve. On the other hand, as in multivariate regression, when more terms are included in the model, the degree of freedom for the sum of square errors is decreased, resulting in possibly less power for generalization. Thus, it is not certain that the classification rate in the test sample will be higher due to a larger number of hidden nodes. The number of hidden nodes is set at 3 (H1), 5 (H2), and 7 (H3) for this experiment. In neural network literature, the popular choice for the number of hidden nodes is $2n+1$ with n being the number of input variables. Since n is two in this study, the recommended number would be five. The other two numbers are used to measure the sensitivity of this factor.

The effects will be examined with respect to the correct classification rates in the training and the test samples. A network is trained with a training sample and the classification rate is calculated based on the training sample. Then the same network is asked to classify the test set from the same problem type and the

Figure 1: Scatterplot of a sample of P1.

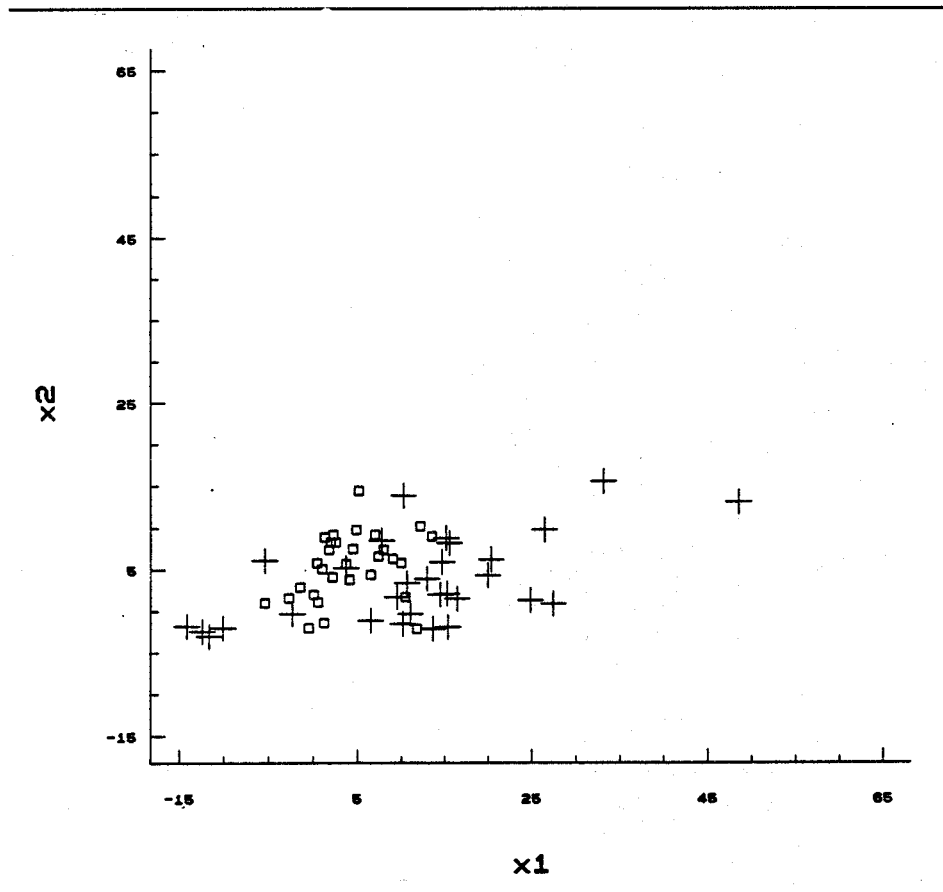


classification rate is calculated. Each pattern, in both training and test samples, has a target value of 0 if the pattern comes from group 1, and a target value of 1 if it comes from group 2. If the activation value at the output node is within .5 of the target value, then the pattern is considered correctly classified.

Results

The average classification rates at the cell level are shown in Table 1 where HN denotes hidden nodes. Each value represents the average of 30 replications. The average classification rate is 91.22 percent for training samples of size 30 of problem type P1. The average classification rate on the test set of the 30 networks trained with samples of size 30 from problem P1 is 78.33 percent. From this table, it is clear that the main determining factor in neural network performance is problem type. This, however, is understandable and primarily due to the distributions and the expected error rate shown in (2). While the expected classification rate is difficult to compute exactly, we speculate that it is about 85 percent for P1, 79 percent for P2, and 81 percent for P3.

Figure 2: Scatterplot of a sample of P2.



The effects of network architecture and sample size are fairly consistent. Across problem types and network architectures, as sample size increases, the average classification rate decreases in training samples and increases in test samples. Both classification rates seem to converge to the expected rate for each problem type (Figure 4). The rate of increase or decrease depends on problem type. For problem P1, the training accuracy is 92.41 percent for size 30 and 88.48 percent for size 90, a decrease of 3.93 percent, but the increase in test set accuracy is only 2.47 percent. For P2, the decrease in training accuracy is 3.35 percent as sample size goes from 30 to 90, and the increase in test accuracy is 5.06 percent.

Across problem types and sample sizes, as the number of hidden nodes increases, the classification rate increases in the training samples and decreases in the test samples. As in sample size, the rates seem to converge to the expected rates (Figure 5). However, the increase in training set accuracy is higher than the decrease in test set accuracy. For example, for problem type P1 the overall increase in accuracy for the training set is $91.34 - 88.82 = 2.52$ percent and the decrease in the test set accuracy is only $80.13 - 79.63 = .50$ percent. For P2, the corresponding changes are 5.93 percent and 1.02 percent.

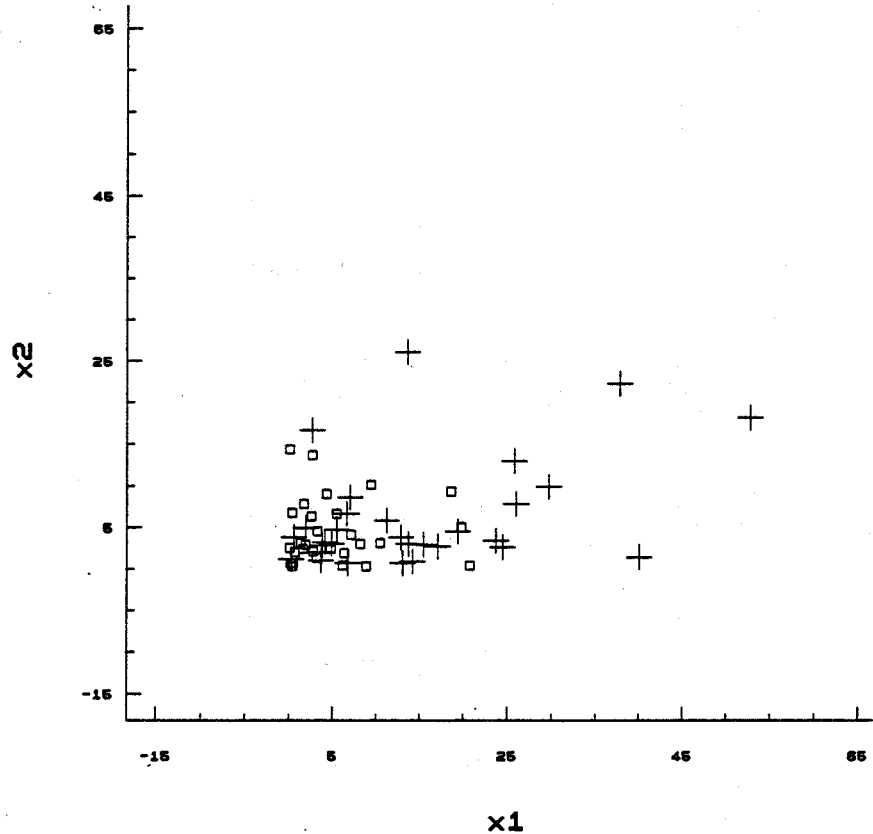
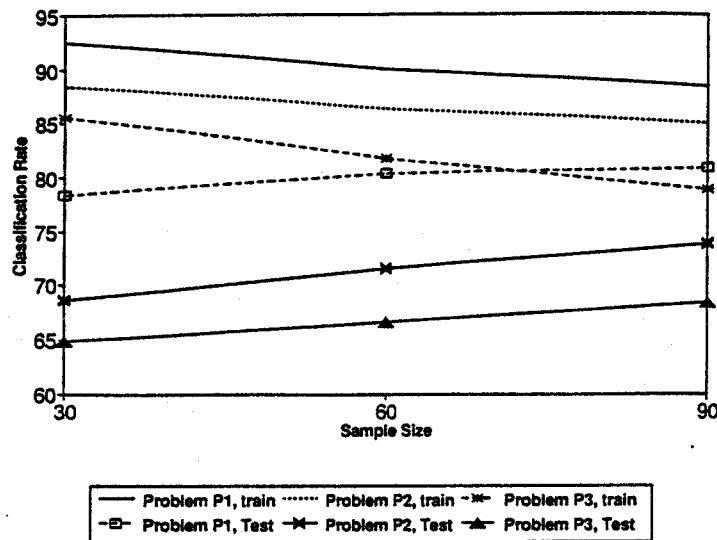
Figure 3: Scatterplot of a sample of P3.

Table 2 contains the summaries of Table 1. A three-way analysis of variance is used to study the effects of problem type, network architecture, and size of training samples on the correct classification rate achieved in the training and test samples. The main effects of all three factors for both dependent measures are highly significant with p -values less than .01, while the interactions terms are found to be insignificant for all practical purposes. In the training samples, an average of 90.26 percent is attained for P1, 86.59 percent for P2, and 82.05 percent for P1. The means of the levels are significantly different (at .05 level) from each other in both training and test samples. A steeper rate of decrease is observed in the averages for the test samples, from 79.84 percent in P1 to 66.62 percent in P3.

The highest classification rate in the training sample is found to be associated with the smallest sample size, 30. As the training sample size increases from 30 to 60, and to 90, classification rate decreases in the training samples from a high of 88.80 percent to 85.99 percent and 84.11 percent, respectively. Yet, a reverse pattern is observed in the test samples. Differences among the three level means are significant for both training and test samples. The phenomenon that training

Table 1: Average classification rates from Experiment 1.

| Problem Property | Network Architecture | Sample Size 30 | | Sample Size 60 | | Sample Size 90 | | Average | |
|--------------------------|----------------------|----------------|----------|----------------|----------|----------------|----------|--------------|----------|
| | | Training (%) | Test (%) | Training (%) | Test (%) | Training (%) | Test (%) | Training (%) | Test (%) |
| Normal, Equal Variance | HN=3 | 91.22 | 78.33 | 88.39 | 80.88 | 86.85 | 81.17 | 88.82 | 80.13 |
| | HN=5 | 92.33 | 78.35 | 89.94 | 80.00 | 89.04 | 80.91 | 90.44 | 79.75 |
| | HN=7 | 93.67 | 78.38 | 91.39 | 80.19 | 89.56 | 80.38 | 91.34 | 79.63 |
| | Average | 92.41 | 78.35 | 89.91 | 80.36 | 88.48 | 80.82 | | |
| Normal, Unequal Variance | HN=3 | 84.22 | 68.92 | 82.95 | 71.78 | 82.67 | 74.49 | 83.28 | 71.73 |
| | HN=5 | 89.33 | 69.13 | 86.67 | 71.78 | 85.85 | 73.59 | 87.28 | 71.50 |
| | HN=7 | 91.67 | 67.95 | 89.28 | 71.07 | 86.67 | 73.12 | 89.21 | 70.71 |
| | Average | 88.41 | 68.67 | 86.30 | 71.54 | 85.06 | 73.73 | | |
| Biexponential | HN=3 | 81.89 | 65.93 | 79.33 | 67.18 | 76.85 | 69.71 | 79.36 | 67.61 |
| | HN=5 | 85.44 | 64.69 | 82.83 | 66.51 | 78.56 | 68.54 | 82.26 | 66.58 |
| | HN=7 | 89.44 | 63.92 | 83.17 | 65.98 | 81.00 | 67.14 | 84.54 | 65.68 |
| | Average | 85.59 | 64.85 | 81.78 | 66.56 | 78.80 | 68.46 | | |

Figure 4: Neural network classification rates by sample size.

accuracy decreases and test accuracy increases as sample size increases may be explained as follows. The smaller the sample, the easier it is for neural networks to adjust themselves to the idiosyncrasies of the individual observations in the sample; therefore, the percentage of correct classification goes up. On the other hand, the separation curve may be so complicated that it is very different from the optimal one; hence the test accuracy decreases.

The classification rate in the training samples increases from 83.82 percent to 88.43 percent as the number of hidden nodes goes up from three to seven. Tukey's pairwise comparison identifies significant differences (at 5 percent significance level) among all three means. On the other hand, with larger number of hidden nodes, the performance of the network on the test sets goes from 73.15 percent down to 72.01 percent. A significant difference between the means associated with H1 and H3 is detected by the Tukey's comparison procedure. The increase in training set accuracy is much higher than the decrease in test set accuracy.

In neural network literature, there is a phenomenon called over-training which refers to the situation when the objective function in network training is reduced too much and the network loses its ability to generalize. We do not agree with this concept as we do not think training is the culprit. We think the reason a network loses its ability to generalize is that it is too large for the training set. This is similar to any statistical modeling exercise. For the same data set, if one increases the number of independent variables or the number of terms (interactions), the coefficient of multiple determination increases. But it does not mean that the model's ability to generalize increases. As we increase the number of hidden nodes of a neural network, we increase the number of variables and terms. As a result the network is better able to configure itself for the individual observations in the training set.