Pergamon

Mathl. Comput. Modelling Vol. 21, No. 1/2, pp. 243–258, 1995

0895-7177/95 $9.50 + 0.00

0895-7177(94)00207-X

Performance Measures, Consistency, and Power for Artificial Neural Network Models*

J. M. TWOMEY AND A. E. SMITH†
Department of Industrial Engineering, University of Pittsburgh
1031 Benedum Hall, Pittsburgh, PA 15261, U.S.A.
aesmith@engrng.pitt.edu

Abstract—Model building in artificial neural networks (ANN) refers to selecting the "optimal" network architecture, network topology, data representation, training algorithm, training parameters, and terminating criteria, such that some desired level of performance is achieved. Validation, a critical aspect of any model construction, is based upon some specified ANN performance measure of data that was not used in model construction. In addition to trained ANN validation, this performance measure is often used to evaluate the superiority of network architecture, learning algorithm, or application of a neural network. This paper investigates the three most frequently reported performance measures for pattern classification networks: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and percent good classification. First the inconsistency of the three metrics for selecting the "better" network is examined empirically. An analysis of error histograms is shown to be an effective means for investigating and resolving inconsistent network performance measures. Second, the focus of this paper is on percent good classification, the most often used measure of performance for classification networks. This measure is satisfactory if no particular importance is given to any single class, however, if one class is deemed more serious than others, percent good classification will mask the individual class components. This deficiency is resolved through a neural network analogy to the statistical concept of power. It is shown that power as a neural network performance metric is tuneable, and is a more descriptive measure than percent correct for evaluating and predicting the "goodness" of a network.

1. INTRODUCTION

The most popular, and perhaps the most successful, application of the backpropagation trained artificial neural networks (ANN) is pattern classification. The majority of research in ANN applications of pattern classification focuses on model design. Pattern classification ANN's have been applied in a wide variety of disciplines including medicine, engineering, economics, and biology, and the domain of the modeler determines what is actually being modeled with the ANN. The cognitive and biological communities use neural networks to model the massively parallel architecture of the brain, while the view taken by the engineering and statistical communities is that neural networks are inference models much like nonparametric regression [1–3]. In either case, model building in neural networks refers to selecting the "optimal" network architecture, network topology, data representation, training algorithm, training parameters, and terminating criteria, such that some desired level of performance is achieved. While building a network is viewed by some as a 'trial and error procedure,' there are a number of articles and books that provide theoretical approaches and heuristic guides to model construction [4–8].

*This research is supported by the Central Research Development Fund of the University of Pittsburgh.
†Author to whom all correspondence should be addressed.

Validation is a critical aspect of any model construction. Although, there does not exist a well formulated or theoretical methodology for ANN model validation, the usual practice is to base model validation upon some specified network performance measure of data that was not used in model construction (a "test set"). In addition to trained network validation, this performance measure is often used to evaluate the superiority of network architecture, learning algorithm, or application of a neural network. There are three frequently reported performance measures: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and percent good classification. Typically, for the sake of parsimony, most researchers present only one performance measure to indicate the "goodness" of a particular network's performance. There is, however, no consensus as to which measure should be reported, and thus, comparisons among techniques and results of different researchers are practically impossible.

The issues addressed in this paper are related to performance measures for classification neural networks and are presented in two parts: Sections 3 and 4. Section 3 examines the relative value of the set of commonly used metrics (RMSE, MAE, percent good classifications) to determine their consistency for selecting the "better" network for the classification problem. Empirical results of a well-established pattern classification problem demonstrates the necessity of examining all three commonly used metrics. An analysis of error histograms is shown to be an effective means for investigating and resolving inconsistent performance measures.

Percent good classification is the preferred measure of performance for classification networks since it is consistent with the task. It is a normalized measure of the number of correct classifications summed over all classes. This measure is satisfactory if no particular importance is given to any single class. However, if determining membership in one class is deemed more important or serious than others, percent good classification could mask the individual class components. In Section 4, this issue is addressed through a neural network analogy to the statistical concept of power. It is shown that power as a neural network performance metric can be adjusted by the relative importance of detecting the target class. In addition, power is demonstrated to be a more descriptive measure than percent correct for evaluating and predicting the "goodness" of a network. A brief outline of power as a statistical concept is included in Section 4 to familiarize the reader with its definition, interpretation, and the effects of its related parameters. The statistical concepts introduced are then adapted to the area of ANN models. Several neural network experiments are presented to empirically support parallels drawn between the power of a statistic and the power of an ANN trained on the same pattern classification problem. Finally, the utility of "power" as a general neural network concept is discussed.

Included in this introduction is a brief description of a typical feedforward network and the backpropagation training algorithm for those readers not familiar with the subject. The reader may refer to [7,8] for a more complete explanation.

2. OVERVIEW OF FEEDFORWARD NEURAL NETWORK ARCHITECTURE AND BACKPROPAGATION TRAINING

Artificial neural networks are massively parallel computing mechanisms. Neural networks store model knowledge in their many interconnecting weights, which during training move from an initially random state to a stable state. These variable weights hierarchically connect nodes both in parallel and in sequence. The entire mechanism processes vector input through the network of nodes and weights, arriving at a vector output. There is typically an input layer, an output layer and one or more hidden layers between the former and the latter. The networks utilized in this work are feedforward; i.e., neuron activation flows from the input layer through the hidden layer(s), to the output layer. Recurrent networks, on the other hand (not used in this work), permit the activation to flow laterally or back to hidden and/or input layers.

The fundamental building block of every neural network is a neuron with multiple input connections and a single output value. The signal flow of the neuron's inputs, x_i , and output, o, are considered to be unidirectional.

How the network processes the data is now described. The output signal for a single neuron is given by the following relationship

$$o = f\left(\sum_{i=1}^{m} w_i\, x_i\right) = f\left(\mathbf{w}^{\mathbf{t}}\mathbf{x}\right), \tag{1}$$

where $\mathbf{w}$ is the weight vector

$$\mathbf{w} = [w_1, w_2, \ldots, w_m]^t, \tag{2}$$

and $\mathbf{x}$ is the input vector

$$\mathbf{x} = [x_1, x_2, \ldots, x_m]^t. \tag{3}$$

The function $f(\mathbf{w}^{\mathbf{t}}\mathbf{x})$ is the transfer or activation function. The net input into each neuron is passed to subsequent neurons as activation; where net is the scalar product computation

$$\text{net} = \mathbf{w}^{\mathbf{t}}\mathbf{x}. \tag{4}$$

There are several suitable continuous valued activation functions, however the most typical—sigmoidal function—was used in this work

$$f(\text{net}) = \frac{1}{1 + \exp^{(-\text{net})}}. \tag{5}$$

This unipolar activation function produces output between 0 and 1. Each layer is constructed of one or more neurons.

Before the model can be used to predict classification, the weight matrix $\mathbf{W}$ consisting of all the weight vectors $\mathbf{w}$, is modified from an initial random state to a fixed equilibrium state. Weight modification or training, is the process of finding the 'best' $\mathbf{W}$. The most well known of the supervised training algorithms—backpropagation—was chosen for this work. The backpropagation algorithm adjusts initially randomized weights during training according to the steepest gradient along the error surface [9,10]. Weights are adjusted in proportion to their contribution to the output by recycling the squared error signal back through the layers of weights.

For a supervised network, data consists of n training pairs, $\{\mathbf{z_1}, \mathbf{d_1}, \mathbf{z_2}, \mathbf{d_2}, \ldots, \mathbf{z_n}, \mathbf{d_n}\}$ where:

$\mathbf{z_n}$ is the normalized input vector,

$\mathbf{d_n}$ is the normalized desired output vector.

The goal in training is to minimize $\lambda(w)$; where for backpropagation training, the function, is the squared error over all n pattern vectors

$$\lambda(w) = \sum_{p=1}^{n} \sum_{k=1}^{K} \left(d_{kp} - o_{kp}\right)^2. \tag{6}$$

The size of vector $\mathbf{z}$ is not necessarily the same size of vector $\mathbf{d}$. For the sigmoidal transfer function, error signals δ_k are calculated for the output neurons $k = 1$ to K by equation (7)

$$\delta_k = (d_k - o_k)\,(1 - o_k)\; o_k \qquad \text{for } k = 1, 2, \ldots, K, \tag{7}$$

where

d_k is the desired output for node k,

o_k is the network output for node k,

$(1 - o_k)\, o_k$ 1^{st} partial derivative of $(1 - o_k)^2$ with respect to w.

The hidden layer y, neurons 1 to J error signal is the weighted sum of

$$\delta_{yj} = y_j (1 - y_j) \sum_{k=1}^{K} \delta_k w_{kj}, \qquad \text{for } j = 1, 2, \ldots, J, \tag{8}$$

where

$$y_j \leftarrow f\left(\mathbf{w}_\mathbf{j}^\mathbf{t} \mathbf{z}\right), \tag{9}$$

and $\mathbf{z}$ is the normalized input vector for the pattern vector p.

Weights between the hidden neurons 1 to J and the output neurons 1 to K are adjusted by

$$w_{kj} \leftarrow w_{kj} + \eta \delta_k y_j, \qquad \text{for } k = 1, 2, \ldots, K \quad \text{and } j = 1, 2, \ldots, J, \tag{10}$$

and weights between the input neurons 1 to I and the hidden neurons 1 to J are adjusted by

$$w_{ji} \leftarrow w_{ji} + \eta \delta_{yj} z_j, \qquad \text{for } j = 1, 2, \ldots, J \quad \text{and } i = 1, 2, \ldots, I, \tag{11}$$

where η is the step size, or rate at which the gradient descent occurs.

Training proceeds by presenting each normalized input vector, $\mathbf{z}$, and its corresponding output target vector $\mathbf{d}$, until each $\lambda(w)$ is smaller than the maximum allowable error $\lambda(w)_{\max}$. Presentation of the entire set of training vectors to the network is called an epoch. Sometimes training is halted after a specified number of epochs if $\lambda(w)_{\max}$ is not reached.

The validation of a trained neural network model is conducted similarly to, but not as rigorously as, statistical regression [11,12], where the "goodness" of the network is judged on how well the network predicts classification of data it has never seen before; i.e., how well the network generalizes to a test set of data not in the training set. Both the training and the test sets should be sampled from i.i.d. (independently and identically distributed) populations.

3. EXAMINATION OF PERFORMANCE MEASURES AND THEIR CONSISTENCY

As stated in the Introduction, performance measures are used to select the "better" network model in terms of architecture, training algorithm, training parameters, etc. This section examines the inconsistencies of the three most commonly used performance measures for selecting the "best" network in terms of training parameter-tolerance.

3.1. Definition of Common Performance Measures

The definitions for RMSE, MAE and percent good classification are given in equations (12), (13) and (14).

RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{p=1}^{n} (d_p - o_p)^2}{n}}. \tag{12}$$

MAE is defined as:

$$MAE = \frac{\sum_{p=1}^{n} |d_i - o_i|}{n}. \tag{13}$$

In neural computing, rules are employed for classifying a correct response to a given test input vector. There are two commonly used interpretive schedules for binary output:

for interpretive schedule I:

$$\text{if } |d_p - o_p| \leq 0.5 \text{ then } Y_p = 1, \text{ else } Y = 0,$$

for interpretive schedule II:

$$\text{if } |d_p - o_p| \leq 0.4 \text{ then } Y_p = 1, \text{ else } Y = 0.$$

Percent good classification is a measure of the number of correct classifications summed over all patterns using one of the interpretive schedules from above:

$$\text{percent good classification} = \frac{\sum_{p=1}^{n} (Y_p)}{n} \times 100. \tag{14}$$

3.2. Statistical Pattern Classification Problem Description, Network Architecture and Training

The classical statistical population classification problem was chosen because of predictable properties and because it has been well studied in the field of statistics. This problem is also familiar to the neural network community; Hush [13] used it in order to investigate the relationship of network size and the number of training set samples to network performance; Kohonen, Barna and Chrisley [14] used it in their benchmarking studies.

Two population classes of data were generated for both training and test sets: Class 0 consisted of data randomly generated from a Gaussian distribution, with a mean of 0.0 and a standard deviation of 1.0, and Class 1 was also from a Gaussian distribution, with a mean of 0.0 and a standard deviation of 2.0 (see Figure 1). These distribution parameters were selected because they pose a difficult two class classification task, and therefore, would present a significant proportion of errors. This problem is considered to be difficult because of the significant amount of overlap of the two classes (Class 1 is almost entirely enclosed in Class 0) requiring the network to form a nonlinear decision boundary. The network task was to classify a given input vector as coming from either the Class 0 distribution or the Class 1 distribution.

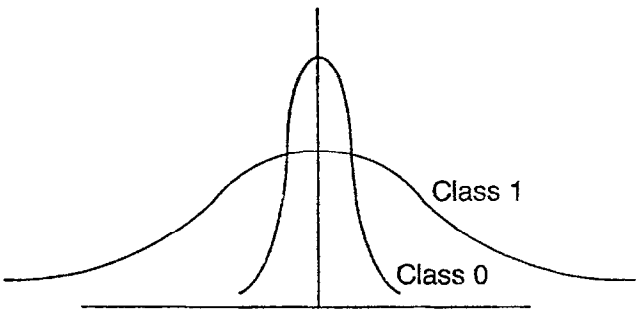

Figure 1. Depiction of first classification problem.

A training or test pair consisted of an input vector of 12 sample data points and a binary (0,1) output vector to indicate the source population for the sample input data: 0's signify Class 0, and 1's signify Class 1. An input vector dimension of 12 was chosen, after a bit of trial and error, in order to accommodate some of the difficulty associated with this task, but at the same time not so large as to impede processing time or make classifications too easy. The training set size contained 198 samples per class for a total of 396 samples. The test set consisted of 500 samples per class, for a total of 1000 vectors. The test set was large so that individual point idiosyncracies would not significantly affect the overall results.

A standard one hidden layer configuration was used since it has been shown to perform well on problems that have convex decision regions such as this one [15,16]. The smallest network that would learn the problem as well as facilitate generalization consisted of 12 nodes in the hidden layer. The assumption here and throughout this paper is that all networks are of optimal size (number of hidden nodes) [2]. All training parameters were held constant with the exception of

training tolerance. The training rate was set to 1.0 with an added smoothing factor of 0.9. The smoothing factor allows for current weight changes to be based in part on past weight changes [17]:

$$\Delta_p w_{ij} = \eta \left(\alpha \Delta_{p-1} w_{ij} (1 - \alpha) o_{pi}\right), \tag{15}$$

where: Δw_{ij} is the change in weight for input vector p, and α is the smoothing factor. All weights were initialized to the same random point on the error surface at the start of each training session.

3.3. Training Tolerance Investigation

Typically, networks are trained to decreasingly lower tolerances $\lambda(w)_{\max}$, in an attempt to achieve the "best" performing network. This assumes that lower training tolerances equates with improved performance. To examine three commonly used performance metrics (RMSE, MAE, percent good classification) several networks were trained and tested in order to measure the effects of lowering the training tolerance on performance measures for the two statistical pattern classification problem described in Section 3.2.

A total of 13 different networks were trained, each to a different training tolerance. Training tolerances ranged from 0.08 to 0.6 for normalized input between 0 and 1. During training, a response was considered correct if the output was within ± a prespecified training tolerance of the target value (0 or 1). Training iterations continued for each network until each of the 396 training samples was deemed correct according to the specified training tolerance. Note that the tolerance that was used to train the network is an upper bound to the final training tolerance of the trained network, since the maximum acceptable error is applied to the worst training vector.

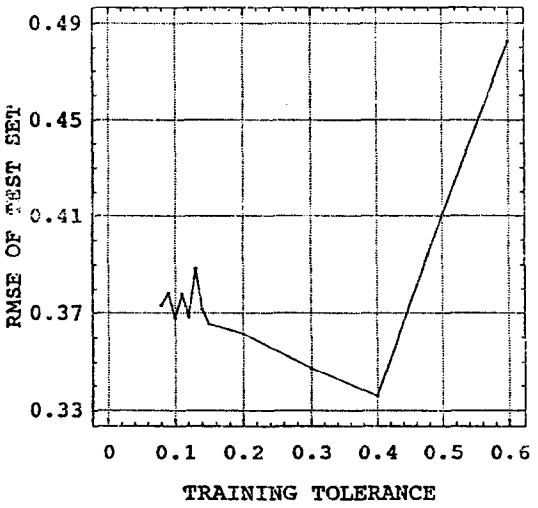

Figure 2. Test set results for Root Mean Squared Error (RMSE) vs. training tolerance.

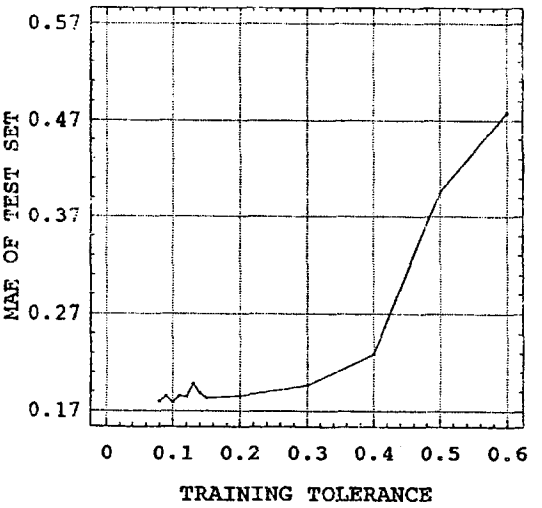

Figure 3. Test set results for Mean Absolute Error (MAE) vs. traning tolerance.

Figure 2 is a plot of Root Mean Squared Error (RMSE) vs. training tolerance for the test set of 1000. This plot indicates that the RMSE resulting from networks trained at tolerances between 0.08 and 0.13 vary to some degree about a RMSE of 0.14. The RMSE for networks trained to tolerances from 0.14 to 0.40 steadily declines to a minimum of 0.11. At training tolerances of 0.5 and 0.6 there is a sharp increase to an RMSE of 0.24. The result of the mid-range training tolerances having the smallest RMSE is not expected. The minimum RMSE is often used as a criteria for selecting the "better" neural network. Based on this criteria, the network trained to a tolerance of 0.4 would be chosen as the "better" network. This inconsistent phenomenon will be explained in Section 3.4.

Figure 3 is a plot of the test set results for Mean Absolute Error (MAE) vs. training tolerance. These results, aside from some slight variations at the lower training tolerances, are closer to what

is usually predicted: a steady decrease in MAE with a decrease in training tolerance. Criteria for selecting the "best" neural network is again, to choose the network with the smallest MAE, in this case a training tolerance of 0.1.

Figure 4 is a plot of the test set results measured in terms of percent correct (using interpretative schedules) vs. training tolerance. This plot displays two curves: curve I (dotted line) represents interpretive schedule I, and curve II (solid line) represent interpretive schedule II. Since the output for these networks are binary (0,1), interpretive schedule I classifies output less than 0.5, as coming from Class 0 and output greater than 0.5, as coming from Class 1. For interpretive schedule II, an output less than 0.4, is classified as coming from Class 0, and an output greater than 0.6, as coming from Class 1. Note that for interpretive schedule II, an output that fell between 0.4 and 0.6 is always counted as incorrect. In both methods, the network with the highest proportion of correct responses would be deemed as the "better" network. As anticipated, curve I is consistently greater than curve II since curve II is a subset of curve I and the networks themselves are identical. The "better" networks according to interpretive schedule I were trained using tolerances between 0.08 and 0.5, while training tolerances between 0.08 and 0.2 yielded the "better" networks according to interpretive schedule II.

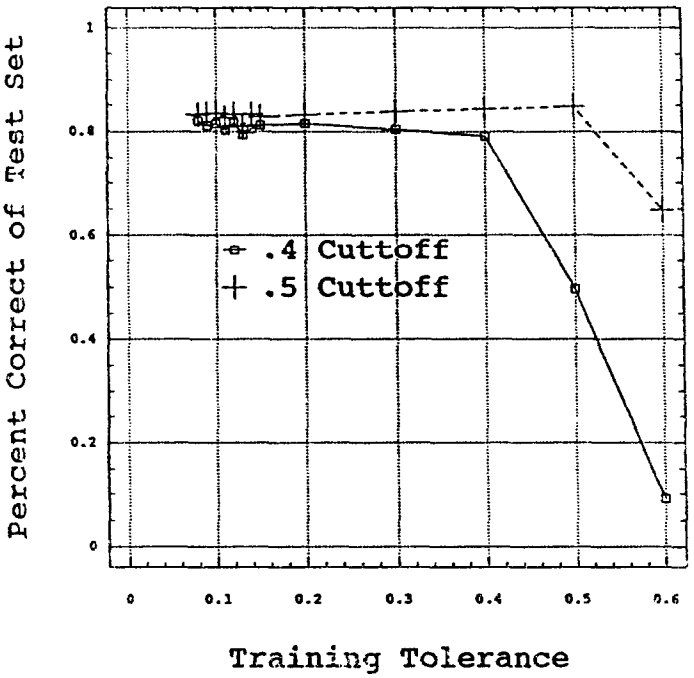

Figure 4. Test set results measured in terms of percent correct (using interpretative schedules) vs. training tolerance.

Selecting the "better" network using three of the most popular network performance criteria have yielded three different networks; RMSE criteria resulted in the network trained at 0.4, MAE criteria and percent correct (schedule II) resulted in the network trained at 0.1, and percent correct (schedule I) resulted in the network trained at 0.5. The results are conflicting and in some cases are contrary to what most practitioners believe make a good network.

3.4. Resolving Inconsistencies

In order to explore these issues, the relative frequency histograms of test set residuals or output errors, were constructed for each network. Six distributions that are representative of the thirteen networks are presented in Figure 5. Error was calculated as [desired output – network output]; $[d_i - o_i]$. The network trained at a 0.1 training tolerance exhibits a trimodal distribution of

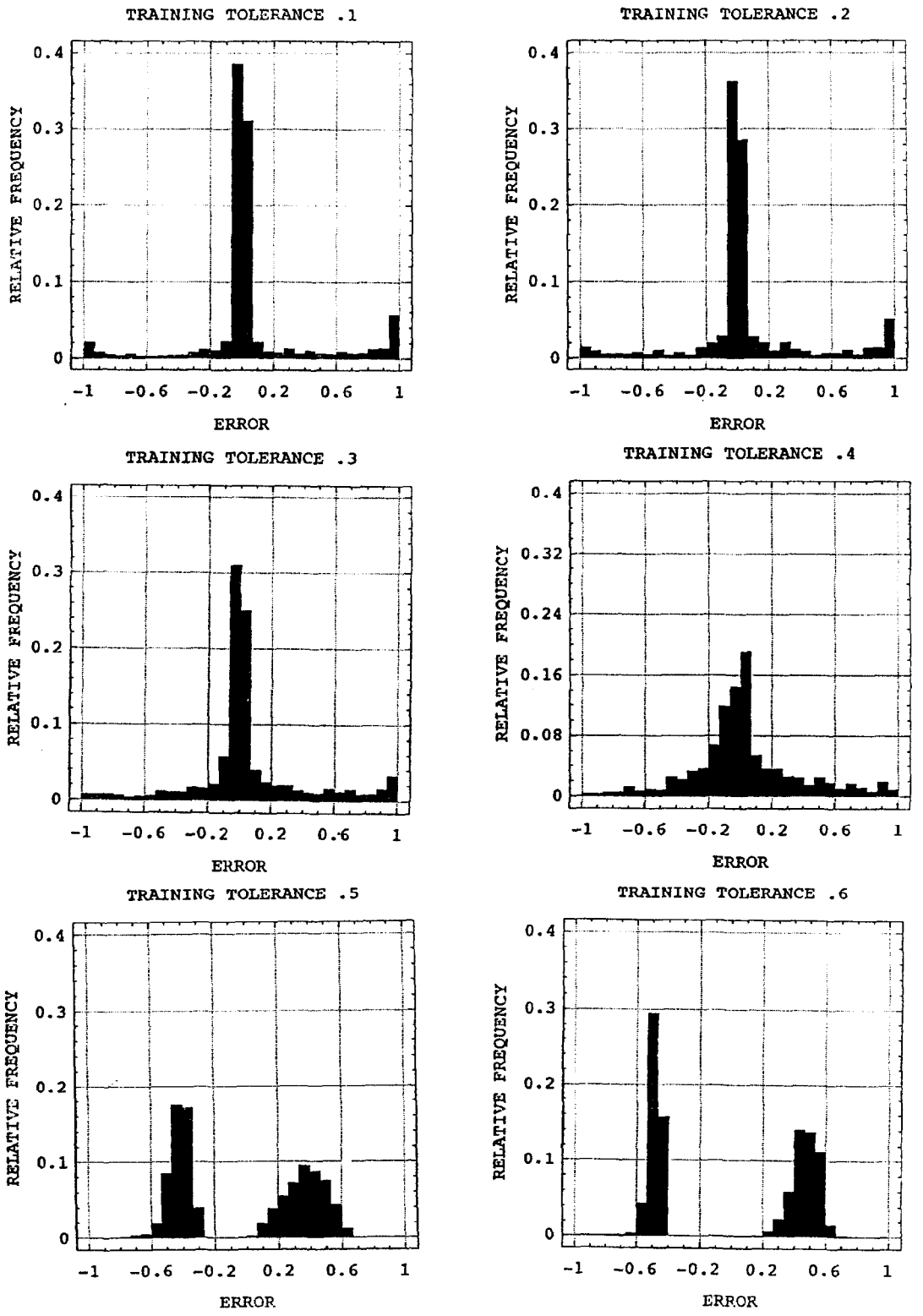

Figure 5. Relative frequency histograms of test set output errors.

errors. The large center peak indicates very small errors or output that is very close to the targeted values. The smaller endpoint spikes denote clearly incorrect responses (i.e., a 1 signal for a 0, or a 0 signal for a 1); the right hand side shows a larger proportion of misclassifications for data sampled from class 1, suggesting a harder classification task. (This is intuitively explainable

since the outer distribution covers a larger range.) There is a smaller proportion of errors in the region between the spikes, representing a few less sure classifications (i.e., a 0.35 for a 0 or a 0.65 for a 1). As the training tolerance increased, all three modes decrease until the error distribution takes on a Gaussian-like distribution shape at tolerance of 0.4. At tolerances of 0.5 and 0.6 the distributions become bimodal with an increase in distance between the modes at a tolerance of 0.6. At these large training tolerances, more vectors are classified in the less sure region because the networks have not been allowed to learn the classification model.

A careful examination of the various error distributions demonstrates how a network trained at 0.4 would have the lowest RMSE. By squaring the errors in calculating the RMSE, a greater penalty is placed on large errors (clearly incorrect responses) even if they are few in quantity. This is evident through a comparison of distributions of 0.1 and 0.4 tolerances. MAE takes large errors into account, but does not weight them more heavily. Percent correct indicated a "better" network when a larger proportion of errors fell within the prescribed range even though there are a number of less sure classifications (see histogram of tolerance 0.5).

In spite of these inconsistencies, percent good classification is the preferred performance measure because it is the most appropriate measure for the task of classification. However, in the next section the deficiencies of percent good classification measure are examined and an alternative measure is posed.

4. PERCENT GOOD AND THE NEED FOR POWER CURVES

Recently, there have been several articles in the neural network literature relating neural networks to nonparametric statistics [1,2,3]. The work presented here, expands on some of these ideas by applying the theory and concepts related to the power of a statistical test, to the power associated with the weights of a trained neural network. The aim of this section is to is to demonstrate how power as a neural network concept and power as a statistical concept possess similar properties. In addition, the superiority of power for evaluating and predicting the "goodness" of a network over percent good is illustrated. Section 4 begins with a review of the power of a statistical test.

4.1. Power as a Statistical Concept

The power (P) of a statistical test is the probability of rejecting the Null hypothesis when the Null hypothesis is false. Or, as stated by Cohen [18] "it is the probability that (the test) will result in the conclusion that a phenomenon exists." The phenomenon is typically described in terms of an assumed value for a statistical parameter, such as mean value, variance or correlation and is stated in the form of a Null hypothesis. The P of a specific statistical test can be determined from the characteristics of the statistical test and the properties of the population. The P of a statistical test is related to three parameters: significance criteria (α), sample size (n) and effect size (ES). A description of the elements of statistical hypothesis testing is important for understanding the terminology used throughout this section, and the reader is referred to any statistical text such as [19].

4.1.1. Significance criteria (α and β)

The significance criteria (α) is a basis for judging whether or not the phenomenon exits. It is the probability of committing a Type I error; i.e., the probability of wrongly rejecting the Null hypothesis, that is a false alarm. α values are typically set very low in order to reduce the probability of falsely rejecting the Null hypothesis. The consequences of reducing the probability of committing a Type I error results in an increase in the probability of committing a Type II error (β) all else being constant; i.e., failing to reject the Null hypothesis if it is false. Since $P = 1 - \beta$, P decreases with a decrease in α. In the design of a statistical test, the investigator

must weigh the seriousness of committing a Type I error vs. a Type II error. An investigator may set α to be .05, and P to equal .80, which results in β equal to .20. Type I error is, therefore four times as serious as a Type II error (.20/.05 = 4).

4.1.2. Reliability of sample results and sample size (n)

The reliability of a sample value is the closeness to which it approximates the population parameter; i.e.,

Sample	$\rightarrow$	Population
$\bar{x}$	$\rightarrow$	μ
s	$\rightarrow$	σ
n	$\rightarrow$	N

The reliability of a sample value may or may not be dependent upon the population parameters and distribution, but is always dependent upon the size of the sample, n. For a given α level, as n increases toward the size of the population, N, the chances of detecting the phenomenon increases; i.e., an increase in P. The natural question which follows is "How large a sample is needed in order to increase P to a specified value?" This question is addressed in the next section.

4.1.3. Effect Size (ES)

Effect size (ES) is the measure of the degree to which the phenomenon is present and is expressed as the difference between two population parameters or the departure of a population parameter from a constant. The larger the ES, the greater the P. Effect size is measured in terms of the given sample statistic, e.g., the difference between two means or the ratio of two standard deviations.

In statistics, P, α, ES, and n are computationally related; knowing any three defines the fourth. Therefore, questions such as the one posed above on n may be readily determined. For example, an investigator has chosen $\alpha = 0.05$, specifies $P = 0.6$, and expects an effect size of ES = 0.4. According to the methods described in [18], $n = 30$.

4.2. "Power" as a Neural Network Concept

Halbert White in his 1989 article "Learning in Artificial Neural Networks: A Statistical Perspective" [1] asserts that the weights resulting from a network trained on sampled data (training set) are a "(vector valued) statistic," much like an average or standard deviation which are based on sample data. As the sample mean is to the population mean so is the results of a set of weights from a training set of size n is to the results of a set of trained weights from a population set of size N:

$$\bar{x} \rightarrow \mu_N,$$
$$\mathbf{W_n} \rightarrow \mathbf{W_N}$$

where for backpropagation: $\mathbf{W_n}$ is a function of gradient descent and a particular training set selected, so that

$$\mathbf{W_n} = T_{L,A,I}(z_n, d_n) \tag{16}$$

n: sample size or number of patterns

T: gradient descent

L: learning algorithm

A: network architecture (# nodes, connections, layers, etc.)

I: starting point on the error surface

z_n: input vector

d_n: desired output vector.

Statistical hypotheses are formulated to make inferences about population parameters. For example, a null hypothesis concerning the mean of some characteristic of the population may be stated as:

$$Ho : \mu_N = \theta.$$

If the population, of size N, were available, μ would be known and the assertion $\mu_N = \theta$, would simply be true or false. However, cases where the entire population (N) are known, are rare. Instead, sample statistics are used to infer, or test the null hypothesis.

For a classification neural network, it is assumed that $\mathbf{W}_N$ is optimal. The Null hypothesis for a classification network model may thus be stated as:

Ho: Given $\mathbf{W}_N$, assign input vector $\mathbf{x}$ as coming from class m.

Again as in statistics, if the population were available, the class of vector input $\mathbf{x}$ would be known with certainty. Since these cases are rare, $\mathbf{w}_n$ may be used to infer the classifications made by $\mathbf{W}_N$. Having stated the Null hypothesis as such, it should not be assumed that a standard statistical test of this hypothesis may ensue. The Null hypothesis has been posed only to provide a framework in which the network analogy of power can be made.

4.2.1. Type I and Type II errors of a neural network (α and β)

White theorizes that "the analytical tools that describe the behavior of statistics generally can be used to describe the behavior of statistics specifically obtained by some neural network learning procedures" [1]. Therefore, the "P" associated with the $\mathbf{w}_n$ resulting from a trained 2-class pattern classification network may be defined as the probability of correctly concluding that for a given input (generalization), the phenomenon (target Class 1) exists or

$$P = p(\text{recognizing Class 1} \mid \text{Class 1}). \tag{17}$$

In addition, for a 2-class problem, Type I and Type II errors may be defined as

$$\begin{aligned} &\text{Type I error } (\alpha) = p(\text{Class 1} \mid \text{Class 0}) \qquad \text{and,} \\ &\text{Type II error } (\beta) = p(\text{Class 0} \mid \text{Class 1}). \end{aligned} \tag{18}$$

In statistical models and in neural network models, the objective is to minimize the sum of these relative errors. In neural networks, the rules used for classifying a correct response to a given test input vector, referred to in Section 3.1, may not meet the desired objective, as shown below in Section 4.2.2. In addition, networks are generally trained with the implicit, and perhaps the unrecognized, assumption that $\alpha = \beta$.

To minimize the total sum of α and β, while enabling the explicit consideration of the relative seriousness of committing each type of error, a classification strategy taken from statistical Discriminant Function analysis [11] is proposed: locate the optimal boundary which discriminates between the two classes.

That is, find a cutoff point that:

$$\min \sum_{i=1}^{2} \text{Type } i \text{ error}, \tag{19}$$

where the cutoff point is defined as:

$$\begin{aligned} &\text{output value} \leq \text{cutoff} \qquad \text{classify as 0;} \\ &\text{output value} > \text{cutoff} \qquad \text{classify as 1.} \end{aligned}$$

In the following section, experimental data is presented to illustrate the parallels drawn between power as a statistical concept and power as a neural network concept.

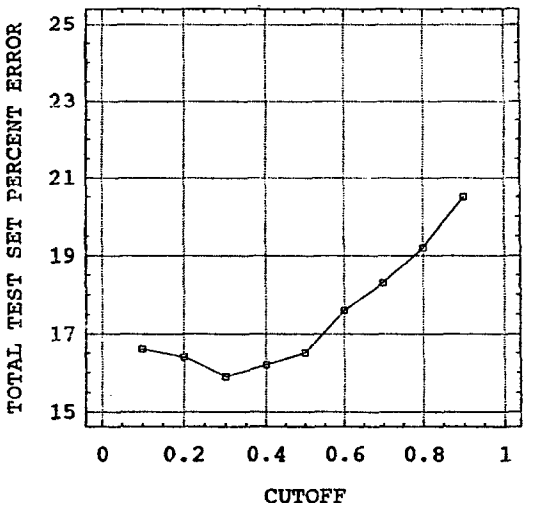

Figure 6. Minimization of sum of Type I and Type II errors.

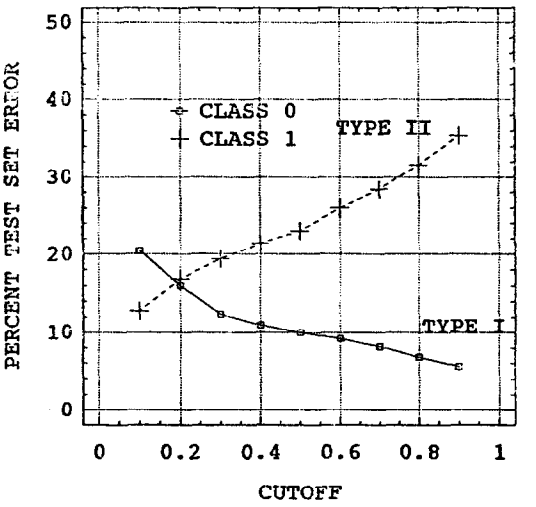

Figure 7. The effects of changing cutoff points on relative errors.

4.2.2. Type I (α) and Type II (β) errors results

A single network was selected from the set of networks described in Section 3.2 (network trained to a tolerance of 0.1), in order to illustrate the proposed classification scheme (19). Figure 6 is a plot of total test set percent error vs. cutoff. Total percent error is equal to the sum of Type I and Type II errors. The plot indicates that the minimal error occurs at cutoff equal to 0.3.

To demonstrate the relationship of α to P, and to examine the effects of weighing the relative seriousness of one error type more heavily than the other, a plot of percent test set error vs. cutoff is shown in Figure 7. The solid line represents percent incorrect classifications of Class 0 or Type I error (α). The dotted line represents percent incorrect classifications of Class 1 or Type II error (β); the complement is P. The plot indicates that as α decreases, so does P. The lines intersect at cutoff 0.2, where percent error is 0.17 for both error types. The ratio or relative seriousness of Type II error to Type I error is 1.0. Moving the cutoff point from 0.2 to 0.1 decreases the ratio of a Type II error to Type I error from 1.0 to 0.6. Moving the cutoff point in the opposite direction toward a cutoff of 0.9 increases the ratio from 1.0 to 6.3, where the chances of committing a Type II error is more than 6 times as great as committing a Type I error; i.e., at 0.9 cutoff a Type I error is much less likely to occur than a Type II error. At cutoff 0.3, where the minimum sum of the two errors occurs, the ratio of Type II to Type I errors is 1.5. For

classification schedule 2, where the cutoff occurs at 0.5, it is incorrectly assumed that both type of errors are given equal importance. In fact there are more than two times as many Type II errors as Type I errors at a cutoff of 0.5.

This plot illustrates how the cutoff point of the classification schedule can be adjusted to reflect the relative seriousness of Type I vs. Type II errors for a given neural application. In fact, it demonstrates that power curves must be examined to ascertain the relative proportion of classification errors to be expected for the model under consideration. Reliance upon an implicit assumption of $\alpha = \beta$ will usually prove false when the actual trained network and its power characteristics are investigated.

4.2.3 Training set size (n) and biased training set results

As indicated earlier, sample data is obtained due to the infeasibility of collecting the entire population of data when building a model. The goal in sampling for training purposes is to obtain a representative sample that will produce a reliable estimate of $\mathbf{W}_N$. To examine the relationship between training set size, n, and P, seven individual networks were trained on the classification problem described in Section 3.2. Each of the seven networks were trained on a different training set size, ranging from 60 to 400 samples per class. The size of the hidden layer was increased to 25 in order to accommodate the larger sample sizes. Not all networks converged to the training tolerance of 0.1.

Figure 8 is a plot of percent test set correct vs. number of training samples per class. The plot parallels the results found in statistics, in that an increase in P (Class 1 percent correct) is associated with a nonlinear increase in training set size. The addition of 340 samples per class (from 60 to 400) increases P from 72 to 88 percent correct. Since Class 0 is an "easier" classification task, it reaches maximum percent correct $[1 - \alpha]$ (90 percent correct) at $n = 60$ per class.

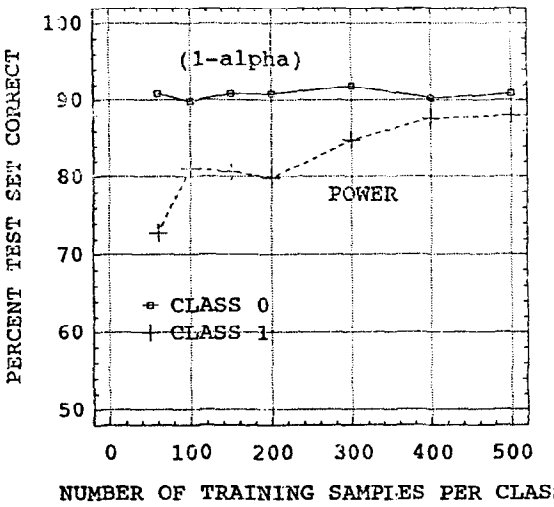

Figure 8. The relationship of increased training set sizes and power.

Not discussed in Section 4.1.2. is the assumption that the data from which the statistic is derived be a representative sample of the population, i.e., an unbiased sample. Large sample sizes can diminish the amount of bias, but again large sample sizes are not easily acquired, and an investigator may have to rely upon sampling techniques to obtain a representative sample of the population. A representative training set for an ANN 2-class classification problem is typically comprised of both classes in equal proportions. To study the effects of nonrepresentative or biased training sets on P, nineteen individual networks were trained for the same classification problem. Each network was trained on a training set size of $n = 200$ with a different proportion of Class 1

samples. Proportion of Class 1 samples ranged from 0.1 to 0.9; where 0.5 is unbiased. The test set contained equal proportions of both classes. Figure 9 is a plot of percent test set correct vs. Class 1 proportion of training set; the solid line represents Class 0 $(1 - \alpha)$ and the dotted line Class 1 (P). The plot shows that an increase in proportion of Class 1 results in an increase of percent correct classification of Class 1 (P) and a decrease in percent correct classification of Class 0. Figure 9 exhibits a well known neural network attribute, that of performance robustness. Classification accuracy is not severely impaired despite biased proportions ranging from 0.3 to 0.8.

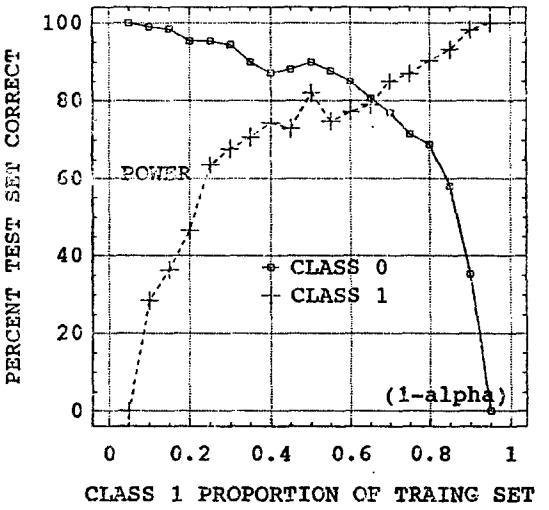

Figure 9. The relationship of biased training sets and power.

4.2.4. Effect Size (ES) results

As indicated in Section 3.1.3, ES can be described as "the degree to which the phenomenon exists" [18] and is measured according to the relevant parameters which describes the phenomenon. The same approach is applied to a single network trained on a somewhat different statistical pattern classification problem (see Figure 10) which clearly demonstrates effect size. Again two classes of data were randomly generated for both training and test sets. Class 0 training set consisted of 150 samples from a Gaussian (0.0, 1.0) distribution. Class 1 training set also consisted of 150 samples, but were generated from 15 different Gaussian distributions having standard deviations equal to 1.0 and means ranging from 0.15 to 4.0. Fifteen test sets from the same 15 Class 1 distributions were generated; one test set per distribution, each containing 500 samples. One test set of 500 samples were generated from Class 0. The network was trained to a tolerance of 0.1. Network architecture and all other training parameters were as described in Section 3.2.

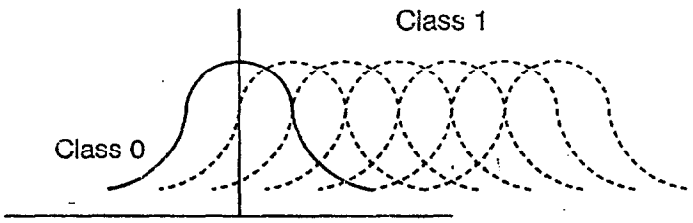

Figure 10. Depiction of second classification problem.

ES was then measured in terms of the distance between the μ from which the classes were sampled. More specifically, it is the number of standard deviations between the μ of Class 0 (0.0) and the μ of Class 1 (0.15 to 4.0). For pattern classification problems, ES is related to "overlap." Figure 11 is a plot of percent test set correct vs. ES. The plot shows a steady increase in P from

an ES of 0.15 to 1.5. This result is not surprising here, or in statistics, since detecting a difference between populations becomes less difficult as the separability between those populations becomes greater, or overlap becomes smaller, and where the easiest case involves no pattern overlap. Also shown in Figure 11 is Class 0 (depicted as stars) located slightly above average Class 1 percent correct. The intersection of the three lines at ES = 0.8 indicates the classification boundary between Class 0 and Class 1 formed by the trained neural network.

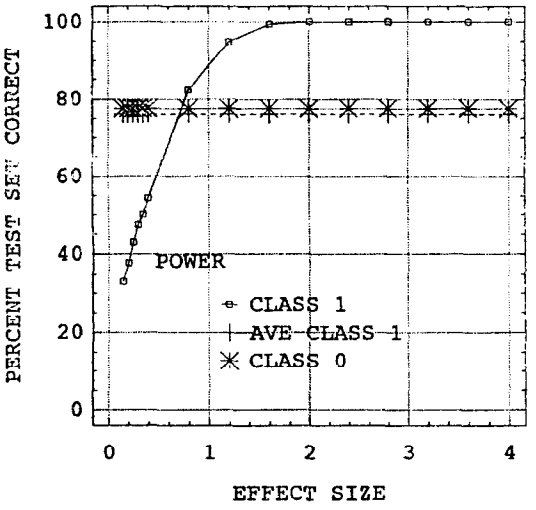

Figure 11. The relationship of increased effect sizes and power.

4.3. The Value of Power (P) as a Classification Network Performance Measure

P, used as a statistical term, allows an investigator to convey, in one word, the strength or effectiveness of a given test. By inserting 'trained ANN' before 'test,' the same assertions may be made about P as a term used in the area of neural networks. P, as it has been defined for a 2-class ANN classification problem, is a more descriptive measure than RMSE, MAE, and percent correct; i.e., it is the probability that a "specific targeted" class will be correctly identified. For example, consider the identification of cancer from an x-ray as a pattern classification problem whose target class (Class 1) is cancer. P is a measure of how effective the ANN is in detecting cancer. Associated with P are measures of Type I and Type II errors. These measures not only incorporate more information, but as it has been demonstrated, may be calibrated to reflect the relative seriousness of committing either type of error. For the x-ray example: the cutoff point may be located in order to be very certain of detecting cancer (reduce the probability of committing a Type II error) at the expense of over detection (increasing the probability of a Type I error). While the results presented here are limited specifically to 2-class classification problems, it is possible to extend this work to multiple classes.

5. CONCLUSIONS

The work presented in this paper addresses issues related to performance measures for pattern classification neural networks models. It was shown in Section 3 that the most widely used network performance criteria may give conflicting results when they are used to determine the "better" networks. How often this occurs and under what circumstances is not known since most papers report only one performance measure. It is recommended that researchers and practitioners examine at a minimum RMSE, MAE, and percent good. If a conflict does occur, the error, or residual, distributions should be plotted to discover the relative number of clear misclassifications to "don't know" classifications. The distribution of errors is an area of neural networks that should be more fully explored. In regression, it is known that a significant F-value or a large R^2 value may be misleading as to the "goodness-of fit" of the regression line, and therefore,

an examination of the residuals is required. In regression a "good-fit" can be demonstrated by a plot of the residuals versus the predicted values being scattered randomly about 0. A pattern in the residuals would indicate that the model is not accounting for all of the variability in the data. The development of a similar methodology for ANN is an area for future work.

In Section 4, it was shown that the P of a specific statistical test is related to the characteristics of the statistical test and the properties of the population. This research has established that there is an ANN analogy to the statistical concept of power. The empirical results have shown that the statistical correlates of P (α and β, n, ES) are also correlated with the P of a trained neural network. In statistics, the real value of P lies in its relationship to α, n, and ES and that knowing any three, the fourth may be computed. It was demonstrated that the P of an ANN parallels those relationships. Further research in this area could result in a methodology for predicting an interval containing P, when n, E S, and α are known for an ANN, or calculating n for a given ES and a prespecified α and β.

REFERENCES

1. H. White, Learning in artificial neural networks: A statistical perspective, *Neural Computation* **1**, 425–464 (1989).
2. H. White, Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings, *Neural Networks* **3**, 535–549 (1990).
3. S. Geman, E. Bienenstock and R. Doursat, Neural networks and the bias/variance dilemma, *Neural Computation* **4**, 1–58 (1992).
4. B. Horn and D. Hush, On the optimality of the sigmoid perceptron, In *Proceedings IEEE International Joint Conference on Neural Networks*, pp. I-269–I-272, (1990).
5. E.B. Baum and D. Haussler, What size nets give valid generalization?, *Neural Computation* **1**, 151–160 (1989).
6. D. Baily and D. Thompson, How to develop neural-network applications, *AI Expert* **5**, 38–47 (1990).
7. J.M. Zurada, *Introduction to Artificial Neural Networks*, West Publishing Co., New York, (1992).
8. J.A. Freeman and D.M. Skapura, *Neural Networks: Algorithms, Applications and Programming Techniques*, Addison-Wesley, Reading, MA, (1991).
9. D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning internal representations by error propagation, In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume 1, (Edited by D.E. Rumelhart and J.L. McClelland), MIT Press, Cambridge, MA, (1986).
10. P.J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph.D. Dissertation in Statistics, Harvard University, (August 1974).
11. N. Draper and H. Smith, *Applied Regression Analysis*, 2nd Edition, John Wiley & Sons, New York, (1981).
12. R.D. Snee, Validation of regression models: Methods and examples, *Technometrics* **19** (4), 415–428 (1977).
13. D.R. Hush, Classification with neural networks: A performance analysis, In *Proceedings of the IEEE International Conference on Systems Engineering*, pp. 277–280, (1989).
14. T. Kohonen, G. Barna and R. Chrisley, Statistical pattern recognition with neural networks: Benchmarking studies, In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. I-61–I-68, (1988).
15. A. Lapedes and R. Farber, How neural networks work, In *Neural Information Processing Systems*, pp. 442–456, American Institute of Physics, New York, (1988).
16. A.E. Smith and C.H. Dagli, Relating binary and continuous problem entropy to backpropagation network architecture, In *Applications of Artificial Neural Networks II*, Volume 2, pp. 551–562, SPIE, Bellingham, WA, (1991).
17. T.J. Sejnowski and C.R. Rosenberg, Parallel networks that learn to pronounce English text, *Complex Systems*, 145–168 (1987).
18. J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, Academic Press, New York, (1969).
19. W. Mendenhall, R.L. Scheaffer and D.D. Wackerly, *Mathematical Statistics with Applications*, Duxbury Press, Belmont, CA, (1981).