

Randomness in Generalization Ability: A Source to Improve It

Dilip Sarkar, *Member, IEEE*

Abstract—Among several models of neurons and their interconnections, feedforward artificial neural networks (FFANN's) are most popular, because of their simplicity and effectiveness. Some obstacles, however, are yet to be cleared to make them truly reliable-smart information processing systems. Difficulties such as long learning time and local minima may not affect FFANN's as much as the question of generalization ability, because a network needs only one training, and then it may be used for a long time. The question of generalization ability of ANN's, however, is of great interest for both theoretical understanding and practical use. This paper reports our observations about randomness in generalization ability of FFANN's. A novel method for measuring generalization ability is defined. This method can be used to identify degree of randomness in generalization ability of learning systems. If an FFANN architecture shows randomness in generalization ability for a given problem, multiple networks can be used to improve it. We have developed a model, called voting model, for predicting generalization ability of multiple networks. It has been shown that if correct classification probability of a single network is greater than half, then as the number of networks in a voting network is increased so does its generalization ability. Further analysis has shown that VC-dimension of the voting network model may increase monotonically as the number of networks in the voting networks is increased. This result is counter intuitive, since it is generally believed that the smaller the VC-dimension, the better the generalization ability.

I. INTRODUCTION

ARTIFICIAL neural networks (ANN's) are mathematical models developed to mimic some information storing and processing capabilities of the brain of higher animals. Although interest of the research community in ANN's as a means for intelligent computing has existed for over 30 years (Widrow and Lehr [57] provide an extensive survey), Rumelhart *et al.* can be credited with for the revitalization of wide interest in it [41]. Because of 1) its simplicity; 2) its power to extract useful information from examples; and 3) its capability of storing information implicitly in the connecting links in the form of weights, currently EBP (error backpropagation) is the most popular learning algorithm for multilayer (feedforward) FFANN's. Despite its successful application in performing many intelligent information processing tasks in a wide range of fields, many of the characteristics of the EBP algorithm are not well understood by researchers. Learning speed, convergence rate, sensitiveness to initial weight sets, and generalization ability are among the most important ones. Difficulties such as long learning-time and local minima may

not affect FFANN's as much as the question of generalization ability, because a network needs only one training, and then it may be used for a long time. The question of generalization ability of ANN's, however, is of great interest for both theoretical understanding and practical use. Improvement of generalization ability is the theme of this work.

Generalization: One measure of performance of a learning system is how closely its actual output approximates the desired output for an input that it has never seen. This is known as generalization ability of the system. A popular method to understand the generalization ability of a system is: 1) divide the inputs into two subsets—one subset is called the training set and the other is called the test set, 2) repeatedly present elements from the training set to the system until error (difference between actual output and desired output) for each input is below a desired level, and 3) use elements from the test set for measuring the generalization ability of the system.

To develop an FFANN system, a network architecture (number of layers, number of nodes in each layer etc.) is selected, and then the selected network (architecture) is trained using some learning algorithm. Once trained, the network is like a "black box" that produces some output for every input, and the generalization ability depends on what is inside the black box. Thus, for a given learning algorithm and a given network architecture, the generalization ability depends on at least two factors—the training set and the initial weights. It is believed that the dependence on initial weights is due to the fact that each unique set of weights starts at a unique point in the weight space, and thus, starting at different initial points different local minima points are reached.

Research on generalization ability of FFANN's can be categorized mainly into two classes—1) structural adaptation before, or after, or during training for improving the generalization ability and 2) theoretical prediction of expected generalization ability. Structural adaptation can be classified as pruning methods, weight decay methods, and multiple network methods [46]. This work concentrates on multiple network methods and their variations. Before we present our results, a brief survey of the methods to improve generalization ability of FFANNAs are presented in Section II. Our observations about dependence of generalization ability of FFANN's on the initial weights are reported in Section III. We define a novel method for measuring generalization ability of learning systems in Section IV. This method also can be used for empirical estimation of dependence of initial weights on the generalization ability of FFANN's. In Section V, an analytical model is developed for predicting the performance of a class of

Manuscript received May 17, 1994; revised November 10, 1994.

The author is with the Department of Mathematics and Computer Science, University of Miami, Coral Gables, FL 33124 USA.

Publisher Item Identifier S 1045-9227(96)01229-5.

multiple networks, called voting networks. Section VI studies VC-dimension of voting networks. Section VII presents general conclusions and discusses possible future extension of work.

II. PREVIOUS AND CURRENT RELATED WORK

Werbos [56] reviews and evaluates supervised learning-generalization ability and learning speed. In Section II-A, a survey of methods for structural adaptation to improve the generalization ability of FFANN's is presented. Section II-B provides a brief review of important theoretical tools for understanding and predicting generalization ability of FFANN's.

A. Structural Adaptation

The structural adaptation approach can be grouped into three classes: pruning methods, weight decay methods, and multiple network methods. In pruning methods nodes or weights are removed when they are found to be redundant in some sense. The weight decay methods decay weights with the goal that the weights that appear to be unnecessary for storing the examples in the network will be removed eventually. The multiple networks methods employ several networks to improve generalization-ability and/or fault-tolerance.

1) *Pruning Methods*: Sietsma and Dow [45] have proposed that after training a network, units that are not useful for representation and classification of the examples may be pruned (removed) to improve its generalization ability. Their pruning method is based on informal intuitions. A more formal method for removing units is skeletonization, proposed by Mozer and Smolensky [32], in which relevance (difference of values of the energy function without the unit and with the unit in the network) of each unit is evaluated and units with lower relevance value are removed. Experimental study of skeletonization on two problems—a four-bit multiplexor problem and a random mapping problem—have shown that it reduces the size of the network by about half [32]. Recently, Ramachandran and Pratt have reported a skeletonization method which computes relevance of units based on their information measure [40]. Chauvin [7] has proposed a modification to the original EBP algorithm by adding a term to the sum-of-the-squared error energy function. Experimental observations and analytical explanations for improved generalization ability of this modified EBP algorithm were reported in [8] and [9].

In optimal brain damage (OBD) method of le Cun *et al.*, once the energy function has reached a local minimum, the importance of a weight is determined by computing saliency $s_i = (\partial^2 E / \partial w_i^2)(w_i^2 / 2)$ for weight w_i . The weights with small value of saliency have very little influence on the value of the energy function and are prime candidates for removal. A test of OBD for a handwritten character recognition task has reduced training time by a factor of about four and has produced networks with better generalization capability [28]. Hassibi and Stock [19], [20] also used a formal method, called optimal brain surgeon (OBS), to prune some weights (like OBD); in addition, OBS adjusts other weights, and hence is capable of obtaining networks with energy value lower than that obtained by OBD method. Hassibi and Stock have shown

that OBD is a special case of OBS. Simulation studies of OBS method have shown an impressive performance for several problems [19], [20].

2) *Weight Decay Methods*: Initial connection weights are selected randomly for breaking symmetry and improving convergence rate. Since some of these randomly selected initial weights may not be necessary for distributed representation of concepts that the network learns, however, they can be removed. Essentially this is done in the weight decay method proposed by Hinton [23]. In this method, a small fraction of the current weight is subtracted from it to get the next weight. Thus, if a weight is not reinforced frequently enough, then its value decays towards zero. Weigend *et al.* [54], [55] have a novel explanation, interpretation, and implementation of the weight decay method. With the usual sum-of-the-squared error function they added a term which would force a balance between performance error and the number of nonzero weights. Application of this novel idea can also be found in [51]–[53]. Simulation studies by Krogh and Hertz have shown higher generalization ability of networks trained using weight decay method [27]. Another variation of the weight decay method is a weight group decay method, proposed by Hanson and Pratt [18], in which all input weights of a unit are reduced by a constant amount. This method has the advantage that when all the weights of a unit decay towards zero, the unit itself can be removed. Experimental study indicates that it produces networks with better generalization ability, although the probability of convergence is reduced considerably [18]. Nowlan and Hinton [36], [37] have proposed a more complicated and sophisticated penalty term in which the distribution of weight values is modeled as a mixture of multiple gaussians; it simplified neural networks by soft weight-sharing. Simulations have shown a better generalization ability of networks created by this method. A great advantage of weight decay methods is that learning and pruning take place simultaneously.

3) *Multiple Networks Methods*: Hampshire and Waibel [17] trained a number of networks, each with a different energy function, on a speech recognition task. The networks' outputs were averaged to form the answer of the combined system during the testing phase, and a significantly higher generalization ability was obtained. Lincoln and Skrzypek [29] trained five identical size networks with the EBP algorithm and considered them as a cluster. During classification, the outputs from these five networks were used by a "judge" to form the output of the cluster. In comparison to a single network, a cluster of multiple networks showed better generalization ability and a significant fault tolerance. Pearlmutter and Rosenfeld [38] analyzed systems obtained by combining multiple networks of identical size and concluded that use of multiple networks increases the expected generalization ability. Druker *et al.* [11] have used voting networks trained with a boosting algorithm (which is based on Schapire's original work [44]). Sarkar's [42], [43] studies on voting networks (composed of multiple FFANN's) and FFANN's with multiple outputs have shown their improved generalization abilities. A more sophisticated version of multiple networks—competing expert networks—was

proposed by Jacobs *et al.* [24]. It has performed better than a single backpropagation network for a vowel recognition task [35].

B. Theoretical Considerations

Several theoretical methods developed, either 1) for analysis of FFANN's or 2) for some other applications have found use in FFANN's, are briefly discussed next.

Learning in a network has a close parallel to function approximation, in which given a few example points in a multidimensional space the objective is to extract parameters of the function such that the error of approximation to each value of the function is minimized [2], [39]. How good an approximation of a function may be obtained by a function approximation method? The answer to this question can be obtained from statistical estimation theory, since the function approximation may be considered as nonparametric statistical inference [14], [6], [61]. It is claimed that bias and variance affect the generalization ability of a network, and to obtain a good generalization ability, their sum has to be minimized. A network that minimizes the sum of bias and variance is very difficult to obtain [14]. Relationship between Bayesian and EBP methods is studied in [6] and [30], and it is also shown how this relationship can be utilized to obtain better FFANN's. Other studies relating neural networks with probability and/or statistics can be found, for example, in [34], [33], and [27].

From the work of Vapnik and Chervonenkis [49], the concept of Vapnik-Chervonenkis (VC) dimensionality has been developed and is being used to estimate bounds on the generalization ability of network families in terms of classes of functions they can realize [10]. Abu-Mostafa [1] in his review article explains how the work in [49] provides the key tools for dealing with the information issues related to neural networks. The pioneering work in [49] has been extended and/or elaborated in [50] and [47]. In addition, VC-dimension has been used for estimation of generalization of PAC (probably approximately correct) models [5], [21]. Vapnik [48] presents general principles of risk minimization for general learning theory. He argues that in learning from examples, there are two competing factors, 1) empirical risk and 2) confidence interval, whose sum should be minimized. A large network has a small empirical risk but it would have a large confidence interval. In particular, he discusses the effect of 1) network architecture, 2) weight decay during training, and 3) preprocessing of training patterns on generalization ability of networks. A continuation of these discussions are found in [16], where Guyon *et al.* explain the effect of principal component analysis, OBD, smoothing and regularization, and higher-order units. Guyon *et al.* [15] have proposed a method for automatic capacity tuning of very large VC-dimension classifiers (when the number of available training patterns is small).

Wolpert [58]–[61] has developed a mathematical theory of generalization, and favorably compared its generalization ability in few applications with that of neural networks. The probability of correct classification as a function of 1) number of training samples, 2) number of interconnection weights, and 3) the number of nodes has been estimated by Baum

and Haussler [3], [4]. Recently, Moody [31] has reported an analytical method for evaluating the expected test set errors, which provides a method for evaluating the generalization ability of neural networks. Relationship between statistical capacity and VC-dimension of neural networks can be found in [22] and [25].

III. RANDOMNESS IN GENERALIZATION ABILITY

In the sequel, an FFANN architecture (FFANNA) is a fixed interconnection of a fixed number of neurons in a fixed number of layers whose connection weights are yet to be determined, whereas an FFANN is an FFANNA whose connection weights have been fixed (possibly) by some learning algorithm. Application problems considered here are some type of classification of different patterns.

Let P be the total number of examples from n_o classes to be learned. Each example x_i is a n_o dimensional vector $(x_{1i}, x_{2i}, \dots, x_{n_o i})$. The number of input units, n_o is equal to the dimension of the input vectors and the number of output units, n_o is equal to the number of classes where patterns are to be classified. Without loss of generality, it can be assumed that the n_o classes are $1, 2, \dots, n_o$. The desired output for a class i is a binary vector $(b_1, b_2, \dots, b_{n_o})$ such that $b_i = 1$, and $b_j = 0$ for $j \neq i$. A training input vector x_i together with the corresponding desired output vector d_i is called a pattern pair. Let the number of hidden units in a layer l be n_l .

A. Randomness

Empirical studies on generalization ability of neural networks trained with the EBP algorithm have reflected that performance varies if initial weight sets are different [42], [43]. A more precise discussion on this variation in generalization ability is presented after introducing a few notations and definitions. Let the available data set be divided into two groups, training data set, and testing data set; and let the elements of each set be indexed with a unique integer. Let the set obtained by pairing each element of the training data set with its corresponding desired output be called training pair set and be denoted by DTR, and let the set obtained by pairing each element of the testing data set with its corresponding desired output be called testing pair set and be denoted by DTS.

Suppose for a pattern recognition problem an FFANNA has been selected, that is, the number of layers, the number of units in each layer, and the interconnection topology of neurons in different layers have been chosen. Now an FFANN can be obtained by assigning initial connection weights randomly, and then training the network with some learning algorithm. Suppose two different FFANN's, N_1 and N_2 are obtained with two different random initializations (of weights), but in both cases the same training algorithm presents elements of the training pair set DTR in the same sequential order. These two networks, N_1 and N_2 , are tested for their generalization ability with elements of the testing pair set DTS, and let their response sets be S_1 and S_2 , respectively. Each element of a response set is a doublet $\langle i, j \rangle$ whose first component is the index of the input x_i and the other component is the class j

where the input is classified. Inputs which are not classified in any class, or those which are classified in multiple classes are rejected and are not part of the response set.

One may look at S_1 and S_2 more closely. If S_1 and S_2 are not identical, the only factor that has contributed to their variation is initial weights. Thus, intersection I of S_1 and S_2 could be a measure of influence of initial weights on the generalization ability of the networks. An empty intersection I implies that the generalization ability of each network depends only on the initial weights; $I = S_1 = S_2$ implies that the generalization ability does not depend on the initial weights; otherwise, the generalization ability of each network depends on the initial weights, to some extent. Our experimental studies show that the intersection of S_1 and S_2 is usually nonempty, and it is a strict subset of either S_1 or S_2 . In the following section, some results from our simulations are reported. These simulation results have motivated us to study the effect of initial weights on the generalization ability of FFANN's. From our empirical studies it has been concluded that the subset of the testing pair set DTS that is recognized by a network depends on at least three factors: 1) the initial weight set, 2) the training data set, and 3) the learning algorithm.

B. Experiments

Experimental studies were conducted on several problems to obtain empirical results [42], [43]. In this section three of those problems are briefly described and empirical evidence of randomness in generalization abilities are reported. In the subsequent subsections, results of different empirical studies on these problems are summarized to show agreement with expected theoretical behavior.

1) *Handwritten Numeral Recognition (HNR)*: The classification task is to recognize handwritten numerals 0, 2, 3, 4, 5, 6, 7, 8, and 9.¹ Samples of handwritten digits were collected from 20 different persons, who were selected randomly. Thus, a total of 180 samples, with 20 samples for each of the digits, was collected to conduct the experiment. Each of these handwritten numerals was digitized and normalized to obtain a 15 by 11 binary matrix. The 180 samples were divided into two groups with 90 samples in each, and every group has 10 samples of each character. One of the two groups was used as the training pair set, and the other was used as the testing pair set.

A two-layer network architecture with nine units in the output layer was trained with the EBP algorithm to perform the classification task. Five different initial weight sets were used to obtain five networks and their responses were recorded as five sets S_1, S_2, \dots, S_5 . Sizes of these sets and their intersections are shown in Table I. Size of the set S_i is denoted by C_i . Size of a set obtained from the intersection of the sets $S_{i_1}, S_{i_2}, \dots, S_{i_j}$ is denoted by $C_{i_1 i_2 \dots i_j}$. Data from studies of two other problems will also be reported in the same format.

2) *Sonar Signal Classification (SSC)*: Data for this study were taken from the benchmark database maintained by Fahlman [12]. The task is to classify sonar signals bounced off a metal cylinder and a roughly cylindrical rock [12]. The

¹Digit 1 was omitted because it would make data normalization a little difficult.

TABLE I
THE SIZES OF THE RESPONSE SETS AND THEIR INTERSECTIONS
FOR HANDWRITTEN NUMERAL RECOGNITION PROBLEM

$C_1 = 30$	$C_2 = 35$	$C_3 = 33$	$C_4 = 28$	$C_5 = 29$
$C_{12} = 19$	$C_{13} = 17$	$C_{14} = 19$	$C_{15} = 18$	$C_{23} = 21$
$C_{24} = 18$	$C_{25} = 18$	$C_{34} = 21$	$C_{35} = 19$	$C_{45} = 16$
$C_{123} = 14$	$C_{124} = 14$	$C_{125} = 12$	$C_{134} = 15$	$C_{135} = 13$
$C_{145} = 14$	$C_{234} = 15$	$C_{235} = 14$	$C_{245} = 13$	$C_{345} = 14$
$C_{1234} = 12$	$C_{1235} = 11$	$C_{1245} = 11$	$C_{1345} = 11$	$C_{2345} = 12$
$C_{12345} = 10$				

TABLE II
THE SIZES OF THE RESPONSE SETS AND THEIR INTERSECTIONS FOR SONAR
SIGNAL CLASSIFICATION PROBLEM WITH THREE HIDDEN UNITS (SSC-3)

$C_1 = 167$	$C_2 = 163$	$C_3 = 167$	$C_4 = 166$	$C_5 = 168$
$C_{12} = 152$	$C_{13} = 153$	$C_{14} = 153$	$C_{15} = 151$	$C_{23} = 157$
$C_{24} = 155$	$C_{25} = 150$	$C_{34} = 157$	$C_{35} = 153$	$C_{45} = 155$
$C_{123} = 148$	$C_{124} = 147$	$C_{125} = 142$	$C_{134} = 147$	$C_{135} = 143$
$C_{145} = 145$	$C_{234} = 150$	$C_{235} = 145$	$C_{245} = 148$	$C_{345} = 149$
$C_{1234} = 143$	$C_{1235} = 138$	$C_{1245} = 141$	$C_{1345} = 141$	$C_{2345} = 143$
$C_{12345} = 137$				

transmitted sonar signal is a "frequency-modulated chirp," rising in frequency. Each received signal was transformed into a set of 60 numbers in the range 0.0 to 1.0. Details of the transformation process can be found in [12]. There were 208 patterns, 111 of which were obtained from signals bounced off a metal cylinder with aspect angle spanning 90 degrees, and the remaining 97 were obtained from signals bounced off a roughly cylindrical rock with aspect angle spanning 180 degrees.

Several two-layer network architectures were considered. All of them showed similar variations in their generalization abilities. The results for two network architectures with two and 24 units in the hidden layer (abbreviated as SSC-2 and SSC-24) are reported. Generalization abilities of these networks were measured following the method reported in [12]. The 208 elements of the data set, after randomization, were divided into 13 subsets. Twelve of these 13 subsets were used for training a network, while the remaining subset was used to measure its generalization ability. This process was repeated 13 times for testing generalization ability of a network architecture with the entire data set. The performance of these networks are summarized in Tables II and III.

3) *Majority-XOR (M-XOR)*: This is one of the problems Cohn and Tesauro used to see "can neural networks do better than Vapnik-Chervonenkis bounds?" [10]. It is an extension of

TABLE III
THE SIZES OF THE RESPONSE SETS AND THEIR INTERSECTIONS FOR SONAR
SIGNAL CLASSIFICATION PROBLEM WITH 24 HIDDEN UNITS (SSC-24)

$C_1 = 174$	$C_2 = 168$	$C_3 = 176$	$C_4 = 171$	$C_5 = 170$
$C_{12} = 155$	$C_{13} = 165$	$C_{14} = 159$	$C_{15} = 161$	$C_{23} = 158$
$C_{24} = 154$	$C_{25} = 154$	$C_{34} = 163$	$C_{35} = 160$	$C_{45} = 157$
$C_{123} = 151$	$C_{124} = 147$	$C_{125} = 148$	$C_{134} = 155$	$C_{135} = 156$
$C_{145} = 152$	$C_{234} = 148$	$C_{235} = 147$	$C_{245} = 146$	$C_{345} = 151$
$C_{1234} = 144$	$C_{1235} = 145$	$C_{1245} = 143$	$C_{1345} = 148$	$C_{2345} = 141$
$C_{12345} = 140$				

TABLE IV
THE SIZES OF THE RESPONSE SETS AND THEIR INTERSECTIONS
FOR MAJORITY-XOR PROBLEM WITH THREE HIDDEN UNITS

$C_1 = 395$	$C_2 = 412$	$C_3 = 444$	$C_4 = 378$	$C_5 = 444$
$C_{12} = 275$	$C_{13} = 297$	$C_{14} = 276$	$C_{15} = 301$	$C_{23} = 330$
$C_{24} = 278$	$C_{25} = 355$	$C_{34} = 314$	$C_{35} = 352$	$C_{45} = 306$
$C_{123} = 231$	$C_{124} = 207$	$C_{125} = 243$	$C_{134} = 229$	$C_{135} = 242$
$C_{145} = 222$	$C_{234} = 237$	$C_{235} = 301$	$C_{245} = 252$	$C_{345} = 259$
$C_{1234} = 181$	$C_{1235} = 214$	$C_{1245} = 188$	$C_{1345} = 192$	$C_{2345} = 218$
$C_{12345} = 169$				

the linearly separable majority function. Majority is a Boolean predicate in which the output is "1" if and only if more than half of the bits are "1." Majority-XOR is a Boolean function of N bits where output of the function is "1" if and only if N th bit disagrees with the majority of the first $N - 1$ bits.

Several networks were obtained by varying both data set size and the number of presentations of the data. All of them showed a similar trend. We present generalization ability of five networks with three hidden units. There were 600 samples, selected from uniformly distributed random binary numbers, which were divided into 10 subsets. Nine of these 10 subsets were used for the training, while the remaining subset was used for testing the generalization ability of the obtained network. The process was repeated 10 times to study generalization ability of a network architecture for the entire data set. The observed performances are summarized in Table IV. It may be recalled that formats of the table entries are identical to that of the earlier tables.

It is clearly evident from the Tables I-IV that for a given problem and a given network architecture, sizes of the response sets are not too different. Also, the intersections of response sets, two or multiple, show consistent results if the network architecture and the problem are fixed. Furthermore, we can see that the size of a set obtained from intersection of three response sets is smaller than that obtained from intersection

of two response sets. In general, as the number of response sets intersected is increased, the size of the intersected set decreases. This suggests that the influence of initial weights is not insignificant. For instance, in Table IV the size of both response sets C_3 and C_5 is 444, but size of their intersection C_{35} is 352. Thus, merely looking at the sizes of two response sets it cannot be concluded that the two networks have identical generalization ability. Also, because of the complex nature of neural networks, there is no available method to compute initial weights that will give a best generalization ability for a given FFANN architecture. Thus, the randomness in the generalization ability of neural networks is expected to remain. And hence, two most natural questions to ask are: 1) How to quantify the randomness in generalization ability of an FFANN architecture? and 2) Can this randomness in generalization ability of an FFANN architecture be used to obtain systems that have better generalization ability? In the next section, a method to measure the randomness in the generalization ability of an FFANN architecture, proposed in [42], is summarized.

IV. QUALITY MEASURE FOR GENERALIZATION ABILITY

Assume that there are m networks N_1, N_2, \dots, N_m whose tests of generalization abilities with the testing pair set DTS have produced the response sets S_1, S_2, \dots, S_m . The principle of inclusion and exclusion is applied on these sets to obtain the overall response of all the networks. Let

$$p_1 = \sum_{i=1}^m |S_i|, p_2 = \sum_{i_1=1}^{m-1} \sum_{i_2=i_1+1}^m |S_{i_1} \cap S_{i_2}|, \dots,$$

$$p_j = \sum_{i_1=1}^{m-j+1} \sum_{i_2=i_1+1}^{m-j+2} \dots \sum_{i_j=i_{j-1}+1}^m |S_{i_1} \cap S_{i_2} \cap \dots \cap S_{i_j}|.$$

Using the inclusion and exclusion principle p_{total} , the total number of elements from the testing pair set DTS recognized by the combined system is

$$p_{total} = p_1 - p_2 + p_3 - p_4 + \dots + (-1)^{m+1} p_m.$$

The number of terms summed to obtain p_j is ${}^m C_j$, the number of ways j objects can be chosen out of m different objects. Thus, the average number of elements in a set that is the intersection of j sets out of m sets is $(p_j / {}^m C_j)$. Now we use this average set size to define j th order generalization ability $g_{j,m}$ of a network architecture as

$$g_{j,m} = (p_j / {}^m C_j) / |DTS|. \quad (1)$$

This new definition of generalization ability of an FFANNA is a generalization of the conventional definition since, $g_{1,1}$ is the conventional definition. Also, $g_{1,m}$ is the average generalization ability of m networks corresponding to an FFANNA. A nice property of this definition of generalization ability of an FFANNA is that it automatically gives a nonincreasing sequence. This is stated next.

TABLE V
FIRST- TO FIFTH-ORDER GENERALIZATION ABILITY
MEASURES FOR HNR, SSC, AND M-XOR PROBLEMS

Problem	HNR	SSC-3	SSC-24	M-XOR
$g_{1,5}$	0.344	0.799	0.824	0.681
$g_{2,5}$	0.207	0.739	0.763	0.514
$g_{3,5}$	0.153	0.704	0.722	0.404
$g_{4,5}$	0.123	0.679	0.693	0.331
$g_{5,5}$	0.111	0.659	0.673	0.282

Property 1: For $1 \leq j < m$, $g_{j,m} \geq g_{j+1,m}$. This new measure of generalization ability can be used in several ways. For instance: 1) If $g_{j,m}$ are equal for all values of j even for a large m , the generalization ability of the networks architecture is (most likely) not sensitive to initial weight sets, and 2) If $g_{j,m} > g_{j+1,m}$, the generalization ability is sensitive to initial weight sets and may be improved by combining multiple networks. Some results from our empirical studies are summarized in Table V.

It can be seen from Table V that for all three problems, $g_{1,5} > g_{2,5}$, $g_{2,5} > g_{3,5}$, and so on. Thus, it is clear for each of these three problems that the generalization abilities of obtained neural networks are influenced by the choice of initial weight sets, (in addition to the training pattern set DTR and the training algorithm). Also, this initial-weight dependency of the generalization ability of an FFANNA could not be determined with existing methods to quantify it.

V. THE VOTING MODEL

In different settings multiple networks have been used and their outputs were combined to get a performance that is better than that of any single network. Some analytical models also were proposed to justify the improvement [38], [11]. In this section we view the improvements from a different angle and propose a model that predicts improved generalization ability of multiple networks. We shall treat FFANN's as function approximators and a combination of outputs of several networks as a composition of functions. The analysis will follow after introduction of a few notations and definitions. The analytical model described next assumes a two-class classification problem.

Let an FFANNA, a learning algorithm A , and a set of training patterns DTR together define a class of FFANN's (or functions), $F(A, DTR)$. In the following discussion it is assumed that, the algorithm presents the training patterns to the network in an identical sequential order each time. Starting from an initial state a learning algorithm A obtains an FFANN, $F(A, DTR, W)$, by presenting the patterns in the DTR for obtaining a set of final weights W . It is clear that $F(A, DTR, W) \in F(A, DTR)$. Let $P_{F(A, DTR, W)}(y)$ be the probability of correct classification by the network $F(A, DTR, W)$ of a pattern y (chosen randomly from outside the DTR, but from the same distribution as the elements of the

DTR). In the voting model, first several networks are obtained from different initializations, and then the final classification of a pattern is to the class (of two possible classes) where majority of the networks classified the pattern. Thus, each voting network, consists of $2n + 1$ networks working together to classify a pattern, can also be considered as a function. Let $G_{2n+1}(y)$ be the class of all voting networks that can be constructed by choosing $2n + 1$ networks from the networks in the class $F(A, DTR)$. Therefore, a voting network can be viewed as a new function $g_{2n+1}(y)$ which is a composition of a set of $(2n + 1)$ functions, $F_i(A, DTR, W_i)$, $1 < i < (2n + 1)$ as follows:

$$g_{2n+1}(y) = 1 \text{ if and only if } \sum_{i=1}^{2n+1} F_i(A, DTR, W_i) \geq (n + 1) \quad (2)$$

where it is assumed that $F_i(A, DTR, W_i) = 1$ if classification by the network is correct. Let $P_{g_{2n+1}}(y)$ be the probability that $g_{2n+1}(y) = 1$ for a test pattern y from the DTS. In an asymptotic case, we show that if $P_{F(A, DTR, W)}(y) > 1/2$, the probability that $g_{2n+1}(y) = 1$ goes to one as n goes to infinity. Before we proceed to find the asymptotic probability of correct classification, we now find an exact expression for the correct classification probability as a function of n and $P_{F(A, DTR, W)}(y)$. For the sake of brevity, $P_{F(A, DTR, W)}(y)$ will be written as $P_F(y)$.

If $2n + 1$ networks, $F_i(A, DTR, W_i)$, $1 \leq i \leq 2n + 1$, are randomly selected from $F(A, DTR)$, then the probability that exactly k of them will correctly classify a pattern y (which is also randomly and independently selected from the same distribution as the elements of DTR) is, $P_F^k(y)(1 - P_F(y))^{2n+1-k}$. Now, from the $2n+1$ networks, the k networks that correctly classify y can be chosen in ${}^{2n+1}C_k$ ways. Thus, the overall probability of obtaining exactly k correct classifications from $2n + 1$ networks is

$${}^{2n+1}C_k P_F^k(y)(1 - P_F(y))^{2n+1-k}. \quad (3)$$

A correct classification is obtained from a corresponding voting network $g_{2n+1}(y)$ when $k > n$, and thus, we have the following lemma.

Lemma 1: $P_{g_{2n+1}}(y)$, the probability of correct classification by a $(2n + 1)$ -voting network is

$$P_{g_{2n+1}}(y) = \sum_{k=n+1}^{2n+1} {}^{2n+1}C_k P_F^k(y)(1 - P_F(y))^{2n+1-k}. \quad (4)$$

For conciseness, writing $(1 - P_F(y)) = Q_F(y)$, and using (4) we have

$$P_{g_{2(n+1)+1}}(y) = \sum_{k=(n+1)+1}^{2(n+1)+1} {}^{2(n+1)+1}C_k P_F^k(y) \cdot Q_F^{2(n+1)+1-k}(y) \quad (5)$$

and

$$P_{g_{2n+1}}(y) = \sum_{k=n+1}^{2n+1} {}^{2n+1}C_k P_F^k(y) Q_F^{2n+1-k}(y). \quad (6)$$

By expanding ${}^{2(n+1)+1}C_k$ in (5) by using ${}^{m+1}C_x = {}^mC_x + {}^mC_{x-1}$, a standard identity for binomial coefficients [26], (and then keeping only the nonzero terms) we can obtain the following equation:

$$\begin{aligned} P_{g_{2(n+1)+1}}(y) &= \sum_{k=(n+1)+1}^{2n+1} {}^{2n+1}C_k P_F^k(y) Q_F^{2(n+1)+1-k}(y) \\ &+ 2 \sum_{k=n+1}^{2n+1} {}^{2n+1}C_k P_F^{k+1}(y) Q_F^{2(n+1)-k}(y) \\ &+ \sum_{k=n}^{2n+1} {}^{2n+1}C_k P_F^{k+2}(y) Q_F^{2n+1-k}(y). \quad (7) \end{aligned}$$

Since $P_F(y) + Q_F(y) = 1$, one can multiply the right side of (6) by $(P_F(y) + Q_F(y))^2$ and then distribute the summation to obtain the equation given next

$$\begin{aligned} P_{g_{2n+1}}(y) &= \sum_{k=n+1}^{2n+1} {}^{2n+1}C_k P_F^{k+2}(y) Q_F^{2n+1-k}(y) \\ &+ 2 \sum_{k=n+1}^{2n+1} {}^{2n+1}C_k P_F^{k+1}(y) Q_F^{2n+2-k}(y) \\ &+ \sum_{k=n+1}^{2n+1} {}^{2n+1}C_k P_F^k(y) Q_F^{2n+3-k}(y). \quad (8) \end{aligned}$$

Now, subtracting (8) from (7), and then doing some simplification we get

$$\begin{aligned} P_{g_{2(n+1)+1}}(y) - P_{g_{2n+1}}(y) \\ = {}^{2n+1}C_n P_F^{n+1}(y) Q_F^{n+1}(y) (P_F(y) - Q_F(y)). \quad (9) \end{aligned}$$

Because it was assumed that $P_F(y) > 1/2$ and $P_F(y) + Q_F(y) = 1$, we note that $P_{g_{2(n+1)+1}}(y) - P_{g_{2n+1}}(y) > 0$, or,

Lemma 2: $P_{g_{2(n+1)+1}}(y) > P_{g_{2n+1}}(y)$.

Thus, we have shown that the probability of correct classification increases monotonically as the number of networks used to construct the voting network is increased.

It can be recognized that in (4), the right-hand side is the sum of last n terms of the binomial distribution of $P_F(y)$. The expected value of the corresponding random variable X , $E(X) = (2n+1)P_F(y)$ and the variance of X is $Var(X) = (2n+1)P_F(y)(1-P_F(y))$. Hence, one can write

$$1 - P_{g_{2n+1}}(y) = \sum_{k=0}^n {}^{2n+1}C_k P_F^k(y) (1 - P_F)^{2n+1-k}. \quad (10)$$

Since $P_F(y) > 1/2$, it is clear that $(2n+1)P_F(y) > n$, and thus the right side of (10) can be bounded from the above. For details of this bounding process one can see, for example, Feller [13, pp. 150–152]. After the necessary approximation

TABLE VI
IMPROVEMENT OF GENERALIZATION ABILITY BY
VOTING NETWORKS FOR SSC AND M-XOR PROBLEMS

Problem	SSC-3	SSC-24	M-XOR
$g_{1,5}$	0.799	0.824	0.681
P_{g_3} - observed	0.808	0.844	0.735
P_{g_3} - theoretical	0.889	0.918	0.76
P_{g_5} - observed	0.813	0.85	0.75
P_{g_5} - theoretical	0.936	0.959	0.811

(10) can be written as the following inequality:

$$1 - P_{g_{2n+1}}(y) \leq ((n+1)P_F(y))/((2n+1)P_F(y) - n)^2. \quad (11)$$

Now limiting value of the expression to the right of the equality sign can be evaluated as the value of n goes to infinity, and it can be shown that

$$\lim_{n \rightarrow \infty} (1 - P_{g_{2n+1}}(y)) = 0, \quad \text{or} \quad \lim_{n \rightarrow \infty} P_{g_{2n+1}} = 1.$$

Therefore, we have the following theorem.

Theorem 1: If the probability of correct classification by a single network is greater than 1/2, then the probability of correct classification by the voting network goes to one as the number of networks used for the construction of the voting network goes to infinity.

Experimental Observations: We conducted extensive simulations for SSC-3, SSC-24, and M-XOR problems to see whether generalization abilities are indeed improved by various voting networks. We found a consistent and improved performance from the simulations (see Table VI). Table VI also contains theoretical predictions calculated from (4) for $n = 1$ and $n = 2$.

In Table VI, average generalization abilities of one (row $g_{1,5}$), three-voting (row P_{g_3} -observed), and five-voting (row P_{g_5} -observed) networks and their theoretical generalization abilities corresponding to the average generalization ability of one network has been shown (rows P_{g_3} -theoretical and P_{g_5} -theoretical). It can be seen that for each problem we obtained some improvement using voting networks. In general, five-voting networks have shown better generalization ability than that from three-voting networks. The theoretical values were computed with a basic assumption that the generalization ability is purely random—the percentage of patterns correctly classified is about the same for different initialization, but the patterns correctly classified is influenced by the choice of initial weights (which were randomly selected). As can be seen in Table VI, theoretically calculated generalization ability of each voting network is higher than that obtained from simulation. This is easily explained by observing that the randomness we observed in generalization abilities (see Table V) are not as high as it would be if it were totally

dependent on the initial weights. A general observation from the experiments is that a higher generalization is obtained by a voting network when the randomness, as measured by the new measure defined in earlier section, is higher in the generalization ability of the corresponding networks. This clearly shows how one can utilize the new measure to find the randomness involved in generalization ability of networks, and use the measured randomness to obtain (possible) voting networks with better performance.

VI. VC-DIMENSION AND VOTING NETWORKS

This section discusses our study on VC-dimension of voting networks (presented in the previous section). We start with necessary definitions. Let $B \subset R^n$, be a set of n dimension vectors. Let F be a set of binary functions from B to $\{0, 1\}$ and let X_i be a subset of B that contains i elements. A dichotomy of X_i induced by $f \in F$ is a partition of X_i into two disjoint subsets X_i^0 and X_i^1 such that for each $x \in X_i^0$, $f(x) = 0$ and for each $x \in X_i^1$, $f(x) = 1$. It is said that X_i is shattered by F if all 2^i dichotomies of X_i can be induced by the functions in F . The Vapnik-Chervonenkis (VC) dimension of F , denoted as $VC-dim(F)$, is the cardinality of a largest subset of B shattered by F , that is, the largest m such that there exists a X_m which is shattered by F .

In this section it is shown that the VC-dimension of voting networks may increase monotonically without bound if the number of networks used to construct them are increased without bound. Following the notation of the previous section, let $F(A, DTR)$ be the set of networks corresponding to 1) an FFANNA; 2) a learning algorithm A ; and 3) a training pattern set DTR . Let, without loss of any generality, $VC-dim(F(A, DTR)) = d$. This implies that there exists 2^d networks, $F_i(A, DTR, W_i) \in F(A, DTR)$, for $1 \leq i \leq 2^d$, such that they shatter a set of d input patterns, $X_d = \{x_1, x_2, \dots, x_d\}$. Let each $G (= \{g_1, g_2, \dots, g_{2^d}\})$ and $H (= \{h_1, h_2, \dots, h_{2^d}\})$ be a set of 2^d networks from $F(A, DTR)$ that shatters X_d . The indexes $1, 2, \dots, 2^d$ are assigned to the functions in G and H such that $g_i(x) = h_i(x)$ for $x \in X_d$. Now when a new pattern $y \in X_d$ is presented to the functions in G and H , let the response sequences be $G(y) = (g_1(y), g_2(y), \dots, g_{2^d}(y))$, and $H(y) = (h_1(y), h_2(y), \dots, h_{2^d}(y))$.

There could be three cases, 1) $G(y) = H(y)$; 2) $G(y) = \bar{H}(y)$, where $\bar{H}(y)$ denotes the bit-wise complement of $H(y)$; and 3) $G(y) \neq H(y)$, or $G(y) \neq \bar{H}(y)$. (Recall that f_i 's and G_i 's are binary functions with values 1 or 0). Case 2) cannot occur, since then $VC-dim(F(A, DTR))$ would be $d + 1$ which is a direct contradiction to our assumption that $VC-dim(F(A, DTR)) = d$. Using a special case of Case 3), we show that even if the VC-dimension of $F(A, DTR) = d$, three-voting networks can be constructed from them that have VC-dimension $d + 1$.

Lemma 3: If $g_i(y) = h_i(y)$ for a given value of i , and $g_j(y) \neq h_j(y)$ for all $j \neq i$, then there exists three-voting networks whose VC-dimension is one greater than that of the networks used to construct them.

Proof: The proof of the lemma is by construction. If we can construct a three-voting network v such that $v(y) =$

$g_i(\bar{y}) = h_i(\bar{y})$ for the given value of i and $v_j(x) = g_j(x) = h_j(x)$ for all $x \in X_d$, then $G \cup (H - \{h_i\}) \cup \{v\}$ will have VC-dimension $d + 1$ (by the discussion we already had just before the statement of this lemma). Construct a three-voting network as follows.

Step 1: Choose g_i as one of the three-voting networks.

The responses of g_i for inputs in X_d and y are $(g_i(x_1), g_i(x_2), \dots, g_i(y))$. Now select two functions h_j and h_k such that $h_j(x_1) = g_i(x_1), h_j(x_2) = g_i(x_2), \dots, h_j(x_l) = g_i(x_l), h_j(x_{l+1}) = g_i(x_{l+1}), h_j(x_{l+2}) = g_i(x_{l+2}), \dots, h_j(x_d) = g_i(x_d)$, and $h_k(x_1) = g_j(x_1), h_k(x_2) = g_j(x_2), \dots, h_k(x_l) = g_i(x_l), h_k(x_{l+1}) = g_i(x_{l+1}), h_k(x_{l+2}) = g_i(x_{l+2}), \dots, h_k(x_d) = g_i(x_d)$. Note that such a choice exists, since H includes 2^d functions that can induce all possible partitions of X_d . Now what remains is to select two networks from among g_j, g_k, h_j , and h_k in such a way that the outcome of a vote for the input y is $g_i(y)$. It may be recalled that $g_j(y) = h_j(\bar{y})$ and $g_k(y) = h_k(\bar{y})$.

Step 2: If $h_j(y) = g_i(\bar{y})$, then select h_j ; else select g_j .

Step 3: If $h_k(y) = g_i(\bar{y})$, then select h_k ; else select g_k .

The three-voting network v , thus constructed from three networks selected in three steps, produces $v(x) = g(x)$ for all $x \in X_d$, but $v(y) = g_i(\bar{y})$. This completes the proof, since the VC-dimension of $G \cup (H - \{h_i\}) \cup \{v\}$ is $d + 1$. \square

Now one can use three-voting networks to obtain voting networks consisting of nine networks whose VC-dimension may be $d + 2$. This process can be used recursively to obtain networks of VC-dimension greater than any given constant. Thus, it is clear that in the worst case one may obtain voting networks whose VC-dimension is greater than that of the networks used to construct them. This appears as a contradiction to our observations that generalization ability of voting networks are higher than that of the networks used to construct them. An explanation to this discrepancy was presented in [38] by using Chaitin-Kolmogorov complexity.

The VC-dimension is a measure of worst-case performance of a set of networks. Among a large set of networks, if only a small subset has poor generalization ability, VC-dimension of the networks will be large. The probability of selecting a poor network, however, will be small. Since voting model predicts performance when multiple networks are used, and if only a few of all the networks are poor, the probability of selecting at least half of the networks that are poor performers is very small. Thus, even though voting model may produce networks with higher VC-dimension, on an average it is very unlikely that one of the networks actually selected is a poor performer.

VII. CONCLUSIONS

It has been observed that networks obtained from a given network architecture for different initializations often correctly recognize different subsets of a given test set. Thus, it is concluded that random initializations of a network architecture may contribute to a type of randomness to its generalization ability. We have defined a measure to quantify generalization ability of learning systems. It also helps to identify initialization-related randomness component in generalization

ability of learning systems. We have developed an analytical model to predict performance of voting networks constructed from networks that have some degree of randomness in their generalization ability. One can use our model to find out (possible) improvement in generalization ability by making voting networks. In some cases, however, the obtained prediction can be an over estimation, and hence this model would require further refinement for more accurate predictions.

REFERENCES

- [1] Y. S. Abu-Mostafa, "The Vapnik-Chervonenkis dimension: Information versus complexity in learning," *Neural Comput.*, vol. 1, 1989.
- [2] A. R. Barron and R. J. Barron, "Statistical learning networks: A unifying view," in *Proc. Symp. Interface: Statist. Comput. Sci.*, 1988.
- [3] E. B. Baum and D. Haussler, "What size net gives valid generalization?" in *Advances in Neural Information Processing Systems 1*. San Mateo, CA: Morgan Kaufmann, 1989.
- [4] ———, "What size net gives valid generalization?" *Neural Comput.*, vol. 1, 1989.
- [5] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth, "Learnability and the Vapnik-Chervonenkis dimension," *J. Assoc. Comput. Machinery*, vol. 36, 1989.
- [6] W. L. Buntine and A. S. Weigend, "Bayesian backpropagation," *Complex Syst.*, 5, 1991.
- [7] Y. Chauvin, "A backpropagation algorithm with optimal use of hidden units," in *Advances in Neural Information Processing Systems 1*. San Mateo, CA: Morgan Kaufmann, 1989.
- [8] ———, "Dynamic behavior of constrained backpropagation networks," in *Advances in Neural Information Processing Systems 2*. San Mateo, CA: Morgan Kaufmann, 1990.
- [9] ———, "Generalization dynamics in LMS trained linear networks," in *Advances in Neural Information Processing Systems 3*. San Mateo, CA: Morgan Kaufmann, 1991.
- [10] D. Cohn and G. Tesauro, "Can neural networks do better than the Vapnik-Chervonenkis bounds?" in *Advances in Neural Information Processing Systems 3*. San Mateo, CA: Morgan Kaufmann, 1991.
- [11] H. Drucker, R. Schapire, and P. Simard, "Improving performance in neural networks using a boosting algorithm," in *Advances in Neural Information Processing Systems 5*. San Mateo, CA: Morgan Kaufmann, 1993.
- [12] S. E. Fahlman *et al.*, "Neural nets learning algorithms and benchmarks database," Carnegie Mellon Univ., Tech. Rep., 1993.
- [13] W. Feller, *An Introduction To Probability Theory And Its Applications*, vol. 1. New York: Wiley, 1988.
- [14] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 4, 1992.
- [15] I. Guyon, B. Boser, and V. Vapnik, "Automatic capacity tuning of very large VC-dimension classifiers," in *Advances in Neural Information Processing Systems 5*. San Mateo, CA: Morgan Kaufmann, 1993.
- [16] I. Guyon, V. Vapnik, B. Boser, L. Bottou, and S. A. Solla, "Structural risk minimization for character recognition," in *Advances in Neural Information Processing Systems 4*. San Mateo, CA: Morgan Kaufmann, 1992.
- [17] J. Hampshire and A. Waibel, "A novel objective function for improved phoneme recognition using time delay neural networks," Carnegie Mellon Univ., Tech. Rep. CMU-CS-89-118, 1989.
- [18] S. J. Hanson and L. Y. Pratt, "Comparing biases for minimal network construction with backpropagation," in *Advances in Neural Information Processing Systems 1*. San Mateo, CA: Morgan Kaufmann, 1989.
- [19] B. Hassibi and D. C. Stork, "Second-order derivatives for network pruning: Optimal brain surgeon," in *Proc. World Congr. Neural Networks*, 1993.
- [20] B. Hassibi, D. C. Stork, and G. J. Wolff, "Optimal brain surgeon and general network pruning," in *Proc. Int. Conf. Neural Networks*, 1993.
- [21] D. Haussler, "Decision theoretic generalization of the PAC model for neural net and other learning applications," *Inform. Comput.*, vol. 100, 1992.
- [22] D. Haussler, M. Kearns, M. Opper, and R. Schapire, "Estimating average-case learning curves using Bayesian, statistical physics, and VC dimension methods," in *Advances in Neural Information Processing Systems 4*. San Mateo, CA: Morgan Kaufmann, 1992.
- [23] G. E. Hinton, "Learning distributed representations of concepts," in *Proc. 8th Annu. Conf. Cognitive Sci. Soc.*, 1986.
- [24] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Comput.*, 1991.
- [25] C. Ji and D. Psaltis, "The VC-dimension versus the statistical capacity of multilayer networks," in *Advances in Neural Information Processing Systems 4*. San Mateo, CA: Morgan Kaufmann, 1992.
- [26] D. E. Knuth, *The Art of Computer Programming*, vol. 1. Reading, MA: Addison-Wesley, 1973.
- [27] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in Neural Information Processing Systems 4*. San Mateo, CA: Morgan Kaufmann, 1992.
- [28] Y. le Cun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems 2*. San Mateo, CA: Morgan Kaufmann, 1990.
- [29] W. P. Lincoln and Skrzypek, "Synergy of clustering multiple back-propagation networks," in *Advances in Neural Information Processing Systems 2*. San Mateo, CA: Morgan Kaufmann, 1990.
- [30] D. J. C. MacKay, "Bayesian model comparison and backpro bets," in *Advances in Neural Information Processing Systems 4*. San Mateo, CA: Morgan Kaufmann, 1992.
- [31] J. Moody, "The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems," in *Advances in Neural Information Processing Systems 4*. San Mateo, CA: Morgan Kaufmann, 1992.
- [32] M. C. Mozer and P. Smolensky, "Skeletonization: A technique for trimming the fat from a network via relevance assessment," in *Advances in Neural Information Processing Systems 1*. San Mateo, CA: Morgan Kaufmann, 1989.
- [33] A. F. Murray and P. J. Edwards, "Synaptic weight noise during MLP learning enhances fault-tolerance, generalization and learning trajectory," in *Advances in Neural Information Processing Systems 5*. San Mateo, CA: Morgan Kaufmann, 1993.
- [34] M. T. Musavi, K. H. Chan, D. H. Hummels, K. Kalantri, and W. Ahmed, "A probabilistic model for evaluation of neural-network classifiers," *Pattern Recognition*, vol. 25, 1992.
- [35] S. J. Nowlan and G. E. Hinton, "Evaluation of adaptive mixtures of competing experts," in *Advances in Neural Information Processing Systems 3*. San Mateo, CA: Morgan Kaufmann, 1991.
- [36] ———, "Adaptive soft weight tying using Gaussian mixtures," in *Advances in Neural Information Processing Systems 4*. San Mateo, CA: Morgan Kaufmann, 1992.
- [37] ———, "Simplifying neural networks by soft weight-sharing," *Neural Comput.*, vol. 4, 1992.
- [38] B. A. Pearlmutter and R. Rosenfield, "Chaitin-Kolmogorov complexity and generalization in neural networks," in *Advances in Neural Information Processing Systems 3*. San Mateo, CA: Morgan Kaufmann, 1991.
- [39] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, 1990.
- [40] S. Ramachandran and L. Y. Pratt, "Information measure based skeletonization," in *Advances in Neural Information Processing Systems 4*. San Mateo, CA: Morgan Kaufmann, 1992.
- [41] D. E. Rumelhart, J. L. McClelland, *et al.*, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, vol. 1. Cambridge, MA: MIT Press, 1986.
- [42] D. Sarkar, "Improving generalization through multiple redundant output units," in *Proc. World Congr. Neural Networks*, 1993.
- [43] ———, "Randomness in generalization ability: A source to improve it?" Univ. Miami, Tech. Rep. TR-CS-03-93, 1993.
- [44] R. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, 1990.
- [45] J. Sietsma and R. J. Dow, "Creating artificial neural networks that generalize," *Neural Networks*, vol. 4, 1991.
- [46] K. Turner, "Structural adaptation and generalization in neural networks," Master's thesis, Univ. Texas, Austin, 1992.
- [47] V. Vapnik, *Estimation of Dependences Based on Empirical Data*. Berlin: Springer-Verlag, 1982.
- [48] ———, "Principles of risk minimization for learning theory," in *Advances in Neural Information Processing Systems 4*. San Mateo, CA: Morgan Kaufmann, 1992.
- [49] V. Vapnik and A. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory Probability Applicat.*, vol. 16, 1971.
- [50] ———, "Necessary and sufficient conditions for the uniform convergence of means to their expectations," *Theory Probability Applicat.*, vol. 26, 1981.
- [51] A. S. Weigend, B. A. Huberman, and D. E. Rumelhart, "Predicting the future: a connectionist approach," *Int. J. Neural Syst.*, vol. 1, 1990.
- [52] ———, "Predicting sunspots and exchange rates with connectionist networks," in *Nonlinear Modeling and Forecasting, SFI Studies in the Sciences of Complexity, Proc. Vol. XII*. Reading, MA: Addison-Wesley, 1992.

- [53] D. E. Rumelhart A. S. Weigend, "Backpropagation, weight-elimination, and time-series prediction," in *Proc. Connectionist Models Summer School*. San Mateo, CA: Morgan Kaufmann, 1990.
- [54] ———, "Weight-elimination applied to currency exchange rate prediction," in *Proc. Int. Joint Conf. Neural Networks*, 1991.
- [55] ———, "Generalization by weight-elimination with application to forecasting," in *Advances in Neural Information Processing Systems 3*. San Mateo, CA: Morgan Kaufmann, 1991.
- [56] P. J. Werbos, "Supervised learning: Can it escape its local minima?" in *Proc. World Congr. Neural Networks*, 1993.
- [57] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: Perceptron, madline, and backpropagation," *Proc. IEEE*, vol. 78, 1990.
- [58] D. H. Wolpert, "Constructing a generalizer superior to NETalk via a mathematical theory of generalization," *Neural Networks*, 1990.
- [59] ———, "A mathematical theory of generalization: Part I," *Complex Syst.*, vol. 4, 1990.
- [60] ———, "A mathematical theory of generalization: Part II," *Complex Syst.*, vol. 4, 1990.
- [61] ———, "On the connection between in-sample testing and generalization error," *Complex Syst.*, vol. 6, 1992.



Dilip Sarkar (S'86-M'88) received the B. Tech. (Hons.) degree in electronics and electrical communication engineering from the Indian Institute of Technology, Kharagpur, India, in May 1983, the M. S. degree in computer science from the Indian Institute of Science, Bangalore, India, in December 1984, and the Ph.D. degree in computer science from the University of Central Florida, Orlando, in May 1988.

From January 1985 to August 1986 he was a Ph.D. student at Washington State University, Pullman. He is currently an Associate Professor of Computer Science at the University of Miami, Coral Gables. His research interest include design and analysis of algorithms, parallel architectures and algorithms, distributed processing, computer communication, parallel compilation, graph theory, and neural networks. In these areas, he has guided several theses and has authored over 40 papers.

Dr. Sarkar is member of the IEEE Computer Society and the Association for Computing Machinery. He was a recipient of the 14th All India Design Competition Award in electronics in 1982.