



EMPIRICAL ESTIMATION OF GENERALIZATION ABILITY OF NEURAL NETWORKS

*Dilip Sarkar**

Abstract: This work concentrates on a novel method for empirical estimation of generalization ability of neural networks. Given a set of training (and testing) data, one can choose a network architecture (number of layers, number of neurons in each layer etc.), an initialization method, and a learning algorithm to obtain a network. One measure of the performance of a trained network is how closely its actual output approximates the desired output for an input that it has never seen before. Current methods provide a “number” that indicates the estimation of the generalization ability of the network. However, this number provides no further information to understand the contributing factors when the generalization ability is not very good. The method proposed uses a number of parameters to define the generalization ability. A set of the values of these parameters provide an estimate of the generalization ability. In addition, the value of each parameter indicates the contribution of such factors as network architecture, initialization method, training data set, etc. Furthermore, a method has been developed to verify the validity of the estimated values of the parameters.

Key words: *Artificial neural networks, cross-validation, error back-propagation learning, feed-forward networks, generalization ability, voting networks*

Received: May 31, 2000

Revised and accepted: August 31, 2000

1. Introduction

Artificial Neural Networks (ANNs) are developed to mimic some information storing and processing capabilities of the brain of higher animals. Although the interest of the research community in ANNs as a means for intelligent computing has existed for over 30 years (Widrow and Lehr[24] provide an extensive survey), Rumelhart, McClelland, and the PDP research group can be given credit for the revitalization of wide interest in it [18], [19]. Different ANN models and their applications can be found in many books and in such surveys as [9], [10], [24]. In depth discussions

*Dilip Sarkar

Department of Computer Science, University of Miami, Coral Gables, FL 33124,
E-mail: sarkar@cs.miami.edu

and knowledge about neural networks for pattern recognition can be found in such books as, [1], [17].

At present, Feed-forward ANNs (FFANNs) are very popular, because of (i) their simplicity, (ii) their power to extract useful information from examples, and (iii) their capability of storing information implicitly in the connecting links in the form of weights. Despite numerous successful applications of EBP (Error Back Propagation) learning algorithm in performing many intelligent information processing tasks in a wide range of fields, many of its characteristics are not well understood by researchers. Learning speed, convergence rate, sensitiveness to initial weight sets, and generalization ability are among the most important ones. This work concentrates on empirical estimation of generalization ability of trained networks. A brief review of research on generalization is presented in Section 1.3.

1.1 Development of a FFANN system

Development of a FFANN system consists of the following steps:

- Select a network architecture (number of layers, number of nodes in each layer etc.).
- Select an initialization method, and initialize the network.
- Select training data set.
- Select a learning algorithm, and present data from the training data set until errors are below a predetermined limit.

The steps just described are *barely* enough to obtain a network from a network architecture. The obtained network is like a “black box” that produces some output for every input. Nothing can be said about its generalization ability (or its performance with data it was not trained with). To use the system with any level of confidence, it is necessary to know at least an estimate of its generalization ability. Thus it is necessary to add a final step in the network development: a method for estimating its generalization ability.

1.2 Generalization

One measure of the 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 system¹. Two popular methods to understand the generalization ability of a system are *test-set validation* and *cross-validation*.

In the test-set validation model, a part of available data is used to train the network, and the remaining exemplars are used to estimate the generalization ability of the trained network. This method is practical only if the available data set is very large or new data can be obtained cheaply. In the cross-validation model, the available data set is randomly partitioned into a number of approximately equal-size subsets. A network is trained with data in all but one of the subsets. The

¹Note that this most commonly used definition for generalization is different from that in [3], where it is called *rule extraction*.

remaining subset is used to estimate the generalization ability of the network. This process is repeated for each subset using it as the test set. The cross-validation estimate of generalization ability of the network architecture is an average of observed generalization abilities of all subsets.

1.3 Research on generalization ability

Research on generalization ability of FFANNs can be classified into three categories: i) theoretical prediction of expected generalization ability, ii) structural adaptation of the network before, or after, or during training for improving the generalization ability, and iii) empirical estimation of generalization ability of a trained network. All three research directions are important, since they complement each other. A better theoretical understanding provides methods to implement better systems, a good estimation of generalization ability of a network provides methods for assessing accuracy of a new theory, and experimental results are often useful to form a hypothesis for establishing new theories. For a survey of literature on the first two research categories refer to [21] and the references therein.

Bayesian ideas have been developed for application to (i) learning and regularization, (ii) evaluation of trained networks, and (iii) comparison of different trained networks [11], [14], [12]. A big advantage of the Bayesian methods is that they can eliminate the requirement of test data. Moreover, the Bayesian methods can be used for information-based selection of training data for a better generalization performance [13]. Despite many good features, the Bayesian methods being relatively new and being relatively complex are not popular yet. Also, like other methods, the Bayesian ideas do not help get around the local minima problem. The following two paragraphs briefly review the issues related to empirical estimation of generalization ability.

It was discussed earlier that after a network had been trained, the estimation of its generalization ability was very important. That is because the trained network may have just “memorized” the data used for training it. In that case, it might be necessary to adjust the structure of the network to improve its generalization ability. There are two simple methods that are widely used (please refer to Sec. 1.2) for estimation of generalization ability. When any of these methods is used, one obtains a “number” as the estimate of the networks generalization ability. The cross-validation method is most commonly used, since it requires relatively fewer data.

Although both methods are quite simple, there are good theoretical works to support their usefulness. The review of different cross-validation methods can be found in a number of books or surveys [4], [6], [8], [22], [23]. The general method of cross-validation is discussed in [15]. The cross-validation procedure was refined by Moody and Utans [16] for use in an environment where multiple local minima exists, such as FFANNs.

1.4 The problem

A limitation of these available methods for estimation of generalization ability is that they provide only a “number” stating the goodness of the generalization ability. Suppose, for example, there were 100 pairs of test data for estimating the

generalization ability of a network. By using the test-set validation method, for example, it has been estimated that 90 of the test data are correctly classified. From this empirical estimation one can assume that when the network is presented with a new datum, there is a 0.9 probability that it will be classified correctly. To make the estimation free of such bias as choice of initial weights, one can use many different sets of initial weights and observe the correct classification for the network trained with each set of initial weights. An average of these values can be used as the final estimation of the network's generalization ability. This use of more and more initializations will probably make the estimation robust. Nonetheless, from such a statement as, a 0.9 probability of correct classification, one has no knowledge about why the remaining 10% are not classified correctly. It could be one or more of the following: i) network architecture, ii) initialization method, iii) training data, and iv) learning algorithm. Efforts to find a contributing factor or factors from these four factors have provided methods to obtain networks with better generalization ability (see [21] for a survey).

2. Overview of the Proposed Method

We believe that to understand the influence of each factor, generalization ability should be expressed as a function of several parameters. The first step in estimating generalization ability of a FFANNA ought to be an estimation of the value of each of these parameters. The next step will be calculation of generalization ability from the estimated values of the parameters. Finally, there should be a method to check the "goodness" of the estimated generalization ability. Thus we propose a three step procedure to attack the problem:

- 1a) Choose the parameters that will be used to define generalization ability. For example, the fraction of the testing data set that is never correctly classified; or the fraction of the testing data set that is always correctly classified.
- 1b) Express generalization ability as a function of these parameters.
- 2a) Design a procedure to estimate the value of each of the parameters.
- 2b) Compute the generalization ability of the given FFANNA.
- 3) Device a method to verify the "goodness" of the parameters and/or generalization ability.

In addition to the novel idea of parameterization of generalization ability estimation, the concept of Step 3 is very new, to the best of our knowledge. We believe such a step should be an integral part of all generalization ability estimation methods. The reason for this belief is that no estimation method can provide excellent estimation for every network. Thus once an estimation has been obtained, one should check its validity.

3. Estimation of Generalization Ability

In the sequel, a *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 a FFANN is a FFANNA whose connection weights have been fixed (possibly) by some learning algorithm. Our approach to estimation of generalization ability is illustrated next (and can also be found in [20]). In this work, three parameters are used. The choice of these three parameters was guided by our experience from training many networks for various classification problems. A possible interpretation of these parameters is presented in Section 6.

3.1 Choosing the parameters

Let S be a set of C data for testing generalization ability of a FFANNA. Let F_{AL} be a fraction of the elements of S that is always correctly classified by any network obtained from the given FFANNA. Let F_{NL} be a fraction of the elements that are never correctly classified by any network obtained from the same FFANNA. Now the remaining fractions of the elements of S are those whose a part is correctly classified by some networks but none classified by all the networks. Let this fraction be denoted by F_{ML} . Note that

$$F_{AL} + F_{NL} + F_{ML} = 1. \quad (1)$$

Let us denote the set of elements corresponding to F_{AL} , F_{NL} , and F_{ML} be S_{AL} , S_{NL} , and S_{ML} , respectively. It should be noted that S_{AL} , S_{NL} , and S_{ML} are subsets of S , and are mutually disjoint. Also their union is the set S . Thus they form a partition of S . It also should be noted that any of this partitions could be empty. Let PR_{ML} be the probability of correctly classifying an element of S_{ML} by a trained network. These parameters can define the estimated generalization ability as follows:

$$G(F_{AL}, F_{ML}, PR_{ML}) = F_{AL} + F_{ML} * PR_{ML} \quad (2)$$

One can obtain very good estimates of F_{AL} and F_{ML} by counting for the fraction of always and sometimes correctly classified examples, and for PR_{ML} the fraction of sometimes correctly classified examples that are correctly classified. However, for a good estimate of the parameters, the counting method requires training and testing of a very large number of networks. An alternative method that worked well, and requires training and testing of only a few networks is described next. For developing this method it is assumed that *errors for the patterns from subset S_{ML} are independent between initializations*. This assumption may appear arbitrary, but we believe there are some connections between the local minima problem and the correct classification of the elements from subset S_{ML} .

3.2 A procedure to estimate values for the parameters

Let C_{AL} , C_{NL} , and C_{ML} be the number of elements in S_{AL} , S_{NL} , and S_{ML} , respectively. Note that,

$$C_{AL} + C_{NL} + C_{ML} = C$$

and

$$F_{AL} = \frac{C_{AL}}{C}, F_{NL} = \frac{C_{NL}}{C}, \quad \text{and} \quad F_{ML} = \frac{C_{ML}}{C}$$

- 1) From a Given FFANNA obtain a network N_1 by training it with the given set of training data. Find the set of elements S_1 that is correctly classified by the network. Let C_1 be the number of elements in S_1 .
- 2) Repeat Step 1 four times, for example, to construct $N_2, N_3, N_4,$ and $N_5,$ and record corresponding correctly classified data sets $S_2, S_3, S_4,$ and $S_5,$ respectively. Let $C_2, C_3, C_4,$ and C_5 be the number of elements in $S_2, S_3, S_4,$ and $S_5,$ respectively.
- 3) Compute $g_{1,5}$, the 1st order generalization ability as follows,

$$g_{1,5} = (C_1 + C_2 + C_3 + C_4 + C_5)/(5 * C). \quad (3)$$

The general j th order generalization was defined by Sarkar in [21]. To make the presentation concise, it is necessary to introduce some notations. Let $C_{i_1 i_2 \dots i_j}$ be the size of the set obtained from intersection of $S_{i_1}, S_{i_2}, \dots, S_{i_j}$. Let $C_{i_1 i_2 \dots i_j}^{av}$ be the average size of the sets obtained from all possible intersections of j sets from S_1, S_2, \dots, S_k . Now j th order generalization is computed as follows:

$$g_{j,k} = C_{i_1 i_2 \dots i_j}^{av} / C \quad (4)$$

- 4) Compute $g_{2,5}$, and $g_{3,5}$.
- 5) Solve the following three equations relating $F_{AL}, F_{ML},$ and PR_{ML} with $g_{1,5}, g_{2,5},$ and $g_{3,5}$.

$$g_{1,5} = F_{AL} + F_{ML} * PR_{ML} \quad (5)$$

$$g_{2,5} = F_{AL} + F_{ML} * (PR_{ML})^2 \quad (6)$$

$$g_{3,5} = F_{AL} + F_{ML} * (PR_{ML})^3 \quad (7)$$

Let us see how one obtains Equation (5). By our definition, $g_{1,5}$ is a fraction of C elements from S that is correctly classified on an average. However, $g_{1,5}$ can also be written as the sum of two fractions. One is obtained from S_{AL} , the subset of S whose elements are always correctly classified. This fraction is the ratio of the sizes of S_{AL} and S , or $(C_{AL}/C) = F_{AL}$. The other fraction comes from S_{ML} . By our assumption, the probability that an element of S_{ML} is correctly classified is PR_{ML} . Since S_{ML} has C_{ML} elements, the number of elements from S_{ML} that, on an average, are correctly classified is $C_{ML} * PR_{ML}$. Thus, to find the fractional contribution from this set one divides $(C_{ML} * PR_{ML})$ by C to obtain $(C_{ML} * PR_{ML})/C = (C_{ML}/C) * PR_{ML} = F_{ML} * PR_{ML}$. This completes the justification for writing Equation (5). Using similar arguments, one can justify Equations (6) and (7).

3.3 Solving the equations

A set of values for the three parameters can be obtained by solving Equations (5), (6), and (7). Since the values of $g_{1,5}$, $g_{2,5}$, and $g_{3,5}$ have been estimated empirically, it will be enough if PR_{ML} , F_{ML} , and F_{AL} are directly (or indirectly) expressed as their function. An outline of a method for solving Equations (5), (6), and (7) to obtain the value of the parameters is given next. From Equations (5) and (6) we can get

$$g_{1,5} - g_{2,5} = PR_{ML} * F_{ML}(1 - PR_{ML}) \quad (8)$$

and from Equations (6) and (7) we can get

$$g_{2,5} - g_{3,5} = (PR_{ML})^2 * F_{ML}(1 - PR_{ML}). \quad (9)$$

One can use Equations (8) and (9) to obtain

$$PR_{ML} = \frac{g_{2,5} - g_{3,5}}{g_{1,5} - g_{2,5}} \quad (10)$$

Similarly, the values of F_{ML} and F_{AL} can be obtained, and given next.

$$F_{ML} = \frac{g_{1,5} - g_{2,5}}{PR_{ML} - (PR_{ML})^2} \quad (11)$$

and

$$F_{AL} = g_{1,5} - F_{ML} * PR_{ML} \quad (12)$$

Following the outlined steps one can estimate the values of the parameters and also give an estimate of generalization ability of the network. However, how good are the estimates? How closely do the estimated values of these parameters model the behavior of the FFANNA? To answer such questions it is necessary to develop a method to verify, in a completely different way, the performance prediction of the network. In the next subsection we present a method to verify the validity of the obtained estimates.

4. Verification of Validity of the Parameters

The values of the parameters were obtained from observed values of $g_{1,5}$, $g_{2,5}$, and $g_{3,5}$. One can take another higher order generalization, such as $g_{4,5}$ to check the validity of the model. However, we believe a totally different method should be employed to check the validity of the estimated values of the parameters. We have used the parameters to predict generalization of *voting-networks*. If a set of values of the parameters, when used to predict the performance of voting-networks, do the prediction with satisfactory accuracy, we assume that the estimated values for the parameters are “good” and acceptable. Otherwise, they are rejected and the generalization ability prediction using them is questionable.

4.1 The voting networks

In a 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 a majority of the networks classified the pattern. Let g_{2n+1} be a voting network consisting of $2n + 1$ networks. Thus each voting network consists of $2n + 1$ networks [21]. Let $P_{g_{2n+1}}(y_{ML})$ be the probability that a test pattern y_{ML} from S_{ML} is correctly classified by g_{2n+1} .

If $2n + 1$ networks are randomly selected from all possible networks of a given FFANNA, then the probability that exactly k of them will correctly classify a pattern y_{ML} is $PR_{ML}^k(y_{ML})(1 - PR_{ML}(y_{ML}))^{2n+1-k}$, where $PR_{ML}(y_{ML})$ is the probability of correctly classifying y_{ML} by a single network. (Please refer to [21] for derivation and discussions.) Now, from the $2n + 1$ networks, the k networks that correctly classify y_{ML} can be chosen in ${}^{2n+1}C_k$ ways. Thus the overall probability of obtaining exactly k correct classifications from $2n + 1$ networks is

$${}^{2n+1}C_k PR_{ML}^k(y_{ML})(1 - PR_{ML}(y_{ML}))^{2n+1-k}. \quad (13)$$

Now when $k > n$, a correct classification is obtained from the corresponding voting network g_{2n+1} , and thus, $P_{g_{2n+1}}(y_{ML})$, the probability of a correct classification by a $(2n + 1)$ -voting network is

$$P_{g_{2n+1}}(y_{ML}) = \sum_{k=n+1}^{2n+1} {}^{2n+1}C_k PR_{ML}^k(y_{ML})(1 - PR_{ML}(y_{ML}))^{2n+1-k}. \quad (14)$$

An element y from the testing set S may have come from the subset S_{AL} with probability F_{AL} , from the subset S_{ML} with probability F_{ML} , or from the subset S_{NL} with probability F_{NL} . Thus $P_{g_{2n+1}}(y)$, the probability that an element y from the testing data set S is correctly classified by a voting network g_{2n+1} is:

$$P_{g_{2n+1}} = F_{AL} * 1 + F_{ML} * P_{g_{2n+1}}(y_{ML}) + F_{NL} * 0 = F_{AL} + F_{ML} * P_{g_{2n+1}}(y_{ML}) \quad (15)$$

Now, if the correct classification probability predicted by this equation matches closely with that obtained from empirical observations, we accept the estimated values of the parameters as "good" values. In the following case study we show that the actual values of a correct prediction probability are in very good agreement with empirical observations.

5. Experiments

Experimental studies were conducted on several problems to obtain empirical results [20], [21]. In this subsection two of these problems are briefly described and empirical results are reported.

5.1 Sonar signal classification (SSC)

Data for this study were taken from the benchmark database maintained by Fahlman [5]. The task is to classify sonar signals bounced off a metal cylinder and a roughly cylindrical rock [5], [7]. The 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 about the transformation process can be found in [5], [7]. There were 208 patterns, 111 of which were obtained from signals bounced off a metal cylinder with aspect angle spanning of 90 degrees, and the remaining 97 were obtained from signals bounced off a roughly cylindrical rock with aspect angle spanning of 180 degrees.

Several two-layer network architectures were considered. All of them showed similar randomness in their generalization abilities. Results of two network architectures with 3 and 24 units in the hidden layer (abbreviated as SSC-3 and SSC-24) are reported. For each architecture five different initial-weight sets were used to obtain five networks. The generalization abilities of these networks were measured following the method reported in [5]. 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, and a set of correctly classified data S_1 was obtained. Similarly, correctly classified data sets S_2 , S_3 , S_4 and S_5 were obtained. The performance of this network architecture is summarized in Tables I and II. The size of the set S_i is denoted by C_i . The size of a set obtained from the intersection of sets $S_{i_1}, S_{i_2}, \dots, S_{i_j}$ is denoted by $C_{i_1 i_2 \dots i_j}$.

$C_1 = 166$	$C_2 = 169$	$C_3 = 163$	$C_4 = 168$	$C_5 = 155$
$C_{12} = 154$	$C_{13} = 151$	$C_{14} = 150$	$C_{15} = 137$	$C_{23} = 151$
$C_{24} = 155$	$C_{25} = 145$	$C_{34} = 149$	$C_{35} = 137$	$C_{45} = 138$
$C_{123} = 144$	$C_{124} = 145$	$C_{125} = 133$	$C_{134} = 141$	$C_{135} = 127$
$C_{145} = 127$	$C_{234} = 142$	$C_{235} = 132$	$C_{245} = 133$	$C_{345} = 125$
$C_{1234} = 137$	$C_{1235} = 125$	$C_{1245} = 125$	$C_{1345} = 119$	$C_{2345} = 123$
$C_{12345} = 118$				

Tab. I The sizes of the response sets and their intersections for sonar signal classification problem with 3 hidden Units (SSC-3).

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

Tab. II The sizes of the response sets and their intersections for sonar signal classification problem with 24 hidden units (SSC-24).

5.1.1 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?’ [2]. It is an extension of 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 the 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 3 hidden units. There were 600 samples of length $N = 24$, selected from uniformly distributed random binary numbers, which were divided into 10 subsets. Nine of these 10 subsets were used for 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 III. It may be recalled that the formats of the table entries are identical to those of the earlier tables.

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

Tab. III The sizes of the response sets and their intersections for majority-XOR problem with 3 hidden units.

Estimation of the values of the parameters From Table I we obtain $g_{1,5} = 0.799$, $g_{2,5} = 0.739$, and $g_{3,5} = 0.704$. The substitution of these values in Equation (10) estimates $PM_{ML} = 0.583$. The value of $F_{ML} = 0.247$ is obtained from Equation (11). Finally, $F_{AL} = 0.655$ is calculated from Equation (12). Similarly, from Tables II and III, and Equations (10) to (12) parameters of other network architectures are estimated. The values of these estimated parameters for all three networks are shown in Table IV.

Problem	SSC-3	SSC-24	M-XOR
$g_{1,5}$	0.799	0.824	0.681
$g_{2,5}$	0.739	0.763	0.514
$g_{3,5}$	0.704	0.722	0.404
PM_{ML}	0.583	0.672	0.659
F_{ML}	0.247	0.277	0.743
F_{AL}	0.655	0.638	0.192
F_{NL}	0.098	0.085	0.065

Tab. IV Estimated values of different parameters of the networks for SSC-3, SSC-24, and M-XOR problems.

Validation of the obtained values of the parameters To validate these estimated values of the parameters F_{AL} , F_{ML} , and PR_{ML} , we now use Equations (14) and (15). The objective is to predict generalization ability of voting networks from these values of the parameters, and compare them with empirical observations. From equation (14) we get $P_3(y_{ML}) = 0.623$, and $P_5(y_{ML}) = 0.653$. Thus the corresponding estimated generalization abilities of the 3-voting and 5-voting networks are $P_3(y) = 0.809$ and $P_5(y) = 0.816$. These values are in very good agreement with the empirical values 0.808 and 0.813 obtained from 3-voting and 5-voting networks, respectively. Thus we conclude that the estimated values of the parameters are robust. Table V shows estimated and empirical values of validation parameters for SSC-3, SSC-24, and M-XOR problems. We can see that the predicted performances of the voting networks are in good agreement with those of the voting networks studied empirically. Thus it can be concluded that the estimated values of the parameters are “good”.

Problem	SSC-3	SSC-24	M-XOR
$P_3(y_{ML})$ from Eqn (14)	0.623	0.748	0.73
$P_3(y)$ from Eqn (15)	0.809	0.845	0.734
3-voting networks	0.808	0.844	0.735
$P_5(y_{ML})$ from Eqn (14)	0.653	0.798	0.778
$P_5(y)$ from Eqn (15)	0.816	0.859	0.769
5-voting networks	0.813	0.85	0.75

Tab. V *Estimated and empirical values of validation parameter for SSC-3, SSC-24, and M-XOR problems.*

6. Discussion

The purpose of this section is to present a discussion on possible interpretations of the parameters F_{AL} , F_{NL} , F_{ML} , and PR_{ML} . We start our discussion with F_{AL} .

Parameter F_{AL} is an indicator to influence the weight selection method used on the generalization ability of the trained network. A value close to one indicates that the FFANNA’s generalization is almost independent of the weight initialization method used. On the other extreme, the value of zero indicates that the generalization of the FFANNA is completely dependent on the weight initialization method. The value of F_{AL} somewhere between zero and one indicates some degree of dependence on the weight initialization method. The degree of dependence is proportional to the value of F_{AL} . Next the parameter F_{NL} is discussed.

The value of this parameter is an indication of the fraction of the training data from the training set which is merely “memorized” by the trained network, but their features are not learned for generalization. The value of zero or close to zero is most desirable, and it indicates that the trained network has memorized none or very few of the examples from the training data set. On the other hand, F_{NL} ’s value one or close to one indicates that the trained network has memorized all or most of the data, and hence, it classifies none or hardly any data from the testing data set. It should be noted that the value of zero for parameter F_{NL}

may not mean that the trained network has generalized very well. To come to the conclusion that a network has generalized very well, one requires to examine the values of parameters F_{AL} , F_{ML} , and PR_{ML} also. An unacceptably large value of F_{NL} indicates that there may be a problem with i) learning algorithm (or values of the parameters associated with it), or ii) network architecture, or iii) (in very few cases) the initialization method used.

Let us see some examples. Suppose a simple learning algorithm stores every pattern presented to it in a table. During the classification of a new pattern, it examines the table entries sequentially and classifies a pattern to a class only if a perfect match is found. This learning algorithm obviously does not do any generalization, and hence, if the testing data set is different from the training data set, the value of F_{NL} will be one.

The effect of network architecture may also be reflected by the value of F_{NL} . For example, if one chooses a relatively large network for the set of patterns to be learned, the network usually memorizes the patterns in this oversized network. When this trained network is presented with a test pattern which is slightly different from all training patterns, it is most likely to be classified incorrectly. The influence of the initialization method may affect the value of this parameter in some cases, but it is very unlikely to be so.

The parameter F_{ML} is an indicator of the influence of the initialization method on the generalization ability. Note that its role is somewhat complementary to F_{AL} . The small value of this parameter indicates that the initialization method has very little influence. But a value close to one indicates a great influence of the initialization method on the generalization ability of a trained network. However, the influence of this parameter must be judged in conjunction with PR_{ML} for the reason discussed next.

The value of PR_{ML} acts as a scaling factor for F_{ML} to compute the contribution of the generalization ability of a trained network. A smaller value diminishes the contribution of F_{ML} to the overall generalization ability, and a value closer to one does not reduce the contribution to the generalization ability very much. We believe that the parameters F_{ML} and PR_{ML} are related to the local minima effect. If the existence of local minimum for a problem does not affect the generalization, the F_{ML} would be zero.

7. Conclusion

The parameterization of the generalization ability has provided us with a good tool to obtain insight into the possible reasons for either generalizing or not generalizing well. For instance, from Table IV we find that for the SSC-3 problem the fraction of data that is always correctly classified is 0.655, that is, $F_{AL} = 0.655$; the fraction of data that may or may not be correctly classified by any given network is 0.247, that is, $F_{ML} = 0.247$; and the fraction of data that is never correctly classified is $(1 - 0.655 - 0.247) = 0.098$, that is, $F_{NL} = 0.098$. This means that the features of about 9.8% data is never captured by the network, and will always be classified incorrectly. (Note that the existing method would find that the probability of correct classification is 0.799, and from this, one could incorrectly conclude that the training data set does not represent features of 20.1% of the testing data.) Thus

one has to add new data representing features of the incorrectly classified data in the training data set.

References

- [1] Bishop C.M.: Neural networks for pattern recognition. Oxford University Press, 1996.
- [2] Cohn D., Tesauro G.: Can neural networks do better than the Vapnik-Chervonenkis bounds? In: Advances in Neural Information Processing Systems 3, Morgan Kaufmann, 1991.
- [3] Denker J., Schwartz D., Wittner B., Solla S., Howard R., Jackel L.: Large automatic learning, rule extraction, and generalization. Complex Systems, 1, 1987.
- [4] Eubank R.L.: Spline smoothing and nonparametric regression. Marcel Dekker, Inc., 1988.
- [5] Fahlman S.E. et al.: Neural nets learning algorithms and benchmarks database. Technical report, Carnegie Mellon University, 1993.
- [6] Geisser S.: The predictive sample reuse method with applications. Journal of the American Statistical Association, 70, 1975.
- [7] Gorman R.P., Sejnowski T.J.: Analysis of hidden units in a layered network trained to classify sonar targets. Neural Networks, 1, 1988.
- [8] Hastie T.J., Tibshirani: Generalized additive models. Chapman and Hall, 1990.
- [9] Hinton G.E.: Connectionist learning procedures. Artificial Intelligence, 40, 1989.
- [10] Lippmann R.P.: An introduction to computing with neural network. IEEE ASSP Magazine, 1987.
- [11] Mackay D.J.C.: Bayesian interpolation. Neural Computation, 1992.
- [12] Mackay D.J.C.: The evidence framework applied to classification networks. Neural Computation, 1992.
- [13] Mackay D.J.C.: Information-based objective functions for active data selection. Neural Computation, 1992.
- [14] Mackay D.J.C.: A practical Bayesian framework for backprop networks. Neural Computation, 1992.
- [15] Moody J.: Prediction risk and architecture selection for neural networks. In: From Statistics to Neural Networks: Theory and Pattern Recognition Applications, Springer-Verlag, 1994.
- [16] Moody J., Utans J.: Principled architecture selection for neural networks: Application to corporate bond rating prediction. In: Advances in Neural Information Processing Systems 4, 4, Morgan Kaufmann, 1992.
- [17] Ripley B.D.: Pattern recognition and neural networks. Cambridge University Press, 1996.
- [18] Rumelhart D.E., McClelland J.L., et al.: Parallel distributed processing: explorations in the microstructures of cognition. MIT Press, 1, 1986.
- [19] Rumelhart D.E., McClelland J.L., et al.: Parallel distributed processing: explorations in the microstructures of cognition. MIT Press, 2, 1986.
- [20] Sarkar D.: A novel method for estimation of generalization ability of neural networks. Technical Report TR-CS-01-95, University of Miami, 1995.
- [21] Sarkar D.: Randomness in generalization ability: A source to improve it. IEEE Transactions on Neural Networks, 7, 1996.
- [22] Stone M.: Cross-validatory choices and assessment of statistical predictions. Roy. Stat. Soc., B36, 1974.
- [23] Stone M.: Cross-validation: A review. Math. Operationsforsch. Statist., Ser. Statistic, 9, 1978.
- [24] Widrow B., Lehr M.A.: 30 years of adaptive neural networks: Perceptron, madline, and backpropagation. Proceedings of the IEEE, 78, 1990.