PERGAMON

Neural Networks 11 (1998) 851–859

Neural
Networks

Contributed article

Statistical estimation of the number of hidden units for feedforward neural networks

Osamu Fujita*

NTT System Electronics Laboratories, Atsugi, Kanagawa 243-01, Japan

Received 15 August 1995; revised 26 March 1998; accepted 26 March 1998

Abstract

The number of required hidden units is statistically estimated for feedforward neural networks that are constructed by adding hidden units one by one. The output error decreases with the number of hidden units by an almost constant rate, if each appropriate hidden unit is selected out of a great number of candidate units. The expected value of the maximum decrease per hidden unit is estimated theoretically as a function of the number of learning data sets in relation to the number of candidates that are obtained by random search. This relation can be expanded to cover other searching methods. In such a case, the number of candidates implies how many steps might be required if random search were used instead. Therefore the number of candidates can be regarded as a parameter that represents the efficiency of the search. Computer simulation shows that estimating this parameter experimentally from the actual decrease in output error is useful for demonstrating the efficiency of the gradient search. It also shows the influence, on the number of hidden units, of the hidden unit's nonlinearity. © 1998 Elsevier Science Ltd. All rights reserved.

Keywords: Neural network; Feedforward network; Least squares approximation; Number of hidden units; Statistics of extremes; Efficiency of search

1. Introduction

The number of hidden units in a feedforward neural network is significant in characterizing the performance of the network. It greatly influences network capacity (Baum, 1988; Akaho and Amari, 1990), generalization ability (Baum and Haussler, 1989), learning speed and output response. For capacity and universality in application to function approximation, it is apparently better for the number of hidden units to be as large as possible. On the other hand, from the standpoint of generalization, the number should not be too large for heuristic learning systems in which the best network is a-priori unknown and has to be determined stochastically. Using the analogy of Akaike's Information Criterion (Akaike, 1974; Kurita, 1990; Moody, 1992), an optimum number of hidden units ought to exist, which depends on the complexity of a given learning task. But estimating the number before the learning task is done is difficult. To meet such a requirement, the actual number of hidden units should be flexible and adjustable to the optimum number during learning. There may be many such kinds of learning systems (for example, Ash, 1989; Hagiwara, 1990; Hirose et al., 1989).

A simple one of flexible networks is the growth network that is constructed by adding hidden units one by one so as to reduce the output error of the network (for example, Fahlman and Lebiere, 1990). For speed learning, the main network is fixed during learning; only the added unit is made to learn, so as to compensate for residual error. The output error decreases with the number of hidden units added. The larger the decrease per hidden unit, the smaller the network that can be constructed. The decrease per hidden unit, that is, the performance of the hidden unit, can be theoretically evaluated for the network whose output units have a linear function (Fujita, 1992). In this case, it is possible to clarify the quantitative relation between the number of hidden units and the network output error. The statistical evaluation of this relation gives valuable information about the efficiency of stochastic search for hidden units of good performance, as will be mentioned later. This information is different from such a kind of information as the lower or upper bound of the capacity of the threshold logic network implementing an arbitrary dichotomy as evaluated by Baum (1988).

This paper describes statistical estimation of the number of hidden units for feedforward neural networks. We first derive the decrease in output error per hidden unit based on the least-squares approximation (restatement of the paper by Fujita, 1992). Then we theoretically estimate the expected

* Requests for reprints should be sent to Osamu Fujita. E-mail: futaro @ba.2.so-net.ne.jp He is now with Justsystem Corporation.

PII: S0893-6080(98)00043-4

maximum value of the decrease as the largest value in the great number of samples from an ideal distribution of the hidden unit. In other words, the best hidden unit is selected out of a finite number of candidate units that have the various functions of hidden units. The expected largest value depends on not only the number of candidate units but also the number of learning data sets which represents the complexity and difficulty of learning tasks. The expected largest decrease per hidden unit is used to estimate the total number of hidden units required to reduce the output error to a desired value. Finally, based on the theoretical estimation, the results of computer simulation for an actual learning task are discussed. We show that the relationship between the decrease in output error and the number of the candidate units represents the efficiency of the search for good hidden units. We also show that it has close relation with the nonlinearity of the hidden unit that is an ability to produce various outputs.

2. Feedforward neural network model

In this section, a feedforward neural network model, consisting of a linear output unit and nonlinear hidden units, is described mathematically as a nonlinear transformation of input matrix data into an output vector. Each unit is represented by a vector whose components are its output values for all learning data. The vectors of nonlinear hidden units span a vector space. This vector space should be as close to the desired output vector of the output unit as possible. The output error is defined by the minimum distance between the desired output vector and the actual output vector in the vector space, which is based on the least-squares approximation. The performance of each hidden unit is evaluated as to its contribution to the decrease in the output error (Fujita, 1992).

Let us consider an n-input–1-output feedforward neural network that consists of a linear output unit and m nonlinear hidden units. Suppose that K data sets for both the inputs and the desired output are given. The input data sets are represented by $K \times n$ matrix $\mathbf{X}$, and the desired output data sets are represented by K-dimensional column vector $\mathbf{z}$. The outputs of the hidden units are represented by $K \times m$ matrix $\mathbf{H}$, which is generally a nonlinear function of $\mathbf{X}$, as follows:

$$\mathbf{H} = f(\mathbf{X}), \tag{1}$$

where f denotes the mapping $\mathbf{X} \rightarrow \mathbf{H}$. The column space of $\mathbf{H}$ can be outside of the column space of $\mathbf{X}$ because of the nonlinearity of f.

In conventional neural networks, there are two types of interconnection between hidden units. Let us call the one a layered network and the other a cascaded network in this paper. For the layered network that has only one hidden layer, there is no interconnection between hidden units, and so the output vector of the jth hidden unit, $\mathbf{h}_j$ (the jth column vector of $\mathbf{H}$) is produced by

$$\mathbf{h}_j = f(b\mathbf{X}\,\mathbf{w}_{\mathbf{h}_j}), \tag{2}$$

where f is a nonlinear function for each component such as sin or tanh function, $\mathbf{w}_{\mathbf{h}_j}$ is an n-dimensional column vector as a weight vector, and b is a coefficient that represents the nonlinearity of the hidden unit used for discussion in Section 4. (In general, it is not necessary to notate b explicitly because it can be included in $\mathbf{w}_{\mathbf{h}_j}$.) For the cascaded network, $\mathbf{h}_j$ is given by

$$\mathbf{h}_j = f(b[\mathbf{X}\mathbf{H}_{j-1}]\mathbf{w}_{\mathbf{h}_j}), \tag{3}$$

where $[\mathbf{X}\mathbf{H}_{j-1}]$ is the augmented matrix consisting of $\mathbf{X}$ and $\mathbf{H}_{j\text{-}1} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_{j-1}]$, and $\mathbf{w}_{\mathbf{h}_j}$ is the $(n+j-1)$-dimensional weight vector. The output of the hidden unit in the cascaded network depends on the output of the other hidden units added previously.

Let y be an actual output vector produced by the linear output unit of the network as a linear combination of the column vectors of $\mathbf{H}$ i.e.

$$\mathbf{y} = \mathbf{H}\mathbf{w}, \tag{4}$$

where $\mathbf{w}$ is an m-dimensional weight vector. Thus, $\mathbf{y}$ satisfies $\mathbf{y} \in L[\mathbf{H}]$ where $L[\mathbf{H}]$ denotes the column space of the matrix $\mathbf{H}$. Let $\mathbf{y}_0$ be an optimum vector such that it minimizes the sum of squared output errors,

$$\|\mathbf{z} - \mathbf{y}\|^2 = \sum_{i=1}^{K} (z_i - y_i)^2, \tag{5}$$

where $\|\mathbf{y}\|$ is the Euclidean norm of $\mathbf{y}$. According to the theory of the least squares approximation, the optimum $\mathbf{y}_0$ is given by

$$\mathbf{y}_0 = \mathbf{P}\mathbf{z}, \tag{6}$$

where $\mathbf{P}$ is the projection matrix onto $L[\mathbf{H}]$. If $\mathbf{H}$ consists of linearly independent columns only, then $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ is non-singular and $\mathbf{P}$ is expressed by

$$\mathbf{P} = \mathbf{H}\,(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\,\mathbf{H}^{\mathrm{T}}. \tag{7}$$

Hence, the least sum of squared output errors is expressed by

$$\|\mathbf{z} - \mathbf{P}\mathbf{z}\|^2 = \|\mathbf{P}_{\mathrm{c}}\mathbf{z}\|^2 = \mathbf{z}^{\mathrm{T}}\mathbf{P}_{\mathrm{c}}\mathbf{z} \tag{8}$$

where $\mathbf{P}_{\mathrm{c}} = \mathbf{I} - \mathbf{P}$ and $\mathbf{I}$ is the identity matrix. $\mathbf{P}_{\mathrm{c}}$ is the projection matrix onto $L^{\perp}[\mathbf{H}]$, which is the orthogonal complement of $L[\mathbf{H}]$. It has two basic properties, $\mathbf{P}_{\mathrm{c}}^2 = \mathbf{P}_{\mathrm{c}}$ and $\mathbf{P}_{\mathrm{c}}^{\mathrm{T}} = \mathbf{P}_{\mathrm{c}}$, in common with $\mathbf{P}$. The optimum weight vector is given by

$$\mathbf{w}_0 = \mathbf{H}^{\dagger}\mathbf{z} \tag{9}$$

where $\mathbf{H}^{\dagger} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}$, i.e. the pseudoinverse of $\mathbf{H}$.

Let us consider the case in which a new hidden unit is added. Let $\mathbf{h}$ be the output vector of the added hidden unit. The column space to which $\mathbf{y}$ belongs is expanded by one dimension from $L(\mathbf{H})$ to $L([\mathbf{H}\mathbf{h}])$, where $[\mathbf{H}\mathbf{h}]$ is an augmented matrix. Such a space expansion brings about the decrease of the minimum value of $\|\mathbf{z} - \mathbf{y}\|^2$ from $\mathbf{z}^{\mathrm{T}}\mathbf{P}_{\mathrm{c}}\mathbf{z}$ to $\mathbf{z}^{\mathrm{T}}\mathbf{P}_{\mathrm{c}}'\mathbf{z}$ where $\mathbf{P}_{\mathrm{c}}'$ is the projection matrix onto $L^{\perp}[\mathbf{Hh}]$, i.e.

$$\begin{aligned}\mathbf{P}_{\mathrm{c}}' &= I - [\mathbf{Hh}]([\mathbf{Hh}]^{\mathrm{T}}[\mathbf{Hh}])^{-1}[\mathbf{Hh}]^{\mathrm{T}} \\ &= \mathbf{P}_{\mathrm{c}} - \mathbf{P}_{\mathrm{c}}\mathbf{h}(\mathbf{h}^{\mathrm{T}}\mathbf{P}_{\mathrm{c}}\mathbf{h})^{-1}\mathbf{h}^{\mathrm{T}}\mathbf{P}_{\mathrm{c}}. \end{aligned} \tag{10}$$

Let Δ be the decrease of the output error. Using Eq. (10), Δ is expressed by

$$\Delta = \mathbf{z}^{\mathrm{T}}\mathbf{p}_{\mathrm{c}}\mathbf{z} - \mathbf{z}^{\mathrm{T}}\mathbf{P}_{\mathrm{c}}'\mathbf{z} = \mathbf{z}^{\mathrm{T}}\mathbf{P}_{\mathrm{c}}\mathbf{h}(\mathbf{h}^{\mathrm{T}}\mathbf{P}_{\mathrm{c}}\mathbf{h})^{-1}\mathbf{h}^{\mathrm{T}}\mathbf{P}_{\mathrm{c}}\mathbf{z} = \frac{(\mathbf{z}^{\mathrm{T}}\mathbf{P}_{\mathrm{c}}\mathbf{h})^2}{(\mathbf{h}^{\mathrm{T}}\mathbf{P}_{\mathrm{c}}\mathbf{h})} \tag{11}$$

The $\mathbf{h}$ that maximizes the Δ is the best one for reducing the output error.

The network can be constructed as small as possible by adding the best **h**s one by one. The cascaded network can be generally smaller than the three-layered network. The overall network size is satisfactory in either network, although it is not necessarily minimum in comparison with the case in which all **h**s are optimized simultaneously.

3. Statistical estimation of the number of hidden units

The number of required hidden units for reducing the output error depends on how large the Δ of each unit can be. In this section, the expected largest value of Δ is estimated statistically, based on a certain supposition that the largest value of Δ is obtained by random search for **h**.

For the convenience of theoretical treatment, Δ is rewritten as

$$\Delta = \left(\frac{\mathbf{P}_{\mathrm{c}}\mathbf{z}}{\|\mathbf{P}_{\mathrm{c}}\mathbf{z}\|}, \frac{\mathbf{P}_{\mathrm{c}}\mathbf{h}}{\|\mathbf{P}_{\mathrm{c}}\mathbf{h}\|}\right)^2 \|\mathbf{P}_{\mathrm{c}}\mathbf{z}\|^2, \tag{12}$$

where the term within the parentheses denotes the inner product. This inner product corresponds to the inner product $(\mathbf{u}, \mathbf{v})$ of $(K-m)$-dimensional unit vectors $\mathbf{u}$ and $\mathbf{v}$ because of the projection $\mathbf{P}_{\mathrm{c}}$. In order to estimate the expected largest value of Δ, let us consider the distribution of $r = (\mathbf{u}, \mathbf{v})^2$ in the range of $0 \le r \le 1$. Suppose that $\mathbf{u}$ is constant and $\mathbf{v}$ is uniformly distributed on the surface of the $(K-m)$-dimensional hypersphere. In this case, the cumulative distribution function $\Phi(r)$ can be expressed by the beta distribution as follows:

$$\Phi(r) = cB_r\left(\frac{1}{2}, \frac{K-m-1}{2}\right), \tag{13}$$

where B_r is the incomplete beta function and c is the inverse of the beta function $B(1/2, (K-m-1/2)$ (see Appendix A). In this distribution, how large can r be, if candidate vectors for $\mathbf{v}$ are obtained by random sampling of unit vectors? Let s be the number of samples of $\mathbf{v}$, and r_s be the largest value of r in those samples. The distribution density function ψ_s of r_s is given by (Gumbel, 1958),

$$\psi_s(r_s) = s\Phi^{s-1}(r_s)\phi(r_s), \tag{14}$$

where ϕ is the distribution density function of Φ. The expected value of r_s is

$$E(r_s) = \int_0^1 r\psi_s(r)\,\mathrm{d}r = 1 - \int_0^1 \Phi^s(r)\,\mathrm{d}r, \tag{15}$$

and it is expressed by the following inequality (see Appendix B),

$$E(r_s) > 1 - \left[\frac{K-m-1}{2cs}\right]^{\frac{2}{K-m-1}} \Gamma\left(\frac{K-m+1}{K-m-1}\right), \tag{16}$$

where $\Gamma()$ is the gamma function. The right side gives a close approximation to $E(r_s)$ for $K-m \gg 1$ and $s \gg 1$ as follows:

$$E(r_s) \approx 1 - \alpha s^{-\frac{2}{K-m-1}} \tag{17}$$

where α is the abbreviation of the coefficient given in Eq. (16) and is close to 1 for $K-m \gg 1$. The expected rate of the squared output error decreasing by one hidden unit is expressed by

$$E\left(\frac{\|\mathbf{P}'_{\mathrm{c}}\mathbf{z}\|^2}{\|\mathbf{P}_{\mathrm{c}}\mathbf{z}\|^2}\right) = 1 - E(r_s) \approx \alpha s^{-\frac{2}{K-m-1}}. \tag{18}$$

The squared output error $\|\mathbf{P}_{\mathrm{c}}^{(m)}\mathbf{z}\|^2$ for m hidden units added one by one can thus be estimated as follows:

$$E\left(\frac{\|\mathbf{P}_{\mathrm{c}}^{(m)}\mathbf{z}\|^2}{\|\mathbf{P}_{\mathrm{c}}^{(0)}\mathbf{z}\|^2}\right) \approx \alpha^m \prod_{j=1}^{m} s^{-\frac{2}{K-j-1}} \approx s^{-\frac{2m}{K}}. \tag{19}$$

Therefore, the number of required hidden units is estimated approximately as

$$m \approx \frac{K\log(\|\mathbf{P}_{\mathrm{c}}^{(0)}\mathbf{z}\|/C)}{\log s}, \tag{20}$$

where C is the criterion of the allowable output error.

4. Computer simulation and discussion

4.1. Largest value of Δ

The actual value of $E(r_s)$ can be examined experimentally by obtaining the largest value of pseudo-random numbers from the beta distribution, which can be generated by the method given by Atkinson (1979). Fig. 1 shows the distribution of 100 samples of r_s for each value of $K-m$ and s. This shows that the theoretical estimation, Eq. (17), can be used as an approximation. The theoretical value is slightly less than the experimental value; this outcome agrees with the inequality in Eq. (16). (Based on an empirical correction, it was found that a better approximation can be obtained by changing α to $(\alpha + 2)/3$. This correction, however, has no intrinsic meaning and serves only to provide a better fit with the experimental data.)

The theoretical estimation is no more than an approximation. The theory in Section 3 is developed on the basis of the supposition that $\mathbf{P}_c\mathbf{h}/\|\mathbf{P}_c\mathbf{h}\|$ is uniformly distributed on the hypersphere. In practice, however, the distribution of **h** not only depends on the distribution of input vectors and weights, but also varies with the nonlinear function f of the hidden unit. Besides, the projection of **h** onto $L^{\perp}[\mathbf{H}]$ makes the situation complex. Even if the distribution of **h** is determined exactly, the distribution of $\mathbf{P}_c\mathbf{h}/\|\mathbf{P}_c\mathbf{h}\|$ is localized in some cases, and dispersed in other cases, depending on $\mathbf{P}_c$. Since the distribution of $\mathbf{P}_c\mathbf{h}/\|\mathbf{P}_c\mathbf{h}\|$ cannot be determined exactly, it is natural for the first step of approximation to assume that $\mathbf{P}_c\mathbf{h}/\|\mathbf{P}_c\mathbf{h}\|$ is, theoretically, uniformly distributed on the hyperspherical surface. However, it is important to examine whether this supposition is good or not for more practical conditions.

For most of neural network models, the output value is restricted to a certain range, for example, $[-1, 1]$ or $(0, 1)$, and so the output vector **h**, whose K components have output values such as these, is restricted in the K-dimensional hypercube. If **h** is uniformly distributed in the hypercube, the distribution of $\mathbf{P}_c\mathbf{h}/\|\mathbf{P}_c\mathbf{h}\|$ is probably non-uniform on the $(K-m)$-dimensional hypersphere in $L^{\perp}[\mathbf{H}]$ and has an anisotropy due to the projection of the hypercube. It is worthwhile examining the largest values of Δ by computer simulation for such a non-uniform distribution of $\mathbf{P}_c\mathbf{h}/\|\mathbf{P}_c\mathbf{h}\|$. For example, suppose that $\mathbf{P}_c\mathbf{h}$ is uniformly distributed in the $(K-m)$-dimensional hypercube (although such a case rarely occurs in practice). Fig. 2 shows the distribution of 100 samples of r_s for the non-uniform distribution of $\mathbf{P}_c\mathbf{h}/\|\mathbf{P}_c\mathbf{h}\|$ in comparison with that for the uniform distribution of $\mathbf{P}_c\mathbf{h}/\|\mathbf{P}_c\mathbf{h}\|$ shown in Fig. 1. Each r_s is obtained as the largest value in s samples of $(\mathbf{u}, \mathbf{v})^2$ for one random sample of **u** and s random samples of **v**. The $(K-m)$-dimensional unit vectors **u** and **v** are produced by $\xi/\|\xi\|$ where ξ is a $(K-m)$-dimensional vector whose components $\xi_i\ (i = 1, \cdots, K-m)$ are pseudo-random numbers uniformly distributed in the range $[-1, 1]$. In addition, Fig. 2 also shows the case that **u** and **v** are random binary vectors such that $\xi_i = \{-1, 1\}$.

The results of the computer simulation show that the distribution of r_s for the non-uniform distribution of $\mathbf{P}_c\mathbf{h}/\|\mathbf{P}_c\mathbf{h}\|$ is almost equal to that for the uniform distribution. The reason can be explained as follows. The distribution density of $\xi/\|\xi\|$ is high in the direction of the corners of the hypercube and very low in the direction of the coordinate axes. There are 2^{K-m} highest-density points that are symmetrically and uniformly distributed on the hyperspherical surface. The frequency of condensation and rarefaction is 2^{K-m}. When the number of samples is much less than 2^{K-m}, the distribution of the samples is too sparse to reflect the anisotropy, and can be regarded as approximately uniform on the hyperspherical surface.

Using this fact, under such a condition as $s \ll 2^{K-m}$, there is another approach to approximation of the largest value of Δ based on the conventional theoretical statistics of extremes in the normal distribution (Appendix C). The distribution of $(\mathbf{u}, \mathbf{v})$ for random binary vectors can be approximated by a kind of the binomial distribution and by the normal distribution after all. According to the conventional theory (Gumbel, 1958), the most probable largest value (that is not the expected largest value) can be

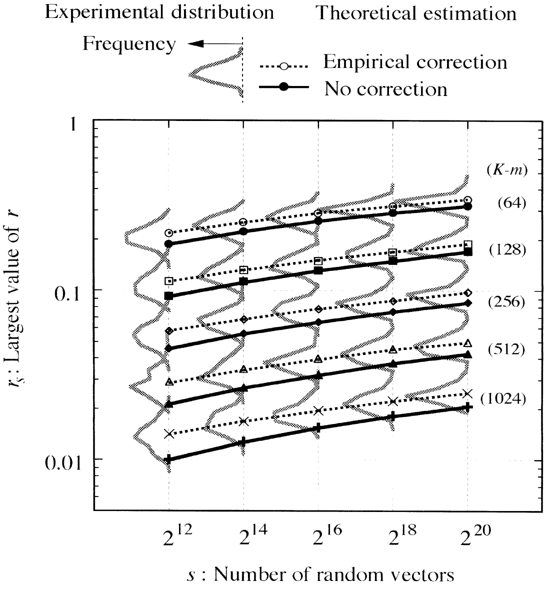

Fig. 1. Estimation of the largest value r_s with respect to s and $K-m$. The experimental distribution of 100 values of r_s for each condition is obtained by computer simulation in which the values of r are generated as pseudo-random numbers from the beta distribution. Theoretical estimation is based on the approximation shown in Eq. (17).

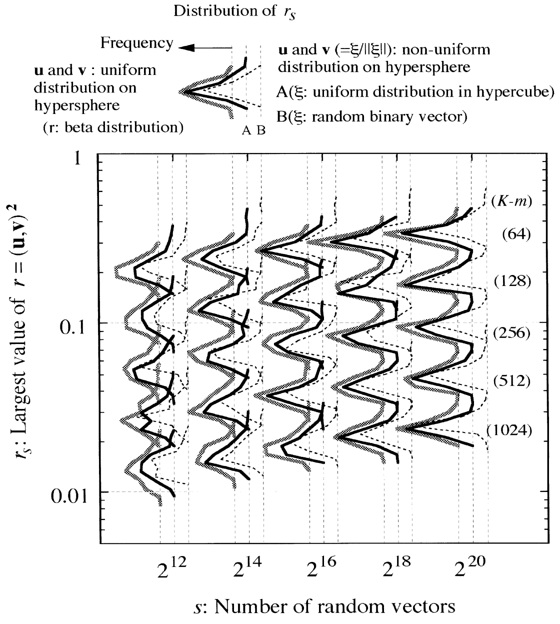

Fig. 2. The distribution of the largest values of $r = (\mathbf{u}, \mathbf{v})^2$ in comparison with r_s shown in Fig. 1. The $(K-m)$-dimensional unit vectors **u** and **v** are produced by $\xi/\|\xi\|$ where ξ is a $(K-m)$-dimensional vector whose components are pseudo-random numbers uniformly distributed in [-1, 1] for case (A) and pseudo-random binary numbers of $\{-1, 1\}$ for case (B).

approximated by

$$\frac{2\log(s/\sqrt{2\pi})}{K-m}. \tag{21}$$

This approximation also fits the experimental data well, but it should not be used for an larger value. In an extreme case, the approximate value of Eq. (21) can be more than 1, which violates the upper limit such that $(\mathbf{u}, \mathbf{v})^2 \leq 1$.

4.2. Actual decrease of output error

According to Eq. (19), the output error decreases with the number of hidden units by an almost constant rate, if s is constant. Such cases are often observed in practice, even if the distribution of $\mathbf{P}_c\mathbf{h}/\|\mathbf{P}_c\mathbf{h}\|$ is not uniform and does not satisfy the supposition used in the theory. For example, Fig. 3 displays curves showing the decreasing values of the squared output error obtained by computer simulation. The network is constructed by adding hidden units one by one. Each hidden unit is the best selected out of 64 candidates. The vector **h** of each candidate unit is made to maximize Δ by modifying its weight vector **w** with a multi-start gradient ascent (hillclimbing) method. Thus, each **h** is the best selected out of 64 candidates that each possess the local maximum of Δ obtained through 100 steps of the gradient ascent method. The input and desired output data are given as random values, which means the input data space is irreducible. As shown in Eqs. (2) and (3), two types of neural networks—layered networks and cascaded networks—are examined, and two types of hidden unit activation functions—tanh and sin—are used.

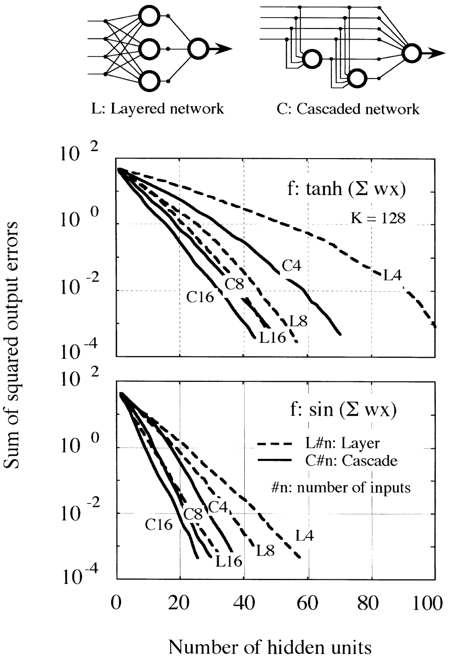

Fig. 3. Decreasing curves of the squared output error with the number of hidden units. Two types of networks—layered network and cascaded network—and two types of hidden unit activation functions—tanh and sin—are examined.

The output error decreases exponentially under any conditions. This means that the rate of decrease per hidden unit is almost constant. Under detailed observation, however, the rate of decrease in fact increases with the number of inputs and the number of hidden units added, and varies with the types of network and the nonlinear function. The rate of decrease of the cascaded network is larger than that of the layered network, and that of the sin-unit is larger than that of the tanh-unit.

The actual rate of decrease of the output error is closely related to the parameter s. Based on Eq. (19), the value of s can be estimated inversely from the actual rate of decrease per hidden unit obtained from the results of the computer simulation, as follows:

$$s \approx \left(\frac{\|\mathbf{P}_c^{(0)}\mathbf{z}\|}{\|\mathbf{P}_c^{(m)}\mathbf{z}\|}\right)^{\frac{K}{m}}. \tag{22}$$

In Eq. (22), the estimated value of s represents how many steps might be required for determining **h** for one hidden unit if random search were used instead. According to the computer simulation shown in Fig. 3, for example, s is estimated at about 10^7, which means that it might take 10^7 steps to search for each **h** if the random search method were

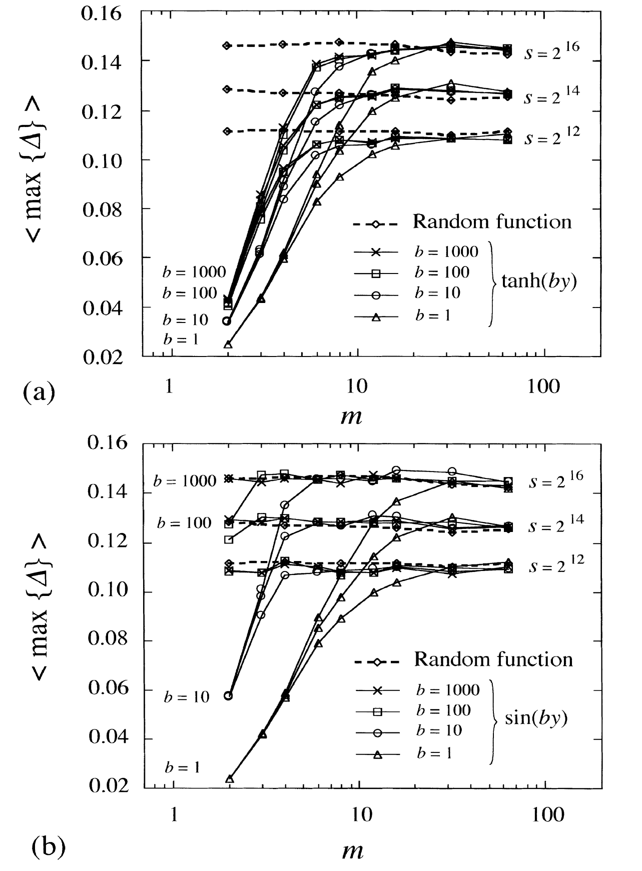

Fig. 4. Dependence of the average of max{Δ} upon s and m. The dimension of the subspace of $\mathbf{P}_c\mathbf{h}$ is set constant as $K - m = 128$. Two types of hidden unit activation functions—tanh and sin—are compared with a random function.

used for determining **h** in order to construct a similarly-sized network. In the computer simulation using the gradient ascent method with multi-start, however, the total number of learning steps for determining one hidden unit is 6400 (64 trials of a 100-step search from various random starting points). The cost of one step for the gradient ascent method is only a few times greater than that for the random search. This implies that **h** obtained by searching only 10^4 points by gradient search is equivalent to that obtained by searching 10^7 points by random search. This result shows that the gradient ascent method is much more efficient for searching than a simple random search in this case. Thus, the evaluation of s based on Eq. (22) shows the efficiency of the search in an actual learning process for determining **h** for one hidden unit.

The rate of decrease in output error depends on how appropriately the vector **h** is determined. Hence the rate depends not only upon the efficiency of the search but also upon the distribution of **h**, i.e. the diversity of **h** produced by the nonlinear function f. To clarify in detail, Fig. 4 shows the actual dependence of Δ on s and m, where $K - m$ is constant for various m. Two types of nonlinear functions for f—tanh and sin—are compared with a random function as a standard of comparison, where the random function produces a random vector for **h**. To investigate the performance of $\mathbf{h} = f(b\mathbf{X}\mathbf{w_h})$ from a statistical standpoint, s random vectors of $\mathbf{w_h}$ are given for each of 100 random examples of **z** and **X**. In this case, it is assumed that all inputs are directly connected with the output unit, i.e. $\mathbf{H} = \mathbf{X}$ and $m = n$. The plotted points indicate $\langle r_s \rangle$, that is, $\langle \max\{\Delta\} \rangle$ for $\|\mathbf{P}_c\mathbf{z}\| = 1$, where $\langle\rangle$ denotes the average of 100 values of $\max\{\Delta\}$, and $\max\{\Delta\}$ is the largest value of Δ in the s samples of $\Delta(\mathbf{h})$. The main characteristic is that $< \max\{\Delta\} >$ increases with m and saturates at the same level as the random function. This implies that the diversity of the nonlinear mapping f of **X** to **h** increases with the dimension of the column space of **X** (the input space), and so the number of hidden units required depends indirectly on the number of inputs to each hidden unit (although n is not explicitly notated in Eq. (20)). This is the reason that the rate of decrease of the output error gets larger with the number of hidden units for the cascaded network, but the diversity and nonlinearity of the random function must be highest, and so $\langle \max\{\Delta\} \rangle$ saturates finally at the level of the random function, as shown in Fig. 4. This effect can easily be seen for the tanh units that have weak nonlinearity.

The dependence of the diversity of **h** upon the nonlinearity of f is also shown in Fig. 3. The rate of decrease of the sin unit is larger than that of the tanh unit. This is because the nonlinearity of the sin function is higher than that of the tanh function. The effect of such a difference between sin and tanh is clearly seen in Fig. 4(a) and (b). In this case, the nonlinearity of f is indicated by the coefficient b, which is used as $\sin(by_\mathrm{h})$ or $\tanh(by_\mathrm{h})$ where $y_h = \mathbf{X}\mathbf{w_h}$, because b has a close relation with the number of folds (or bend) in the curve of the nonlinear function in the unit range of y_h. For the sin unit, $\langle \max\{\Delta\} \rangle$ for small m increases with b up to the level of the random function. By comparison, the tanh unit saturates at a level much lower than that of the random function. These facts can be explained as follows. If b is small, both functions are approximately linear for small y. As b becomes large, the number of folds of the sin function increases infinitely, but that of the tanh function does not increases by more than two. The higher the nonlinearity is, the wider is the range of directions that h can cover in its vector space, and therefore the better is the h that can be selected. In that sense, the sin function is better than the tanh function for constructing as small a network as possible. It does have a drawback, however, in that it is less robust; this results in high noise sensitivity.

The parameter s obtained experimentally can be used for evaluating the efficiency of learning, if the above-mentioned effect of the nonlinearity of f is taken into account. Once s is evaluated for various sets of conditions, the number of hidden units can be roughly estimated by using s for a similar set of conditions. There are two points of basic importance in the use of s. One is that the optimization problem is simply formulated by the maximization of the inner product of unit vectors that have restrictions. The other is that s means the number of steps required for random search and it can be used as a standard of comparison with respect to the efficiency of search. Although the nonlinearity of f exerts influence upon the distribution of $\phi(r)$, this effect can be regarded as negligible in practice when dimensions of vector spaces such as n, K and m are very large in comparison with $\log s$. Under such conditions, it is supposed that s random points of $f(\mathbf{X})$ are projected on $L^{\perp}[\mathbf{X}]$ with sparse and approximately uniform distribution. Conversely, it can be said that evaluation of s under other conditions shows the effect of the nonlinearity of f on producing various output vectors of the hidden unit.

5. Summary

The number of required hidden units is statistically estimated for feedforward neural networks constructed by adding hidden units one by one. This number is approximately proportional to the number of learning data sets, the logarithm of the decreasing rate of the output error, and the inverse of $\log s$. This parameter s is introduced, for theoretical estimation, as the total number of candidates that are randomly searched for the optimum hidden unit. This formulation can be applied for evaluating an actual learning method. In such a case, s can be considered as a parameter that represents the efficiency of the search for better hidden units and is equivalent to the number of steps required for determining one hidden unit by random search. The computer simulation shows that the output error decreases exponentially with the number of hidden units; this agrees with the theoretical results. Although the decrease in the output error depends not only upon the number of inputs but also upon the nonlinearity of hidden units, these effects are negligible when the dimension of the vector space is very large.

Acknowledgements

The author wishes to thank Professor Tetsuo Nishi and anonymous reviewers for their comments and suggestion.

Appendix A The cumulative distribution function $\Phi(r)$

$\Phi(r)$ is expressed by

$$\Phi(r) = P[r \geq v(\mathbf{u}, \mathbf{v})^2] \tag{A1}$$

where $P[e]$ denotes the probability of event e. The vectors $\mathbf{u}$ and $\mathbf{v}$ are supposed to be uniformly distributed on the surface of the $(K - m)$-dimensional hypersphere. On this supposition, we may put $\mathbf{u} = [1\ 0\ \cdots\ 0]$ only if $\mathbf{v}$ is uniformly distributed on the hyperspherical surface, which does not change the distribution of r. Thus, $\Phi(r)$ is represented by

$$\Phi(r) = P[r \geq v_1^2 | \sum_{i=1}^{K-m} v_i^2 = 1] = P[\sqrt{r} \geq v_1 \geq -\sqrt{r} | \sum_{i=1}^{K-m} v_i^2 = 1]. \tag{A2}$$

This probability can be estimated by using a function $\chi(v; A)$, which is equal to 1 for $v \in A$ and 0 for $v \notin A$, as follows:

$$\begin{aligned}
\Phi(r) &= \lim_{\varepsilon \to 0} \frac{\int_{-\sqrt{r}}^{\sqrt{r}} \mathrm{d}v_1 \int_{-1}^{1} \cdots \int_{-1}^{1} \chi(\{v_i\}; \{1 - v_1^2 \geq \sum_{i=2}^{K-m} v_i^2 > 1 - v_1^2 - \varepsilon\})\, \mathrm{d}v_2 \cdots \mathrm{d}v_k}{\int_{-1}^{1} .. \int_{-1}^{1} \chi(\{v_i\}; \{1 \geq \sum_{i=1}^{K-m} v_i^2 > 1 - \varepsilon\})\, \mathrm{d}v_1 \cdots \mathrm{d}v_k} \\
&= \lim_{\varepsilon \to 0} \frac{\int_{-\sqrt{r}}^{\sqrt{r}} b\left[(1 - v_1^2)^{(K-m-1)/2} - (1 - v_1^2 - \varepsilon)^{(K-m-1)/2}\right] \mathrm{d}v_1}{a\left[1 - (1 - \varepsilon)^{(K-m)/2}\right]} \\
&= \lim_{\varepsilon \to 0} c \int_{-\sqrt{r}}^{\sqrt{r}} \left[(1 - v_1^2)^{(K-m-3)/2} + O(\varepsilon)\right] \mathrm{d}v_1 \\
&= c \int_{-\sqrt{r}}^{\sqrt{r}} (1 - v_1^2)^{(K-m-3)/2}\, \mathrm{d}v_1 \\
&= c \int_{0}^{r} t^{-1/2} (1 - t)^{(K-m-3)/2}\, \mathrm{d}t \\
&= c B_r\left(\frac{1}{2}, \frac{K - m - 1}{2}\right),
\end{aligned} \tag{A3}$$

where B_r is the incomplete beta function and c is a normalization constant equal to the inverse of the beta function $B(1/2, (K - m - 1)/2)$.

Appendix B The expected largest value of $E(r_s)$

$E(r_s)$ is expressed by

$$E(r_s) = \int_0^1 r \varphi_s(r)\, \mathrm{d}r = 1 - \int_0^1 \Phi^s(r)\, \mathrm{d}r. \tag{A4}$$

$\Phi(r)$ can be rewritten as

$$\Phi(r) = 1 - c \int_0^{1-r} (1 - t)^{-1/2} t^{(k-m-3)/2}\, \mathrm{d}t. \tag{A5}$$

For approximation, let us introduce the following function,

$$\begin{aligned}
G(r) &= 1 - c \int_0^{1-r} t^{(k-m-3)/2}\, \mathrm{d}t \\
&= 1 - \frac{2c}{k - m - 1}(1 - r)^{(k-m-1)/2},
\end{aligned} \tag{A6}$$

which satisfies $\Phi(r) < G(r)$ and therefore $\Phi^s(r) < G^s(r)$. Let

G_s^{-1} be the inverse of $q = G_s(r)$ as follows:

$$r = G_s^{-1}(q) = 1\left[\frac{K-m-1}{2c}\left(1-q^{1/s}\right)\right]^{2/(K-m-1)}. \qquad \text{(A7)}$$

Putting $q_0 = G_s(0)$, $E(r_s)$ can be expressed by the following inequality:

$$E(r_s) = 1 - \int_0^1 \Phi^s(r)\,\mathrm{d}r > 1 - \int_0^1 G^s(r)\,\mathrm{d}r = \int_{q_0}^1 G_s^{-1}(q)\,\mathrm{d}q. \qquad \text{(A8)}$$

Using an inequality $1 - q^{1/s} < -(1/s)\ln q$ for $q \in [q_0,\ 1]$, we have

$$\begin{aligned} E(r_s) &> 1 - \left(\frac{K-m-1}{2c}\right)^{2/(K-m-1)} \\ &\quad \times \int_{q_0}^1 \left(-\frac{1}{s}\ln q\right)^{2/(K-m-1)} \mathrm{d}q = 1 \\ &\quad - \left(\frac{K-m-1}{2cs}\right)^{2/(K-m-1)} \\ &\quad \times \int_{-\ln q_0}^{0} -\mathrm{e}^{-t} t^{2/(K-m-1)}\,\mathrm{d}t = 1 \\ &\quad - \left(\frac{K-m-1}{2cs}\right)^{2/(K-m-1)} \\ &\quad \times \gamma\left(\frac{K-m-1}{K-m-1}, -\ln q_0\right) > 1 \\ &\quad - \left(\frac{K-m-1}{2cs}\right)^{2/(K-m-1)} \Gamma\left(\frac{K-m-1}{K-m-1}\right) \\ &= 1 - \alpha s^{-2/(K-m-1)}, \end{aligned} \qquad \text{(A9)}$$

where γ is the incomplete gamma function, Γ is the gamma function, and

$$\alpha = \left(\frac{K-m-1}{2c}\right)^{2/(K-m-1)} \Gamma\left(\frac{K-m-1}{K-m-1}\right). \qquad \text{(A10)}$$

The coefficient α is approximately equal to 1 for $K - m \gg 1$. The right and left sides of the inequality Eq. (A9) are asymptotically equal as $s \to \infty$.

Appendix C The most probable largest value in the normal distribution

Suppose that **u** and **v** are K-dimensional random binary vectors represented by $\xi/\|\xi\|$ where $\xi_i = \{-1, 1\}$ $(i = 1, \cdots, K)$. The distribution of (**u**, **v**) is represented by a kind of the binomial distribution. If K is large, the distribution is approximated by the normal distribution $N(0,\ 1/K)$. The largest value of $(\mathbf{u},\ \mathbf{v})^2$ is considered as the largest value in the sample of χ^2 distribution with 1 degree of freedom, but it is too difficult to simply express the expected largest value. For simple approximation, it is better to represent as the square of the largest value (**u**, **v**) in the normal distribution. According to the conventional theory (Gumbel, 1958), the characteristic largest value x_s in s samples is defined by $F(x_s) = 1 - 1/s$ where $F(x)$ is the distribution function. For the normal distribution $N(0,\ 1/K)$, x_s can be approximated as

$$\sqrt{\frac{2\log s}{K}\left\{1 - \frac{(\log 4\pi + \log\log s)}{2\log s}\right\}}. \qquad \text{(A11)}$$

As for the most probable largest value, u_s, it can be approximated as

$$\sqrt{\frac{2\log(s/\sqrt{2\pi})}{K}}. \qquad \text{(A12)}$$

Therefore, using the later approximation, the largest value of $(\mathbf{u},\ \mathbf{v})^2$ for $K - m$ dimensional vectors **u** and **v** can be approximated by

$$\frac{2\log(s/\sqrt{2\pi})}{K-m}. \qquad \text{(A13)}$$

References

Akaho, S., & Amari, S. (1990). On the capacity of three-layer networks. *Proceedings of the International Joint Conference on Neural Networks* (pp. 1–6), San Diego, CA, Vol. III.

Akaike H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, AC-19*, 716–723.

Ash, T. (1989). Dynamic node creation in backpropagation networks. *Proceedings of the International Joint Conference on Neural Networks* (p. 623), Washington, DC, Vol. 11.

Atkinson A.C. (1979). A family of switching algorithms for the computer generation of beta random variables. *Biometrica, 66*, 141–145.

Baum E.B. (1988). On the capabilities of multilayer perceptrons. *Journal of Complexity, 4*, 193–215.

Baum E.B., & Haussler D. (1989). What size net gives valid generalization?. *Neural Computation, 1*, 151–160.

Fahlman, S. E., & Lebiere, C. (1990). The cascade-correlation learning architecture. In D. S. Touretzky (Eds.), *Advances in Neural Information Processing Systems 2* (pp. 524–532). Morgan Kaufmann.

Fujita O. (1992). Optimization of the hidden unit function in feedforward neural networks. *Neural Networks, 5 (5)*, 755–764.

Gumbel, E. J. (1958). *Statistics of Extremes*. New York: Columbia University Press.

Hagiwara, M. (1990). Novel back propagation algorithm for reduction of hidden units and acceleration of convergence using artificial selection. *Proceedings of the International Joint Conference on Neural Networks* (pp. 625–630), San Diego, Vol. 1.

Hirose, Y., Yamashita, K. & Hijiya, S. (1989). Back propagation method which varies the number of hidden units. *1989 Spring National Convention Record* (p. 18). Japan: Institute of Electronics, Information and Communication Engineers, D-18, Vol. 7.

Kurita T. (1990). A method to determine the number of hidden units of three-layered neural networks by information criteria. *Transactions of the Institute of Electronics Information and Communication Engineers, J73-D-II*, 1872–1878. (in Japanese).

Moody, J. E. (1992). The effective number of parameters: an analysis of generalization and regularization in nonlinear learning systems. In J. E. Moody, S. J. Hanson, & R. P. Lippmann (Eds.), *Advances in Neural Information Processing Systems 4* (pp. 847–854). Morgan Kaufmann.