Communicated by Gerard Dreyfus

Exhaustive Learning

D. B. Schwartz
V. K. Samalam
GTE Laboratories, Waltham, MA 02254 USA

Sara A. Solla
J. S. Denker
AT&T Bell Laboratories, Holmdel, NJ 07733 USA

Exhaustive exploration of an ensemble of networks is used to model learning and generalization in layered neural networks. A simple Boolean learning problem involving networks with binary weights is numerically solved to obtain the entropy S_m and the average generalization ability G_m as a function of the size m of the training set. Learning curves G_m vs m are shown to depend solely on the distribution of generalization abilities over the ensemble of networks. Such distribution is determined prior to learning, and provides a *novel* theoretical tool for the *prediction* of network performance on a *specific* task.

1 Introduction

Layered networks are useful in their ability to implement input–output maps $y = f(x)$. The problem that arises is that of designing networks to implement a desired map $\tilde{f}$. Supervised learning searches for networks that satisfy the map $\tilde{f}$ on a restricted set of points, the training examples. An outstanding theoretical question is that of predicting the generalization ability of the resulting networks, defined as the ability to *correctly* extend the domain of the function beyond the training set.

Theoretical and predictive analysis of the performance of networks that are trained from examples are few (Denker et al. 1987; Carnevali and Patarnello 1987; Baum and Haussler 1989), in contrast to the large effort devoted to the experimental application and optimization of various learning algorithms. Such experimental results offer useful solutions to specific problems but shed little light on general theoretical issues, since the solutions are heavily influenced by the intrinsic dynamics of the chosen algorithm. A theoretical analysis based on the global and statistical properties of an ensemble of networks (Denker et al. 1987; Carnavali and Patarnello 1987) requires reliable information about such ensemble, unbiased by the peculiarities of the specific strategy adopted to search for appropriate networks within the ensemble.

Neural Computation **2**, 374–385 (1990) © 1990 Massachusetts Institute of Technology

Progress in the theoretical understanding of complex systems is often triggered by intuition obtained through carefully designed numerical experiments. We have therefore chosen a Boolean classification task involving low resolution weights, which enables us to explore the network ensemble exhaustively. Such unbiased search, although hardly useful as a practical tool, is free from the constraints intrinsic to current learning algorithms. It reveals the true properties of the network ensemble as determined by the choice of architecture, and is used here to monitor, without introducing any additional bias, the evolution of the ensemble through training with examples of the desired task.

The insight gained from the numerical experiments led to a theoretical analysis of supervised learning and the emergence of generalization ability, presented in Section 2 of this paper. The numerical experiments that motivated the theoretical framework are described in Section 3. An analysis of the numerical results according to the theory, as well as some applications of the theory to more realistic problems, are provided in Section 4.

2 Theoretical Framework

Consider an ensemble of layered networks with fixed architecture and varying couplings. Such ensemble is described by its configuration space $\{\mathbf{W}\}$: every point $\mathbf{W}$ is a list of values for all couplings needed to select a network design within the chosen architecture. The resulting network realizes a specific input-output function, $y = f_{\mathbf{W}}(\mathbf{x})$. For simplicity, consider Boolean functions $y \in \{0, 1\}$ on a Boolean $\mathbf{x} \in \{0, 1\}^N$ or real $\mathbf{x} \in \mathcal{R}^N$ domain.

A prior density $\rho_0(\mathbf{W})$ constrains the effective volume of configuration space to

$$Z_0 = \int d\mathbf{W} \rho_0(\mathbf{W}) \tag{2.1}$$

Regions corresponding to the implementation of the function f are identified by the masking function

$$\Theta_f(\mathbf{W}) = \begin{cases} 1 & \text{if} \quad f_{\mathbf{W}} = f \\ 0 & \text{if} \quad f_{\mathbf{W}} \neq f \end{cases}$$

and occupy a volume

$$Z(f) = \int d\mathbf{W} \rho_0(\mathbf{W}) \Theta_f(\mathbf{W}) \tag{2.2}$$

The specification of an architecture and its corresponding configuration space thus defines a probability on the space of functions:

$$P_0(f) = \frac{Z(f)}{Z_0} \tag{2.3}$$

which results from a full exploration of configuration space. $P_0(f)$ is the probability that a randomly chosen network in configuration space will realize the function f. The class of functions implementable by a given architecture is

$$\mathcal{F} = \{f | P_0(f) \neq 0\} \tag{2.4}$$

The entropy of the prior distribution

$$S_0 = -\sum_{\{f\}} P_0(f) \ln P_0(f) \tag{2.5}$$

is a measure of the functional diversity of the chosen architecture. The maximum value of $S_0 = \ln(n_{\mathcal{F}})$, where $n_{\mathcal{F}}$ is the number of functions in class $\mathcal{F}$, is attained when all realizable functions are equally likely, and corresponds to the uniform distribution, $P_0(f) = 1/n_{\mathcal{F}}$ for all $f \in \mathcal{F}$.

Supervised learning results in a monotonic reduction of the effective volume of configuration space. An example $\boldsymbol{\xi}^\alpha = (\mathbf{x}^\alpha, y^\alpha)$ of the desired function $\tilde{f}$ is learned by removing from $\mathcal{F}$ every function that contradicts it. A sequence of m input-output pairs $\boldsymbol{\xi}^\alpha = (\mathbf{x}^\alpha, y^\alpha)$, $1 \leq \alpha \leq m$, which are examples of $\tilde{f}$ thus defines a sequence of classes of functions,

$$\mathcal{F}_m \subseteq \mathcal{F}_{m-1} \subseteq \ldots \mathcal{F}_1 \subseteq \mathcal{F}$$

where every function $f \in \mathcal{F}_m$ correctly classifies all of the training examples $\boldsymbol{\xi}^\alpha$, $1 \leq \alpha \leq m$. The effective volume of configuration space is reduced to

$$Z_m = \int d\mathbf{W} \rho_0(\mathbf{W}) \sum_{f \in \mathcal{F}_m} \Theta_f(\mathbf{W}) \tag{2.6}$$

by learning a training set of size m.

The probability on the space of functions is modified by learning and becomes

$$P_m(f) = \frac{Z(f)}{Z_m} \tag{2.7}$$

$P_m(f)$ is the probability that f has not been eliminated by one of the m examples and is thus a member of $\mathcal{F}_m$. The total volume of configuration space occupied by functions $f \in \mathcal{F}_m$ is Z_m.

The entropy of the posterior distribution,

$$S_m = -\sum_{\{f\}} P_m(f) \ln P_m(f) \tag{2.8}$$

reflects the narrowing of the probability distribution: $S_m < S_0$. The entropy decrease $\eta_m = S_{m-1} - S_m$ defines the efficiency of learning the mth example.

Learning corresponds to a monotonic contraction of the effective volume of configuration space: $Z_m \subseteq Z_{m-1}$. Exhaustive learning, as defined here, leads to the complete exclusion of networks incompatible with each training example. Such error-free learning excludes the possibility of data so noisy as to contain intrinsic incompatibilities in the training set. A recent extension of the theory (Tishby et al. 1989) provides the tools to analyze the case of learning with error.

The entropy decrease $(S_0 - S_m)$ is the information gain, that is, the information extracted from the examples in the training set. The residual entropy S_m measures the functional diversity of the ensemble of trained networks. The optimal case of $S_m = 0$ corresponds to the elimination of all ambiguity about the function to be implemented. In general $S_m \neq 0$, and its value measures the lack of generalization ability of the trained networks.

A more detailed description of the generalization ability achieved by supervised learning is based on the generalization ability $g(f)$ of the individual functions $f \in \mathcal{F}$, defined as the probability that f will correctly classify a randomly chosen example of the desired function $\tilde{f}$. As an illustration of the intrinsic ability of f to reproduce $\tilde{f}$, consider the simple case of a Boolean function from N inputs onto 1 output. The function $\tilde{f}$ is specified by 2^N bits, indicating the output for every possible input. In this case

$$g(f) = \frac{2^N - d_{\text{H}}(f, \tilde{f})}{2^N} \tag{2.9}$$

where $d_{\text{H}}(f, \tilde{f})$ is the Hamming distance between f and $\tilde{f}$, that is, the number of bits by which their truth tables differ.

The survival probability $P_m(f)$ can be expressed recursively by noting that the probability of surviving a single additional example is on average just $g(f)$. Thus

$$P_m(f) = \frac{P_{m-1}(f)g(f)}{\sum\limits_{\{f'\}} P_{m-1}(f')g(f')} \tag{2.10}$$

where the denominator is required to maintain proper normalization. The recursion relation equation 2.10 is based on the assumption that $g(f)$ is independent of m, and thus it is valid provided m remains small compared to the total number of possible inputs $\{\mathrm{x}\}$. Such limitation is not severe: learning experiments are of interest when the network can indeed be trained with a set of examples that is a small subset of the total space.

The generalization ability of trained networks is an ensemble property described by the probability density

$$\rho_m(g) = \sum_{\{f\}} P_m(f)\delta[g - g(f)] \tag{2.11}$$

The product $\rho_m(g)dg$ is the probability of generating networks with generalization ability in the range $[g, g + dg]$ by training with m examples. The average generalization ability

$$G_m = \int_0^1 g\rho_m(g)dg \tag{2.12}$$

given by

$$G_m = \sum_{\{f\}} P_m(f)g(f) \tag{2.13}$$

is the probability that a randomly chosen surviving network will correctly classify an arbitrary test example, distinct from the m training examples.

The recursion relation equation 2.10 for $P_m(f)$ can be rewritten as

$$P_m(f) = \frac{P_{m-1}(f)g(f)}{G_{m-1}} \tag{2.14}$$

and substituted onto equation 2.11 to yield

$$\rho_m(g) = \frac{1}{G_{m-1}} \sum_{\{f\}} P_{m-1}(f)g(f)\delta[g - g(f)] \tag{2.15}$$

or

$$\rho_m(g) = \frac{g\rho_{m-1}(g)}{\int_0^1 g'\rho_{m-1}(g')dg'} \tag{2.16}$$

The recursion relation equation 2.16 is a crucial result of this theoretical analysis, since it provides a fundamental tool to both analyze and predict the outcome of supervised learning. Iterative applications of equation 2.16 lead to the relation

$$\rho_m(g) = \frac{g^m\rho_0(g)}{\int_0^1 g'^m\rho_0(g')dg'} \tag{2.17}$$

The probability density $\rho_m(g)$ is thus fully determined by the initial distribution $\rho_0(g)$. Its average value G_m (equation 2.12), given by

$$G_m = \frac{\int_0^1 g^{m+1}\rho_0(g)dg}{\int_0^1 g^m\rho_0(g)dg} \tag{2.18}$$

is simply the ratio between the $(m + 1)$th and the mth moments of $\rho_0(g)$, and can be computed if $\rho_0(g)$ is given or estimated.

The entropy S_m (equation 2.8) and average generalization ability G_m (equation 2.12) are the fundamental tools to monitor the learning process. The picture that emerges is that of learning as a monotonic decrease of the effective volume of configuration space, measured by a monotonic entropy decrease with increasing m. The contraction is not arbitrary: it emphasizes regions of configuration space with intrinsically high generalization ability. The iterated convolution with g to obtain $\rho_m(g)$ from $\rho_0(g)$ (equation 2.17) results in an increasing bias toward $g = 1$, and a monotonic increase of the average generalization ability with increasing m.

3 Numerical Experiments

Consider a layered network with L levels of processing. The network architecture is specified by the number $\{N_\ell\}$, $0 \leq \ell \leq L$ of units per layer, and its configuration by the weights $\{W_{ij}^{(\ell)}\}$ and biases $\{W_i^{(\ell)}\}$ for $1 \leq \ell \leq L$, $1 \leq i \leq N_\ell$, $1 \leq j \leq N_{\ell-1}$. The configuration space $\{\mathbf{W}\}$, of dimensionality $D_{\mathbf{W}} = \sum_{\ell=1}^{L} N_\ell(1 + N_{\ell-1})$, describes a canonical ensemble of networks with fixed architecture and varying couplings.

Full explorations of configuration space are in general impractical due to the vast number of possible networks in $\{\mathbf{W}\}$ and the correspondingly large number $n_{\mathcal{F}}$ of realizable functions. Statistical sampling techniques are thus needed to extract reliable information on the prior distributions $P_0(f)$ and $\rho_0(g)$. Simplified problems with restricted architectures and binary weights $W_{ij}^{(\ell)} = \pm 1$ result in ensembles amenable to exhaustive exploration. Ensembles containing about a million networks have allowed here for the accurate computation of various ensemble averaged quantities, and led to the theoretical insight described in the preceding section.

Consider the contiguity problem (Denker et al. 1987; Solla 1989), a classification of binary input patterns $\mathbf{x} = (x_1, \ldots, x_N)$, $x_i = 0, 1$ for all $1 \leq i \leq N$. Periodic boundary conditions are imposed on the N-bit input vectors, so that the last bit is adjacent to the first. The patterns are classified according to the number k of blocks of 1's in the pattern. For example, for $N = 10$, $\mathbf{x} = (1110001111)$ corresponds to $k = 1$, $\mathbf{x} = (0110110111)$ to $k = 3$, and $\mathbf{x} = (0010011111)$ to $k = 2$. The task is simplified into a dichotomy: the two categories correspond to $k \leq k_0$ and $k > k_0$. This problem can be solved by an $L = 2$ layered network (Denker et al. 1987) with $N_0 = N_1 = N$ and $N_2 = 1$, and receptive fields of size 2.

In the numerical results reported here all processing units are thresholding units: their output is 1 or 0 according to whether their input is positive or negative. The bias $W^{(2)}{}_1$ of the output unit, the biases $W^{(1)}{}_i$ of the hidden units, and the weights $W^{(2)}{}_{1i}$ between hidden units and output unit are fixed at the values determined by the solution to the contiguity problem for $k_0 = 2$: $W^{(2)}{}_1 = -2.5$, and $W^{(1)}{}_i = -0.5$, $W^{(2)}{}_{1i} = 1$ for all $1 \leq i \leq N$. The only degrees of freedom are thus the couplings between input units and hidden units. A receptive field of size 2 corresponds to the only nonzero couplings being $W^{(1)}{}_{i,i}$ and $W^{(1)}{}_{i,i-1}$, providing input to each hidden unit $1 \leq i \leq N$ from two input units: the one immediately below, and the adjacent one to the left.

The configuration space corresponds to $W^{(1)}{}_{ij} = \pm 1$ for $j = i, i-1$ and $1 \leq i \leq N$. Even for such a simple example the configuration space is large: it consists of 2^{2N} distinct points. Two of them correspond to equivalent solutions to the contiguity problem: $W^{(1)}{}_{i,i} = +1$, $W^{(1)}{}_{i,i-1} = -1$, based on left-edge detection; and $W^{(1)}{}_{i,i} = -1$, $W^{(1)}{}_{i,i-1} = +1$, based on right-edge detection. The degeneracy of the remaining $(2^{2N} - 2)$ networks, that is to say to which extent they implement distinct functions, has not been investigated in depth.

The learning experiments are performed as follows: an explicit representation of the ensemble is constructed by listing all possible 2^{2N} networks. To generate a training set, randomly distributed examples within the 2^N points in input space are obtained by blocking in groups of N bits the output of a high quality random number generator. A training set is prepared by labeling subsequent examples $(\mathbf{x}^\alpha, y^\alpha)$. The αth example is learned by eliminating from the listing all the networks that misclassify it. The entropy S_m is estimated by the logarithm of the number of surviving networks. The number of surviving networks is an upper bound to the number of surviving functions, and the two quantities are monotonically related. The average generalization ability G_m is computed by testing each surviving network on a representative set of examples not included in the training set. The size of the testing set is chosen so as to guarantee a precision of at least 1% in the determination of G_m.

Results reported here correspond to N = 9, 10, and 11. Smaller values of N yield poor results due to limits in the available number of examples: there are only 256 possible inputs for N=8. Values of N larger than 11 exceed reasonable requirements in computer time and memory, even on a 64-Mbyte machine capable of 5×10^7 connections/sec.

Curves for the entropy S_m and the prediction error $\mathcal{E}_m = 1 - G_m$ as a function of the size m of the training set are shown in Figure 1 (for N=9) and Figure 2 (for N=9 and 11), respectively. The curves are averages over 1000 separate runs, the runs being distinguished by different sequences of training examples.

The prior distribution of generalization abilities $\rho_0(g)$ is computed by testing all networks in the initial list on a randomly chosen set of 300 examples, large enough to obtain the intrinsic generalization ability of each network with a precision of at least 6%. The accumulated histograms are shown in Figure 3 (for N= 9 and 11). The dependence of the average generalization ability G_m on the number m of training examples can be predicted from $\rho_0(g)$ according to equation 2.18. The predicted curve for N=11 is shown in Figure 4, and compared to the curve computed through direct measurement of the average generalization ability. Discrepancies are to be expected, since uncertainties in the estimation of

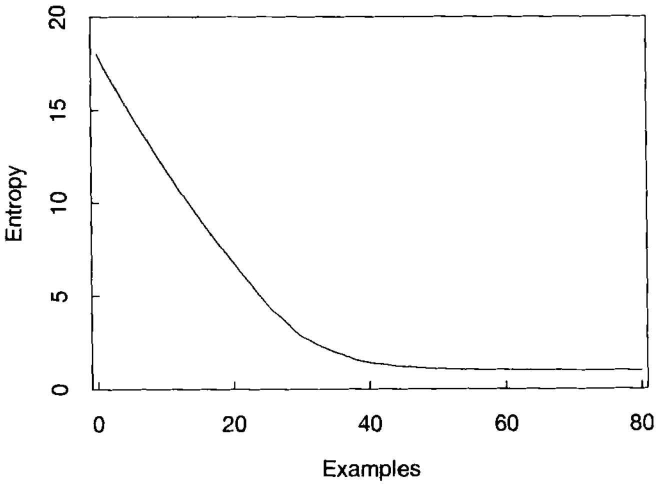

Figure 1: Numerical estimate of the ensemble entropy S_m as function of the size m of the training set for the contiguity problem, $N = 9$. The entropy is computed as the logarithm of the number of surviving networks.

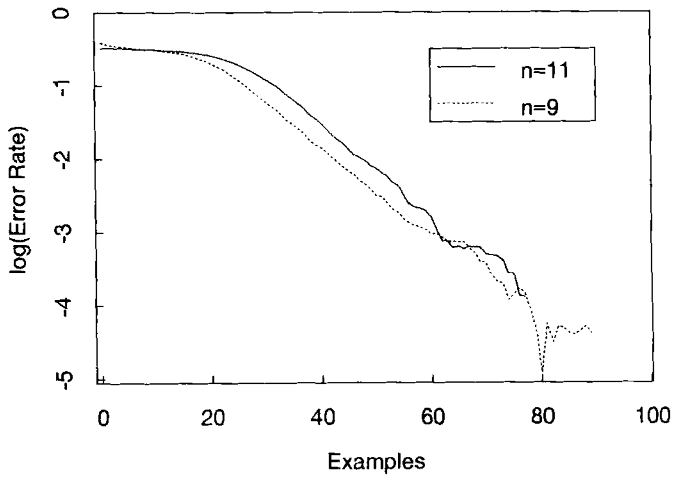

Figure 2: Numerical evaluation of the prediction error $\mathcal{E}_m$ as function of the size m of the training set for the contiguity problem, N=9 and 11.

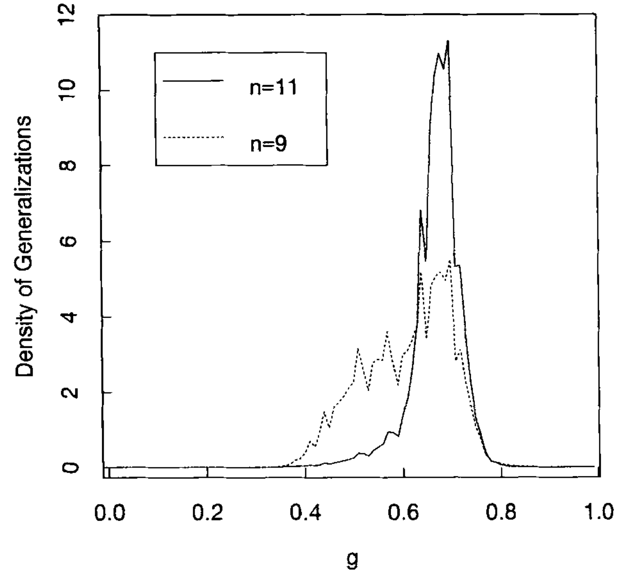

Figure 3: Initial distribution $\rho_0(g)$ for the generalization ability of the chosen network architecture to solve the contiguity problem, N=9 and 11.

$\rho_0(g)$ affect the prediction of G_m. Lack of accuracy in the determination of g for the individual networks results in a systematic broadening of $\rho_0(g)$ and overestimation of the prediction error $\mathcal{E}_m = 1 - G_m$. A more detailed analysis of such effects will be reported in a subsequent paper (Samalam and Schwartz 1989).

4 Discussion of Results

Results for the ensemble entropy S_m and the generalization ability G_m shown in Figures 1 and 2 as function of the size m of the training set confirm that supervised learning results in a monotonic decrease of the ensemble entropy and the prediction error.

The rate of entropy decrease $\eta_m = S_{m-1} - S_m$ measures the information content of the mth training example. The continuous decrease in the slope of the entropy in Figure 1 indicates that the effective information content of each new example is a decreasing function of m. The early stages of learning rapidly eliminate networks implementing functions with a very low intrinsic generalization ability $g(f)$. Such functions can

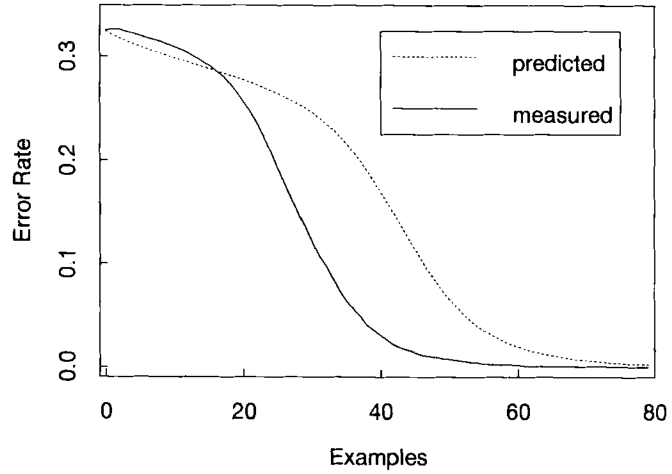

Figure 4: Prediction error $\mathcal{E}_m = 1 - G_m$ as function of the size m of the training set for the contiguity problem, $N = 11$. The numerical result of Figure 2 is compared to the prediction resulting from applying the recursion relation equation 2.18 to the initial histogram of Figure 3.

be eliminated with a small number of examples, and learning is very efficient. As learning proceeds, the surviving functions are characterized by $g(f)$ close to one. Such functions require a large number of examples, of order $(1 - g)^{-1}$, to be eliminated. The decrease in learning efficiency is intimately tied to the decrease in prediction error: an additional example carries new information and results in further entropy reduction to the extent to which it is unpredictable (Tishby et al. 1989).

The monotonic decrease of the prediction error $\mathcal{E}_m$ with m shown in Figure 2 is characterized by an exponential tail for sufficiently large m. Such exponential tail has also been observed in learning experiments on one layer ($L = 1$) networks using gradient descent (Ahmad and Tesauro 1989). The theoretical formalism presented here predicts such exponential decay for the learning of any Boolean function. Consider the case of Boolean functions from N inputs onto 1 output. There are 2^N possible inputs, and the intrinsic generalization ability can only be of the form $g_r = r/2^N$, with r an integer in the range $0 \leq r \leq 2^N$. Then

$$\rho_0(g) = \sum_{r=0}^{2^N} p_r \delta(g - g_r) \tag{4.1}$$

where p_r is the probability of $g = g_r$. The average generalization ability of equation 2.18 is easily computed for a density of the form equation 4.1:

$$G_m = \frac{\sum_{r=0}^{2^N} p_r g_r^{m+1}}{\sum_{r=0}^{2^N} p_r g_r^m} \tag{4.2}$$

and it is dominated at large m by the by the two largest values of r for which $p_r \neq 0$. If g=1 is attainable with probability p, and the next highest value $\hat{g} = 1 - \hat{\epsilon}$ is attainable with probability q, then for large m

$$\mathcal{E}_m = 1 - G_m \sim \frac{q}{p}\, \hat{\epsilon}\, \hat{g}^m \tag{4.3}$$

indicating an exponential decay of the form ϵ^{-m/m_0}, with $m_0^{-1} = -\ln \hat{g} \approx \hat{\epsilon}$.

The parameter m_0 controlling the rate of exponential decay is inversely proportional to the gap $\hat{\epsilon}$ between g=1 and $g = \hat{g}$. If $\hat{\epsilon} \rightarrow 0$ the exponential decay is replaced by a power law of the form

$$\mathcal{E}_m \sim \frac{m_0}{m + m_0} \tag{4.4}$$

Such asymptotic form follows from the moment ratio equation 2.18 for G_m whenever $\rho_0(g) \sim (1 - g)^{m_0}$ as $g \rightarrow 1$ (Tishby et al. 1989).

As a simple example of the continuous case, consider learning to separate points in $\mathcal{R}^N$ with a plane through the origin using an $L = 1$ network. Restricting the weights to the unit sphere results in an initial distribution of the form

$$\rho_0(g) \propto \sin^{N-2}(\pi g) \tag{4.5}$$

as follows from the Jacobian of a spherical coordinate system in N dimensions. The average generalization ability equation 2.18 is computed to be

$$\mathcal{E}_m = \frac{0.5 m_0}{m + m_0} \tag{4.6}$$

with m_0 controlled by the dimension N of the input.

It is intuitively obvious that the outcome of supervised learning is hard to predict, in that the dependence of the generalization ability of a trained network on the number of training examples is determined by both the problem and the architecture. The fundamental result of this paper is to demonstrate that knowledge of the initial distribution $\rho_0(g)$ suffices to predict network performance (equation 2.18). The specific details of the chosen architecture and the desired map $y = \tilde{f}(\mathrm{x})$ matter only to the extent that they influence and determine $\rho_0(g)$. The asymptotic form of learning curves $\mathcal{E}_m$ vs. m is controlled by the properties of $\rho_0(g)$

close to g=1: the existence of a gap results in exponential decay, while the continuous case leads to power-law decay.

The approach is based on analyzing the statistical properties of an ensemble of networks (Gardner 1988) at fixed architecture. In contrast to more general analysis based on the VC dimension of the network (Baum and Haussler 1989; Devroye 1988), which produce bounds on the prediction error, the performance of the ensemble is evaluated here in reference to a specific task. The question being asked is not how difficult it is to train a given network architecture in general, but how difficult it is to train it for the specific task of interest. It is in this ability to yield specific predictions that resides the potential power of the method.

Acknowledgments

One of the authors (D. B. S.) would like to acknowledge helpful discussions with R. L. Rivest, S. Judd, and O. Selfridge, as well as the support of AT&T Bell Laboratories where this work was begun.

References

Ahmad, S., and Tesauro, G. 1989. Scaling and generalization in neural networks: A case study. In *Advances in Neural Network Information Processing Systems I*, D. S. Touretzky, ed., pp. 160–168. Morgan Kaufmann, San Mateo.

Baum, E. B., and Haussler, D. 1989. What size net gives valid generalization? *Neural Comp.* **1**, 151–160.

Carnevali, P., and Patarnello, S. 1987. Exhaustive thermodynamic analysis of Boolean learning networks. *Europhys. Lett.* **4**, 1199–1204.

Denker, J., Schwartz, D., Wittner, B., Solla, S., Howard, R., Jackel, L., and Hopfield, J. 1987. Automatic learning, rule extraction, and generalization. *Complex Syst.* **1**, 877–922.

Devroye, L. 1988. Automatic pattern recognition, a study of the probability of error. *IEEE Trans. PAMI* **10**, 530–543.

Gardner, E. 1988. The space of interactions of neural network models. *J. Phys. A* **21**, 257–270.

Samalam, V. K., and Schwartz, D. B. 1989. A study of learning and generalization by exhaustive analysis. GTE Laboratories Tech. Rep. TM-0224-12-89-401.

Solla, S. A. 1989. Learning and generalization in layered neural networks: The contiguity problem. In *Neural Networks: From Models to Applications*, L. Personnaz and G. Dreyfus, eds., pp. 168–177. I. D. S. E. T. , Paris.

Tishby, N., Levin, E., and Solla, S. A. 1989. Consistent inference of probabilities in layered networks: Predictions and generalization. In *IJCNN International Joint Conference on Neural Networks*, Vol. II, 403–409. IEEE, New York.

Received 7 September 1989; accepted 2 May 1990.

This article has been cited by:

1. J Polhill. 2001. An approach to guaranteeing generalisation in neural networks. *Neural Networks* **14**, 1035-1048. [CrossRef]
2. Hanzhong Gu, H. Takahashi. 2000. How bad may learning curves be?. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, 1155-1167. [CrossRef]
3. Hanzhong Gu, Haruhisa Takahashi. 2000. Exponential or Polynomial Learning Curves? Case-Based Studies. *Neural Computation* **12**:4, 795-809. [Abstract] [PDF] [PDF Plus]
4. C. Van den Broeck, G. Bex. 1998. Multifractal a priori probability distribution for the perceptron. *Physical Review E* **57**, 3660-3663. [CrossRef]
5. Marco MuselliSequential constructive techniques 81-144. [CrossRef]
6. H Gu. 1997. Estimating Learning Curves of Concept Learning. *Neural Networks* **10**, 1089-1102. [CrossRef]
7. G. Bex, C. Van den Broeck. 1997. Domain sizes of the Gardner volume for the Ising reversed wedge perceptron. *Physical Review E* **56**, 870-876. [CrossRef]
8. Marc M. Van Hulle. 1997. Topology-preserving Map Formation Achieved with a Purely Local Unsupervised Competitive Learning Rule. *Neural Networks* **10**, 431-446. [CrossRef]
9. David Haussler, Michael Kearns, H. Sebastian Seung, Naftali Tishby. 1997. Rigorous learning curve bounds from statistical mechanics. *Machine Learning* **25**, 195-236. [CrossRef]
10. Peter Auer, Robert C. Holte, Wolfgang MaassTheory and Applications of Agnostic PAC-Learning with Small Decision Trees 21-29. [CrossRef]
11. Alexander V. Lukashin, Apostolos P. Georgopoulos. 1994. A Neural Network for Coding of Trajectories by Time Series of Neuronal Population Vectors. *Neural Computation* **6**:1, 19-28. [Abstract] [PDF] [PDF Plus]
12. A Minai. 1994. Perturbation response in feedforward networks. *Neural Networks* **7**, 783-796. [CrossRef]
13. B AMIRIKIAN, H NISHIMURA. 1994. What size network is good for generalization of a specific task of interest?. *Neural Networks* **7**, 321-329. [CrossRef]
14. J M R Parrondo, C Van den Broeck. 1993. *Journal of Physics A: Mathematical and General* **26**, 2211-2223. [CrossRef]
15. Thorsteinn Rögnvaldsson. 1993. Pattern Discrimination Using Feedforward Networks: A Benchmark Study of Scaling Behavior. *Neural Computation* **5**:3, 483-491. [Abstract] [PDF] [PDF Plus]
16. Timothy Watkin, Albrecht Rau, Michael Biehl. 1993. The statistical mechanics of learning a rule. *Reviews of Modern Physics* **65**, 499-556. [CrossRef]
17. Sara Solla, Esther Levin. 1992. Learning in linear neural networks: The validity of the annealed approximation. *Physical Review A* **46**, 2124-2130. [CrossRef]

18. Shun-ichi Amari, Naotake Fujita, Shigeru Shinomoto. 1992. Four Types of Learning Curves. *Neural Computation* **4**:4, 605-618. [Abstract] [PDF] [PDF Plus]

19. H. Seung, H. Sompolinsky, N. Tishby. 1992. Statistical mechanics of learning from examples. *Physical Review A* **45**, 6056-6091. [CrossRef]

20. A Krogh, J A Hertz. 1992. *Journal of Physics A: Mathematical and General* **25**, 1135-1147. [CrossRef]

21. David Cohn, Gerald Tesauro. 1992. How Tight Are the Vapnik-Chervonenkis Bounds?. *Neural Computation* **4**:2, 249-269. [Abstract] [PDF] [PDF Plus]

22. Stuart Geman, Elie Bienenstock, René Doursat. 1992. Neural Networks and the Bias/Variance Dilemma. *Neural Computation* **4**:1, 1-58. [Abstract] [PDF] [PDF Plus]

23. H.S. Seung, H. Sompolinsky, N. TishbyLearning Curves in Large Neural Networks 112-127. [CrossRef]