PHYSICAL REVIEW E VOLUME 47, NUMBER 4 APRIL 1993

Learning from examples in feedforward Boolean networks

Prabodh Shukla and Tapas Kumar Sinha
Department of Physics, North Eastern Hill University, Shillong 793 003, India
(Received 1 September 1992)

We examine the ability of a feedforward Boolean network to learn to implement the parity function of n input bits. It is shown, on the basis of exact enumerations, as well as theoretical analysis, that, in the absence of teaching, the probability of implementing correctly increasing lengths of the truth table has a staircase structure. This allows us to predict that, in the process of teaching the network, a sharp learning transition would take place at a fraction of teaching which goes to zero as $n2^{-n}$ in the limit $n \to \infty$.

PACS number(s): 87.10.+e, 02.50.−r, 05.20.−y

I. INTRODUCTION

Feedforward neural networks often show a remarkable ability to learn general rules from a small set of examples. For example, Paternello and Carnevali [1] demonstrated that a network consisting of 160 Boolean gates added correctly all pairs of two eight-bit numbers after being trained on only 224 random examples. Van den Broeck and Kawai [2] showed that a network consisting of of 25 gates implemented correctly the parity (sum modulo 2) of seven single-bit numbers after being trained on approximately 17% of the examples. They also studied networks with varying number of gates N, and the number of input bits n, and showed that the learning transition shifts to lower fractions of teaching and becomes sharper as n increases. The learning transition thus appears to approach a genuine thermodynamic phase transition in the large-n limit. In this paper we wish to examine the nature of this phase transition and the underlying mechanism.

II. STRUCTURE OF THE NETWORK

We use a network very similar to the one used in Refs. [1] and [2]. However, we describe it briefly to set up our notation. N Boolean gates are arranged in a linear feedforward manner. Each gate has two binary inputs and one output, and therefore it can be one of the g types where $g \leq 16$. An input to a gate can come from the n inputs to the network or from the outputs of the previous gates in the line. We allow configurations where both inputs to a gate may come from the same source. In Ref. [1] all 16 gates were used. In Ref. [2] only eight of them were used, i.e., AND, OR, XOR, ALL, and their negations. The results are not very sensitive to whether all 16 gates are used or only the eight listed above. We have restricted our work to $g=1$, 2, 6, and 8; $g=8$ means the eight gates used in Ref. [2], $g=6$ means AND, OR, XOR, and their negations, $g=2$ means XOR and its negation, and $g=1$ means XOR only.

III. EXACT ENUMERATIONS

A. Probability landscape

Although exact enumerations of even small networks are very time-consuming they provide valuable information and serve to caution us against straightforward application of Monte Carlo methods in larger systems. As an illustration, we show in Fig. 1 the results of exact enumerations of networks with $n=4$, $N=3$, and $g=6$. We enumerate all configurations of the network and classify them according to the Boolean function they implement. Although 2^{16} Boolean functions are possible with $n=4$, the $N=3$ network implements only 2160 functions. As is well known, all the functions are not implemented with equal probability but there are groups of functions which are implemented with equal probability. We have

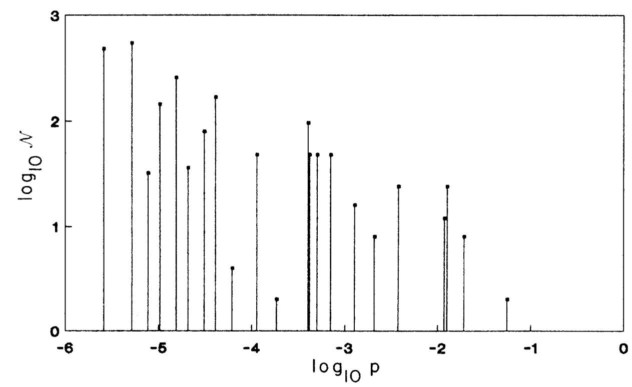

FIG. 1. Probability of Boolean functions implemented by a feedforward network with $n=4$, $N=3$, $g=6$. The x axis shows the logarithm of the probability, and the y axis shows the logarithm of the number $\mathcal{N}$ of Boolean functions having the same probability.

shown the probability of functions in a group versus the number of functions in that group in a log-log plot. Figure 2 shows a similar graph for $n=4$, $N=4$, and $g=6$. In this case a total of 9568 Boolean functions are implemented. The two shortest vertical lines in Figs. 1 and 2 correspond to two groups of functions which are the easiest to learn by the network. The number of functions in each group is only two. The two functions in each group are mirror images of each other so the hamming distance between them is the largest possible. The groups are relatively isolated (no other vertical lines which are too close). All these features contribute to the easy learning of these functions. One of the functions in the highest probability group corresponds to all bits being zero. In the other group one function corresponds to the parity of the input bits.

Comparisons of Figs. 1 and 2 show that as we go from $N=3$ to 4, more groups come up. The number of functions in new groups is high but their probability is very low. Thus the contribution of additional Boolean functions to the information entropy of the network is small. This is in accordance with the Monte Carlo based result [2] that the information entropy of the network is insensitive to N for large N. It is important to note that as N increases, the new functions which are implemented have such low probability that they remain invisible to the Monte Carlo methods unless the number of Monte Carlo steps is also appropriately increased. In view of the limitations of our computing resources (it takes nearly 60 h on a PC486 to generate the data shown in Fig. 2), and the difficulty of applying Monte Carlo techniques to systems with hierarchical probability landscapes, we have confined ourselves to exact enumerations of small networks.

B. Parity function

It is known [2] that for networks of a fixed size N, the learning of the parity of n bits from examples improves dramatically with increasing n. In this connection we note that this can be true only for $n \leq N+1$. The upper limit on n can be checked numerically and is easy to understand. If n is larger than $N+1$ then even a conventional circuit of N XOR gates will not be able to implement the XOR function of n variables. Neural networks have greater redundancy than the conventional chip and therefore no neural net configuration will be able to implement the parity of n bits if $n>N+1$. We have performed exact enumeration at the limiting value $n=N+1$ as we expect to find the sharpest learning transitions in this case. Indeed the numerical results bear out our expectation. We have carried out exact enumerations on networks with $n=2$, $N=1$; $n=3$, $N=2$; $n=4$, $N=3$; $n=5$, $N=4$; and $n=6$, $N=5$. In each case we took $g=1,2,6,8$. Figure 3 shows the results for the case $n=5$, $N=4$, $g=2,6,8$. The important features of Fig. 3 to be discussed below are also shared by all other cases.

The parity of five bits may be characterized as a 32-bit Boolean function where each bit represents the parity of five input bits in a particular configuration. Various configurations of five input bits can be conveniently labeled by a five-bit Boolean function or by its decimal equivalent integer from 0 to 31. We label the 32 bits of the parity function by the decimal equivalents of the five-bit Boolean function representing the configurations of the input bits. Thus the least significant bit represents the parity of five input bits when they are all zero, and the highest significant bit represents the parity of inputs when they are all unity. With this notation the 32-bit Boolean function desired to be implemented by the network is $BAAB\ ABBA$ where A stands for 0110 and B stands for 1001. We enumerate all configurations of the network which implement correctly L consecutive bits of the above function starting from the least significant bit. This yields the probability $P(L)$ of implementing L consecutive bits of the parity function correctly by the network. Figure 3 shows the graph of $\log_{10}P(L)$ vs L.

The graph in Fig. 3 has been obtained from a study of the network in the absence of teaching. However, the same graph can help in examining the ability of a network to learn from examples. This can be illustrated by a simple example. Suppose we confine ourselves to a $g=2$

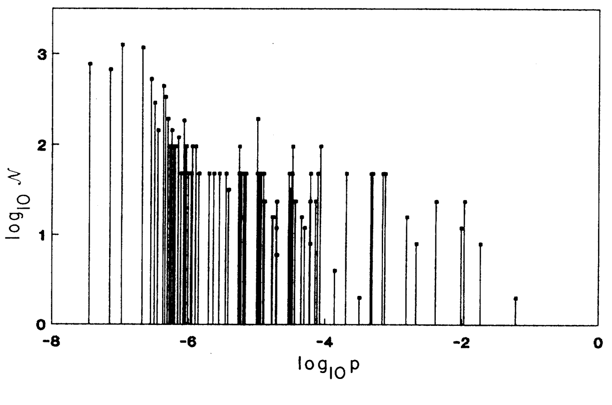

FIG. 2. Probability of Boolean functions implemented by a feedforward network with $n=4$, $N=4$, $g=6$.

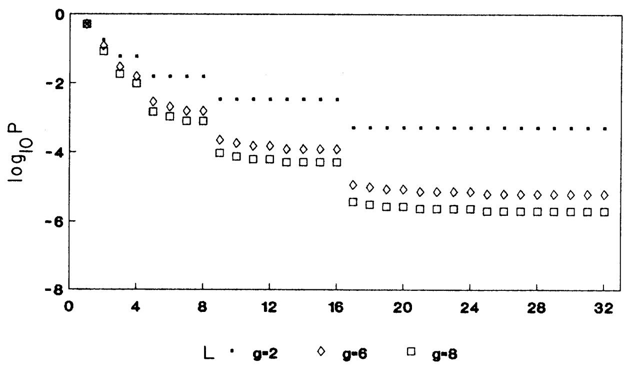

FIG. 3. Implementation of the 32-bit parity function by a network with $n=5$, $N=4$, $g=2,6,8$. The y axis shows the logarithm of the probability P of implementing L consecutive bits of the Boolean function (x axis).

network, and train it to learn the first 17 bits of the parity function. Then this training will suffice to implement the remaining 15 bits of the parity function faultlessly because there are no configurations of the network which implement 17 bits but not all 32. The crucial feature of Fig. 3 is that the probability decreases in sharp steps at $L=1,2,4,8$, and 16 and stays constant between these steps. This property is true for $g=2$ networks of all sizes as far as we have checked numerically.

Writing the total number of bits in the parity function as $T=2^n$, and $L=lT$, the probability changes in sharp steps at $l=\frac{1}{2},\frac{1}{4},\frac{1}{8},\ldots$ for $g=2$ networks. For higher values of g, the overall structure remains intact although the horizontal parts of the steps acquire some curvature as shown in Fig. 3. Let $p(g,T)$ be the probability that a network based on g types of gates implements all T bits of the parity function correctly. Let $p(g,L)$ denote the probability that a network will implement all T bits correctly after being trained on only L bits. It follows from the staircase structure of $p(g,L)$ that the probability that a $g=2$ network implements L bits correctly and still does not implement all the bits of the parity function is equal to the probability that all L training bits fall in the first half of the T-bit Boolean function. This probability is $(\frac{1}{2})^L$. For $g>2$ also we can take this probability to be $(\frac{1}{2})^L$ approximately if we neglect the curvature of the horizontal portion of the steps [3]. Thus we can write

$$p(g,L)=\frac{p(g,T)}{p(g,T)+(1/2)^L}=\frac{p(g,T)}{p(g,T)+(1/2^T)^l}\ . \tag{3.1}$$

The above formula was derived in Ref. [2] from a different point of view and shown to give a good description of the learning curve of the network. In Fig. 4 we have shown the above equation graphically for the case $n=5$, $N=4$, $g=8$ which has $p(g=8,T)=1.99\times10^{-6}$. For comparison, we have also shown in Fig. 4 the graph of Eq. (3.1) for the case $n=7$, $N=6$, $g=8$, with $p=1.00\times10^{-9}$. As n increases, p decreases and the learning transition given by Eq. (3.1) becomes sharper

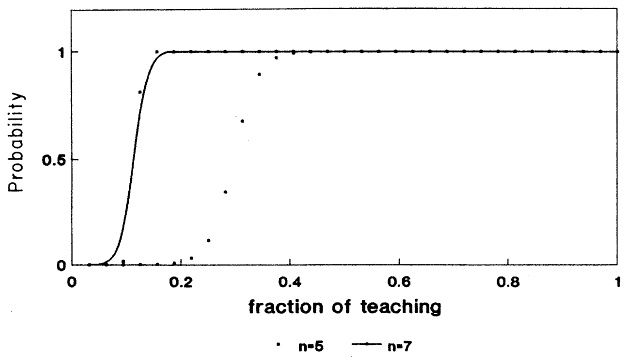

FIG. 4. Learning of the parity rule by two networks with (i) $n=5$, $N=4$, $g=8$; and (ii) $n=7$, $N=6$, $g=8$. The x axis shows the number of bits taught, and the y axis shows the corresponding probability of faultless learning.

and shifts to lower values of teaching fraction l. The critical learning fraction l_c may be defined as the teaching fraction where $p(g,L)=\frac{1}{2}$. We get

$$l_c=-\frac{\log_{10}p(g,T)}{T\log_{10}2}\ ,\quad T=2^n\ . \tag{3.2}$$

IV. LEARNING TRANSITION

In this section we derive an analytic expression for learning of the parity function of n bits by networks with $N=n-1$ and $g=1$, 2, 6, and 8. The analytic expression initially arose as a conjecture based on numerical studies of small-size networks, but reflection showed that it could be deduced rather simply. Let us for convenience call correct configurations those network configurations which correctly implement the parity of n input bits. Let $F(g,N)$ denote the number of correct configurations in a network with N gates where each gate can be of g types, and $n=N+1$. We note that the correct configurations with $g=1$ are those in which the first gate is fed by any two input bits, and each successive gate is fed by the output of its immediate predecessor gate and one of the remaining unused input bits. Thus $F(1,N)=(N+1)NF(1,N-1)$ with $N\geq2$ and $F(1,1)=2$. The second point to note is that in a correct configuration of $g=1$ network if any two successive XOR gates are both replaced by their inverse gates then the configuration still implements the desired function. Thus the correct configurations in a $g=2$ network can be obtained from the correct configurations of a $g=1$ network by replacing any two successive XOR gates by their inverses. This exhausts all the correct configurations of the networks with $g\geq2$, and $n=N+1$. Thus $F(2,N)=2^{N-1}F(1,N)$. The total number of configurations in a network is equal to $\prod_{k=1}^{N}(N+k)^2g^N$. Thus the probability $P(g,N)$ of a random configuration of the network being a correct configuration is given by

$$\begin{aligned}P(g,N)&=(N+1)\prod_{k=1}^{N}\left[\frac{k}{N+k}\right]^2 \quad \text{for } g=1\\ &=\left[\frac{2}{g}\right]^N\frac{1}{2}(N+1)\prod_{k=1}^{N}\left[\frac{k}{N+k}\right]^2\\ &\qquad\qquad \text{for } g=2,6,8\ .\end{aligned} \tag{4.1}$$

Note that $P(g,N)$ is equal to $p(g,T)$, which occurs in Eq. (3.2). Thus we get

$$l_c=-\frac{1}{T\log_{10}2}\left[N\log_{10}\left[\frac{2}{g}\right]+\log_{10}\left[\frac{N+1}{2}\right]+2\sum_{k=1}^{k=N}\log_{10}\left[\frac{k}{N+k}\right]\right]\ .$$

In the limit $N\rightarrow\infty$, $l_c\rightarrow0$ as $N2^{-N}$.

V. DISCUSSION

We have analyzed the capacity of a feedforward neural net to learn from examples the task of implementing the parity of n bits. We have restricted our analysis to networks with $N = n - 1$ gates (nodes). This is not a serious restriction in view of earlier work [2] which has established that the learning transition becomes sharper and shifts to a lower fraction of teaching as n increases. However, the number of bits n cannot be increased beyond $N + 1$ if the network is to implement the desired parity function of n bits. Thus our restriction to networks with $n = N + 1$ means that we have studied networks where the learning transition is expected to be sharpest and at the least value of the teaching fraction. These expectations are borne out by our analysis as well as exact enumerations on finite-size systems. Somewhat surprisingly our analysis predicts that the teaching fraction goes to zero as $n2^{-n}$ (apart from logarithmic corrections) in the limit $n \rightarrow \infty$. The origin of this result lies in the staircase structure of the probability landscape of the parity function. The probability of implementing consecutive L bits of the T-bit parity function ($T = 2^n$) has a staircase structure where each successive step is approximately half the height of its predecessor and twice as long. Thus the T bits can be divided into n groups of adjacent bits which constitute the horizontal portions of the staircase. The condition that the teaching should lead to faultless learning means that at least one bit from each of the n groups of bits should be taught. Thus the learning threshold may be expected to scale directly in proportion to n, and inversely in proportion to the total number of bits 2^n. Our analysis bears this out within logarithmic corrections.

In retrospect, the analysis presented in this paper takes some of the mystery out of the learning phenomena of feedforward neural nets. The feedforward Boolean network of logic gates emerges as a simple neural net which is naturally programed to learn the parity of a large number of bits from a vanishingly small fraction of examples. It is rather tantalizing to speculate if similar principles operate behind the functions of more complex neural nets. For example, there is a school of thought [4] that the human brain is biologically programed to learn language, i.e., the acquisition and use of language is facilitated by certain fundamental structural properties of the brain. The simple example studied in this paper supports the broad perspective that certain units (logic gates) arranged in certain structure (feedforward) can (in view of underlying symmetries) learn all 2^N manifestations of a rule from only N examples. It is therefore plausible that other units (neurons) in some other structure may possess similar facility for learning language. It is our hope that further study of very simple neural net models can support at least a rudimentary understanding of more complex phenomena.

[1] S. Paternello and P. Carnevali, Europhys. Lett. **4**, 503 (1987); **4**, 1199 (1987).

[2] C. Van den Broeck and R. Kawai, Phys. Rev. A **42**, 6210 (1990).

[3] It may be observed that the curvature of each step for $g = 6, 8$ is confined to the first half-length of the step. The second half-length corresponding to larger L values is flat. Figure 4 shows the plot of Eq. (3.1) with the factor $\frac{1}{2}$ replaced by $\frac{1}{4}$.

[4] Noam Chomsky, *Cartesian Linguistics* (Harper and Row, New York, 1966), pp. 59–73.