Antony Martin and Leonardo Franco

The influence of oppositely classified examples on the generalization complexity of Boolean functions

Article (Published version)
(Refereed)

Anthony, Martin and Franco, Leonardo (2006) *The influence of oppositely classified examples on the generalization complexity of Boolean functions.* Ieee transactions on neural networks, 17 (3). pp. 578-590. ISSN 1045-9227

DOI: 10.1109/TNN.2006.872352

The Influence of Oppositely Classified Examples on the Generalization Complexity of Boolean Functions

Leonardo Franco, *Member, IEEE,* and Martin Anthony

***Abstract*—In this paper, we analyze Boolean functions using a recently proposed measure of their complexity. This complexity measure, motivated by the aim of relating the complexity of the functions with the generalization ability that can be obtained when the functions are implemented in feed-forward neural networks, is the sum of a number of components. We concentrate on the case in which we use the first two of these components. The first is related to the "average sensitivity" of the function and the second is, in a sense, a measure of the "randomness" or lack of structure of the function. In this paper, we investigate the importance of using the second term in the complexity measure, and we consider to what extent these two terms suffice as an indicator of how difficult it is to learn a Boolean function. We also explore the existence of very complex Boolean functions, considering, in particular, the symmetric Boolean functions.**

***Index Terms*—Average sensitivity, Boolean functions, complexity, generalization, learning, randomness, symmetric functions.**

I. Introduction

A. Background

THE complexity of Boolean functions is one of the central and classical topics in the theory of computation. Scientists have long tried to classify Boolean functions according to various complexity measures, such as the minimal size of Boolean circuits needed to compute specific functions [1]–[3]. Franco [4], [5] introduced a complexity measure for Boolean functions that appears to be related to the generalization error when learning the functions by neural networks. This complexity measure has been derived from results showing that the generalization ability obtained for Boolean functions and for the number of examples (or similarly queries) needed to learn the functions when implemented in neural networks is related to the number of pairs of examples that are similar (close with respect to the Hamming distance), but have opposite outputs [6], [7]. (The Hamming distance between two binary vectors $x, y \in \{0,1\}^N$ of length N is simply the number of components on which they differ.) When only the bordering (or boundary) examples (those at Hamming distance 1) are considered, the complexity measure becomes equivalent to average sensitivity, a measure introduced by Kahn *et al.* [8]. Average sensitivity has been linked by Linial *et al.* [9] to the complexity of learning in the probabilistic "PAC" model of learning introduced by Valiant [10]; and many results about the average sensitivity of Boolean functions have been obtained for different classes of Boolean functions [11], [12]. It has been shown in [4], and is further analyzed in this paper, that terms that account for the number of pairs of opposite examples at Hamming distance larger than 1 are important in obtaining a better match between the complexity of different kinds of Boolean functions and the observed generalization ability.

Knowing and characterizing which functions are the most complex has a number of implications. It could help us to understand what functions can be most easily learned. (For human learning, the difficulty of learning has been linked to function complexity in [13].) In physics, a link has been established between the complexity measure and the Hamiltonian of spin systems [5], and a correspondence has been shown to exist between the most complex Boolean functions and the ground state of magnetic systems. It is worth noting that in recent years physics and computational complexity theory have both benefited from their interaction, and the analogy just mentioned offers a new point of contact between the disciplines. (For an introduction to some of the relationships between statistical mechanics and computational complexity, see [14] and [15].) In this way, the study of the complexity measure and of the most complex functions that it defines are of interest in a number of disciplinary contexts, including mathematical properties of Boolean functions, the physics of magnetic systems, and learning in real and artificial systems.

B. Overview of the Paper

In Section II, we introduce the complexity measure that is the focus of this paper. In its most general form, for a Boolean function $f : \{0,1\}^N \rightarrow \{0,1\}$, the complexity $C[f]$ is a weighted sum of $\lfloor N/2 \rfloor$ terms, $C_i[f]$, where $C_i[f]$ can be interpreted, broadly speaking, as indicating how many pairs of inputs to the Boolean function that are Hamming distance i apart have *different* output values of the function. We briefly indicate that some of the terms of this complexity measure can be related to previously-studied concepts in the theory of Boolean functions, such as average sensitivity and the Fourier coefficients.

In Section III, we report on some experiments conducted to investigate the usefulness of the second-order complexity term C_2 (used in conjunction with C_1) and it is demonstrated that the complexity measure $C_{12}[f] = C_1[f] + C_2[f]$ seems, for the families of functions considered, to correlate well with the

Manuscript received January 13, 2004; revised September 29, 2005. This work was supported by CICYT (España) by Grant TIN2005-02984 (that includes FEDER funds). The work of L. Franco was supported by the Ramón y Cajal Programme of the Ministerio de Educación y Ciencia (España). The work of M. Anthony was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

L. Franco is with the Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Campus de Teatinos S/N, 29071, Málaga, Spain (e-mail: lfranco@lcc.uma.es).

M. Anthony is with the Department of Mathematics, London School of Economics and Political Sciences, London WC2A 2AE, U.K.

Digital Object Identifier 10.1109/TNN.2006.872352

generalization error when the functions are learned using neural network architectures.

Section IV contains a theoretical analysis of the complexity of functions possessing some "irrelevant attributes," meaning that the functions do not depend on all the coordinates of their inputs. We then look closely at a particular case of such functions, namely those corresponding to the parity function on some subset of the attributes. Here, we find that the correspondence between the C_{12} complexity and generalization error is not extremely precise, and this therefore provides some evidence that, in some cases, it may be wise to use additional, higher-order, terms such as $C_3[f]$. This class of functions with irrelevant attributes is, however, rather small and we do find, empirically, a stronger correspondence between C_{12} and the generalization error when considering a larger class of functions derived from the parity functions, with some irrelevant attributes, but also a degree of "randomness"' in their definition.

Sections V and VI consider a well studied class of Boolean functions known as the *symmetric functions*. These are functions whose output depends only on the number of ones in the input (or, equally, these are the functions whose values are preserved under any permutation of the coordinates of the inputs). This class has been much-studied in a number of contexts: Many general results on sample and computational complexity have been obtained [1], [4], [16]. Section V presents some theoretical bounds on the complexity of these functions. Section VI investigates how the complexity of symmetric functions can be approximated by examining only the inputs that have a number of ones close to $N/2$.

The paper finishes with some discussion, conclusions, and suggestions for future work.

II. The Complexity Measure and Its Interpretation

A. A Complexity Measure for Boolean Functions

In its most general form, the Boolean function complexity measure considered here consists of a sum of terms, C_i, each of which accounts for the number of neighboring examples at a given Hamming distance having different outputs (that is, different values of the function). The complexity measure can be written in a general form as

$$C[f] = C_1[f] + \sum_{i=2}^{\lfloor \frac{N}{2} \rfloor} \alpha_i C_i[f] \tag{1}$$

where α_i are constant values that weight how pairs of oppositely classified examples (that is, elements of $\{0,1\}^N$ with different outputs) at Hamming distance i contribute to the total complexity, and N is the number of input bits. In all sections of the paper, apart from where is clearly indicated, the complexity measure used was equal to $C[f] = C_1[f] + C_2[f]$ (i.e., only the first two terms of (1) were used and the value of α_2 used was equal to 1).

Each term C_i has a normalization factor that takes into account both the number of neighboring examples at Hamming distance i and the total number $N_{ex} = 2^N$ of examples. Explicitly, if the examples are enumerated as $\{e_i : 1 \leq i \leq 2^N\}$, then

$$\mathcal{C}_i[f] = \frac{1}{N_{\text{ex}} * N_n(i)} \sum_{j=1}^{N_{\text{ex}}} \left(\sum_{\{l | \text{Hamm.}(e_j, e_l) = i\}} |f(e_j) - f(e_l)| \right) \tag{2}$$

where $N_n(i)$ is the number of examples at Hamming distance i from a given example, equal to the Binomial coefficient $\binom{N}{i}$. Thus, $C_i[f]$ may be interpreted as the probability, uniformly over choice of example x, and uniformly over the choice of an example y at Hamming distance i from x, that $f(y) \neq f(x)$.

B. Relationship to Other Measures Associated With Boolean Functions

The first term, C_1 is proportional to the number of bordering (or boundary) examples of the function f; that is, those with an immediate neighbor having opposite outputs. Equivalently, it is the probability that flipping a uniformly chosen bit in a uniformly chosen example will change the output of f. This is proportional to the average sensitivity $as(f)$ of the function f [8], [9], [17]. For a Boolean function on N variables and $v \in \{0,1\}^N$, the *sensitivity* of f at v, which we shall denote $s_f(v)$, is the number of neighbors $v\prime$ of v (that is, $v' \in \{0,1\}^N$ differing from v only in one entry) such that $f(v') \neq f(v)$. The average sensitivity is defined to be the average, over all elements v of $\{0,1\}^N$, of the sensitivities, $as(f) = 2^{-N} \sum_{v \in \{0,1\}^N} s_f(v')$. The average sensitivity $as(f)$ is also related to the notion of the "influence" of a variable; see [8], for instance. The influence of variable x_i is defined to be the proportion of $x \in \{0,1\}^N$ such that if x^i is obtained from x by changing the ith entry of x, we have $f(x^i) \neq f(x)$. It can be seen that $s(f)$ is the sum of the influences of the N variables. The connection between C_1 and the average sensitivity is that $C_1[f] = as(f)/N$.

The number of bordering examples (equivalently, the average sensitivity or $C_1[f]$) has been shown to be related to the generalization ability that can be obtained when Boolean functions are implemented in neural networks [6], to the number of examples needed to obtain perfect generalization [6], [7], to a bound on the number of examples needed to specify a linearly separable function [18], and to the query complexity of monotone Boolean functions [19]. Moreover, links between the sensitivity of a Boolean function and its learnability in the PAC sense have been established [9] and many results regarding the average sensitivity of Boolean functions have been derived [8], [11], [12], [20].

The complexity measures can also be related to another important idea in the theory of Boolean functions, namely the Fourier coefficients of f. Fourier (or harmonic) analysis of Boolean functions has recently proven to be extremely useful in a number of areas, such as learning theory ([9], [21] for instance) and circuit complexity [22]. (References [23] and [24] provide good surveys of the harmonic analysis of Boolean functions and its uses.) It is shown in [8] that the average sensitivity of a Boolean function f, and hence the complexity measure $C_1(f)$ may be written in terms of the Fourier coefficients of f. Furthermore, results of Kahn, Kalai, and Linial also show that the higher-order complexity terms $C_i[f]$ can be expressed in terms of the Fourier coefficients.

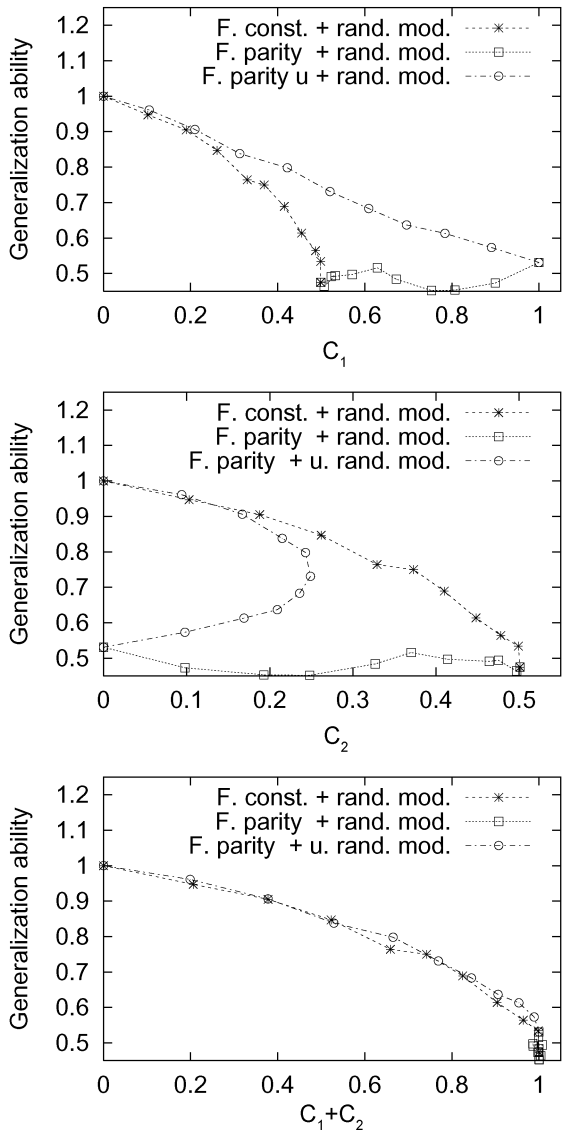

Fig. 1. Generalization ability versus first term (top graph), second term (middle graph), and first plus second terms (bottom graph) of the complexity measure for three different classes of functions. Each point in the graphs corresponds to the average obtained across 30 generated functions with approximately the complexity shown. (See text for more details.).

III. Importance of the Second-Order Term of the Complexity Measure

The second-order term has been shown to be relevant in order to produce an accurate match between the complexity measure and the observed generalization ability when the functions are implemented in neural networks [4]. Experimental results indicate that the first-order complexity term $C_1[f]$ alone does not give as good a correspondence as does the combination $C_{12}[f] = C_1[f] + C_2[f]$.

Fig. 1 (top) shows the generalization ability versus the first-order complexity (C_1) obtained from simulations performed for three different classes of functions. Each point in the graphs of Fig. 1 corresponds to the average obtained across 30 generated functions with approximately the complexity shown. The first class of functions (indicated as $\mathrm{F.const} + \mathrm{rand.mod}$ in the figure) was generated by modifications of the constant, identically-1, function, producing functions with first-order complexities C_1 between 0 and 0.5. These were generated as a function of a parameter p in the following way: For every example, a random uniform number in the range [0, 1] was selected and then compared to the value of p. If the random value was smaller than p then the output of the function on that example was randomly selected with equal probability to be 0 or 1. Thus, for each example, with probability p, the output is randomly chosen, and with probability $1-p$ the output is 1. The second set of functions ($\mathrm{F.parity} + \mathrm{rand.mod.}$ in the figure) was generated in the same way but through random modifications of the parity function, to obtain functions with a complexity between 1 and 0.5. (The parity function is the function that has output 1 on an example precisely when the example has an odd number of entries equal to 1.) The third set of functions ($\mathrm{F.parity\ u} + \mathrm{rand.mod}$ in the figure) was generated as follows: Starting with the parity function, for each positive example (that is, an example with output 1 on the parity function), the output is changed to 0 with probability p. This yields functions with complexities ranging from 1 to 0, all of which, except the initial parity function, are unbalanced (in the sense that the number of outputs equal to 0 and 1 are different). The classes of functions span a range of complexities according to the way they are generated. In particular, for the case of the first term of the complexity measure, C_1, the maximum possible range of C_1 from 0 to 1 is covered only when the class includes the constant function (with all outputs having the same value), for which $C_1 = 0$, and the parity function or its complementary one, the only functions with complexity C_1 equals to 1.

Fig. 1 (top graph) shows, for each of these three classes, the generalization ability computed by simulations performed in a neural network architecture with $N = 8$ inputs and eight neurons in the hidden layer trained with backpropagation, using half of the total number of examples for training, one fourth for validation and one fourth for testing the generalization ability (by which we mean the proportion of the test set that is classified correctly by the network after training). The generalization error is measured at the point at which the validation error achieved its minimum value. In Fig. 1 (middle) the generalization ability is plotted against the second-order term of the complexity measure, C_2, and it can be observed that the general behavior of the generalization seems uncorrelated to this second term (Note the different scale in this graph in comparison to the other two). But when we plot, in figure (bottom), the generalization ability versus $C_1 + C_2$ better agreement is obtained for the three different classes of Boolean functions. The discrepancy observed in the generalization ability in Fig. 1 for functions with similar complexity C_1 appears to be almost totally corrected when $C_1 + C_2$ is used. The shape of the curves in Fig. 1 (middle graph) is better understood if we consider at the same time the behavior of the generalization ability as a function of C_1, as the generalization ability is much more correlated to C_1 than to C_2 when C_1 and C_2 are considered separately. In particular for the case of the functions labeled $\mathrm{F.parity+u.rand.mod.}$ the shape of the curve is understood by following the curve in Fig. 1 (middle graph) starting from the point where generalization ability equals to 1 and C_2 equals to 0 in clockwise sense and consider that along

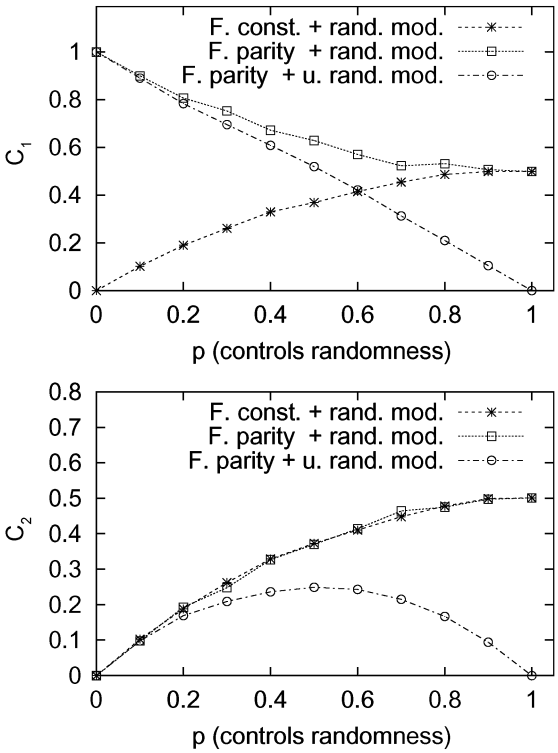

Fig. 2. Complexity terms C_1 (top) and C_2 (bottom) versus the parameter p that controls randomness for the class of functions used in Fig. 1. For the first two classes of functions the amount of randomness is directly proportional to p while for the third class the maximum randomness is achieved for $p = 0.5$. A good agreement between the amount of randomness and C_2 is obtained.

this trajectory the value of C_1 monotonically increases Fig. 1 (top graph) and thus, as expected, the generalization ability decreases.

We also show the values of C_1 and C_2 as a function of the parameter p that controls the amount of randomness in the functions in Fig. 2 (top and middle). The first (second) class of functions starts with the constant (parity) function when $p = 0$ and ends in a totally random function when $p = 1$. We see that C_1 increases (decreases) and C_2 increases with p, as expected. For the third class of functions the maximum amount of randomness corresponds to 0.5, at which point all of the outputs that were originally 1 are changed to 0 with probability 0.5. (Note that, for this class of functions, the value of p is proportional to the fraction of output bits 1 remaining in the definition of the function.)

The results for the three different classes of functions presented in Fig. 2 (middle) show a clear correlation between the amount of randomness or lack of structure (or regularity) present in the Boolean functions and the value of C_2. The term C_2 includes the contribution of pairs of examples at Hamming distance 2 and belonging to the same "level" (that is, having the same weight, the number of bits that are ON) and also from examples that are at Hamming distance 2 from each other and have weights differing by 2. It is worth noting that the first contribution, from examples in the same level (i.e., of the same weight) measures how "nonsymmetric" the function is. (For symmetric functions, all examples of a particular weight have identical outputs.)

We have also investigated the behavior of the generalization error versus the complexity value of the functions when

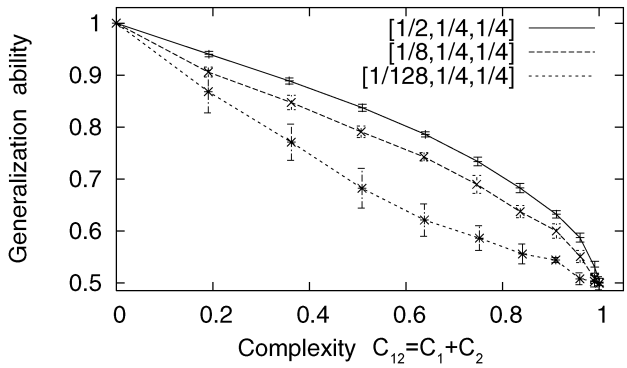

Fig. 3. Generalization ability versus the complexity C_{12} (first and second terms included) for Boolean functions with $\mathrm{N} = 14$ inputs created starting from the constant function and adding random modifications. The three curves shown were computed for different sizes of the training set while the size of the validation and generalization sets was held constant. It is indicated in the figure legend the fraction of examples used in each set. (See text for more details.) The error bars indicate the standard deviation of the mean computed over sets of 10 functions.

different sizes of sets are used for the training, validation and generalization measurement procedures. To analyze this, we ran simulations using the set of Boolean functions generated by random modifications of the constant function (previously named $\mathrm{F.const} + \mathrm{rand.mod}$) that generates a set of Boolean functions with complexity $C_{12} = C_1 + C_2$ in the range from 0 to 1 for the case of $N = 14$ inputs. The total number of different input-output pairs (examples) is equal in this case to $2^{14} = 16\,384$. In Fig. 3 the results are shown for three different analyzed cases. The upper curve in Fig. 3 corresponds to the case of using half of the total number of examples for training (8192), one fourth (4096) for validation, and one fourth for measuring the generalization error. The middle curve was constructed using [1/8, 1/4, 1/4] of the total number of examples, corresponding to 2048, 4096, and 4096 examples. The lower curve was computed using 128 examples for training and the same size sets for the validation and generalization sets (indicated by [1/128, 1/4, 1/4]). The figure indicates that the monotonic relationship between complexity and generalization ability is preserved for different selection of the training set sizes. The figure also shows that as the fraction of examples for training diminishes the generalization ability is reduced, something that, of course, is to be expected from standard models of generalization (such as discussed in [25] and [26]), as a reduction in training sample size implies a reduction in knowledge about the target function. The results presented in Fig. 3 show that when a comparison between different functions is done the size of the training set used has to be approximately the same for all the different functions, as the size of the training set clearly affects the level of generalization that can be obtained. Thus, if the complexity of a function is measured it is not straightforward to predict the level of generalization that can be expected as the size of the training set used plays also an important role. Nevertheless, the complexity value obtained can be used as a comparison against other functions that will be trained with the same size training set.

Another indication of the importance of the second-order complexity term C_2 comes from the fact that when only C_1 is considered, the functions with highest complexity turn out to

be the very well known parity function and its complement [4], [27], yet, as we shall see, numerical simulations have shown that there are functions which are more complex to implement on a neural network (in the sense that the generalization error is higher). Indeed, for this class of very complex functions the generalization error obtained is greater than 0.5 (which is what would be expected for random functions). In [4] it has been shown that the average generalization error over a whole set of functions using the same architecture is 0.5, indicating that there exist functions for which the generalization error is higher than 0.5. Similar results have been obtained for time series implemented by perceptrons by Zhu & Kinzel [28].

IV. A Method for Finding Very Complex Boolean Functions and Ground States

A. Irrelevant Attributes

In this section, we investigate how to find Boolean functions with a high C_{12} complexity. One technique that seems to be useful is to consider functions having a number of irrelevant attributes. Such an approach is motivated by considerations from statistical mechanics. In [4], [5] an analogy is established between the Boolean function complexity measure and the Hamiltonian of magnetic systems. This analogy implies that there is a correspondence between the ground state of magnetic systems and the most complex functions. Ground states of many magnetic systems have been observed often to have a certain type of order (short or long range order) and it is a subject of controversy under which conditions this order does not arise [5], [29]. In some cases the ordered ground state consists of two equal size antiferromagnetic domains, corresponding in the language of Boolean functions to a parity function on $N-1$ variables, with the Nth variable being irrelevant. Finding the ground states of magnetic systems is a complicated task only rigorously undertaken in very few cases. It has been shown that in most cases the problem of rigorously establishing that a state is the ground state is computationally intractable [30], [31].

A Boolean function is said to have A irrelevant attributes if there are indexes $i_1, i_2, \ldots, i_A$ such that the value $f(x_1, x_2, \ldots, x_N)$ of the function does not depend on $x_{i_1}, x_{i_2}, \ldots, x_{i_A}$. For the sake of simplicity, let us suppose these "irrelevant attributes" are $x_{N-A+1}, x_{N-A+2}, \ldots, x_N$. Then, the value of f is determined entirely by its projection f^* onto the relevant $N-A$ attributes, given, in this case, by

$$f^*(x_1, x_2, \ldots, x_{N-A}) = f(x_1, x_2, \ldots, x_{N-A}, 0, 0, \ldots, 0).$$

(The choice of 0 for the last A coordinates here is arbitrary, the point being that the value of f is independent of these.)

B. C_{12} Complexity With Irrelevant Attributes

The complexity $C_{12}[f]$ of f can be related to that of f^* as follows.

Theorem 4.1: With this notation, we have

$$C_{12}[f] = \left(\frac{N-A}{N} + \alpha_2 \frac{2A(N-A)}{N(N-1)}\right) C_1[f^*] + \alpha_2 \frac{(N-A)(N-A-1)}{N(N-1)} C_2[f^*]. \quad (3)$$

Proof: To see why (3) holds, it is useful to recall the probabilistic interpretations of C_1 and C_2. For $x \in \{0,1\}^N$, and $1 \le i < j \le N$, let x^i denote the example obtained from x by "flipping" the ith component—that is, by changing x_i from 0 to 1 or from 1 to 0—and let x^{ij} be the example obtained from x by flipping components i and j. Then, we have the following, where all probabilities indicated are uniform over choice of x and over the choice of components to be flipped

$$\begin{aligned}
C_1[f] &= \Pr\left(f(x^i) \neq f(x)\right) \\
&= \Pr\left(f(x^i) \neq f(x) | i \le N-A\right) \Pr(i \le N-A) \\
&\quad + \Pr\left(f(x^i) \neq f(x) | i > N-A\right) \Pr(i > N-A) \\
&= \Pr\left(f^*(x^i) \neq f^*(x)\right) \frac{N-A}{N} + 0 \\
&= \frac{N-A}{N} C_1[f^*]. \\
C_2[f] &= \Pr\left(f(x^{ij}) \neq f(x)\right) \\
&= \Pr\left(f(x^{ij}) \neq f(x) | i, j \le N-A\right) \Pr(i, j \le N-A) \\
&\quad + \Pr\left(f(x^{ij}) \neq f(x) | i \le N-A, j > N-A\right) \\
&\quad \times \Pr(i \le N-A, j > N-A) \\
&\quad + \Pr\left(f(x^{ij}) \neq f(x) | i, j > N-A\right) \Pr(i, j > N-A) \\
&= \Pr\left(f^*(x^{ij}) \neq f^*(x)\right) \frac{\binom{N-A}{2}}{\binom{N}{2}} \\
&\quad + \frac{(N-A)A}{\binom{N}{2}} \Pr\left(f^*(x^i) \neq f^*(x)\right) + 0 \\
&= \frac{(N-A)(N-A-1)}{N(N-1)} C_2[f^*] + \frac{2(N-A)A}{N(N-1)} C_1[f^*].
\end{aligned}$$

From these, (3) follows. □

Consider the N-dimensional Boolean functions defined as the parity function on $N-A$ variables for $A = 0, 1, \ldots, N$. The complexity of these functions including the first- and second-order terms, C_1 and C_2, can be written in terms of A as

$$\begin{aligned}
C_{12}[f] &= C_1[f] + \alpha_2 C_2[f] \\
&= \frac{N-A}{N} + \alpha_2 \frac{2A(N-A)}{N(N-1)} \quad (4) \\
&= \frac{(N-A)(N-1+2\alpha_2 A)}{N(N-1)} \quad (5)
\end{aligned}$$

This follows from (3), because the functions concerned have A irrelevant attributes, and because the projection f^* onto the $N-A$ relevant attributes is the parity function on $N-A$ variables, having $C_1[f^*] = 1$ and $C_2[f^*] = 0$. It can also be seen directly: The two terms in (4) represent the fraction of pairs of examples with opposite outputs at Hamming distances 1 and 2, respectively. To find the function of this particular type with highest C_{12} complexity, we take the derivative of C_{12} with respect to A (assuming, temporarily, that A is a continuous parameter)

$$\begin{aligned}
\frac{dC_{12}}{dA} &= \frac{d}{dA}\left(\frac{(N-A)(N-1-2\alpha_2 A)}{N(N-1)}\right) \\
&= \frac{(2\alpha_2 - 1)N + 1 - 4\alpha_2 A}{N(N-1)} \quad (6)
\end{aligned}$$

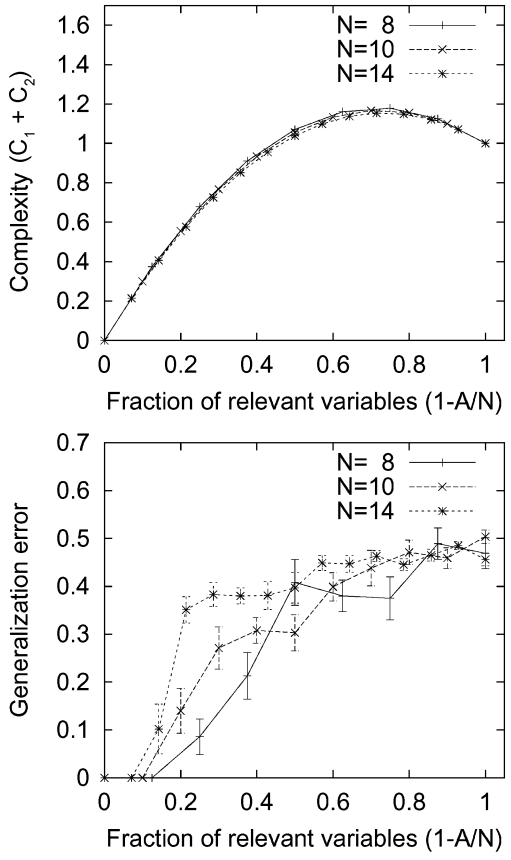

Fig. 4. (Top graph) Complexity $(C_1 + C_2)$ for Boolean functions that implement the parity function on $N - A$ variables as a function of the fraction of relevant variables $(1 - (A/N))$, for different values of N. (Bottom graph) Generalization error versus the fraction of relevant variables for the case of the parity function on $(N - A)$ bits.

Assuming that $\alpha_2 \geq 1/2$, and calculating the maximizing value $A_{\max}$, the root of (6), we obtain

$$A_{\max} = \frac{N}{2} - \frac{N-1}{4\alpha_2} > 0. \tag{7}$$

The complexity of the corresponding function is

$$C_{12}\left[f(A_{\max})\right] = \frac{\left(\frac{N}{2} + \frac{N-1}{4\alpha_2}\right)\left(\frac{N-1}{2} + \alpha_2 N\right)}{N(N-1)}. \tag{8}$$

In particular, for the case $\alpha_2 = 1$ we obtain $A_{\max} = (N+1)/4$ which yields, when $(N+1)/4$ is an integer, a Boolean function with complexity

$$C_{12}\left[f(A_{\max})\right] = \frac{(3N-1)^2}{8N(N-1)} > \frac{9}{8}. \tag{9}$$

The complexity of the function found is larger than 1.125 for any N, indicating that we have found a very complex function. (For comparison, we note that the complexity $(C_1 + C_2)$ of the parity function and of a random function is approximately 1.0.) As mentioned in Section IV-A, the problem of finding the most complex functions is analogous to finding the ground state of magnetic systems, that can be rigorously analyzed only in few cases. It is a specially difficult problem to analyze when competing (or frustrating) interactions exists, as is the case of the complexity measure comprising at least two terms. Our results show that there exists functions with a complexity $(C_{12} = C_1 + C_2)$ of at least 1.125 but we can not prove that these are the maximum complexity functions. An exhaustive analysis of all functions with $N = 4$ inputs showed that the most complex function in that case had $(C_{12} = C_1 + C_2)$ equals to 1.25 [4].

C. Empirical Investigation

We empirically study the generalization error obtained when Boolean functions are implemented on feed-forward neural networks, to analyze its correlation with the complexity measure. We show in Fig. 4 the values of the complexity $C_1 + C_2$ (top) and generalization error (bottom) of parity functions that depend on $N - A$ variables for the cases $N = 8, 10,$ and 14. The behavior of the generalization error (Fig. 4, bottom) follows approximately the shape of the curve obtained for the complexity $(C_1 + C_2)$ (Fig. 4, top) but only approximately. The C_{12} complexity reaches a maximum value for a value of $1 - (A/N)$ approximately equal to 0.7 (Fig. 4, top). The results from the simulations for $N = 8, 10,$ and 14 show a generalization error having an upward trend, as the fraction of relevant variables increases, reaching a maximum when there are no irrelevant variables. This might be explained by the fact these functions are quite regular and maybe easier to implement than other functions with the same amount of randomness. We also note that if the third term (or higher-order terms) is considered in the complexity measure then the prediction is that a local maximum value for the complexity $(C_1+C_2+C_3)$ as a function of the fraction of relevant variables no longer exists. (More information about this is given later in this section.) We analyze in the discussion what criteria to use when considering how many terms to include in the definition of the complexity measure.

To continue with the empirical analysis, we decided to look at some related functions with a greater element of randomness in their definition. We considered functions that implement the parity function of $N - A$ variables, where the final A variables on any given input example were subject to random alteration: Each of the final A variables of an example were, with probability 0.5, left unchanged, and with probability 0.5, were set to 0. On each example, then, the function constructed computed the parity function of $(N-A+\delta)$ variables on that example, where δ is a value between 0 and A, distributed according to a binomial distribution with mean $A/2$. The set of functions found is a complex one for which the generalization error can, for some values of A, be larger than 0.5.

The simulations were performed in one hidden layer architectures, trained with backpropagation, using half of the total number of examples for training, one fourth for validation and the remaining fourth to measure the generalization error. The validation set was used to prevent overfitting, i.e., the validation error is constantly monitored and at its minimum value the generalization error is measured. The training was done using standard backpropagation with momentum with learning rate equal to 0.25 and momentum constant equal to 0.2. The number of neurons in the single hidden layer used in the architectures were equal to the number of inputs. (Other possibilities were explored and no significant differences were found.) The error bars plotted show the standard error of the mean (SME), when 50 averages were taken. The SME decreases as the number of input bits, N, increases and for fixed N it was approximately

constant as a function of A, except for the case $A = 0$, in which case the SME was normally larger.

In Fig. 5 the values of the complexity $C_1 + C_2$ (top) and generalization error (bottom) obtained for the Boolean functions that implement the parity function on $(N - A + \delta)$ variables (in the sense described earlier) are shown for the cases of N =8, 10, 14. The generalization error is larger than 0.5 for some values of $1 - (A/N)$. Fig. 5 (bottom) indicates that a correlation is found with the computed values (from the numerical simulations) for the complexity (C_1+C_2) Fig. 5 (top). Furthermore, by using the empirical value obtained for the maximum of the generalization error as a function of $1 - (A/N)$ we estimate, through (7), a value for α_2. From the results, plotted in Fig. 5 (bottom), we obtain values of $A_{\max}$ equal to 7/8 and 8/10 for N =8 and 10, respectively, that we correct to 7.5/8 and 9/10 because the functions were created by using the extra variables indicated by (δ) as mentioned before and thus δ can be considered to be equivalent for $A/2$ extra variables. These values lead to α_2 equals to 0.5 and 0.5625 for the cases $\mathrm{N} = 8$ and $\mathrm{N} = 10$, respectively (the simulations for the case $\mathrm{N} = 14$ do not show a clear maximum value for the generalization error and this is why we did not do the estimation for this case.

We can also find functions for which C_2 is very high, independently of C_1. In this case, we repeat the procedure used in (4) to find that functions defined as the parity on $N/2$ variables have a complexity C_2 equal to

$$C_2\left[f\left(A = \frac{N}{2}\right)\right] = \frac{N}{2(N-1)} \tag{10}$$

which is larger than 0.5 for all N. For the case $N = 4$, this gives a function with $C_2 = 2/3$ and we have exhaustively verified that this is the largest value that can be obtained for any Boolean function with $N = 4$. For cases where $N \neq 4$, we do not know how close the value obtained for C_2 is to its maximum possible; and, as aforementioned, it could be that the demonstration that this value corresponds to the functions with maximum complexity is intractable (Istrail, 2000). For symmetric Boolean functions, however, as we observe in Section V, the maximum value that can be achieved for C_2 is 0.5.

D. Higher-Order Complexity Terms

Suppose that f is a Boolean function of N variables, of which A are irrelevant. Using the same notation as earlier, for $1 \leq k \leq \lfloor N/2 \rfloor$, the kth complexity term $C_k[f]$ of f can be related to the complexity terms $C_i[f^*]$ (for $1 \leq i \leq k$) of f^* as follows.

Theorem 4.2: With this notation

$$C_k[f] = \frac{1}{\binom{N}{k}} \sum_{j=r}^{s} \binom{N-A}{j}\binom{A}{k-j} C_j[f^*] \tag{11}$$

where $r = \max(0, k - A)$ and $s = \min(k, N - A)$.

Proof: Again, we use the probabilistic interpretation of C_k. For $x \in \{0,1\}^N$, and $I \subseteq [N] = \{1, 2, \ldots, N\}$, let x^I denote the element of $\{0,1\}^N$ obtained from x by "flipping" the components x_i for $i \in I$—that is, by changing these x_i from 0 to 1 or from 1 to 0. For $|I| = k$, the number of relevant attributes in I (which is $|I \cap [N-A]|$) has to be at least $r = \max(0, k-A)$ (since there are A irrelevant attributes); and, the number of relevant attributes can be no more than $s = \min(k, N - A)$ (since there are $N - A$ relevant attributes in total). In the Theorem statement, if $r = 0$ then $C_0[f^*]$ should be considered equal to 0. Then we have the following, where all probabilities indicated are uniform over choice of x and over the choice of coordinate set I of cardinality k

$$\begin{aligned} C_k[f] &= \Pr\left(f(x^I) \neq f(x)\right) \\ &= \sum_{j=r}^{s} \Pr\left(f(x^I) \neq f(x) \,|\, |I \cap [N-A]| = j\right) \\ &\quad \times \Pr\left(|I \cap [N-A]| = j\right) \\ &= \sum_{j=r}^{s} C_j[f^*] \frac{\binom{N-A}{j}\binom{A}{k-j}}{\binom{N}{k}} \end{aligned}$$

which is as required. □

Consider now, as earlier, the function defined as the parity function on $N - A$ variables, where A is between 0 and N. (So, the case $A = 0$ is the usual parity function, and the case $A = N$ is a function that is constant (either identically 1 or identically 0).

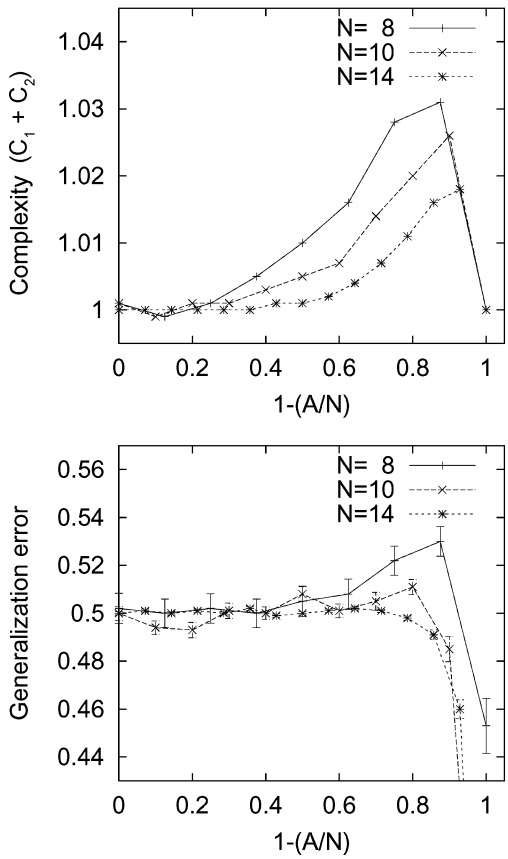

Fig. 5. (Top graph) Complexity $(C_1 + C_2)$ and bottom graph) Generalization error vs the fraction of relevant variables $(1 - (A/N))$ for Boolean functions described in the text as the parity on $(N - A + \delta)$ variables for different values of N.

Corollary 4.3: Suppose that $0 \leq A \leq N$ and that f is the parity function on $N - A$ variables (having A irrelevant attributes). Then, for $1 \leq k \leq \lfloor N/2 \rfloor$

$$C_k[f] = \frac{1}{\binom{N}{k}} \sum_{l=R}^{S} \binom{N-A}{2l+1} \binom{A}{k-2l-1}$$

where $R = \max(0, \lceil (k - A - 1)/2 \rceil)$ and $S = \min(\lfloor (k - 1)/2 \rfloor, \lfloor (N - A - 1)/2 \rfloor)$.

This follows from (11), because the projection of f^* onto the $N - A$ relevant attributes is the parity function on $N - A$ variables, having $C_j[f^*] = 1$ if $j = 2l+1$ is odd, and $C_j[f^*] = 0$ otherwise.

For $1 \leq k \leq \lfloor N/2 \rfloor$, let $C_{1,\ldots,k}[f] = C_1[f] + C_2[f] + \cdots + C_k[f]$ be the complexity of f when all complexity terms up to order k have been included. We commented that in considering $C_{12}[f] = C_{1,2}[f] = C_1[f] + C_2[f]$, the parity functions on $N - A$ attributes have a $C_{12}[f]$ complexity measure that increases to a maximum and then decreases again, as the number of relevant attributes is increased (that is, as A is decreased). However, this behavior appears *not* to be present when we consider $C_{123}[f] = C_{1,2,3}[f]$; and also when we consider the full complexity $C[f] = C_{1,2,\ldots,\lfloor N/2 \rfloor}[f]$. In considering these "fuller" measures of complexity, it appears that as the number of relevant attributes is increased, the complexity increases monotonically, so that the parity function itself (with no irrelevant attributes) has the highest complexity among the $N + 1$ functions. This perhaps explains the empirical results on the generalization error for these functions (Fig. 4, bottom), something that we noted could not be explained simply by the first- and second-order complexity measure $C_{12}[f]$. We have seen that the first-order complexity C_1 is often an inadequate measure of complexity and that this can be corrected by adding the second-order term. Now we see, additionally, that, for some classes of function, even higher-order complexity terms are required to reveal a correspondence with generalization error.

V. Complexity of Symmetric Boolean Functions

An important class of Boolean functions is the class of symmetric functions, those for which the output depends only on the number of input bits ON (or, equivalently, on the weight of the example). This class includes many important functions, such as the parity function and the majority function, and many results regarding different properties of these functions have been obtained [1], [6], [7], [16], [32], [33]. We first determine independently the maximum values of C_1 and C_2 that such functions can achieve and then by using an approximation, in which we consider only the input examples with a balanced or almost balanced number of input bits ON and OFF, we analyze which symmetric functions have high C_{12} complexity measure.

For the case of C_1 it is trivial to see that the parity function and its complement, for which $C_1 = 1$, are the only Boolean functions for which C_1 is maximum, and they are symmetric. The maximum possible value for C_2 among symmetric functions is 0.5, as we now show.

For a given number of input bits, N, we organize the examples in levels according to the number of bits ON (number of bits equal to 1), N_n. The number of OFF bits in a given example is then equal to $N_f = N - N_n$. A useful picture to see how the examples are organized in levels is the poset diagram (see Fig. 6) where neighboring examples at a Hamming distance 1 are linked by bonds or edges. (If we identify examples with subsets of $[N]$, the poset is the power set of $[N]$ with respect to set inclusion.) In Fig. 6 the poset is shown for the case $N = 4$.

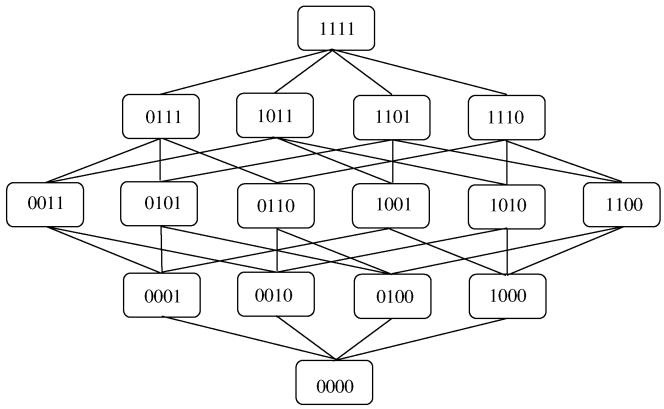

Fig. 6. Poset for N = 4.

The number of examples $N_{H=2}(x)$ at Hamming distance 2 from any given example x is $\binom{N}{2}$. For any example x, this number may be decomposed as

$$N_{H=2}(x) = N^{SL}_{H=2}(x) + N^{DL}_{H=2}(x) \tag{12}$$

where $N^{SL}_{H=2}(x)$ is the number of examples at distance 2 in the same level of the poset as x, and $N^{DL}_{H=2}(x)$ is the number of examples at distance 2 and in a different level (either level $i+2$ or $i-2$). Now, $N^{SL}_{H=2}(x)$ and $N^{DL}_{H=2}(x)$ depend only on the level of the poset to which x belongs. Explicitly, for all x in level i

$$N^{SL}_{H=2}(x) = i(N - i) \tag{13}$$

as this is the number of examples in the same level that have one different bit ON and one different bit OFF but the same total number of bits ON. For x in level i

$$N^{DL}_{H=2}(x) = \binom{i}{2} + \binom{N-i}{2} \tag{14}$$

as this is the number of different examples that can be obtained from x by flipping two ON bits to OFF or by flipping two OFF bits to ON. (The binomial coefficient $\binom{m}{k}$ is interpreted as 0 if $k > m$.)

For a symmetric Boolean function, examples in the same level have the same output. It follows that, in considering the complexity measure $C_2[f]$, only examples at distance 2 and in a different level need be considered. Therefore, if L_i denotes level i of the poset, we have

$$C_2[f] \leq \frac{1}{2^N} \frac{1}{\binom{N}{2}} \sum_{i=0}^{N} \sum_{x \in L_i} N^{DL}_{H=2}(x) \tag{15}$$

$$= \frac{1}{2^N} \frac{1}{\binom{N}{2}} \sum_{i=0}^{N} \binom{N}{i} \left(\binom{i}{2} + \binom{N-i}{2} \right) \tag{16}$$

$$= \frac{1}{2^N} \sum_{i=0}^{N} C(i, N) \tag{17}$$

where

$$C(i, N) = \frac{1}{\binom{N}{2}} \binom{N}{i} \left(\binom{i}{2} + \binom{N-i}{2} \right) \tag{18}$$

1 0 1
1 1 1 1
1 2 2 2 1
1 3 4 4 3 1
1 4 7 8 7 4 1
1 5 11 15 15 11 5 1
1 6 16 26 30 26 16 6 1
1 7 22 42 56 56 42 22 7 1
1 8 29 64 98 112 98 64 29 8 1
1 9 37 93 162 210 210 162 93 37 9 1
1 10 46 130 255 372 420 372 255 130 46 10 1

Fig. 7. Triangle created from (18) starting from $N = 2$ up to $N = 12$. The ith coefficient $C_{i,j}$ in row j ($j = 2$ being the first row) has value $C_{i,j} = C(i,j) = (1/\binom{j}{2})\binom{j}{i}(\binom{i}{2} + \binom{j-i}{2}) = ((i^2 + (j-i)^2 - j)/(j^2 - j))\binom{j}{i}$.

is 2^N times the maximum possible contribution to $C_2[f]$ from the examples in level i.

The numbers $C(i,N)$ of (18) are indicated (for $N = 2$ to 12) in the triangular array of Fig. 7.

It is apparent that, in this triangle, an entry in a given row can be obtained by adding the two elements that are located above it in the preceding row. It can also be seen that the sum of the numbers in each row is double that of the sum in the preceding row. Both these observations are easily verified. It can be checked that for any N and $0 \le i \le N-1$

$$C(i,N) + C(i+1,N) = C(i+1,N+1). \tag{19}$$

Additionally, the sum of the numbers in row N can be seen to be 2^{N-1} as follows. We note that

$$\sum_{i=0}^{N} \binom{N}{i}\binom{i}{2} = 2^{N-2}\binom{N}{2}$$

since both sides of this identity are different expressions for the number of pairs $(\{x,y\}, S)$ where $x, y \in [N] = \{1, 2, \ldots, N\}$ and $x, y \in S \subseteq [N]$. (Clearly, for each choice of $\{x,y\}$ there are 2^{N-2} possible S, giving the right-hand side. But the same quantity can be calculated as follows. Choose some subset S of $[N]$ and then some 2-subset $\{x,y\}$ of S. This second approach gives the left-hand side.) Also

$$\sum_{i=0}^{N} \binom{N}{i}\binom{N-i}{2} = 2^{N-2}\binom{N}{2}$$

since both sides are the number of pairs $(\{x,y\}, S)$ where $x, y \in [N]$, $S \subseteq [N]$ and $x, y \notin S$. It follows that

$$\begin{aligned}\sum_{i=0}^{N} C(i,N) &= \frac{1}{\binom{N}{2}} \sum_{i=0}^{N} \binom{n}{i}\left(\binom{i}{2} + \binom{N-i}{2}\right) \\ &= \frac{1}{\binom{N}{2}} 2 2^{N-2}\binom{N}{2} \\ &= 2^{N-1}.\end{aligned} \tag{20}$$

This shows, in particular, that for any symmetric Boolean function, we have

$$C_2[f] \le \frac{1}{2^N} \sum_{i=0}^{N} C(i,N) = \frac{1}{2}. \tag{21}$$

The maximum value of 0.5 for C_2 can be achieved for any N by the symmetric functions with the property that the outputs chosen for the different levels alternate between 0 and 1 every two levels.

VI. Approximating the Complexity of Symmetric Boolean Functions

We now analyze approximately the C_{12}-complexity of symmetric functions, where $C_{12}[f] = C_1[f] + C_2[f]$. The idea of the approximation is to focus on the middle layers of the poset, these being the largest, to compute locally the complexity around these layers.

The middle layer (in the case of even N) and the middle two layers (in the case of odd N) are the most populated ones. Since the complexity measure that we are considering involves examples up to Hamming distance 2, to compute our approximation, we consider only the layers within distance 2 of these largest ones. We first consider the case N even and base our analysis, therefore, on the levels $(N/2)-2$, $(N/2)-1$, $N/2$, $(N/2)+1$, $(N/2)+2$. We analyze the validity of the approximation by considering the fraction of pairs of examples at Hamming distance 1 (those contributing to C_1) that are taken into account by the approximation in proportion to the total number of pairs of examples at Hamming distance 1.

Fraction of pairs in the approx.

$$= \frac{2\sum_{i=\frac{N}{2}-2}^{\frac{N}{2}-1} \{\text{pairs (i, i+1)}\}}{\text{Total number of pairs}} \tag{22}$$

where pairs (i, i+1) indicates the number of pairs between levels i and $i+1$. For the case of $N \gg 1$ (22) leads to

$$\text{Fraction of pairs in the approx.} \sim \frac{2N\frac{2^N}{\sqrt{N}}}{\frac{N}{2}2^N} = \frac{4}{\sqrt{N}}. \tag{23}$$

Statistical tests of significance for the approximation were carried for the cases $N = 14$ and $N = 20$ and they are detailed later.

We express the approximated complexity of the symmetric Boolean functions in terms of the "interactions" of the examples at Hamming distance 1 and 2 of these five levels and introduce a function F that will account for this value. $F(X_1, X_2, X_3, X_4)$, where $X_1, X_2, X_4 \in \{0,1,2\}$ and $X_3 \in \{0,1\}$ reflect the values of the interactions between the different levels considered. Explicitly, X_1 reflects the value of the interaction at Hamming distance 1 between levels $(N/2)-1$ and $N/2$ and between levels $N/2$ and $(N/2)+1$. That is, X_1 takes value 0 if the Boolean function assigns the same output to all these three layers; it has value 1 if, for one of these pairs of layers, the outputs are different and for the other pair they are equal; and it has value 2 if for each pair of layers, the outputs are different. Similarly, X_2 reflects the value of the interaction at Hamming distance 1 between levels $(N/2)-2$ and $(N/2)-1$ and between levels $(N/2)+2$ and $(N/2)+1$. The parameters X_3 and X_4 describe distance-2 interactions: X_3 reflects the value of the interaction at Hamming distance 2 between levels $(N/2)-1$ and $(N/2)+1$, and X_4 accounts for the interaction at Hamming distance 2 between the middle level $N/2$ and levels $(N/2) \pm 2$.

TABLE I
THE ASSIGNMENTS OF VALUES TO THE MIDDLE FIVE LAYERS OF THE POSET AND THE CORRESPONDING PARAMETER VECTORS

Output values	Parameters (X_1, X_2, X_3, X_4)
...10101 ...	(2,2,0,0)
...10110 ...	(1,2,1,1)
...10100 ...	(2,1,0,1)
...00110 ...	(1,1,1,2)
...00100...	(2,0,0,2)
...01110 ...	(0,2,0,2)
...10111 ...	(1,1,1,0)
...00111 ...	(1,0,1,1)
...01111 ...	(0,1,0,1)
...11111 ...	(0,0,0,0)

Now we approximate and assess numerically, for all configurations of assignments of output values to these middle layers, the complexity of symmetric functions whose outputs are consistent with these. Without any loss of generality we assume outputs of examples in level $N/2$ to be 1. By symmetry, we then need only consider the ten configurations of output values to the five layers that are shown in the first column of Table I. Here, the assignment $\dots 10\,100 \dots$ means, for example, that layer $N/2-2$ is assigned 1, layer $N/2-1$ is assigned 0, and so on. We also indicate, in the second column, the corresponding parameters (X_1, X_2, X_3, X_4).

We use two different ways of measuring the C_{12} complexity in these five central layers. These approximations are made by two functions $F^{(1)}$ and $F^{(2)}$ of the parameters $X = (X_1, X_2, X_3, X_4)$. These are given (for $i = 1, 2$) by

$$F^{(i)}(X_1, X_2, X_3, X_4) = f(X_1, X_2) + g_i(X_3, X_4)$$

where f measures the C_1 complexity, relativized to these layers, and g_1, g_2 are two approximations for the C_2 complexity, locally around these layers. The function f is defined by

$$f(X_1, X_2) = \frac{X_1\binom{N}{\frac{N}{2}}\frac{N}{2} + X_2\binom{N}{\frac{N}{2}-1}\left(\frac{N}{2}-1\right)}{2\binom{N}{\frac{N}{2}}\frac{N}{2} + 2\binom{N}{\frac{N}{2}-1}\left(\frac{N}{2}-1\right)}. \quad (24)$$

The approximation g_1 is defined as follows:

$$g_1(X_3, X_4) = \frac{1}{2}\frac{X_3\binom{N}{\frac{N}{2}-1}\binom{\frac{N}{2}+1}{2} + X_4\binom{N}{\frac{N}{2}}\binom{\frac{N}{2}}{2}}{\binom{N}{\frac{N}{2}-1}\binom{\frac{N}{2}+1}{2} + 2\binom{N}{\frac{N}{2}}\binom{\frac{N}{2}}{2}}. \quad (25)$$

Here, the leading factor of 1/2 reflects the fact that, for symmetric functions, C_2 can be no more than 1/2 (as shown in the previous section). The function g_2 is given by

$$g_2(X_3, X_4) = \frac{X_3\binom{N}{\frac{N}{2}-1}\binom{\frac{N}{2}+1}{2} + X_4\binom{N}{\frac{N}{2}}\binom{\frac{N}{2}}{2}}{\binom{N}{\frac{N}{2}-1}\binom{\frac{N}{2}+1}{2} + 2\binom{N}{\frac{N}{2}}\binom{\frac{N}{2}}{2} + S} \quad (26)$$

where S, the number of pairs of examples in these five layers which are at distance 2 and in the same layer is

$$S = \binom{N}{\frac{N}{2}+2}\left(\frac{N}{2}+2\right)\left(\frac{N}{2}-2\right) + \binom{N}{\frac{N}{2}+1} \times \left(\frac{N}{2}+1\right)\left(\frac{N}{2}-1\right) + \frac{1}{2}\binom{N}{\frac{N}{2}}\left(\frac{N}{2}\right)^2. \quad (27)$$

Generally, we would expect g_2 to underestimate C_2, because although S accounts for all distance-2 pairs within the same layer, in these five layers, g_2 does not account for possible distance-2 interactions between points of these five layers and points outside these layers. (There is no term corresponding to a possible interaction between layer $N/2-1$ and layer $N/2-3$, and so on.)

Table II shows the values of f, g_1, g_2, $F^{(1)}$ and $F^{(2)}$ for the configurations of interest (see Table I), for the case $N = 14$. (The first column indicates the appropriate parameter values.) In the final column of the table, we give the mean values of C_1, C_2, C_{12} over all symmetric functions on $N = 14$ variables which extend the given configuration on the five central layers. So, for instance, for the last entry of the first column, we consider all those symmetric Boolean functions on $\{0,1\}^{14}$ which assign values 1, 0, 1, 0, 1 to layers 5, 6, 7, 8, 9 (respectively)—that is, all those that extend the pattern $\dots 10101 \dots$ of the central layers—and we compute the mean values of C_1, C_2, and C_{12} over all such functions.

Table II shows the mean values of the complexity measures for extensions of the given configurations of the middle layers. More information is provided by the distribution of these. For instance, Fig. 8 shows the distribution of complexities C_1, C_2 and C_{12} for extensions of the configuration $\dots 10100 \dots$ [corresponding to the third row of Table II with X equal to (2, 1, 0, 1)]. As noted in Table II, in this case the means of C_1, C_2 and C_{12} are (respectively) 0.709, 0.194, and 0.903. The corresponding standard deviations are 0.067, 0.047, and 0.082. The minimum values of each are 0.576, 0.097 and 0.733, and the maximum values are 0.843, 0.290, and 1.0, respectively. We also assessed statistically how good the approximations of $F^{(1)}$ and $F^{(2)}$ using only examples from the five middle layers were in relationship to the real values. For the case of $N = 14$ we computed the regression value between the true complexity of all the symmetric Boolean functions and their respective approximated value, finding the Pearson correlation coefficient to be highly significant $R = 0.926$ and $R = 0.929$ for F^1 and F^2, respectively. For the case $N = 20$ we computed the correlation between the values given by the approximations and the complexity values of all the symmetric Boolean functions compatible with each of the 10 different function approximations defined in Table I (2^{16} functions were considered for each of the cases) to find values of the Pearson correlation equal to $R = 0.896$ and $R = 0.899$ for $F^{(1)}$ and $F^{(2)}$, respectively. The correlation was statistically significant in all cases ($p < 0.001$).

So far we have considered the case of even N. To analyze the case of odd N, we would consider the four most populated levels $(N-i)/2$ with $i = \{\pm 1, \pm 3\}$. Suppose (without loss of generality) that a Boolean function f assigns value 1 to the examples in level $(N-1)/2$. If f then alternates its values between immediate layers of the poset, we obtain the parity function or its complement, for which $C_1 = 1$ and $C_2 = 0$. Another interesting configuration is that in which f assigns value 1 to examples in layer $(N+1)/2$ and then alternates its values every two layers. As we have seen, this gives the maximum possible value (for a symmetric function) of C_2, namely $C_2 = 0.5$. The function also has $C_1 = 0.5$, as we now show. Suppose that N is of the form $4k+1$ for a positive integer k. (A very similar

TABLE II
APPROXIMATIONS TO THE COMPLEXITIES C_1, C_2 AND C_{12} OF FUNCTIONS HAVING A GIVEN CONFIGURATION OF OUTPUTS ON THE MIDDLE FIVE LAYERS (IN THE CASE $N = 14$). (SEE TEXT AND TABLE I.) THE LAST COLUMN SHOWS THE MEAN VALUES OF COMPLEXITIES OF ALL FUNCTIONS CONSISTENT WITH THESE GIVEN OUTPUT PATTERNS OF THE MIDDLE LAYERS. APPROXIMATIONS AND MEANS OF C_{12} COMPLEXITIES ARE HIGHLIGHTED IN BOLD

X	f	g_1	g_2	$F^{(1)}$	$F^{(2)}$	mean $C_1/C_2/C_{12}$
(2,2,0,0)	1	0	0	**1**	**1**	0.867/0.097/**0.963**
(1,2,1,1)	0.714	0.342	0.286	**1.056**	**1**	0.657/0.306/**0.963**
(2,1,0,1)	0.786	0.158	0.132	**0.944**	**0.918**	0.709/0.194/**0.903**
(1,1,1,2)	0.5	0.5	0.418	**1**	**0.918**	0.5/0.403/**0.903**
(2,0,0,2)	0.571	0.316	0.264	**0.887**	**0.835**	0.552/0.290/**0.843**
(0,2,0,2)	0.429	0.316	0.264	**0.744**	**0.692**	0.448/0.290/**0.738**
(1,1,1,0)	0.5	0.184	0.154	**0.684**	**0.654**	0.5/0.210/**0.710**
(1,0,1,1)	0.286	0.342	0.286	**0.628**	**0.571**	0.342/0.306 /**0.649**
(0,1,0,1)	0.214	0.158	0.1329	**0.372**	**0.346**	0.290/0.194/**0.484**
(0,0,0,0)	0	0	0	**0**	**0**	0.133/ 0.097/**0.230**

argument can be made if, instead, N is of the form $4k+3$.) The contributions to $C_1[f]$ are of two types: Those arising from, for each example in layer $2i$, the $2i$ neighbors in layer $2i-1$ (for $1 \leq i \leq 2k$) and (equally) those arising from, for each example in layer $2i-1$, the $N-(2i-1)$ neighbors in layer $2i$. These two types of contribution are equal, since each is the total number of edges between layers $2i$ and $2i-1$. We, therefore, have

$$2^N N C_1[f] = 2\sum_{i=1}^{2k} 2i \binom{4k+1}{2i} = 2\sum_{r \text{ even}} r\binom{N}{r}. \quad (28)$$

Now, we can argue combinatorially, as follows: $\sum_{r \text{ even}} r\binom{N}{r}$ is equal to the number of pairs (x, S) where $S \subseteq [N]$ is of even cardinality and $x \in S$. (For, there are $\binom{N}{r}$ subsets S of cardinality r and, for each, there are r choices for x.) Now, suppose that $x \in [N]$. Let y be any element of $[N]$ not equal to x. Then the 2^{N-1} subsets T of $[N]$ containing x can be partitioned into pairs $\{R, R \cup \{y\}\}$ where R runs through all subsets of $[N]$ containing x but not containing y. Since precisely one member of each such pair is of even cardinality, and since there are N choices for x, it follows that the number of pairs (x, S) where $x \in S \subseteq [N]$ and $|S|$ is even is exactly $N2^{N-1}/2$. Hence

$$C_1[f] = \frac{1}{2^N N} 2\left(N\frac{2^{N-1}}{2}\right) = \frac{1}{2}$$

as claimed.

VII. DISCUSSION AND CONCLUSION

The ability to generalize is a very interesting and important property of many intelligent systems like humans and neural networks, and a lot of effort has been devoted to its understanding. We analyzed in this paper a recently proposed measure for the complexity of Boolean functions related to the difficulty of generalization when neural networks are trained using examples of the function. We studied the first-order (C_1) and second-order (C_2) terms of the complexity measure and demonstrated the importance of the second term in obtaining accurate comparisons with the generalization error. Furthermore, we analyzed the relationship between C_2 and the amount of randomness introduced in three different classes of functions and found a very clear correlation, suggesting that the second-order complexity term can be used as an estimate of the randomness existing in a Boolean function. This is, we believe, an important feature, since measuring randomness is a very complicated and delicate matter [34], [35], and also because it is important to quantify randomness in applications such as cryptography [12].

By using an assumption on the nature of the most complex functions based on some results from statistical mechanics, we have been able to obtain very complex Boolean functions, with a complexity larger than 1. We showed empirically that the difficulty of generalization for these functions was related to the complexity measure (once the functions were modified by adding a controlled element of randomness). We have seen in the experiments in Section III that the use of the first and second term were enough to get a nice match between generalization and complexity for different classes of functions but when in Section IV we analyzed the parity function using some irrelevant variables we found that the two first terms were not enough and that higher-order terms were needed to obtain an accurate match. Our opinion is that when analyzing large classes of functions, in which the values are averaged across many different functions around a certain complexity the two first terms are enough, but when the analysis is done for very few functions, as is the case of the functions analyzed in Section III it might be necessary to include terms of third-order or higher, but this is an issue that needs further investigations. Regarding the values of the constants α_i that weigh the different terms in the complexity measure in relationship to the first term, we have been able to estimate the value of α_2 to be close to 0.5 (the estimations were 0.5 and 0.5625 for $N = 8$ and 10 inputs, respectively). We note that the estimation relies on some approximations, and so they should be not considered as definitive values, and that is why we take a conservative approach in some parts of this paper and use the value of α_2 used in previous analysis [4], [5] in which α_2 was equal to 1. This value permits, as demonstrated in Section III, to obtain a good match between generalization error and complexity for different classes of functions. We have also observed in the different studies carried out in this paper and in [4], [5] that for large classes of functions higher-order terms normally have a similar value to the first and second, in particular the first, third, fifth terms tends to have similar values and the same for the even terms (second, fourth, sixth, etc.). The consequence of the previous observations in relationship to the values of the constants α_i would be that consecutive terms will have to be weighted by an amount proportional to the relationship between first and second terms (i.e, $(\alpha_4/\alpha_3) \sim (\alpha_2/\alpha_1)$; $\alpha_1 = 1$) but we do not have yet experimental or theoretical justification for a proper choice of their absolute values. As higher-order terms

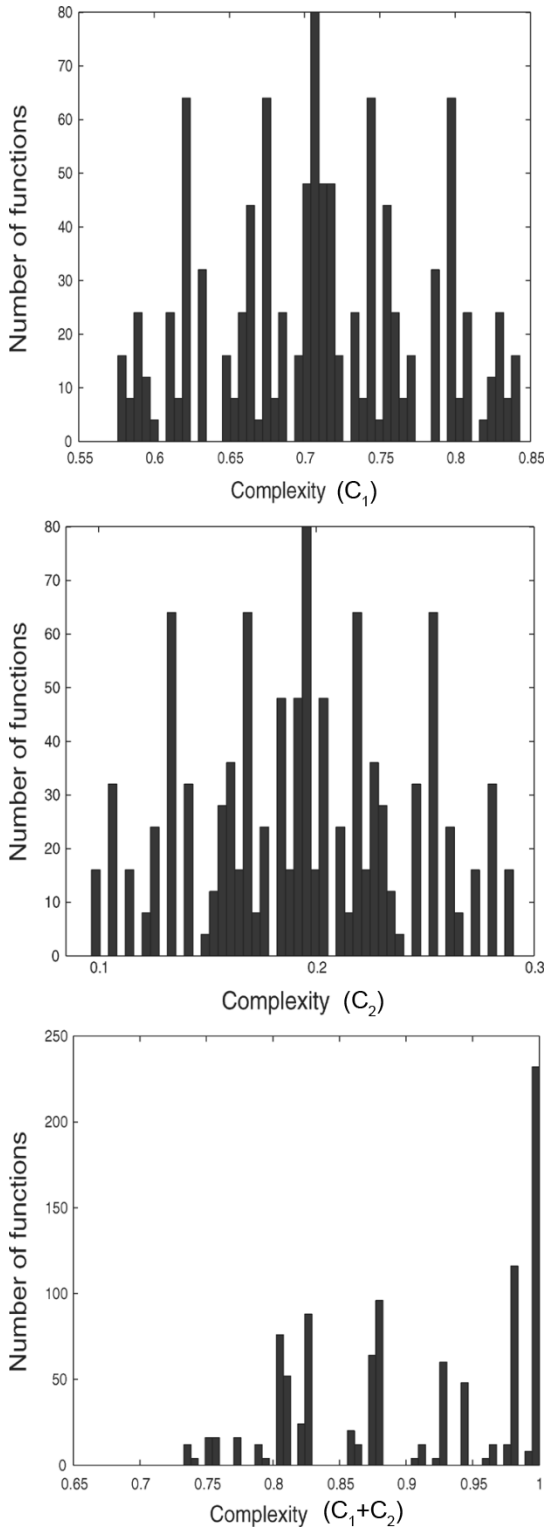

Fig. 8. Number of functions compatible with the values of the middle layer being 10100, and having given C_1, C_2 and $C_{12} = C_1 + C_2$ complexities (top, middle, bottom histograms, respectively).

weight the effect of examples at larger Hamming distances the logic indicates that their weight should be not higher than those for the first and second terms.

For the class of symmetric Boolean functions, we first obtained a general bound on the maximum value that the second-order term of the complexity can take and, secondly, by focusing on the most populated levels of inputs, we found approximate values for the complexity of certain symmetric functions (and, in particular, we were able to obtain an indication that some complex symmetric functions existed). We have performed statistical tests that showed that these approximations compared well (for $N = 14$ and $N = 20$) with the computationally calculated actual values of the complexities.

As a whole, the results presented in this paper show that the complexity measure introduced in [4] can be used to characterize different classes of Boolean functions in relationship to the complexity of generalization, and we think that this may lead to new lines of research contributing to a better understanding of the difficulty of learning Boolean functions by neural networks and in particular to the study of how changes in the neural architecture affect the computability of functions [36], [37].

Classical statistical learning theory (see [25] or [26], for example) suggests that the difference between generalization error and training error (that is, between the error on a test set and on the training set) can be bounded (with high probability) by a quantity that depends on the size of the network (more precisely, its VC-dimension) and is independent of the function being learned. In particular, this leads to an upper bound on the generalization error that increases as the training error increases. It is possible that a perceived correlation between complexity and generalization error is in large part due to a correlation between complexity and training error (a higher complexity indicating that it is more difficult to "fit" the network to the function). More recently, certain nonuniform bounds depending on the sizes of the weights in the network after training have been obtained [38]. (These weights can depend on the particular function being learned, and in this sense are nonuniform.) As mentioned, the classical VC-dimension based results from statistical learning theory give a (high-probability) upper bound on the difference between generalization and training errors, and this bound depends on the size of the network, but not on the function. We conjecture that such a uniform bound can be replaced by one that involves not only the size of the network, but also the complexity of the particular Boolean function being learned. Further experiments would indicate whether this is the case.

We are currently exploring different extensions of this work, such as the generalization of the measure to continuous input functions, the use of the complexity measure for individual patterns to improve learning, the construction of neural networks architectures for restricted classes of Boolean functions, the use of the second term of the complexity measure to estimate randomness, and also applications to the statistical mechanics of magnetic systems. In particular, we think that the complexity measure will be an important element in order to study how changes in the architecture (number of hidden layers and number of neurons on each layer) affect the generalization ability and we are currently doing such analysis on the class of symmetric Boolean functions.

ACKNOWLEDGMENT

Fruitful discussions with Dr. S. A. Cannas are gratefully acknowledged. The authors also acknowledge valuable comments and suggestions from three anonymous reviewers.

REFERENCES

[1] I. Wegener, *The Complexity of Boolean Functions*. New York: Wiley, 1987.

[2] J. Hastad, "Almost optimal lower bounds for small depth circuits," *Adv. Comput. Res.*, vol. 5, pp. 143–170, 1989.

[3] I. Parberry, *Circuit Complexity and Neural Networks*. Cambridge, MA: MIT Press, 1994.

[4] L. Franco, "Generalization ability of Boolean functions implemented in feedforward neural networks," *Neurocomputing*, 2006, to be published.

[5] L. Franco and S. A. Cannas, "Non-glassy ground state in a long-range antiferromagnetic frustrated model in the hypercubic cell," *Physica A*, vol. 332, pp. 337–348, 2004.

[6] ——, "Generalization and selection of examples in feedforward neural networks," *Neural Comput.*, vol. 12, pp. 2405–2426, 2000.

[7] ——, "Generalization properties of modular networks implementing the parity function," *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1306–1313, Nov. 2001.

[8] J. Kahn, G. Kalai, and N. Linial, "The influence of variables on Boolean functions," in *Proc. 29th IEEE Symp. Foundations of Computer Science (FOCS'88)*, 1988, pp. 68–80.

[9] N. Linial, Y. Mansour, and N. Nisan, "Constant depth circuits, Fourier transform and learnability," *J. ACM*, vol. 40, pp. 607–620, 1993.

[10] L. G. Valiant, "A Theory of the learnable," *Commun. ACM*, vol. 27, pp. 1134–1142, 1984.

[11] R. B. Boppana, "The average sensitivity of bounded-depth circuits," *Inf. Process. Lett.*, vol. 63, pp. 257–261, 1997.

[12] A. Bernasconi, C. Damm, and I. Shparlinski, "The average sensitivity of square-freeness," *Comput. Complex.*, vol. 9, pp. 39–51, 2000.

[13] J. Feldman, "Minimization of Boolean complexity in human concept learning," *Nature*, vol. 407, pp. 572–573, 2000.

[14] O. C. Martin, R. Monasson, and R. Zecchina, "Statistical mechanics methods and phase transitions in optimization problems," *Theor. Comput. Sci.*, vol. 265, pp. 3–67, 2001.

[15] S. Mertens, "Computational complexity for physicists," *Comput. Sci. Eng.*, vol. 4, pp. 31–47, 2002.

[16] K. Y. Siu, V. P. Roychowdhury, and T. Kailath, "Depth-size tradeoffs for neural computation," *IEEE Trans. Comput.*, vol. 40, no. 12, pp. 1402–1412, Dec. 1991.

[17] M. Ben-Or and N. Linial, "Collective coin flipping," in *Randomness and Computation*, S. Micali, Ed. New York: Academic, 1989, pp. 91–115.

[18] M. Anthony, G. Brightwell, and J. Shawe-Taylor, "On specifying Boolean functions by labeled examples," *Discrete Appl. Math.*, vol. 61, pp. 1–25, 1995.

[19] V. I. Torvik and E. Triantaphyllou, "Minimizing the average query complexity of learning monotone Boolean functions," *INFORMS J. Comput.*, vol. 14, pp. 142–172, 2002.

[20] A. Gál and A. Rosén, "A theorem on sensitivity and applications in private computation," in *Proc. 31st ACM Symp. Theory of Computing*, 1999, pp. 348–357.

[21] M. L. Furst, J. C. Jackson, and S. W. Smith, "Improved learning of AC^0 functions," in *Proc. 4th Annu. Workshop on Computational Learning Theory*, 1991, pp. 317–325.

[22] J. Bruck and R. Smolensky, "Polynomial threshold functions, AC^0 functions, and spectral norms," *SIAM J. Comput.*, vol. 21, pp. 33–42, 1992.

[23] M. Saks, "Slicing the hypercube," in *Surveys in Combinatorics (Invited Talks of the 1993 British Combinatorial Conf.)*. Cambridge, U.K.: Cambridge Univ. Press, 1993, pp. 211–255.

[24] V. P. Roychowdhury, K. Y. Siu, and A. Orlitsky, "Neural models and spectral methods," in *Theoretical Advances in Neural Computation and Learning*, V. P. Roychowdhury, K. Y. Siu, and A. Orlitsky, Eds. Boston, MA: Kluwer Academic, 1994.

[25] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[26] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[27] S. Haykin, *Neural Networks: A Comprehensive Foundation*: Macmillan, 1994.

[28] H. Zhu and W. Kinzel, "Antipredictable sequences: harder to predict than random sequences," *Neural Comput.*, vol. 10, pp. 2219–2230, 1998.

[29] P. Chandra, P. Coleman, and I. Ritchey, "The anisotropic kagome antiferromagnet: a topical spin glass?," *J. Phys. I*, vol. 3, pp. 591–610, 1993.

[30] F. Barahona, "On the computational complexity of ising spin glasses," *J. Phys. A: Math. Gen.*, vol. 15, pp. 3241–3253, 1982.

[31] S. Istrail, "Statistical mechanics, three-dimensionality and NP-completeness I. Universality of intractability for the partition function of the Ising model accross nonplanar lattices," in *Proc. 32nd ACM Symp. Theory of Computing (STOC 2000)*, 2000, pp. 87–96.

[32] S. D. Cotofana and S. Vassiliadis, "Periodic symmetric functions, serial addition, and multiplication with neural networks," *IEEE Trans. Neural Netw.*, vol. 9, no. 6, pp. 1118–1128, Nov. 1998.

[33] L. Franco and S. A. Cannas, "Solving arithmetic problems using feedforward neural networks," *Neurocomput.*, vol. 18, pp. 61–79, 1965.

[34] A. Kolmogorov, "Three approaches to the quantitative definition of information," *Probl. Info. Transmiss.*, vol. 1, pp. 1–17, 1965.

[35] G. Chaitin, "Randomness and mathematical proof," *Scientif. Amer.*, vol. 232, pp. 47–52, 1975.

[36] V. Deolalikar, "Mapping Boolean functions with neural networks having binary weights and zero thresholds," *IEEE Trans. Neural Netw.*, vol. 12, no. 3, pp. 639–642, May 2001.

[37] G.-B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 274–281, Mar. 2003.

[38] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 525–536, Mar. 1998.

Leonardo Franco (M'06) was born in Tucumán, Argentina. He received the M.S. and Ph.D. degrees in physics from the University of Córdoba, Argentina, where he analyzed the generalization properties of feed-forward neural networks.

He then became a Postdoctoral Fellow to SISSA, Trieste, Italy, where he became involved in computational neuroscience. He then moved to the University of Oxford, U.K., as a Research Scientist where he applied and developed information theory methods to the analysis of neuronal recordings. He is currently with Málaga University, Spain, as a Ramón y Cajal researcher working on neural networks, their applications to biomedical problems, and also in computational neuroscience. He has authored approximately 25 publications in journals and international conferences.

Martin Anthony was raised in Paisley, Scotland. He received the B.Sc. degree in mathematics from the University of Glasgow, Scotland, and the Ph.D. degree, also in mathematics from the University of London, U.K.

Since 1990, he has been on the Mathematics faculty of the London School of Economics (LSE). He is now Professor of Mathematics and Convener of the Mathematics Department at LSE. He has published widely on the mathematical theory of machine learning and neural networks, including *Computational Learning Theory: An Introduction* (with N. L. Biggs), *Neural Network Learning: Theoretical Foundations* (with P. L. Bartlett), and *Discrete Mathematics of Neural Networks: Selected Topics*.