The influence of opposite examples and randomness on the generalization complexity of Boolean functions

Leonardo Franco *
Department of Experimental Psychology,
University of Oxford,
South Parks Road,
Oxford OX1 3UD, UK.

Martin Anthony †
Department of Mathematics,
London School of Economics,
Houghton Street,
London WC2A 2AE, UK.

December 2003
CDAM Research Report LSE-CDAM-2003-21

Abstract

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 two components. The first of these 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. We also explore the existence of very complex Boolean functions, considering, in particular, the symmetric Boolean functions.

* e-mail: Leonardo.Franco@psy.ox.ac.uk, Tel.: +44-(0)1865-271419, Fax: +44-(0)1865-310447
† email: m.anthony@lse.ac.uk, Tel.: +44-(0)207-9557623, Fax: +44-(0)207-9556877

1 Introduction

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. Franco (2002) and Franco & Cannas (2003) 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 (Franco & Cannas, 2000; 2001). 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 Linial et al. (1989). Average sensitivity has been linked by Linial et al. (1993) to the complexity of learning in the probabilistic 'PAC' model of learning introduced by Valiant (1984); and many results about the average sensitivity of Boolean functions have been obtained for different classes of Boolean functions (Boppana, 1997 ; Bernasconi et al., 2000). It has been shown in (Franco, 2002) 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 Feldman (2000)). Additionally, very complex Boolean functions are of interest in cryptographic applications (Bernasconi, 2000). In physics, a link has been established between the complexity measure and the Hamiltonian of spin systems (Franco & Cannas, 2003), 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 mehanics and computational complexity, see Martin, Monasson and Zecchina (2001) and Mertens (2002).) 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. In this paper, we pay particular attention to symmetric Boolean functions, an important class about which many general results on sample and computational complexity have been obtained (Siu et al., 1991; Wegener, 1987; Franco, 2002).

2 The complexity measure and its interpretation

In its most general form, the 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. The complexity measure can be written in a general form as:

$$C = C_1 + \sum_{i=2}^{N/2} \alpha_i C_i, \tag{1}$$

where α_i are constant values that weight how pairs of oppositely-classified examples (that is, elements of $\{0,1\}^N$) at Hamming distance i contribute to the total complexity, and N is the number of input bits. 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

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

where $N_{neigh}(i)$ is the number of examples at Hamming distance i from a given example, equal to the Binomial coefficient $C(N,i)= \binom{N}{i}$. 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 $f(y) \neq f(x)$.

The first term, C_1 is proportional to the number of bordering (or boundary) examples, those with an immediate neighbor having opposite output. 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' $s(f)$ (Linial et al., 1993; Ben-Or and Linial, 1989; Kahn, Kalai and Linial, 1988): in fact, $C_1[f] = s(f)/N$. (The average sensitivity $s(f)$ is related to the notion of the 'influence' of a variable; see Kahn, Kalai and Linial (1988), for instance: $s(f)$ is the sum of the influences of the N variables.) The number of bordering examples has been shown to be related to the generalization ability that can be obtained when Boolean functions are implemented in neural networks (Franco & Cannas, 2000), to the number of examples needed to obtain perfect generalization (Franco & Cannas, 2000; 2001), to a bound on the number of examples needed to specify a linearly separable function (Anthony et al., 1995), and to the query complexity of monotone Boolean functions (Torvik & Triantaphyllou, 2002). Moreover, links between the sensitivity of a Boolean function and its learnability in the PAC sense has been established (Linial et al., 1993) and many results regarding the average sensitivity of Boolean functions have been derived (Boppana, 1997; Kahn, Kalai and Linial, 1988; Gál & Rosén, 1999; Bernasconi, 2002).

Following Kahn, Kalai and Linial (1988), the complexity measures can be related to the Fourier coefficients of f. Fourier (or harmonic) analysis of Boolean functions has proven to be extremely useful in a number of areas, such as learning theory (Linial, Mansour and Nisan, 1993; Furst, Jackson and Smith, 1991, for instance) and circuit complexity (Bruck and Smolensky, 1992). (Saks (1993) and Siu et al. (1995) provide good surveys of the harmonic analysis of Boolean functions and its uses.) To describe the Fourier coefficients, it is useful to identify in the obvious way examples of $\{0,1\}^N$ with subsets of $[N] = \{1, 2, \ldots, N\}$ (with an example corresponding to the set of entries equal to 1). For a given $f : \{0,1\}^N \to \{0,1\}$, the Fourier coefficients $\{f_S : S \subseteq [N]\}$ are defined by

$$f_S = 2^{-N} \sum_{B \subseteq [N]} (-1)^{|B \cap S|} f(B). \tag{3}$$

It is shown in Kahn, Kalai and Linial (1988) that the average sensitivity may be written in terms of the Fourier coefficients as $s(f) = 4 \sum_{S \subseteq [N]} |S| f_S^2$, so

$$C_1[f] = \frac{4}{N} \sum_{S \subseteq [N]} |S| f_S^2. \tag{4}$$

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: there are polynomials Q_i of degree i (related to the Krawtchouk polynomials of coding theory; see MacWilliams and Sloane (1977) and Kalai and Linial (1995)) such that

$$C_i[f] = \frac{1}{N\binom{N}{i}} \sum_{S \subseteq [N]} Q_i(|S|) f_S^2. \tag{5}$$

3 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 (Franco, 2002). Experimental results indicate that the first-order complexity term $C_1[f]$ alone does not give as good a correspondence as does the combination $C_1[f] + C_2[f]$.

Figure 1a shows the generalization ability vs. the first order complexity (C_1) obtained from simulations performed for three different classes of functions. The first class of functions (indicated as 'F. const + 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 function ('F. parity + 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 ('F. parity u + 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). Figure 1a shows, for each of these three classes, the generalization ability computed by simulations performed in a neural network architecture with $N = 8$ inputs and 8 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. In figure 1b the generalization ability is plotted against the second order term of the complexity measure, C_2 and it can be observed that the general behaviour of the generalization seems uncorrelated to this second term. But when we plot, in figure 1c, 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.

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 Figures 2a and b. 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.)

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 (Haykin, 1995; Franco & Cannas, 2001), 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 Franco (2002) 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 (1998).

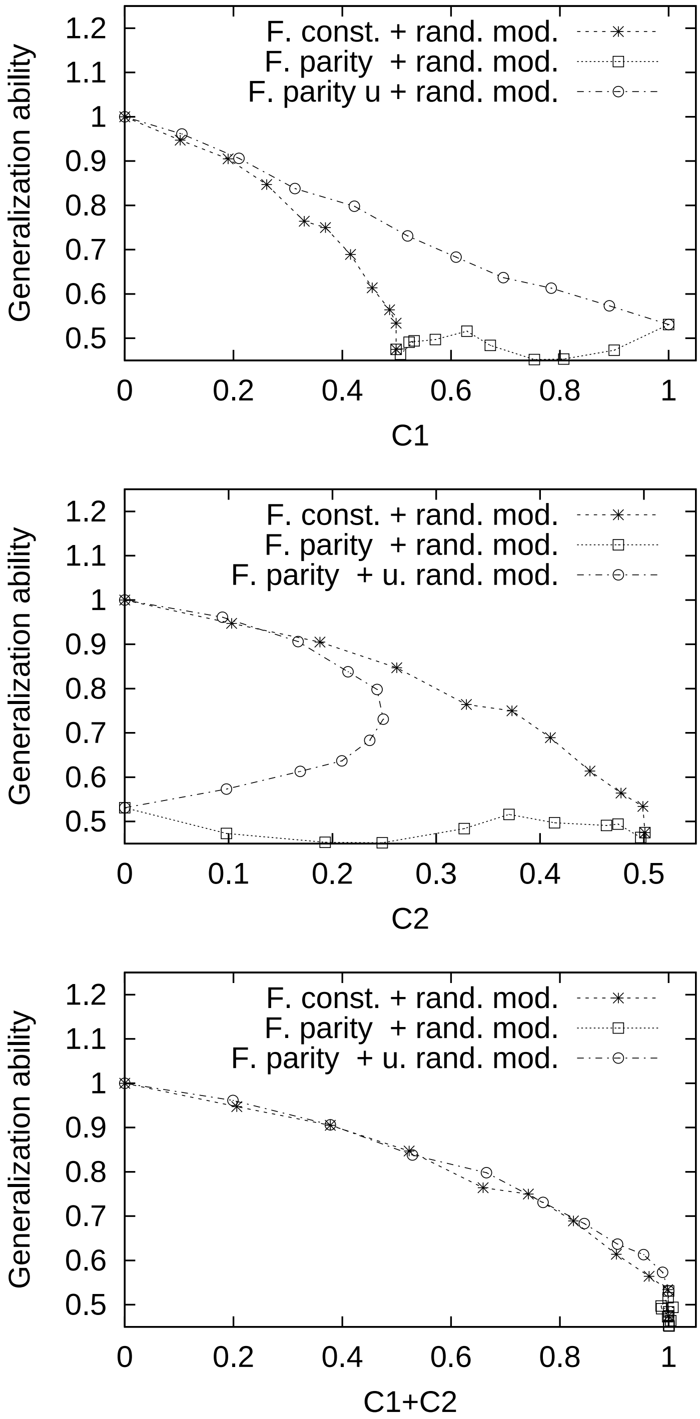

Figure 1: Generalization ability vs. first term (top graph), second term (middle graph), and first plus second terms (bottom graph) of the complexity measure for three different classes of functions. (See text for details.)

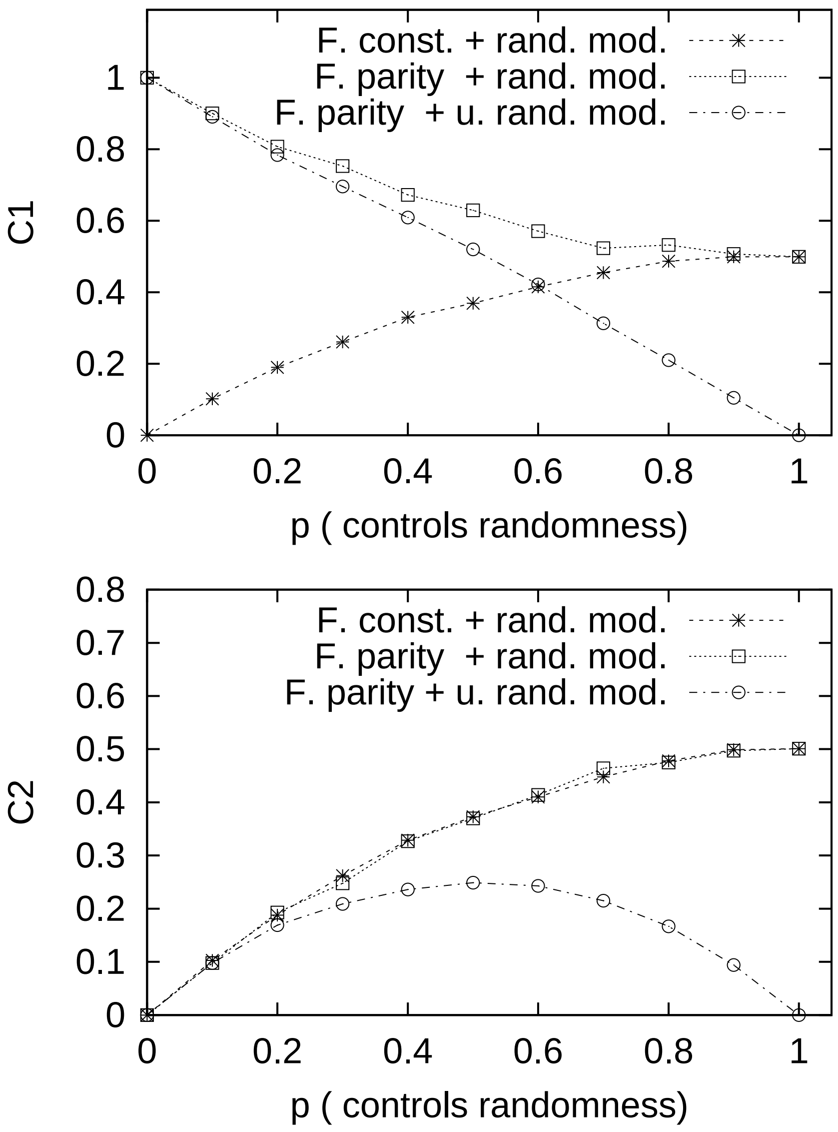

Figure 2: Complexity terms C_1 (top) and C_2 (bottom) vs. the parameter p that controls randomness for the class of functions used in Figure 1. For the first two classes of functions the amount of randomness is directly proportional to p while for the third class the maximum randomness is achieved for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

4 A method for finding very complex Boolean functions and ground states

In this section we investigate how to find Boolean functions with a high 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 Franco, 2002 and Franco & Cannas, 2003 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 (Franco & Cannas, 2003; Chandra et al., 1993). 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 (Barahona, 1982; Istrail,2000).

A Boolean function is said to have A irrelevant attributes if there are $i_1, i_2, \ldots, i_A$ 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 co-ordinates here is arbitrary, the point being that the value of f is independent of these.) The complexity of f can be related to that of f^* as follows:

$$C_{12}[f] = \left(\frac{N-A}{N} + \alpha_2 \frac{2A(N-A)}{N(N-1)}\right) C_1[f^*] + \alpha_2 \frac{(N-A)(N-A-1)}{N(N-1)} C_2[f^*]. \tag{6}$$

To see why (6) holds, it is useful to recall the probabilistic interpretations of C_1 and C_2. For $x \in \{0,1\}^N$, and $1 \leq i, j \leq N$, 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:

$$\begin{aligned} C_1[f] &= \Pr\left(f(x^i) \neq f(x)\right) \\ &= \Pr\left(f(x^i) \neq f(x) \mid i \leq N-A\right) \Pr(i \leq N-A) \\ &\quad + \Pr\left(f(x^i) \neq f(x) \mid i > N-A\right) \Pr(i > N-A) \end{aligned}$$

$$
\begin{aligned}
&= \Pr\Big(f^*(x^i) \neq f^*(x)\Big)\,\frac{N-A}{N} + 0 \\
&= \frac{N-A}{N}\,C_1[f^*].
\end{aligned}
$$

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

From these, (6) 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:

$$
\begin{aligned}
C_{12}[f] &= C_1[f] + \alpha_2 C_2[f] \\
&= \frac{N-A}{N} + \alpha_2 \frac{2\,A(N-A)}{N(N-1)} \qquad (7) \\
&= \frac{(N-A)(N-1+2\alpha_2 A)}{N(N-1)} \qquad (8)
\end{aligned}
$$

This follows from (6), 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 Eq. 7 represent the fraction of pairs of examples with opposite outputs at Hamming distances 1 and 2 respectively. To find the most complex function of this particular type, we take the derivative of C_{12} respect to A (assuming, temporarily, that A is a continuous parameter):

$$
\begin{aligned}
\frac{dC_{12}}{dA} &= \frac{d}{dA}\left(\frac{(N-A)(N-1-2\alpha_2 A)}{N(N-1)}\right) \\
&= \frac{(2\alpha_2-1)N + 1 - 4\alpha_2 A}{N(N-1)}, \qquad (9)
\end{aligned}
$$

Assuming that $\alpha_2 \geq 1/2$ and calculating the maximizing value A_{max}, the root of (9), we obtain:

$$
A_{max} = \frac{N}{2} - \frac{N-1}{4\alpha_2} > 0. \qquad (10)
$$

The complexity of the corresponding function is

$$C_{12}[f(A_{max})] = \frac{\left(\frac{N}{2} + \frac{N-1}{4\alpha_2}\right)\left(2N - 1 - \left(\frac{N-1}{2\alpha_2}\right)\right)}{N(N-1)} \tag{11}$$

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

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

$$\tag{13}$$

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 of the parity function and of a random function are approximately 1.0)

We empirically analyzed the generalization error obtained when Boolean functions are implemented on feed-forward neural networks, to see if this correlates with the complexity measure. For the functions that implement the parity function on $(N - A)$ variables, however, we did not find that the complexity of generalization (that is, the generalization error) correlated well with the complexity measure. It seems that this might be explained by the fact these functions are quite regular. (In fact as the number of relevant variables decreased, the functions seemed to be less complex to implement on neural networks.) Thus we decided to look instead some related functions with a greater element of randomness in their definition. We considered functions that implement the parity function of N 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.

In Fig. 3a we show the values of the complexity $C_1 + C_2$ of parity functions that depend on $N - A$ variables for the cases $N = 14, 10, 8, 4$. In Fig. 3b the generalization error obtained for the Boolean functions that implement the parity function on $(N - A + \delta)$ variables (in the sense described above) is shown for the same cases. 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 error bars plotted show the standard error of the mean (SME), when 50 averages were taken. The SME decreases as 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. As the graphs indicate, we generally found that the value of A_{max} for which the generalization error is a maximum increases with N, to a maximum, and then falls again, as we indicated by the analysis for the parity function with irrelevant attributes.

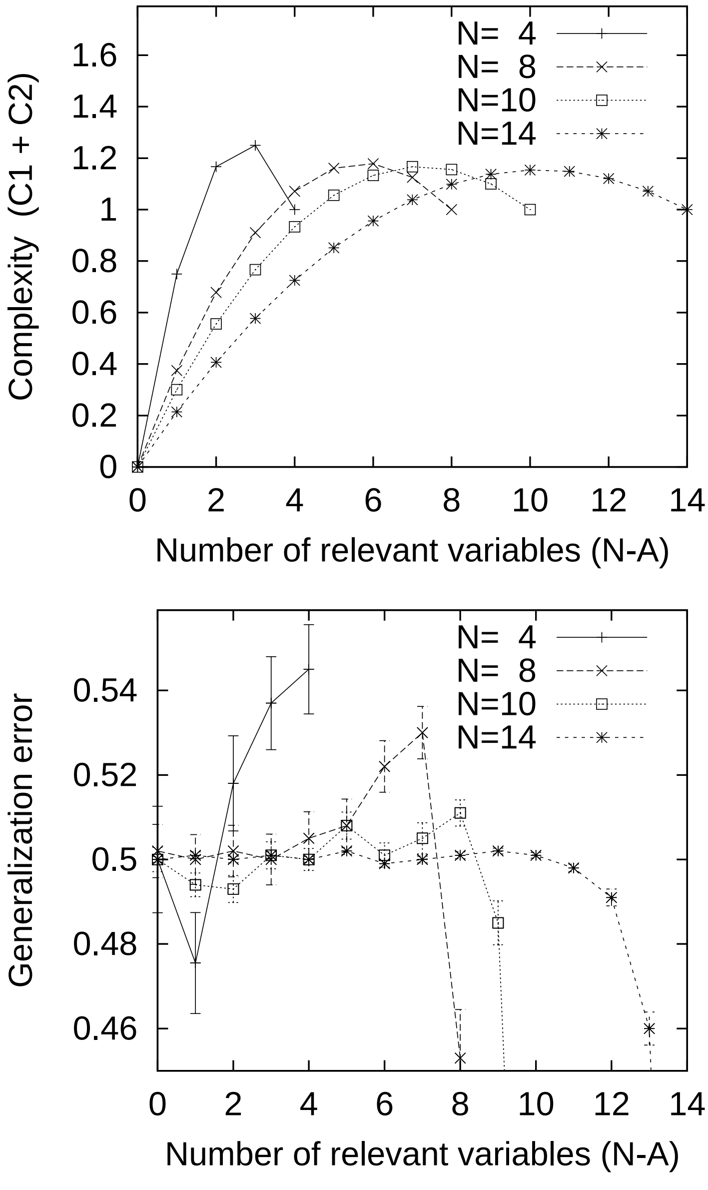

Figure 3: a) The complexity $(C_1 + C_2)$ for Boolean functions that implement the parity function on $N - A$ variables as a function of the number of relevant variables $(N - A)$, for different values of N. b) Generalization error versus the number of relevant variables for the case of the Boolean functions described in the text as the parity on $(N - A + \delta)$ bits, for different values of N.

We can also find functions for which C_2 is very high, independently of C_1. In this case, we repeat the procedure use in Eq. 8-10 to find that functions defined as the parity on $\frac{N}{2}$ variables have a complexity C_2 equal to

$$C_2[f(A = \frac{N}{2})] = \frac{N}{2(N-1)} \tag{14}$$

which is larger than 0.5 for all N. For the case $N = 4$, this gives a function with $C_2 = \frac{2}{3}$ 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 we mentioned above, 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 the next section, the maximum value that can be achieved for C_2 is 0.5.

5 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 have been obtained (Wegener, 1987; Siu et. al. 1991; Cotofana & Vassiliadis, 1998; Franco & Cannas, 1998; Franco & Cannas, 2000; Franco & Cannas, 2001). 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 is 0.5, as we now show.

For a given number of inputs 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 figure 4) 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 figure 4 the poset is shown for the case $N = 4$.

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

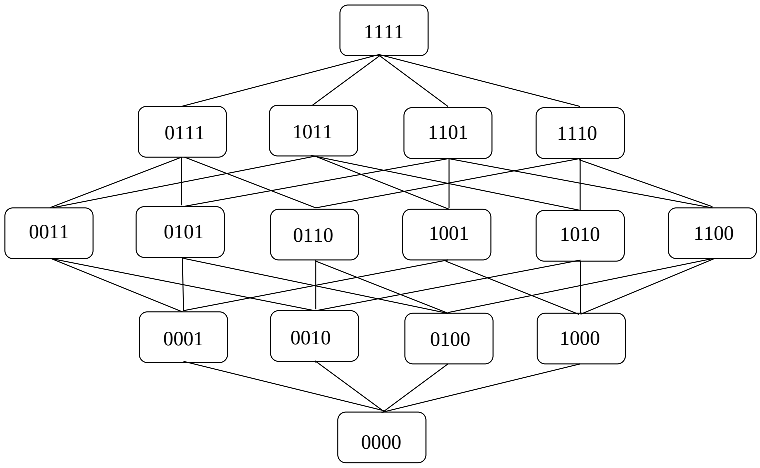

Figure 4: Poset for N=4

example x, this number may be decomposed as

$$N_{H=2}(x) = N^{SL}_{H=2}(x) + N^{DL}_{H=2}(x) \tag{15}$$

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 $H^{DL}_{H=2}(x)$ depend only on the level of the poset to which x belongs. Explicitly, for all x in level i,

$$N^{SL}_{H=2}(x) = i(N-i), \tag{16}$$

as this is the number 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,

$$N^{DL}_{H=2}(x) = \binom{i}{2} + \binom{N-i}{2}, \tag{17}$$

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

$$\begin{aligned} C_2[f] &\leq \frac{1}{2^N}\frac{1}{\binom{N}{2}}\sum_{i=0}^{N}\sum_{x\in L_i} N^{DL}_{H=2}(x) && (18)\\ &= \frac{1}{2^N}\frac{1}{\binom{N}{2}}\sum_{i=0}^{N}\binom{N}{i}\left(\binom{i}{2}+\binom{N-i}{2}\right) && (19)\end{aligned}$$

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

Figure 5: The triangle created from Eq. 21 starting from $N = 2$ up to $N = 12$. The i-th coefficient $C_{i,j}$ in row j ($j = 2$ being the first row) has value $C_{i,j} = C(i,j) = \frac{1}{\binom{j}{2}}\binom{j}{i}\left(\binom{i}{2}+\binom{j-i}{2}\right) = \frac{i^2+(j-i)^2-j}{j^2-j}\ \binom{j}{i}$

$$= \frac{1}{2^N}\sum_{i=0}^{N} C(i,N), \tag{20}$$

where

$$C(i,N) = \frac{1}{\binom{N}{2}}\binom{N}{i}\left(\binom{i}{2}+\binom{N-i}{2}\right) \tag{21}$$

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

The numbers $C(i,N)$ of Equation (21) are indicated (for $N = 2$ to 11) in the triangular array of Figure 5.

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 $0 \le i \le N-1$,

$$C(i,N) + C(i+1,N) = C(i+1,N+1). \tag{22}$$

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 $[X]$ 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

$$\sum_{i=0}^{N} C(i, N) = \frac{1}{\binom{N}{2}} \sum_{i=0}^{N} \binom{n}{i} \left(\binom{i}{2} + \binom{N-i}{2} \right) = \frac{1}{\binom{N}{2}} 2\, 2^{N-2} \binom{N}{2} = 2^{N-1}. \tag{23}$$

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

$$C_2[f] \leq \frac{1}{2^N} \sum_{i=0}^{N} C(i, N) = \frac{1}{2}. \tag{24}$$

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.

6 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. These local approximations will give approximations to the complexity of functions consistent with the same outputs on the middle five layers, provided the values assigned to higher and lower layers in the poset are such that the complexity observed around the middle layers 'extrapolates', at least on average, to the rest of the poset. Such an approach does not give exact results, but we believe the approximations obtained are useful indicators of the values of C_{12} for symmetric functions.

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 $\frac{N}{2} - 2$, $\frac{N}{2} - 1$, $\frac{N}{2}$, $\frac{N}{2} + 1$, $\frac{N}{2} + 2$. (Note that, although these layers are the largest, they only account for a fraction of order $1/\sqrt{N}$ of all 2^N examples on $\{0, 1\}^N$, so for the approximations to be valid, it has to be assumed that the complexity observed locally around these central layers is continued throughout the rest of the poset. A possible alternative approach would be to work with a number of central layers of order $\sqrt{N}$, but this is clearly more difficult.

We express the approximated complexity of the symmetric Boolean functions in terms of the value of the interactions of the examples at Hamming distance 1 and 2 of these 5 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 $\frac{N}{2} - 1$ and $\frac{N}{2}$ and between levels $\frac{N}{2}$ and $\frac{N}{2} + 1$. Thus, X_1 takes value 0 if the Boolean function assigns the same output to all these four 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 $\frac{N}{2} - 2$ and $\frac{N}{2} - 1$ and between levels $\frac{N}{2} + 2$ and $\frac{N}{2} + 1$. 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 $\frac{N}{2} - 1$ and $\frac{N}{2} + 1$, and X_4 accounts for the interaction at Hamming distance 2 between the middle level $\frac{N}{2}$ and levels $\frac{N}{2} \pm 2$.

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 $\frac{N}{2}$ 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 1. Here, the assignment $\ldots 10100 \ldots$ 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).

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

Table 1: The assignments of values to the middle five layers and the corresponding parameter vectors.

We use two different ways of measuring the C_{12} complexity 'locally' 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

$$f(X_1, X_2) = \frac{X_1\binom{N}{\frac{N}{2}}\frac{N}{2} + X_2\binom{N}{\frac{N}{2}-1}\left(\frac{N}{2}-1\right)}{2\binom{N}{\frac{N}{2}}\frac{N}{2} + 2\binom{N}{\frac{N}{2}-1}\left(\frac{N}{2}-1\right)}. \tag{25}$$

The approximation g_1 is defined as follows.

$$g_1(X_3, X_4) = \frac{1}{2}\frac{X_3\binom{N}{\frac{N}{2}-1}\binom{\frac{N}{2}+1}{2} + X_4\binom{N}{\frac{N}{2}}\binom{\frac{N}{2}}{2}}{\binom{N}{\frac{N}{2}-1}\binom{\frac{N}{2}+1}{2} + 2\binom{N}{\frac{N}{2}}\binom{\frac{N}{2}}{2}}. \tag{26}$$

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

$$g_2(X_3, X_4) = \frac{X_3\binom{N}{\frac{N}{2}-1}\binom{\frac{N}{2}+1}{2} + X_4\binom{N}{\frac{N}{2}}\binom{\frac{N}{2}}{2}}{\binom{N}{\frac{N}{2}-1}\binom{\frac{N}{2}+1}{2} + 2\binom{N}{\frac{N}{2}}\binom{\frac{N}{2}}{2} + S}, \tag{27}$$

where S, the number of pairs of examples in these 5 layers which are at distance 2 and in the same layer is

$$S = \binom{N}{\frac{N}{2}+2}\left(\frac{N}{2}+2\right)\left(\frac{N}{2}-2\right) + \binom{N}{\frac{N}{2}+1}\left(\frac{N}{2}+1\right)\left(\frac{N}{2}-1\right) + \frac{1}{2}\binom{N}{\frac{N}{2}}\left(\frac{N}{2}\right)^2. \tag{28}$$

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 2 shows the values of $f, g_1, g_2, F^{(1)}$ and $F^{(2)}$ for the configurations of interest (see Table 1), 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 $\ldots 10101 \ldots$ of the central layers—and we compute the mean values of C_1, C_2 and C_{12} over all such functions.

Table 2 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, Figure 6 shows the distribution of complexities C_1, C_2 and C_{12} for extensions of the configuration $\ldots 11111 \ldots$

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**	0.8666/0.0969/**0.9635**
(1,2,1,1)	0.7143	0.3421	0.2857	**1.0564**	**1**	0.6571/0.3064/**0.9635**
(2,1,0,1)	0.7857	0.1579	0.1319	**0.9436**	**0.9176**	0.7095/0.1936/**0.9031**
(1,1,1,2)	0.5	0.5	0.4176	**1**	**0.9176**	0.5/0.4031/**0.9031**
(2,0,0,2)	0.5714	0.3158	0.2637	**0.8872**	**0.8352**	0.5524/0.2903/**0.8427**
(0,2,0,2)	0.4286	0.3158	0.2637	**0.7444**	**0.6923**	0.4476/0.2903/**0.7379**
(1,1,1,0)	0.5	0.1842	0.1538	**0.6842**	**0.6538**	0.5/0.2097/**0.7097**
(1,0,1,1)	0.2857	0.3421	0.2857	**0.6278**	**0.5714**	0.3429/0.3064 /**0.6493**
(0,1,0,1)	0.2143	0.1579	0.1319	**0.3722**	**0.3462**	0.2905/0.1936/**0.4841**
(0,0,0,0)	0	0	0	**0**	**0**	0.1334/ 0.0969/**0.2303**

Table 2: 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 1.) 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.

(corresponding to the final row of Table 2). As noted in Table 2, in this case the means of C_1, C_2 and C_{12} are (respectively) 0.1334, 0.0969 and 0.2303. The corresponding standard deviations are 0.0471, 0.0669 and 0.1094. The minimum values of each are (of course) 0, and the maximum values are, respectively, 0.2668, 0.1938 and 0.3877.

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 $\frac{N-i}{2}$ with $i = \{\pm 1, \pm 3\}$. Suppose (without loss of generality) that a Boolean function f assigns value 1 to the examples in level $\frac{N-1}{2}$. 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 $\frac{N+1}{2}$ 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 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 $1 \le i \le 2k$) 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

$$2^N N C_1[f] = 2\sum_{i=1}^{2k} 2i\binom{4k+1}{2i} = 2\sum_{r\text{ even}} r\binom{N}{r}. \tag{29}$$

Now, we can argue combinatorially, as follows: $\sum_{r\text{ even}} r\binom{N}{r}$ is equal to the number of pairs (x, S)

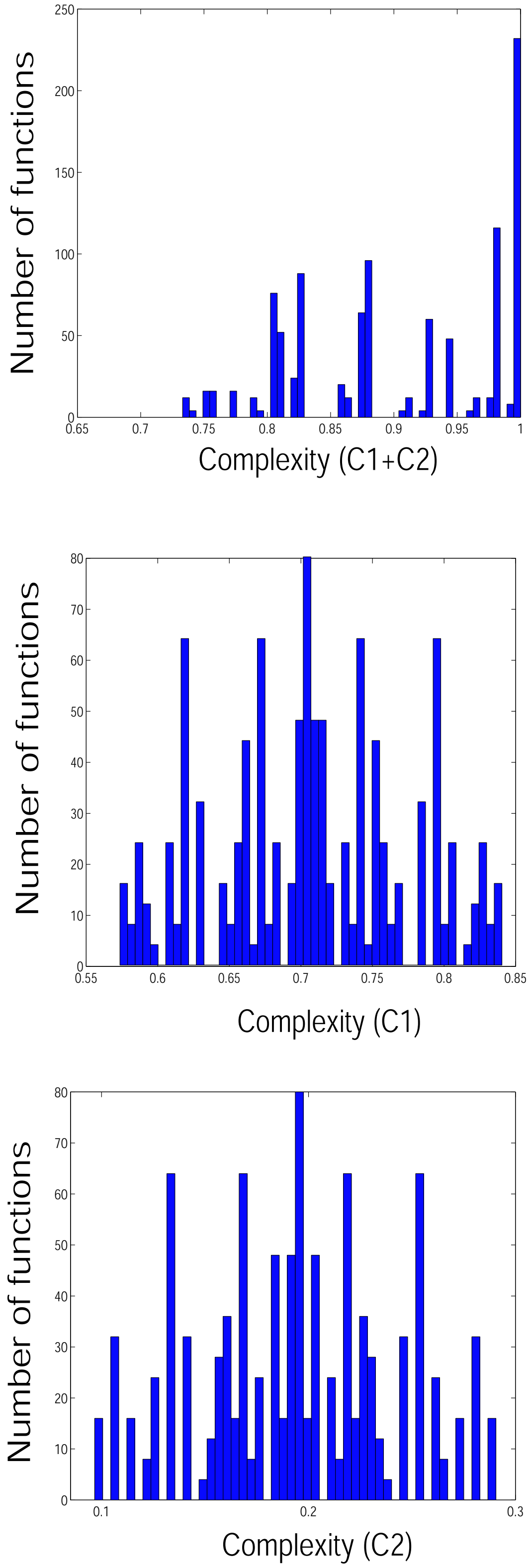

Figure 6: Number of functions compatible with the values of the middle layer being 11111, and having given C_1, C_2 and $C_{12} = C1 + C2$ complexities (top, middle, bottom histograms, respectively).

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.

7 Discussion and conclusions

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 and second order terms of the complexity measure and demonstrated the importance of the second term in obtaining accurate comparisons with generalization error. Furthermore, we indicated that the second-order complexity term provides an estimate of the randomness existing in the output of a Boolean function. This is, we believe, an important feature, since measuring randomness is a very complicated and delicate matter (Kolmogorov, 1965 ; Chaitin, 1975), and also because it is important to quantify randomness in applications such as cryptography (Bernasconi, 2000). An important issue that we have not investigated in detail concerns the appropriate choice of the constant α_2 that weights the effect of the second-order term in relation to the first-order term. We have usually simply taken $\alpha_2 = 1$ (and in one calculation, we assumed $\alpha_2 \geq 1/2$). It may be useful to use both the C_1 and C_2 terms not in linear combination, but as independent quantities that help to characterize the complexity of a Boolean function.

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).

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). In most cases, these approximations compared well (for $N = 14$) with computationally calculated actual values of the complexities.

As a whole, the results presented in this paper show that the complexity measure introduced in (Franco, 2002) 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 emergence of generalization in neural networks. 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.

Acknowledgements

Fruitful discussions with Dr. S.A. Cannas are gratefully acknowledged. L.F. acknowledge support from MRC IRC Oxford UK.

References

Anthony, M., Brightwell, G. and Shawe-Taylor, J. (1995). On specifying Boolean functions by labelled examples. *Discrete Applied Mathematics*, *61*, 1-25.

Anthony, M. (2001). *Discrete Mathematics of Neural Networks: Selected Topics.* SIAM Monographs, Philadelphia.

Barahona,F. (1982) On the computational complexity of ising spin glasses. J.Phys A: Math. Gen., 15: 3241-3253.

Ben-Or, M. and N. Linial, N. (1989). Collective coin flipping. In *Randomness and Computation* (S. Micali ed.) Academic Press, New York, pp. 91-115.

Bernasconi, A., Damm, C. and Shparlinski, I. (2000). The average sensitivity of square-freeness. *Computational Complexity*, *9*, 39-51.

Boppana, R.B. (1997). The Average Sensitivity of Bounded-Depth Circuits. *Information Processing Letters*, *63*, 257-261.

Bruck, J. and Smolensky, R. (1992). Polynomial threshold functions, AC^0 functions, and spectral norms. *SIAM Journal on Computing*, 21 (1), 33-42.

Chaitin, G. (1975). Randomness and Mathematical Proof. *Scientific American. 232*, 47-52.

Chandra, P., Coleman, P. and Ritchey, I. (1993). The Anisotropic Kagome Antiferromagnet: A Topical Spin Glass? *Journal de Physique I, 3*, 591-610.

Cotofana, S. D. and Vassiliadis, S. (1998). Periodic Symmetric Functions, Serial Addition, and Multiplication with Neural Networks. *IEEE Transactions on Neural Networks*, *9*, 1118-1128.

Feldman, J. (2000). Minimization of Boolean complexity in human concept learning. *Nature*, *407*, 572-573.

Franco, L. and Cannas, S.A. (2000). Generalization and Selection of Examples in feedforward Neural Networks. *Neural Computation*, *12*, 2405-2426.

Franco, L. and Cannas, S.A. (2001). Generalization properties of modular networks implementing the parity function. *IEEE Transactions in Neural Networks*, *12*, 1306-1313.

Franco, L. (2002). A measure for the complexity of Boolean functions related to their implementation in neural networks. Submitted. Available from: http://arxiv.org/pdf/cond-mat/0111169

Franco, L. and Cannas, S.A. (2003). Non glassy ground-state in a long-range antiferromagnetic frustrated model in the hypercubic cell. Physica A. In press.
Available on-line from http://dx.doi.org/10.1016/j.physa.2003.10.011

Furst, M. L., Jackson, J. C., and Smith, S. W. (1991). Improved Learning of AC^0 Functions. In *Proceedings of Fourth Annual Workshop on Computational Learning Theory*, 317-325.

Gál, A. and Rosén, A. (1999). A theorem on sensitivity and applications in private computation. In: *Proceedings of the 31st ACM Symposium on the Theory of Computing* , 348-357.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. Macmillan.

Istrail, S. (2000) Statistical Mechanics, Three-dimensionality and NP-completeness I. Universality of intractability for the partition function of the Ising model Across non-planar lattices. STOC 2000 Portland Oregon USA.

Kahn, J., Kalai, G. and Linial, N. (1988). The influence of variables on Boolean functions. In: *Proceedings 29th IEEE Symposium on the Foundations of Computer Science (FOCS' 88)*, 68-80.

Kolmogorov, A. (1965). Three Approaches to the Quantitative Definition of Information. *Problems of Information Transmission. 1*, 1-17.

Linial, N., Mansour, Y. and Nisan, N. (1993). Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, *40*, 607-620.

MacWilliams, F.J and Sloan, N.J.A. (1977). *The Theory of Error Correcting Codes*. North-Holland.

Martin, O.C., Monasson, R. and Zecchina, R. (2001). Statistical mechanics methods and phase transitions in optimization problems. Theoretical computer science, 265, pp.3-67.

Mertens, S. (2002). Computational Complexity for physicists. Computing in Science and Engineering, 31-47.

Saks, M. (1993). Slicing the hypercube. In *Surveys in Combinatorics (Invited Talks of the 1993 British Combinatorial Conference)*, Cambridge University Press, 211-255.

Siu, K.Y., Roychowdhury, V.P. and Kailath, T. (1991). Depth-Size Tradeoffs for Neural Computation. *IEEE Transactions on Computers, 40*, 1402-1412.

Torvik, V.I. and Triantaphyllou, E. (2002). Minimizing the Average Query Complexity of Learning Monotone Boolean Functions. *INFORMS Journal on Computing*, *14*, 142-172.

Valiant, L.G. (1984). A Theory of Learnable. *Communications of the ACM*, *27*, 1134-1142 .

Wegener, I. (1987). *The complexity of Boolean functions*. Wiley, New York.

Zhu, H and Kinzel, W. (1998). Antipredictable sequences: harder to predict than random sequences. *Neural Computation*, *10*, 2219-2230.