Harmonic Analysis of Neural Networks

Jehoshua Bruck
IBM Almaden Research Center, K54/802
650 Harry Road
San Jose, CA 95120-6099

Abstract

Neural networks models have attracted a lot of interest in recent years mainly because there were perceived as a new idea for computing. These models can be described as a network in which every node computes a linear threshold function. One of the main difficulties in analyzing the properties of these networks is the fact that they consist of nonlinear elements. I will present a novel approach, based on harmonic analysis of Boolean functions, to analyze neural networks. In particular I will show how this technique can be applied to answer the following two fundamental questions (i) what is the computational power of a polynomial threshold element with respect to linear threshold elements? (ii) Is it possible to get exponentially many spurious memories when we use the outer-product method for programming the Hopfield model?

1 Introduction

The main purpose of this paper is to introduce a useful tool for the analysis of discrete neural networks in which every node is a Boolean threshold gate. The difficulty in the analysis of neural networks arises from the fact that the basic processing elements (linear threshold gates) are nonlinear. The key idea in harmonic analysis of threshold functions is to represent the functions as polynomials over the field of real numbers. Answering different questions regarding neural networks becomes equivalent to answering questions related to the coefficients of this polynomials.

We will introduce the basic concepts of harmonic analysis of Boolean functions and mention two applications to neural networks. The first application is related to feedforward networks—we prove that a two layer feedforward network of linear threshold gates can compute strictly more than a a single polynomial threshold gate [1]. The second application is

23ACSSC-12/89/0142 $1.00 © 1989 MAPLE PRESS

related to the Hopfield model—we prove that the outer-product method for programming the Hopfield model can result in many spurious stable states(exponential in the number of vectors that we want to store) [3]. Also, we describe applications of harmonic analysis for proving that a given function is not linear threshold and for getting upper bounds on the number of polynomial threshold functions.

The paper is organized as follows, in the next section we present the representation theorem. In Section 3 we present a necessary and sufficient condition for a function to be polynomial threshold and its applications. In Section 4 we describe the application for getting lower bounds on the number of spurious memories in the Hopfield model.

2 The Representation Theorem

In this section the representation of Boolean functions as polynomials over the field of rational numbers is presented. For more detailes see [7] it contains an excellent presentation of the subject.

Throughout the paper a boolean function of f of n variables is a mapping, $f : \{1,-1\}^n \mapsto \{1,-1\}$. Note that we use the multiplicative representation of $\{0,1\}$ via the transformation $a \mapsto (-1)^a$.

Definition: Given a Boolean function f of order n, p is a polynomial (with coefficients over the field of rational numbers) *equivalent* to f iff for all $X \in \{1,-1\}^n$: $f(X) = p(X)$.

As an example, let $f = x_1 \oplus x_2$; that is, f is the XOR function of two variables. It is easy to check that in the $\{1,-1\}$ representation $p(x_1, x_2) = x_1 x_2$. Notice that for every Boolean function f, the polynomial p is linear in each of its variables because $x^2 = 1$ for $x \in \{-1, 1\}$. It is known that every Boolean function has a unique representation as a polynomial [7]. This representation is derived by using the Hadamard matrix, as described by Theorem 1 below. Let's start by defining Hadamard matrices,

Definition: A *Hadamard matrix* of order m, to be denoted by H_m, is an $m \times m$ matrix of +1's and -1's such that

$$H_m H_m^T = m I_m \tag{1}$$

where I_m is the $m \times m$ identity matrix.

Hadamard matrices of order 2^k exist for all $k \geq 0$. The so called Sylvester construction is as follows [8]:

$$\begin{aligned} H_2 &= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ H_{2^{n+1}} &= \begin{bmatrix} H_{2^n} & H_{2^n} \\ H_{2^n} & -H_{2^n} \end{bmatrix} \end{aligned} \tag{2}$$

Theorem 1 *Let f be a Boolean function of order n. Let p be a polynomial equivalent to f. Let A_{2^n} denote the vector of coefficients of p. Let P_{2^n} denote the vector of the 2^n values of p (and f). Then:*

1. *The polynomial p always exists and is unique.*
2. *The coefficients of p are computed as follows,*

$$A_{2^n} = \frac{1}{2^n} H_{2^n} P_{2^n}.$$

Proof: The proof is constructive. The idea is to compute A_{2^n} by solving a system of linear equations. □

Example: Consider the function $f(x_1, x_2) = x_1 \wedge x_2$. Then

$$f(x_1, x_2) = \frac{1}{2}(1 + x_1 + x_2 - x_1 x_2).$$

Notation: The entries of the vector A are denoted by $\{a_\alpha \mid \alpha \in \{0,1\}^n\}$ and are called the spectral representation of a function. Note that a_α is the coefficient of X^α in the polynomial representation where $X^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$. Hence, every Boolean function can be written as:

$$f(X) = \sum_{\alpha \in \{0,1\}^n} a_\alpha X^\alpha.$$

3 Necessary and Sufficient Conditions

The results in this section are based on [1], where more details can be found. We use the polynomial representation of Boolean functions presented in the previous section to derive a necessary and sufficient condition for a function to be an S-threshold function, for arbitrary S. A function $f(X)$ is an S-threshold, for a given set $S \subset \{0,1\}^n$, iff there exist weights such that $f(X) = sgn(\sum_{\alpha \in S} w_\alpha X^\alpha)$.

Theorem 2 *Fix $S \subseteq \{0,1\}^n$. Let $F(X) = \sum_{\alpha \in S} w_\alpha X^\alpha$. Let $f(X)$, $X \in \{-1,1\}^n$, be a Boolean function with spectral representation $\{a_\alpha \mid \alpha \in \{0,1\}^n\}$. Then:*

$$f(X) = sgn(F(X)) \qquad \forall X \in \{1,-1\}^n$$

iff

$$\sum_{X \in \{1,-1\}^n} \mid F(X) \mid = 2^n \sum_{\alpha \in S} w_\alpha a_\alpha. \qquad (3)$$

Theorem 2 is interesting because it suggests that an S-threshold function is fully characterized by the set of spectral coefficients that correspond to S. To see the power of the condition we mention three applications.

1. Using the necessary and sufficient condition we can prove that the PARITY function (outputs 1 iff the number of -1's in X is even) can not be computed as a sign of a polynomial which does not include the term $x_1 x_2 \cdots x_n$. This is a generalization of the result that PARITY is not a linear threshold function.

2. We can also obtain upper bounds on the number of threshold functions. The result is that for a fixed set of m monomials the number of different threshold functions (the weights are arbitrary) is at most 2^{mn}. Again, this is a generalization of a known result for linear threshold function for which that upper bound is 2^{n^2}.

3. The necessary and sufficient condition that is derived above is used to derive lower

bounds on the number of monomials in a threshold function, again, by using the spectral representation.

Let m be the number of monomials in an S-threshold function $f(X)$. Then it turns out that

$$m \geq \hat{a}^{-1}.$$

Where,

$$\hat{a} = \max_{\alpha \in S} | a_\alpha | .$$

The lower bound can can be used to prove that a 2-layer feedforward network of linear threshold elements can compute strictly more than a polynomial threshold function. Assuming that the number of gates in the network and the number of monomials in the polynomial threshold function is bounded by a polynomial in the number of variables.

4 Spurious Memories in the Hopfield Model

The results in this section are based on [3], where more details can be found. One of the most important properties of the Hopfield model is the fact that when it operates in a serial mode it will always get to a stable state (provided W is a symmetric matrix with nonnegative diagonal); see [2,4,6] for more details on convergence properties. This property suggests the use of the model as an associative memory device. An associative memory is a device which "memorizes" a set M of distinct n-bit vectors. It gets as an input an n-bit vector and its output is a vector which belongs to M and is the closest (e.g. in Hamming distance) to the input vector. The idea is that a network can implement an associative memory with the vectors in M being stable states and the association done by convergence to the closest stable state.

One of the interesting issues concerning the use of the network as an associative memory is: how should one *program the network?*

Programming of a network can be defined as follows: Consider the set $M = \{V_1, V_2, \ldots, V_s\}$ that consists of s vectors over $\{1, -1\}^n$. Construct a network such that the set M is a subset of the set of stable states of the network. Hopfield [6] suggested computing W by the *outer-product* method (which is a Hebb-type of rule [5]). Namely,

$$W = \sum_{i=1}^{s} (V_i V_i^T - I_n)$$

where I_n is the $n \times n$ identity matrix. Using this method, T is chosen to be the all-zero vector. Note that if the V_i's are orthogonal then, for all $1 \leq i \leq s$,

$$W V_i = (n - s) V_i.$$

So if $n > s$ every one of the V_i's is stored. Hence, a natural question is: are there any other (spurious) stable states? Namely, what can be said about the number of stable states that are not in M?

Using the harmonic analysis approach it is proved that in certain cases the number of spurious memories can be exponentially (in s) big. Our results hold for the three following cases which cover all the possibilities for s:

1. s is "small" : $1 \leq s \leq \log n$.
2. s is "big" : $n - \log n \leq s < n$.
3. The intermediate cases: $s = 2^k$, where $0 \leq k < \log n$.

The results are the first constructive evidence for the results in [9,10] where such a phenomenon was suggested based on probabilistic arguments.

References

[1] J. Bruck. Harmonic analysis of polynomial threshold functions. *SIAM Journal on Discrete Mathematics*, February 1990. To Appear.

[2] J. Bruck and J. W. Goodman. A generalized convergence theorem for neural networks. *IEEE Transactions on Information Theory*, 34:1089–1092, September 1988.

[3] J. Bruck and V. Roychowdhury. On the number of spurious memories in the hopfield model. *IEEE Transactions on Information Theory*, 1989. To appear.

[4] E. Goles, F. Fogelman, and D. Pellegrin. Decreasing energy functions as a tool for studying threshold networks. *Discrete Applied Mathematics*, 12:261–277, 1985.

[5] D. O. Hebb. *The Organization of Behavior.* Joh Wiley & Sons, New York, 1949.

[6] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences USA*, 79:2554–2558, 1982.

[7] R. J. Lechner. Harmonic analysis of switching functions. In A. Mukhopadhyay, editor, *Recent Development in Switching Theory*, Academic Press, 1971.

[8] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland, New York, 1977.

[9] R. J. McEliece and E. C. Posner. The number of stable points of an infinite-range spin glass. 1985. Unpublished Manuscript.

[10] F. Tanaka and S. F. Edwards. Analytic theory of ground state properties of a spin glass: I. ising spin glass. *J. Phys. F: Metal Phys.*, 10:2769–2778, 1980.