# Neural Networks with Quadratic VC Dimension

**Pascal Koiran***
Lab. de l'Informatique du Parallélisme
Ecole Normale Supérieure de Lyon – CNRS
69364 Lyon Cedex 07, France

**Eduardo D. Sontag**†
Department of Mathematics
Rutgers University
New Brunswick, NJ 08903, USA

## Abstract

This paper shows that neural networks which use continuous activation functions have VC dimension at least as large as the square of the number of weights $w$. This result settles a long-standing open question, namely whether the well-known $O(w \log w)$ bound, known for hard-threshold nets, also held for more general sigmoidal nets. Implications for the number of samples needed for valid generalization are discussed.

## 1 Introduction

One of the main applications of artificial neural networks is to pattern classification tasks. A set of labeled training samples is provided, and a network must be obtained which is then expected to correctly classify previously unseen inputs. In this context, a central problem is to estimate the amount of training data needed to guarantee satisfactory learning performance. To study this question, it is necessary to first formalize the notion of learning from examples.

One such formalization is based on the paradigm of *probably approximately correct (PAC)* learning, due to Valiant (1984). In this framework, one starts by fitting some function $f$, chosen from a predetermined class $\mathcal{F}$, to the given training data. The class $\mathcal{F}$ is often called the "hypothesis class", and for purposes of this discussion it will be assumed that the functions in $\mathcal{F}$ take binary values $\{0,1\}$ and are defined on a common domain $X$. (In neural networks applications, typically $\mathcal{F}$ corresponds to the set of all neural networks with a given architecture and choice of activation functions. The elements of $X$ are the inputs, possibly multidimensional.) The training data consists of labeled samples $(x_i, \varepsilon_i)$, with each $x_i \in X$ and each $\varepsilon_i \in \{0,1\}$, and

*koiran@lip.ens-lyon.fr.
†sontag@hilbert.rutgers.edu.

"fitting" by an $f$ means that $f(x_i) = \varepsilon_i$ for each $i$. Given a new example $x$, one uses $f(x)$ as a guess of the "correct" classification of $x$. Assuming that both training inputs and future inputs are picked according to the same probability distribution on $X$, one needs that the space of possible inputs be well-sampled by the training data, so that $f$ is an accurate fit. We omit the details of the formalization of PAC learning, since there are excellent references available, both in textbook (e.g. Anthony and Biggs (1992), Natarajan (1991)) and survey paper (e.g. Maass (1994)) form, and the concept is by now very well-known.

After the work of Vapnik (1982) in statistics and of Blumer et. al. (1989) in computational learning theory, one knows that a certain combinatorial quantity, called the *Vapnik-Chervonenkis (VC) dimension* $\mathrm{VC}(\mathcal{F})$ of the class $\mathcal{F}$ of interest completely characterizes the sample sizes needed for learnability in the PAC sense. (The appropriate definitions are reviewed below. In Valiant's formulation one is also interested in quantifying the computational effort required to actually fit a function to the given training data, but we are ignoring that aspect in the current paper.) Very roughly speaking, the number of samples needed in order to learn reliably is proportional to $\mathrm{VC}(\mathcal{F})$. Estimating $\mathrm{VC}(\mathcal{F})$ then becomes a central concern. Thus from now on, we speak exclusively of VC dimension, instead of the original PAC learning problem.

The work of Cover (1988) and Baum and Haussler (1989) dealt with the computation of $\mathrm{VC}(\mathcal{F})$ when the class $\mathcal{F}$ consists of networks built up from hard-threshold activations and having $w$ weights; they showed that $\mathrm{VC}(\mathcal{F}) = O(w \log w)$. (Conversely, Maass (1993) showed that there is also a lower bound of this form.) It would appear that this definitely settled the VC dimension (and hence also the sample size) question.

However, the above estimate assumes an architecture based on hard-threshold ("Heaviside") neurons. In contrast, the usually employed gradient descent learning algorithms ("backpropagation" method) rely upon *continuous* activations, that is, neurons with graded responses. As pointed out in Sontag (1989), the use of analog activations, which allow the passing of rich (not just binary) information among levels, may result in higher memory capacity as compared with threshold nets. This has serious potential implications in learning, essentially because more memory capacity means that a given function $f$ may be able to "memorize" in a "rote" fashion too much data, and less generalization is therefore possible. Indeed, Sontag (1992) showed that there are conceivable (though not very practical) neural architectures with extremely high VC dimensions. Thus the problem of studying $\mathrm{VC}(\mathcal{F})$ for analog networks is an interesting and relevant issue. Two important contributions in this direction were the papers by Maass (1993) and by Goldberg and Jerrum (1995), which showed upper bounds on the VC dimension of networks that use piecewise polynomial activations. The last reference, in particular, established for that case an upper bound of $O(w^2)$, where, as before, $w$ is the number of weights. However it was an open problem (specifically, "open problem number 7" in the recent survey by Maass (1993) if there is a matching $w^2$ lower bound for such networks, and more generally for arbitrary continuous-activation nets. It could have been the case that the upper bound $O(w^2)$ is merely an artifact of the method of proof in Goldberg and Jerrum (1995), and that reliable learning with continuous-activation networks is still possible with far smaller sample sizes, proportional to $O(w \log w)$. But this is not the case, and in this paper we answer Maass' open question in the affirmative.

Assume given an activation $\sigma$ which has different limits at $\pm\infty$, and is such that there is at least one point where it has a derivative and the derivative is nonzero (this last condition rules out the Heaviside activation). Then there are architectures with arbitrary large numbers of weights $w$ and VC dimension proportional

to $w^2$. The proof relies on first showing that networks consisting of two types of activations, Heavisides and linear, already have this power. This is a somewhat surprising result, since purely linear networks result in VC dimension proportional to $w$, and purely threshold nets have, as per the results quoted above, VC dimension bounded by $w \log w$. Our construction was originally motivated by a related one, given in Goldberg and Jerrum (1995), which showed that real-number programs (in the Blum-Shub-Smale (1989) model of computation) with running time $T$ have VC dimension $\Omega(T^2)$. The desired result on continuous activations is then obtained, approximating Heaviside gates by $\sigma$-nets with large weights and approximating linear gates by $\sigma$-nets with small weights. This result applies in particular to the standard sigmoid $1/(1+e^{-x})$. (However, in contrast with the piecewise-polynomial case, there is still in that case a large gap between our $\Omega(w^2)$ lower bound and the $O(w^4)$ upper bound which was recently established in Karpinski and Macintyre (1995).) A number of variations, dealing with Boolean inputs, or weakening the assumptions on $\sigma$, are discussed. The full version of this paper also includes some remarks on thresholds networks with a constant number of linear gates, and threshold-only nets with "shared" weights.

## Basic Terminology and Definitions

Formally, a (first-order, feedforward) *architecture* or *network* $\mathcal{A}$ is a connected directed acyclic graph together with an assignment of a function to a subset of its nodes. The nodes are of two types: those of fan-in zero are called *input nodes* and the remaining ones are called *computation nodes* or *gates*. An *output node* is a node of fan-out zero. To each gate $g$ there is associated a function $\sigma_g : \mathbb{R} \to \mathbb{R}$, called the *activation* or *gate function* associated to $g$.

The *number of weights* or *parameters* associated to a gate $g$ is the integer $n_g$ equal to the fan-in of $g$ plus one. (This definition is motivated by the fact that each input to the gate will be multiplied by a weight, and the results are added together with a "bias" constant term, seen as one more weight; see below.) The *(total) number of weights* (or parameters) of $\mathcal{A}$ is by definition the sum of the numbers $n_g$, over all the gates $g$ of $\mathcal{A}$. The *number of inputs* $m$ of $\mathcal{A}$ is the total number of input nodes (one also says that "$\mathcal{A}$ has inputs in $\mathbb{R}^m$"); it is assumed that $m > 0$. The *number of outputs* $p$ of $\mathcal{A}$ is the number of output nodes (unless otherwise mentioned, we assume by default that all nets considered have one-dimensional outputs, that is, $p = 1$).

Two examples of gate functions that are of particular interest are the identity or *linear* gate: $\mathrm{Id}(x) = x$ for all $x$, and the *threshold* or *Heaviside* function: $H(x) = 1$ if $x \geq 0$, $H(x) = 0$ if $x < 0$.

Let $\mathcal{A}$ be an architecture. Assume that nodes of $\mathcal{A}$ have been linearly ordered as $\pi_1, \ldots, \pi_m, g_1, \ldots, g_l$, where the $\pi_j$'s are the input nodes and the $g_j$'s the gates. For simplicity, write $n_i := n_{g_i}$, for each $i = 1, \ldots, l$. Note that the total number of parameters is $n = \sum_{i=1}^{l} n_i$ and the fan-in of each $g_i$ is $n_i - 1$. To each architecture $\mathcal{A}$ (strictly speaking, an architecture together with such an ordering of nodes) we associate a function

$$F : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^p,$$

where $p$ is the number of outputs of $\mathcal{A}$, defined by first assigning an "output" to each node, recursively on the distance from the the input nodes. Assume given an input $x \in \mathbb{R}^m$ and a vector of weights $w \in \mathbb{R}^n$. We partition $w$ into blocks $(w_1, \ldots, w_l)$ of sizes $n_1, \ldots, n_l$ respectively. First the coordinates of $x$ are assigned as the outputs of the input nodes $\pi_1, \ldots, \pi_m$ respectively. For each of the other gates $g_i$, we proceed as follows. Assume that outputs $y_1, \ldots, y_{n_i-1}$ have already

been assigned to the predecessor nodes of $g_i$ (these are input and/or computation nodes, listed consistently with the order fixed in advance). Then the output of $g_i$ is by definition

$$\sigma_{g_i}\left(w_{i,0} + w_{i,1}y_1 + w_{i,2}y_2 + \ldots + w_{i,n_i-1}y_{n_i-1}\right) ,$$

where we are writing $w_i = (w_{i,0}, w_{i,1}, w_{i,2}, \ldots, w_{i,n_i-1})$. The value of $F(x,w)$ is then by definition the vector (scalar if $p = 1$) obtained by listing the outputs of the output nodes (in the agreed-upon fixed ordering of nodes). We call $F$ the *function computed by the architecture* $\mathcal{A}$. For each choice of weights $w \in \mathbb{R}^n$, there is a function $F_w : \mathbb{R}^m \to \mathbb{R}^p$ defined by $F_w(x) := F(x,w)$; by abuse of terminology we sometimes call this also the function computed by $\mathcal{A}$ (if the weight vector has been fixed).

Assume that $\mathcal{A}$ is an architecture with inputs in $\mathbb{R}^m$ and scalar outputs, and that the (unique) output gate has range $\{0,1\}$. A subset $A \subseteq \mathbb{R}^m$ is said to be *shattered* by $\mathcal{A}$ if for each Boolean function $\beta : A \to \{0,1\}$ there is some weight $w \in \mathbb{R}^n$ so that $F_w(x) = \beta(x)$ for all $x \in A$. The *Vapnik-Chervonenkis (VC) dimension* of $\mathcal{A}$ is the maximal size of a subset $A \subseteq \mathbb{R}^m$ that is shattered by $\mathcal{A}$. If the output gate can take non-binary values, we implicitly assume that the result of the computation is the sign of the output. That is, when we say that a subset $A \subseteq \mathbb{R}^m$ is shattered by $\mathcal{A}$, we really mean that $A$ is shattered by the architecture $H(\mathcal{A})$ in which the output of $\mathcal{A}$ is fed to a sign gate.

## 2 Networks Made up of Linear and Threshold Gates

**Proposition 1** *For every $n \geq 1$, there is a network architecture $\mathcal{A}$ with inputs in $\mathbb{R}^2$ and $O(\sqrt{N})$ weights that can shatter a set of size $N = n^2$. This architecture is made only of linear and threshold gates.*

*Proof.* Our architecture has $n$ parameters $W_1, \ldots, W_n$; each of them is an element of $T = \{0.w_1 \ldots w_n ; w_i \in \{0,1\}\}$. The shattered set will be $S = [n]^2 = \{1, \ldots, n\}^2$.

For a given choice of $W = (W_1, \ldots, W_n)$, $\mathcal{A}$ will compute the boolean function $f_W : S \to \{0,1\}$ defined as follows: $f_W(x,y)$ is equal to the $x$-th bit of $W_y$. Clearly, for any boolean function $f$ on $S$, there exists a (unique) $W$ such that $f = f_W$.

We first consider the obvious architecture which computes the function:

$$f^1_W(y) = W_1 + \sum_{z=2}^{n} (W_z - W_{z-1}) H(y - z + 1/2) \tag{1}$$

sending each point $y \in [n]$ to $W_y$. This architecture has $n-1$ threshold gates, $3(n-1)+1$ weights, and just one linear gate.

Next we define a second multi-output net which maps $w \in T$ to its binary representation $f^2(w) = (w_1, \ldots, w_n)$. Assume by induction that we have a net $\mathcal{N}^2_i$ that maps $w$ to $(w_1, \ldots, w_i, 0.w_{i+1} \ldots w_n)$. Since $w_{i+1} = H(0.w_{i+1} \ldots w_n - 1/2)$ and $0.w_{i+2} \ldots w_n = 2 \times 0.w_{i+1} \ldots w_n - w_{i+1}$, $\mathcal{N}^2_{i+1}$ can be obtained by adding one threshold gate and one linear gate to $\mathcal{N}^2_i$ (as well as 4 weights). It follows that $\mathcal{N}^2_n$ has $n$ threshold gates, $n$ linear gates and $4n$ weights.

Finally, we define a net $\mathcal{N}^3$ which takes as input $x \in [n]$ and $w = (w_1, \ldots, w_n) \in \{0,1\}^n$, and outputs $w_x$. We would like this network to be as follows:

$$f^3(x,w) = w_1 + \sum_{z=2}^{n} w_z H(x - z + 1/2) - \sum_{z=2}^{n} w_{z-1} H(x - z + 1/2).$$

This is not quite possible, because the products between the $w_i$'s (which are inputs in this context) and the Heavisides are not allowed. However, since we are dealing with binary variables one can write $uv = H(u + v - 1.5)$. Thus $\mathcal{N}_3$ has one linear gate, $4(n-1)$ threshold gates and $12(n-1)+n$ weights. Note that $f_W(x,y) = f^3(x, f^2(f^1_W(y))$. This can be realized by means of a net that has $n+2$ linear gates, $(n-1)+n+4(n-1) = 6n-5$ threshold gates, and $(3n-2)+4n+(12n-11) = 19n-13$ weights. □

The following is the main result of this section:

**Theorem 1** *For every $n \geq 1$, there is a network architecture $\mathcal{A}$ with inputs in $\mathbb{R}$ and $O(\sqrt{N})$ weights that can shatter a set of size $N = n^2$. This architecture is made only of linear and threshold gates.*

*Proof.* The shattered set will be $S = \{0, 1, \ldots, n^2 - 1\}$. For every $x \in S$, there are unique integers $x, y \in \{0, 1, \ldots, n-1\}$ such that $u = nx + y$. The idea of the construction is to compute $x$ and $y$, and then feed $(x+1, y+1)$ to the network constructed in Proposition 1. Note that $x$ is the unique integer such that $u - nx \in \{0, 1, \ldots, n-1\}$. It can therefore by computed by brute force search as follows:

$$x = \sum_{k=0}^{n-1} kH[H(u - nk) + H(n - 1 - (u - nk)) - 1.5].$$

This network has $3n$ threshold gates, one linear gate and $8n$ weights. Then of course $y = u - nx$. □

A Boolean version is as follows.

**Theorem 2** *For every $d \geq 1$, there is a network architecture $\mathcal{A}$ with $O(\sqrt{N})$ weights that can shatter the $N = 2^{2d}$ points of $\{0,1\}^{2d}$. This architecture is made only of linear and threshold gates.*

*Proof.* Given $u \in \{0,1\}^{2d}$, one can compute $x = 1 + \sum_{i=1}^{d} 2^{i-1} u_i$ and $y = 1 + \sum_{i=1}^{d} 2^{i-1} u_{i+d}$ with two linear gates. Then $(x, y)$ can be fed to the network of Proposition 1 (with $n = 2^d$). □

In other words, there is a network architecture with $2^d$ weights that can compute all boolean functions on $2d$ variables.

## 3 Arbitrary Sigmoids

We now extend the preceding VC dimension bounds to networks that use just one activation function $\sigma$ (instead of both linear and threshold gates). All that is required is that the gate function have a sigmoidal shape and satisfy a very weak smoothness property:

1. $\sigma$ is differentiable at some point $x_0$ (i.e., $\sigma(x_0+h) = \sigma(x_0)+\sigma'(x_0)h+o(h)$) where $\sigma'(x_0) \neq 0$.
2. $\lim_{x \to -\infty} \sigma(x) = 0$ and $\lim_{x \to +\infty} \sigma(x) = 1$ (the limits 0 and 1 can be replaced by any distinct numbers).

A function satisfying these two conditions will be called *sigmoidal.* Given any such $\sigma$, we will show that networks using only $\sigma$ gates provide quadratic VC dimension.

**Theorem 3** *Let $\sigma$ be an arbitrary sigmoidal function. There exist architectures $\mathcal{A}_1$ and $\mathcal{A}_2$ with $O(\sqrt{N})$ weights made only of $\sigma$ gates such that:*

- *$\mathcal{A}_1$ can shatter a subset of $\mathbb{R}$ of cardinality $N = n^2$;*
- *$\mathcal{A}_2$ can shatter the $N = 2^{2d}$ points of $\{0,1\}^{2d}$.*

This follows directly from Theorems 1 and 2, together with the following simulation result:

**Theorem 4** *Let $\sigma$ be a an arbitrary sigmoidal function. Let $\mathcal{N}$ be a network of $T$ threshold and $L$ linear gates, with a threshold gate at the output. Then $\mathcal{N}$ can be simulated on any given finite set of inputs by a network $\mathcal{N}'$ of $T + L$ gates that all use the activation function $\sigma$ (except the output gate which is still a threshold). Moreover, if $\mathcal{N}$ has $n$ weights then $\mathcal{N}'$ has $O(n)$ weights.*

*Proof.* Let $S$ be a finite set of inputs. We can assume, by changing the thresholds of threshold gates if necessary, that the net input $I_g(x)$ to any threshold gate $g$ of $\mathcal{N}$ is different from 0 for all inputs $x \in S$.

Given $\epsilon > 0$, let $\mathcal{N}_\epsilon$ be the net obtained by replacing the output functions of all gates by the new output function $x \mapsto \sigma(x/\epsilon)$ if this output function is the sign function, and by $x \mapsto \sigma_\epsilon(x) = [\sigma(x_0+\epsilon x)-\sigma(x_0)]/[\epsilon\sigma'(x_0)]$ if it is the identity function. Note that for any $a > 0$, $\lim_{\epsilon\to 0+} \sigma(x/\epsilon) = H(x)$ uniformly for $x \in ]-\infty,-a] \cup [a,+\infty]$ and $\lim_{\epsilon\to 0} \sigma_\epsilon(x) = x$ uniformly for $x \in [-1/a, 1/a]$.

This implies by induction on the depth of $g$ that for any gate $g$ of $\mathcal{N}$ and any input $x \in S$, the net input $I_{g,\epsilon}(x)$ to $g$ in the transformed net $\mathcal{N}_\epsilon$ satisfies $\lim_{\epsilon\to 0} I_{g,\epsilon}(x) = I_g(x)$ (here, we use the fact that the output function of every $g$ is continuous at $I_g(x)$). In particular, by taking $g$ to be the output gate of $\mathcal{N}$, we see that $\mathcal{N}$ and $\mathcal{N}_\epsilon$ compute the same function on $S$ if $\epsilon$ is small enough. Such a net $\mathcal{N}_\epsilon$ can be transformed into an equivalent net $\mathcal{N}'$ that uses only $\sigma$ as gate function by a simple transformation of its weights and thresholds. The number of weights remains the same, except at most for a constant term that must be added to each net input to a gate; thus if $\mathcal{N}$ has $n$ weights, $\mathcal{N}'$ has at most $2n$ weights. □

## 4 More General Gate Functions

The objective of this section is to establish results similar to Theorem 3, but for even more arbitrary gate functions, in particular weakening the assumption that limits exist at infinity. The main result is, roughly, that any $\sigma$ which is piecewise twice (continuously) differentiable gives at least quadratic VC dimension, save for certain exceptional cases involving functions that are almost everywhere linear.

A function $\sigma : \mathbb{R} \to \mathbb{R}$ is said to be *piecewise $C^2$* if there is a finite sequence $a_1 < a_2 < \ldots < a_p$ such that on each interval $I$ of the form $]-\infty, a_1[$, $]a_i, a_{i+1}[$ or $]a_p, +\infty[$, $\sigma_{|I}$ is $C^2$.

(Note: our results hold even if it is only assumed that the second derivative exists in each of the above intervals; we do not use the continuity of these second derivatives.)

**Theorem 5** *Let $\sigma$ be a piecewise $C^2$ function. For every $n \geq 1$, there exists an architecture made of $\sigma$-gates, and with $O(n)$ weights, that can shatter a subset of $\mathbb{R}^2$ of cardinality $n^2$, except perhaps in the following cases:*

1. *$\sigma$ is piecewise-constant, and in this case the VC dimension of any architecture of $n$ weights is $O(n \log n)$;*

2. *$\sigma$ is affine, and in this case the VC dimension of any architecture of $n$ weights is at most $n$.*

3. *there are constants $a \neq 0$ and $b$ such that $\sigma(x) = ax + b$ except at a finite nonempty set of points. In this case, the VC dimension of any architecture of $n$ weights is $O(n^2)$, and there are architectures of VC dimension $\Omega(n \log n)$.*

Due to the lack of space, the proof cannot be included in this paper. Note that the upper bound of the first special case is tight for threshold nets, and that of the second special case is tight for linear functions in $\mathbb{R}^n$.

## Acknowledgements

Pascal Koiran was supported by an INRIA fellowship, DIMACS, and the International Computer Science Institute. Eduardo Sontag was supported in part by US Air Force Grant AFOSR-94-0293.

## References

M. ANTHONY AND N.L. BIGGS (1992) *Computational Learning Theory: An Introduction*, Cambridge U. Press.

E.B. BAUM AND D. HAUSSLER (1989) *What size net gives valid generalization?*, Neural Computation 1, pp. 151-160.

L. BLUM, M. SHUB AND S. SMALE (1989) *On the theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines*, Bulletin of the AMS 21, pp. 1–46.

A. BLUMER, A. EHRENFEUCHT, D. HAUSSLER, AND M. WARMUTH (1989) *Learnability and the Vapnik-Chervonenkis dimension*, J. of the ACM 36, pp. 929-965.

T.M. COVER (1988) *Capacity problems for linear machines*, in: Pattern Recognition, L. Kanal ed., Thompson Book Co., pp. 283-289.

P. GOLDBERG AND M. JERRUM (1995) *Bounding the Vapnik-Chervonenkis dimension of concept classes parametrized by real numbers*, Machine Learning 18, pp. 131-148.

M. KARPINSKI AND A. MACINTYRE (1995) *Polynomial bounds for VC dimension of sigmoidal neural networks*, in Proc. 27th ACM Symposium on Theory of Computing, pp. 200-208.

W. MAASS (1993) *Bounds for the computational power and learning complexity of analog neural nets*, in Proc. of the 25th ACM Symp. Theory of Computing, pp. 335-344.

W. MAASS (1994) *Perspectives of current research about the complexity of learning in neural nets*, in *Theoretical Advances in Neural Computation and Learning*, V.P. Roychowdhury, K.Y. Siu, and A. Orlitsky, editors, Kluwer, Boston, pp. 295-336.

B.K. NATARAJAN (1991) *Machine Learning : A Theoretical Approach*, M. Kaufmann Publishers, San Mateo, CA.

E.D. SONTAG (1989) *Sigmoids distinguish better than Heavisides*, Neural Computation 1, pp. 470-472.

E.D. SONTAG (1992) *Feedforward nets for interpolation and classification*, J. Comp. Syst. Sci 45, pp. 20-48.

L.G. VALIANT (1984) *A theory of the learnable*, Comm. of the ACM 27, pp. 1134–1142

V.N. VAPNIK (1982) *Estimation of Dependencies Based on Empirical Data*, Springer, Berlin.