VC Dimension in Circuit Complexity

Pascal Koiran
LIP, Ecole Normale Supérieure de Lyon – CNRS
46 allée d'Italie, 69364 Lyon Cedex 07
France
email: koiran@lip.ens-lyon.fr

February 8, 1996

Abstract

The main result of this paper is a $\Omega(n^{1/4})$ lower bound on the size of a sigmoidal circuit computing a specific AC_2^0 function. This is the first lower bound for the computation model of sigmoidal circuits with unbounded weights. We also give upper and lower bounds for the same function in a few other computation models: circuits of AND/OR gates, threshold circuits, and circuits of piecewise-rational gates.

1 Introduction

One of the main trends in circuit complexity has been to establish lower bounds for circuits made of increasingly powerful gates. Threshold circuits have been quite popular. Here we study generalizations of this computation model. A gate with p inputs $x_1, \ldots, x_p$ outputs $y = \phi(\sum_{i=1}^{p} w_i x_i - \theta)$ where $\phi : \mathbb{R} \to \mathbb{R}$ is a fixed real function. The numerical parameters $w_1, \ldots, w_p$ and θ are called *weights* (θ is also often called a *threshold*). When ϕ is the sign function H, the threshold circuit model is recovered. Of particular interest is the so-called standard sigmoid $\sigma(x) = 1/(1+e^{-x})$. This computation model has been studied due to its frequent usage in empirical machine learning work, and perhaps also because it gives rise to mathematically challenging problems.

Up to now, it had been possible to prove lower bounds in this model only by assuming that an upper bound on the magnitude of the weights is given[1] [3, 4, 10]; one also had to assume a certain separation $\epsilon > 0$ between network outputs for accepted and rejected inputs. In this paper, we give the first lower bound which does not rely on any weight bound or separation. This progress was made possible by a recent breakthrough regarding the VC dimension of sigmoidal circuits [7]. It is to the author's knowledge the first application of VC dimension to circuit complexity. Our $\Omega(n^{1/4})$ lower bound on the size of sigmoidal circuits holds for a certain function in AC_2^0 which we call SELECTION. This function can be computed by a circuit of about n AND/OR gates, which is optimal. The same upper bound holds for sigmoidal gates since they are more powerful than AND/OR gates. In fact, we show that one can do better: SELECTION can be computed by a circuit of $O(\sqrt{n})$ sigmoidal gates. The same upper bound holds for circuits using the so-called linear-bounded sigmoid, and this is optimal up to a constant factor.

[1]Here we give direct lower bounds. These papers contain only indirect bounds, in the sense that they compare the computational power of sigmoidal circuits and other models, such as threshold circuits, splines, and rational functions.

2 Technical preliminaries

We shall just recall the definition of the Vapnik-Chervonenkis dimension. A few references on this topic can be found in [7].

Definition 1 *Let $\mathcal{F}$ be a class of $\{0,1\}$-valued functions on a domain X. We say that $\mathcal{F}$ shatters a set $A \subseteq X$ if for every function $f : A \to \{0,1\}$ there exists $g \in \mathcal{F}$ such that $f = g_{|A}$. The* VC dimension *of $\mathcal{F}$ is the cardinality of the largest set that is shattered by $\mathcal{F}$.*

Here is a more formal definition of our computation model.

Definition 2 *A network (or circuit) is a directed acyclic graph with only one vertex of fan-out 0 (the output gate). Let n be the number of vertices of fan-in*

0 (the input gates, labeled by input variables). The network computes a function $F : \{0,1\}^n \to \mathbb{R}$ which is defined in the usual inductive way. Namely, each vertex (or gate*) g of fan-in $p > 0$ outputs $\phi(\sum_{i=1}^{p} w_i x_i - \theta)$. Here $\phi : \mathbb{R} \to \mathbb{R}$ is a real function and the x_i's are either input variables or outputs of preceding gates linked to g, enumerated in some fixed order.*

It is often the case that the output function ϕ is the same for all gates. Since we are interested in computing boolean functions, the output of a sigmoidal network has to be converted to a binary value. We use the following convention: an output $y \geq 1/2$ is converted to 1, and it is converted to 0 if $y < 1/2$.

In Definition 2, weights and thresholds have fixed numerical values. One could also treat some of them as *programmable parameters*. This gives rise to a *network architecture*. For each setting of the programmable parameters, we have a different sigmoidal network. One can thus associate a family of boolean functions to a network architecture. By VC dimension of an architecture, we mean the VC dimension of the corresponding family of functions.

The lower bound proof method is quite general. Given a function $f : \{0,1\}^n \to \{0,1\}$, we fix a subset $I \subset \{1, \ldots, n\}$ of indices. For notational simplicity, let us assume that I is of the form $\{p+1, p+2, \ldots, n\}$. Consider the following family of restrictions:

$$\mathcal{F} = \{f_v : \{0,1\}^p \to \{0,1\}; f_v(u) = f(u,v)\}$$

where $v \in \{0,1\}^{n-p}$. The method goes as follows:

1. From the definition of f, compute a lower bound on the VC dimension of $\mathcal{F}$.
2. Assume that f can be computed by a "small" network C. Then $\mathcal{F}$ can be computed by a "small" network architecture. Show that $\mathcal{F}$ must then have "small" VC dimension. If C is too small, this will be in contradiction with the lower bound established at step 1.

This method will be applied to the SELECTION function, which is defined as follows. When n is a power of 2, the restriction SELECTION_n of SELECTION to inputs of length $n + \log n$ maps an input (x, y) to $\text{SELECTION}_n(x,y) = y_x$. Here, $x \in \{0,1\}^{\log n}$ and $y \in \{0,1\}^n$. So x is interpreted as an integer, and we just select bit number x of y. In the following, we consider without loss of generality only inputs of this form. SELECTION can be extended to $\{0,1\}^*$ by padding with dummy inputs for other lengths.

We shall use the following family of restrictions:

$$\mathcal{F}_n = \{f_y : \{0,1\}^{\log n} \to \{0,1\}; f_y(x) = f(x,y)\} \quad (1)$$

where $y \in \{0,1\}^n$. It is clear that $\mathcal{F}_n$ shatters $\{0,1\}^{\log n}$: an arbitrary function $f : \{0,1\}^{\log n} \to \{0,1\}$ can be implemented by f_y where $y = f(0)f(1)f(2)\ldots f(n-1)$. Hence $\mathcal{F}_n$ has VC dimension n.

SELECTION is in AC_2^0 (i.e., can be computed by a depth-2 boolean circuit of polynomial size) since

$$\text{SELECTION}_n(x,y) = \bigvee_{i=0}^{n-1} [y_i \wedge (x = i)]$$

and

$$(x = i) \equiv \bigwedge_{j=1}^{\log n} (x_j = i_j) \quad (2)$$

(here i_j is the j-th bit in the binary representation of i.) The corresponding circuit is made of one OR gate and n AND gates. SELECTION_n can be computed by a threshold circuit of $n+1$ gates since an OR or an AND gate can be simulated by a threshold gate. It is a folk theorem that a threshold circuit can be simulated by a sigmoidal circuit of the same size. (the weights have to be multiplied by a large constant, and some thresholds must be shifted; see e.g. [8] for more details.) Hence SELECTION_n can be computed by a circuit of $n+1$ sigmoidal gates. It will be shown in Theorem 7 that one can do better: SELECTION_n can be computed by a sigmoidal circuit of size $O(\sqrt{n})$.

3 Upper and lower bounds

Our main result is as follows.

Theorem 1 *There is a lower bound of $\Omega(n^{1/4})$ on the size (number of gates) of any sigmoidal network computing* SELECTION_n.

This is a straightforward consequence of the following more general result.

Theorem 2 *Assume that SELECTION_n can be computed by a sigmoidal network $\mathcal{N}$ with M gates, m of which are linked to one of the last n inputs. Then $m^2 M^2 \geq Cn$ for some universal constant C.*

Proof. For any $y \in \{0,1\}^n$, the function f_y defined in (1) can be implemented by a circuit which is obtained from $\mathcal{N}$ by changing the thresholds of the m gates linked to the last n inputs, and removing the corresponding links. Therefore all functions in $\mathcal{F}$ can be computed by a fixed architecture of M gates with m programmable parameters. By [7], the VC dimension of such an architecture is $O(m^2 M^2)$. We are done since $\mathcal{F}_n$ has VC dimension n. □

Theorem 1 follows from the obvious observation that $m \leq M$. In many cases, m is significantly smaller than M. For instance, if one works with layered networks, m will be bounded by the number of gates in the first hidden layer.

The same technique can be used to give lower bounds for networks that use an arbitrary Pfaffian function instead of σ, or apply a polynomial function to their inputs instead of taking a linear combination. The suitable VC dimension bounds are in [7]. Interestingly, this technique also makes it possible to prove the optimality for circuits of AND/OR gates of the construction showing that SELECTION $\in \mathrm{AC}_2^0$ (as is customary in circuit complexity, we can assume that there are no negations in the circuit by pushing them down to the input level).

Theorem 3 *Let $s(n)$ be the minimum number of AND and OR gates needed to compute* SELECTION$_n$*. Then $n \leq s(n) \leq n+1$.*

Proof. We have already seen that $s(n) \leq n+1$. In order to show that $s(n) \geq n$, consider a circuit of AND/OR gates where m gates are linked to at least one of the y inputs. Assign some fixed values to $y_1, \ldots, y_n$ and let g be one of the m gates, for instance an OR gate. Gate g computes a function of the form

$$(\bigvee_{i=1}^{k} u_i) \vee (\bigvee_{i=1}^{l} v_i)$$

where u_i is a (possibly negated) x input or the output of another gate, and v_i is a (possibly negated) y input. Depending on the values of $y_1, \ldots, y_n$, the second term is the constant 0 or the constant 1. Hence gate g can compute only two different functions. The same holds for an AND gate. This implies that the cardinality of $\mathcal{F}_n$ is upper-bounded by 2^m. Thus $m \geq n$ since $|\mathcal{F}_n| = 2^n$. $\square$

We do not have such a tight bound for threshold circuits.

Theorem 4 *Let $s(n)$ be the minimum number of threshold gates needed to compute* SELECTION$_n$*. Then $C.n/\log n \leq s(n) \leq n+1$ for some universal constant C.*

Proof. The upper bound $s(n) \leq n+1$ was discussed in section 3. The lower bound follows from the fact that for a circuit of M threshold gates the VC dimension of $\mathcal{F}_n$ is $O(M \log M)$ [1]. $\square$

We now study more powerful computation models.

Theorem 5 *Let $\phi : \mathbb{R} \to \mathbb{R}$ be some fixed piecewise-rational function. There is a $\Omega(\sqrt{n})$ lower bound on the size of a ϕ-network computing* SELECTION$_n$*.*

Proof. By [5], the VC dimension of $\mathcal{F}$ for a circuit of M ϕ-gates is $O(M^2)$. $\square$

This lower bound can actually be achieved with a very simple activation function: the linear-bounded sigmoid π. This function is defined as follows: $\pi(x) = 0$ for $x \leq 0$, $\pi(x) = x$ for $0 \leq x \leq 1$ and $\pi(x) = 1$ for $x \geq 1$. Threshold gates can be simulated by π-gates with "large" weights, just as for sigmoidal gates. The linear part of π also makes it possible to simulate linear gates (i.e., gates where the output function is the identity). These two properties are used in the proof of the next theorem.

Theorem 6 SELECTION$_n$ *can be computed by a network of $O(\sqrt{n})$ π-units.*

Proof. Let us first assume for simplicity that n is of the form 2^{2k}, and let $p = \sqrt{n} = 2^k$. The construction is in several stages. In the first stage, we encode the n 'y' bits into p groups of p bits. This can be done by p units in the first hidden layer computing

$$W_0 = \pi\left(\sum_{i=1}^{p} 2^{-i} y_{i-1}\right), \ldots, W_{p-1} = \pi\left(\sum_{i=1}^{p} 2^{-i} y_{n-p+i-1}\right).$$

Let $u, v \in \{0, \ldots, p-1\}$ be the unique integers such that $x = pu + v$. We need to compute bit number v of W_u. Such a construction is described in [8], with the difference that the W_i's were programmable parameters in that paper. This implies a difference in the construction of the second stage, where we select W_u. To the unit computing W_i we associate another unit outputting $b_i = 1$ if $u = i$ (i.e., if W_i should be selected) and $b_i = 0$ otherwise. Note that u is simply given by the k most significant bits of x. The test $u = i$ can thus be performed as in (2) with a single AND gate, and therefore also by a single π gate. The selection of W_u can now be implemented as follows:

$$W_u = \pi\left[\sum_{i=0}^{p-1} \pi(W_i - b_i)\right]. \qquad (3)$$

In the two final stages, we need to decode the p digits of W_u and output bit number v. This can be done almost as in [8] with $O(p)$ units. We describe that construction here for the sake of completeness. First, we define a multi-output network which maps W_u to its binary representation $f(W_u) = (W_{u1}, \ldots, W_{up})$.

Assume by induction that we have a net $\mathcal{N}_i$ that maps W_u to $(W_{u1}, \ldots, W_{ui}, 0.W_{u,i+1} \ldots W_{up})$. Since

$$\begin{aligned} W_{u,i+1} &= H(0.W_{u,i+1} \ldots W_{up} - 1/2) \\ &= \pi\left[2^{p-i}\left(x - \left(\frac{1}{2} - 2^{-(p-i)}\right)\right)\right] \end{aligned}$$

and

$$0.W_{u,i+2} \ldots W_p = \pi(2 \times 0.W_{u,i+1} \ldots W_{un} - W_{u,i+1}),$$

$\mathcal{N}_{i+1}$ can be obtained by adding two gates to $\mathcal{N}_i$ (as well as 4 weights)[2]. It follows that $\mathcal{N}_p$ has $2p$ gates and $4p$ weights. Finally, we define a net $\mathcal{N}'$ which takes as input v, $f(W_u)$ and outputs W_{uv}. We would like this network to be as follows:

$$\begin{aligned} f'(v, W_u) &= \pi\left[W_{u1} + \sum_{z=2}^{p} W_{uz}\pi(2v - 2z + 1)\right. \\ &\qquad \left. - \sum_{z=2}^{p} W_{u,z-1}\pi(2v - 2z + 1)\right] \end{aligned}$$

(v can be computed as a linear combination of k input bits). This is not quite possible, because the products between the W_{uz}'s and the π-gates are not allowed. However, since we are dealing with binary variables one can write $\alpha\beta = \pi(2\alpha + 2\beta - 1)$. Thus $\mathcal{N}'$ has $4p - 3$ gates and $12(p-1) + p$ weights.

In the case where n is of the form 2^{2k+1} (recall that n is always assumed to be a power of two) we break the n 'y' bits into 2^k groups of length 2^{k+1}. The rest of the construction is essentially unchanged. □

The same $O(\sqrt{n})$ upper bound applies to the standard sigmoid. We first need the following intermediate result.

Lemma 1 SELECTION$_n$ *can be computed by a network of $O(\sqrt{n})$ linear, threshold, and product units.*

Proof. This follows from a simple modification of the construction of Theorem 5. In that construction, threshold or linear units can be used instead of π-units everywhere except in (3). However, the selection stage is trivial to implement with product units: $W_u = \sum_{i=0}^{p-1} b_i W_i$. □

The following result applies in particular to the standard sigmoid.

Theorem 7 *Let $\phi : \mathbb{R} \to \mathbb{R}$ be such that $\lim_{x \to -\infty} \phi(x) = 0$, $\lim_{x \to +\infty} \phi(x) = 1$ and $\phi''(a)$ exists and is non-zero for some $a \in \mathbb{R}$. Then* SELECTION$_n$ *can be computed by a network of $O(\sqrt{n})$ ϕ-units.*

[2] with the usual convention $H(x) = 1$ for $x \geq 0$ and $H(x) = 0$ otherwise.

Proof. This follows from Lemma 1 and the fact that for any finite set of inputs, a network of linear, threshold and product gates can be simulated by a network of ϕ-gates of the same size, up to a constant factor [8]. □

4 Final Remarks

The $\Omega(n^{1/4})$ lower bound for sigmoidal circuits is still rather small. It would be very interesting to improve it, possibly by choosing another target function than SELECTION. The ultimate goal could be to prove an exponential lower bound for constant-depth sigmoidal networks, as was done for threshold nets [6]. (note however that to the author's knowledge, such bounds are known to hold only for threshold nets with "small" weights.) Proving such superlinear lower bounds would require other tools than the VC dimension since, as a rule of thumb, the VC dimension of an architecture is at least linear in the number of programmable parameters. It is probably safe to say that purely combinatorial arguments will not suffice: deep algebraic and geometric tools, such as those used in the proof of the Karpinski-Macintyre bound [7], will have to play an important role.

It would also be interesting to close the gap between the upper and the lower bound in Theorem 4. The optimal upper bound in Theorem 6 was obtained from a modification of the construction of piecewise-linear circuits of quadratic VC dimension [8], which is as large as possible. It has also been known that the $O(w \log w)$ upper bound on the VC dimension of threshold net is optimal [9]. However, it seems harder to obtain a $O(n/\log n)$ upper bound for SELECTION$_n$ from the corresponding construction.

References

[1] E.B. Baum and D. Haussler. What size net gives valide generalization ? *Neural Computation*, 1:151–160, 1989.

[2] B. DasGupta and G. Schnitger. The power of approximating: a comparison of activation functions. In *Advances in Neural Information Processing Systems 5*, pages 615–622. Morgan Kaufmann, 1993.

[3] B. DasGupta and G. Schnitger. Analog versus discrete neural networks. preprint, 1995. extended abstract in [2].

[4] B. DasGupta and G. Schnitger. Efficient approximation with neural networks: a comparison of gate functions. preprint, 1995. extended abstract in [2].

[5] P. Goldberg and M. Jerrum. Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, 18:131–148, 1995.

[6] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turán. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 6:129–154, 1993.

[7] M. Karpinski and A. Macintyre. Polynomial bounds for VC dimension of sigmoidal neural networks. In *Proc. 27th ACM Symposium on Theory of Computing*, pages 200–208, 1995.

[8] P. Koiran and E. D. Sontag. Neural networks with quadratic VC dimension. In *Advances in Neural Information Processing Systems 8*. Morgan Kaufmann, 1996.

[9] W. Maass. Neural nets with superlinear VC dimension. *Neural Computation*, 6:877–884, 1994. extended abstract in *Proc. 25th ACM Symposium on Theory of Computing, pages 335-344, 1993.*

[10] W. Maass, G. Schnitger, and E. D. Sontag. On the computational power of sigmoid versus boolean threshold circuits. In *Proc. 32nd IEEE Symposium on Foundations of Computer Science*, 1991.