The Perceptron Strikes Back*

preliminary report

Richard Beigel† Yale University

Nick Reingold† Yale University

Daniel Spielman‡ Yale College

Abstract

We show that every AC^0 predicate is computed by a low-degree probabilistic polynomial over the reals. We show that circuits composed of a symmetric gate at the root with AND-OR subcircuits of constant depth can be simulated by probabilistic depth-2 circuits with essentially the same symmetric gate at the root and AND gates of small fanin at the bottom. In particular, every language recognized by a depth-d AC^0 circuit is decidable by a probabilistic perceptron of size $2^{O\left(\log^{4d} n\right)}$ and order $O\left(\log^{4d} n\right)$ that uses $O\left(\log^{3} n\right)$ probabilistic bits. As a corollary, we present a new proof that depth-d AND-OR circuits computing the parity of n binary inputs require size $2^{n^{\Omega(1/d)}}$.

1. AC^0

AC^0 is the class of predicates computed by polynomial-size circuits consisting of unbounded fanin AND and OR gates, and having depth bounded by some constant. It has been the subject of much recent research.

Razborov [13] and Smolensky [14] showed that every AC^0 predicate can be approximated by a low-degree polynomial over any finite field. Their upper bounds yield circuit lower bounds related to Hastad's [10]. Applying Hastad's lower bound for parity, Linial, Mansour, and Nisan [11] proved an important upper bound: every AC^0 predicate can be approximated by a low-degree polynomial over the reals. Their upper bound implies an efficient learning algorithm for AC^0 predicates, under the uniform distribution. An open problem is to efficiently learn AC^0 predicates under an arbitrary distribution.

We define probabilistic polynomials, and show that every AC^0 predicate can be computed *exactly* by a low-degree *probabilistic* polynomial over the reals. It is hoped that this is a step towards distribution-free learning of AC^0 predicates.

Our result has an important consequence for neural nets: every AC^0 predicate can be computed by a probabilistic perceptron. This is related to Allender and Hertrampf's [1, 2] and Yao's [19] results for circuits with mod-gates.

In addition, our upper bound for AC^0 can be combined with Smolensky's easy proof that parity cannot be approximated by a low degree polynomial; thus we obtain a very elegant proof that constant-depth circuits that compute parity require exponential size (cf. [8, 18, 10, 7, 14]). The result of Linial, Mansour, and Nisan [11] could be applied in a similar way, but that would be circular, because their work depends on prior lower bounds for parity. Our work does not depend on

---

*The authors may be reached by writing to Department of Computer Science, P.O. Box 2158, Yale Station, New Haven, CT 06520-2158, or by sending electronic mail to lastname-firstname@cs.yale.edu. Similar results were obtained by Tarui. Except for appendix 1, where we simplify his 0-sided error construction, all of our results were obtained independently.

†Supported in part by NSF grants CCR-8808949 and CCR-8958528.

‡Supported in part by NSF grant CCR-8958528 under an REU supplement.

prior lower bounds for parity.

2. Perceptrons

Studied since the 1950s, perceptrons are depth-2 threshold circuits consisting of a single threshold gate at the root with AND gates at the next level. Though perceptrons were proposed as a powerful component of vision systems, Minsky and Papert's monograph [12] showed that they were computationally very weak. Thus debunked, perceptrons all but disappeared from the scientific literature. Subsequently, however, the success of neural nets, which are networks of perceptrons, generated circumstantial evidence in favor of the perceptron. Now individual perceptrons are sometimes used for real computation. In this paper, we show that a single perceptron is powerful enough to recognize every language in AC^0. To be precise, we show that probabilistic perceptrons of size $2^{\mathrm{polylog}\, n}$ and bottom fanin polylog n can simulate AC^0 circuits with small error probability. Thus we vindicate the reputation of the much maligned perceptron.

Highly contrasting results are known for deterministic perceptrons. Minsky and Papert [12] constructed a depth-2 AC^0 predicate that cannot be computed by a deterministic perceptron of order less than $\frac{1}{2}\sqrt{n}$.

Our approach is inspired by Allender [1], who showed that every AC^0 circuit can be simulated by a probabilistic depth-2 circuit with a parity gate at the root and $2^{\mathrm{polylog}\, n}$ AND gates of fanin polylog n on the bottom. Allender and Hertrampf [2] extended this result by reducing the number of random bits used. We also build on techniques of Toda and Ogiwara [16]. Similar results were obtained independently by Tarui [15], who reports that his work was independent of Toda and Ogiwara.

3. Probabilistic Polynomials and Circuits

Definition 1. A *probabilistic circuit* is a circuit with two types of inputs: actual inputs $x_1, \ldots, x_n$ from $\{0,1\}$, and probabilistic inputs $\rho_1, \ldots, \rho_l$ where each ρ_i is chosen uniformly and independently from $\{0,1\}$. If C is a probabilistic circuit, then for any given $x_1, \ldots, x_n$ we think of the output $C(x_1, \ldots, x_n)$ as a random variable. Let C be a probabilistic circuit, and let f be a function of n inputs. *C computes f with error ϵ* if for every $x_1, \ldots, x_n$, the probability that $C(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$ is at least $1 - \epsilon$.

Definition 2. A *probabilistic polynomial* is a polynomial in two types of variables: actual variables $x_1, \ldots, x_n$ from $\{0,1\}$, and probabilistic variables $\rho_1, \ldots, \rho_l$ where each ρ_i is chosen uniformly and independently from $\{0,1\}$. If P is a probabilistic polynomial, then for any given $x_1, \ldots, x_n$ we think of $P(x_1, \ldots, x_n)$ as a random variable. Let P be a probabilistic polynomial, and let f be a function of n variables. *P computes f with error ϵ* if for every $x_1, \ldots, x_n$, the probability that $P(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$ is at least $1 - \epsilon$.

Definition 3. A *perceptron of size s and order m* is a depth-2 circuit with a threshold gate at the top and s AND gates of fanin m on the bottom.

We will make use of the following result due to Valiant and Vazirani [17]:

Theorem 4. *Let $S \subseteq \{0,1\}^k$ be nonempty. Suppose $w_0, w_1, \ldots, w_k$ are randomly chosen from $\{0,1\}^k$. Let $S_0 = S$ and let S_i be*

$$\{v \in S : v \cdot w_0 \equiv v \cdot w_1 \equiv \cdots \equiv v \cdot w_i \equiv 0 \pmod 2\}$$

for each $i \in \{0, \ldots, k\}$, where $\cdot$ denotes inner product. Let $Pr_k(S)$ be the probability that $|S_i| = 1$ for some $i \in \{0, \ldots, k\}$. Then $Pr_k(S) \geq 1/4$.

Lemma 5. *For any $\epsilon > 0$, there exists a probabilistic polynomial of degree $O\left(\log(1/\epsilon)\log^2 n\right)$, with n actual variables and $O\left(\log(1/\epsilon)\log^2 n\right)$ probabilistic variables which computes the OR of the n actual variables with error ϵ.*

Proof: Let $b_0, \ldots, b_{n-1}$ be the actual variables and let $k = \lceil \log n \rceil$. Consider elements of $\{0,1\}^k$ as numbers from 0 to $n-1$ represented in binary. Let $S = \left\{ v \in \{0,1\}^k : b_v = 1 \right\}$, and let the S_i's be as in Theorem 4. The bits of each w_i are taken from distinct sets of the probabilistic bits of size k. Notice that $|S_i|$ can be computed as

$$\sum_{v \in \{0,1\}^k} (b_v \wedge (v \cdot w_0 \equiv v \cdot w_1 \equiv \cdots \cdots \equiv v \cdot w_i \equiv 0 \pmod 2))). \quad (1)$$

Each summand in (1) can be computed as a polynomial in at most $O\left(\log^2 n\right)$ variables, where b_v and each bit in the w_i's are considered separate variables. Thus, for each $i \in \{0, \ldots, k\}$ we can compute $P_i = |S_i|$ as a polynomial with the inputs and the probabilistic bits as its variables. Consider $P = \prod_{i=1}^{k}(1 - P_i)$. P is a probabilistic polynomial with the b_i's as its actual variables and the bits of the w_i's as its probabilistic variables. If every b_i is zero, then every $|S_i| = 0$, so $P(b_0, \ldots, b_{n-1}) = 1$. If at least one b_i is 1, then, by Theorem 4, $P(b_0, \ldots, b_{n-1}) = 0$ with probability at least 1/4.

We can amplify the probability by repeating the test, each time using new probabilistic bits. For each test, we construct a new probabilistic polynomial P which differs only in the probabilistic bits used. Let P' be the probabilistic polynomial obtained by multiplying together $O(\log(1/\epsilon))$ different such P's. Notice that if every b_i is zero, then $P'(b_0, \ldots, b_{n-1}) = 1$, and if at least one b_i is one, then $P'(b_0, \ldots, b_{n-1}) = 0$ with probability at least $1 - \epsilon$. Since we repeat the test $O(\log(1/\epsilon))$ times, we use $O\left(\log(1/\epsilon)\log^2 n\right)$ probabilistic bits in all. ∎

Lemma 6. *For any $\epsilon > 0$ and any function f computed by a depth-d, size-s AND-OR circuit with n inputs, there exists a probabilistic polynomial of degree $O\left(\left(\log(1/\epsilon)\log^2 n \log s\right)^d\right)$ of n actual variables and $O\left(\log(1/\epsilon)\log^2 n\right)$ probabilistic variables that computes f with error ϵ.*

Proof: Change each AND gate in the circuit to the negation of an OR gate of the negations of its inputs. By Lemma 5, for each OR gate in the circuit there is a probabilistic polynomial that computes the value of the gate as a function of its inputs and the probabilistic bits with error ϵ/s. Let $\hat{P}$ be the composition of these polynomials. $\hat{P}$ has degree $O\left(\left(\log(1/\epsilon)\log^2 n \log s\right)^d\right)$ and computes f with error ϵ. ∎

Definition 7. A set of gates G is *discerning* if for any sequence of weights $w_1, \ldots, w_f$ there exists a gate $g \in G$ of fanin f such that for all $x_1, \ldots, x_f$, if $\sum_{i=1}^{f} x_i w_i = 0$ then $g(x_1, \ldots, x_f) = 0$ and if $\sum_{i=1}^{f} x_i w_i = 1$ then $g(x_1, \ldots, x_f) = 1$. For example, the collection of all (weighted) threshold gates is discerning.

Theorem 8. *For any $\epsilon > 0$, any discerning set of gates G, and any function f computed by a depth-d, size-s AND-OR circuit with n inputs, there exists a probabilistic depth-2 circuit of size $2^{O\left(\left(\log(1/\epsilon)\log^2 n \log s\right)^d\right)}$ composed of a gate from G at the root and AND gates of fanin $O\left(\left(\log(1/\epsilon)\log^2 n \log s\right)^d\right)$ that uses $O\left(\log(1/\epsilon)\log^2 n\right)$ probabilistic bits and computes f with error ϵ.*

Proof: By Lemma 6, there exists a probabilistic polynomial that computes f with error ϵ. Let $\hat{P}$ be such a polynomial. Since the variables of the polynomial are idempotent, it can be expanded into $2^{O\left(\left(\log(1/\epsilon)\log^2 n \log s\right)^d\right)}$ monomials of degree $O\left(\left(\log(1/\epsilon)\log^2 n \log s\right)^d\right)$, each of which can be computed by an AND gate of fanin $O\left(\left(\log(1/\epsilon)\log^2 n \log s\right)^d\right)$. Thus, f can be computed with error ϵ by a gate from G with these AND gates as inputs. ∎

Definition 9. A *symmetric gate* is a gate with inputs $x_1, \ldots, x_n$ which computes $h(\sum_{i=1}^{n} x_i)$ for some underlying function h. We say that two symmetric gates are of the same *type* if the two underlying functions are the same.

Theorem 10. *For any $\epsilon > 0$ and any function f computed by a depth-d circuit of n inputs with a symmetric gate g at the root and m size-s AND-OR subcircuits, there exists a probabilistic depth-2 circuit with a symmetric gate of the same type as g at the root and $2^{O\left(\left(\log(m/\epsilon)\log^2 n \log s\right)^d\right)}$ AND gates of fanin $O\left(\left(\log(m/\epsilon)\log^2 n \log s\right)^d\right)$ on the bottom that uses $O\left(\log(m/\epsilon)\log^2 n\right)$ probabilistic bits and computes f with error ϵ.*

Proof: By Lemma 6, for each subcircuit which computes an input to G, there exists a polynomial of degree $O\left(\left(\log(m/\epsilon)\log^2 n \log s\right)^d\right)$ that computes the value of the subcircuit with error ϵ/m. As in Theorem 8, each polynomial can be computed by a weighted sum of AND gates. So, there is a symmetric gate of the same type as g with these AND gates as inputs which computes f with error ϵ. ∎

Corollary 11. *For any function f computed by a depth-d, size $2^{\log^k n}$ AND-OR circuit of n inputs, there exists a probabilistic perceptron of size $2^{O\left(\log^{2d(k+1)} n\right)}$ and order $O\left(\log^{2d(k+1)} n\right)$ that uses $O\left(\log^{k+2} n\right)$ probabilistic bits and computes f with error $1/2^{\log^k n}$.*

Corollary 12. *For any polynomial p, and any function f computed by a depth-d AC^0 circuit with n inputs, there exists a probabilistic perceptron of size $2^{O\left(\log^{4d} n\right)}$ and order $O\left(\log^{4d} n\right)$ that uses $O\left(\log^3 n\right)$ probabilistic bits and computes f with error $1/p(n)$.*

It is a routine matter to verify that all the polynomials constructed in this paper have coefficients of size $2^{\mathrm{polylog}(n/\epsilon)}$; therefore the threshold gates used in the circuits built above have weights of size $2^{\mathrm{polylog}(n/\epsilon)}$ as well.

4. Parity and AC^0: a new proof

There have been many proofs that the parity function is not computed by any AND-OR circuit having small depth and size [18, 4, 7, 10, 14]. We present a new proof based on our result and an important observation due to Smolensky [14].

Suppose that the parity of n binary inputs is computed by an AND-OR circuit having size s and depth d. Then there exists a probabilistic polynomial of degree $(\log s)^{O(d)}$ that computes parity with error probability at most $1/4$. Therefore there exists a polynomial of degree $(\log s))^{O(d)}$ that computes parity on at least $3/4$ of all inputs. Our approximating polynomial has domain $\{0,1\}^n$. By considering the map to be a function into the integers mod 3, and applying the transformation $x \mapsto x-1$ to each variable we can consider our inputs to be from $\{1,-1\}$. Smolensky's idea [14] will complete the proof. Consider a polynomial mapping $\{-1,1\}^n$ to the integers mod 3. If a monomial has degree $\geq n/2$, note that it is equal to $m(x_1,\ldots,x_n)p(x_1,\ldots,x_n)$, where m is the product of the variables that do not appear in the monomial and p computes parity. Thus, every polynomial mapping can be approximated by a polynomial of degree $n/2 + (\log s)^{O(d)}$, with agreement on at least $3/4$ of all inputs. If $(\log s)^d = n^{o(1)}$, then

$$3^{2^n} \leq 3^{\frac{1}{4}2^n} \cdot 3^{(\frac{1}{2}+o(1))2^n},$$

by the law of large numbers, so $1 \leq \frac{1}{4} + \frac{1}{2} + o(1)$, which is false. Therefore $(\log s)^d = n^{\Omega(1)}$, so

$$s = 2^{n^{\Omega(1/d)}}.$$

Instead of using Smolensky's idea, one could use the result from [3] that a small order perceptron cannot even approximate parity.

5. A Suggestion on Terminology

Many papers [8, 9, 1, 5, 19, 6] have considered circuits whose bottom level consists of AND gates or OR gates having fanin polylog n, while gates at other levels have unbounded fanin. Perceptrons have a single threshold gate at the root and polylog-fanin AND gates at the bottom level. Since they have only one threshold gate, it is unnatural for us to think of perceptrons as depth-2 threshold circuits. We prefer to call them depth-1^+ threshold circuits. In general we suggest the notation "depth-d^+" for circuits having d levels of unbounded fanin gates, followed by one level consisting of polylog-fanin AND gates or polylog-fanin OR gates. For example, Yao's result [19] says that every bounded depth ACC circuit can be simulated by a deterministic depth-2^+ threshold circuit of size $2^{\mathrm{polylog}\, n}$.

6. Concluding Remarks

We have shown that every AC^0 predicate is computed by a probabilistic perceptron of order $O\left(\log^{4d} n\right)$. On the other hand, Minsky and Papert [12] have found a depth-2 AC^0 predicate which cannot be computed by any deterministic perceptron of order less than $\frac{1}{2}\sqrt{n}$. Thus, we find that randomization strictly increases the power of perceptrons. The power of randomization has been similarly established in other areas of complexity theory. We wonder whether the use of randomness can simplify the design of more general neural nets.

Acknowledgments. We would like to thank Dana Angluin, Fu Bin, Mike Fischer, Fred Green, Simon Kasif, Noam Nisan, Steven Rudich, and Jun Tarui for helpful discussions.

References

[1] E. Allender. A note on the power of threshold circuits. In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, pages 580–584, 1989.

[2] E. Allender and U. Hertrampf. On the power of uniform families of constant depth threshold circuits. In *Proceedings of the 15th International Symposium on Mathematical Foundations of Computer Science*, pages 158–164, Aug. 1990. Lecture Notes in Computer Science 452.

[3] J. Aspnes, R. Beigel, M. Furst, and S. Rudich. On the expressive power of voting polynomials. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, 1991. To appear.

[4] L. Babai. A random oracle separates PSPACE from the polynomial hierarchy. *Inf. Process. Lett.*, 26:51–53, 1987.

[5] R. Beigel. Polynomial interpolation, threshold circuits, and the polynomial hierarchy. YALEU/DCS/TR 843, Yale University, Dept. of Computer Science, Jan. 1991.

[6] R. Beigel, N. Reingold, and D. Spielman. PP is closed under intersection. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, 1991. To appear.

[7] J. Cai. With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 21–29, 1986.

[8] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, April 1984.

[9] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, and G. Turán. Threshold circuits of bounded depth. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pages 99–110, 1987.

[10] J. Hastad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 6–20, 1986.

[11] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform, and

learnability. In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, pages 574–579, 1989.

[12] M. L. Minsky and S. A. Papert. *Perceptrons.* MIT Press, Cambridge, MA, 1988. Expanded Edition. The first edition appeared in 1968.

[13] A. A. Razborov. Lower bounds for the size of circuits of bounded depth with basis $\{\wedge, \oplus\}$. *Math. notes of the Academy of Science of the USSR*, 41(4):333–338, Sept. 1987.

[14] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 77–82, 1987.

[15] J. Tarui. Randomized polynomials, threshold circuits, and the polynomial hierarchy. In *Proceedings of the 8th Annual Symposium on Theoretical Aspects of Computer Science.* Springer-Verlag, 1991.

[16] S. Toda and M. Ogiwara. Counting classes are at least as hard as the polynomial-time hierarchy. In *Proceedings of the 6th Annual Conference on Structure in Complexity Theory.* IEEE Computer Society Press, 1991. To appear.

[17] L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Comput. Sci.*, 47:85–93, 1986.

[18] A. C. Yao. Separating the polynomial-time hierarchy by oracles. In *Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science*, pages 1–10, 1985.

[19] A. C.-C. Yao. On ACC and threshold circuits. In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science*, pages 619–627, 1990.

7. Appendix 1: Zero-sided error

In his paper [15], Jun Tarui constructs probabilistic polynomials P that either give the correct answer, or else give an answer that does not belong to $\{0,1\}$. We call that zero-sided error. By considering P^2 and $1-(1-P)^2$, he obtains perceptrons that compute with 1-sided error. Tarui was definitely the first to discover that, and the results in this section are not independent of him. However, we have a cleaner construction of polynomials with 0-sided error.

Let $P(x_1,\ldots,x_n)$ be any randomized polynomial that is 0 if $x_1+\ldots+x_n$ is 0, and is 1 with high probability if $x_1+\ldots+x_n>0$. (Such polynomials are given by Valiant and Vazirani.) We construct a polynomial E which is 0 when P computes OR correctly, and non-zero when P computes OR incorrectly. Let

$$E=(P-1)(x_1+\ldots+x_n).$$

Let

$$E^* = \sum_{\text{all OR gates simulated}} E^2 \text{ for that OR gate.}$$

Let F be the probabilistic polynomial produced by our circuit simulation. Then $F-(F^2+1)E^*$ is equal to F when there are no errors, and is negative when there is an error. Thus $F-(F^2+1)E^*$ computes with 0-sided error.