Efficient Approximation with Neural Networks: A Comparison of Gate Functions*

Bhaskar DasGupta† Georg Schnitger‡

Department of Computer Science

The Pennsylvania State University

University Park

PA 16802

Abstract

We compare different gate functions in terms of the approximation power of their circuits. Evaluation criteria are circuit size s, circuit depth d and the approximation error $e(s,d)$. We consider two different error models, namely "extremely tight" approximations (i.e. $e(s,d) = 2^{-s}$) and the "more relaxed" approximations (i.e. $e(s,d) = s^{-d}$). Our goal is to determine those gate functions that are equivalent to the standard sigmoid $\sigma(x) = \frac{1}{1+\exp(-x)}$ under these two error models.

For error $e(s,d) = 2^{-s}$, the class of equivalent gate functions contains, among others, (non-polynomial) rational functions, (non-polynomial) roots and most radial basis functions. Newman's approximation of $|x|$ by rational functions is obtained as a corollary of this equivalence result. Provably not equivalent are polynomials, the *sine*-function and linear splines. For error $e(s,d) = s^{-d}$, the class of equivalent activation functions grows considerably, containing for instance linear splines, polynomials and the *sine*-function. This result only holds if the number of input units is counted when determining circuit size. The standard sigmoid is distinguished in that relatively small weights and thresholds suffice.

Finally, we consider the computation of boolean functions. Now the binary threshold gains considerable power. Nevertheless, we show that the standard sigmoid is capable of computing a certain family of n-bit functions in constant size, whereas circuits composed of binary thresholds require size at least proportional to $\log n$.

1 Introduction.

We consider efficient approximations of a given multivariate function $f : [-1,1]^m \rightarrow \mathcal{R}$ by feedforward neural networks. Let us first introduce the notion of a feedforward net (or in our terminology, the notion of a Γ-circuit).

Let Γ be a class of real-valued functions, where each function is defined on some subset of $\mathcal{R}$. A Γ-circuit C is an unbounded fan-in circuit whose edges and vertices are labeled by real numbers. The real number assigned to an edge (resp. vertex) is called its *weight* (resp. its *threshold*). Moreover, to each vertex v a gate function $\gamma_v \in \Gamma$ is assigned.

*Partially supported by NSF Grant CCR-9114545

†`email: bhaskar@cs.psu.edu`

‡`email: georg@cs.psu.edu`

The circuit C computes a function $f_C : [-1,1]^m \to \mathcal{R}$ as follows. The components of the input vector $x = (x_1, \ldots, x_m) \in [-1,1]^m$ are assigned to the sources of C. Let $v_1, \ldots, v_n$ be the immediate predecessors of a vertex v. The input for v is then $s_v(x) = \sum_{i=1}^n w_i y_i - t_v$, where w_i is the weight of the edge (v_i, v), t_v is the threshold of v and y_i is the value assigned to v_i. If v is not a sink, then we assign the value $\gamma_v(s_v(x))$ to v. Otherwise we assign $s_v(x)$ to v.

We assume from now on that C has a single sink w. Then $f_C = s_w$ is the function computed by C.

A great deal of work has been done showing that circuits of depth 2 can approximate (in various norms) large function classes (including continuous functions) arbitrarily well (Arai, 1989; Carrol and Dickinson, 1989; Cybenko, 1989; Funahashi, 1989; Gallant and White, 1988; Hornik *et al.* 1989; Irie,1988; Lapades and Farber, 1987; Nielson, 1989; Poggio and Girosi, 1989; Wei *et al.*, 1991). Various gate functions have been used, among others, *the cosine squasher, the standard sigmoid, radial basis functions, generalized radial basis functions, polynomials, trigonometric polynomials and binary thresholds.* Still, as we will see, these functions differ greatly in terms of their approximation power.

Research on efficient approximations has only started recently (Williamson and Paice, 1991).

Our goal is to compare gate functions in terms of *efficiency* and *quality of approximation.* We measure efficiency by the *size* of the circuit (i.e. the number of nodes, excluding input units) and by its *depth.* Another resource of interest is *the Lipschitz-bound* of the circuit, which is a measure of the numerical stability of the circuit. We say that circuit C has Lipschitz-bound L if all weights and thresholds of C are bounded in absolute value by L and for each vertex v of C and for all inputs $x, y \in [-1,1]^m$,

$$|\gamma_v(s_v(x)) - \gamma_v(s_v(y))| \leq L \cdot |s_v(x) - s_v(y)|.$$

(Thus we do not demand that gate function γ_v has Lipschitz-bound L, but only that γ_v has Lipschitz-bound L for the inputs it receives.) We measure the quality of an approximation of function f by function f_C by the Chebychev norm; i.e. by the maximum distance between f and f_C.

Let $\gamma : \mathcal{R} \to \mathcal{R}$ be some gate function. The following questions are the topic of our paper.

- Given a function f, how well can we approximate f by a $\{\gamma\}$-circuit of size s, depth d and Lipschitz-bound L? We are particularly interested in the behavior of the approximation error $e(s,d)$ as a function of size and depth. This set-up allows us to investigate how much the approximation error decreases with increased size and/or depth.
- Given two gate functions γ_1 and γ_2, when do $\{\gamma_1\}$-circuits and $\{\gamma_2\}$-circuits have *essentially* the same "approximation power" with respect to some error function $e(s,d)$?

Let us first introduce the notions of *simulation* and *equivalence* with respect to error $e(s,d)$.

Definition 1.1 *Let $e : \mathcal{N}^2 \to \mathcal{R}_+$ be a function. Γ_1 and Γ_2 are classes of gate functions.*
(a). *We say that Γ_1 simulates Γ_2 with respect to e if and only if there is a constant k such that for all functions $f : [-1,1]^m \to \mathcal{R}$ with Lipschitz-bound $1/e(s,d)$,*

> *if f can be $e(s,d)$-approximated by a Γ_2-circuit of size s, depth d and Lipschitz-bound 2^s, then f can also be $e(s,d)$-approximated by a Γ_1-circuit of size $(s+1)^k$, depth $k(d+1)$ and Lipschitz-bound 2^{s^k}.*

(b). *We say that Γ_1 and Γ_2 are equivalent with respect to e if and only if Γ_2 simulates Γ_1 with respect to e and Γ_1 simulates Γ_2 with respect to e.*

In other words, we allow size to increase polynomially and depth to increase by a constant factor, but we insist on at least the same approximation error. Observe that we have linked the approximation error $e(s,d)$ and the Lipschitz-bound of the function to be approximated. The reason is that approximations of functions with high Lipschitz-bound "tend" to have an inversely proportional approximation error. Moreover observe that the Lipschitz-bounds of the involved circuits are allowed to be exponential in the size of the circuit. We will see in section 1.2, that for some gate functions far smaller Lipschitz-bounds suffice.

Below we discuss our results. In section 1.1 we consider the case of tight approximations, i.e. $e(s,d) = 2^{-s}$. Then in section 1.2 the more relaxed error model $e(s,d) = s^{-d}$ is discussed. Section 1.3 deals with the computation of boolean functions. In section 2 we show how to approximately compute polynomials assuming a sufficiently smooth gate function. Proofs are given in the remaining sections.

1.1 Error $e(s,d) = 2^{-s}$.

Theorem 1.1 *Assume that $e(s,d) = 2^{-s}$. Then the following gate functions are equivalent to the standard sigmoid $\sigma(x) = \frac{1}{1+\exp(-x)}$.*

- *any rational function which is not a polynomial,*
- *any root x^α provided α is not a natural number,*
- *the logarithm (for any basis $b > 1$),*
- *$exp(x)$,*

This result is the consequence of identifying a large class of gate functions γ that are capable of simulating the standard sigmoid with respect to error $e(s,d) = 2^{-s}$ (Theorem 3.1). Informally, such a gate function γ has to satisfy two properties. Firstly, γ has a convergent Taylor-series in some small interval; in this case, polynomials can be approximated tightly (Proposition 2.1). Secondly, γ-circuits have to be capable of approximating the binary threshold (over the interval $[-1,1] - [-2^{-s}, 2^s]$) with size $poly(s)$, depth $O(1)$ and error 2^{-s}.

In Definition 3.1, we describe a Lipschitz-type property that guarantees a tight approximation of the binary threshold. This result is quite powerful. For instance, we obtain Newman's approximation of $|\ x\ |$ (resp. of the binary threshold) by rational functions (Newman, 1964) as a corollary (Remark 3.1).

Notable exceptions from the list of functions equivalent to the standard sigmoid are polynomials, trigonometric polynomials and splines (see section 2). We *do* obtain an equivalence to the standard sigmoid by allowing splines of degree s as activation functions for circuits of size s. Linear splines, on the other hand, are properly weaker (Proposition 3.1). The same applies to polynomials, even if we allow any polynomial of degree s as gate function for circuits of size s (Proposition 3.1).

As a consequence of the equivalence of standard sigmoids and splines, we prove that "small and fast" sigmoidal circuits (with weights of size at most 2^s) are incapable of approximating oscillating functions even with modest error. In particular, an approximation error of at most $1/2$ for $sine(Ax)$ requires size $s = A^{\Omega(1/d)}$, where d is the given depth of the circuit (Lemma 3.1). Thus $sine$ and σ are not equivalent with respect to error $e(s,d) = 2^{-s}$.

On the other hand, constant depth and size polynomial in $\log A$ suffice to approximate $sine(Ax)$, provided the input x is given *in binary* (Reif, 1987). Therefore, the complexity of extracting bits from the given analog input is high as well.

1.2 Error $e(s, d) = s^{-d}$.

Now we are facing a far less demanding error requirement (if depth is small). But, as we will see in section 1.3, linear splines and the standard sigmoid are still not equivalent. This changes if we count the number of input units when determining the size of the circuit.

With this modification, the standard sigmoid turns out to be equivalent to linear splines as well as to the *sine*-function. More generally, any suitable gate function (Definition 2.2) is capable of simulating the standard sigmoid with respect to error s^{-d} (Theorem 4.1).

Again, we have to allow a Lipschitz-bound of (at most) 2^s for circuits of size s. **But**, assuming that the standard sigmoid σ simulates γ with respect to error s^{-d}, then σ-circuits with weights and thresholds of absolute value at most s suffice. Observe that γ-circuits are allowed to have weights as large as 2^s.

The equivalence proof involves a first phase of extracting $O(d \log s)$ bits from the analog input. In a second stage, a binary computation is mimicked. The extraction process can be carried out once the binary threshold is approximated with error s^{-1} (over the domain $[-1, 1] - [-1/s, 1/s]$). It is quite surprising that mimicking a binary computation leads to an almost optimal analog computation when approximating, say, $sine(Ax)$ by σ-circuits; this computation is provably optimal if we disregard a polynomial increase in size and a constant factor increase in depth. Finally, this process leads to a characterization of those gate functions that are equivalent to σ with respect to error s^{-d} (Theorem 4.1).

1.3 Computing boolean functions.

The binary threshold gains considerable power when computing boolean functions (as compared to approximating analog functions). *If* the input units are counted when determining circuit size, then the binary threshold is equivalent to the standard sigmoid for the case of constant depth and small weights even if we do not allow depth to increase (Maass *et al.*, 1991). Our simulation results for error $e(s, d) = 2^{-s}$ also imply that the binary threshold and the standard sigmoid are equivalent when computing boolean functions (Remark 3.2); but we do not have to impose any depth restriction and we can allow weights of size up to 2^s. Observe however that the depth of a simulating threshold circuit increases by a constant factor.

As a consequence of the equivalence of binary threshold and standard sigmoid we obtain that, when computing boolean functions, small weights suffice for sigmoidal circuits (if we are willing to increase size polynomially and depth by a constant factor).

But the above equivalence only holds if we count the number of input units when determining the size of the circuit. Sigmoidal circuits will be far more powerful otherwise. This difference in efficiency is demonstrated by the the problem of squaring. In particular, we are interested in the language

$$M_n = \{(x, y) : \ x \in \{0,1\}^n, y \in \{0,1\}^{n^2} \text{ and } [x]^2 \geq [y]\}.$$

($[z]$ denotes the bit sum of the binary string z; i.e. $[z] = \sum_i z_i$). We obtain the following result,

Theorem 1.2 *A threshold circuit computing M_n must have size at least $\Omega(\log n)$. But M_n can be computed by a σ-circuit with constantly many gates.*

This lower bound is "almost" tight, since it is possible to design a threshold circuit of size $O(\log n \cdot \log\log n \cdot \log\log\log n)$ which computes the square of a given sum of bits. Technically, M_n is not sensitive; for instance, if we set all x-bits to 0 and one y-bit to 1, then M_n is trivialized. Also, intermediate forms of sensitivity have to be ruled out: setting half of the x-bits to 0 and setting half of the y bits to 1 again trivializes the language. Therefore, Wegener's lower bound for sensitive functions (Wegener, 1991) does not apply. Our lower bound proof does proceed by trying to unravel a given circuit gate by gate. But we can not trivialize each gate; instead we construct a subdomain which allows to trivialize threshold gates while not trivializing M_n.

The previously best known separation of threshold circuits and sigmoidal circuits is due to Maass, Schnitger and Sontag (Maass *et al.*, 1991). But their result only applies to threshold circuits of depth 2; our result holds without any depth restriction. Theorem 1.2 can be generalized to separate threshold circuits and gate functions which are 3-times differentiable in some small neighborhood, but this smoothness requirement is more severe than the one assumed in (Maass *et al.*, 1991).

2 Polynomials and Splines.

First we briefly discuss splines (Ahlberg *et al.*, 1967).

Definition 2.1
(a) *A* **continuous** *function $f : \mathcal{R} \rightarrow \mathcal{R}$ is called a k-spline of degree δ if and only if there are knots $-\infty = x_0 < x_1 < \ldots < x_k < x_{k+1} = \infty$ and polynomials $p_1, \ldots, p_{k+1}$ of degree at most δ such that $f(x) = p_r(x)$ for all x with $x \in [x_{r-1}, x_r]$.*
(b) *A spline circuit (resp. polynomial circuit) of size s is an unbounded fan-in circuit whose gates compute 1-splines (resp. univariate polynomials)) of degree at most s. The coefficients of all splines involved, as well as the weights and thresholds of the circuit have to be bounded in absolute value by 2^s.*

Observe that a k-spline can be expressed as a spline circuit of depth 2 and size $k+1$. In particular, assume that f is a k-spline with polynomials $p_1, \ldots, p_{k+1}$ and knots $x_1, \ldots, x_k$. Consider the 1-splines $f_1, \ldots, f_k$ and the polynomial $f_{k+1} = p_{k+1}$. Set $f_i(x) = p_i(x) - p_{i+1}(x)$ for $x \leq x_i$ and $f_i(x) = 0$ for $x > x_i$. Then $f = \sum_{i=1}^{k+1} f_i$.

Splines are quite powerful; intuitively they achieve tight approximations whenever the domain can be partitioned into not "too many" intervals where the function in question has a sufficiently fast converging taylor series. This applies for instance to rational functions.

In the next result we show that circuits composed of sufficiently smooth gate functions are capable of approximating polynomials within any degree of accuracy.

Definition 2.2 *Let $\gamma : \mathcal{R} \rightarrow \mathcal{R}$ be a function. We call γ suitable if and only if there exists real numbers α, β ($\alpha > 0$) and an integer k such that*
(a) *γ can be represented by the power series $\sum_{i=0}^{\infty} a_i (x-\beta)^i$ for all $x \in [-\alpha, \alpha]$. The coefficients are rational numbers of the form $a_i = \frac{P_i}{Q_i}$ with $|P_i|, |Q_i| \leq 2^{ki}$ (for $i > 1$).*
(b) *For each $i > 2$ there exists j with $i \leq j \leq i^k$ and $a_j \neq 0$.*

Proposition 2.1 *Assume that γ is suitable with parameter k.*

Then, over the domain $[-D, D]$, any degree n polynomial p can be ε-approximated by a γ-circuit C_p. C_p has depth 2 and size $O(n^{2k})$; its weights are rational numbers whose numerator and denominator are bounded in absolute value by

$$p_{max}(2+D)^{poly(n)} ||\gamma^{(N+1)}||_{[-\alpha,\alpha]} \frac{1}{\varepsilon}.$$

Here we have assumed that the coefficients of p are rational numbers with numerator and denominator bounded in absolute value by p_{max}.

Proof. We assume that $\beta = 0$ and furthermore that $\gamma(x) = \sum_{i=0}^{\infty} a_i x^i$ for all $x \in [-\alpha, \alpha]$. By assumption on γ we will find N such that $a_N \neq 0$ and $n \leq N \leq n^k$. It suffices to show that x^N can be approximately computed since $x^N, (x+1)^N, \ldots, (x+N)^N$ are linearly independent and therefore span $\mathcal{P}_N$, the space of all degree N polynomials. When computing x^n (instead of x^N) circuit size increases by at most a factor of N, while depth remains at 2.

Let $T_N(x) = \sum_{i=1}^{N} a_i x^i$ denote the N'th degree Taylor polynomial of γ. Assume that T_N has s non-zero coefficients, say $a_{k_1}, \ldots, a_{k_s}$. We observe that the monomials $x^{k_1}, \ldots, x^{k_s}$ form a Tchebycheff system over $[0, \infty]$ (page 9 in (Karlin and Studden, 1966)). Hence, the the matrix $({x_i}^{k_j})$ is non-singular for any collection of pairwise distinct, positive reals $x_1, \ldots, x_s$.

Let $0 < u_1 < \ldots < u_s \leq 1$ be arbitrary. Then the matrix $(a_{k_j}(u_i)^{k_j})$ is non-singular and therefore, the polynomials $T_N(u_1 x), \ldots, T_N(u_s x)$ are linearly independent. Consequently, there are $c_1 \ldots, c_s$ such that $\sum_{i=1}^{s} c_i T_N(u_i x) = x^N$. Set $c_{max} = \max\{|c_i| : 1 \leq i \leq s\}$.

Let $L > 1$ be an integer. In order to compute x^N from h (instead of T_N) we compute $\gamma_N(x) = L^N \sum_{i=1}^{s} c_i h(u_i x/L)$. Observe that for $-\alpha \leq x \leq \alpha$, $\gamma_N(x) = x^N + L^N \sum_{i=1}^{s} c_i (h(u_i x/L) - T_N(u_i x/L))$. We claim that γ_N approximates x^N with error at most ε, provided L is suitably chosen.

The error is bounded by $e = N L^N c_{max} (\frac{||\gamma^{(N+1)}||_{[-\alpha,\alpha]}(|x|/L)^{N+1}}{(N+1)!})$. Here we have applied the Lagrange error estimate. Since the coefficients a_i are size-restricted rationals, we obtain $c_{max} \leq 2^{poly(N)}$. Furthermore we can assume that $[-\alpha, \alpha] \subseteq [-D, D]$. Thus

$$e \leq c_{max} ||\gamma^{(N+1)}||_{[-\alpha,\alpha]} |x|^{N+1}/L \leq c_{max} ||\gamma^{(N+1)}||_{[-\alpha,\alpha]} D^{N+1}/L \leq (2+D)^{poly(N)} ||\gamma^{(N+1)}||_{[-\alpha,\alpha]}/L.$$

Therefore, for

$$L = (\frac{D}{\alpha})^N (2+D)^{poly(n)} ||\gamma^{(N+1)}||_{[-\alpha,\alpha]} \frac{1}{\varepsilon} = (2+D)^{poly(n)} ||\gamma^{(N+1)}||_{[-\alpha,\alpha]} \frac{1}{\varepsilon}$$

and $x \in [-\alpha, \alpha]$, we can approximate x^N with error $(\frac{\alpha}{D})^N \cdot \varepsilon$.

Finally, we achieve the desired ε-approximation of x^N over the interval $[-D, D]$ by computing $(\frac{D}{\alpha})^N \gamma_N(\frac{\alpha}{D} x)$.
♠

3 Equivalence of gate functions for error 2^{-s}.

Let us observe that an approximation of individual gate functions establishes an approximation of the entire circuit. In particular, assume that there is k such that each function in Γ_1 can be (2^{-2s^2-s})-approximated by a depth k, size s^k Γ_2-circuit which has a Lipschitz-bound of 2^{s^k}. Let C be an arbitrary Γ_1-circuit of depth d and size s and assume that C has Lipschitz-bound 2^s. If we approximate the gate functions of C individually, then we obtain a circuit D of depth $k \cdot d$, size s^{k+1} and Lipschitz-bound 2^{s^k}. Since a weighted sum can amplify the approximation error by at most a factor of $s \cdot 2^s \leq 2^{2s}$, D will approximate with error bounded by $2^{-2s^2-s} 2^{2sd} \leq 2^{-s}$.

The following classes of gate functions will be able to simulate the standard sigmoid with respect to error 2^{-s}.

Definition 3.1 *Let Γ be a class of gate functions and let $g : [1, \infty] \rightarrow \mathcal{R}$ be a function.*
(a). *We say that g is fast converging if and only if*

$$| g(x) - g(x+\varepsilon) | = O(\varepsilon/x^2) \text{ for } x \geq 1, \varepsilon \geq 0,$$

$$0 < \int_0^\infty g(1+u^2)du < \infty \text{ and } \left| \int_{2^N}^\infty g(u^2)du \right| = O(1/N) \text{ for all } N \geq 1.$$

(b). *We say that Γ is powerful if and only if at least one function in Γ is suitable and there is a fast converging function g which can be approximated for all $s > 1$ (over the domain $[-2^s, 2^s]$) with error 2^{-s} by a $\{\Gamma\}$-circuit of constant depth, size polynomial in s and Lipschitz-bound 2^s.*

The crucial concept of fast convergence can be checked easily for differentiable functions by applying the mean value theorem. Examples of fast converging functions include $x^{-\alpha}$ for $\alpha \geq 1$, $exp(-x)$ and $\sigma(-x)$.

Below we characterize the approximation power of splines. When we say that Γ simulates splines, we mean that if spline-circuits (Definition 2.1) of size s and depth d approximate a function f with approximation error 2^{-s}, then f can also be approximated with error 2^{-s} by a Γ-circuit of size $(s+1)^k$, depth $k(d+1)$ and Lipschitz-bound 2^{s^k}.

Theorem 3.1 *Assume that Γ is powerful.*
(a) *Γ simulates splines with respect to error $e(s,d) = 2^{-s}$.*
(b) *Assume that each gate function in Γ can be approximated (over the domain $[-2^s, 2^s]$) with error 2^{-s} by a spline-circuit N_s of size s and constant depth. Then Γ is equivalent to splines.*

Proof. We only verify part (a). Part (b) is an immediate consequence of part (a).

Assume that Γ-circuits of constant depth and size polynomial in s can compute a function $t : [-1,1] \rightarrow \mathcal{R}$ which approximates the binary threshold (with error at most 2^{-s}) over the domain $[-1,1] - [-2^{-s}, 2^{-s}]$. Moreover, assume that $|t|$ is bounded by a constant (independent of s).

This will establish the claim. Why? Let $\pi(x) = \begin{cases} p(x) & \text{if } x < 0 \\ q(x) & \text{otherwise.} \end{cases}$ As discussed earlier, it suffices to find a Γ-circuit of constant depth and size polynomial in r which approximates π with error at most (2^{-2r^2-r}). An inductive argument shows that spline-circuits of size s and Lipschitz-bound 2^r generate gate input of size at most $L = 2^{O(r^2)}$. Therefore, an approximation of π over the domain $[-L, L]$ is sufficient as well. Observe that Γ-circuits can approximate polynomials and therefore can also approximately multiply (since $x \cdot y = ((x+y)^2 - x^2 - y^2)/2$). Since $|t|$ is bounded over $[-1,1]$, the required approximation is now achieved by

$$p(0) + (q^*(x) - p(0)) \cdot t(x/L) + (p^*(x) - p(0)) \cdot t(-x/L),$$

where p^* and q^* are approximations (with error at most 2^{-r^3}) of p and q respectively. (Observe that $p(0) = q(0)$ since we only allow continuous splines.) Thus the claim will follow for $s = r^3$.

The remaining task is to construct t. Since Γ is powerful, we can approximate a fast converging function g quite tightly. We set $h(x) = g(1+x^2)$. The construction of $t(x)$ (for $x \in [-1,1]$) proceeds as follows.

Let $K = 4$. First x is scaled down by computing $y(x) = x/2^{s^{K/4}}$. Then the sum

$$z(x) = \sum_{i=0}^{s^{3K/2}} (u_{i+1} - u_i) \cdot h(u_i)$$

is determined, where $u_i = y(x)a^i$ and $a = (1+1/s^K)$. Below we show that, for $x > 2^{-s^{K/4}}$, $z(x)$ approximates $\alpha = \int_0^\infty h(u)du$ quite tightly; therefore $z(x)$ approximates $-\alpha$ for $x < -2^{-s^{K/4}}$. This approximation is improved by first computing $u(x) = (\frac{\alpha - z(x)}{2\alpha})^{s^{K/4}}$ and then by computing $t(x) = (1 - u(x))^{s^{K/4}}$.

We claim that $t(x)$ has the desired properties. The crucial point is to verify that $z(x)$ approximates α for positive x. First we observe that h is continuous since g is continuous. Therefore,

$$\int_{u_i}^{u_{i+1}} h(u)du = (u_{i+1} - u_i))h(w), \text{ for some } w \text{ with } u_i \leq w \leq u_{i+1}.$$

Next we utilize that g is fast converging. In particular,

$$\mid h(w) - h(u_i) \mid = O(\frac{u_{i+1}^2 - u_i^2}{(1+u_i^2)^2}) = O(\frac{u_{i+1} - u_i}{1+u_i^2})$$

and consequently

$$\mid \int_{u_i}^{u_{i+1}} h(u)du - (u_{i+1} - u_i)h(u_i) \mid = O(\frac{(u_{i+1} - u_i)^2}{1+u_i^2}).$$

This leads to the estimate

$$\mid z(x) - \int_{u_0}^{\infty} h(u)du \mid = O(\sum_{i=0}^{s^{3K/2}} \frac{(u_{i+1} - u_i)^2}{1+u_i^2}) + \int_{u_{s^{3K/2}+1}}^{\infty} h(u)du.$$

But $u_{i+1} - u_i = u_i(a-1)$ and therefore

$$\sum_{i=0}^{s^{3K/2}} \frac{(u_{i+1} - u_i)^2}{1+u_i^2} = O((a-1)^2 \sum_{i=0}^{s^{3K/2}} 1) = O((a-1)^2 s^{3K/2}) = O(s^{-K/2}).$$

Now, utilizing that $(1+1/m)^m \geq 2$, we obtain for $i = js^K$,

$$u_i = y(x)a^i = y(x)(1+1/s^K)^i \geq y(x)2^j = x2^{j-s^{K/4}}.$$

It suffices to consider x with $x > 2^{-s^{K/4}}$ since we do not claim any approximation result otherwise. In this case, $u_i \geq 2^{j-2s^{K/4}}$, and therefore $u_{s^{3K/2}} + 1 \geq 2^{s^{K/2}/2}$. Since g is fast converging,

$$\mid \int_{u_{s^{3K/2}+1}}^{\infty} h(u)du \mid = O(s^{-K/2}).$$

Finally, let us observe that $u_0 \leq 2^{-s^{K/4}}$ and since h is continuous, $\int_0^{u_0} h(u)du$ can be neglected. Summarizing,

$$\mid z(x) - \int_0^{\infty} h(u)du \mid = O(s^{-K/2}).$$

So far we disregarded that g can only be approximated. But it suffices to achieve an error of 2^{-s^K} over the interval $[-2^{s^K}, 2^{s^K}]$ which is achievable by circuits of size polynomial in s.

Let us consider the final two operations, namely $u(x) = (\frac{\alpha - z(x)}{2\alpha})^{s^{K/4}}$ and $t(x) = (1-u(x))^{s^{K/4}}$. First u maps an approximation of α into a neighborhood of 0 and an approximation of $-\alpha$ into a neighborhood of

1. The powering decreases the distance to 0 "exponentially" and increases the distance to 1 polynomially (by a factor of $O(s^{K/4})$). t on the other hand interchanges 0 and 1. Consequently, the previously poor approximation of 1 is turned into a tight approximation of 0, whereas the previously tight approximation of 0 deteriorates only polynomially. (Here we have used $\exp(yx/(1+x)) \leq (1+x)^y \leq \exp(yx)$ for $x, y \geq 0$ and its corresponding inequality for $x < 0$.) The result will be an approximation error of at most $2^{-s^{K/4}}$. ♠

Remark 3.1 *Obviously, $1/x$ is powerful. Therefore Theorem 3.1 implies that $\{1/x\}$-circuits of constant depth and size s approximate the binary threshold (with error 2^{-s}) over the domain $[-1,1] - [-2^{-s}, 2^{-s}]$ (and the approximation is bounded over $[-1,1]$, independently of s). The degree of the resulting rational function will be polynomial in s. Thus Theorem 3.1 generalizes Newman's approximation of the binary threshold (Newman, 1964) by rational functions.*

In Theorem 1.1 we claimed the equivalence (with respect to error $e(s,d) = 2^{-s}$) of the standard sigmoid, (non-polynomial) rational functions, (non-polynomial) roots x^α, the logarithm and the exponential function. We can now give the proof.

Proof of Theorem 1.1. All involved functions have efficient spline-approximations in the sense of Theorem 3.1 (b). Moreover all involved functions are suitable and thus their circuits can (approximately) compute polynomials; in particular, they can (approximately) square.

The result follows from Theorem 3.1 (b), once all gate functions are shown to be equivalent to spline-circuits.

Let $\gamma(x) = \sigma(x)$. Since $\sigma(-x)$ is fast converging, σ is equivalent to splines.

Let $\gamma(x) = p(x)/q(x)$ be a rational function. First we mimic polynomial division. We obtain the rational $\gamma^* = p^*/q^*$, where the degree of p^* is less than the degree of q^*. Let us assume that the leading coefficients of both p^* and q^* is 1. Then we will find B and k ($k \geq 2$) such that $| \gamma^* - 1/x^k | \leq 2^{-s}$ for $x \geq B$. Thus we have established that γ is powerful, since $\gamma^*(x+B)$ is fast converging.

Let $\gamma(x) = x^\alpha$. For $k = \lceil \alpha \rceil$ we approximately differentiate γ $k+1$ times by computing the corresponding differential quotients. We are done since $\gamma^{(k)}$ is fast converging.

Let $\gamma(x) = \ln x$. Obviously $ln'(x)$ is fast converging. Finally, let $\gamma(x) = \exp(x)$. This time $\gamma(-x)$ is fast converging. ♠.

Remark 3.2
(a) *Of course, the equivalence of spline circuits and $\{\sigma\}$-circuits also holds for* **binary** *input. Since threshold circuits can add and multiply m m-bit numbers in constant depth and size polynomial in m (Reif, 1987), threshold circuits can efficiently approximate polynomials and splines.*

Thus, we obtain that σ-circuits of depth d, size s and Lipschitz-bound L can be simulated by circuits of binary thresholds. The depth of the simulating threshold circuit will increase by a constant factor and its size will increase by a polynomial in $(s+n)\log(L)$, where n is the number of input bits. (The inclusion of n accounts for the additional increase in size when approximately computing a weighted sum by a threshold circuit.)
(b) *If we allow size to increase by a polynomial in $s+n$, then Threshold circuits and σ-circuits are actually equivalent with respect to error bound 2^{-s}. This follows, since a threshold gate can easily be implemented by a sigmoidal gate (Maass et al., 1991). As a consequence, if we allow size to increase polynomially (in $s+n$) and depth to increase by a constant factor, then σ-circuits with weights that are at most exponential (in $s+n$) can be simulated by σ-circuits with weights of size polynomial in s.*

(c) *The equivalence of σ-circuits and threshold circuits does* **not** *hold for analog input, as we will see below. Also, for analog inputs, the size of the weights can* **not** *be restricted, since otherwise large Lipschitz-bounds can not be obtained.*

Our next goal is to verify that neither degree-bounded polynomials, the *sine*-function nor linear splines are equivalent to σ with respect to $e(s,d) = 2^{-s}$. We start with polynomials and linear splines.

Proposition 3.1
(a) *Let $abs(x) = | x |$. If a polynomial circuit of size s and depth d approximates $abs(x)$ over the interval $[-1,1]$, then the approximation error will be at least $s^{-O(d)}$.*
(b) *Let $sq(x) = x^2$. If a circuit of linear 1-splines (of size s and depth d) approximates $sq(x)$ over the interval $[-1,1]$, then its approximation error will be at least $s^{-O(d)}$.*

Proof.
(a) Let C be a polynomial circuit of depth d and size s. We assume that all gates of C are polynomials of degree at most s. Then, assuming that C computes a univariate function, C will compute a polynomial of degree at most s^d. But a polynomial of degree δ approximates $| x |$ with error at least $O(1/\delta)$ (page 59 in (Feinerman and Newman, 1974)).
(b) Let C be a circuit of linear 1-splines. Assume that C has size s and depth d. Obviously, C computes a linear spline with at most s^{d-1} knots. Thus there will be an interval I of length at least $s^{-(d-1)}$ such that C computes a linear function on I. Therefore, C will approximate $sq(x)$ with error at least $s^{-O(d)}$. ♠

Since both $sq(x)$ and $abs(x)$ can be computed exactly by degree 2 splines, they can be approximated with error 2^{-s} by σ-circuits. Thus, linear 1-splines and degree-bounded polynomials are **not** equivalent to σ with respect to error 2^{-s}. Moreover, linear splines and degree-bounded polynomials are incomparable with respect to error 2^{-s}. Finally, we consider the *sine*-function.

Definition 3.2 *Let $f : [a,b] \rightarrow \mathcal{R}$ be a function and let ε be a positive real number. We say that f ε-oscillates t times if and only if there are real numbers $a \leq x_1 < \ldots < x_{t+1} \leq b$ such that*

(a) $f(x_1) = f(x_2) = \ldots = f(x_{t+1})$,

(b) $|x_{i+1} - x_i| \geq \varepsilon$ *for all i and*

(c) *there are real numbers $y_1, \ldots, y_t$ such that $x_i \leq y_i \leq x_{i+1}$ and $|f(x_i) - f(y_i)| \geq \varepsilon$ for all i.*

Part (b) of the next theorem establishes that $\{\sigma\}$- and $\{sine\}$-circuits are not equivalent with respect to error 2^{-s}.

Lemma 3.1 *Assume that Γ can be simulated by σ with respect to error $e(s,d) = 2^{-s}$.*
(a) *Let $f : [-1,1] \rightarrow \mathcal{R}$ be a function that ε-oscillates t times and let C be a Γ-circuit of depth d, size s and Lipschitz-bound 2^s.*

If C approximates f with error at most $\frac{\varepsilon}{2}$, then $s \geq t^{\alpha/d}$. (The positive constant α depends only on Γ.)
(b) *The function $sine(Ax)$ can be approximated by a $\{\sigma\}$-circuit C_A with size $s = A^{O(1/d)}$, depth d and error at most $s^{O(-d)}$. On the other hand, every depth d Γ-circuit which approximates $sine(Ax)$ with error at most $\frac{1}{2}$, has to have size at least $A^{\Omega(1/d)}$.*
(c) *(Bit extraction.) Let f_A be any function with $f_A(i/A) = parity(i)$. Then the results of part (b) also apply to f_A.*

Proof.
(a) According to Theorem 3.1, C can be approximated (with error 2^{-s}) by a spline-circuit S of size $(s+1)^k$, depth $k(d+1)$, degree s and Lipschitz-bound 2^{s^k}. S will compute a spline of degree at most $s^{k(d+1)}$ and with at most $(s+1)^{k^2(d+1)-1}$ knots. Therefore, S oscillates at most $s^{O(d)}$ times. To achieve t oscillations, its size has to satisfy $s \geq t^{\Omega(1/d)}$. The claim follows by observing that less than t oscillations lead to an error larger than $\varepsilon/2$.
(b) The lower bound is an immediate consequence of part (a). The upper bound follows from Theorem 4.1., where it shown that σ and *sine* are equivalent with respect to error bound s^{-d} (when approximating functions with Lipschitz-bound s^d).
(c) Again, the lower bound is an immediate consequence of part (a). The upper bound follows from the proof technique of Theorem 4.1 (a). ♠

4 Equivalence of gate functions for error s^{-d}.

The lower bounds in the previous section suggest that the relaxed error bound $e(s,d) = s^{-d}$ is of importance. Indeed, it will turn out that most non-trivial smooth gate functions lead to circuits that simulate $\{\sigma\}$-circuits, **provided** the number of input units is counted when determining the size of the circuit. The concept of *threshold-property* will be crucial.

Definition 4.1 *Let Γ be a collection of gate functions. We say that Γ has the threshold-property if there is a constant c such that the following two properties are satisfied for all $m > 1$.*
(a) *For each $\gamma \in \Gamma$ there is a threshold circuit $T_{\gamma,m}$ of depth c, size $(s+m)^c$ which computes the binary representation of a rational number $T(x)$ with $|T(x) - \gamma(x)| \leq 2^{-m}$. The input x of $T_{\gamma,m}$ is given in binary and consists of $2m+1$ bits; m bits describe the integral part of x, m bits describe its fractional part and one bit indicates the sign. $s+m$ specifies the required number of output bits, i.e. $s = \lceil \log_2(\sup\{\gamma(x) : -2^{m+1} < x < 2^{m+1}\}) \rceil$.*
(b) *There is a Γ-circuit of depth c, size m^c and Lipschitz bound 2^{m^c} which $\frac{1}{m}$-approximates the binary threshold over $D = [-1,1] - [-1/m, 1/m]$.*

Theorem 4.1 *Assume that $e(s,d) = s^{-d}$.*
(a) *Let Γ be a class of gate functions and assume that Γ has the threshold property. Then, σ and Γ are equivalent with respect to e.* **Moreover**, *σ-circuits require weights and thresholds of absolute value at most s. (Observe that Γ-circuits are allowed to have weights as large as 2^s).*
(b) *If Γ and σ are quivalent with respect to error 2^{-s}, then Γ and σ are equivalent with respect to error s^{-d}.*
(c) *Additionally, the following classes are equivalent to σ-circuits with respect to e.*
- *polynomial circuits,*
- *γ-circuits, where γ is a suitable function and γ satisfies part (a) of Definition 4.1. (The sine-function is a special case.)*
- *linear spline circuits (assuming that all coefficients, weights and thresholds are bounded by 2^s for circuits of size s).*

Proof.
(a) Assume that Γ is given and that Γ has the threshold property.

Let C be a Γ-circuit of size s and depth d which computes a function f with Lipschitz-bound $1/e(s,d) = s^d$. We will construct a σ-circuit D approximating C with error at most s^{-d}. The first task of D is to compute the leading $2 \cdot d \log s$ bits of x.

Assume that this is achieved. We already know that each gate of Γ can be efficiently approximated by a threshold circuit if the input is given in binary. But the same also applies to a weighted sum, since threshold circuits can add and multiply n n-bit numbers in constant time and poly(n) size (Reif, 1987). Thus, we will be able to build a threshold circuit of depth $O(d)$ and size poly(s) which approximates C with *exponential accuracy* (if the input is given in binary). Finally, as already mentioned, σ-circuits can efficiently simulate threshold circuits. Thus C can be simulated with exponential accuracy by a σ-circuit (for binary input). But the binary input only approximates the original input with accuracy $O(s^{-2d})$. Nevertheless we achieve the desired accuracy of s^{-d}, since the functions to be approximated have Lipschitz-bound s^d. Observe that threshold circuits require only 0-1 weights; the corresponding simulation by $\{\sigma\}$-circuits can be achieved reliably with weights of size at most s.

To complete the simulation of C, we have to design a σ-circuit of depth $O(d)$ and size $s^{O(1)}$ which extracts the leading $2 \cdot d \log s$ bits of input $x \in [-1,1]$. With two layers, we can determine the interval(s) $[i/s, (i+1)/s]$ that x belongs to and consequently we can determine the leading $\frac{1}{2} \log s$ bits of x. Let x_1 be the corresponding approximation of x. We repeat the above procedure but now applied to $s^{1/2}(x - x_1)$ instead of x.

The corresponding simulation of $\{\sigma\}$-circuits by Γ-circuits is analogous.

(b) If Γ and σ are equivalent with respect to error 2^{-s}, then, by definition, Γ and σ are equivalent with respect to error s^{-d}.

(c) Part (a) of the threshold property is satisfied for all involved gate functions. Part (b) is obvious for linear splines and has to be verified only for polynomials.

Over the domain $D = [-1,1] - [-1/s, 1/s]$ the binary threshold function is a Lipschitz-bounded function with Lipschitz bound $s/2$. Hence, by Jackson's theorem (page 36 of (Feinerman and Newman, 1974)) it can be $\frac{1}{s}$-approximated by a polynomial of degree s. Moreover, weights of size at most $2^{poly(s)}$ suffice: interpolate the best polynomial approximation of a continuous version of the binary threshold and apply Cramer's rule to verify the size bound. ♠

5 A Lower Bound for Squaring

In this section we prove Theorem 1.2. We have to consider

$$M_n = \{(x,y) : \; x \in \{0,1\}^n, y \in \{0,1\}^{n^2} \;\text{ and }\; [x]^2 \geq [y]\} \quad (\text{where } \; [z] = \sum_i z_i).$$

By proposition 2.1, M_n can be recognized by a σ-circuit with constantly many gates. We have to show that any threshold circuit recognizing M_n must have at least $\Omega(\log n)$ gates.

Proof of Theorem 1.2. Let us assume that C_s is a threshold circuit with s gates recognizing M_n. Our approach will be to successively trivialize (i.e. *partially* fix the outcomes of) the gates of C_s by fixing appropriate bits of the input (x,y). The process of trivializing starts with source gates, continues with gates all of whose immediate predecessors have been trivialized and finally terminates with the sink gate of C_s.

Let us assume that the process has reached the gate

$$g(x,y) \equiv (\sum_{i=1}^{k} a_i x_i + \sum_{i=1}^{l} b_i y_i \geq t),$$

where we have only shown the remaining free bits (after appropriate renumbering). Observe that g will in general depend on the outcomes of other gates. But all these gates are already processed and their outcome is (partially) fixed; their remaining influence is captured by the threshold value t. We demand that the previously fixed bits $x_{k+1},\ldots,x_n$ and $y_{l+1},\ldots,y_{n^2}$ satisfy the following invariant.

(a) there exists $\alpha \geq 0$ (with $l = 2\alpha k + k^2$) so that C_s recognizes the language

$$M_n(k,l) = \{(x,y):\ x \in \{0,1\}^k, y \in \{0,1\}^l \text{ and, } (\alpha + [x])^2 \geq \alpha^2 + [y]\}.$$

(b) Each processed gate is constant for all inputs (x,y) with $x \in \{0,1\}^k, y \in \{0,1\}^l$ and $2\alpha[x] \leq [y] \leq 2\alpha[x] + x^2$.

Observe that once the sink of C_s is processed the entire circuit has been trivialized in the sense of the above invariant. Moreover, the circuit will compute *incorrectly* unless k, the number of free x-bits is zero. We will show below that k decreases by at most a factor of $1/360$ for each processed gate. This establishes the claim.

We now start to trivialize the gate g. Our first goal is enforce that all a_i's have the same sign and that all b_i's have the same sign. To achieve identical sign (positive or non-negative) for the a_i's, we set an appropriate collection of $k/2$ x-bits to 0. Let $p = k/2$. We repeat this procedure for the y-bits by setting $l/2$ y-bits to 0. Additionally, $\frac{3}{4}k^2$ y-bits are fixed to be 0. Thus, with $q = l/2 - \frac{3}{4}k^2$, we have $q = 2\alpha p + p^2$. Since α remains unchanged and we only set bits to 0, the invariant is still satisfied. Finally, we assume that

(a) bits $x_1,\ldots,x_p$ and $y_1,\ldots,y_q$ are free and

(b) $|a_1| \leq \ldots \leq |a_p|$ as well as $|b_1| \leq \ldots \leq |b_q|$.

Set $q^* = q - \alpha p + p^2/4 + 1$.
Case 1. $a_1 > 0$ and $b_1 > 0$.
Let $\beta_1 = \sum_{i=p/2+1}^{p} a_i$ and $\beta_2 = \sum_{i=q^*}^{q} b_i$.
Case 1.1. $\beta_1 + \beta_2 \geq t$.
We fix $x_{p/2+1} = \cdots = x_p = 1$ and $y_{q^*} = \cdots = y_q = 1$. Moreover, we set $\alpha^* = \alpha + p/2$. It is now immediate that the invariant is satisfied for α^* and that gate g is trivialized. We are left with $p/2 = k/4$ free x-bits.
Case 1.2. $\beta_1 + \beta_2 < t$.
We fix $x_{p/2+1} = \cdots = x_p = 0$ and $y_{q^*} = \cdots = y_q = 0$. The invariant is satisfied for α and gate g is again trivialized. $k/4$ free x-bits remain.

Case 2. $a_1 \leq 0$ and $b_1 \leq 0$.
The construction is analogous to Case 1.

Case 3. $a_1 > 0$ and $b_1 \leq 0$.

First we partition the x-positions into three classes $S_x = \{1, \ldots, p/3\}$, $M_x = \{p/3+1, \ldots, 2p/3\}$ and $L_x = \{2p/3+1, \ldots, p\}$. The y-positions are partitioned into $S_y = \{1, \ldots, 2\alpha p/3\}$, $M_y = \{2\alpha p/3+1, \ldots, 4\alpha p/3\}$, $L_y = \{4\alpha p/3+1, \ldots, 2\alpha p\}$ and the set Y of the remaining p^2 positions.

Next, we pair up positions in S_x and L_y. In particular, associate the last 2α elements of L_y with position 1, the smallest element of S_x. We define $c_1 = a_1 + \sum_{i=2\alpha p/3-2\alpha+1}^{2\alpha p/3} b_i$ as the corresponding sum of coefficients. We continue this process by associating the next 2α elements of L_y with position 2 and define c_2 as their corresponding sum of coefficients. Finally, we pair up the first 2α elements of L_y with position $p/3$, the largest ' element of S_x, and define $c_{p/3}$. Due to our case assumption we obtain $c_1 \leq \cdots \leq c_{p/3}$.

We continue by pairing up positions in M_x and M_y and then by pairing up positions in L_x and S_y. The corresponding coefficient sums satisfy $c_1 \leq \cdots \leq c_p$.

We set $d_i = b_{2\alpha p+i}$ and $h(u,v) \equiv (\sum_{i=1}^{p} c_i u_i + \sum_{i=1}^{p^2} d_i v_i \geq t)$ for $u \in \{0,1\}^p$ and $v \in \{0,1\}^{p^2}$. Observe that $g(x,y) \equiv h(u,v)$ **provided** $x_i = u_i$, $x_i = y_j$ for any j associated with the x-position i and **provided** $y_{2\alpha p+i} = v_i$.

This new representation allows us to interpret the "squaring problem after fixing" as the original squaring problem. Namely, assume that (x,y) is the input induced by (u,v). Then (x,y), together with the already fixed bits, belongs to M_n if and only if $[u]^2 \geq [v]$. Why is this so? The invariant guarantees that membership in M_n is equivalent to satisfying the inequality $(\alpha + [x])^2 \geq \alpha^2 + [y]$. Obviously this inequality is equivalent to $2\alpha[x] + [x]^2 \geq [y]$. But, since (x,y) is induced by (u,v), $[x] = [u]$ and $[v] = [y] - 2\alpha[x]$.

We set $V = \{p^2/3+1, \ldots, p^2/3 + p^2/9\}$. Let $\xi = \sum_{i \in M_x} c_i$ and $\eta = \sum_{i \in V} d_i$. Observe that η is non-positive. Moreover, let $\xi(r)$ denote the sum of c_i's where r varies over the first r elements of M_x. We define $\eta(r)$ to be the sum of d_i's where i varies over the last r elements of V. Then we obtain, for the average $a(r) = \frac{r}{p/3}$, $\xi(r) \leq a(r)\xi$ and $\eta(r^2) \leq a(r)^2\eta$. Consequently, $\xi(r) + \eta(r^2) \leq a(r)(\xi + a(r)\eta)$.

Case 3.1 $\xi \geq 0$.

Our first attempt at trivializing gate g will be to find a value for r ($1/60 \leq a(r) \leq 1$) such that $a(r)(2\xi + a(r)\eta) < t$ is satisfied. Assume that such an r exists. We then set $u_i = 1$ for the first r positions $i \in M_x$ and $v_j = 1$ for the last r^2 positions $j \in V$. Next, except for the first r positions from S_x, set all remaining free u-bits to zero. Finally, set $v_j = 0$ unless $j > p^2 - r^2$, and unless it is set to 1 in the previous step.

In other words, only r u-bits are free and r^2 v-bits are free. The free u-positions belong to S_x and the free v-positions belong to the smallest d-coefficients. Consequently, since we assume $\xi > 0$, $h(u,v) \leq a(r)(2\xi + a(r)\eta) < t$. Thus, we have trivialized the gate h.

We really need to trivialize the gate g. To achieve this, we first fix x-bits and y-bits as dictated by our fixing of u-bits and v-bits. Observe that gate g won't be forced to give constant output, since setting all free x-bits to 1 and all free y-bits to zero might result in a weighted sum that exceeds t. But our invariant only requires a modest trivialization, namely, gate g has to be constant provided $2\alpha[x] \leq [y] \leq 2\alpha[x] + [x]^2$.

Under this restriction, the most promising way of exceeding the threshold t is to set (say) r' x-bits (resp. $2\alpha r'$ y-bits) corresponding to the *largest* x-weights (y-weights) to 1. But the resulting input (x,y) is induced by some input (u,v) and, consequently, we have satisfied the invariant for gate g. Observe also that the invariant still holds for previously processed gates. The reason is that the domain over which gates have to be constant is shrinking.

Now let us assume that the first attempt of trivialization has failed. Our next attempt will be to find a value for r ($1/60 \leq a(r) \leq 1$) such that $a(r)(\xi + 2a(r)\eta) \geq t$. Let us assume that we are lucky.

First we redefine $\xi(r)$ and $\eta(r)$. $\xi(r)$ denotes now the sum of c_i's where r varies over the last r elements of M_x. $\eta(r)$ is the sum of d_i's where i varies over the first r elements of V. We obtain $\xi(r) \geq a(r)\xi$ and $\eta(r^2) \geq a(r)^2\eta$. Consequently, $\xi(r) + \eta(r^2) \geq a(r)(\xi + a(r)\eta)$.

We set $u_i = 1$ for the last r positions $i \in M_x$ and $v_j = 1$ for the first r^2 positions $j \in V$. Finally, except for the last r positions from L_x, we set all remaining free u-bits to zero and, except for the first r^2 positions of V we set all remaining v-bits to zero (unless already set to 0 in the previous step).

Thus, only r u-bits and r^2 v-bits are free. The free u-bits (v-bits) belong to the largest u-weights (v-weights) and we obtain $h(u,v) \geq a(r)(2\xi + a(r)\eta) \geq t$. As a consequence, gate h is trivialized.

Finally, we fix x-bits and y-bits according to our fixing of u-bits and v-bits. This time, the most promising way of "dipping" below the threshold value t is to set (say) r' x-bits (resp. $2\alpha r' + (r')^2$ y-bits) corresponding to the smallest x-weights (resp. y-weights) to 1. But again the resulting input (x,y) is induced by a (u,v) string, which establishes the invariant.

Observe that at least $k/(2 \cdot 3 \cdot 60)$ x-bits remain free for both attempts. To conclude the argument we have to face the failure of both attempts. In other words, for all r with $1/60 \leq a(r) \leq 1$,

$$a(r)(2\xi + a(r)\eta) \geq t \quad \text{and} \quad a(r)(\xi + 2a(r)\eta) < t.$$

Consequently, $2\xi + \eta \geq t$ (with $a(r) = 1$) and $\frac{1}{2}(\xi + \eta) < t$ (with $a(r) = 1/2$). We obtain $2\xi + \eta > \frac{1}{2}(\xi + \eta)$ and thus $\xi > -\frac{1}{3}\eta$.

On the other hand, we obtain $\frac{2}{60}\xi + {\frac{1}{60}}^2\eta \geq t$ (with $a(r) = 1/60$) and $\frac{1}{20}\xi + \frac{1}{200}\eta < t$ (with $a(r) = 1/20$). We get $\frac{2}{60}\xi + {\frac{1}{60}}^2\eta > \frac{1}{20}\xi + \frac{1}{200}\eta$. This inequality is equivalent to $\xi < -\frac{17}{60}\eta$.

As a consequence, $-\frac{17}{60}\eta > -\frac{1}{3}\eta$. But $\eta \leq 0$ and therefore $-\frac{1}{3}\eta > -\frac{17}{60}\eta$. We have derived a contradiction, implying that at least one of our attempts will succeed.

Case 3.2 $\xi < 0$. This case is essentially similar to Case 3.1. But this time the inequalities $a(r)(\xi + a(r)\eta) < t$, respectively $a(r)(2\xi + 2a(r)\eta) \geq t$, have to be considered. Observe that, if none of the inequalities is satisfied, we obtain

$$\xi + \eta < t \leq 2(\xi + \eta).$$

And this is impossible since both ξ and η are negative.

Case 4. $a_1 \leq 0$ and $b_1 > 0$.
Similar to case 3.

Remark 5.1. The lower bound of Theorem 5.1 is "almost" tight. It is quite easy to construct a threshold circuit of $O(\log n)$ gates that computes the binary representation of the sum of n bits. Now apply the Schoenhage-Strassen multiplication algorithm to obtain a threshold circuit of size $O(\log n \cdot \log\log n \cdot \log\log\log n)$ which computes the square of a given sum of bits.

6 Conclusions and open Problems.

In summary, the standard sigmoid reaches an approximation power comparable to or better than classes of more established functions investigated in approximation theory (i.e. splines and polynomials). An additional advantage of the standard sigmoid is that its approximation performance scales with respect to the error bound: larger error results in decreasing weight size. Moreover, the standard sigmoid is actually more powerful than the binary threshold even when computing boolean functions.

Our results show that good approximation performance (for error 2^{-s}) hinges on two properties, namely efficient approximation of polynomials and efficient approximation of the binary threshold. These two properties are shared by a quite large class of gate function; i.e. powerful functions. Since (non-polynomial) rational functions are powerful, we were able to generalize Newman's approximation of $| x |$ by rational functions.

For a good approximation performance relative to the relaxed error bound s^{-d} it is sufficient to efficiently approximate the binary threshold. Consequently, the class of equivalent gate functions grows considerably.

Finally, compared to the binary threshold, the standard sigmoid is able to take advantage of its (non-trivial) smoothness to allow more efficient circuits.

The following problems remain open.

- Determine good upper bounds on the number of zeroes of a $\{\sigma\}$-circuit of size s and depth d independent of the size of the weights.
- Does the considerable approximation power of the standard sigmoid translate into a good learning performance (when learning to approximate)?

Acknowledgements.

We wish to thank R. Paturi, K. Y. Siu and V. P. Roychowdhury for helpful discussions. Special thanks go to W. Maass for suggesting this research, to E. Sontag for continued encouragement and very valuable advice and to J. Lambert for his never-ending patience.

References

AHLBERG, J. H., NILSON, E. N., and WALSH, J. L. (1967), The theory of splines and their applications, Academic Press.

ARAI, W. (1989), Mapping abilities of three-layer networks, *in* "Proc. of the International Joint Conference on Neural Networks", pp. 419-423.

CARROL, S. M., and DICKINSON, B. W. (1989), Construction of neural nets using the Radon Transform,*in* "Proc. of the International Joint Conference on Neural Networks", pp. 607-611.

CYBENKO, G. (1989), Approximation by superposition of a sigmoidal function, *Mathematics of Control, Signals, and Systems*, 2, pp. 303-314.

FEINERMAN, R. P., and NEWMAN, D. J. (1974), Polynomial Approximation, The Williams and Wilkins Company, pp. 36-38.

FUNAHASHI, K. (1989), On the approximate realization of continuous mappings by neural networks, *Neural Networks*, 2, pp. 183-192.

GALLANT, A. R., and WHITE, H. (1988), There exists a neural network that does not make avoidable mistakes, *in* "Proc. of the International Joint Conference on Neural Networks", pp. 657-664.

HORNIK, K., STINCHCOMBE, M., and WHITE, H. (1989), Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, 2, pp. 359-366.

IRIE, B., and MIYAKE, S. (1988), Capabilities of the three-layered perceptrons, *in* "Proc. of the International Joint Conference on Neural Networks", pp. 641-648.

KARLIN, S., and STUDDEN, W. J. (1966), Tchebycheff systems: with applications in analysis and statistics, J. Wiley.
LAPADES, A., and FARBAR, R. (1987), How neural nets work, *in* "Advances in Neural Information Processing Systems", pp. 442-456.
MAASS, W., SCHNITGER, G., and SONTAG, E. (1991), On the computational power of sigmoid versus boolean threshold circuits,*in* "Proc. of the 32nd Annual Symp. on Foundations of Computer Science", pp. 767-776.
NEWMAN, D. J. (1964), Rational approximation to $\mid x \mid$, *Michigan Math. Journal*, 11, pp. 11-14.
HECHT-NIELSON, R. (1989), Theory of backpropagation neural networks, *in* "Proc. of the International Joint Conference on Neural Networks", pp. 593-611.
POGGIO, T., and GIROSI, F. (1989), A theory of networks for Approximation and learning, *Artificial Intelligence Memorandum*, no 1140.
REIF, J. H. (1987), On threshold circuits and polynomial computation, *in* "Proceedings of the 2nd Annual Structure in Complexity theory", pp. 118-123.
WEGENER, I. (1991), The complexity of the parity function in unbounded fan-in, unbounded depth circuits, Theo. Comp. Sci, 85, pp. 155-170.
WEI, Z., YINGLIN, Y., and QING, J. (1991), Approximation property of multi-layer neural networks (MLNN) and its application in nonlinear simulation, *in* "Proc. of the International Joint Conference on Neural Networks", pp. 171-176.
WILLIAMSON, R. C., and PAICE, A. D. B. (1991), The number of nodes required in a feed-forward neural network for functional representation, *manuscript.*