JMLR: Workshop and Conference Proceedings vol 49:1–19, 2016

# Cortical Computation via Iterative Constructions

**Christos Papadimitriou** CHRISTOS@BERKELEY.EDU
*UC Berkeley*

**Samantha Petti** SPETTI3@GATECH.EDU
*Georgia Tech*

**Santosh Vempala** VEMPALA@GATECH.EDU
*Georgia Tech*

## Abstract

We study Boolean functions of an arbitrary number of input variables that can be realized by simple iterative constructions based on constant-size primitives. This restricted type of construction needs little global coordination or control and thus is a candidate for neurally feasible computation. Valiant's construction of a majority function can be realized in this manner and, as we show, can be generalized to any uniform threshold function. We study the rate of convergence, finding that while linear convergence to the correct function can be achieved for any threshold using a fixed set of primitives, for quadratic convergence, the size of the primitives must grow as the threshold approaches 0 or 1. We also study finite realizations of this process and the learnability of the functions realized. We show that the constructions realized are accurate outside a small interval near the target threshold, where the size of the construction grows as the inverse square of the interval width. This phenomenon, that errors are higher closer to thresholds (and thresholds closer to the boundary are harder to represent), is a well-known cognitive finding.

**Keywords:** Cortical Computation, Iterative Constructions, Monotone functions, Threshold functions. [1]

## 1. Introduction

Among the many unexplained faculties of the mammalian cortex is its ability learn complex patterns and invariants from relatively few examples. This is manifested in a range of cognitive functions including visual and auditory categorization, motor learning and language. In spite of the highly varied perceptual and cognitive tasks accomplished, the substrate appears to be relatively uniform in the distribution and type of cells. How could these 80 billion cells organize themselves so effectively?

Cortical computation must therefore be highly distributed, require little *synchrony* (number of pairs of events that must happen in lock-step across neurons), little *global control* (longest chain of events that must happen in sequence) and be based on very simple primitives (Papadimitriou and Vempala, 2015b). Assuming that external stimuli are parsed as sets of binary sensory features, our central question is the following:

*What functions can be represented and learned by algorithms so simple that one could imagine them happening in the cortex?*

1. Extended abstract. Full version appears as arXiv:1602.08357, v2

© 2016 C. Papadimitriou, S. Petti & S. Vempala.

Perhaps the most natural primitives are the AND and OR functions on two input variables. These functions are arguably neurally plausible. They were studied as JOIN and LINK by Valiant (1994, 2000, 2005); Feldman and Valiant (2009), who showed how to implement them in the neuroidal model. An *item* is a collection of neurons (corresponding to a *neural assembly* in neuroscience) that represents some learned or sensed concept. Given two items $A, B$, the JOIN operation forms a new item $C = \text{JOIN}(A, B)$, which "fires" when both $A$ and $B$ fire, i.e., $C$ represents $A \wedge B$. LINK$(A, B)$ captures association, and causes $B$ to fire whenever $A$ fires. By setting LINK$(A, C)$ and LINK$(B, C)$, we achieve that $C$ is effectively $A \vee B$. While the precise implementation and neural correlates of JOIN and LINK are unclear, there is evidence that the brain routinely engages in hierarchical memory formation.

**Monotone Boolean functions.** Functions constructed by recursive processes based on AND/OR trees have been widely studied in the literature, motivated by the design of reliable circuits as in (Moore and Shannon, 1956) and more recently, understanding the complexity-theoretic limitations of monotone Boolean functions. One line of work studies the set of functions that could be the limits of recursive processes, where at each step, the leaves of a tree are each replaced by constant-size functions. Moore and Shannon (1956), showed that a simple recursive construction leads to a threshold function, which can be applied to construct stable circuits. Valiant (1984) used their 4-variable primitive function $(A \vee B) \wedge (C \vee D)$ to derive a small depth and size threshold function that evaluates to 1 if at least $(2 - \phi) \approx 0.38$ fraction of the inputs are set to 1 and to zero otherwise. The depth and size were $O(\log n)$ and $O(n^{5.3})$ respectively. Calling it the *amplification method*, Boppana (1985) showed that Valiant's construction is optimal. Dubiner and Zwick (1992) extended the lower bound to classes of read-once formulae. Hoory et al. (2006) gave smaller size Boolean *circuits* (where each gate can have fan out more than 1), of size $O(n^3)$ for the same threshold function. Luby et al. (1998) gave an alternative analysis of Valiant's construction along with applications to coding. The construction of a Boolean formula was extended by Servedio (2004) to monotone linear threshold functions, in that they can be approximated on most inputs by monotone Boolean formulae of polynomial size. Friedman (1986) gave more efficient constructions for threshold functions with small thresholds.

Savicky gives conditions under which the limit of such a process is the uniform distribution on all Boolean functions with $n$ inputs (Savicky, 1987, 1990) (see also Brodsky and Pippenger (2005); Fournier et al. (2009)). In a different application, Goldman et al. (1993) showed how to use properties of these constructions to identify read-once formulae from their input-output behavior.

**Our work.** Unlike previous work, where a single constant-sized function is chosen and applied recursively, we will allow constructions that randomly choose one of two constant-sized functions. To be neurally plausible, our constructions are *bottom-up* rather than *top-down*, i.e., at each step, we apply a constant-size function to an existing set of outputs. In addition, the algorithm itself must be very simple — our goal is not to find ways to realize all Boolean functions or to optimize the size of such realizations. Here we address the following questions: What functions of $n$ input items can be constructed in this iterative manner? Can arbitrary uniform threshold functions be realized? What size and depth of iterative constructions suffices to guarantee accurate computations? Can such functions and constructions be learned from examples, where the learning algorithm is also neurally plausible?

Our rationale for uniform threshold functions is two-fold. First, uniform threshold functions are fundamental in computer science and likely also for cognition. Second, the restriction to JOIN and LINK as primitives ensures that any resulting function will be monotone since negation is not possible in this framework. Moreover, if we require the construction to be symmetric, it would seem that the only obtainable family of Boolean functions are uniform thresholds. However, as we will see, there is a surprise here, and in fact we can get *staircase functions*, i.e., functions that take value $p_i$ on the interval $(a_i, a_{i+1})$ where $a_0 = 0 < a_1 < a_2 < \cdots < a_k < a_{k+1} = 1$ and $0 = p_0 < p_1 < p_2 < \cdots < p_{k-1} < p_k = 1$.

To be able to describe our results precisely, we begin with a definition of iterative constructions.

### 1.1. Iterative constructions

A sequence of AND/OR operations can be represented as a tree, as depicted in Figure 2. Such a tree $T$ with $n$ leaves naturally computes a function $g_T : \{0,1\}^n \to \{0,1\}$. We can build larger trees in a neurally plausible way by using a set of small AND/OR trees as building blocks. Let $C$ be a probability distribution on a finite set of trees. We define a *iterative tree for $C$* as follows.

**IterativeTree($L, m, C, X$):**

For each level $j$ from $1$ to $L$, apply the following iteration $m$ times:
(level $0$ consists of the input items $X$)

1. Choose a tree $T$ according to $C$.
2. Choose items uniformly at random from the items on level $j - 1$.
3. Build the tree $T$ with these items as leaves.

The construction of small AND/OR trees is a decentralized process requiring a short sequence of steps, i.e., the synchrony and control parameters are small. Therefore, we consider them to be neurally plausible.

The iterative tree construction has a well-defined sequence of *levels*, with items from the next level having leaves only in the current level. A construction that needs even less coordination is the following: each item has a *active* period and the probability that it participates in future item creation decays exponentially with time. The weight of an item starts at 1 when it is created and decays by a factor of $e^{-\alpha}$ each time unit. We refer to such constructions as *exponential* iterative constructions. An extreme version of this, which we call *wild* iterative construction, is to have $\alpha = 0$, i.e, all items are equally likely to participate in the creation of new items. Figure 1 illustrates these constructions.

### 1.2. Results

We are interested in the functions computed by high level items of iterative constructions. In particular, we design iterative constructions so that high level items compute a threshold function with high probability. We state our results here. A full version with complete proofs is available on the arXiv.

**Definition 1** *The function $f : [0,1] \to [0,1]$ is a t-threshold if $f(x) = 0$ for $x < t$ and $f(x) = 1$ for $x > t$.*

**ExponentialConstruction($k, C, \alpha$):**

Initialize the weights of input items to 1.
Construct $k$ items as follows:

1. Choose a tree $T$ according to $C$.
2. Choose the leaves for $T$ independently from existing items with probability proportional to the weight of the item.
3. Build the tree $T$ with these items as leaves.
4. Multiply the weight of every item by $e^{-\alpha}$.

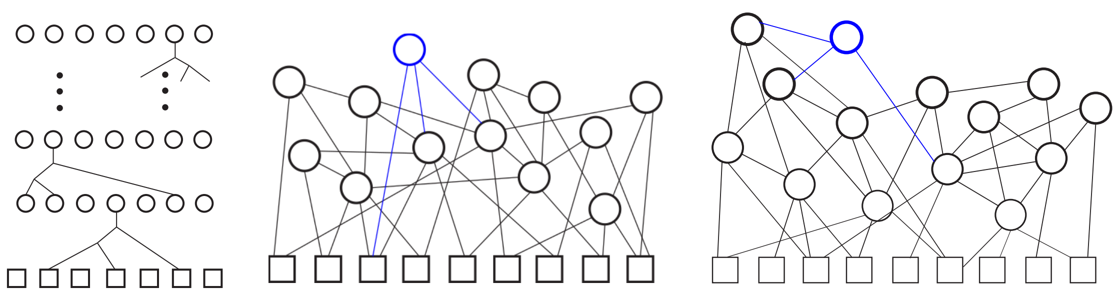

Figure 1: Left: an iterative construction. Middle: a wild construction. Right: an exponential construction. In the latter two images, the thickness of the outline representing each item indicates the probability the item will be selected in the construction of the next item.

For given probability distribution on a set of trees, the output of high level items of a corresponding iterative construction depends on the following: $(i)$ the fraction of input items firing, $(ii)$ the width of the levels, and $(iii)$ the number of levels. For an $n$ item input, the fraction of input items firing must take the form $k/n$, $k \in \mathbb{Z}$. Throughout the paper, we assume that the distance between the desired threshold and the fraction of input items firing is at least $1/n$. To address $(ii)$, we first analyze the functions computed by high level items of an iterative construction when the width of the levels is infinite, which is equivalent to the "top down" approach. Then, we remove this assumption and analyze the "bottom-up" construction in which the items at level $j-1$ are fixed before the items at level $j$ are created. The following theorems give a guarantee on the probability that an iterative tree with infinite width levels accurately computes a threshold function in terms of the number of levels. To start, we restate Valiant's result Valiant (1984). Here $\phi = (\sqrt{5}+1)/2$ is the golden ratio ($2-\phi \approx 0.38$).

**Theorem 2** *Let $R$ be the tree that computes $(A \vee B) \wedge (C \vee D)$. Then, an item at level $\Omega(\log n + \log k)$ of an infinite width iteratively constructed tree for $R$ computes a $(2-\phi)$-threshold function accurately with probability at least $1-2^{-k}$.*

In this construction, the iterative tree that computes the $2-\phi$ threshold function is built using only one small tree. We show that it is possible to achieve arbitrary threshold functions if we allow our iterative tree to be built according to a probability distribution on two distinct trees.

**Theorem 3** *Let $0 < t < 1$ and let $R = \{\mathsf{Pr}(T_1) = t, \mathsf{Pr}(T_2) = 1-t\}$ where $T_1$ is the tree that computes $(A \vee B) \wedge C$ and $T_2$ is the tree that computes $(A \wedge B) \vee C$. Then, an item at level $\Omega(\log n + k)$ of an infinite width iteratively constructed tree for $R$ computes a $t$-threshold function accurately with probability at least $1 - 2^{-k}$.*

The rate of convergence of this more general construction is linear rather than quadratic. While both are interesting, the latter allows us to guarantee a correct function on every input with depth only $O(\log n)$, since there are $2^n$ possible inputs.

**Definition 4** *A construction exhibits linear convergence if items at level $\Omega(\log n + k)$ of an infinite width iterative tree accurately compute the threshold function with probability at least $1 - 2^{-k}$. A construction exhibits quadratic convergence if items at level $\Omega(\log n + \log k)$ of an infinite width iterative tree accurately compute the threshold function with probability at least $1 - 2^{-k}$.*

The next theorem gives constructions using slightly larger trees with 4 and 5 leaves respectively (illustrated in Figure 2) that converge quadratically to a $t$-threshold function for a range of values of $t$, with more leaves giving a larger range. Moreover, these ranges are tight, i.e. no construction on trees with 4 or 5 leaves yields quadratic convergence to a $t$-threshold function for $t$ outside these ranges.

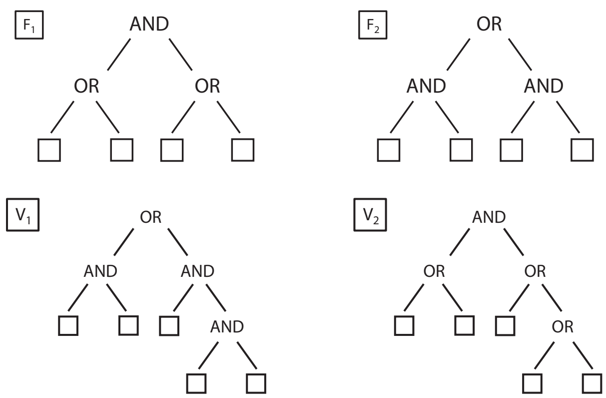


Figure 2: For $.38 \lesssim t \lesssim 0.62$, there exists a probability distribution on $F_1$ and $F_2$ that yields an iterative tree that converges quadratically to a $t$-threshold function. For $.26 \lesssim t \lesssim 0.74$, there exists a probability distribution on $V_1$ and $V_2$ that yields an iterative tree that converges quadratically to a $t$-threshold function.

**Theorem 5**

*(A) Let $2-\phi \leq t \leq \phi - 1$ and $\alpha(t) = \frac{1-t-t^2}{2t(t-1)}$. Define $R = \{\mathsf{Pr}(F_1) = \alpha(t), \mathsf{Pr}(F_2) = 1-\alpha(t)\}$ be the probably distribution on trees in Figure 2. Then, an item at level $\Omega(\log n + \log k)$ of an infinite*

*width iteratively constructed tree for $R$ computes a $t$-threshold function accurately with probability at least $1-2^{-k}$. Moreover, for $t$ outside this range, there exists no such construction on trees with four leaves that converge quadratically to a $t$-threshold function.*

*(B) Let $\alpha(t) = \frac{-1+5t-4t^2+t^3}{5t(t-1)}$ and let $t$ be a value for which $0 \leq \alpha(t) \leq 1$, so $0.26 \lesssim t \lesssim 0.74$. Let $R = \{\mathsf{Pr}(V_1) = \alpha(t), \mathsf{Pr}(V_2) = 1-\alpha(t)\}$ be the probably distribution on trees in Figure 2. Then, an item at level $\Omega(\log n + \log k)$ of an infinite width iteratively constructed tree for $R$ computes a $t$-threshold function accurately with probability at least $1-2^{-k}$ . Moreover, for $t$ outside this range, there exists no such construction on trees with five leaves that converges quadratically to a $t$-threshold function.*

As the desired threshold $t$ approaches 0 or 1, we show that an iterative tree that computes the $t$-threshold function must use increasingly large trees as building blocks.

**Theorem 6** *Let $t$ be a threshold, $0 < t < 1$ and let $s = \min\{t, 1-t\}$. Then, the construction of an iterative tree whose level $\Omega(\log n + \log k)$ items compute a $t$-threshold function with probability at least $1-2^{-k}$ must be defined over a probability distribution on trees with at least $\frac{1}{\sqrt{2s}}$ leaves.*

This raises the question of whether it is possible to have quadratic convergence for any threshold. We can extend the constructions described in Theorem 5 by using analogous trees with six and seven leaves to obtain quadratic convergence for thresholds in the ranges $0.15 \lesssim t \lesssim 0.85$ and $0.11 \lesssim t \lesssim 0.89$ respectively. However, it is not possible to generalize this construction beyond this point. Instead, to achieve quadratic convergence for thresholds near the boundaries, we turn to the following construction, which asymptotically matches the lower bound of Theorem 6. We define $A_k$ as a tree on $2k$ leaves that computes $(x_1 \vee x_2 \vee \cdots \vee x_k) \wedge (x_{k+1} \vee x_{k+2} \vee \cdots \vee x_{2k})$ and $B_k$ as a tree on $2k$ leaves that computes $(x_1 \wedge x_2 \wedge \cdots \wedge x_k) \vee (x_{k+1} \wedge x_{k+2} \wedge \cdots \wedge x_{2k})$.

**Theorem 7** *For any $0 < t \leq 2-\phi$, there exists $k$ and a probability distribution on $A_k$ and $A_{k+1}$ that yields an iterative tree with quadratic convergence to a $t$-threshold function. Similarly for any $\phi - 1 \leq t < 1$, there exists $k$ and a probability distribution on $B_k$ and $B_{k+1}$ that yields an iterative tree with quadratic convergence to a $t$-threshold function.*

There is a trade-off between constructing iterative trees that converge faster and requiring minimal coordination in order to build the subtrees. Building a specified tree on a small number of leaves requires less coordination than building a specified tree on many leaves. Therefore, as $t$ approaches 0 or 1, constructing an iterative tree with quadratic convergence becomes less neurally plausible because the construction of each subtree requires much coordination. These results are in line with behavioral findings (Rosch et al., 1976; Rosch, 1978) and computational models (Arriaga and Vempala, 2006; Arriaga et al., 2015) about categorization being easier when concepts are more robust.

We characterize the class of functions that can be achieved by iterative constructions allowing building block trees of any size. We show that it is possible to achieve an arbitrarily close approximation of any staircase function in which each step intersects the line $y = x$.

Next we turn to finite realizations of iterative trees. The above theorems analyze the behavior of an iterative construction where the width of the levels is infinite. We assumed that for any input the number of items firing at given level of the tree is equal to its expectation. However, imagining a "bottom up" construction, we note that the chance that the number of items firing at a given level deviates from expectation is non-trivial. Such deviations percolate up the tree and effect the probability that high level items compute the threshold function accurately. The smaller the width of a level, the more likely that the number of items firing at that level deviates significantly from expectation, rendering the tree less accurate. How large do the levels of an iteratively constructed tree need to be in order to ensure a reasonable degree of accuracy?

**Theorem 8** *Consider a construction of a $t$-threshold function with quadratic convergence described in Theorem 5 or Theorem 7 in which each level $\ell$ has $m_\ell$ items and the fraction of input items firing is at least $\varepsilon$ from the threshold $t$. Then, with probability at least $1-\gamma$, items at level $\Omega(\log \frac{1}{\gamma} + \log \frac{1}{\varepsilon}))$ will accurately compute the threshold function for $m_1 = \Omega\left(\frac{\ln(1/\gamma)}{\varepsilon^2}\right)$ and $\sum_\ell m_\ell = O(m_1)$.*

As a direct corollary, by setting $\varepsilon = O(1/n)$ and $\gamma = 2^{-n-1}$, we realize a $t$-threshold construction of size $O(n^3)$ for any $t$, matching the best-known construction which was for a specific threshold (Hoory et al., 2006). The finite-width version of Theorem 3 is given in Section 4.

The exponential iterative construction also converges to a $t$-threshold function for appropriate $\alpha$. We give the statements here for the wild iterative construction (with no weight decay) and the general exponential construction.

**Theorem 9** *Consider a wild construction on $n$ inputs for the $t$-threshold function given in Theorem 3 in which $n > \log(\frac{1}{8\varepsilon\delta}) \max\{\frac{1}{\varepsilon^2}, \frac{1}{\delta^2}\}$ where $\varepsilon$ is the distance between $t$ and the fraction of inputs firing. Then, there is an absolute constant $c$ such that for $k = \Omega\left(n\left(\frac{1}{\delta^c} + \frac{1}{\varepsilon^c}\right)\right)$, the $k^{th}$ item accurately computes the $t$-threshold function with probability at least $1-\delta$.*

**Theorem 10** *Consider an exponential construction on $n$ inputs for the $t$-threshold function given in Theorem 3 in which $\alpha \le \frac{\min\{\varepsilon^2,\delta^2\}}{2\log(1/\varepsilon\delta)}$ and $n > 1/\alpha$. Then for*

$$k = \Omega\left(\frac{n}{\min\{\varepsilon^2, \delta^2\}}\left(\log n + \log^2 \frac{1}{\varepsilon\delta}\right)\right),$$

*with probability at least $1-\delta$, the $k^{th}$ item will compute the $t$-threshold function.*

Finally, we give a simple cortical algorithm to learn a uniform threshold function from a single example, described more precisely by the following theorem.

**Theorem 11** *Let $X \in \{0,1\}^n$ such that $||X||_1 = tn$, $L = \Omega\left(\log \frac{1}{\gamma} + \log \frac{1}{\varepsilon}\right)$, and $\varepsilon = \Omega\left(\sqrt{\frac{\ln(1/\gamma)}{m}}\right)$. Then, on any input in which the fraction of input items firing is outside $[t-\varepsilon, t+\varepsilon]$, items at level $L$ of an iterative tree produced by LearnThreshold($L, m, X$) will compute a $t$-threshold function with probability at least $1-\gamma$.*

## 2. Polynomials of AND/OR Trees

Let $g_T : \{0,1\}^n \to \{0,1\}$ be the Boolean function computed by an AND/OR tree $T$ with $n$ leaves. We define $f_T$ as the probability that $T$ evaluates to 1 if each input item is independently set to 1 with probability $p$.

$$f_T(p) = \mathsf{Pr}\left(g_T(X) = 1 \,|\, X \sim B(n,p)\right).$$

We analogously define $f_C(p)$ for probability distributions on trees; let $f_C$ be the probability that a tree chosen according to $C$ evaluates to 1 if each input item is independently set to 1 with probability $p$. Let $\lambda_T$ be the probability of $T$ in distribution $C$. We have

$$f_C(p) = \sum_{T \in C} \lambda_T f_T(p).$$

In an iterative construction for the probability distribution $C$, an item at level $k$ evaluates to 1 with probability $f_C(p_{k-1})$ where $p_{k-1}$ is the probability that an item at level $k-1$ evaluates to 1. In the case where the width of the levels is infinite, the fraction of inputs firing any level is exactly equal its expectation. Therefore, the probability that items at level $k$ evaluate to 1 is $f_C^{(k)}(p)$ where $p$ is the probability an input is set to 1. This follows directly from the recurrence relation:

$$f_C^{(k)}(p) = f_C(f_C^{(k-1)}(p)).$$

We call a polynomial *achievable* if it can be written as $f_T$ for some AND/OR tree $T$. We call a polynomial *achievable through convex combinations* if it can be written as $f_C$ for some probability distribution on AND/OR trees $C$. Note that $\mathcal{A}$ is closed under the AND and OR operations. If $a, b \in \mathcal{A}$, then $a \cdot b \in \mathcal{A}$ and $a + b - a \cdot b \in \mathcal{A}$. The set of polynomials achievable through convex combinations is the convex hull of $\mathcal{A}$.

**Lemma 12** *Let $f \in \mathcal{A}$ be an achievable polynomial of degree $d$, $f = a_0 + a_1 x + a_2 x^2 + \dots a_d x^d$. Then $|a_\ell| \le d^\ell$.*

**Proof** Proceed by induction. The only achievable polynomial of degree 1 is $f(x) = x$, so the statement clearly holds. Next, assume $|a_{l'}| \le d^{l'}$ holds for all $l' < l$. Let $f$ be a degree $d$ achievable polynomial. We may assume $f = g + h - gh$ or $f = gh$ where $g$ and $h$ are achievable polynomials with degree $k$ and $d - k$ respectively where $k \le \frac{l}{2}$. First consider the case when $f = g + h - gh$, meaning the root of the tree corresponding to $f$ is an OR operation. Observe

$$\begin{aligned}
|a_\ell(f)| &= \big| a_\ell(g) + a_\ell(h) - \sum_{i=1}^{l-1} a_i(g) a_{l-i}(h) \big| \\
&\le k^l + (d-k)^l + \sum_{i=1}^{l-1} k^i (d-k)^{l-i} \\
&\le ((d-k)+k)^l \\
&= d^l.
\end{aligned}$$

Next consider the case when $f = gh$, meaning the root of the tree corresponding to $f$ is an AND operation. Observe that

$$|a_\ell(f)| = \big| \sum_{i=1}^{l-1} a_i(g) a_{l-i}(h) \big| \le \sum_{i=1}^{l-1} k^i (d-k)^{l-i} < d^l.$$

■

We observe a relationship between the polynomial of a tree and the polynomial of its complement. We define the complement of the AND/OR tree $T$ to be the tree obtained from $T$ by switching the operation at each node.

**Lemma 13** *Let $A$ and $B$ be complementary AND/OR trees and let $f_A$ and $f_B$ be the corresponding polynomials. Then $f_B(1-p) = 1 - f_A(p)$ for all $0 < p < 1$.*

Let $f_A$ be a polynomial achievable through convex combinations, $f_A = \sum_{i=1}^n \lambda_i f_{A_i}$. Let $A_i$ and $B_i$ be complementary AND/OR trees. Let $f_B = \sum_{i=1}^n \lambda_i f_{B_i}$. We say that $f_A$ and $f_B$ are *complementary polynomials*.

**Corollary 14** *Let $f_A$ and $f_B$ be complementary polynomials. Then*

1. *For all $0 < p < 1$, $f_B(1-p) = 1 - f_A(p)$*
2. *If $p$ is a fixed point of $f_A$ then $1-p$ is a fixed point of $f_B$*
3. *For all $0 < p < 1$, $f_B^{(k)}(1-p) = 1 - f_A^{(k)}(p)$.*

Finally, we make some observations about the polynomials associated with the specific family of trees we use in many of our constructions.

**Definition 15** *Let $A_k$ be a tree on $2k$ leaves that computes $(x_1 \vee x_2 \vee \cdots \vee x_k) \wedge (x_{k+1} \vee x_{k+2} \vee \cdots \vee x_{2k})$. Let $B_k$ be a tree on $2k$ leaves that computes $(x_1 \wedge x_2 \wedge \cdots \wedge x_k) \vee (x_{k+1} \wedge x_{k+2} \wedge \cdots \wedge x_{2k})$.*

**Lemma 16** *Let $f_{A_k}$ and $f_{B_k}$ be the polynomials corresponding to $A_k$ and $B_k$ respectively. Then $f_{A_k}$ has a unique fixed point in the interval $\left(\frac{1}{k^2}, \frac{1}{k(k-1)}\right)$ and $f_{B_k}$ has a fixed point in the interval $\left(1 - \frac{1}{k(k-1)}, 1 - \frac{1}{k^2}\right)$.*

**Lemma 17** *Let $f_{A_k}$ and $f_{B_k}$ be the polynomials corresponding to $A_k$ and $B_k$ respectively. For $0 < t \le 2 - \phi$, there exists some $k$ and $\alpha$ such that $f_A = \alpha f_{A_k} + (1-\alpha) f_{A_{k+1}}$ has fixed point $t$. Moreover, $\frac{t - f_A(p)}{t-p} \ge \left(1 + \frac{p(1-p)}{t}\right)$. Similarly, for $\phi - 1 \le t < 1$, there exists some $k$ and $\alpha$ such that $f_B = \alpha f_{B_k} + (1-\alpha) f_{B_{k+1}}$ has fixed point $t$. Moreover, $\frac{t - f_B(p)}{t-p} \ge \left(1 + \frac{p(1-p)}{t}\right)$.*

## 3. Convergence of iterative trees to threshold functions

In the previous section, we showed that if the width of each level is infinite, then items at level $k$ of an iterative tree evaluate to 1 with probability $f_C^{(k)}(p)$ when each input is independently set to 1 with probability $p$. In this section we prove that the probability distribution given in Theorem 3 converges to a $t$-threshold function.

By an abuse of notation, we say that $f(p)$ converges to a $t$-threshold function if

$$\lim_{k\to\infty} f^{(k)}(p) = \begin{cases} 0 & 0 \le p < t \\ 1 & t < p \le 1 \\ t & p = t. \end{cases}$$

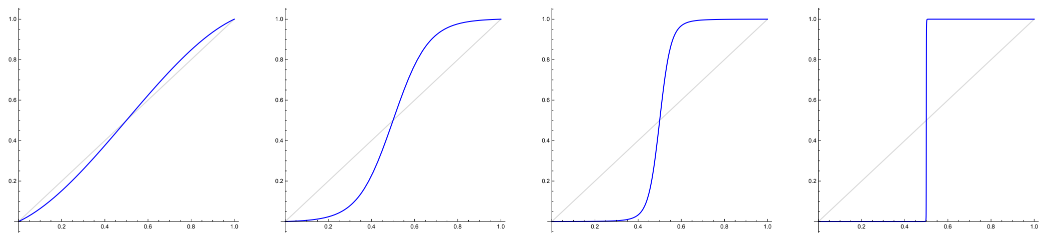

Figure 3: A function $f$ that converges to a $1/2$-threshold function. Left to right: $f(p), f^{(5)}(p), f^{(10)}(p), f^{(30)}(p)$.

Moreover, we say that $f$ converges quadratically to a $t$-threshold function if the corresponding iterative construction exhibits quadratic convergence. The function depicted in Figure 3 converges to a $1/2$-threshold function.

The following lemma gives sufficient conditions for convergence to a $t$-threshold function.

**Lemma 18** *Let $f$ be a function corresponding to an iterative construction on $n$ inputs.*

1. *On the interval $[0, 1]$, $f$ has precisely three fixed points: $0, t,$ and $1$.*
2. *(Linear Divergence) There exists constants $u, v$ satisfying $0 < u < t$ and $t < v < 1$ and constants $c_1, c_2 > 1$ such that*
   - *(a) $t - f(p) \geq c_1(t - p)$ for $p \in [u, t - \frac{1}{n}]$, and*
   - *(b) $f(p) - t \geq c_2(p - t)$ for $p \in [t + \frac{1}{n}, v]$.*
3. *(Linear Convergence) For the constants $u, v$ as above, there exists constants $c_3, c_4$ such that $c_3 u < 1$ and $c_4(1 - v) < 1$ and*
   - *(a) $f(p) < c_3 p$ for $p \in (0, u)$, and*
   - *(b) $1 - f(p) < c_4(1 - p)$ for $p \in (v, 1)$.*
4. *(Quadratic Convergence) For the constants $u, v$ as above, there exists constants $c_3, c_4$ such that $c_3 u < 1$ and $c_4(1 - v) < 1$ and*
   - *(a) $f(p) < c_3 p^2$ for $p \in (0, u)$, and*
   - *(b) $1 - f(p) < c_4(1 - p)^2$ for $p \in (v, 1)$.*

*If $f$ satisfies conditions $1, 2,$ and $3$, then $f$ exhibits linear convergence to a $t$-threshold function, meaning items at level $\Omega(\log n + k)$ of the corresponding infinite width iterative construction compute a $t$-threshold function with probability at least $1 - 2^{-k}$. If $f$ satisfies conditions $1, 2,$ and $4$, then $f$ exhibits quadratic convergence to a $t$-threshold function, meaning items at level $\Omega(\log n + \log k)$ of the corresponding infinite width iterative construction compute a $t$-threshold function with probability at least $1 - 2^{-k}$.*

We now prove that the construction described in Theorem 3 converges linearly to a $t$-threshold function.

**Proof** [of Thm. 3.] Let $f_R$ be the polynomial that describes the iterative construction in which $T_1$ and $T_2$ are selected with probability $t$ and $1-t$ respectively. Since, $f_{T_1}(p) = 2p^2 - p^3$ and $f_{T_2}(p) = p + p^2 - p^3$,

$$f_R(p) = tf_{T_1}(p) + (1-t)f_{T_2}(p) = (1-t)p + (1+t)p^2 - p^3.$$

Since $f_R(p) - p = p(1-p)(p-t)$, the fixed points of $f_R$ are $0, t$, and $1$. We claim that $f_R$ exhibits linear convergence to a $t$-threshold function.

Let $p$ be the probability that an input item fires. It suffices to consider the case when $p \le t - 1/n$. By Corollary 14, convergence to 1 for $p \ge t + \frac{1}{n}$ follows from the complementary construction.

First we show that the probability an item at level $\Omega(\log n)$ fires is less than $\frac{t}{2}$. By definition $p - f(p) = p(1-p)(t-p)$. Observe that for $t/2 < p \le t - 1/n$

$$\frac{t - f(p)}{t-p} = 1 + \frac{p - f(p)}{t-p} = 1 + p(1-t) \ge 1 + \frac{t(1-t)}{2}.$$

It follows that for all $\ell$ either $f^{(\ell)}(p) < \frac{t}{2}$ or

$$t - f^{(l)}(p) \ge \left(1 + \frac{t(1-t)}{2}\right)^l (t-p) \ge \left(1 + \frac{t(1-t)}{2}\right)^l \frac{1}{n}.$$

For $l = \log_{1+\frac{t(1-t)}{2}} \frac{tn}{2}$, $f^l(p) < \frac{t}{2}$.

Next, we show that at $\Omega(k)$ additional levels, the probability an items fires is less than $2^{-k}$. For $p < \frac{t}{2}$,

$$f(p) = p(1-p)(p-t) + p = p(1 - (1-p)(t-p)) \le p\left(1 - \left(1 - \frac{t}{2}\right)\frac{t}{2}\right).$$

It follows

$$f^{(l)}(p) < \left(1 - \left(1 - \frac{t}{2}\right)\frac{t}{2}\right)^l p < \left(1 - \left(1 - \frac{t}{2}\right)\frac{t}{2}\right)^l \frac{t}{2}.$$

Thus, for $l = \log_{(1-(1-\frac{t}{2})\frac{t}{2})} \frac{1}{t2^{k-1}}$, $f^l(p) < 2^{-k}$. We have shown that when the input items fire with probability $p \le t - 1/n$, items level $\Omega(k + \log n)$ will evaluate to 1 with probability less than $2^{-k}$. ■

### 3.1. Quadratic convergence for arbitrary thresholds.

In this section we show that as $t$ approaches 0 or 1, increasingly large building blocks trees are needed to construct an iterative tree that converges quadratically to a $t$-threshold function. Further, we give a construction that exhibits quadratic convergence for arbitrary thresholds near 0 and 1. We begin by proving Theorem 6, which can also be restated as follows: Let $f$ be an achievable polynomial with fixed points $0$, $t$, and $1$ that exhibits quadratic convergence to a $t$-threshold function. Then, $f$ has degree at least $\frac{1}{\sqrt{2s}}$ where $s = \min\{t, 1-t\}$.

**Proof** [of Thm. 6.] Let $f$ be a degree $d$ achievable polynomial with fixed points $0$, $t$, and $1$ that exhibits quadratic convergence. Then for $\varepsilon$ sufficiently small, $f(\varepsilon) = O(\varepsilon^2)$, which implies $a_1 = 0$. For $x < \frac{1}{2d}$, by Lemma 12 we have

$$f(x) = a_2x^2 + a_3x^2 + \cdots + a_dx^d \leq d^2x^2 + d^3x^3 + \ldots d^dx^d < d^2x^2\left(\frac{1}{1-dx}\right) < 2d^2x^2.$$

Since $t$ is a fixed point of $f$, $f(t) = t$. Thus, $t < 2d^2t^2$. It follows that $d > \frac{1}{\sqrt{2t}}$. By Lemma 13, if there exists an achievable polynomial with fixed point $t$, then there also exists a complementary achievable polynomial with fixed point $1-t$. Thus, $d > \frac{1}{\sqrt{2(1-t)}}$. ∎

We now prove that a nearly matching iterative construction exists. To achieve quadratic convergence to thresholds near 0 or 1, we average trees of the form $A_k$ and $A_{k+1}$ or $B_k$ and $B_{k+1}$ respectively.

**Proof** [of Thm. 7.] By Corollary 14, it suffices to prove the theorem for $1 - \phi \leq t < 1$. The complement of a construction that achieves quadratic convergence to a $t$-threshold function yields quadratic convergence for to a $(1-t)$-threshold function. By Lemma 17, there exists $k$ and $\alpha$ such that $f = \alpha f_{B_k} + (1-\alpha)f_{B_{k+1}}$has fixed point $t$. Moreover, $\frac{t-f(p)}{t-p} \geq \left(1 + \frac{p(1-p)}{t}\right)$.

We apply Lemma 18 to prove that $f$ converges to a $t$-threshold function. Let $p$ be the probability an input item is on. First suppose that $p \leq t - \frac{1}{n}$. We show linear divergence away from $t$. For any constant $0 < u < t$, and $u \leq p \leq t - \frac{1}{n}$ by Lemma 17 we have

$$t - f(p) \geq (t-p)\left(1 + \frac{p(1-p)}{t}\right) \geq (t-p)\left(1 + \frac{u(1-t)}{t}\right).$$

Thus, $c_1 = 1 + \frac{u(1-t)}{t}$ is a valid choice for $c_1$ in Lemma 18.

Next, we claim that $u = 1 - \frac{1}{k-1}$ is a valid starting point for quadratic convergence towards 0. We write $f(p) = p^2(\alpha d_k(p) + (1-\alpha)d_{k+1}(p))$ where $d_k(p) = 2p^{k-2} - p^{2k-2}$. Let $d(p) = \alpha d_k(p) + (1-\alpha)d_{k+1}(p)$. Note that $d(p)$ is increasing on the interval $(0, u)$ since each $d_k$ increases on this interval. For $p < u$,

$$\frac{2k-4}{2k-2} = u > u^k > p^k.$$

It follows that $d_k'(p) = p^{k-3}((2k-4) - (2k-2)p^k) > 0$. Thus, $d_k$ is increasing on the interval $(0, u)$. Thus, $c_3 = d(u)$ is a valid choice for $c_3$ in Lemma 18.

It remains to show that for $p \geq t + \frac{1}{n}$ we observe linear divergence from $t$ then quadratic convergence to 1. We show linear divergence away from $t$. For any constant $t < v < 1$, and $t + \frac{1}{n} \leq p \leq 1$ by Lemma 17 we have

$$f(p) - t \geq (p-t)\left(1 + \frac{p(1-p)}{t}\right) \geq (p-t)\left(1 + \frac{t(1-v)}{t}\right).$$

Thus, $c_2 = 1 + \frac{t(1-v)}{t}$ is a valid choice for $c_2$ in Lemma 18.

We claim that $v > 1 - \frac{1}{8(k+1)^2}$ is a valid starting point for quadratic convergence to 1. By Corollary 14, $f_{A_k}(1-p) = 1 - f_{B_k}(p)$. It follows

$$1 - f(p) = \alpha - \alpha f_{B_k}(p) + (1-\alpha) - (1-\alpha) f_{B_{k+1}}(p) = \alpha f_{A_k}(1-p) + (1-\alpha) f_{A_{k+1}}(1-p).$$

Recall from the proof of Theorem 6, $f(x) < 2dx^2$ where $d$ is the degree of $x$. Therefore,

$$f_{A_k}(1-p) < 8k^2(1-p)^2 < 8(k+1)^2(1-p)^2 \text{ and } f_{A_{k+1}}(1-p) < 8(k+1)^2(1-p)^2.$$

Since $(1-v)8(k+1)^2 < 1$, $c_4 = 8(k+1)^2$ is a valid choice for $c_4$ in Lemma 18. ■

## 4. Finite iterative constructions of threshold trees

In the above section, we analyzed the behavior of iterative trees in the limit with respect to level width. We assumed that for any input the number of items firing at level $l$ of the tree is equal to its expectation, $mf^{(l)}(p)$ where $m$ is the width of level $l$ and $p$ is the fraction of the inputs firing. In a "bottom up" construction in which the items of one level are fixed before the next level is built, we note that the chance that the number of items that fire at a given level deviates from expectation is non-trivial. In this section, we give a bound on the width of the levels required to achieve a desired degree of accuracy for a finite realization of iterative constructions.

We will use the following concentration inequality.

**Lemma 19** *(Chernoff) Let $Y_1, Y_2, \ldots Y_m$ be independent with $0 \le Y_i \le 1$ and $Y = \sum_{i=1}^{n} Y_i$. Then, for any $\delta > 0$,*

$$\Pr(Y - E(Y) \ge \delta E(Y)) \le \exp\left(\frac{-\delta^2 E(Y)}{2+\delta}\right).$$

For ease of notation, all statements in this section about the probability of $X_{i+1}$ taking some values refers to the probability of $X_{i+1}$ taking some values given $X_i$. The following lemma describes linear divergence for finite width constructions.

**Lemma 20** *Consider the construction of a $t$-threshold function in which each level $\ell$ has $m_\ell$ items and the fraction of input items firing is at least $\varepsilon$ below the threshold $t$. Let $d$ be the minimum value of $\frac{f(p)-p}{p(1-p)(p-t)}$ on the interval $[0,1]$. Then, with probability at least $1-\gamma$, the fraction of inputs firing at level $\Omega(\frac{1}{\varepsilon})$ will be less than any fixed constant $u$ when*

$$m_\ell = \frac{8\ln(\frac{1}{u(1-t)\gamma})}{d^2 u(1-t)^2 \left(1+\frac{c_1}{2}\right)^{\ell-1} \varepsilon^2}$$

*where $c_1$ is the linear divergence constant.*

**Proof** Let $X_i$ be the fraction of items firing at level $i$. Then $E(X_i) = f(X_{i-1})$. In expectation, the sequence $X_1, X_2, X_3, \ldots$ converges to 0. We will show that with probability at least $1-\gamma$, the sequence obeys the *half-progress* relation $X_{i+1} \le \frac{X_i + f(X_i)}{2}$ and therefore $X_L < u$ for $L = \Omega(\frac{1}{\varepsilon})$.

Write $f(p) - p = p(1-p)(p-t)g(p)$ where $g$ is a polynomial in $p$. Let $d$ be the minimum value obtained by $g$ on the interval $[0,1]$. First we compute probability that $X_{i+1} > \frac{X_i + f(X_i)}{2}$ given $X_i$ by applying Lemma 19. Observe

$$\begin{aligned}
\Pr\left(X_{i+1} > \frac{X_i + f(X_i)}{2}\right) &= \Pr\left(X_{i+1} - E(X_{i+1}) > \frac{X_i - f(X_i)}{2}\right) \\
&\leq \exp\left(\frac{-\left(\frac{X_i - f(X_i)}{2f(X_i)}\right)^2 mf(X_i)}{2 + \frac{X_i - f(X_i)}{2f(X_i)}}\right) \\
&= \exp\left(-\frac{(X_i(1-X_i)(X_i - t)g(X_i))^2 m}{2(X_i + 3(X_i + X_i(1-X_i)(X_i - t)g(X_i)))}\right) \\
&\leq \exp\left(-\frac{X_i(1-X_i)^2(t-X_i)^2 d^2 m}{8}\right)
\end{aligned}$$

Let $\varepsilon_i = t - X_i$ and $\alpha = \frac{u(1-t)^2 d^2}{8}$. Then for $u \leq X_i \leq t - \varepsilon$,

$$\Pr\left(X_{i+1} > \frac{X_i + f(X_i)}{2}\right) < \exp\left(-\alpha m \varepsilon_i^2\right).$$

Next we compute the probability that $i$ is the first value for which the half-progress relation is not satisfied given $X_i > u$. If the half-progress relation is satisfied meaning $X_{i+1} > \frac{X_i + f(X_i)}{2}$, then $\varepsilon_{i+1} \geq \varepsilon_i \beta$ where $\beta = 1 + \frac{u}{2}(1-t)$. It follows that if the half-progress relation is satisfied for all $j < i$, then $\varepsilon_{i+1} \geq \varepsilon \beta^i$. Thus,

$$\Pr\left(i \text{ is the first value for which } X_{i+1} > \frac{X_i + f(X_i)}{2}\right) \leq \exp\left(-\alpha m \varepsilon^2 \beta^{2i}\right).$$

By linear divergence, there exists $L = \Omega(\log(\frac{1}{\varepsilon}))$ such that if the sequence satisfies the half-progress relation for all $i < L$, then $X_L < u$. We bound the probability that this does not happen. Let $m_\ell = \frac{8 \ln(\frac{1}{u(1-t)\gamma})}{d^2 u(1-t)^2 \beta^i \varepsilon^2}$. For ease of notation, let $c = \ln \frac{1}{u(1-t)\gamma} < 1$. Observe

$$\begin{aligned}
\Pr(X_L > u) \leq \sum_{i=0}^{L} \exp\left(-\alpha m_\ell \varepsilon^2 \beta^{2i}\right) &= \sum_{i=0}^{L} \exp\left(-c\beta^i\right) \\
&\leq \sum_{i=0}^{L} \exp\left(-c(1 + iu(1-t))\right) \\
&< \exp(-c) \sum_{i=0}^{L} e^{-iu(1-t))} \\
&< \frac{\exp(-c)}{1 - \exp(-u(1-t))} \\
&< \frac{\exp(-c)}{u(1-t)} = \gamma.
\end{aligned}$$

■

**Theorem 21** *Consider the construction of a $t$-threshold function with linear convergence given in Theorem 3 in which each level $\ell$ has $m_\ell$ items and the fraction of input items firing is at least $\varepsilon$ from the threshold $t$. Then, with probability at least $1-\gamma$, items at level $\Omega(\log \frac{1}{\gamma} + \log \frac{1}{\varepsilon})$ will accurately compute the threshold function for $m = \Omega\left(\ln(\frac{1}{\gamma})(\frac{1}{\gamma} + \frac{1}{\varepsilon^2})\right)$.*

**Proof** Let $X_i$ be the fraction of items firing at level $i$. Then $E(X_i) = f(X_{i-1})$. By Corollary 14, it suffices to consider the case when the fraction of inputs firing is less than $t - \varepsilon$. As proved in Theorem 3, the sequence $X_1, X_2, X_3, \ldots$ convergences to 0 if each $X_i$ achieves its expectation. We will show that with probability at least $1 - \frac{\gamma}{2}$, the sequence drops below $\frac{\gamma}{2}$. First we apply Lemma 20. Recall that the polynomial corresponding to this construction is $f(p) = p + p(1-p)(p-t)$ and therefore $d$ in the statement of Lemma 20 is 1. Let $u$ be a constant $0 < u < t$, $m \geq \frac{8\ln(\frac{4}{u(1-t)\gamma})}{u(1-t)^2\varepsilon^2}$ and $L = \Omega(\frac{1}{\varepsilon})$. Thus, $X_L < u$ with probability at least $1 - \frac{\gamma}{4}$.

Next we show that given $X_L < u$ the probability that the sequence continues to obey the half-progress relation (as defined in Lemma 20) and drops below $\frac{\gamma}{2}$ is at least $1 - \frac{\gamma}{4}$. Let $\alpha = \frac{(1-u)^2(t-u)^2}{8}$. Given a fixed value $X_i < u$,

$$\mathsf{Pr}\left(X_{i+1} > \frac{X_i + f(X_i)}{2}\right) < \mathsf{exp}\left(-\alpha m X_i^2\right).$$

We compute the probability that $N + i$ is the first value for which the half-progress relation is not satisfied given $X_L < u$. If $X_i < u$ and the half-progress relation is satisfied at $i$ then $X_{i+1} \leq X_i(1-\beta)$ where $\beta = \frac{1}{2}(1-u)(t-u)$. It follows that if the half-progress relation is satisfied for all $j < i$, then $X_{N+i} \leq (1-\beta)^i u$. Let $L' = \frac{4}{(1-u)(t-u)} \log_2\left(\frac{2u}{\gamma}\right)$. If for all $L \leq i \leq L + L'$, the half-progress relation is satisfied then $X_{L+L'} < u(1-\beta)^{L'} < \frac{\gamma}{2}$. We bound the probability that this does not happen. Let $m \geq \frac{16\ln\left(\frac{16}{(1-u)(t-u)\gamma}\right)}{(1-u)^2(t-u)^2\gamma}$. For ease of notation, let $c = \ln\left(\frac{8}{\beta\gamma}\right)$. Observe

$$\begin{aligned}
\mathsf{Pr}\left(X_{L+L'} > \frac{\gamma}{2}\right) &\leq \sum_{i=0}^{L'} \mathsf{exp}\left(-mX_i\alpha\right) \\
&= \sum_{i=1}^{L'} \mathsf{exp}\left(-\frac{2cX_i}{\gamma}\right) \\
&\leq \sum_{i=0}^{L'} \mathsf{exp}\left(-c(1-\beta)^{-(L'-i)}\right) \\
&= \sum_{i=0}^{L'} \mathsf{exp}\left(-c(1-\beta)^{i}\right) \\
&\leq \sum_{i=0}^{\beta L'} \frac{1}{\beta}\mathsf{exp}\left(-ce^{i}\right) \\
&\leq \frac{2\mathsf{exp}\left(-c\right)}{\beta} \\
&= \frac{\gamma}{4}.
\end{aligned}$$

Therefore, with probability at least $1 - \frac{\gamma}{2}$, items at level $\Omega\left(\log\frac{1}{\gamma} + \log\frac{1}{\varepsilon}\right)$ of an iterative construction with width $m$ fire with probability at most $\frac{\gamma}{2}$ for $m = \Omega\left(\ln(\frac{1}{\gamma})(\frac{1}{\gamma} + \frac{1}{\varepsilon^2})\right)$. Thus, the iterative construction accurately computes the threshold function with probability at least $(1-\frac{\gamma}{2})^2 > 1-\gamma$. ■

## 5. Learning

So far we have studied the realizability of thresholds via neurally plausible simple iterative constructions. These constructions were based on prior knowledge of the target threshold. Here we study the learnability of thresholds from examples. It is important that the learning algorithm should be neurally plausible and not overly specialized to the learning task. We believe the simple results presented here are suggestive of considerably richer possibilities.

We begin with a one-shot learning algorithm. We show that given a single example of a string $X \in \{0,1\}^n$ with $\|X\|_1 = tn$, we can build an iterative tree that computes a $t$-threshold function with high probability. Let $T_1$ and $T_2$ be the building block trees in the construction given in Theorem 3. The simple LearnThreshold algorithm, described below, has the guarantee stated in Theorem 11, which follows from Theorem 3.

**LearnThreshold($L, m, X$):**
*Input:* Levels parameter $L$, a string $X \in \{0,1\}^n$ such that $\|X\|_1 = tn$, width parameter $m$.
*Output:* A finite realization of iterative tree with width $m$.

For each level $j$ from 1 to $L$, apply the following iteration $m$ times:
(level 0 consists of the input items $X$)

1. Pick a random input item $i$.
2. If $X_i = 1$ then let $T = T_1$, else let $T = T_2$.
3. Pick 3 items uniformly at random from the previous level.
4. Build $T$ with these items as leaves.

## 6. Discussion

We have seen that very simple, distributed algorithms requiring minimal global coordination and control can lead to stable and efficient constructions of important classes of functions. Our work raises several interesting questions.

1. What are the ways in which threshold functions are applied in cognition? Object recognition is one application of threshold functions in cognition. For instance, suppose we have items representing features such as “trunk,” “grey,” “wrinkled skin,” and “big ears,” and an item representing our concept of an “elephant.” If a certain threshold of items representing the features we associate with an elephant fire, then the “elephant” item will fire. This structure

lends itself to a hierarchical organization of concepts that is consistent with the fact that as we learn, we build on our existing set of knowledge. For example, when a toddler learns to identify an elephant, he does not need to re-learn how to identify an ear. The item representing "ear" already exists and will fire as a result of some threshold function created when the toddler learned to identify ears. Now the item representing "ear" may be used as an input as the toddler learns to identify elephants and other animals.

2. The hierarchical structure of iterative trees makes them a promising representation for learning in a setting where knowledge is built on existing knowledge. An ideal algorithm for this setting should successfully learn from only a few examples and non-examples. Further, the algorithm must be neurally plausible. In addition to being highly distributed, and requiring little synchrony and global control, the algorithm should only use information from items that are currently firing. At any moment in the brain the majority of neurons are not firing. In our current model there is no way to distinguish between relevant but not currently firing items from any item that is not currently firing. To remedy this, it might be beneficial introduce a third "predictive state" in which items that are predicted to be relevant are primed to fire, e.g. as done by Papadimitriou and Vempala (2015a).

   The one-shot learning algorithm described in Section 5 relies on sampling items not currently firing and therefore fails this last measure of neural plausibility. However, we believe the simple result of the one-shot learning algorithm is suggestive of richer possibilities. The following two items outline more specific learning tasks pertaining to iterative constructions.

3. What is an interesting model and neurally plausible algorithm for learning threshold functions of $k$ relevant input items? In this scenario, the input is a set of sparse binary strings of length $n$ representing examples in which at least $tk$ of $k$ relevant items are firing. The output is an iterative tree that computes a $t$-threshold function on the $k$ relevant items. We can formulate the previously described example of learning to identify an elephant as an instance of this problem. Each time the toddler sees an example of an elephant, many features associated with elephant will fire in addition to some features that are not associated with elephants. There may also be features associated with an elephant that are not present in this example and therefore not firing. A learning algorithm must rely on information about the items that are currently firing to learn both the set of relevant items and a threshold function on this set of items.

4. More generally, given a set of inputs and outputs, can we devise an algorithm to learn the distribution on building block trees for which the corresponding iterative tree would produce the outputs?

5. To what extent can general linear threshold functions with general weights be constructed/learned by cortical algorithms?

6. A concrete question is whether the construction of Theorem 7 is optimal, similar to the optimality of the constructions in Theorem 5.

7. A simple way to include non monotone Boolean functions with the same constructions as we study here, would be to have input items together with their negations (as in e.g., (Savicky, 1990)). What functions can be realized this way, using a distribution on a small set of fixed-size trees?

**Acknowledgements.** We thank an anonymous referee for helpful comments. This work was supported in part by NSF awards CCF-1217793, EAGER-1415498 and EAGER-1555447.

## References

Rosa I. Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine Learning*, 63(2):161–182, 2006. doi: 10.1007/s10994-006-6265-7. URL http://dx.doi.org/10.1007/s10994-006-6265-7.

Rosa I. Arriaga, David Rutter, Maya Cakmak, and Santosh S. Vempala. Visual categorization with random projection. *Neural Computation*, 27(10):2132–2147, 2015. doi: 10.1162/NECO_a_00769. URL http://dx.doi.org/10.1162/NECO_a_00769.

Ravi B. Boppana. Amplification of probabilistic boolean formulas. In *Foundations of Computer Science, 1985., 26th Annual Symposium on*, pages 20–29, Oct 1985. doi: 10.1109/SFCS.1985.5.

Alex Brodsky and Nicholas Pippenger. The boolean functions computed by random boolean formulas or how to grow the right function. *Random Structures & Algorithms*, 27(4):490–519, 2005. ISSN 1098-2418. doi: 10.1002/rsa.20095. URL http://dx.doi.org/10.1002/rsa.20095.

M. Dubiner and U. Zwick. Amplification and percolation [probabilistic boolean functions]. In *Foundations of Computer Science, 1992. Proceedings., 33rd Annual Symposium on*, pages 258–267, Oct 1992. doi: 10.1109/SFCS.1992.267766.

Vitaly Feldman and Leslie G. Valiant. Experience-induced neural circuits that achieve high capacity. *Neural Computation*, 21(10):2715–2754, 2009. doi: 10.1162/neco.2009.08-08-851.

Hervé Fournier, Danièle Gardy, and Antoine Genitrini. Balanced and/or trees and linear threshold functions. In *Proceedings of the Meeting on Analytic Algorithmics and Combinatorics*, ANALCO '09, pages 51–57, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics. URL http://dl.acm.org/citation.cfm?id=2791158.2791166.

Joel Friedman. Constructing O($nlogn$) size monotone formulae for the $k^{th}$ threshold function of $n$ boolean variables. *SIAM Journal on Computing*, 15(3):641–654, 1986.

Sally A Goldman, Michael J Kearns, and Robert E Schapire. Exact identification of read-once formulas using fixed points of amplification functions. *SIAM Journal on Computing*, 22(4):705–726, 1993.

Shlomo Hoory, Avner Magen, and Toniann Pitassi. Monotone circuits for the majority function. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 410–425. Springer, 2006.

Michael Luby, Michael Mitzenmacher, and Mohammad Amin Shokrollahi. Analysis of random processes via and-or tree evaluation. In *SODA*, volume 98, pages 364–373, 1998.

Edward F Moore and Claude E Shannon. Reliable circuits using less reliable relays. *Journal of the Franklin Institute*, 262(3):191–208, 1956.

Christos H. Papadimitriou and Santosh S. Vempala. Cortical learning via prediction. In *Proc. of COLT*, 2015a.

Christos H. Papadimitriou and Santosh S. Vempala. Cortical computation. In *Proc. of PODC*, 2015b.

E. Rosch. Principles of categorization. In Eleanor Rosch and Barbara Lloyd, editors, *Cognition and Categorization*. Lawrence Elbaum Associates, 1978.

Eleanor Rosch, Carolyn B. Mervis, Wayne D. Gray, David M. Johnson, and Penny Boyes-braem. Basic objects in natural categories. *COGNITIVE PSYCHOLOGY*, 8:382–439, 1976.

Petr Savicky. Boolean functions represented by random formulas. *Commentationes Mathematicae Universitatis Carolinae*, 28(2):397–398, 1987.

Petr Savicky. Random boolean formulas representing any boolean function with asymptotically equal probability. *Discrete Mathematics*, 83(1):95 – 103, 1990. ISSN 0012-365X. doi: http://dx.doi.org/10.1016/0012-365X(90)90223-5. URL http://www.sciencedirect.com/science/article/pii/0012365X90902235.

Rocco A. Servedio. Monotone boolean formulas can approximate monotone linear threshold functions. *Discrete Applied Mathematics*, 142(13):181 – 187, 2004. ISSN 0166-218X. doi: http://dx.doi.org/10.1016/j.dam.2004.02.003. URL http://www.sciencedirect.com/science/article/pii/S0166218X04000174. Boolean and Pseudo-Boolean Functions.

Leslie G. Valiant. Short monotone formulae for the majority function. *Journal of Algorithms*, 5(3), 1984.

Leslie G. Valiant. *Circuits of the mind*. Oxford University Press, 1994. ISBN 978-0-19-508926-4.

Leslie G. Valiant. A neuroidal architecture for cognitive computation. *J. ACM*, 47(5):854–882, 2000. doi: 10.1145/355483.355486.

Leslie G. Valiant. Memorization and association on a realistic neural model. *Neural Computation*, 17(3):527–555, 2005. doi: 10.1162/0899766053019890.