Lower bounds over Boolean inputs for deep neural networks with ReLU gates.

Anirbit Mukherjee[*]   Amitabh Basu[†]

Abstract

Motivated by the resurgence of neural networks in being able to solve complex learning tasks we undertake a study of high depth networks using ReLU gates which implement the function $x \mapsto \max\{0, x\}$. We try to understand the role of depth in such neural networks by showing size lowerbounds against such network architectures in parameter regimes hitherto unexplored. In particular we show the following two main results about neural nets computing Boolean functions,

- We use the method of random restrictions to show almost linear, $\Omega(\epsilon^{2(1-\delta)} n^{1-\delta})$, lower bound for completely weight unrestricted LTF-of-ReLU circuits to match the Andreev function on at least $\frac{1}{2} + \epsilon$ fraction of the inputs for $\epsilon > \sqrt{2 \frac{\log^{\frac{2}{2-\delta}}(n)}{n}}$ for any $\delta \in (0, \frac{1}{2})$

- We use the method of sign-rank to show exponential in dimension (n) lower bounds for ReLU circuits ending in a LTF gate and of depths upto $O(n^\xi)$ with $\xi < \frac{1}{8}$ with some restrictions on the weights in the bottom most layer. All other weights in these circuits are kept unrestricted. This in turns also implies the same lowerbounds for LTF circuits with the same architecture and the same weight restrictions on their bottom most layer.

Along the way we also show that there exists a $\mathbb{R}^n \to \mathbb{R}$ Sum-of-ReLU-of-ReLU function which Sum-of-ReLU neural nets can never represent no matter how large they are allowed to be.

1 Introduction

There has been a recent surge of activity in using neural networks for complex artificial intelligence tasks (like this very recent spectacular demonstration [34] of the power of neural nets). This has rekindled interest in understanding neural networks from a complexity theory perspective. A myriad of hard mathematical questions have surfaced in the attempts to rigorously explain the power of neural networks and a comprehensive overview of these can be found in this recent three part series of articles from The Center for Brains, Minds and Machines (CBMM), [26, 25, 42].There is a rich literature investigating the complexity of the function classes represented by neural networks with various kinds of gates (or "activation functions" which is the more common parlance in machine learning). Many papers, a canonical example being the classic paper by Maass [23], establish complexity results for the entire class of functions represented by circuits where the gates can come

---

[*]Department of Applied Mathematics and Statistics, Johns Hopkins University, Email: `amukhe14@jhu.edu`

[†]Department of Applied Mathematics and Statistics, Johns Hopkins University, Email: `basu.amitabh@jhu.edu`

from a very general family. This is complemented by papers that study a very specific family of gates such as the sigmoid gate or the LTF gate [16], [35], [31] [19], [3], [32], [28], [4]. Many associated results can also be found in these reviews [20, 27]. Recent circuit complexity results in [18], [38], [6], [17] stand out as significant improvements over known lower (and upper) bounds on circuit complexity with threshold gates. The results of Maass [23] also show that very general families of neural networks can be converted into circuits with only LTF gates with at most a constant factor blow up in depth and polynomial blow up in size of the circuits.

In the last 5 years or so, a particular family of gates called the *Rectified Linear Unit (ReLU)* gates have been reported to have significant advantages over more traditional gates in practical applications of neural networks. Such a gate with n real inputs computes the following output,

$$\mathbb{R}^n \to \mathbb{R} \tag{1}$$

$$\mathbf{x} \mapsto \max\{0, b + \langle \mathbf{w}, \mathbf{x} \rangle\} \tag{2}$$

where $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are fixed parameters associated with the gate (b is called the bias of the gate). In comparison, the ± 1 valued LTF gate mentioned above computes (for the same weights as above) the function, $(2\mathbf{1}_{(b+\langle \mathbf{w}, \mathbf{x} \rangle \geq 0)} - 1)$ where $\mathbf{1}_{(b+\langle \mathbf{w}, \mathbf{x} \rangle \geq 0)}$ is the $0/1$ indicator function for the stated halfspace condition.

Some of the prior results which apply to general gates, such as the ones in [23], also apply to ReLU gates, because those results apply to gates that compute a piecewise polynomial function (ReLU is a piecewise linear function with only two pieces). However, as witnessed by results on LTF gates, one can usually make much stronger claims about specific classes of gates. To the best of our knowledge, no prior results have been obtained for ReLU gates from the perspective of *Boolean* complexity theory, i.e., the study of such circuits when restricted to Boolean inputs. The main focus of this work is to study circuits computing Boolean functions mapping $\{-1, 1\}^m \to \{-1, 1\}$ which use ReLU gates in their intermediate layers, and have an LTF gate at the output node (to ensure that the output is in $\{-1, 1\}$). We remark that using an LTF gate at the output node while allowing more general analog gates in the intermediate nodes is a standard practice when studying the Boolean complexity of analog gates (see, for example, [23]).

Although we are not aware of an analysis of lower bounds for ReLU circuits when applied to only Boolean inputs, there has been recent work on the analysis of such circuits when viewed as a function from $\mathbb{R}^n$ to $\mathbb{R}$ (i.e., allowing real inputs and output). From [8] and [7] (with restrictions on the domain and the weights) we know of (super-)exponential lowerbounds on the size of Sum-of-ReLU circuits for certain easy Sum-of-ReLU-of-ReLU functions . Depth v/s size tradeoffs for such circuits have recently also been studied in [39, 12, 21, 41, 30] and in a recent paper [2] by the current authors. To the best of our knowledge no lowerbounds scaling exponentially with the dimension are known for analog deep neural networks of depths more than 2.

In what follows, the *depth* of a circuit will be the length of the longest path from the output node to an input variable, and the *size* of a circuit will be the total number of gates in the circuit. We will also use the notation *Sum-of-ReLU* to refer to circuits whose inputs feed into a single layer of ReLU gates, whose outputs are combined into a weighted sum to give the final output. Similarly,

Sum-of-ReLU-of-ReLU denotes the circuit with depth 3, where the output node is a simple weighted sum, and the intermediate gates are all ReLU gates in the two "hidden" layers. We analogously define Sum-of-LTF, LTF-of-LTF, LTF-of-ReLU, LTF-of-LTF-of-LTF, LTF-of-ReLU-of-ReLU and so on. We will also use the notation LTF-of-$(\text{ReLU})^k$ for a circuit of the form LTF-of-ReLU-of-RELU-...-ReLU with $k \geq 1$ levels of ReLU gates.

2 Statement and discussion of results

Boolean v/s real inputs. We begin our study with the following observation which shows that ReLU circuits have markedly different behaviour when the inputs are restricted to be Boolean, as opposed to arbitrary real inputs. Since AND and OR gates can both be implemented by ReLU gates, it follows that *any* Boolean function can be implemented by a ReLU-of-ReLU circuit. In fact, it is not hard to show something slightly stronger:

Lemma 2.1. Any function $f : \{-1, 1\}^n \to \mathbb{R}$ can be implemented by a Sum-of-ReLU circuit using at most $\min\{2^n, \sum_{\hat{f}(S) \neq 0} |S|\}$ number of ReLU gates, where $\hat{f}(S)$ denotes the Fourier coefficient of f for the set $S \subseteq \{1, \ldots, n\}$.

The Lemma follows by observing that the indicator functions of each vertex of the Boolean hypercube $\{-1, 1\}^n$ can be implemented by a single ReLU gate, and the parity function on k variables can be implemented by k ReLU gates (see Appendix C). Thus, if one does not restrict the size of the circuit, then Sum-of-ReLU circuits can represent any pseudo-Boolean function. In contrast, we will now show that if one allows real inputs, then there exist functions with just 2 inputs (i.e., $n = 2$) which cannot be represented by any Sum-of-ReLU circuit, no matter how large.

Proposition 2.2. The function $\max\{0, x_1, x_2\}$ cannot be computed by any Sum-of-ReLU circuit, no matter how many ReLU gates are used. It can be computed by a Sum-of-ReLU-of-ReLU circuit.

The first part of the above proposition (the impossibility result) is proved in Appendix A. The second part follows from Corollary 2.2 of a previous paper by the authors [2], which states that any $\mathbb{R}^n \to \mathbb{R}$ function that can be implemented by a circuit of ReLU gates, can always be implemented with at most $\lceil \log(n+1) \rceil$ layers of ReLU gates (with a weighted Sum to give the final output).

Restricting to Boolean inputs. From this point on, we will focus entirely on the situation where the inputs to the circuits are restricted to $\{-1, 1\}$. One motivation behind our results is the desire to understand the strength of the ReLU gates vis-a-vis LTF gates. It is not hard to see that any circuit with LTF gates can be simulated by a circuit with ReLU gates with at most a constant blow-up in size (because a single LTF gate can be simulated by 2 ReLU gates when the inputs are a discrete set – see Appendix B). The question is whether ReLU gates can do significantly better than LTF gates in terms of depth and/or size.

A quick observation is that Sum-of-ReLU circuits can be linearly (in the dimension n) smaller than Sum-of-LTF circuits. More precisely,

3

Proposition 2.3. The function $f : \{-1,1\}^n \to \mathbb{R}$ given by $f(x) = \sum_{i=1}^{n} 2^i \left(\frac{1+x_i}{2}\right)$ can be implemented by a Sum-of-ReLU circuit with 2 ReLU gates, and any Sum-of-LTF that implements f needs $\Omega(n)$ gates.

The above result follows from the following two facts: 1) any linear function is implementable by 2 ReLU gates, and 2) any Sum-of-LTF circuit with w LTF gates gives a piecewise constant function that takes at most 2^w different values. Since f takes 2^n different values (it evaluates every vertex of the Boolean hypercube to the corresponding natural number expressed in binary), we need $w \geq n$ gates.

In the context of these preliminary results, we now state our main contributions. For the next result we recall the definition of the Andreev function [1] which has previously many times been used to prove computational lower bounds [24, 15, 14].

Definition 1 (Andreev's function). The Andreev's function is the following mapping,

$$
A_n : \{0,1\}^{\lfloor \frac{n}{2} \rfloor} \times \{0,1\}^{\lfloor \log(\frac{n}{2}) \rfloor \times \lfloor \frac{n}{2\lfloor \log(\frac{n}{2}) \rfloor} \rfloor} \longrightarrow \{0,1\}
$$
$$
(\mathbf{x}, [a_{ij}]) \longmapsto x_{\text{bin}\left(\{(\sum_{j=1}^{\lfloor \frac{n}{2\lfloor \log(\frac{n}{2}) \rfloor} \rfloor} a_{ij}) \mod 2\}_{i=1,2,..,\lfloor \log(\frac{n}{2}) \rfloor} \right)}
$$

where "bin" is the function that gives the decimal number that can be represented by its input bit string.

We are particularly inspired by the most recent use of the Andreev function by Kane and Williams [18] to get the first super linear lower bounds for approximating it using LTF-of-LTF circuits. We will give an almost linear lower bound on the size of LTF-of-ReLU circuits approximating this Andreev function with no restriction on the weights $\mathbf{w}, b$ for each gate.

Theorem 2.4. For any $\delta \in (0, \frac{1}{2})$, there exists $N(\delta) \in \mathbb{N}$ such that for all $n \geq N(\delta)$ and $\epsilon > \sqrt{\frac{2 \log^{\frac{2}{2-\delta}}(n)}{n}}$, any LFT-of-ReLU circuit on n bits that matches the Andreev function on $n-$bits for at least $1/2 + \epsilon$ fraction of the inputs, has size $\Omega(\epsilon^{2(1-\delta)} n^{1-\delta})$.

It is well known that proving lower bounds without restrictions on the weights is much more challenging even in the context of LTF circuits. In fact, the recent results in [18] are the first superlinear lower bounds for LTF circuits with no restrictions on the weights. With restrictions on some or all the weights, e.g., assuming $poly(n)$ bounds on the weights (typically termed the "small weight asssumption") in certain layers, exponential lower bounds have been established for LTF circuits [11, 16, 32, 33]. Our next results are of this flavor: under certain kinds of weight restrictions, we prove exponential size lower bounds on the size of LTF-of-(ReLU)$^{d-1}$ circuits. *One thing to note is that our weight restrictions are assumed only on the bottom layer (closest to the input). The other layers can have gates with unbounded weights.* Nevertheless, our weight restrictions are somewhat unconventional.

4

Definition 2. [Weight restriction condition] Let $m \in \mathbb{N}$ and σ be any permutation of $\{1, \ldots, 2^m\}$. Let us also consider an arbitrary sequencing $\{\mathbf{x}^1, \ldots, \mathbf{x}^{2^m}\}$ of the vertices of the hypercube $\{-1, 1\}^m$. Define the polyhedral cone

$$P_{m,\sigma} := \{\mathbf{a} \in \mathbb{R}^m : \langle \mathbf{a}, \mathbf{x}^{\sigma(1)} \rangle \leq \langle \mathbf{a}, \mathbf{x}^{\sigma(2)} \rangle \leq \ldots \langle \mathbf{a}, \mathbf{x}^{\sigma(2^m)} \rangle\}.$$

In words, $P_{m,\sigma}$ is the set of all linear objectives that order the vertices of the m-dimensional hypercube in the order specified by σ. We will impose the condition that there exists a σ such that for each ReLU gate in the bottom layer, the vector $\mathbf{w} \in P_{m,\sigma}$ ($\mathbf{w}$ as defined in (1)) and all weights are integers with magnitude bounded by some $W > 0$.

We will prove our lower bounds against the function proposed by Arkadev and Nikhil in [5],

$$g : OMB_n^0 \circ OR_{n^{\frac{1}{3}} - \log n} \circ XOR_2 : \{-1, 1\}^{2(n^{\frac{4}{3}} - n \log n)} \rightarrow \{-1, 1\} \tag{3}$$

which we will refer to as the Arkadev-Nikhil function in the remainder of the paper. Here OMB is the ODD-MAX-BIT function which is a ± 1 threshold gate which evaluates to -1 on say a $n-$bit input $\mathbf{x}$ if $\sum_{i=1}^{n} (-1)^{i+1} 2^i (1 + x_i) \geq \frac{1}{2}$. We show the following exponential lowerbound against this function,

Theorem 2.5. Let $m, d, W \in \mathbb{N}$. Any depth d LTF-of-(ReLU)$^{d-1}$ circuits on $2m$ bits such that the weights in the bottom layer are restricted as per Definition 2 that implements the Arkadev-Nikhil function on $2m$ bits will require a circuit size of

$$\Omega \left((d-1) \frac{2^{\frac{m^{\frac{1}{8}}}{d-1}}}{(mW)^{\frac{1}{d-1}}} \right).$$

Consequently, one obtains the same size lower bounds for circuits with only LTF gates of depth d.

Note that this is an exponential in dimension size lowerbound for even super-polynomially growing bottom layer weights (and additional constraints as per Definition 2) and upto depths scaling as $d = O(m^\xi)$ with $\xi < \frac{1}{8}$.

We note that the Arkadev-Nikhil function can be represented by an $O(m)$ size LTF-of-LTF circuit with no restrictions on weights (see Theorem 2.6 below). In light of this fact, Theorem 2.5 is somewhat surprising as it shows that for the purpose of representing Boolean functions a deep ReLU circuit (ending in a LTF) gate can get exponentially weakened when just its bottom layer weights are restricted as per Definition 2, even if the integers are allowed to be super-polynomially large. Moreover, the lower bounds also hold of LTF circuits of arbitrary depth d, under the same weight restrictions on the bottom layer. We are unaware of any exponential lower bounds on LTF circuits of arbitrary depth under any kind of weight restrictions.

We will use the method of sign-rank to obtain the exponential lowerbounds in Theorems 2.5. The sign-rank of a real matrix A with all non-zero entries is the least rank of a matrix B of the same

dimension with all non-zero entries such that for each entry (i, j), $sign(B_{ij}) = sign(A_{ij})$. For a Boolean function f mapping, $f : \{-1, 1\}^m \times \{-1, 1\}^m \to \{-1, 1\}$ one defines the "sign-rank of f" as the sign-rank of the $2^m \times 2^m$ dimensional matrix $[f(\mathbf{x}, \mathbf{y})]_{\mathbf{x}, \mathbf{y} \in \{-1, 1\}^m}$. This notion of a sign-rank has been used to great effect in diverse fields from communication complexity to circuit complexity to learning theory. Explicit matrices with a high sign-rank were not known till the breakthrough work by Forster, [9]. Forster et. al. showed elegant use of this complexity measure to show exponential lowerbounds against LTF-of-MAJ circuits in [10]. Lot of the previous literature about sign-rank has been reviewed in the book by Satya Lokam [22]. Most recently the following result was obtained by Arkadev and Nikhil in [5] leading to a proof of strict containment of LTF-of-MAJ in LTF-of-LTF.

Theorem 2.6. [**Theorem** 4.2 **and Corollary** 1.2 **in** **[5]**]
The Akradev-Nikhil function g in equation 3 can be represented by a linear sized LTF-of-LTF circuit and sign-rank$(g) \geq \frac{2^{n^{\frac{1}{3}} - 2\log n}}{16}$

We will prove our theorem by showing a small upper bound on the sign-rank of LTF-of-$(ReLU)^{d-1}$ circuits which have their bottom most layer's weight restricted in the said way.

3 Lower bounds for LTF-of-ReLU against the Andreev function (Proof of Theorem 2.4)

We will use the classic "method of random restrictions" [37, 36, 13, 40, 29] to show a lowerbound for weight unrestricted LTF-of-ReLU circuits for representing the Andreev function. The basic philosophy of this method is to take any arbitrary LTF-of-ReLU circuit which supposedly matches the Andreev function on a large fraction of the inputs and to randomly fix the values on some of its input coordinates and also do the same fixing on the same coordinates of the input to the Andreev function. Then we show that upon doing this restriction the Andreev function collapses to an arbitrary Boolean function on the remaining inputs (what it collapses to depends on what values were fixed on its inputs that got restricted). But on the other hand we show that the LTF-of-ReLU collapses to a circuit which is of such a small size that with high-probability it cannot possibly approximate a randomly chosen Boolean function on the remaining inputs. This contradiction leads to a lowerbound.

There are two important concepts towards implementing the above idea. First one has to precisely define as to when can a ReLU gate upon a partial restriction of its inputs be considered to be removable from the circuit. Once this notion is clarified it will automatically turn out that doing random restrictions on ReLU is the same as doing random restriction on a LTF gate as was recently done in [18]. The secondly it needs to be true that at any fixed size LTF-of-ReLU circuits cannot represent too many of all the Boolean functions possible at the same input dimension. For this very specific case of LTF-of-ReLU circuits where ReLU gates necessarily have a fan-out of 1, Theorem 2.1 in [23] applies and we have from there that LTF-of-ReLU circuits over $n-$bits with w ReLU gates can represent at most $N = 2^{O((wn+w+w+1+1)^2 \log(wn+w+w+1+1))} = 2^{O((wn+2w+2)^2 \log(wn+2w+2))}$ number of Boolean functions. We note that slightly departing from the usual convention with neural networks here in this work by Wolfgaang Mass he allows for direct wires from the input nodes to the output LTF gate. This flexibility ties in nicely with how we want to define a ReLU gate to be becoming

removable under the random restrictions that we use.

Random Boolean functions vs any circuit class

In everything that follows all samplings being done (denoted as $\sim$) are to be understood as sampling from an uniform distribution unless otherwise specified. Firstly we note this well-known lemma,

Claim 1. Let $f : \{-1,1\}^n \to \{-1,1\}$ be any given Boolean function. Then the following is true,

$$\mathbb{P}_{g \sim \{\{-1,1\}^n \to \{-1,1\}\}} \left[\mathbb{P}_{\mathbf{x} \sim \{-1,1\}^n}[f(\mathbf{x}) = g(\mathbf{x})] \geq \frac{1}{2} + \epsilon \right] \leq e^{-2^{n+1}\epsilon^2}$$

From the above it follows that if N is the total number of functions in any circuit class (whose members be called C) then we have by union bound,

$$\mathbb{P}_{g \sim \{\{-1,1\}^n \to \{-1,1\}\}} \left[\exists C \text{ s.t } \mathbb{P}_{\mathbf{x} \sim \{-1,1\}^n}[C(\mathbf{x}) = g(\mathbf{x})] \geq \frac{1}{2} + \epsilon \right] \leq N e^{-2^{n+1}\epsilon^2} \qquad (4)$$

Equipped with these basics we are now ready to begin the proof of the lowerbound against weight unrestricted LTF-of-ReLU circuits,

Proof.

Definition 3. Let D denote arbitrary LTF-of-ReLU circuits over $\lfloor \log(\frac{n}{2}) \rfloor$ bits.

For some $\frac{\epsilon}{3} \leq \frac{1}{2}$ and a size function denoted as $s(n, \epsilon)$ we use equation 4 , the definition of D above and the upperbound given earlier for the number of LTF-of-ReLU functions at a fixed circuit size (now used for circuits on $\lfloor \log(\frac{n}{2}) \rfloor$ bits) to get,

$$\mathbb{P}_{f \sim \{0,1\}^{\lfloor \log(\frac{n}{2}) \rfloor} \to \{0,1\}} \left[\forall D \text{ s.t } |D| \leq s(n, \epsilon) \; \mathbb{P}_{\mathbf{y} \sim \{0,1\}^{\lfloor \log(\frac{n}{2}) \rfloor}}[f(\mathbf{y}) = D(\mathbf{y})] \leq \left(\frac{1}{2} + \frac{\epsilon}{3} \right) \right]$$

$$\geq 1 - 2^{O(s^2 \log^2(\frac{n}{2}) \log(\log(\frac{n}{2})s))} e^{-\left(\frac{\epsilon^2}{9} \right) 2^{1 + \lfloor \log(\frac{n}{2}) \rfloor}}$$

$$\geq 1 - 2^{O(s^2 k^2 \log(ks))} e^{-\left(\frac{2\epsilon^2}{9} \right) 2^k}$$

whereby in the last inequality above we have assumed that $n = 2^{k+1}$. This assumption is legitimate because we want to estimate certain large n asymptotics. For any arbitrarily chosen constant $C < \frac{2}{9}$ we try to satisfy the following condition, $O(s^2 k^2 \log(ks)) - \frac{2\epsilon^2 2^k}{9} \leq -C\epsilon^2 2^k \implies O(s^2 k^2 \log(ks)) \leq O(\epsilon^2 2^k)$. For any constant $\theta > 0$ for large enough $x > 0$ we would have $\log(x) < x^\theta$ and hence the above constraint on s gets satisfied if we work in the regime, $s \leq O(\frac{(\epsilon^2 2^k)^{\frac{1}{2+\theta}}}{k})$. So for this range of s we have, $2^{O(s^2 k^2 \log(ks))} e^{-\left(\frac{2\epsilon^2}{9} \right) 2^k} \leq e^{O(s^2 k^2 \log(ks)) - \left(\frac{2\epsilon^2}{9} \right) 2^k} \leq e^{-C\epsilon^2 2^k}$. Now we want, $e^{-C\epsilon^2 2^k} \leq \frac{\epsilon}{3}$. But on the otherhand for the upperbound on s to make sense we need, $\epsilon^2 2^k \geq k^{2+\theta}$. Its clear

that both the conditions get satisfied if for asymptotically large n we choose $\epsilon > \sqrt{\frac{2\log^{2+\theta}(\frac{n}{2})}{n}}$. And corresponding to this we have for $s(n,\epsilon) \le O(\frac{\epsilon^{\frac{2}{2+\theta}} n^{\frac{1}{2+\theta}}}{2^{\frac{1}{2+\theta}}\log(\frac{n}{2})})$

$$\mathbb{P}_{f\sim\{0,1\}^{\lfloor\log(\frac{n}{2})\rfloor}\to\{0,1\}}\left[\forall D \text{ s.t } |D| \le s(n,\epsilon)\ |\mathbb{P}_{\mathbf{y}\sim\{0,1\}^{\lfloor\log(\frac{n}{2})\rfloor}}[f(\mathbf{y}) = D(\mathbf{y})] \le \left(\frac{1}{2}+\frac{\epsilon}{3}\right)\right]$$
$$\ge 1 - \frac{\epsilon}{3} \tag{5}$$

Definition 4 (**F***)**.** Let F^* be the subset of all these f above for which the above event is true.

Now we recall the definition of the Andreev function in equation 1 for the following definition and the claim,

Definition 5 (ρ)**.** Let ρ denote the set of all possible "random restrictions" where one is fixing all the input bits of A_n except 1 bit in each row of the matrix a. So the restricted function (call it $A_n|_\rho$ by overloading the notation for simplicity) computes a function of the form,

$$A_n|_\rho : \{0,1\}^{\lfloor\log(\frac{n}{2})\rfloor} \to \{0,1\}$$

From the definitions of A_n and ρ above the following is immediate,

Claim 2. The truth table of $A_n|_\rho$ is the $\mathbf{x}$ string in the input to A_n that gets fixed by ρ. Thus we observe that if ρ is chosen uniformly at random then $A_n|_\rho$ is a $\lfloor\log(\frac{n}{2})\rfloor$ bit Boolean function chosen uniformly at random.

Let f^* be any arbitrary member of F^*. Let $\mathbf{x}^* \in \{0,1\}^{\lfloor\frac{n}{2}\rfloor}$ be the truth-table of f^*. Let $\rho(\mathbf{x}^*)$ be restrictions on the input of A_n which fix the $\mathbf{x}$ part of its input to $\mathbf{x}^*$. So when we are sampling restrictions uniformly at random from the restrictions of the type $\rho(\mathbf{x}^*)$ these different instances differ in which bit of each row of the matrix a (of the input to A_n) they left unfixed and to what values did they fix the other entries of a. Let C be a n bit LTF-of-ReLU Boolean circuit of size say $w(n,\epsilon)$. Thus under the restriction $\rho(\mathbf{x}^*)$ both C and A_n are $\lfloor\log(\frac{n}{2})\rfloor$ bit Boolean functions.

Now we note that a ReLU gate over n bits upon a random restriction becomes redundant (and hence removable) iff its linear argument either reduces to a non-positive definite function or a positive definite function. In the former case the gate is computing the constant function zero and in the later case it is computing a linear function which can be simply implemented by introducing wires connecting the inputs directly to the output LTF gate. Thus in both the cases the resultant function no more needs the ReLU gate for it to be computed. (We note that such direct wires from the input to the output gate were allowed in how the counting was done of the total number of LTF-of-ReLU Boolean functions at a fixed circuit size.) Combining both the cases we note that the conditions

for collapse (in this sense) of a ReLU gate is identical to that of the conditions of collapse for a LTF gate with the same linear argument. Hence corresponding to the random restrictions ρ we can just directly utilize the random restriction lemma 1.1 from [18] to say that,

$$\mathbb{P}_{\rho(\mathbf{x}^*)}[\text{ReLU}|_{\rho(\mathbf{x}^*)}\text{is removable }] \geq \eta$$

where for $\eta = 1 - O(\frac{\log n}{\sqrt{n}})$
The above definition of η implies,

$$\mathbb{P}_{\rho(\mathbf{x}^*)}[\text{ A } n-\text{bit ReLU is } \textit{not} \text{ forced to a constant }] \leq 1 - \eta$$
$$\implies \mathbb{E}_{\rho(\mathbf{x}^*)}[\text{ Number of ReLUs of C } \textit{not} \text{ forced to a constant }] \leq w(n, \epsilon)(1 - \eta)$$
$$\implies \mathbb{P}_{\rho(\mathbf{x}^*)}[\text{ Number of ReLUs of C } \textit{not} \text{ forced to a constant } > s(n, \epsilon)]$$
$$\leq \frac{\mathbb{E}_{\rho(\mathbf{x}^*)}[\text{ Number of ReLUs of C } \textit{not} \text{ forced to a constant }]}{s(n, \epsilon)}$$
$$\implies \mathbb{P}_{\rho(\mathbf{x}^*)}[\text{ Number of ReLUs of C } \textit{not} \text{ forced to a constant } \geq s(n, \epsilon)] \leq \frac{w(n, \epsilon)(1 - \eta)}{s(n, \epsilon)}$$
$$\implies \mathbb{P}_{\rho(\mathbf{x}^*)}[\text{ Size of } C|_{\rho(\mathbf{x}^*)} \leq s(n, \epsilon)] \geq 1 - \frac{w(n, \epsilon)(1 - \eta)}{s(n, \epsilon)}$$

$$(6)$$

Now we compare with the definitions of ϵ and f^* to observe that (a) with probability at least $1 - \frac{w(n,\epsilon)(1-\eta)}{s(n,\epsilon)}$, $C|_{\rho(\mathbf{x}^*)}$ is of the circuit type as in the event in equation 2 and (b) by definition of the Andreev function it follows that $A_n|_{\rho(\mathbf{x}^*)}$ has its truth table given by $\mathbf{x}^*$ and hence it specifies the same function as $f^* \in F^*$. Hence $\forall \mathbf{x}^*$ and $\rho(\mathbf{x}^*)$ this can as well write this as,

$$\mathbb{P}_{\mathbf{y}\sim\{0,1\}^{\lfloor\log(\frac{n}{2})\rfloor}}[C|_{\rho(\mathbf{x}^*)}(\mathbf{y}) = A_n|_{\rho(\mathbf{x}^*)}(\mathbf{y})| \text{ Size of } C|_{\rho(\mathbf{x}^*)} \leq s(n, \epsilon)] \leq \frac{1}{2} + \frac{\epsilon}{3} \qquad (7)$$

$\forall \mathbf{x}^*$ equation 6 can be rewritten as,

$$\mathbb{P}_{\rho(\mathbf{x}^*)}[\text{ Size of } C|_{\rho(\mathbf{x}^*)} \leq s(n, \epsilon)] \geq 1 - \frac{w(n, \epsilon)(1 - \eta)}{s(n, \epsilon)} \qquad (8)$$

The equation 5 can be written as,

$$\mathbb{P}_{f\sim\{0,1\}^{\lfloor\log(\frac{n}{2})\rfloor}\to\{0,1\}}[f \in F^*] \geq 1 - \frac{\epsilon}{3} \qquad (9)$$

Claim 3. Circuits C have low correlation with the Andreev function

$$\mathbb{P}_{\mathbf{z}\sim\{0,1\}^n}[C(z) = A_n(z)] \leq \frac{\epsilon}{3} + \frac{w(n, \epsilon)(1 - \eta)}{s(n, \epsilon)} + \frac{1}{2} + \frac{\epsilon}{3}$$

9

Proof. We think of sampling a $z \sim \{0,1\}^n$ as a two step process of first sampling a $\tilde{f}$, a $\lfloor \log(\frac{n}{2}) \rfloor$ bit Boolean function and fixing the first $\lfloor \frac{n}{2} \rfloor$ bits of z to be the truth-table of $\tilde{f}$ and then we randomly assign values to the remaining $\lfloor \frac{n}{2} \rfloor$ bits of z. Call these later $\lfloor \frac{n}{2} \rfloor$ bit string to be $\mathbf{x}_{other}$.

$$
\begin{aligned}
\mathbb{P}_{\mathbf{z} \sim \{0,1\}^n}[C(z) = A_n(z)] &= \mathbb{E}_{\mathbf{z} \sim \{0,1\}^n}[\mathbf{1}_{C(z)=A_n(z)}] \\
&= \mathbb{E}_{\mathbf{z} \sim \{0,1\}^n}[\mathbf{1}_{C(z)=A_n(z)}\mathbf{1}_{\tilde{f} \in F^*}] + \mathbb{E}_{\mathbf{z} \sim \{0,1\}^n}[\mathbf{1}_{C(z)=A_n(z)}\mathbf{1}_{\tilde{f} \notin F^*}] \\
&= \mathbb{P}_{\mathbf{z} \sim \{0,1\}^n}[(C(z) = A_n(z)) \cap (\tilde{f} \in F^*)] + \mathbb{P}_{\mathbf{z} \sim \{0,1\}^n}[(C(z) = A_n(z)) \cap (\tilde{f} \notin F^*)] \\
&= \mathbb{P}_{\mathbf{z} \sim \{0,1\}^n}[(C(z) = A_n(z)) \mid (\tilde{f} \in F^*)]\mathbb{P}_{\mathbf{z} \sim \{0,1\}^n}[\tilde{f} \in F^*] \\
&\quad + \mathbb{P}_{\mathbf{z} \sim \{0,1\}^n}[(C(z) = A_n(z)) \cap (\tilde{f} \notin F^*)] \\
&\leq \mathbb{P}_{\mathbf{z} \sim \{0,1\}^n}[(C(z) = A_n(z)) \mid (\tilde{f} \in F^*)] + \mathbb{P}_{\mathbf{z} \sim \{0,1\}^n}[\tilde{f} \notin F^*] \\
&\leq \mathbb{P}_{\mathbf{z} \sim \{0,1\}^n}[(C(z) = A_n(z)) \mid (\tilde{f} \in F^*)] + \frac{\epsilon}{3}
\end{aligned}
$$

In the last line above we have invoked equation 9. Now we note that sampling the n bit string z such that $\tilde{f} \in F^*$ is the same as doing a random restriction of the type $\rho(\tilde{f})$ and then randomly picking a $\lfloor \log(\frac{n}{2}) \rfloor$ bit string say $\mathbf{y}$. So we can rewrite the last inequality as,

$$
\begin{aligned}
\mathbb{P}_{\mathbf{z} \sim \{0,1\}^n}[C(z) = A_n(z)] &\leq \mathbb{P}_{(\rho(\tilde{f}),\mathbf{y})}[C(\rho(\tilde{f}),\mathbf{y}) = A_n(\rho(\tilde{f}),\mathbf{y})] + \frac{\epsilon}{3} \\
&\leq \mathbb{E}_{(\rho(\tilde{f}),\mathbf{y})}[\mathbf{1}_{C(\rho(\tilde{f}),\mathbf{y})=A_n(\rho(\tilde{f}),\mathbf{y})} \mid (\tilde{f} \in F^*)] + \frac{\epsilon}{3} \\
&\leq \mathbb{E}_{(\rho(\tilde{f}),\mathbf{y})}[\mathbf{1}_{C(\rho(\tilde{f}),\mathbf{y})=A_n(\rho(\tilde{f}),\mathbf{y})}\mathbf{1}_{\text{Size of } C|_{\rho(\tilde{f})}<s(n,\epsilon)} \mid (\tilde{f} \in F^*)] \\
&\quad + \mathbb{E}_{(\rho(\tilde{f}),\mathbf{y})}[\mathbf{1}_{C(\rho(\tilde{f}),\mathbf{y})=A_n(\rho(\tilde{f}),\mathbf{y})}\mathbf{1}_{\text{Size of } C|_{\rho(\tilde{f})}\geq s(n,\epsilon)} \mid (\tilde{f} \in F^*)] + \frac{\epsilon}{3} \\
&\leq \mathbb{P}_{(\rho(\tilde{f}),\mathbf{y})}[C(\rho(\tilde{f}),\mathbf{y}) = A_n(\rho(\tilde{f}),\mathbf{y}) \mid \left((\text{Size of } C|_{\rho(\tilde{f})} < s(n,\epsilon)) \cap (\tilde{f} \in F^*)\right)] \\
&\quad + \mathbb{P}_{(\rho(\tilde{f}),\mathbf{y})}[\text{Size of} C|_{\rho(\tilde{f})} \geq s(n,\epsilon) \mid (\tilde{f} \in F^*)] + \frac{\epsilon}{3} \\
&\leq \left(\frac{1}{2} + \frac{\epsilon}{3}\right) + \frac{w(n,\epsilon)(1-\eta)}{s(n,\epsilon)} + \frac{\epsilon}{3}
\end{aligned}
$$

In the last step above we have used equations 7 and 8. $\qquad \square$

So after putting back the values of η and the largest scaling of $s(n,\epsilon)$ that we can have (from equation 5), the upperbound on the above probability becomes,

$$
\frac{1}{2} + \frac{2\epsilon}{3} + O\left(\frac{w(n,\epsilon)\log(n)}{\sqrt{n}\left(\frac{\epsilon^{\frac{2}{2+\theta}} n^{\frac{1}{2+\theta}}}{2^{\frac{1}{2+\theta}}\log(\frac{n}{2})}\right)}\right)
$$

Thus the probability is upperbounded by $\frac{1}{2} + \epsilon$ as long as $w(n,\epsilon) = O\left(\frac{\epsilon^{1+\frac{2}{2+\theta}} n^{\frac{1}{2}+\frac{1}{2+\theta}} \log\left(\frac{n}{2}\right)}{\log(n)}\right)$

Stated as a lowerbound we have that if a LTF-of-ReLU has to match the $n-$bit Andreev function on more than $\frac{1}{2} + \epsilon$ fraction of the inputs for $\epsilon > \sqrt{\frac{2\log^{2+\theta}(\frac{n}{2})}{n}}$ for some $\theta > 0$ (asymptotically this is like having a constant ϵ) then the LTF-of-ReLU needs to be of size $\Omega(\epsilon^{\frac{4+\theta}{2+\theta}} n^{\frac{1}{2} + \frac{1}{2+\theta}})$. Now we define $\delta \in (0, \frac{1}{2})$ such that $\delta = \frac{\theta}{2(2+\theta)}$ and that gives the form of the almost linear lowerbound as stated in the theorem. $\qquad\square$

4 Smaller upper bounds on the sign-rank of LTF-of-(ReLU)$^{\mathrm{d}-1}$ with weight restrictions only on the bottom most layer (Proof of Theorem 2.5)

For a $\{-1,1\}^M \to \{-1,1\}$ LTF-of-ReLU circuit with any given weights on the network the inputs to the threshold function of the top LTF gate are some set of 2^M real numbers (one for each input). Over all these inputs let $p > 0$ be the distance from 0 of the largest negative number on which the LTF gate ever gets evaluated. Then by increasing the bias at this last LTF gate by a quantity less then p we can ensure that no input to this LTF gate is 0 while the entire circuit still computes the same Boolean function as originally. So we can assume without loss of generality that the input to the threshold function at the top LTF gate is never 0. We also recall that the weights at the bottom most layer are constrained to be integers of magnitude at most $W > 0$.

Let this depth d LTF-of-(ReLU)$^{d-1}$ circuit map $\{-1,1\}^m \times \{-1,1\}^m \to \{-1,1\}$. Let $\{w_k\}_{k=1}^{d-1}$ be the widths of the ReLU layers at depths indexed by increasing k with increasing distance from the input. Thus, the output LTF gate gets w_{d-1} inputs; the j-th input, for $j = 1, 2, .., w_{d-1}$, is the output of a circuit C_j of depth $d - 1$ composed of only ReLU gates. Let $f_j(\mathbf{x}, \mathbf{y}) : \{-1,1\}^m \times \{-1,1\}^m \to \mathbb{R}$ be the pseudo-Boolean function implemented by C_j. Thus the output of the overall LTF-of-(ReLU)$^{d-1}$ circuit is,

$$f(\mathbf{x}, \mathbf{y}) := \mathrm{LTF}\left[\beta + \sum_{j=1}^{w_{d-1}} \alpha_j f_j(\mathbf{x}, \mathbf{y})\right] \tag{10}$$

Lemma 4.1. Let $k \geq 0$ and $w_1, \ldots, w_k \geq 1$ be natural numbers. Consider a circuit with $2m$ inputs and a single output, consisting of only ReLU gates of depth $k + 1$ with w_i ReLU gates at each depth, with $i = 1$ corresponding to the layer closest to the input (note that single output ReLU gate is not counted here). We restrict the inputs to $\{-1,1\}^m \times \{-1,1\}^m$, so the circuit implements a pseudo-Boolean function $g : \{-1,1\}^m \times \{-1,1\}^m \to \mathbb{R}$.
Assume that the weights of the w_1 ReLU gates in the layer closest to the input are restricted as per Definition 2. Define the $2^m \times 2^m$ matrix $G(\mathbf{x}, \mathbf{y})$ whose rows and columns are indexed by $(\mathbf{x}, \mathbf{y}) \in \{-1,1\}^m \times \{-1,1\}^m$ as

$$G(\mathbf{x}, \mathbf{y}) = g(\mathbf{x}, \mathbf{y}).$$

Then G has a block structure, where the rows and columns can be partitioned *contiguously* into $O\big((\prod_{i=1}^k w_i)(mW)\big)$ blocks (thus, G has $O\big((\prod_{i=1}^k w_i)^2(mW)^2\big)$ blocks), and within each block G is constant valued.

Before we prove the Lemma, let us see why it implies Theorem 2.5. Let $F_j(\mathbf{x}, \mathbf{y})$ be the matrix obtained from the ReLU circuit outputs $f_j(\mathbf{x}, \mathbf{y})$ from (10), and let $F(\mathbf{x}, \mathbf{y})$ be the matrix obtained from $f(\mathbf{x}, \mathbf{y})$. Let $J_{2^m \times 2^m}$ be the matrix of all ones. Then

$$
\begin{aligned}
\text{sign-rank}(F(\mathbf{x}, \mathbf{y})) &= \text{sign-rank}\left(\text{sign}\left[\beta J_{2^m \times 2^m} + \sum_{j=1}^{w_{d-1}} \alpha_j F_j(\mathbf{x}, \mathbf{y})\right]\right) \\
&\leq \text{rank}\left(\beta J_{2^m \times 2^m} + \sum_{j=1}^{w_{d-1}} \alpha_j F_j(\mathbf{x}, \mathbf{y})\right) \\
&\leq 1 + \sum_{j=1}^{w_{d-1}} \text{rank}(F_j(\mathbf{x}, \mathbf{y})) \\
&= O\left(\left(\prod_{k=1}^{d-1} w_k\right)^2 (mW)^2\right)
\end{aligned}
$$

where the first inequality follows from the definition of sign-rank, the second inequality follows from the subadditivty of rank and the last inequality is a consequence of Lemma 4.1. Indeed, a matrix with block structure as in the conclusion of Lemma 4.1 has rank at most $O\left(\left(\prod_{i=1}^{k} w_i\right)^2 (mW)^2\right)$ by expressing it as a sum of these many matrices of rank one and using subaddivity of rank.

Now we recall that the Arkadev-Nikhil function g (which is linear sized depth 2 LTF) on $2m = 2(n^{\frac{4}{3}} - n \log n)$ bits has sign-rank $\Omega(2^{n^{\frac{1}{3}} - 2\log n})$. It follows that $n^{\frac{4}{3}} \geq m$ and for any constant C s.t $C \in (0, 1)$ for large enough n we would have, sign-rank$(g) = \Omega(2^{Cn^{\frac{1}{3}}}) = \Omega(2^{Cm^{\frac{1}{4}}})$. From the above upper bound on the sign-rank of our bottom layer weight restricted LTF-of-(ReLU)$^{d-1}$ with widths $\{w_k\}_{k=1}^{d-1}$ it follows that for this to represent this Arkadev-Nikhil function it would need, $\left(\left(\prod_{k=1}^{d-1} w_k\right)^2 (mW)^2\right) = \Omega(m^{\frac{1}{4}})$. Hence it follows that the size $(1 + \sum_{k=1}^{d-1} w_i)$ required for such

LTF-of-(ReLU)$^{d-1}$ circuits to represent the Arkadev-Nikhil function is $\Omega\left((d-1)\frac{2^{\frac{m^{\frac{1}{8}}}{d-1}}}{(mW)^{\frac{1}{d-1}}}\right)$.

The statement about LTF circuits is a straightforward consequence of the above result and Claim 5 in Appendix B which says that any LTF gate can be simulated by 2 ReLU gates.
We now prove Lemma 4.1.

Proof of Lemma 4.1. We will prove this Lemma by induction on k.

The base case of the induction $k = 0$: A single ReLU gate. A single ReLU gate's output is given by $\max\{0, \langle \mathbf{a}^1, \mathbf{x} \rangle + \langle \mathbf{a}^2, \mathbf{y} \rangle + b\}$, where $\mathbf{a}^1, \mathbf{a}^2 \in \mathbb{R}^m$ and $b \in \mathbb{R}$. Since the entries of $\mathbf{a}^1, \mathbf{a}^2$ and b are assumed to be integers bounded by $W > 0$, the terms $\langle \mathbf{a}^1, \mathbf{x} \rangle$ and $\langle \mathbf{a}^2, \mathbf{y} \rangle$ can each take at most $O(mW)$ different values, since $\mathbf{x}, \mathbf{y} \in \{-1, 1\}^m$. So we can arrange the rows and columns in increasing order of $\langle \mathbf{a}^1, \mathbf{x} \rangle$ and $\langle \mathbf{a}^2, \mathbf{y} \rangle$ and then partition the rows and columns contiguously according to these values, and the base case is proved.

The induction step. We first make a simple claim about the sum of matrices which are block wise constant.

Claim 4. Let w, M, D be fixed natural numbers. Let $A_1, \ldots, A_w$ be any $M \times M$ matrices such that for each A_i the rows and columns can be partitioned contiguously into D blocks (not necessarily equal in size), such that A_i is constant valued within each of the D^2 blocks. Then $A := A_1 + \ldots + A_w$ is an $M \times M$ matrix whose rows and columns can be partitioned contiguously into $w(D-1) + 1$ blocks such that A is constant valued within each block defined by this partition of the rows and columns.

Proof. The partition of the rows of A_i into D contiguous blocks is equivalent to a choice of $D-1$ lines out of $M-1$ lines. When we sum the matrices, the refined partition in the sum is a selection of $w(D-1)$ lines out of $M-1$ lines, giving us $w(D-1) + 1$ contiguous blocks. The same argument holds for the columns. □

To complete the induction step, we observe that a ReLU circuit with depth $k+1$ layers can be seen as computing $g(\mathbf{x}, \mathbf{y}) = \max\{0, b + \sum_{i=1}^{w_k} a_j g_i(\mathbf{x}, \mathbf{y})\}$, where $g_i(\mathbf{x}, \mathbf{y})$ is the output of a ReLU circuit of depth k. Thus, the corresponding matrices satisfy $G(\mathbf{x}, \mathbf{y}) = \max\{0, bJ_{2^m \times 2^m} + \sum_{i=1}^{w_k} a_j G_i(\mathbf{x}, \mathbf{y})\}$, where $J_{2^m \times 2^m}$ is the matrix of all ones, and the "max" is taken entrywise. the induction hypothesis tells us that the rows and columns each matrix G_i can be partitioned contiguously into $O\big((\prod_{i=1}^{k-1} w_i)(mW)\big)$ such that G_i is constant valued within each block. Thus, by Claim 4, the rows and columns of the matrix $bJ_{2^m \times 2^m} + \sum_{i=1}^{w_k} a_j G_i(\mathbf{x}, \mathbf{y})$ can be partitioned into $O\big((\prod_{i=1}^{k} w_i)(mW)\big)$ contiguous blocks. □

5 Acknowledgements

References

[1] A. E. Andreev. About one method of obtaining more than quadratic effective lower bounds of complexity of pi-schemes, 1987.

[2] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee. Understanding deep neural networks with rectified linear units. *https://eccc.weizmann.ac.il/report/2017/098/*, 2016.

[3] H. Buhrman, N. Vereshchagin, and R. de Wolf. On computation and communication with small bias. In *Computational Complexity, 2007. CCC'07. Twenty-Second Annual IEEE Conference on*, pages 24–32. IEEE, 2007.

[4] M. Bun and J. Thaler. Improved bounds on the sign-rank of ac^ 0. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 55. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[5] A. Chattopadhyay and N. S. Mande. Weights at the bottom matter when the top is heavy. *arXiv preprint arXiv:1705.02397*, 2017.

[6] R. Chen, R. Santhanam, and S. Srinivasan. Average-case lower bounds and satisfiability algorithms for small threshold circuits. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 50. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[7] A. Daniely. Depth separation for neural networks. *arXiv preprint arXiv:1702.08489*, 2017.

[8] R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In *Conference on Learning Theory*, pages 907–940, 2016.

[9] J. Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *Journal of Computer and System Sciences*, 65(4):612–625, 2002.

[10] J. Forster, M. Krause, S. V. Lokam, R. Mubarakzjanov, N. Schmitt, and H. U. Simon. Relations between communication complexity, linear arrangements, and computational complexity. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 171–182. Springer, 2001.

[11] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turan. Threshold circuits of bounded depth. In *Foundations of Computer Science, 1987., 28th Annual Symposium on*, pages 99–110. IEEE, 1987.

[12] B. Hanin. Universal function approximation by deep neural nets with bounded width and relu activations. *arXiv preprint arXiv:1708.02691*, 2017.

[13] J. Hastad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 6–20. ACM, 1986.

[14] R. Impagliazzo, R. Meka, and D. Zuckerman. Pseudorandomness from shrinkage. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 111–119. IEEE, 2012.

[15] R. Impagliazzo and M. Naor. Decision trees and downward closures. In *Structure in Complexity Theory Conference, 1988. Proceedings., Third Annual*, pages 29–38. IEEE, 1988.

[16] R. Impagliazzo, R. Paturi, and M. E. Saks. Size–depth tradeoffs for threshold circuits. *SIAM Journal on Computing*, 26(3):693–707, 1997.

[17] V. Kabanets, D. Kane, and Z. Lu. A polynomial restriction lemma with applications. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 24, page 26, 2017.

[18] D. M. Kane and R. Williams. Super-linear gate and super-quadratic wire lower bounds for depth-two and depth-three threshold circuits. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 633–643. ACM, 2016.

[19] M. Krause and P. Pudlák. On the computational power of depth 2 circuits with threshold and modulo gates. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 48–57. ACM, 1994.

[20] T. Lee, A. Shraibman, et al. Lower bounds in communication complexity. *Foundations and Trends® in Theoretical Computer Science*, 3(4):263–399, 2009.

[21] S. Liang and R. Srikant. Why deep neural networks? *arXiv preprint arXiv:1610.04161*, 2016.

[22] S. V. Lokam et al. Complexity lower bounds using linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 4(1–2):1–155, 2009.

[23] W. Maass. Bounds for the computational power and learning complexity of analog neural nets. *SIAM Journal on Computing*, 26(3):708–732, 1997.

[24] M. S. Paterson and U. Zwick. Shrinkage of de morgan formulae under restriction. *Random Structures & Algorithms*, 4(2):135–150, 1993.

[25] T. Poggio and Q. Liao. Theory ii: Landscape of the empirical risk in deep learning. *arXiv preprint arXiv:1703.09833*, 2017.

[26] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, pages 1–17, 2017.

[27] A. A. Razborov. On small depth threshold circuits. In *Scandinavian Workshop on Algorithm Theory*, pages 42–52. Springer, 1992.

[28] A. A. Razborov and A. A. Sherstov. The sign-rank of ac ^0. *SIAM Journal on Computing*, 39(5):1833–1855, 2010.

[29] B. Rossman. On the constant-depth complexity of k-clique. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 721–730. ACM, 2008.

[30] I. Safran and O. Shamir. Depth separation in relu networks for approximating smooth non-linear functions. *arXiv preprint arXiv:1610.09887*, 2016.

[31] A. A. Sherstov. Powering requires threshold depth 3. *Information processing letters*, 102(2-3):104–107, 2007.

[32] A. A. Sherstov. Separating ac^0 from depth-2 majority circuits. *SIAM Journal on Computing*, 38(6):2113–2129, 2009.

[33] A. A. Sherstov. The unbounded-error communication complexity of symmetric functions. *Combinatorica*, 31(5):583–614, 2011.

[34] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[35] K.-Y. Siu, V. P. Roychowdhury, and T. Kailath. Rational approximation techniques for analysis of neural networks. *IEEE Transactions on Information Theory*, 40(2):455–466, 1994.

[36] J. H. stad. The shrinkage exponent of de morgan formulas is 2. *SIAM Journal on Computing*, 27(1):48–64, 1998.

[37] B. A. Subbotovskaya. Realizations of linear functions by formulas using +. *Doklady Akademii Nauk SSSR*, 136(3):553–555, 1961.

[38] S. Tamaki. A satisfiability algorithm for depth two circuits with a sub-quadratic number of symmetric and threshold gates. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 23, page 4, 2016.

[39] M. Telgarsky. Benefits of depth in neural networks. *arXiv preprint arXiv:1602.04485*, 2016.

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

[41] D. Yarotsky. Error bounds for approximations with deep relu networks. *arXiv preprint arXiv:1610.01145*, 2016.

[42] C. Zhang, Q. Liao, A. Rakhlin, K. Sridharan, B. Miranda, N. Golowich, and T. Poggio. Theory of deep learning iii: Generalization properties of sgd. Technical report, Center for Brains, Minds and Machines (CBMM), 2017.

A   Proof of Proposition 2.2

We first observe that the set of points where $\max\{0, x_1, x_2\}$ is not differentiable is precisely the union of the three half-lines (or rays) $\{(x_1, x_2) : x_1 = x_2, x_1 \geq 0\} \cup \{(0, x_2) : x_2 \leq 0\} \cup \{(x_1, 0) : x_1 \leq 0\}$. On the other hand, consider any Sum-of-ReLU circuit, which can be expressed as a function of the form

$$f(x) = \sum_{i=1}^{w} c_i \max\{0, \langle a^i, x \rangle + b_i\},$$

where $w \in \mathbb{N}$ is the number of ReLU gates in the ciruit, and $a^i \in \mathbb{R}^2$, $b_i, c_i \in \mathbb{R}$ for all $i = 1, \ldots, w$. This implies that $f(x)$ is piecewise linear and the set of points where $f(x)$ is not differentiable is *precisely* the union of the w lines $\langle a^i, x \rangle + b_i = 0$, $i = 1, \ldots, w$. Since a union of lines cannot equal the union of the three half-lines $\{(x_1, x_2) : x_1 = x_2, x_1 \geq 0\} \cup \{(0, x_2) : x_2 \leq 0\} \cup \{(x_1, 0) : x_1 \leq 0\}$, we obtain the consequence that $\max\{0, x_1, x_2\}$ cannot be represented by a Sum-of-ReLU circuit, no matter how many ReLU gates are used.

B   Simulating an LTF gate by a ReLU gate

Claim 5. Any LTF gate $\{-1, 1\}^n \to \{-1, 1\}$ can be simulated by a Sum-of-ReLU circuit with at most 2 ReLU gates.

Proof. Given a LTF gate $(2\mathbf{1}_{\langle a,x \rangle + b \geq 0} - 1)$ it separates the points in $\{-1, 1\}^n$ into two subsets such that the plane $\langle a, x \rangle + b = 0$ is a separating hyperplane between the two sets. Let $-p < 0$ be the value of the function $\langle a, x \rangle + b$ at that hypercube vertex on the "-1" side which is closest to this separating plane. Now imagine a continuous piecewise linear function $f : \mathbb{R} \to \mathbb{R}$ such that $f(x) = -1$ for $x \leq -p$, $f(x) = 1$ for $x \geq 0$ and for $x \in (-p, 0)$ f is the straight line function connecting $(-p, -1)$ to $(0, 1)$. It follows from Corollary 3.1 of our previous work, [2] that this f can be implemented by a $\mathbb{R} \to \mathbb{R}$ Sum-of-ReLU with at most 2 ReLU gates hinged at the points $-p$ and 0 on the domain.

Because the affine transformation $\langle a, x \rangle + b$ can be implemented by the wires connecting the n input nodes to the layer of ReLUs it follows that there exists a $\mathbb{R}^n \to \mathbb{R}$ Sum-of-ReLU with at most 2 ReLU gates implementing the function $g(x) = f(\langle a, x \rangle + b) : \mathbb{R}^n \to \mathbb{R}$. Its clear that $g(\mathbf{x}) = \text{LTF}(\mathbf{x})$ for all $\mathbf{x} \in \{-1, 1\}^n$. $\qquad\square$

C   PARITY on $k-$bits can be implemented by a $O(k)$ Sum-of-ReLU circuit

For this proof its convenient to think of the PARITY function as the following map,

$$\text{PARITY} : \{0, 1\}^k \to \{0, 1\}$$

$$\mathbf{x} \mapsto \left(\sum_{i=1}^{k} x_i \right) \mod 2$$

Its clear that that in the evaluation of the PARITY function as stated above the required sum over the coordinates of the input Boolean vector will take as value every integer in the set, $\{0, 1, 2, .., k\}$. The PARITY function can then be lifted to a $f : \mathbb{R} \to \mathbb{R}$ function such that, $f(y) = 0$ for all $y \leq 0$, $f(y) = y \mod 2$ for all $y \in 1, 2, .., k$, $f(y) = k \mod 2$ for all $y > k$ and for any $y \in (p, p+1)$ for $p \in \{0, 1, .., k-1\}$ f is the straight line function connecting the points, $(p, p \mod 2)$ and $(p+1, (p+1) \mod 2)$. Thus f is a continuous piecewise linear function on $\mathbb{R}$ with $k + 2$ linear pieces. Then it follows from Theorem 2.3 of our previous work, [2] that this f can be implemented by a $\mathbb{R} \to \mathbb{R}$ Sum-of-ReLU circuit with at most $k + 1$ ReLU gates hinged at the points $\{0, 1, 2, .., k\}$ on the domain. The wires from the k inputs of the ReLU gates can implement the linear function $\sum_{i=1}^{k} x_i$. Thus it follows that there exists a $\mathbb{R}^k \to \mathbb{R}$ Sum-of-ReLU circuit (say C) such that, $\text{C}(\mathbf{x}) = \text{PARITY}(\mathbf{x})$ for all $\mathbf{x} \in \{0, 1\}^k$.