Tropical Geometry of Deep Neural Networks

Liwen Zhang [1] **Gregory Naitzat** [2] **Lek-Heng Lim** [2 3]

Abstract

We establish, for the first time, explicit connections between feedforward neural networks with ReLU activation and tropical geometry — we show that the family of such neural networks is equivalent to the family of tropical rational maps. Among other things, we deduce that feedforward ReLU neural networks with one hidden layer can be characterized by zonotopes, which serve as building blocks for deeper networks; we relate decision boundaries of such neural networks to tropical hypersurfaces, a major object of study in tropical geometry; and we prove that linear regions of such neural networks correspond to vertices of polytopes associated with tropical rational functions. An insight from our tropical formulation is that a deeper network is exponentially more expressive than a shallow network.

1. Introduction

Deep neural networks have recently received much limelight for their enormous success in a variety of applications across many different areas of artificial intelligence, computer vision, speech recognition, and natural language processing (LeCun et al., 2015; Hinton et al., 2012; Krizhevsky et al., 2012; Bahdanau et al., 2014; Kalchbrenner & Blunsom, 2013). Nevertheless, it is also well-known that our theoretical understanding of their efficacy remains incomplete.

There have been several attempts to analyze deep neural networks from different perspectives. Notably, earlier studies have suggested that a deep architecture could use parameters more efficiently and requires exponentially fewer parameters to express certain families of functions than a shallow architecture (Delalleau & Bengio, 2011; Bengio & Delalleau, 2011; Montufar et al., 2014; Eldan & Shamir, 2016; Poole et al., 2016; Telgarsky, 2016; Arora et al., 2018). Recent work (Zhang et al., 2016) showed that several successful neural networks possess a high representation power and can easily shatter random data. However, they also generalize well to data unseen during training stage, suggesting that such networks may have some implicit regularization. Traditional measures of complexity such as VC-dimension and Rademacher complexity fail to explain this phenomenon. Understanding this implicit regularization that begets the generalization power of deep neural networks remains a challenge.

The goal of our work is to establish connections between neural network and tropical geometry in the hope that they will shed light on the workings of deep neural networks. Tropical geometry is a new area in algebraic geometry that has seen an explosive growth in the recent decade but remains relatively obscure outside pure mathematics. We will focus on feedforward neural networks with rectified linear units (ReLU) and show that they are analogues of *rational functions*, i.e., ratios of two multivariate polynomials f, g in variables $x_1, \dots, x_d$,

$$\frac{f(x_1, \dots, x_d)}{g(x_1, \dots, x_d)},$$

in *tropical algebra*. For standard and trigonometric polynomials, it is known that *rational approximation* — approximating a target function by a ratio of two polynomials instead of a single polynomial — vastly improves the quality of approximation without increasing the degree. This gives our analogue: An ReLU neural network is the tropical ratio of two tropical polynomials, i.e., a tropical rational function. More precisely, if we view a neural network as a function $\nu : \mathbb{R}^d \to \mathbb{R}^p$, $x = (x_1, \dots, x_d) \mapsto (\nu_1(x), \dots, \nu_p(x))$, then ν is a tropical rational map, i.e., each ν_i is a tropical rational function. In fact, we will show that:

> *the family of functions represented by feedforward neural networks with rectified linear units and integer weights is exactly the family of tropical rational maps.*

It immediately follows that there is a *semifield* structure on this family of functions. More importantly, this establishes a

[1] Department of Computer Science, University of Chicago, Chicago, IL [2] Department of Statistics, University of Chicago, Chicago, IL [3] Computational and Applied Mathematics Initiative, University of Chicago, Chicago, IL. Correspondence to: Lek-Heng Lim <lekheng@galton.uchicago.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

bridge between neural networks[1] and tropical geometry that allows us to view neural networks as well-studied tropical geometric objects. This insight allows us to closely relate boundaries between linear regions of a neural network to tropical hypersurfaces and thereby facilitate studies of decision boundaries of neural networks in classification problems as tropical hypersurfaces. Furthermore, the number of linear regions, which captures the complexity of a neural network (Montufar et al., 2014; Raghu et al., 2017; Arora et al., 2018), can be bounded by the number of vertices of the polytopes associated with the neural network's tropical rational representation. Lastly, a neural network with one hidden layer can be completely characterized by zonotopes, which serve as building blocks for deeper networks.

In Sections 2 and 3 we introduce basic tropical algebra and tropical algebraic geometry of relevance to us. We state our assumptions precisely in Section 4 and establish the connection between tropical geometry and multilayer neural networks in Section 5. We analyze neural networks with tropical tools in Section 6, proving that a deeper neural network is exponentially more expressive than a shallow network — though our objective is not so much to perform state-of-the-art analysis but to demonstrate that tropical algebraic geometry can provide useful insights. All proofs are deferred to Section D of the supplement.

2. Tropical Algebra

Roughly speaking, tropical algebraic geometry is an analogue of classical algebraic geometry over $\mathbb{C}$, the field of complex numbers, but where one replaces $\mathbb{C}$ by a semifield[2] called the tropical semiring, to be defined below. We give a brief review of tropical algebra and introduce some relevant notations. See (Itenberg et al., 2009; Maclagan & Sturmfels, 2015) for an in-depth treatment.

The most fundamental component of tropical algebraic geometry is the *tropical semiring* $\mathbb{T} := \big(\mathbb{R} \cup \{-\infty\}, \oplus, \odot\big)$, also known as the max-plus algebra. The two operations $\oplus$ and $\odot$, called *tropical addition* and *tropical multiplication* respectively, are defined as follows.

Definition 2.1. *For $x, y \in \mathbb{R}$, their* tropical sum *is $x \oplus y := \max\{x, y\}$; their* tropical product *is $x \odot y := x + y$; the* tropical quotient *of x over y is $x \oslash y := x - y$.*

For any $x \in \mathbb{R}$, we have $-\infty \oplus x = 0 \odot x = x$ and $-\infty \odot x = -\infty$. Thus $-\infty$ is the tropical additive identity and 0 is the tropical multiplicative identity. Furthermore, these operations satisfy the usual laws of arithmetic: associativity, commutativity, and distributivity. The set $\mathbb{R} \cup \{-\infty\}$ is therefore a semiring under the operations $\oplus$ and $\odot$. While it is not a ring (lacks additive inverse), one may nonetheless generalize many algebraic objects (e.g., matrices, polynomials, tensors, etc) and notions (e.g., rank, determinant, degree, etc) over the tropical semiring — the study of these, in a nutshell, constitutes the subject of tropical algebra.

Let $\mathbb{N} = \{n \in \mathbb{Z} : n \geq 0\}$. For an integer $a \in \mathbb{N}$, raising $x \in \mathbb{R}$ to the ath power is the same as multiplying x to itself a times. When standard multiplication is replaced by tropical multiplication, this gives us *tropical power*:

$$x^{\odot a} := x \odot \cdots \odot x = a \cdot x,$$

where the last $\cdot$ denotes standard product of real numbers; it is extended to $\mathbb{R} \cup \{-\infty\}$ by defining, for any $a \in \mathbb{N}$,

$$-\infty^{\odot a} := \begin{cases} -\infty & \text{if } a > 0, \\ 0 & \text{if } a = 0. \end{cases}$$

A tropical semiring, while not a field, possesses one quality of a field: Every $x \in \mathbb{R}$ has a tropical multiplicative inverse given by its standard additive inverse, i.e., $x^{\odot(-1)} := -x$. Though not reflected in its name, $\mathbb{T}$ is in fact a *semifield*.

One may therefore also raise $x \in \mathbb{R}$ to a negative power $a \in \mathbb{Z}$ by raising its tropical multiplicative inverse $-x$ to the positive power $-a$, i.e., $x^{\odot a} = (-x)^{\odot(-a)}$. As is the case in standard real arithmetic, the tropical additive inverse $-\infty$ does not have a tropical multiplicative inverse and $-\infty^{\odot a}$ is undefined for $a < 0$. For notational simplicity, we will henceforth write x^a instead of $x^{\odot a}$ for tropical power when there is no cause for confusion. Other algebraic rules of tropical power may be derived from definition; see Section B in the supplement.

We are now in a position to define tropical polynomials and tropical rational functions. In the following, x and x_i will denote variables (i.e., indeterminates).

Definition 2.2. *A* tropical monomial *in d variables $x_1, \dots, x_d$ is an expression of the form*

$$c \odot x_1^{a_1} \odot x_2^{a_2} \odot \cdots \odot x_d^{a_d}$$

where $c \in \mathbb{R} \cup \{-\infty\}$ and $a_1, \dots, a_d \in \mathbb{N}$. As a convenient shorthand, we will also write a tropical monomial in multiindex notation as cx^α where $\alpha = (a_1, \dots, a_d) \in \mathbb{N}^d$ and $x = (x_1, \dots, x_d)$. Note that $x^\alpha = 0 \odot x^\alpha$ as 0 is the tropical multiplicative identity.

Definition 2.3. *Following notations above, a* tropical polynomial *$f(x) = f(x_1, \dots, x_d)$ is a finite tropical sum of tropical monomials*

$$f(x) = c_1 x^{\alpha_1} \oplus \cdots \oplus c_r x^{\alpha_r},$$

where $\alpha_i = (a_{i1}, \dots, a_{id}) \in \mathbb{N}^d$ and $c_i \in \mathbb{R} \cup \{-\infty\}$, $i = 1, \dots, r$. We will assume that a monomial of a given multiindex appears at most once in the sum, i.e., $\alpha_i \neq \alpha_j$ for any $i \neq j$.

[1] Henceforth a "neural network" will always mean a feedforward neural network with ReLU activation.

[2] A semifield is a field sans the existence of additive inverses.

Definition 2.4. *Following notations above, a* tropical rational function *is a standard difference, or, equivalently, a tropical quotient of two tropical polynomials $f(x)$ and $g(x)$:*

$$f(x) - g(x) = f(x) \oslash g(x).$$

We will denote a tropical rational function by $f \oslash g$, where f and g are understood to be tropical polynomial functions.

It is routine to verify that the set of tropical polynomials $\mathbb{T}[x_1, \dots, x_d]$ forms a semiring under the standard extension of $\oplus$ and $\odot$ to tropical polynomials, and likewise the set of tropical rational functions $\mathbb{T}(x_1, \dots, x_d)$ forms a semifield. We regard a tropical polynomial $f = f \oslash 0$ as a special case of a tropical rational function and thus $\mathbb{T}[x_1, \dots, x_d] \subseteq \mathbb{T}(x_1, \dots, x_d)$. Henceforth any result stated for a tropical rational function would implicitly also hold for a tropical polynomial.

A d-variate tropical polynomial $f(x)$ defines a function $f : \mathbb{R}^d \to \mathbb{R}$ that is a *convex function* in the usual sense as taking max and sum of convex functions preserve convexity (Boyd & Vandenberghe, 2004). As such, a tropical rational function $f \oslash g : \mathbb{R}^d \to \mathbb{R}$ is a *DC function* or *difference-convex function* (Hartman, 1959; Tao & Hoai An, 2005).

We will need a notion of vector-valued tropical polynomials and tropical rational functions.

Definition 2.5. *$F : \mathbb{R}^d \to \mathbb{R}^p$, $x = (x_1, \dots, x_d) \mapsto (f_1(x), \dots, f_p(x))$, is called a* tropical polynomial map *if each $f_i : \mathbb{R}^d \to \mathbb{R}$ is a tropical polynomial, $i = 1, \dots, p$, and a* tropical rational map *if $f_1, \dots, f_p$ are tropical rational functions. We will denote the set of tropical polynomial maps by $\operatorname{Pol}(d, p)$ and the set of tropical rational maps by $\operatorname{Rat}(d, p)$. So $\operatorname{Pol}(d, 1) = \mathbb{T}[x_1, \dots, x_d]$ and $\operatorname{Rat}(d, 1) = \mathbb{T}(x_1, \dots, x_d)$.*

3. Tropical Hypersurfaces

There are tropical analogues of many notions in classical algebraic geometry (Itenberg et al., 2009; Maclagan & Sturmfels, 2015), among which are *tropical hypersurfaces*, tropical analogues of algebraic curves in classical algebraic geometry. Tropical hypersurfaces are a principal object of interest in tropical geometry and will prove very useful in our approach towards neural networks. Intuitively, the tropical hypersurface of a tropical polynomial f is the set of points x where f is not linear at x.

Definition 3.1. *The* tropical hypersurface *of a tropical polynomial $f(x) = c_1 x^{\alpha_1} \oplus \dots \oplus c_r x^{\alpha_r}$ is*

$$\mathcal{T}(f) := \bigl\{x \in \mathbb{R}^d : c_i x^{\alpha_i} = c_j x^{\alpha_j} = f(x) \\ \text{for some } \alpha_i \neq \alpha_j\bigr\}.$$

i.e., the set of points x at which the value of f at x is attained by two or more monomials in f.

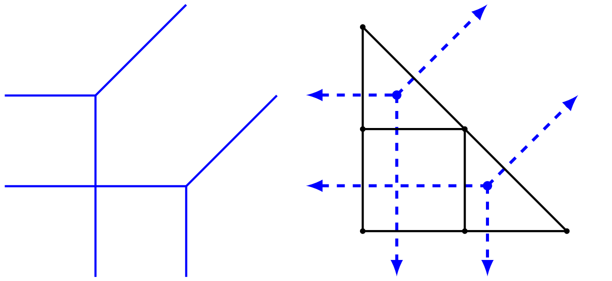

Figure 1. $1 \odot x_1^2 \oplus 1 \odot x_2^2 \oplus 2 \odot x_1x_2 \oplus 2 \odot x_1 \oplus 2 \odot x_2 \oplus 2$. Left: Tropical curve. Right: Dual subdivision of Newton polygon and tropical curve.

A tropical hypersurface divides the domain of f into convex cells on each of which f is linear. These cells are convex polyhedrons, i.e., defined by linear inequalities with integer coefficients: $\{x \in \mathbb{R}^d : Ax \leq b\}$ for $A \in \mathbb{Z}^{m \times d}$ and $b \in \mathbb{R}^m$. For example, the cell where a tropical monomial $c_j x^{\alpha_j}$ attains its maximum is $\{x \in \mathbb{R}^d : c_j + \alpha_j^{\mathsf{T}} x \geq c_i + \alpha_i^{\mathsf{T}} x \text{ for all } i \neq j\}$. Tropical hypersurfaces of polynomials in two variables (i.e., in $\mathbb{R}^2$) are called *tropical curves*.

Just like standard multivariate polynomials, every tropical polynomial comes with an associated *Newton polygon*.

Definition 3.2. *The* Newton polygon *of a tropical polynomial $f(x) = c_1 x^{\alpha_1} \oplus \dots \oplus c_r x^{\alpha_r}$ is the convex hull of $\alpha_1, \dots, \alpha_r \in \mathbb{N}^d$, regarded as points in $\mathbb{R}^d$,*

$$\Delta(f) := \operatorname{Conv}\bigl\{\alpha_i \in \mathbb{R}^d : c_i \neq -\infty,\ i = 1, \dots, r\bigr\}.$$

A tropical polynomial f determines a dual subdivision of $\Delta(f)$, constructed as follows. First, lift each α_i from $\mathbb{R}^d$ into $\mathbb{R}^{d+1}$ by appending c_i as the last coordinate. Denote the convex hull of the lifted $\alpha_1, \dots, \alpha_r$ as

$$\mathcal{P}(f) := \operatorname{Conv}\{(\alpha_i, c_i) \in \mathbb{R}^d \times \mathbb{R} : i = 1, \dots, r\}. \quad (1)$$

Next let $\operatorname{UF}\bigl(\mathcal{P}(f)\bigr)$ denote the collection of upper faces in $\mathcal{P}(f)$ and $\pi : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ be the projection that drops the last coordinate. The dual subdivision determined by f is then

$$\delta(f) := \bigl\{\pi(p) \subseteq \mathbb{R}^d : p \in \operatorname{UF}\bigl(\mathcal{P}(f)\bigr)\bigr\}.$$

$\delta(f)$ forms a polyhedral complex with support $\Delta(f)$. By (Maclagan & Sturmfels, 2015, Proposition 3.1.6), the tropical hypersurface $\mathcal{T}(f)$ is the $(d-1)$-skeleton of the polyhedral complex dual to $\delta(f)$. This means that each vertex in $\delta(f)$ corresponds to one "cell" in $\mathbb{R}^d$ where the function f is linear. Thus, the number of vertices in $\mathcal{P}(f)$ provides an upper bound on the number of linear regions of f.

Figure 1 shows the Newton polygon and dual subdivision for the tropical polynomial $f(x_1, x_2) = 1 \odot x_1^2 \oplus 1 \odot x_2^2 \oplus 2 \odot x_1x_2 \oplus 2 \odot x_1 \oplus 2 \odot x_2 \oplus 2$. Figure 2 shows how we

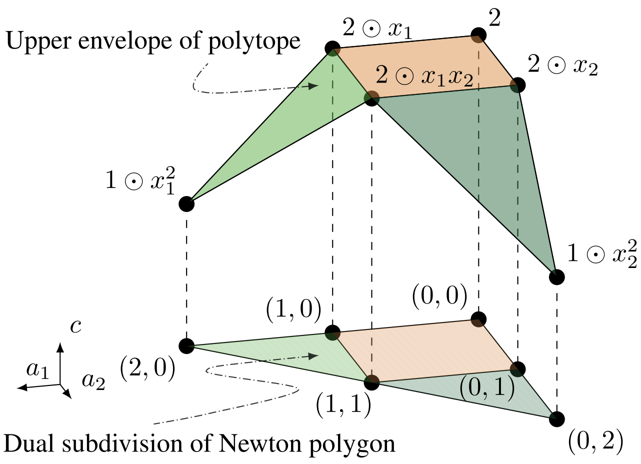

Figure 2. $1 \odot x_1^2 \oplus 1 \odot x_2^2 \oplus 2 \odot x_1 x_2 \oplus 2 \odot x_1 \oplus 2 \odot x_2 \oplus 2$. The dual subdivision can be obtained by projecting the edges on the upper faces of the polytope.

may find the dual subdivision for this tropical polynomial by following the aforementioned procedures; with step-by-step details given in Section C.1.

Tropical polynomials and tropical rational functions are clearly piecewise linear functions. As such a tropical rational map is a piecewise linear map and the notion of *linear region* applies.

Definition 3.3. *A* linear region *of $F \in \mathrm{Rat}(d,m)$ is a maximal connected subset of the domain on which F is linear. The number of linear regions of F is denoted $\mathcal{N}(F)$.*

Note that a tropical *polynomial* map $F \in \mathrm{Pol}(d,m)$ has convex linear regions but a tropical *rational* map $F \in \mathrm{Rat}(d,n)$ generally has nonconvex linear regions. In Section 6.3, we will use $\mathcal{N}(F)$ as a measure of complexity for an $F \in \mathrm{Rat}(d,n)$ given by a neural network.

3.1. Transformations of Tropical Polynomials

Our analysis of neural networks will require figuring out how the polytope $\mathcal{P}(f)$ transforms under tropical power, sum, and product. The first is straightforward.

Proposition 3.1. *Let f be a tropical polynomial and let $a \in \mathbb{N}$. Then*

$$\mathcal{P}(f^a) = a\mathcal{P}(f).$$

$a\mathcal{P}(f) = \{ax : x \in \mathcal{P}(f)\} \subseteq \mathbb{R}^{d+1}$ is a scaled version of $\mathcal{P}(f)$ with the same shape but different volume.

To describe the effect of tropical sum and product, we need a few notions from convex geometry. The *Minkowski sum* of two sets P_1 and P_2 in $\mathbb{R}^d$ is the set

$$P_1 + P_2 := \bigl\{x_1 + x_2 \in \mathbb{R}^d : x_1 \in P_1, x_2 \in P_2\bigr\};$$

and for $\lambda_1, \lambda_2 \geq 0$, their *weighted Minkowski sum* is

$$\lambda_1 P_1 + \lambda_2 P_2 := \bigl\{\lambda_1 x_1 + \lambda_2 x_2 \in \mathbb{R}^d : x_1 \in P_1, x_2 \in P_2\bigr\}.$$

Weighted Minkowski sum is clearly commutative and associative and generalizes to more than two sets. In particular, the Minkowski sum of line segments is called a *zonotope*.

Let $\mathcal{V}(P)$ denote the set of vertices of a polytope P. Clearly, the Minkowski sum of two polytopes is given by the convex hull of the Minkowski sum of their vertex sets, i.e., $P_1 + P_2 = \mathrm{Conv}\bigl(\mathcal{V}(P_1) + \mathcal{V}(P_2)\bigr)$. With this observation, the following is immediate.

Proposition 3.2. *Let $f, g \in \mathrm{Pol}(d,1) = \mathbb{T}[x_1, \dots, x_d]$ be tropical polynomials. Then*

$$\mathcal{P}(f \odot g) = \mathcal{P}(f) + \mathcal{P}(g),$$
$$\mathcal{P}(f \oplus g) = \mathrm{Conv}\bigl(\mathcal{V}(\mathcal{P}(f)) \cup \mathcal{V}(\mathcal{P}(g))\bigr).$$

We reproduce below part of (Gritzmann & Sturmfels, 1993, Theorem 2.1.10) and derive a corollary for bounding the number of verticies on the upper faces of a zonotope.

Theorem 3.3 (Gritzmann–Sturmfels)**.** *Let $P_1, \dots, P_k$ be polytopes in $\mathbb{R}^d$ and let m denote the total number of non-parallel edges of $P_1, \dots, P_k$. Then the number of vertices of $P_1 + \cdots + P_k$ does not exceed*

$$2\sum_{j=0}^{d-1} \binom{m-1}{j}.$$

The upper bound is attained if all P_i's are zonotopes and all their generating line segments are in general positions.

Corollary 3.4. *Let $P \subseteq \mathbb{R}^{d+1}$ be a zonotope generated by m line segments $P_1, \dots, P_m$. Let $\pi : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ be the projection. Suppose P satisfies:*

(i) *the generating line segments are in general positions;*

(ii) *the set of projected vertices $\{\pi(v) : v \in \mathcal{V}(P)\} \subseteq \mathbb{R}^d$ are in general positions.*

Then P has

$$\sum_{j=0}^{d} \binom{m}{j}$$

vertices on its upper faces. If either (i) or (ii) is violated, then this becomes an upper bound.

As we mentioned, linear regions of a tropical polynomial f correspond to vertices on $\mathrm{UF}\bigl(\mathcal{P}(f)\bigr)$ and the corollary will be useful for bounding the number of linear regions.

4. Neural Networks

While we expect our readership to be familiar with feedforward neural networks, we will nevertheless use this short section to define them, primarily for the purpose of fixing notations and specifying the assumptions that we retain throughout this article. We restrict our attention to fully connected feedforward neural networks.

Viewed abstractly, an L-layer feedforward neural network is a map $\nu : \mathbb{R}^d \to \mathbb{R}^p$ given by a composition of functions

$$\nu = \sigma^{(L)} \circ \rho^{(L)} \circ \sigma^{(L-1)} \circ \rho^{(L-1)} \cdots \circ \sigma^{(1)} \circ \rho^{(1)}.$$

The *preactivation* functions $\rho^{(1)}, \dots, \rho^{(L)}$ are affine transformations to be determined and the *activation* functions $\sigma^{(1)}, \dots, \sigma^{(L)}$ are chosen and fixed in advanced.

We denote the *width*, i.e., the number of nodes, of the lth layer by n_l, $l = 1, \cdots, L-1$. We set $n_0 := d$ and $n_L := p$, respectively the dimensions of the input and output of the network. The output from the lth layer will be denoted by

$$\nu^{(l)} := \sigma^{(l)} \circ \rho^{(l)} \circ \sigma^{(l-1)} \circ \rho^{(l-1)} \cdots \circ \sigma^{(1)} \circ \rho^{(1)},$$

i.e., it is a map $\nu^{(l)} : \mathbb{R}^d \to \mathbb{R}^{n_l}$. For convenience, we assume $\nu^{(0)}(x) := x$.

The affine function $\rho^{(l)} : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$ is given by a *weight* matrix $A^{(l)} \in \mathbb{Z}^{n_l \times n_{l-1}}$ and a *bias* vector $b^{(l)} \in \mathbb{R}^{n_l}$:

$$\rho^{(l)}(\nu^{(l-1)}) := A^{(l)}\nu^{(l-1)} + b^{(l)}.$$

The (i,j)th coordinate of $A^{(l)}$ will be denoted $a_{ij}^{(l)}$ and the ith coordinate of $b^{(l)}$ by $b_i^{(l)}$. Collectively they form the *parameters* of the lth layer.

For a vector input $x \in \mathbb{R}^{n_l}$, $\sigma^{(l)}(x)$ is understood to be in the coordinatewise sense; so $\sigma : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}$. We assume the final output of a neural network $\nu(x)$ is fed into a *score function* $s : \mathbb{R}^p \to \mathbb{R}^m$ that is application specific. When used as an m-category classifier, s may be chosen, for example, to be a soft-max or sigmoidal function. The score function is quite often regarded as the last layer of a neural network but this is purely a matter of convenience and we will not assume this. We will make the following mild assumptions on the architecture of our feedforward neural networks and explain next why they are indeed mild:

(a) the weight matrices $A^{(1)}, \dots, A^{(L)}$ are integer-valued;

(b) the bias vectors $b^{(1)}, \dots, b^{(L)}$ are real-valued;

(c) the activation functions $\sigma^{(1)}, \dots, \sigma^{(L)}$ take the form

$$\sigma^{(l)}(x) := \max\{x, t^{(l)}\},$$

where $t^{(l)} \in (\mathbb{R} \cup \{-\infty\})^{n_l}$ is called a *threshold* vector.

Henceforth all neural networks in our subsequent discussions will be assumed to satisfy (a)–(c).

(b) is completely general but there is also no loss of generality in (a), i.e., in restricting the weights $A^{(1)}, \dots, A^{(L)}$ from real matrices to integer matrices, as:

- real weights can be approximated arbitrarily closely by rational weights;
- one may then 'clear denominators' in these rational weights by multiplying them by the least common multiple of their denominators to obtain integer weights;
- keeping in mind that scaling all weights and biases by the same positive constant has no bearing on the workings of a neural network.

The activation function in (c) includes both ReLU activation ($t^{(l)} = 0$) and identity map ($t^{(l)} = -\infty$) as special cases. Aside from ReLU, our tropical framework will apply to piecewise linear activations such as leaky ReLU and absolute value, and with some extra effort, may be extended to max pooling, maxout nets, etc. But it does not, for example, apply to activations such as hyperbolic tangent and sigmoid.

In this work, we view an ReLU network as the simplest and most canonical model of a neural network, from which other variants that are more effective at specific tasks may be derived. Given that we seek general theoretical insights and not specific practical efficacy, it makes sense to limit ourselves to this simplest case. Moreover, ReLU networks already embody some of the most important elements (and mysteries) common to a wider range of neural networks (e.g., universal approximation, exponential expressiveness); they work well in practice and are often the go-to choice for feedforward networks. We are also not alone in limiting our discussions to ReLU networks (Montufar et al., 2014; Arora et al., 2018).

5. Tropical Algebra of Neural Networks

We now describe our tropical formulation of a multilayer feedforward neural network satisfying (a)–(c).

A multilayer feedforward neural network is generally nonconvex, whereas a tropical polynomial is always convex. Since most nonconvex functions are a difference of two convex functions (Hartman, 1959), a reasonable guess is that a feedforward neural network is the difference of two tropical polynomials, i.e., a tropical rational function. This is indeed the case, as we will see from the following.

Consider the output from the first layer in neural network

$$\nu(x) = \max\{Ax + b, \, t\},$$

where $A \in \mathbb{Z}^{p \times d}, b \in \mathbb{R}^p$, and $t \in (\mathbb{R} \cup \{-\infty\})^p$. We will decompose A as a difference of two nonnegative integer-valued matrices, $A = A_+ - A_-$ with $A_+, A_- \in \mathbb{N}^{p \times d}$; e.g., in the standard way with entries

$$a_{ij}^+ := \max\{a_{ij}, 0\}, \qquad a_{ij}^- := \max\{-a_{ij}, 0\}$$

respectively. Since

$$\max\{Ax + b, t\} = \max\{A_+x + b, \, A_-x + t\} - A_-x,$$

we see that every coordinate of one-layer neural network is a difference of two tropical polynomials. For networks with more layers, we apply this decomposition recursively to obtain the following result.

Proposition 5.1. *Let $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{R}^m$ be the parameters of the $(l+1)$th layer, and let $t \in (\mathbb{R} \cup \{-\infty\})^m$ be the threshold vector in the $(l+1)$th layer. If the nodes of the lth layer are given by tropical rational functions,*

$$\nu^{(l)}(x) = F^{(l)}(x) \oslash G^{(l)}(x) = F^{(l)}(x) - G^{(l)}(x),$$

i.e., each coordinate of $F^{(l)}$ and $G^{(l)}$ is a tropical polynomial in x, then the outputs of the preactivation and of the $(l+1)$th layer are given by tropical rational functions

$$\rho^{(l+1)} \circ \nu^{(l)}(x) = H^{(l+1)}(x) - G^{(l+1)}(x),$$
$$\nu^{(l+1)}(x) = \sigma \circ \rho^{(l+1)} \circ \nu^{(l)}(x) = F^{(l+1)}(x) - G^{(l+1)}(x)$$

respectively, where

$$\begin{aligned}
F^{(l+1)}(x) &= \max\{H^{(l+1)}(x),\, G^{(l+1)}(x) + t\},\\
G^{(l+1)}(x) &= A_+ G^{(l)}(x) + A_- F^{(l)}(x),\\
H^{(l+1)}(x) &= A_+ F^{(l)}(x) + A_- G^{(l)}(x) + b.
\end{aligned}$$

We will write $f_i^{(l)}$, $g_i^{(l)}$ and $h_i^{(l)}$ for the ith coordinate of $F^{(l)}$, $G^{(l)}$ and $H^{(l)}$ respectively. In tropical arithmetic, the recurrence above takes the form

$$\begin{aligned}
f_i^{(l+1)} &= h_i^{(l+1)} \oplus (g_i^{(l+1)} \odot t_i),\\
g_i^{(l+1)} &= \left[\bigodot_{j=1}^{n} (f_j^{(l)})^{a_{ij}^-}\right] \odot \left[\bigodot_{j=1}^{n} (g_j^{(l)})^{a_{ij}^+}\right],\\
h_i^{(l+1)} &= \left[\bigodot_{j=1}^{n} (f_j^{(l)})^{a_{ij}^+}\right] \odot \left[\bigodot_{j=1}^{n} (g_j^{(l)})^{a_{ij}^-}\right] \odot b_i.
\end{aligned} \tag{2}$$

Repeated applications of Proposition 5.1 yield the following.

Theorem 5.2 (Tropical characterization of neural networks)**.** *A feedforward neural network under assumptions (a)–(c) is a function $\nu : \mathbb{R}^d \to \mathbb{R}^p$ whose coordinates are tropical rational functions of the input, i.e.,*

$$\nu(x) = F(x) \oslash G(x) = F(x) - G(x)$$

where F and G are tropical polynomial maps. Thus ν is a tropical rational map.

Note that the tropical rational functions above have real coefficients, not integer coefficients. The integer weights $A^{(l)} \in \mathbb{Z}^{n_l \times n_{l-1}}$ have gone into the powers of tropical monomials in f and g, which is why we require our weights to be integer-valued, although as we have explained, this requirement imposes little loss of generality.

By setting $t^{(1)} = \cdots = t^{(L-1)} = 0$ and $t^{(L)} = -\infty$, we obtain the following corollary.

Corollary 5.3. *Let $\nu : \mathbb{R}^d \to \mathbb{R}$ be an ReLU activated feedforward neural network with integer weights and linear output. Then ν is a tropical rational function.*

A more remarkable fact is the converse of Corollary 5.3.

Theorem 5.4 (Equivalence of neural networks and tropical rational functions)**.**

(i) *Let $\nu : \mathbb{R}^d \to \mathbb{R}$. Then ν is a tropical rational function if and only if ν is a feedforward neural network satisfying assumptions (a)–(c).*

(ii) *A tropical rational function $f \oslash g$ can be represented as an L-layer neural network, with*

$$L \le \max\{\lceil \log_2 r_f \rceil,\, \lceil \log_2 r_g \rceil\} + 2,$$

where r_f and r_g are the number of monomials in the tropical polynomials f and g respectively.

We would like to acknowledge the precedence of (Arora et al., 2018, Theorem 2.1), which demonstrates the equivalence between ReLU-activated L-layer neural networks with *real* weights and d-variate continuous piecewise functions with *real* coefficients, where $L \le \lceil \log_2(d+1) \rceil + 1$.

By construction, a tropical rational function is a continuous piecewise linear function. The continuity of a piecewise linear function automatically implies that each of the pieces on which it is linear is a polyhedral region. As we saw in Section 3, a tropical polynomial $f : \mathbb{R}^d \to \mathbb{R}$ gives a tropical hypersurface that divides $\mathbb{R}^d$ into *convex* polyhedral regions defined by linear inequalities with integer coefficients: $\{x \in \mathbb{R}^d : Ax \le b\}$ with $A \in \mathbb{Z}^{m\times d}$ and $b \in \mathbb{R}^m$. A tropical rational function $f \oslash g : \mathbb{R}^d \to \mathbb{R}$ must also be a continuous piecewise linear function and divide $\mathbb{R}^d$ into polyhedral regions on each of which $f \oslash g$ is linear, although these regions are *nonconvex* in general. We will show the converse — any continuous piecewise linear function with integer coefficients is a tropical rational function.

Proposition 5.5. *Let $\nu : \mathbb{R}^d \to \mathbb{R}$. Then ν is a continuous piecewise linear function with integer coefficients if and only if ν is a tropical rational function.*

Corollary 5.3, Theorem 5.4, and Proposition 5.5 collectively imply the equivalence of

(i) tropical rational functions,

(ii) continuous piecewise linear functions with integer coefficients,

(iii) neural networks satisfying assumptions (a)–(c).

An immediate advantage of this characterization is that the set of tropical rational functions $\mathbb{T}(x_1, \ldots, x_d)$ has a semifield structure as we pointed out in Section 2, a fact that we have implicitly used in the proof of Proposition 5.5. However, what is more important is not the algebra but the *algebraic geometry* that arises from our tropical characterization. We will use tropical algebraic geometry to illuminate our understanding of neural networks in the next section.

The need to stay within tropical algebraic geometry is the reason we did not go for a simpler and more general characterization (that does not require the integer coefficients assumption). A *tropical signomial* takes the form

$$\varphi(x) = \bigoplus_{i=1}^{m} b_i \bigodot_{j=1}^{n} x_j^{a_{ij}},$$

where $a_{ij} \in \mathbb{R}$ and $b_i \in \mathbb{R} \cup \{-\infty\}$. Note that a_{ij} is not required to be integer-valued nor nonnegative. A *tropical rational signomial* is a tropical quotient $\varphi \oslash \psi$ of two tropical signomials φ, ψ. A *tropical rational signomial map* is a function $\nu = (\nu_1, \dots, \nu_p) : \mathbb{R}^d \to \mathbb{R}^p$ where each $\nu_i : \mathbb{R}^d \to \mathbb{R}$ is a tropical rational signomial $\nu_i = \varphi_i \oslash \psi_i$. The same argument we used to establish Theorem 5.2 gives us the following.

Proposition 5.6. *Every feedforward neural network with ReLU activation is a tropical rational signomial map.*

Nevertheless tropical signomials fall outside the realm of tropical algebraic geometry and we do not use Proposition 5.6 in the rest of this article.

6. Tropical Geometry of Neural Networks

Section 5 defines neural networks via tropical algebra, a perspective that allows us to study them via tropical algebraic geometry. We will show that the decision boundary of a neural network is a subset of a tropical hypersurface of a corresponding tropical polynomial (Section 6.1). We will see that, in an appropriate sense, zonotopes form the geometric building blocks for neural networks (Section 6.2). We then prove that the geometry of the function represented by a neural network grows vastly more complex as its number of layers increases (Section 6.3).

6.1. Decision Boundaries of a Neural Network

We will use tropical geometry and insights from Section 5 to study decision boundaries of neural networks, focusing on the case of two-category classification for clarity. As explained in Section 4, a neural network $\nu : \mathbb{R}^d \to \mathbb{R}^p$ together with a choice of score function $s : \mathbb{R}^p \to \mathbb{R}$ give us a classifier. If the output value $s(\nu(x))$ exceeds some decision threshold c, then the neural network predicts x is from one class (e.g., x is a CAT image), and otherwise x is from the other category (e.g., a DOG image). The input space is thereby partitioned into two disjoint subsets by the *decision boundary* $\mathcal{B} := \{x \in \mathbb{R}^d : \nu(x) = s^{-1}(c)\}$. Connected regions with value above the threshold and connected regions with value below the threshold will be called the *positive regions* and *negative regions* respectively.

We provide bounds on the number of positive and negative regions and show that there is a tropical polynomial whose tropical hypersurface contains the decision boundary.

Proposition 6.1 (Tropical geometry of decision boundary)**.** *Let $\nu : \mathbb{R}^d \to \mathbb{R}$ be an L-layer neural network satisfying assumptions (a)–(c) with $t^{(L)} = -\infty$. Let the score function $s : \mathbb{R} \to \mathbb{R}$ be injective with decision threshold c in its range. If $\nu = f \oslash g$ where f and g are tropical polynomials, then*

(i) *its decision boundary $\mathcal{B} = \{x \in \mathbb{R}^d : \nu(x) = s^{-1}(c)\}$ divides $\mathbb{R}^d$ into at most $\mathcal{N}(f)$ connected positive regions and at most $\mathcal{N}(g)$ connected negative regions;*

(ii) *its decision boundary is contained in the tropical hypersurface of the tropical polynomial $s^{-1}(c) \odot g(x) \oplus f(x) = \max\{f(x),\, g(x) + s^{-1}(c)\}$, i.e.,*

$$\mathcal{B} \subseteq \mathcal{T}(s^{-1}(c) \odot g \oplus f). \tag{3}$$

The function $s^{-1}(c) \odot g \oplus f$ is not necessarily linear on every positive or negative region and so its tropical hypersurface $\mathcal{T}(s^{-1}(c) \odot g \oplus f)$ may further divide a positive or negative region derived from $\mathcal{B}$ into multiple linear regions. Hence the "$\subseteq$" in (3) cannot in general be replaced by "$=$".

6.2. Zonotopes as Geometric Building Blocks of Neural Networks

From Section 3, we know that the number of regions a tropical hypersurface $\mathcal{T}(f)$ divides the space into equals the number of vertices in the dual subdivision of the Newton polygon associated with the tropical polynomial f. This allows us to bound the number of linear regions of a neural network by bounding the number of vertices in the dual subdivision of the Newton polygon.

We start by examining how geometry changes from one layer to the next in a neural network, more precisely:

Question. *How are the tropical hypersurfaces of the tropical polynomials in the $(l+1)$th layer of a neural network related to those in the lth layer?*

The recurrent relation (2) describes how the tropical polynomials occurring in the $(l+1)$th layer are obtained from those in the lth layer, namely, via three operations: tropical sum, tropical product, and tropical powers. Recall that a tropical hypersurface of a tropical polynomial is dual to the dual subdivision of the Newton polytope of the tropical polynomial, which is given by the projection of the upper faces on the polytopes defined by (1). Hence the question boils down to how these three operations transform the polytopes, which is addressed in Propositions 3.1 and 3.2. We follow notations in Proposition 5.1 for the next result.

Lemma 6.2. *Let $f_i^{(l)}$, $g_i^{(l)}$, $h_i^{(l)}$ be the tropical polynomials produced by the ith node in the lth layer of a neural network,* *i.e., they are defined by* (2). *Then* $\mathcal{P}\big(f_i^{(l)}\big)$, $\mathcal{P}\big(g_i^{(l)}\big)$, $\mathcal{P}\big(h_i^{(l)}\big)$ *are subsets of* $\mathbb{R}^{d+1}$ *given as follows:*

(i) $\mathcal{P}\big(g_i^{(1)}\big)$ *and* $\mathcal{P}\big(h_i^{(1)}\big)$ *are points.*

(ii) $\mathcal{P}\big(f_i^{(1)}\big)$ *is a line segment.*

(iii) $\mathcal{P}\big(g_i^{(2)}\big)$ *and* $\mathcal{P}\big(h_i^{(2)}\big)$ *are zonotopes.*

(iv) *For* $l \geq 1$,

$$\mathcal{P}\big(f_i^{(l)}\big) = \operatorname{Conv}\big[\mathcal{P}\big(g_i^{(l)} \odot t_i^{(l)}\big) \cup \mathcal{P}\big(h_i^{(l)}\big)\big]$$

if $t_i^{(l)} \in \mathbb{R}$, *and* $\mathcal{P}\big(f_i^{(l)}\big) = \mathcal{P}\big(h_i^{(l)}\big)$ *if* $t_i^{(l)} = -\infty$.

(v) *For* $l \geq 1$, $\mathcal{P}\big(g_i^{(l+1)}\big)$ *and* $\mathcal{P}\big(h_i^{(l+1)}\big)$ *are weighted Minkowski sums,*

$$\mathcal{P}\big(g_i^{(l+1)}\big) = \sum_{j=1}^{n_l} a_{ij}^{-}\mathcal{P}\big(f_j^{(l)}\big) + \sum_{j=1}^{n_l} a_{ij}^{+}\mathcal{P}\big(g_j^{(l)}\big),$$

$$\mathcal{P}\big(h_i^{(l+1)}\big) = \sum_{j=1}^{n_l} a_{ij}^{+}\mathcal{P}\big(f_j^{(l)}\big) + \sum_{j=1}^{n_l} a_{ij}^{-}\mathcal{P}\big(g_j^{(l)}\big) + \{b_i e\},$$

where a_{ij}, b_i *are entries of the weight matrix* $A^{(l+1)} \in \mathbb{Z}^{n_{l+1}\times n_l}$ *and bias vector* $b^{(l+1)} \in \mathbb{R}^{n_{l+1}}$, *and* $e := (0,\dots,0,1) \in \mathbb{R}^{d+1}$.

A conclusion of Lemma 6.2 is that zonotopes are the building blocks in the tropical geometry of neural networks. Zonotopes are studied extensively in convex geometry and, among other things, are intimately related to hyperplane arrangements (Greene & Zaslavsky, 1983; Guibas et al., 2003; McMullen, 1971; Holtz & Ron, 2011). Lemma 6.2 connects neural networks to this extensive body of work but its full implication remains to be explored. In Section C.2 of the supplement, we show how one may build these polytopes for a two-layer neural network.

6.3. Geometric Complexity of Deep Neural Networks

We apply the tools in Section 3 to study the complexity of a neural network, showing that a deep network is much more expressive than a shallow one. Our measure of complexity is geometric: we will follow (Montufar et al., 2014; Raghu et al., 2017) and use the number of linear regions of a piecewise linear function $\nu : \mathbb{R}^d \to \mathbb{R}^p$ to measure the complexity of ν.

We would like to emphasize that our upper bound below does not improve on that obtained in (Raghu et al., 2017) — in fact, our version is more restrictive given that it applies only to neural networks satisfying (a)–(c). Nevertheless our goal here is to demonstrate how tropical geometry may be used to derive the same bound.

Theorem 6.3. *Let* $\nu : \mathbb{R}^d \to \mathbb{R}$ *be an* L*-layer real-valued feedforward neural network satisfying* (a)–(c). *Let* $t^{(L)} = -\infty$ *and* $n_l \geq d$ *for all* $l = 1,\dots,L-1$. *Then* $\nu = \nu^{(L)}$ *has at most*

$$\prod_{l=1}^{L-1}\sum_{i=0}^{d}\binom{n_l}{i}$$

linear regions. In particular, if $d \leq n_1,\dots,n_{L-1} \leq n$, *the number of linear regions of* ν *is bounded by* $\mathcal{O}\big(n^{d(L-1)}\big)$.

Proof. If $L = 2$, this follows directly from Lemma 6.2 and Corollary 3.4. The case of $L \geq 3$ is in Section D.7 in the supplement. □

As was pointed out in (Raghu et al., 2017), this upper bound closely matches the lower bound $\Omega\big((n/d)^{(L-1)d}n^d\big)$ in (Montufar et al., 2014, Corollary 5) when $n_1 = \cdots = n_{L-1} = n \geq d$. Hence we surmise that the number of linear regions of the neural network grows polynomially with the width n and exponentially with the number of layers L.

7. Conclusion

We argue that feedforward neural networks with rectified linear units are, modulo trivialities, nothing more than tropical rational maps. To understand them we often just need to understand the relevant tropical geometry.

In this article, we took a first step to provide a proof-of-concept: questions regarding decision boundaries, linear regions, how depth affect expressiveness, etc, can be translated into questions involving tropical hypersurfaces, dual subdivision of Newton polygon, polytopes constructed from zonotopes, etc.

As a new branch of algebraic geometry, the novelty of tropical geometry stems from both the algebra and geometry as well as the interplay between them. It has connections to many other areas of mathematics. Among other things, there is a tropical analogue of linear algebra (Butkovič, 2010) and a tropical analogue of convex geometry (Gaubert & Katz, 2006). We cannot emphasize enough that we have only touched on a small part of this rich subject. We hope that further investigation from this tropical angle might perhaps unravel other mysteries of deep neural networks.

Acknowledgments

The authors thank Ralph Morrison, Yang Qi, Bernd Sturmfels, and the anonymous referees for their very helpful comments. The work in this article is generously supported by DARPA D15AP00109, NSF IIS 1546413, the Eckhardt Faculty Fund, and a DARPA Director's Fellowship.

References

Arora, R., Basu, A., Mianjy, P., and Mukherjee, A. Understanding deep neural networks with rectified linear units. *International Conference on Learning Representations*, 2018.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*, 2014.

Bengio, Y. and Delalleau, O. On the expressive power of deep architectures. In *International Conference on Algorithmic Learning Theory*, pp. 18–36. Springer, 2011.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge University Press, 2004.

Butkovič, P. *Max-linear systems: theory and algorithms*. Springer Science & Business Media, 2010.

Delalleau, O. and Bengio, Y. Shallow vs. deep sum-product networks. In *Advances in Neural Information Processing Systems*, pp. 666–674, 2011.

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

Gaubert, S. and Katz, R. Max-plus convex geometry. In *International Conference on Relational Methods in Computer Science*. Springer, 2006.

Greene, C. and Zaslavsky, T. On the interpretation of whitney numbers through arrangements of hyperplanes, zonotopes, non-radon partitions, and orientations of graphs. *Transactions of the American Mathematical Society*, 280 (1):97–126, 1983.

Gritzmann, P. and Sturmfels, B. Minkowski addition of polytopes: computational complexity and applications to gröbner bases. *SIAM Journal on Discrete Mathematics*, 6(2):246–269, 1993.

Guibas, L. J., Nguyen, A., and Zhang, L. Zonotopes as bounding volumes. In *Proceedings of 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 803–812. SIAM, 2003.

Hartman, P. On functions representable as a difference of convex functions. *Pacific Journal of Mathematics*, 9(3): 707–713, 1959.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

Holtz, O. and Ron, A. Zonotopal algebra. *Advances in Mathematics*, 227(2):847–894, 2011.

Itenberg, I., Mikhalkin, G., and Shustin, E. I. *Tropical algebraic geometry*, volume 35. Springer Science & Business Media, 2009.

Kalchbrenner, N. and Blunsom, P. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1700–1709, 2013.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.

Maclagan, D. and Sturmfels, B. *Introduction to tropical geometry*, volume 161. American Mathematical Society, 2015.

McMullen, P. On zonotopes. *Transactions of the American Mathematical Society*, 159:91–109, 1971.

Montufar, G. F., Pascanu, R., Cho, K., and Bengio, Y. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 2924–2932, 2014.

Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. Exponential expressivity in deep neural networks through transient chaos. In *Advances in Neural Information Processing Systems*, pp. 3360–3368, 2016.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Sohl-Dickstein, J. On the expressive power of deep neural networks. In *International Conference on Machine Learning*, pp. 2847–2854, 2017.

Tao, P. D. and Hoai An, L. T. The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. *Annals of Operations Research*, 133(1–4):23–46, 2005.

Telgarsky, M. benefits of depth in neural networks. In *Conference on Learning Theory*, 2016.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv:1611.03530*, 2016.