IMPERIAL

A Tropical Geometric Perspective on Deep Learning

Shiv Bhatia[1,2], **Yueqi Cao**[1], **Paul Lezeau**[1,3], **Anthea Monod**[2]

Tropical Geometry

The **tropical semiring** is $\overline{\mathbb{R}} = (\mathbb{R} \cup \{\infty\}, \oplus, \odot)$ where $a \oplus b = \max(a, b)$ and $a \odot b = a + b$.

Algebraic geometry studies zero sets of polynomial equations. **Tropical geometry focuses on analogous structures defined by tropical polynomials**, i.e.,

$$\bigoplus_n a_n t^n$$

where n ranges over a finite subset of $\mathbb{N}^m$.

We can study more general formal expressions. For instance: tropical division is given by $a \oslash b = a - b$ and quotients of tropical polynomials are tropical rational functions. Tropical exponentiation corresponds to usual multiplication, so tropical exponentiation makes sense for arbitrary real exponents: in particular, tropical polynomials with arbitrary real exponents are functions on $\mathbb{R}$.

Neural Networks

Neural networks are a way of obtaining complicated families of functions by composing linear maps and elementary nonlinear functions.

Formally, a neural network is a function $\mathbb{R}^m \to \mathbb{R}^\ell$

$$\sigma \circ L_d \circ \sigma \circ L_{i-1} \circ \cdots \circ L_1$$

where the L_i are affine maps and $\sigma(t) = \max(0, t)$.

Any neural network can be written in terms of tropical algebra: the nonlinear activation σ is given by tropical addition; applying a linear map can be expressed using tropical multiplication and exponentiation.

Neural networks are given by composing functions of the form $\max(Ax + b, t)$ (applied component-wise on vectors), so we can represent neural networks tropically by finding a tropical expression for $\max(Ax + b, t)$. For computational reasons, it is useful to restrict to expressions with nonnegative exponents.

Write $A^+ = \max(A, 0)$ and $A^- = \max(-A, 0)$. Then we have

$$\max(Ax + b, t) = \max(A^+x + b, A^-x + t) - A^-x,$$

and the RHS can be written as a vector of tropical rational functions in x (with nonnegative real exponents), where the ith entry is

$$\left(b_i \odot \bigodot_j x_j^{a_{ij}^+} \oplus t_i \odot \bigodot_j x_j^{a_{ij}^-}\right) \oslash \bigodot_j x_j^{a_{ij}^-}.$$

We get a tropical expression for any neural network by composing such expressions.

Linear Regions

Neural networks are locally given by linear functions. Intuitively, complicated neural networks should have complicated representations in terms of linear maps. To quantify this representation, we count the number of linear regions of a neural network, i.e., the number of maximal connected regions of the input space on which the neural network is linear.

Any neural network with 1-dimensional output takes the form $f \oslash g$ where f, g are tropical polynomials with real nonnegative exponents. If $f = \bigoplus_n a_n T^n$ and $g = \bigoplus_n b_n T^n$ then the activation of the neural network at an input x is given by the difference

$$\max_n\{a_n + \sum_i n_i x_i\} - \max_n\{b_n + \sum_i n_i x_i\}.$$

The linear regions of the neural network correspond to regions where both the numerator and the denominator are given by one of the terms in the maxima.

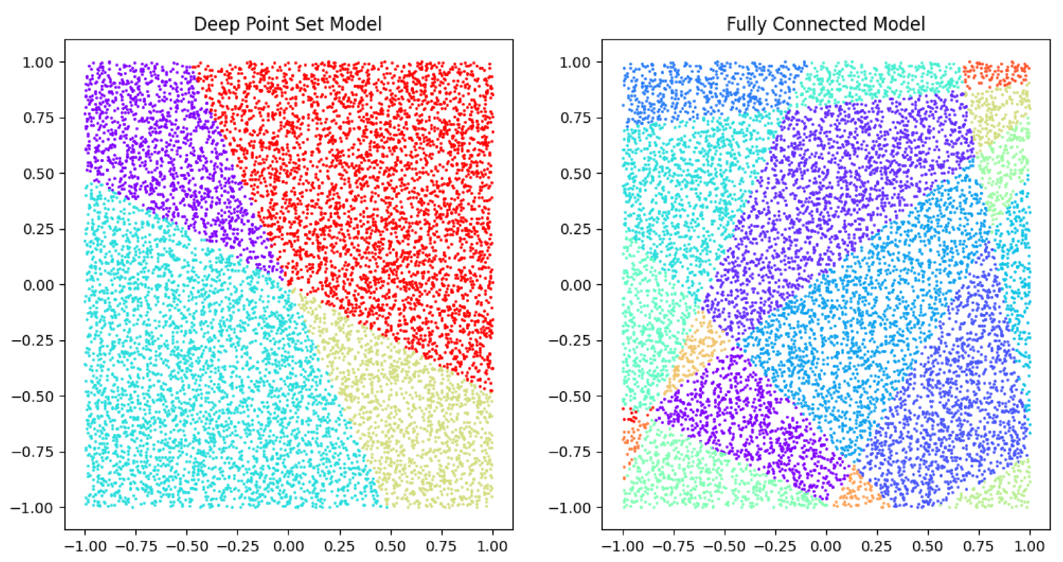


Figure 1: By sampling points and classifying them by their Jacobians, we obtain a visualization of the linear regions. Note that the deep set architecture is symmetric and this is reflected in its linear regions.

The linear regions of a tropical polynomial with positive real exponents are polyhedra in $\mathbb{R}^m$ and the linear region of tropical quotients of such function are unions of polyhedra. This means that **polyhedral geometry can be used to understand the geometry of the linear regions.** Computatioanlly, the connection to polyhedral geometry opens up the use of existing computational tools.

Estimating Linear Regions

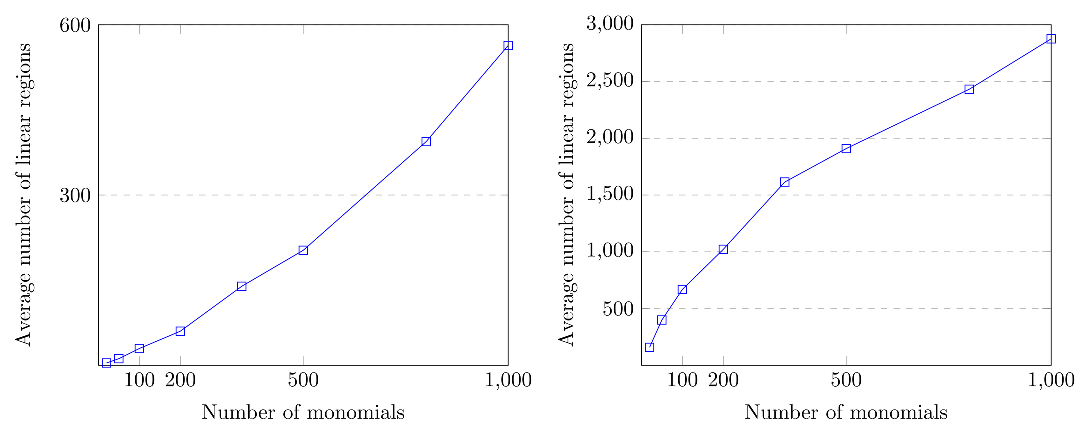


Figure 2: Linear regions of neural network in tropical form. The two curves show the number of linear regions versus number of monomials in tropical Puiseux rational functions in 3 variables (Left) and 4 variables (Right).

We compute the number of linear regions of a neural network in two ways: (1) we make use the extensive polyhedral geometry library in **OSCAR** for symbolic computation for linear regions of tropical polynomials; (2) we can classify points into linear regions by finding the **Jacobian matrix** at points in the input domain.

Hoffman Constants

To estimate the number of linear regions of a neural network, **we sample points in the input domain and determine in which region each point lies**. To guarantee that each linear region has been hit, we need to ensure the sampling domain is large enough. At the same time, we want the sampling domain to be as small as possible for efficient sampling. **In other words, we want the smallest ball from which we can sample a point in each linear region.**

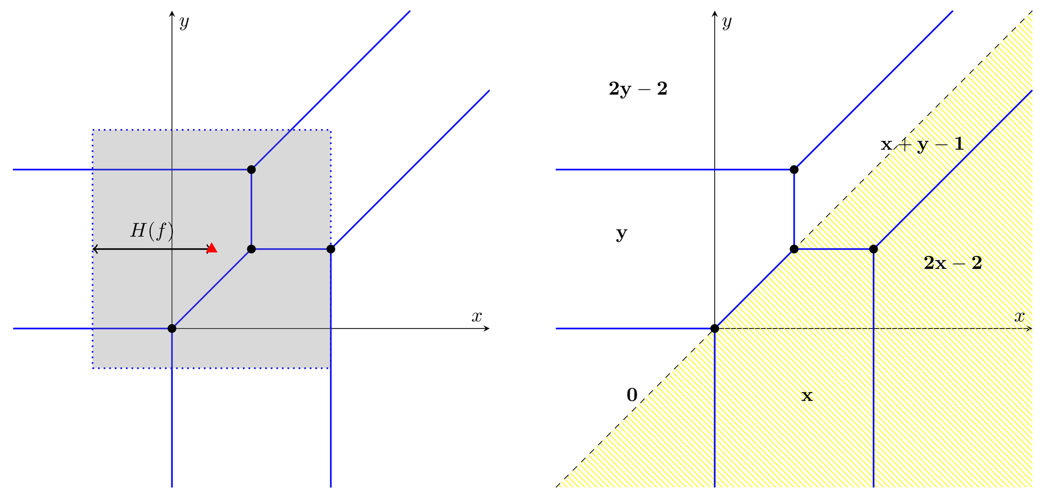
Consider a nonempty polyhedron $P(A, b) = \{x \in \mathbb{R}^n : Ax \leq b\}$. Let $d(u, P(A, b))$ denote the distance of u to the polyhedron measured under an arbitrary norm $\|\cdot\|$ on $\mathbb{R}^n$. Then there exists a constant $H(A)$ only depending on A such that

$$d(u, P_{A,b}) \leq H(A)\|(Au - b)_+\|$$

where $x_+ = \max\{x, 0\}$ is applied coordinate-wise. The constant $H(A)$ is called the Hoffman constant of A.

The Hoffman constant of a tropical rational function $f = p \oslash q$ is defined as the maximal Hoffman constant over its linear regions. Let $\check{f}$ be the min-conjugate of f. For any $x \in \mathbb{R}^n$, we have the following inequality relating the radius of sampling domain and Hoffman constant:

$$R_f(x) \leq H(p \oslash q) \max\{p(x) - \check{p}(x),\, q(x) - \check{q}(x)\}.$$



(a) Illustration of the Hoffman constant of a tropical polynomial f. The Hoffman constant $H(f)$ is characterized by the property that for any point $x \in \mathbb{R}^n$, the ball/square of radius R intersecting all linear regions of f is bounded by $H(f)$ and the function value at x.

(b) Illustration of symmetry of linear regions. The tropical polynomial is symmetric under group action $\mathbb{Z}_2$, which is equivalent to reflection along $y = x$. It suffices to count linear regions inside the fundamental domain $y \leq x$ which is shaded in yellow.

Symmetries

Symmetries of neural networks induce symmetries of linear regions. We can leverage these to estimate the number of linear regions with less samples.

If a neural network is invariant under a group action, then we can restrict sampling to a fundamental domain of this group action. **For S_n-invariant networks, this reduces the number of samples needed by a factor of $n!$.**

References

1. Shiv Bhatia, Yueqi Cao, Paul Lezeau, Anthea Monod (2024). “Tropical Expressivity of Neural Networks”, arXiv: 2405.20174.
2. Liwen Zhang, Naitzat Gregory, Lek-Heng Lim (2018). “Tropical geometry of deep neural networks”, International Conference on Machine Learning.
3. The OSCAR Team (2023) OSCAR – Open Source Computer Algebra Research system, Version 0.12.1-DEV.

Affiliations

[1] Department of Mathematics, Imperial College London
[2] Department of Computing, Imperial College London
[3] London School of Geometry and Number Theory