Alternating Minimization for Regression with Tropical Rational Functions

Alex Dunbar* Lars Ruthotto†

Abstract

We propose an alternating minimization heuristic for regression over the space of tropical rational functions with fixed exponents. The method alternates between fitting the numerator and denominator terms via tropical polynomial regression, which is known to admit a closed form solution. We demonstrate the behavior of the alternating minimization method experimentally. Experiments demonstrate that the heuristic provides a reasonable approximation of the input data. Our work is motivated by applications to ReLU neural networks, a popular class of network architectures in the machine learning community which are closely related to tropical rational functions.

1 Introduction

Tropical algebra uses a semiring structure on $\mathbb{R} \cup \{-\infty\}$ where the tropical sum of two elements is their maximum and tropical multiplication is standard addition. In this setting, n-variable tropical polynomials are functions that are the pointwise maximum of finitely many affine functions with slopes in a finite set $W \subseteq \mathbb{Z}^n_{\geq 0}$. Such functions are piecewise linear and convex. Tropical rational functions are the standard difference between two tropical polynomials and therefore continuous piecewise linear functions.

In this paper, we are interested in fitting tropical rational functions to data and developing a numerical method for solving regression problems for this function class. Specifically, we consider the ℓ^∞ regression problem over tropical rational functions with exponents in a fixed finite set $W = \{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \dots, \mathbf{w}^{(D)}\} \subseteq \mathbb{Z}^n_{\geq 0}$: Given a dataset $\mathcal{D} = \left\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\right\} \subseteq \mathbb{R}^n \times \mathbb{R}$, find

$$\arg\min_{\mathbf{p},\mathbf{q}} \left\| \begin{bmatrix} \max_{1\leq i\leq D} \left(\langle \mathbf{w}^{(i)}, \mathbf{x}^{(1)}\rangle + p_i\right) \\ \max_{1\leq i\leq D} \left(\langle \mathbf{w}^{(i)}, \mathbf{x}^{(2)}\rangle + p_i\right) \\ \vdots \\ \max_{1\leq i\leq D} \left(\langle \mathbf{w}^{(i)}, \mathbf{x}^{(N)}\rangle + p_i\right) \end{bmatrix} - \begin{bmatrix} \max_{1\leq i\leq D} \left(\langle \mathbf{w}^{(i)}, \mathbf{x}^{(1)}\rangle + q_i\right) \\ \max_{1\leq i\leq D} \left(\langle \mathbf{w}^{(i)}, \mathbf{x}^{(2)}\rangle + q_i\right) \\ \vdots \\ \max_{1\leq i\leq D} \left(\langle \mathbf{w}^{(i)}, \mathbf{x}^{(N)}\rangle + q_i\right) \end{bmatrix} - \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix} \right\|_\infty, \tag{1}$$

where the coefficient vectors $\mathbf{p} = \begin{bmatrix} p_1 & p_2 & \dots & p_D \end{bmatrix}^\top$ and $\mathbf{q} = \begin{bmatrix} q_1 & q_2 & \dots & q_D \end{bmatrix}^\top$ define the tropical polynomials $p(\mathbf{x}) = \max_{1\leq i\leq D} \left(\langle \mathbf{w}^{(i)}, \mathbf{x}\rangle + p_i\right)$ and $q(\mathbf{x}) = \max_{1\leq i\leq D} \left(\langle \mathbf{w}^{(i)}, \mathbf{x}\rangle + q_i\right)$, respectively. Since optimizing both vectors simultaneously is difficult, we propose a heuristic for the solution of (1) that alternates between solving the tropical polynomial regression problem of finding the optimal vector $\mathbf{p}$ for fixed $\mathbf{q}$ and the similar problem of finding the optimal $\mathbf{q}$ given fixed $\mathbf{p}$. Alternating methods have a rich history in optimization; see e.g., [4, Chapter 14] for an overview and [1, 39] for applications to structured problems.

Our proposed heuristic alternates between updating $\mathbf{p}$ and $\mathbf{q}$ using results from tropical polynomial regression; see, e.g., [2, 26, 27]. In each substep, we leverage the algebraic structure of tropical polynomials and use the fact that the closed-form solution involves only (min-plus and max-plus) matrix-vector products and vector addition. This renders each iteration of our heuristic computationally cheap. Geometrically, our proposed heuristic searches the nondifferentiability locus of the ℓ^∞ loss in a way such that the loss is nonincreasing. In our experiments, this heuristic provides a reasonable approximation of the input data.

Problem (1) fits into the framework of piecewise linear regression. More specifically, a tropical rational function is a difference of convex functions. Difference of convex functions are known to be widely expressive

*Department of Mathematics, Emory University, Atlanta, Georgia, USA. (alex.dunbar@emory.edu)
†Departments of Mathematics and Computer Science, Emory University, Atlanta, Georgia, USA. (lruthotto@emory.edu)

[15]. Piecewise linear regression problems have received some attention from the optimization community [19, 24, 36] and have recently seen great interest from the deep learning community. In this setting, piecewise linear functions are commonly parametrized using *ReLU neural networks*, functions which are expressible as the repeated compositions of affine transformations with a ReLU activation function $\sigma(\mathbf{x}) = \max(\mathbf{x}, 0)$; see, e.g., [3, 9, 29]. Such functions have proven to be very expressive, and the optimization problem over ReLU networks, although being plagued by non-convexity and non-smoothness, can often be solved to a reasonable accuracy with variants of stochastic gradient descent.

Recent work [6, 25, 43] has shown that ReLU neural networks correspond to tropical algebraic objects. Precisely, if a function can be represented by a ReLU neural network with integral weights, then it also has a representation as a tropical rational function [43]. This connection has been leveraged to analyze the complexity of a neural network by counting its linear regions [6, 43], minimize trained networks [33, 34], and extract linear regions of a trained network [38]. These works use tropical methods to better understand the theoretical properties of neural networks. By considering regression problems over the space of tropical rational functions, we begin to apply tropical methods to understand the training problem for ReLU networks. However, the exact correspondence between network architecture and the function a network represents is poorly understood, limiting a direct translation from the tropical setting.

Piecewise linear regression utilizing a parametrization through max-plus algebra is studied in [19, 36], where the optimization problem is interpreted through mixed integer programming. In these works, the authors allow the set W to vary in $\mathbb{R}^n$ during the optimization. Our approach differs in that we fix W and use the ℓ^∞ norm as an objective function, allowing the heuristic to utilize the algebraic structure of tropical polynomials.

Concurrent work by Krivulin [21, 22] views problem (1) in terms of two-sided tropical linear systems to independently derive the alternating method. Specifically, the method is used to solve two-sided tropical linear systems in [22] and applied to regression problems in [21]. In [21], the author notes that because the error is nonincreasing and bounded below (compare with Proposition 3.1), the error at each iteration converges to some optimal error. Additionally, [21] presents a sampling approach to choosing monomials in numerical experiments for univariate regression problems. Our numerical experiments in Section 4 suggest that the problem becomes substantially more difficult in higher dimensional settings.

The remainder of the paper is organized as follows: Section 2 reviews the relevant background from tropical algebra and ReLU neural networks. Section 3 presents the alternating algorithm for tropical regression. Section 4 details numerical experiments with tropical rational regression. Finally, Section 5 presents concluding remarks and directions for future work.

2 Background

In this section, we review relevant background from tropical algebra, tropical polynomial regression, and ReLU neural networks. We adopt the following notational conventions: Bold lowercase letters denote vectors and bold uppercase letters denote matrices. If $\mathbf{x}$ is a vector, x_i is the i^{th} component of $\mathbf{x}$. The standard inner product of vectors is denoted $\langle \cdot, \cdot \rangle$. Collections of vectors are indexed by superscripts in parentheses. Sequences are denoted by superscripts without parentheses. The all ones vector is denoted $\mathbf{1}$.

2.1 Tropical Algebra

This section briefly recalls relevant ideas and notation from tropical algebra. A more thorough introduction can be found in [5] and a standard reference is [23]. Tropical geometry has recently seen applications outside of algebraic geometry in optimization and statistics [10, 18, 30, 35, 42]. The survey article [25] provides an overview of applications of tropical geometry in machine learning.

The main object of study in tropical algebra is the *tropical semiring* $\mathbb{T} := (\mathbb{R} \cup \{-\infty\}, \oplus, \odot)$. Tropical addition is $a \oplus b = \max(a, b)$ and tropical multiplication is $a \odot b = a + b$. Tropical addition and tropical multiplication are both associative and commutative. The multipicative identity is 0 and every finite element a has a tropical multiplicative inverse $-a$. The additive identity is $-\infty$ and no element of $\mathbb{T}$ has an additive inverse. Tropical exponentials are repeated tropical multiplication and denoted $a^{\odot w} := wa$ for $w \in \mathbb{Z}$.

Given a collection of n variables $x_1, x_2, \ldots, x_n$, we use multi-index notation to describe the *tropical monomial*

$$c \odot \mathbf{x}^{\odot \mathbf{w}} := c + \langle \mathbf{w}, \mathbf{x} \rangle = c + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n.$$

Analogously to standard algebra, the *tropical polynomials* in n variables are defined as finite sums of tropical monomials $\mathbf{x}^{\odot \mathbf{w}}$. That is,

$$\mathbb{T}[x_1, x_2, \ldots, x_n] := \left\{ \bigoplus_{\mathbf{w} \in W} c_{\mathbf{w}} \odot \mathbf{x}^{\odot \mathbf{w}} \middle| c_{\mathbf{w}} \in \mathbb{T}, W \subseteq \mathbb{Z}^n_{\geq 0} \text{ finite} \right\} = \left\{ \max_{\mathbf{w} \in W} (c_{\mathbf{w}} + \langle \mathbf{w}, \mathbf{x} \rangle) \middle| c_{\mathbf{w}} \in \mathbb{T}, W \subseteq \mathbb{Z}^n_{\geq 0} \text{ finite} \right\}.$$

The set of tropical polynomials is also a semiring with operations extending tropical addition and tropical multiplication. Tropical polynomials are convex piecewise linear functions.

Finally, *tropical rational functions* are functions which are tropical quotients of tropical polynomials,

$$\mathbb{T}(x_1, x_2, \ldots, x_n) := \{ p(\mathbf{x}) \oslash q(\mathbf{x}) \mid p, q \in \mathbb{T}[x_1, x_2, \ldots, x_n] \} = \{ p(\mathbf{x}) - q(\mathbf{x}) \mid p, q \in \mathbb{T}[x_1, x_2, \ldots, x_n] \}.$$

Tropical rational functions are piecewise linear but not necessarily convex. However, they are the difference of convex functions. For a fixed set $W \subseteq \mathbb{Z}^n_{\geq 0}$, we use $\mathbb{T}[\mathbf{x}]_W$ to denote the tropical polynomials with exponents in W. Similarly, $\mathbb{T}(\mathbf{x})_W$ denotes the tropical rational functions with exponents in W. When $W = \{\mathbf{w} \in \mathbb{Z}^n | \, 0 \leq w_i \leq d \text{ for } i = 1, 2, \ldots, n\}$, we say that functions $f \in \mathbb{T}(\mathbf{x})_W$ have *degree* d. We adopt the notation $\mathbb{T}[\mathbf{x}]$ and $\mathbb{T}(\mathbf{x})$ to emphasize the connections with commutative algebra.

Tropical Linear Algebra Many concepts from classical linear algebra over a field generalize to $\mathbb{T}^n$. Given $\mathbf{u}, \mathbf{v} \in \mathbb{T}^n$, define vector addition as the componentwise maximum

$$\mathbf{u} \oplus \mathbf{v} = \max(\mathbf{u}, \mathbf{v}) := \begin{bmatrix} \max(u_1, v_1) & \max(u_2, v_2) & \cdots & \max(u_n, v_n) \end{bmatrix}^\top.$$

For $\lambda \in \mathbb{T}$ and $\mathbf{u} \in \mathbb{T}^n$, define scalar multiplication as

$$\lambda \odot \mathbf{u} := \lambda \mathbf{1} + \mathbf{u} = \begin{bmatrix} u_1 + \lambda & u_2 + \lambda & \cdots & u_n + \lambda \end{bmatrix}^\top.$$

As in classical linear algebra, maps $\mathbb{T}^n \to \mathbb{T}^m$ which are compatible with the vector addition and scalar multiplication on $\mathbb{T}^n$ can be represented by matrix multiplication [2]. Given an $m \times n$ matrix $\mathbf{A}$ and a vector $\mathbf{u} \in \mathbb{T}^n$, *max-plus matrix-vector multiplication* is defined as

$$\mathbf{A} \boxplus \mathbf{x} := \begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \ldots & a_{m,n} \end{bmatrix} \boxplus \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} \max_{1 \leq j \leq n}(a_{1,j} + u_j) \\ \max_{1 \leq j \leq n}(a_{2,j} + u_j) \\ \vdots \\ \max_{1 \leq j \leq n}(a_{m,j} + u_j) \end{bmatrix}.$$

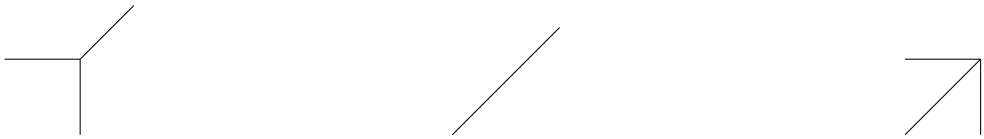
We also work with the dual *min-plus matrix-vector multiplication*

$$\mathbf{A} \boxplus' \mathbf{x} := \begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \ldots & a_{m,n} \end{bmatrix} \boxplus' \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} \min_{1 \leq j \leq n}(a_{1,j} + u_j) \\ \min_{1 \leq j \leq n}(a_{2,j} + u_j) \\ \vdots \\ \min_{1 \leq j \leq n}(a_{m,j} + u_j) \end{bmatrix}.$$

In our setting, we will frequently identify a tropical polynomial $g \in \mathbb{T}[\mathbf{x}]_W$ with its vector of coefficients $\mathbf{g} = (g_{\mathbf{w}})_{\mathbf{w} \in W} \in \mathbb{T}^{|W|}$.

Tropical Hypersurfaces In classical algebraic geometry, the zero set of a polynomial is called a *hypersurface*. The tropical analog for a tropical polynomial $p \in \mathbb{T}[x_1, x_2, \ldots, x_n]$ given by $p(\mathbf{x}) = \max_{\mathbf{w} \in W}(\langle \mathbf{w}, \mathbf{x} \rangle + p_{\mathbf{w}})$ is the *tropical hypersurface*

$$\begin{aligned} \mathcal{V}(p) &:= \{\mathbf{x} \in \mathbb{R}^n \mid p(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + p_{\mathbf{w}} = \langle \mathbf{v}, \mathbf{x} \rangle + p_{\mathbf{v}} \text{ for some } \mathbf{w} \neq \mathbf{v} \in W\} \\ &= \{\mathbf{x} \in \mathbb{R}^n \mid p \text{ is not differentiable at } \mathbf{x}.\} \end{aligned}$$



(a) $\mathcal{V}(p)$ (b) $\mathcal{V}(q)$ (c) X

Figure 1: The tropical hypersurfaces $\mathcal{V}(p)$ and $\mathcal{V}(q)$ and the nondifferentiability locus X of $f = p - q$ in Example 2.1. The tropical hypersurfaces $\mathcal{V}(p)$ and $\mathcal{V}(q)$ divide $\mathbb{R}^2$ into polyhedral regions while X divides $\mathbb{R}^2$ into regions which can be described as a finite union of polyhedra.

Tropical hypersurfaces can be given the structure of a polyhedral complex and the connected components of the set $\mathbb{R}^n \setminus \mathcal{V}(p)$ are open polyhedra. A well-known result about tropical hypersurfaces is that they are determined by the polyhedral geometry of their coefficients; see, e.g., [23, Proposition 3.1.6].

If $f = p \oslash q$ is a tropical rational function, then the nondifferentiability locus of f is contained in $\mathcal{V}(p) \cup \mathcal{V}(q)$ and this containment can be proper.

Example 2.1. Consider the tropical polynomials $p = 0 \oplus x_1 \oplus x_2$ and $q = x_1 \oplus x_2$ and the tropical rational function $f = p - q = 0 \oplus x_1 \oplus x_2 - x_1 \oplus x_2$. Now,

$$\mathcal{V}(p) = \{(x_1, x_2) \in \mathbb{R}^2 | x_1 = x_2 \geq 0 \text{ or } x_1 = 0 \geq x_2 \text{ or } x_2 = 0 \geq x_1\}, \quad \mathcal{V}(q) = \{(x_1, x_2) \in \mathbb{R}^2 | x_1 = x_2\},$$

and the nondifferentiability locus of f is

$$X = \{(x_1, x_2) \in \mathbb{R}^2 | x_1 = x_2 \leq 0\} \cup \{(x_1, x_2) \in \mathbb{R}^2 | x_1 = 0, x_2 \leq 0\} \cup \{(x_1, x_2) \in \mathbb{R}^2 | x_2 = 0, x_1 \leq 0\}.$$

These sets are shown in Figure 1.

2.2 Weighted Lattices and Tropical Polynomial Regression

The vector addition and scalar multiplication on $\mathbb{T}^n$ are compatible with the partial order on $\mathbb{T}^n$ given by componentwise comparison. In [27], this compatibility is studied from an optimization perspective in the framework of *weighted lattices*, algebraic structures where operations are compatible with partial orders. Similar structures are discussed in detail in [8]. In [17, 26, 27, 40, 41], this framework is leveraged to find optimal subsolutions to tropical-linear systems of equations and in particular solve tropical polynomial regression problems. We summarize the key points in this section, closely following the presentation in [27].

The vector addition on $\mathbb{T}^n$ is idempotent and gives a partial order where $\mathbf{u} \leq \mathbf{v}$ if and only if $\max(\mathbf{u}, \mathbf{v}) = \mathbf{v}$. Similarly, the componentwise minimum gives the dual lattice structure. If $h : \mathbb{T}^n \to \mathbb{T}^m$ and $g : \mathbb{T}^m \to \mathbb{T}^n$ are two functions, then h is a *dilation* if $h(\max(\mathbf{u}, \mathbf{v})) = \max(h(\mathbf{u}), h(\mathbf{v}))$, the function g is an *erosion* if $g(\min(\mathbf{u}, \mathbf{v})) = \min(g(\mathbf{u}), g(\mathbf{v}))$, and the pair (h, g) is an *adjunction* if $h(\mathbf{u}) \leq \mathbf{v}$ is equivalent to $\mathbf{u} \leq g(\mathbf{v})$. As an example, for an $m \times n$ matrix $\mathbf{A}$, the map $\mathbb{T}^n \to \mathbb{T}^m$ given by $\mathbf{A} \boxplus \mathbf{u}$ is a dilation and the map $\mathbb{T}^m \to \mathbb{T}^n$ given by $(-\mathbf{A}^\top) \boxplus' \mathbf{u}$ is an erosion.

Theorem 2.1 ([27])**.** *Given a dilation $h : \mathbb{T}^n \to \mathbb{T}^m$, there is a unique erosion $g : \mathbb{T}^m \to \mathbb{T}^n$ given by*

$$g(\mathbf{v}) = \max\{\mathbf{u} \in \mathbb{T}^n | h(\mathbf{u}) \leq \mathbf{v}\}$$

such that (h, g) is an adjunction.

Applying Theorem 2.1 to max-plus matrix multiplication map $\mathbf{u} \mapsto \mathbf{A} \boxplus \mathbf{u}$ gives that the unique erosion to make an adjunct pair is the map $\mathbf{v} \mapsto (-\mathbf{A}^\top) \boxplus' \mathbf{v}$. This follows because $\max_{1 \leq j \leq n}(u_j + a_{ij}) \leq v_i$ for all $1 \leq i \leq m$ if and only if $u_j + a_{ij} \leq v_i$ for all i, j, which happens if and only if $u_j \leq \min_{1 \leq i \leq m}(v_i - a_{ij})$ for all j. So, if $\mathbf{A} \boxplus \mathbf{u} \leq \mathbf{v}$, then $\mathbf{u} \leq (-\mathbf{A}^\top) \boxplus' \mathbf{v}$.

Theorem 2.2 ([8])**.** *Let $\mathbf{A} \in \mathbb{T}^{m \times n}$ and $\mathbf{b} \in \mathbb{T}^m$.*

- *For any $\ell = 1, 2, 3, \ldots$, the optimal solution to*

$$\arg\min_{\mathbf{u}} \|\mathbf{A} \boxplus \mathbf{u} - \mathbf{b}\|_\ell \quad \text{s.t.} \quad \mathbf{A} \boxplus \mathbf{u} \leq \mathbf{b} \tag{2}$$

  is $\hat{\mathbf{u}} = (-\mathbf{A}^\top) \boxplus' \mathbf{b}$.

- *The optimal solution to*

$$\arg\min_{\mathbf{u}} \|\mathbf{A} \boxplus \mathbf{u} - \mathbf{b}\|_\infty \tag{3}$$

  is $\hat{\mathbf{u}} + \frac{1}{2}\|\mathbf{A} \boxplus \hat{\mathbf{u}} - \mathbf{b}\|_\infty$, where $\hat{\mathbf{u}}$ is defined as in the previous part.

We sketch a proof of the first part of Theorem 2.2 to demonstrate how the algebraic structure on $\mathbb{T}^n$ interacts with optimization. In particular, we use the algebraic structure to demonstrate why the solution to (2) is independent of ℓ.

Proof. Note that because max-plus matrix-vector multiplication is a dilation, we have that if $\mathbf{u}, \mathbf{v} \in \mathbb{T}^n$ are such that $\mathbf{u} \leq \mathbf{v}$ then $\mathbf{A} \boxplus \mathbf{u} \leq \mathbf{A} \boxplus \mathbf{v}$. Now, if $\mathbf{u}$ is feasible to (2), then $\mathbf{u} \leq (-\mathbf{A}^\top) \boxplus' \mathbf{b} = \hat{\mathbf{u}}$. This implies that $\mathbf{A} \boxplus \mathbf{u} \leq \mathbf{A} \boxplus \hat{\mathbf{u}} \leq \mathbf{b}$ and therefore for each component $i = 1, 2, \ldots, m$, it must be the case that $(\mathbf{b} - \mathbf{A} \boxplus \mathbf{u})_i \geq (\mathbf{b} - \mathbf{A} \boxplus \hat{\mathbf{u}})_i$. So, for any finite ℓ and any feasible $\mathbf{u}$, $\|\mathbf{A} \boxplus \mathbf{u} - \mathbf{b}\|_\ell \geq \|\mathbf{A} \boxplus \hat{\mathbf{u}} - \mathbf{b}\|_\ell$. □

Theorem 2.2 allows us to solve the tropical polynomial regression problem. Given data points $(\mathbf{x}^{(1)}, y^{(1)})$, $(\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)}) \in \mathbb{R}^n \times \mathbb{R}$ and a finite subset $W = \{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(D)}\} \subseteq \mathbb{Z}^n_{\geq 0}$, set $\mathbf{X}$ to be the $n \times D$ matrix with (i, j) entry $X_{i,j} = \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle$. This is the tropical analog of a Vandermonde matrix. Then, if $\mathbf{y}$ is the vector $\left[y^{(1)}, y^{(2)}, \ldots, y^{(N)}\right]^\top$, the tropical polynomial p which minimizes $\max_{1 \leq i \leq N} |p(\mathbf{x}^{(i)}) - y^{(i)}|$ is

$$p(\mathbf{x}) = \max_{j=1,2,\ldots,D} (p_j + \langle \mathbf{w}^{(j)}, \mathbf{x} \rangle), \tag{4}$$

where the vector of coefficients is

$$\mathbf{p} = (-\mathbf{X}^\top) \boxplus' \mathbf{y} + \frac{1}{2} \left\| \mathbf{X} \boxplus ((-\mathbf{X}^\top) \boxplus' \mathbf{y}) - \mathbf{y} \right\|_\infty . \tag{5}$$

Variants of Tropical Polynomial Regression In addition to the closed form solution of the tropical regression problem in the ∞-norm described above, other variants of tropical polynomial regression have been explored recently. In [17], the authors present two algorithms to solve the 2-norm max plus regression problem. The first involves a brute force search over sparsity patterns of solution vectors and the second involves a variant of Newton's method. In [24], the authors present a method for convex piecewise linear regression with the 2-norm loss which involves iteratively partitioning the input data and fitting affine functions to each partition. Finally, in [40, 41], the authors leverage the weighted lattice framework above to find sparse solutions to the problem (3). Specifically, the authors present a greedy algorithm to find a sparse solution to (2) for $p < \infty$ then shift the finite entries by half of the infinity norm of the residual.

2.3 Relations to ReLU Neural Networks

This section fixes notation for and briefly overviews the relationship between neural networks and tropical algebraic objects. We closely follow the presentation in [43]. A recent survey on tropical algebraic techniques for machine learning is [25].

An *L-layer neural network* with *ReLU* activation functions is a function $\nu : \mathbb{R}^n \to \mathbb{R}$ that can be expressed as a composition of functions

$$\nu = \rho^{(L)} \circ \sigma^{(L-1)} \circ \rho^{(L-1)} \circ \sigma^{(L-2)} \circ \cdots \circ \sigma^{(1)} \circ \rho^{(1)},$$

where $\sigma^{(\ell)}(\mathbf{x}) = \max(\mathbf{x}, 0)$ is the ReLU activation function and $\rho^{(\ell)}(\mathbf{x}) = \mathbf{A}^{(\ell)}\mathbf{x} + \mathbf{b}^{(\ell)}$ is an affine map $\mathbb{R}^{n_{\ell-1}} \to \mathbb{R}^{n_\ell}$. The sequence of dimensions $(n = n_0, n_1, n_2, \ldots, n_{L-1}, 1)$ is called the *architecture* of the network

representing ν. The matrix $\mathbf{A}^{(\ell)}$ and the vector $\mathbf{b}^{(\ell)}$ encode the *weights* and *bias* of layer ℓ, respectively. ReLU neural networks are continuous and piecewise linear by construction. Under assumptions on the entries of the $\mathbf{A}^{(\ell)}$, they can additionally be written as tropical rational functions.

Theorem 2.3 ([43]). *The following classes of functions are the same*

(i) Tropical rational functions

(ii) Continuous piecewise linear functions with integer coefficients

(iii) ReLU neural networks with integer weights

The authors of [43] note that if ν is a neural network with nonintegral weights, then rounding weights to rational numbers and clearing denominators gives a network with integer weights. More precisely,

Corollary 2.3.1. *If $\nu : \mathbb{R}^n \to \mathbb{R}$ is a ReLU neural network, then there is a real number c and a tropical rational function $f \in \mathbb{T}(x_1, \ldots, x_n)$ such that $\nu(\mathbf{x})$ is approximated arbitrarily closely by $f(c\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$.*

Removing the integrality condition on the weights of ν gives the following result:

Theorem 2.4 ([3]). *If $\nu : \mathbb{R}^n \to \mathbb{R}$ is a ReLU neural network, then ν is a piecewise linear function. Conversely, if $r : \mathbb{R}^n \to \mathbb{R}$ is piecewise a piecewise linear function, then r can be represented as a ReLU neural network with at most $\lceil \log_2(n+1) \rceil + 1$ layers.*

The above results suggest that, up to the scaling of the inputs, tropical rational functions and ReLU networks describe the same class of functions. So, tropical algebra and geometry can provide an alternative theoretical framework for understanding neural networks.

In machine learning applications, one typically fixes an architecture and optimizes over the parameters $\mathbf{A}^{(\ell)}$ and $\mathbf{b}^{(\ell)}$ using a variant of stochastic gradient descent. This approach presents challenges in the initialization of network parameters and in the choice of network architecture for a given application.

Popular network initialization methods such as those presented in [11, 16] draw initial weights independently from distributions that depend on network architecture. These methods help with stability while training.

An additional challenge in understanding ReLU neural networks lies in the nonuniqueness of the architecture used to represent the function ν. The proof of [43, Theorem 5.4] describes one method to write a tropical rational function $f(\mathbf{x}) = p(\mathbf{x}) - q(\mathbf{x})$ as a ReLU neural network. If g and h are two tropical polynomials represented by neural networks ν and μ, respectively, then

$$(g \oplus h)(\mathbf{x}) = \sigma((\nu - \mu)(\mathbf{x})) + \sigma(\mu(\mathbf{x})) - \sigma(-\mu(\mathbf{x}))) = \begin{bmatrix} 1 & 1 & -1 \end{bmatrix} \sigma\left(\begin{bmatrix} \nu(\mathbf{x}) - \mu(\mathbf{x}) \\ \mu(\mathbf{x}) \\ -\mu(\mathbf{x}) \end{bmatrix} \right). \tag{6}$$

In particular, the expression (6) can be applied to the case in which $g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + g_{\mathbf{w}}$ is a tropical monomial. This allows us to take the maximum of two networks by adding a layer and appropriately concatenating weight matrices in the hidden layers. In the resulting architecture, each hidden layer decreases in width. For example, a univariate degree 15 tropical rational function f can be represented via repeated applications of (6) as a neural network where the compositions are

$$\mathbb{R}^1 \to \mathbb{R}^{48} \to \mathbb{R}^{24} \to \mathbb{R}^{12} \to \mathbb{R}^6 \to \mathbb{R}^1.$$

Here, the input space is $\mathbb{R}^1$ because f is univariate. More generally, in a representation of an n-variate tropical rational function, the first term in the composition is $\mathbb{R}^n$. While this approach gives one network architecture representing the function ν, it is not unique. In [12, 13, 14, 28, 32], the authors investigate the relationship between network architectures and the polyhedral geometry of the function represented by a network ν.

In section 4.5, we present a preliminary investigation into these challenges from the tropical perspective by applying tropical rational regression as a heuristic for neural network initialization.

Algorithm 1: Alternating fit for tropical rational functions

Input: Dataset $\mathcal{D} = (\mathbf{x}^{(i)}, y^{(i)})_{i=1}^N \subseteq \mathbb{R}^n \times \mathbb{R}$,
Set of permissible exponents $W = \{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(D)}\} \subseteq \mathbb{Z}^n_{\geq 0}$,
Maximum number of iterations $k_{\max}$
Output: Vectors $\mathbf{p}$ and $\mathbf{q}$ of coefficients of tropical polynomials $p, q \in \mathbb{T}[\mathbf{x}]_W$ such that $p(\mathbf{x}^{(i)}) - q(\mathbf{x}^{(i)}) \approx y^{(i)}$

Set $\mathbf{X} \in \mathbb{R}^{N \times D}$ to be the matrix with entries $\mathbf{X}_{i,j} = \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle$;
$\mathbf{p}^0, \mathbf{q}^0 \leftarrow -\infty, \mathbf{q}^0_{\mathbf{0}} \leftarrow -\text{mean}(\mathbf{y})$;
for $k \leq k_{max}$ **do**
  $\mathbf{p}^k \leftarrow \arg\min_{\mathbf{p}} \|\mathbf{X} \boxplus \mathbf{p} - \mathbf{X} \boxplus \mathbf{q}^{k-1} - \mathbf{y}\|_\infty$;
  $\mathbf{q}^k \leftarrow \arg\min_{\mathbf{q}} \|\mathbf{X} \boxplus \mathbf{p}^k - \mathbf{X} \boxplus \mathbf{q} - \mathbf{y}\|_\infty$;
end
$\mathbf{p} \leftarrow \mathbf{p}^{k_{\max}}$; $\mathbf{q} \leftarrow \mathbf{q}^{k_{\max}}$

3 Alternating Method For Tropical Rational Regression

We adapt the polynomial regression method described in Section 2.2 to fit tropical rational functions to a dataset. Recall that a tropical rational function is a function of the form $f(\mathbf{x}) := p(\mathbf{x}) - q(\mathbf{x})$, where p and q are tropical polynomials. So, for some finite subset $W = \{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(D)}\} \subseteq \mathbb{Z}^n_{\geq 0}$,

$$f(\mathbf{x}) = p(\mathbf{x}) - q(\mathbf{x}) = \max_{j=1,2,\ldots,D}(\langle \mathbf{w}^{(j)}, \mathbf{x} \rangle + p_j) - \max_{j=1,2,\ldots,D}(\langle \mathbf{w}^{(j)}, \mathbf{x} \rangle + q_j).$$

Given a set of points $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)} \in \mathbb{R}^n$, set $\mathbf{X}$ to be the matrix whose rows are indexed by $i \in \{1, \ldots, N\}$ and columns are indexed by $\mathbf{w} \in W$ with $\mathbf{X}_{(i,j)} = \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle$. Evaluation of the tropical rational function f at the points $\mathbf{x}^{(i)}$ is then given by

$$\begin{bmatrix} f(\mathbf{x}^{(1)}) & f(\mathbf{x}^{(2)}) & \cdots & f(\mathbf{x}^{(N)}) \end{bmatrix}^\top = \mathbf{X} \boxplus \mathbf{p} - \mathbf{X} \boxplus \mathbf{q}, \tag{7}$$

where $\mathbf{p}$ and $\mathbf{q}$ are the vectors of coefficients of p and q. Using the representation (7), it follows that we can rewrite the problem (1) as

$$\arg \min_{f \in \mathbb{T}(\mathbf{x})_W} \left\| \begin{bmatrix} f(\mathbf{x}^{(1)}) & f(\mathbf{x}^{(2)}) & \cdots & f(\mathbf{x}^{(N)}) \end{bmatrix}^\top - \mathbf{y} \right\|_\infty = \arg \min_{\mathbf{p},\mathbf{q}} \|\mathbf{X} \boxplus \mathbf{p} - \mathbf{X} \boxplus \mathbf{q} - \mathbf{y}\|_\infty . \tag{8}$$

For fixed $\mathbf{q}$, the problem $\arg\min_{\mathbf{p}} \|\mathbf{X} \boxplus \mathbf{p} - (\mathbf{X} \boxplus \mathbf{q} + \mathbf{y})\|_\infty$ is a tropical polynomial regression problem. By Theorem 2.2, this problem has the analytical solution

$$\mathbf{p}_*(\mathbf{q}) = (-\mathbf{X}^\top) \boxplus' (\mathbf{X} \boxplus \mathbf{q} + \mathbf{y}) + \frac{1}{2} \left\| \mathbf{X} \boxplus \left((-\mathbf{X}^\top) \boxplus' (\mathbf{X} \boxplus \mathbf{q} + \mathbf{y}) \right) - (\mathbf{X} \boxplus \mathbf{q} + \mathbf{y}) \right\|_\infty .$$

Similarly, for fixed $\mathbf{p}$, the problem $\arg\min_{\mathbf{q}} \|\mathbf{X} \boxplus \mathbf{q} - (\mathbf{X} \boxplus \mathbf{p} - \mathbf{y})\|_\infty$ has the analytical solution

$$\mathbf{q}_*(\mathbf{p}) = (-\mathbf{X}^\top) \boxplus' (\mathbf{X} \boxplus \mathbf{p} - \mathbf{y}) + \frac{1}{2} \left\| \mathbf{X} \boxplus \left((-\mathbf{X}^\top) \boxplus' (\mathbf{X} \boxplus \mathbf{p} - \mathbf{y}) \right) - (\mathbf{X} \boxplus \mathbf{p} - \mathbf{y}) \right\|_\infty .$$

Moreover, these analytical solutions can be found quickly, as they rely only on max-plus and min-plus matrix-vector products and do not need to solve a linear system. We exploit this to search over the space of tropical rational functions by alternating between fitting the numerator polynomial and the denominator polynomial. This method is summarized below as Algorithm 1.

While Algorithm 1 is defined for general choices of $W \subseteq \mathbb{Z}^n_{\geq 0}$, our implementation takes W to be of the form $W = \{(w_1, w_2, \ldots, w_n) \in \mathbb{Z}^n_{\geq 0} \mid w_i \leq d_i\}$ for some $\mathbf{d} = (d_1, d_2, \ldots, d_n)^\top \in \mathbb{Z}^n_{\geq 0}$. In this case, $|W|$ becomes very large if $\mathbf{d}$ has large entries or if n is large. In [27], the authors discuss choosing W by clustering approximated gradients from the data to reduce the number of parameters used in fitting tropical polynomials. The choice of initialization is such that f is initialized to the constant function $f(x_1, \ldots, x_n) = \text{mean}(\mathbf{y})$.

As a step towards understanding convergence properties of Algorithm 1, we show that the error at each iteration is nonincreasing.

Proposition 3.1. *The error* $e^k = \|\mathbf{X} \boxplus \mathbf{p}^k - \mathbf{X} \boxplus \mathbf{q}^k - \mathbf{y}\|_\infty$ *is nonincreasing.*

Proof. We show that $e^{k+1} \leq e^k$ for any k. By construction, $\mathbf{p}^{k+1}$ satisfies

$$\|\mathbf{X} \boxplus \mathbf{p}^{k+1} - \mathbf{X} \boxplus \mathbf{q}^k - \mathbf{y}\|_\infty \leq \|\mathbf{X} \boxplus \mathbf{p}^k - \mathbf{X} \boxplus \mathbf{q}^k - \mathbf{y}\|_\infty = e^k.$$

Similarly,

$$e^{k+1} = \|\mathbf{X} \boxplus \mathbf{p}^{k+1} - \mathbf{X} \boxplus \mathbf{q}^{k+1} - \mathbf{y}\|_\infty \leq \|\mathbf{X} \boxplus \mathbf{p}^{k+1} - \mathbf{X} \boxplus \mathbf{q}^k - \mathbf{y}\|_\infty.$$

It then follows that $e^{k+1} \leq e^k$. □

The decrease in error between iterations is bounded by a constant multiple of the norm of the update step.

Proposition 3.2. *Let* $\eta^k = \left\| \begin{bmatrix} \mathbf{p}^{k+1} & \mathbf{q}^{k+1} \end{bmatrix}^\top - \begin{bmatrix} \mathbf{p}^k & \mathbf{q}^k \end{bmatrix}^\top \right\|_\infty$. *Then, the change in error between iterations* $e^k - e^{k+1}$ *is bounded:*

$$e^k - e^{k+1} \leq 2\eta^k.$$

Proof. First, note that for each $j = 1, 2, \ldots, D$,

$$p_j^k - \eta^k \leq p_j^{k+1} \leq p_j^k + \eta^k \quad \text{and} \quad q_j^k - \eta^k \leq q_j^{k+1} \leq q_j^k + \eta^k.$$

Because for fixed $i \in \{1, 2, \ldots, N\}$,

$$\max_{j=1,2,\ldots,D} (p_j^k + \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle) - \eta^k = \max_{j=1,2,\ldots,D} (p_j^k - \eta^k + \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle),$$

this in turn implies that

$$\max_{j=1,2,\ldots,D} (p_j^k + \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle) - \eta^k \leq \max_{j=1,2,\ldots,D} (p_j^{k+1} + \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle) \leq \max_{j=1,2,\ldots,D} (p_j^k + \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle) + \eta^k. \quad (9)$$

The analogous statement holds with $\mathbf{q}^k$ replacing $\mathbf{p}^k$.

Let $\ell \in \{1, 2, \ldots, N\}$ be such that $e^k = |p^k(\mathbf{x}^{(\ell)}) - q^k(\mathbf{x}^{(\ell)}) - y^{(\ell)}|$. Now,

$$\begin{aligned}
e^k - e^{k+1} &= |p^k(\mathbf{x}^{(\ell)}) - q^k(\mathbf{x}^{(\ell)}) - y^{(\ell)}| - \max_{i=1,2,\ldots,N} |p^{k+1}(\mathbf{x}^{(i)}) - q^{k+1}(\mathbf{x}^{(i)}) - y^{(i)}| \\
&\leq |p^k(\mathbf{x}^{(\ell)}) - q^k(\mathbf{x}^{(\ell)}) - y^{(\ell)}| - |p^{k+1}(\mathbf{x}^{(\ell)}) - q^{k+1}(\mathbf{x}^{(\ell)}) - y^{(\ell)}| \\
&\leq |p^k(\mathbf{x}^{(\ell)}) - q^k(\mathbf{x}^{(\ell)}) - p^{k+1}(\mathbf{x}^{(\ell)}) + q^{k+1}(\mathbf{x}^{(\ell)})| \\
&\leq |p^k(\mathbf{x}^{(\ell)}) - p^{k+1}(\mathbf{x}^{(\ell)})| + |q^k(\mathbf{x}^{(\ell)}) - q^{k+1}(\mathbf{x}^{(\ell)})| \\
&\leq 2\eta^k
\end{aligned}$$

□

We observe in our experiments that η^k is nonincreasing, motivating its use as a stopping criterion for Algorithm 1.

3.1 Nondifferentiability of the Loss Function

In this section, we investigate the geometry of the problem (1) by viewing the loss function as a tropical rational function. In particular, we show that (1) always has a minimizer for which the loss function is nondifferentiable. Moreover, the iterates produced by Algorithm 1 are always elements of the nondifferentiability locus of the loss function. Finally, we discuss preliminary consequences of nondifferentiability at a minimizer and present open problems involving the geometry of the loss function.

Proposition 3.3. *The loss function*

$$\mathcal{L}(\mathbf{p},\mathbf{q}) = \|\mathbf{X} \boxplus \mathbf{p} - \mathbf{X} \boxplus \mathbf{q} - \mathbf{y}\|_\infty$$

is a tropical rational function of the coefficients p_j, q_j, $j = 1, 2, \ldots, D$.

Proof. Note that

$$\begin{aligned}\mathcal{L}(\mathbf{p},\mathbf{q}) &= \|\mathbf{X} \boxplus \mathbf{p} - \mathbf{X} \boxplus \mathbf{q} - \mathbf{y}\|_\infty \\ &= \max_{i=1,2,\ldots,N} \left[\max(p(\mathbf{x}^{(i)}) - q(\mathbf{x}^{(i)}) - y^{(i)}, y^{(i)} + q(\mathbf{x}^{(i)}) - p(\mathbf{x}^{(i)}))\right].\end{aligned}$$

Now, for each $i = 1, 2, \ldots, N$, the evaluation map on $\mathbb{T}[\mathbf{x}]_W$ which sends $g \mapsto g(\mathbf{x}^{(i)})$ is tropically linear in the coefficients g_j. In particular, for each $i = 1, 2, \ldots, N$, both $p(\mathbf{x}^{(i)}) - q(\mathbf{x}^{(i)}) - y^{(i)}$ and $y^{(i)} + q(\mathbf{x}^{(i)}) - p(\mathbf{x}^{(i)})$ are tropical rational functions of the parameters p_j, q_j. Because the set of tropical rational functions is closed under tropical addition, this implies that $\mathcal{L}(\mathbf{p},\mathbf{q})$ is a tropical rational function. □

Proposition 3.3 allows us to use the polyhedral geometry of tropical hypersurfaces to study the geometry of the optimization problem (1).

Proposition 3.4. *There is an optimal solution to (1). Moreover, there is an optimal solution $(\mathbf{p}^*, \mathbf{q}^*)$ such that $\nabla\mathcal{L}(\mathbf{p}^*, \mathbf{q}^*)$ does not exist.*

Proof. By Proposition 3.3, there are tropical polynomials g, h in $2D$ indeterminates such that

$$\mathcal{L}(\mathbf{p},\mathbf{q}) = g(\mathbf{p},\mathbf{q}) - h(\mathbf{p},\mathbf{q}).$$

Note that the tropical polynomials g and h each depend on the coefficients of both p and q and are, therefore, polynomials in 2D indeterminates. The nondifferentiability locus of $\mathcal{L}$ is a subset of $\Sigma = \mathcal{V}(g) \cup \mathcal{V}(h) \subseteq \mathbb{R}^{2D}$. There are finitely many connected components of $\mathbb{R}^{2D} \setminus \Sigma$, each of which are open polyhedra. Label these polyhedra $A_1, A_2, \ldots, A_s$.

For the first claim, note that $\mathcal{L}$ is linear on $\mathrm{cl}(A_i)$ for each $i = 1, 2, \ldots, s$. Because $\mathcal{L}(\mathbf{p},\mathbf{q}) \geq 0$ for all $(\mathbf{p},\mathbf{q}) \in \mathbb{R}^{2|W|}$, the restriction of $\mathcal{L}$ to $\mathrm{cl}(A_i)$ achieves a minimum value z_i on $\mathrm{cl}(A_i)$. Then $\mathcal{L}$ achieves the minimum value $z = \min_{i=1,2,\ldots,s} z_i$.

For the second claim, note that the restriction of $\mathcal{L}$ to $\mathrm{cl}(A_i)$ achieves its minimum on the boundary ∂A_i for each $i = 1, 2 \ldots, s$. So, there must be an optimal solution in $\Sigma = \cup_{i=1}^s \partial A_i$. Let $(\hat{\mathbf{p}}, \hat{\mathbf{q}})$ be an optimal solution in Σ such that $\nabla\mathcal{L}(\hat{\mathbf{p}}, \hat{\mathbf{q}})$ exists. By the hypothesis that $(\hat{\mathbf{p}}, \hat{\mathbf{q}})$ is an optimal solution, it is necessary that $\nabla\mathcal{L}(\hat{\mathbf{p}}, \hat{\mathbf{q}}) = 0$. Relabeling the A_i if necessary, let $A_1, \ldots, A_k$ be such that $(\hat{\mathbf{p}}, \hat{\mathbf{q}}) \in \cap_{i=1}^k \mathrm{cl}(A_i)$. Because $\mathcal{L}$ is linear on each A_i, it must be the case that $\nabla\mathcal{L}|_{A_i} = 0$ for each i. This implies that every point in $A = \cup_{i=1}^k \mathrm{cl}(A_i)$ is a minimizer of $\mathcal{L}$. Set B to be the smallest connected subset containing A on which $\mathcal{L}$ is minimized. Note that $B = \cup_{i=1}^r \mathrm{cl}(A_i)$ where $r \geq k$. If $B \neq \mathbb{R}^{2D}$, then there is a point $(\mathbf{p}^*, \mathbf{q}^*)$ on the boundary of B where $\nabla\mathcal{L}$ does not exist. Otherwise, $B = \mathbb{R}^{2D}$ and therefore $\mathcal{L}$ is constant. However, $\mathcal{L}$ cannot be constant because for fixed $\mathbf{w}^{(j)} \in W$, fixed $\mathbf{q}$, and fixed p_r for $r \neq j$,

$$\mathcal{L}(\mathbf{p},\mathbf{q}) = \max_{i=1,2,\ldots,N} \left| \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle + p_j - \max_{j=1,2,\ldots,D} (\langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle + q_j) - y^{(i)} \right| = p_j - C$$

for some constant C for sufficiently large values of p_j. □

There are problems for which every optimal solution is in the nondifferentiability locus of $\mathcal{L}$.

Example 3.1. Consider the problem with $\mathcal{D} = \{(0,0)\}$, $W = \{0\} \subseteq \mathbb{Z}$. A tropical rational function with W as its support is

$$f(x) = p_0 - q_0.$$

The minimum of $\mathcal{L}$ is 0, achieved on the line span$\{(1,1)\}$. However, along this line, the loss is

$$\mathcal{L}(p_0, q_0) = \|p_0 - q_0\|_\infty = |p_0 - q_0|$$

so that $\nabla\mathcal{L}(p_0, q_0)$ does not exist when $p_0 = q_0$.

Algorithm 1 produces iterates in the nondifferentiablity locus of $\mathcal{L}$.

Theorem 3.5. *The gradient $\nabla\mathcal{L}(\mathbf{p}^k, \mathbf{q}^k)$ does not exist, where $\mathbf{p}^k$ and $\mathbf{q}^k$ are defined as in Algorithm 1 and $k \geq 1$.*

Proof. For fixed $k \geq 1$, the functions $\mathcal{L}(\mathbf{p}, \mathbf{q}^{k-1})$ and $\mathcal{L}(\mathbf{p}^k, \mathbf{q})$ are the infinity norm of the residual of a tropical polynomial regression problem. Because the updates $\mathbf{p}^k$ and $\mathbf{q}^k$ are minimizers of the infinity norm of such residuals, it suffices to show that in the setup of Theorem 2.2,

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \|\mathbf{A} \boxplus \mathbf{u} - \mathbf{b}\|_\infty$$

is a nondifferentiable point of the function $\mathcal{R}(\mathbf{u}) = \|\mathbf{A} \boxplus \mathbf{u} - \mathbf{b}\|_\infty$.

Suppose for the sake of a contradiction that $\nabla\mathcal{R}(\mathbf{u}^*)$ exists. Because $\mathbf{u}^*$ minimizes $\mathcal{R}$ by hypothesis, it follows that $\nabla\mathcal{R}(\mathbf{u}^*) = 0$. Fix indices i and j such that

$$\mathcal{R}(\mathbf{u}^*) = |\max_\ell(a_{i,\ell} + u^*_\ell) - b_i| = |a_{i,j} + u^*_j - b_i|.$$

Set $J = \{k \mid a_{i,k} + u^*_k = \max_\ell(a_{i,\ell} + u^*_\ell)\}$ and $\mathbf{e}_J$ to be the vector with 1 in component k if $k \in J$ and 0 otherwise. Note that the fixed index $j \in J$. Then, if $\epsilon > 0$ is small enough that $a_{i,k} + u^*_k - \epsilon > a_{i,\ell} + u^*_\ell$ when $k \in J$ and $\ell \notin J$, then there exists $c \in \{-1, 1\}$ such that

$$\mathcal{R}(\mathbf{u}^* + c\epsilon\mathbf{e}_J) \geq |a_{i,j} + u^*_j + c\epsilon - b_i| = |a_{i,j} + u^*_j - b_i| + \epsilon.$$

But then, the difference quotient

$$\left|\frac{\mathcal{R}(\mathbf{u}^* + c\epsilon\mathbf{e}_J) - \mathcal{R}(\mathbf{u}^*)}{\epsilon}\right| \geq 1$$

is bounded away from 0 for $\epsilon > 0$ sufficiently small, a contradiction with the hypothesis that $\nabla\mathcal{R}(\mathbf{u}^*) = 0$. □

Proposition 3.4 and Example 3.1 demonstrate the importance of understanding the nondifferentiability locus of $\mathcal{L}$. The two sources of nondifferentiability in $\mathcal{L}$ are the nondifferentiability of $\|\mathbf{u} - \mathbf{v}\|_\infty$ as a function of $\mathbf{u}$ and the nondifferentiability of the tropical rational functions $f(\mathbf{x}^{(i)}) = p(\mathbf{x}^{(i)}) - q(\mathbf{x}^{(i)})$ as a function of the coefficients p_j and q_j. This connects the geometry of the dataset to that of a tropical rational function produced as an iterate of Algorithm 1 by providing a certificate that $(\mathbf{p}, \mathbf{q})$ is in the nondifferentiability locus of $\mathcal{L}$ in terms of the input data.

Proposition 3.6. *There exists a minimizer $(\mathbf{p}^*, \mathbf{q}^*)$ of $\mathcal{L}$ such that at least one of the following holds:*

1. *There is an i such that $\mathbf{x}^{(i)} \in \mathcal{V}(p^*) \cup \mathcal{V}(q^*)$*
2. *The infinity norm in $\mathcal{L}(\mathbf{p}^*, \mathbf{q}^*)$ is achieved by at least two data points $(\mathbf{x}^{(i)}, y^{(i)}), (\mathbf{x}^{(j)}, y^{(j)})$.*

Proof. We show the contrapositive. Let $(\mathbf{p}^*, \mathbf{q}^*)$ be a minimizer of $\mathcal{L}$ such that $\nabla\mathcal{L}(\mathbf{p}^*, \mathbf{q}^*)$ does not exist and suppose that neither condition holds. Let $1 \leq i \leq N$ be such that

$$\mathcal{L}(\mathbf{p}^*, \mathbf{q}^*) = \left| \max_{j=1,2,\dots,D} (p_j^* + \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle) - \max_{j=1,2,\dots,D} (q_j^* + \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle) - y^{(i)} \right|$$
$$> \left| \max_{j=1,2,\dots,D} (p_j^* + \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle) - \max_{j=1,2,\dots,D} (q_j^* + \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle) - y^{(\ell)} \right|$$

for all $\ell \neq i$. Then there is an open neighborhood $U \subseteq \mathbb{R}^{2D}$ of $(\mathbf{p}^*, \mathbf{q}^*)$ such that

$$\mathcal{L}(\mathbf{p}, \mathbf{q}) = \left| \max_{j=1,2,\dots,D} (p_j + \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle) - \max_{j=1,2,\dots,D} (q_j + \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle) - y^{(i)} \right|$$

for all $(\mathbf{p}, \mathbf{q}) \in U$. Because $\mathcal{L}(\mathbf{p}^*, \mathbf{q}^*) > 0$, it suffices to show that if $\mathbf{x}^{(i)} \notin \mathcal{V}(p^*) \cup \mathcal{V}(q^*)$ then the evaluation map $(\mathbf{p}, \mathbf{q}) \mapsto p(\mathbf{x}^{(i)}) - q(\mathbf{x}^{(i)})$ is differentiable at $(\mathbf{p}^*, \mathbf{q}^*)$. By the hypothesis that $\mathbf{x}^{(i)} \notin \mathcal{V}(p^*) \cup \mathcal{V}(q^*)$, there are $\mathbf{w}^{(j)}, \mathbf{w}^{(k)}$ such that $p_j^* + \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle > p_\ell^* + \langle \mathbf{w}^{(\ell)}, \mathbf{x}^{(i)} \rangle$ for all $\ell \neq j$ and $q_k^* + \langle \mathbf{w}^{(k)}, \mathbf{x}^{(i)} \rangle > q_\ell^* + \langle \mathbf{w}^{(\ell)}, \mathbf{x}^{(i)} \rangle$ for all $\ell \neq k$. Restricting U to a smaller open neighborhood of $(\mathbf{p}^*, \mathbf{q}^*)$ if necessary then gives that

$$\mathcal{L}(\mathbf{p}, \mathbf{q}) = \left| p_j^* + \langle \mathbf{w}^{(j)}, \mathbf{x}^{(i)} \rangle - q_k^* - \langle \mathbf{w}^{(k)}, \mathbf{x}^{(i)} \rangle - y^{(i)} \right|$$

near $(\mathbf{p}^*, \mathbf{q}^*)$. This is an affine function of $(\mathbf{p}, \mathbf{q})$ on U because $\mathcal{L}(\mathbf{p}, \mathbf{q}) > 0$ and therefore $\nabla \mathcal{L}(\mathbf{p}^*, \mathbf{q}^*)$ exists. □

The above discussion highlights the importance of understanding the nondifferentiability locus of tropical rational functions for developing convergence results for Algorithm 1. Such objects are studied in [37], but have not received much attention in the literature.

Based on the experimental results in Section 4, provable convergence of the iterates $(\mathbf{p}^k, \mathbf{q}^k)$ to some (neighborhood of a) stationary point $(\mathbf{p}^*, \mathbf{q}^*)$ appears likely. In light of the piecewise linear geometry of the loss function $\mathcal{L}$, the natural notion of a neighborhood of a stationary point is a polyhedron in $\mathbb{R}^{2D}$ on which $\mathcal{L}$ is constant. So, it is desirable to relate the iterates produced by Algorithm 1 to the geometry of $\mathcal{L}$. We present two open geometric questions that could be useful in the development of a convergence proof.

First, how do the iterates produced by Algorithm 1 traverse the nondifferentiability locus of $\mathcal{L}$? Specifically, if $(\mathbf{p}^k, \mathbf{q}^k)$ and $(\mathbf{p}^{k+1}, \mathbf{q}^{k+1})$ are consecutive iterates of Algorithm 1, is there a polyhedron $\mathcal{P}$ on which $\mathcal{L}$ is linear and such that $(\mathbf{p}^k, \mathbf{q}^k), (\mathbf{p}^{k+1}, \mathbf{q}^{k+1}) \in \mathcal{P}$? Second, if $(\mathbf{p}^*, \mathbf{q}^*)$ is a minimizer of $\mathcal{L}$, then because $\mathcal{L}$ is piecewise linear, it must be convex on a neighborhood of $(\mathbf{p}^*, \mathbf{q}^*)$. How do the iterates produced by Algorithm 1 compare to those produced by a subgradient method? Understanding these geometric problems would allow one to apply the extensive existing theory of linear programming and subgradient methods, respectively, to develop optimality conditions for the Problem 1.

3.2 Polynomial Evaluation

In order to effectively use Algorithm 1, we need to be able to efficiently perform the matrix-vector multiplications involved in solving the minimization problem. This amounts to evaluating a tropical polynomial and performing a min-plus matrix-vector product using the negative transpose of a "Vandermonde" type matrix. This structure can be leveraged to perform computations without storage of the matrix $\mathbf{X}$.

Univariate Polynomial Evaluation Let $n = 1$ and consider the case of evaluating the degree d univariate tropical polynomial $p(t) = \max_{0 \leq j \leq d}(jt + p_j)$ at the points $x^{(1)}, x^{(2)}, \dots, x^{(N)} \in \mathbb{R}$. In this case the matrix $\mathbf{X}$ is given by $\mathbf{X} = \begin{bmatrix} \mathbf{0} & \mathbf{x} & \cdots & d\mathbf{x} \end{bmatrix}$, where $\mathbf{x}$ is the vector of the $x^{(i)}$. Then, $\mathbf{v} = \mathbf{X} \boxplus \mathbf{p}$ a vector of evaluations of p at the $x^{(i)}$. This computation does not require the explicit formation of the highly structured matrix $\mathbf{X}$. The solution $\mathbf{v}$ can be computed by setting $\mathbf{v}^0 = p_0 \mathbf{1}$ and computing

$$\mathbf{v}^k = \max(\mathbf{v}^{k-1}, k\mathbf{x} + p_k \mathbf{1}), \quad 1 \leq k \leq d,$$

so that $\mathbf{v}^d = \mathbf{v}$. This approach avoids the storage of the $N \times (d+1)$ matrix $\mathbf{X}$ and instead only uses the length N vector $\mathbf{x}$.

Similarly, an explicit storage of the matrix $\mathbf{X}$ can be avoided when computing $\hat{\mathbf{p}} = (-\mathbf{X}^\top) \boxplus' \mathbf{y}$. This follows because $\hat{p_j} = \min_{1 \leq i \leq N}(y_i - jx_i)$, so that there is no need to store the matrix $\mathbf{X}$ (compare with [26, Equation 17]).

Finally, to compute $\mathbf{v} = \mathbf{X} \boxplus \left((-\mathbf{X})^\top \boxplus' \mathbf{y}\right)$, we set $\mathbf{v}^0 = \min_{1 \leq i \leq N}(y_i)\mathbf{1}$ and compute

$$\mathbf{v}^k = \max\left(\mathbf{v}^{k-1}, k\mathbf{x} + \min_{1 \leq i \leq N}(y_i - kx_i)\mathbf{1}\right), \quad 1 \leq k \leq d.$$

Then, $\mathbf{v}^d = \mathbf{v} = \mathbf{X} \boxplus \left((-\mathbf{X})^\top \boxplus' \mathbf{y}\right)$. In this way, we are able to evaluate the product $\mathbf{X} \boxplus \left((-\mathbf{X})^\top \boxplus' \mathbf{y}\right)$, which gives the evaluation of the polynomial subfit problem as the coefficients of the polynomial are found. In particular, we can find the coefficients and evaluate the polynomial with a total cost of $\mathcal{O}(dN)$ operations.

The methods in the univariate case form the basis for effective computations with multivariate tropical polynomials as the number of columns in the matrix $\mathbf{X}$ grows as d^n.

Multivariate Polynomial Evaluation We extend the univariate polynomial evaluation method to the multivariate case by considering a polynomial $p \in \mathbb{T}[x_1, \ldots, x_n]$ as a polynomial in the variable x_n with coefficients in $\mathbb{T}[x_1, \ldots, x_{n-1}]$ and evaluating the coefficients.

For example, in the bivariate case, the polynomial

$$p(x_1, x_2) = \max_{0 \leq i \leq d_1, 0 \leq j \leq d_2}(ix_1 + jx_2 + p_{i,j})$$

is to be evaluated at a given set of evaluation points $\left(x_1^{(1)}, x_2^{(1)}\right), \left(x_1^{(2)}, x_2^{(2)}\right), \ldots, \left(x_1^{(N)}, x_2^{(N)}\right) \in \mathbb{R}^2$. Rewrite the polynomial p, collecting all terms of the same degree in x_2. Using tropical notation, this gives

$$p(x_1, x_2) = \bigoplus_{j=0}^{d_2} x_2^{\odot j} \odot \left(\bigoplus_{i=0}^{d_1} x_1^{\odot i} \odot p_{i,j}\right) = \max_{j=0,\ldots,d_2}\left(jx_2 + \max_{i=0,\ldots,d_1}(ix_1 + p_{i,j})\right).$$

Now, the term $\max_{i=0,\ldots,d_1}(ix_1 + p_{i,j})$ is a univariate tropical polynomial for each j and can therefore be evaluated without the storage of the matrix $\mathbf{X}$. Once these terms are each evaluated, p is a univariate polynomial in x_2. Ultimately, this avoids the storage of the large $N \times (d_1+1)(d_2+1)$ matrix $\mathbf{X}$ and instead only uses the N pairs $\left(x_1^{(i)}, x_2^{(i)}\right)$ and the degree bounds d_1, d_2.

We also compute the solution to the polynomial subfit problem without explicitly storing the matrix $\mathbf{X}$. Similarly to the univariate case, each entry in the output of $\hat{\mathbf{p}} = (-\mathbf{X}^\top) \boxplus' \mathbf{y}$ has the form

$$\hat{p}_{\mathbf{w}} = \min_{1 \leq i \leq N}\left(y_i - \sum_{j=1}^{n} w_j x_j^{(i)}\right).$$

So, it is not necessary to store more than the evaluation points $\left(x_1^{(i)}, x_2^{(i)}\right)$.

Finally, to evaluate the product $\mathbf{v} = \mathbf{X} \boxplus \left((-\mathbf{X})^\top \boxplus' \mathbf{y}\right)$, we initialize $\mathbf{v}^1 = \min_{1 \leq i \leq N}(y_i)\mathbf{1}$. For an enumeration $W \setminus \{0\} = \left\{\mathbf{w}^{(2)}, \mathbf{w}^{(3)}, \ldots, \mathbf{w}^{(D)}\right\}$ set

$$\mathbf{u}^k = \begin{bmatrix} \sum_{j=1}^n w_j^{(k)} x_j^{(1)} & \sum_{j=1}^n w_j^{(k)} x_j^{(2)} & \cdots & \sum_{j=1}^n w_j^{(k)} x_j^{(N)} \end{bmatrix}^\top, \quad k = 2, 3, \ldots, D$$

and update

$$\mathbf{v}^k = \max\left(\mathbf{v}^{k-1}, \mathbf{u}^k + \min_{1 \leq i \leq N}(y_i - u_i^k)\mathbf{1}\right), \quad k = 2, 3, \ldots, D.$$

Then $\mathbf{v}^D = \mathbf{v} = \mathbf{X} \boxplus \left((-\mathbf{X})^\top \boxplus' \mathbf{y}\right)$. Note that if $\mathbf{w}^{(k)} - \mathbf{w}^{(k-1)}$ is the standard basis vector $\mathbf{e}_j$, then $\mathbf{u}^k$ can be constructed from $\mathbf{u}^{k-1}$ as $\mathbf{u}^k = \mathbf{u}^{k-1} + \begin{bmatrix} x_j^{(1)} & x_j^{(2)} & \cdots & x_j^{(N)} \end{bmatrix}^\top$ and therefore the updates to $\mathbf{u}^k$ and $\mathbf{v}^k$ can both be computed efficiently from the input data. In particular, in this case, the update to $\mathbf{u}^k$ is computed using N additions instead of the $\mathcal{O}(nN)$ operations needed to construct $\mathbf{u}^k$ from scratch.

3.3 Relationship with Two-Sided Tropical Linear Systems

An alternate perspective on the optimization problem (8) is as minimizing the residual in the solution of a two-sided system of tropically linear equations. Indeed, if Y is the $N \times N$ matrix with entries

$$\mathbf{Y}_{i,j} = \begin{cases} y_\ell, & i = j = \ell \\ -\infty, & i \neq j \end{cases},$$

then $\mathbf{X} \boxplus \mathbf{q} + \mathbf{y} = (\mathbf{Y} \boxplus \mathbf{X}) \boxplus \mathbf{q}$ and the objective $\|\mathbf{X} \boxplus \mathbf{p} - \mathbf{X} \boxplus \mathbf{q} - \mathbf{y}\|$ is the same as the norm of the residual of the two-sided tropical linear system $\mathbf{X} \boxplus \mathbf{p} = (\mathbf{Y} \boxplus \mathbf{X}) \boxplus \mathbf{q}$. Two-sided systems of equations in the form $\mathbf{A} \boxplus \mathbf{p} = \mathbf{B} \boxplus \mathbf{q}$ are considered in [7], where an alternating algorithm is presented which converges to a solution $(\mathbf{p}, \mathbf{q})$ in finitely many steps when a solution exists. The algorithm in [7] proceeds by initializing a vector $\mathbf{q}^0$ and alternately setting $\mathbf{p}^k = (-\mathbf{A}^\top) \boxplus' (\mathbf{B} \boxplus \mathbf{q}^k)$ and $\mathbf{q}^{k+1} = (-\mathbf{B}^\top) \boxplus' (\mathbf{A} \boxplus \mathbf{p}^k)$. Note that this is equivalent to sequentially solving the constrained problems

$$\mathbf{p}^k = \arg\min_{\mathbf{p}} \|\mathbf{A} \boxplus \mathbf{p} - \mathbf{B} \boxplus \mathbf{q}^k\|_\ell \quad \text{s.t.} \quad \mathbf{A} \boxplus \mathbf{p} \leq \mathbf{B} \boxplus \mathbf{q}^k$$

and

$$\mathbf{q}^{k+1} = \arg\min_{\mathbf{q}} \|\mathbf{A} \boxplus \mathbf{p}^k - \mathbf{B} \boxplus \mathbf{q}\|_\ell \quad \text{s.t.} \quad \mathbf{B} \boxplus \mathbf{q} \leq \mathbf{A} \boxplus \mathbf{p}^k$$

for any finite $\ell = 1, 2, 3, \ldots$. If this is applied with $\mathbf{A} = \mathbf{X}$ and $\mathbf{B} = (\mathbf{Y} \boxplus \mathbf{X})$ then at each k, it must be the case that $\mathbf{X} \boxplus \mathbf{p}^k - \mathbf{X} \boxplus \mathbf{q}^k \leq \mathbf{y}$. From a regression point of view, where we expect a nonzero residual, it is more desirable to solve unconstrained problems at each iteration to avoid such a constraint. Replacing the constrained updates in the alternating method of [7] with the unconstrained ∞-norm minimization results in Algorithm 1.

4 Computational Experiments

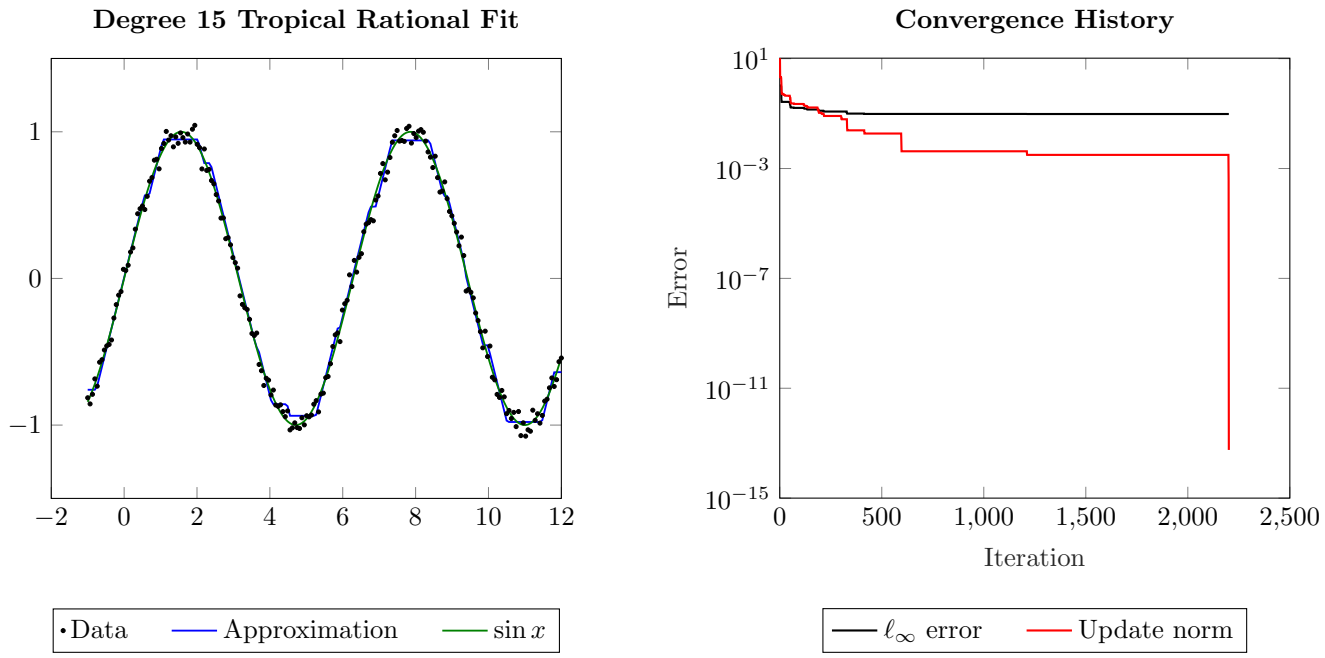
In this section, we use Algorithm 1 for regression tasks and examine its convergence behavior empirically. We provide univariate, bivariate, and higher dimensional examples. In the univariate case we analyze the relationship between the degree hyperparameter and the error in the computed fit. In the bivariate case, we analyze the effect of precomposition with a scaling parameter c as in Corollary 2.3.1. For six variable functions, we examine the use of Algorithm 1 on data generated from tropical rational functions. We then present the performance of our approach on the existing datasets used by [19, 26, 31]. Finally, we present preliminary experiments using the output of Algorithm 1 to initialize ReLU neural networks. All Matlab and Python codes to reproduce our experiments can be found at

https://github.com/Alex-Dunbar/Tropical-Data.git.

4.1 Univariate Data

We apply Algorithm 1 to a dataset consisting of 200 equally spaced points $x^{(i)} \in [-1, 12]$ and corresponding y values $y^{(i)} = \sin(x^{(i)}) + \epsilon^{(i)}$, where $\epsilon^{(i)}$ is drawn independently from a Gaussian distribution with mean 0 and standard deviation 0.05. Figure 2 shows an example, with $d = 15$. We use a stopping criterion of $\eta^k \leq 10^{-12}$. The infinity norm of the error and the infinity norm of the update step at each iteration are plotted in Figure 2b. Both the training loss and the update norm are nonincreasing and have regions on which they are constant.

Effect of Degree Here, we investigate the relationship between the degree of tropical rational function and the error in the fit. Specifically, we generate a dataset as in the above example and use Algorithm 1 to fit a tropical rational function of degree d to the dataset for $d = 1, 2, \ldots, 20$. As a stopping criterion in Algorithm 1, we use $\eta^k \leq 10^{-12}$ or a maximum $k_{\max} = 10000$. Figure 3 shows the relationship between the degree of the rational function and the error in the fit. Note that the error decreases as a function of the degree with a large decrease in error when the degree is 5. The number of iterations needed to achieve the stopping criterion is generally increasing but is not monotonic.



(a) Tropical rational approximation

(b) Training loss and update norm

Figure 2: Results of applying Algorithm 1 with degree 15 tropical rational functions to noisy data from a sine curve. Figure 2a shows the training data, the approximation by a tropical rational function, and the function $\sin x$. The approximating function captures the general behavior of the dataset. Figure 2b shows the ℓ_∞ error $e^k = \|\mathbf{X} \boxplus \mathbf{p}^k - \mathbf{X} \boxplus \mathbf{q}^k - \mathbf{y}\|_\infty$ and the update norm $\eta^k = \| \begin{bmatrix} \mathbf{p}^{k+1} & \mathbf{q}^{k+1} \end{bmatrix}^\top - \begin{bmatrix} \mathbf{p}^k & \mathbf{q}^k \end{bmatrix}^\top \|_\infty$. Both the training loss and the update norm are nonincreasing and contain intervals on which they are nearly constant.

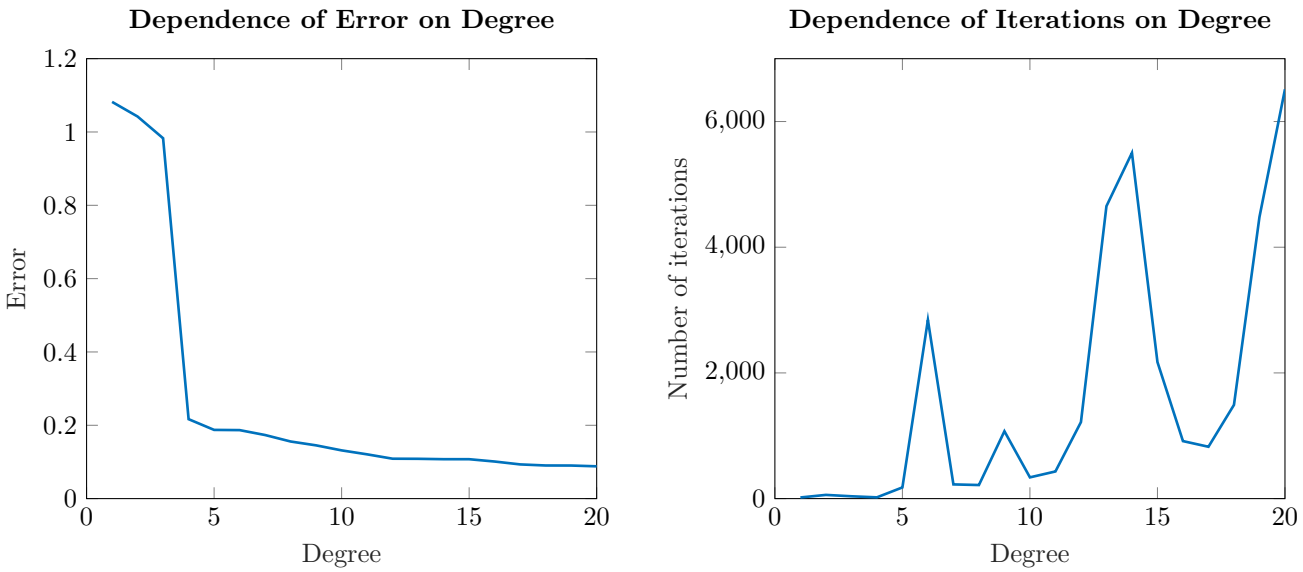


Figure 3: Dependence of error and number of iterations on degree of tropical rational function fit to noisy data from a sine curve. The error decreases monotonoically as a function of degree with a large drop at degree 5. The number of iterations needed to reach the stopping criterion of $\eta^k \leq 10^{-12}$ generally increases with the degree.

4.2 Bivariate Data

We use the method to approximate the Matlab `peaks` dataset using $\mathbf{d} = (10, 10)$ and $\mathbf{d} = (31, 31)$ and training until $\eta^k \leq 10^{-12}$. Explicitly, the `peaks` dataset consists of $2401 = 49^2$ equally spaced (x_1, x_2) pairs in $[-3, 3]^2$ and their evaluations

$$\texttt{peaks}(x_1, x_2) = 3(1-x_1)^2 e^{-x_1^2-(x_2+1)^2} - 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right) e^{-x_1^2-x_2^2} - \frac{1}{3} e^{-(x_1+1)^2-x_2^2}.$$

The fits and the error are shown below in Figure 4. Note that in both cases there is error in the regions on which the data is nearly constant despite the piecewise linear nature of the tropical rational functions. As in the univariate case, the training error and the update norm are nonincreasing and have regions where they are constant over many iterations.

Effect of Scaling Parameter In the above experiments, we directly fit a tropical rational function to the data. However, Corollary 2.3.1 suggests that we should fit a function of the form $f(c\mathbf{x})$, where $c \in \mathbb{R}$ and f is a tropical rational function. To this end, we fit functions of the form $f(c\mathbf{x})$ for 21 equally spaced values of $c \in [1, 3]$ and f a tropical rational function of degree 35. For each value of c, we use a stopping criterion of $\eta^k \leq 10^{-12}$ or a maximum of 500 iterations of the alternating method described in Algorithm 1 to find a tropical rational function f. The dependence of the training error on c is shown in Figure 5 below. Note that the optimal value of c in this range is roughly 1.3. More generally, for fixed degree d, changing the value of c gives a trade-off between maximum slope and resolution between slopes. Due to this trade-off, there will, in general, be large errors for very large c because each affine piece of the tropical polynomials $p(c\mathbf{x})$ and $q(c\mathbf{x})$ will have large slopes. Conversely, there will be large errors for very small values of c because the slopes of the affine pieces of the polynomials $p(c\mathbf{x})$ and $q(c\mathbf{x})$ will be bounded.

4.3 Higher Dimensional Examples

We test Algorithm 1 on functions with many variables. These experiments suggest that the alternating minimization method is able to find solutions with low training loss. However, these solutions do not appear to generalize well, even on data generated from tropical rational functions.

Regression on 6 Variable Function We fit a tropical rational function to the 6 variable function

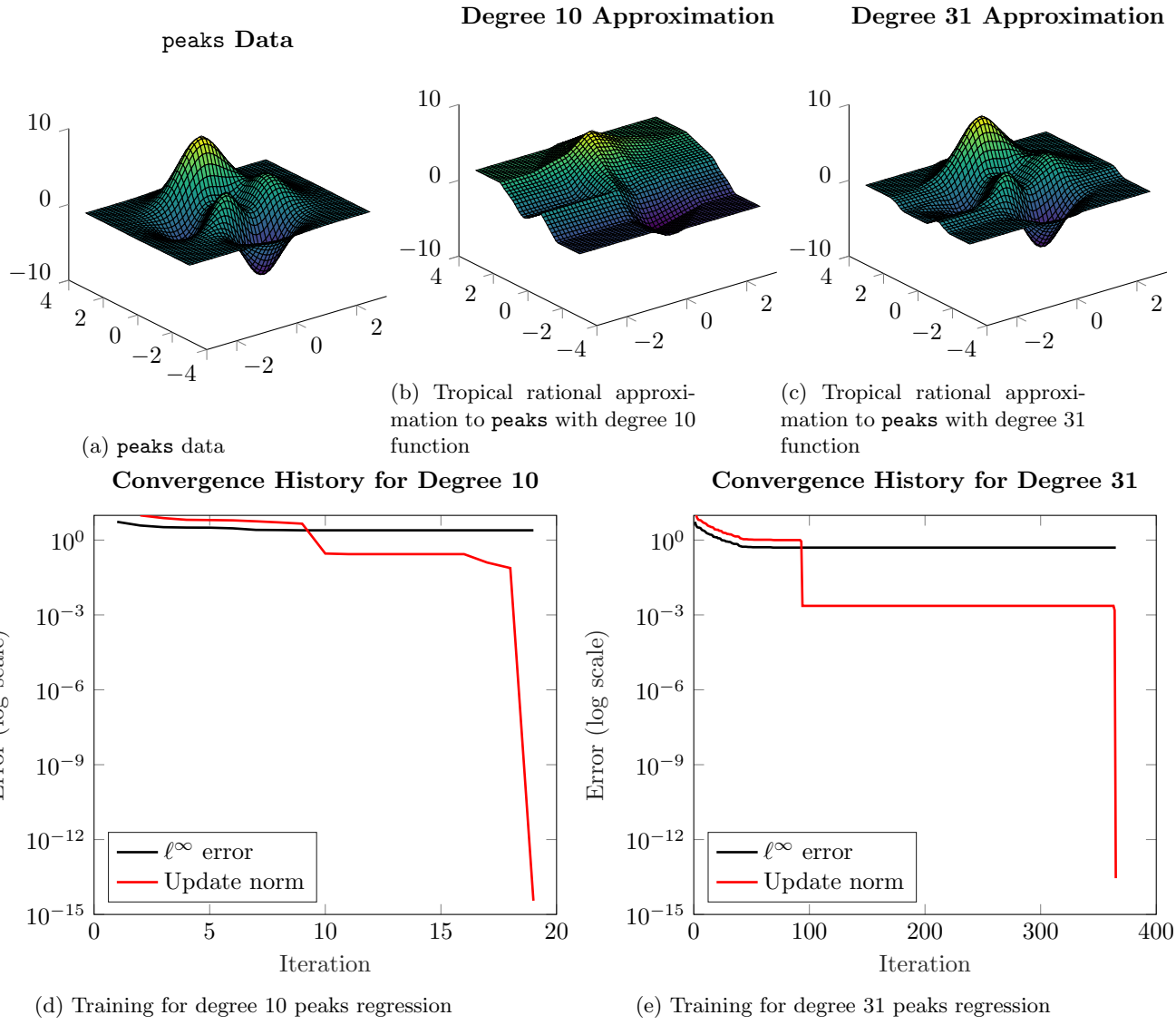
$$g(\mathbf{x}) = x_1x_2x_3 + 2x_4x_5^2\sin(x_6^2)$$

on a training set consisting of $N = 10000$ points drawn uniformly at random from $[0, 1]^6$ and then test on a test set generated in the same way. Here, we fix the maximum degree of the numerator and denominator to be 3 for each variable and train until $\eta^k \leq 10^{-12}$ or for a maximum of 500 iterations. There are 8192 trainable parameters. The convergence behavior during training is shown in Figure 6a. The ℓ^∞ error on the test set is 0.2721, which is roughly 9.75 times the final training error of 0.0279.

Regression on 10 Variable Function We fit a tropical rational function to the 10 variable function

$$h(\mathbf{x}) = x_1x_2x_3 + 2x_4x_5^2\sin(x_6^2) - e^{x_7x_8x_9x_{10}}$$

on a training set consisting of $N = 10000$ points drawn uniformly at random from $[0, 1]^{10}$ and then test on a test set generated in the same way. Here, we fix the maximum degree of the numerator and denominator to be 1 for each variable and train until $\eta^k \leq 10^{-12}$ or for a maximum of 500 iterations. There are 2048 trainable parameters. The convergence behavior during training is shown in Figure 6b. The ℓ^∞ error on the test set is 0.6828, which is roughly 2.9 times the final training error of 0.2342.



(a) peaks data

(b) Tropical rational approximation to peaks with degree 10 function

(c) Tropical rational approximation to peaks with degree 31 function

(d) Training for degree 10 peaks regression

(e) Training for degree 31 peaks regression

Figure 4: Results of applying Algorithm 1 with degree 10 and 31 tropical rational functions to the peaks dataset. The resulting degree 31 function sketches the general behavior of the dataset (Figure 4c), while the degree 10 function fails to approximate the data (Figure 4b). Figures 4d and 4e display the the ℓ^∞ error $e^k = \|\mathbf{X} \boxplus \mathbf{p}^k - \mathbf{X} \boxplus \mathbf{q}^k - \mathbf{y}\|_\infty$ and the update norm $\eta^k = \| \begin{bmatrix} \mathbf{p}^{k+1} & \mathbf{q}^{k+1} \end{bmatrix}^\top - \begin{bmatrix} \mathbf{p}^k & \mathbf{q}^k \end{bmatrix}^\top \|_\infty$. For both degrees, the training loss and the update norm are each nonincreasing and contain intervals on which they are nearly constant.

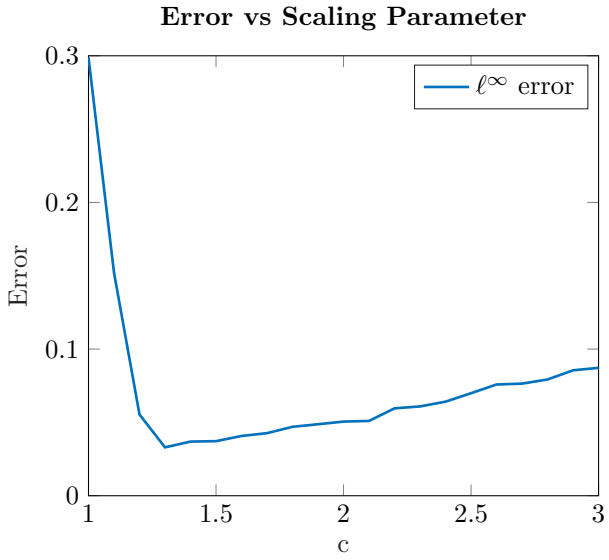
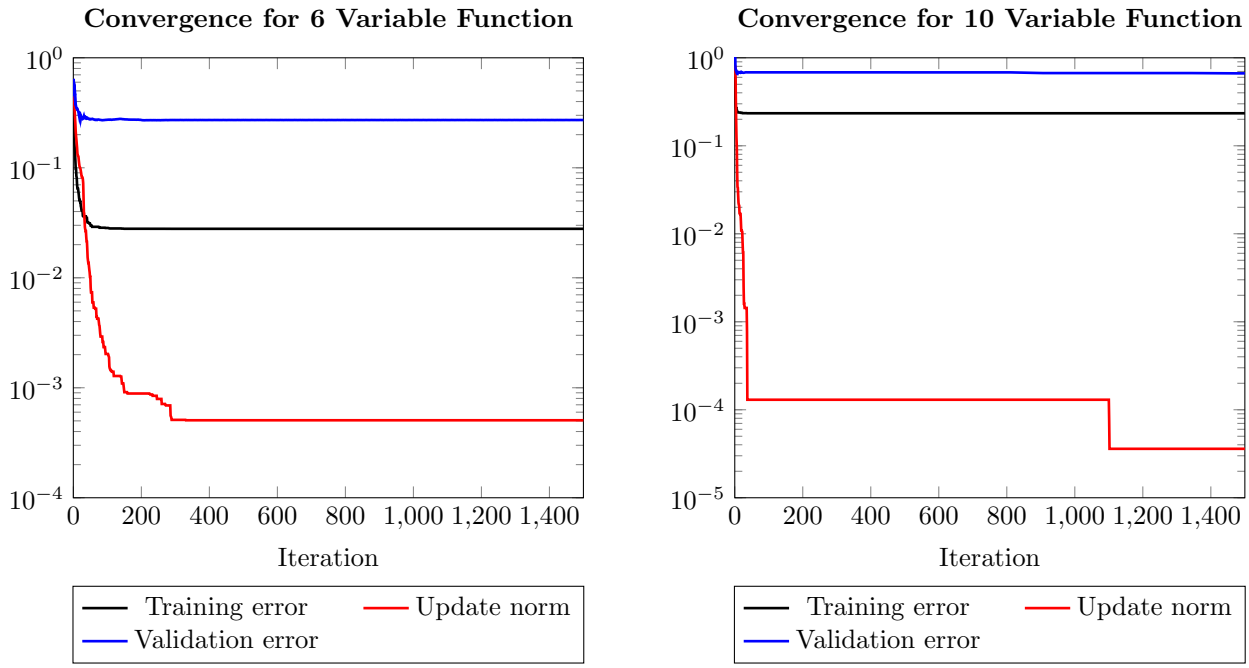


Figure 5: Error in approximation to the `peaks` dataset when using a degree $(35, 35)$ tropical rational function with inputs scaled by c. Here, the optimal value of c in the range $1 \leq c \leq 3$ is roughly 1.3 and gives a much lower training error than the function obtained as the output of Algorithm 1 with unscaled inputs.

Degree	Relative Training Error	Relative Validation Error
1	2.372×10^{-15}	0.1271
2	5.869×10^{-15}	0.2019
3	9.108×10^{-15}	0.2869
4	1.286×10^{-14}	0.3631
5	9.373×10^{-6}	0.3598

Table 1: Average training and validation error on data generated from 6 variable tropical rational functions. For each degree, the training loss is low, but the validation error is high and increasing as a function of degree.

Data Generated from Tropical Rational Functions Here, we investigate the use of Algorithm 1 on data generated by tropical rational functions. Specifically, for $n = 6$ we investigate the use of Algorithm 1 for the recovery of a tropical rational function of degrees 1 through 5 (i.e. $W_d = \{0, 1, 2, \ldots, d\}^6$ for $1 \leq d \leq 5$). For each trial we generate a tropical rational function with coefficients sampled uniformly at random from $[-5, 5]$ as well as training and validation datasets of $N = 10000$ points sampled uniformly at random from $[-5, 5]^6$. We then fit a tropical rational function $\hat{f}$ of the same degree using Algorithm 1 with a stopping criterion of $\eta^k \leq 10^{-8}$ or a maximum of 1000 iterations. In degrees at most 4, the method reached the stopping criterion in fewer than 1000 iterations for each trial. For degree 5, the method terminated after reaching 1000 iterations in 3 trials. In this experiment, $\mathbf{p}^0$ and $\mathbf{q}^0$ are initialized with entries drawn uniformly at random from $[-5, 5]^6$. Table 1 shows the average relative training and validation loss $\|\hat{f}(\mathbf{x}) - \mathbf{y}\|_\infty / \|\mathbf{y}\|_\infty$ across the five trials in each degree. Here, the training loss is low, indicating that Algorithm 1 finds a near optimal solution. However, the validation loss is high and increasing as a function of the degree. This indicates that when run to completion, Algorithm 1 solves the optimization problem (1) well. However, the higher validation errors suggest that the solution to (1) is nonunique. In particular, Algorithm 1 does not necessarily recover the coefficients of the tropical rational function used to generate the data.



(a) Degree 3 fit for 6 variable function with 10,000 data points

(b) Degree 1 fit for 10 variable function and 10,000 data points

Figure 6: Convergence for tropical rational approximation of 6 and 10 variable functions. The training error and update norm display similar behavior as in the low dimensional cases with regions on which they remain constant.

4.4 Performance on Existing Datasets

In this section, we test the performance of our method on datasets generated from convex functions presented in [26] and datasets generated from nonconvex functions presented in [19, 31].

Convex Functions Here, we use datasets from [26] and demonstrate that Algorithm 1, as a generalization of tropical polynomial regression, can be used to approximate convex functions. First, we generate data points $(x^{(i)}, g(x^{(i)}))$, where the $x^{(i)}$ are 200 equally spaced points on the interval $[-1, 12]$ and $g(x^{(i)}) = \max(3, x^{(i)} - 2) + \epsilon^{(i)}$ is a tropical line, where $\epsilon^{(i)}$ is drawn independently from a uniform distribution on $[-0.5, 0.5]$. We fit a tropical rational function and a tropical polynomial with $W = \{0, 1\}$. The results are plotted in Figure 7. Note that there is a deviation between the two approximations near $x = 11$, where the tropical rational approximation becomes nonconvex.

Next, we generate 500 pairs $(x_1^{(i)}, x_2^{(i)}) \in [-1, 1]^2$ uniformly at random and set $y^{(i)} = (x_1^{(i)})^2 + (x_2^{(i)})^2 + \epsilon^{(i)}$, where the $\epsilon^{(i)}$ are drawn independently from a normal distribution with mean 0 and variance 0.25^2. We then fit tropical rational functions of degrees $d = 1, 2, \ldots, 6$ to the data and record the error. Figure 8 shows the average and worst error across 25 such trials as well as the results reported in [26, Table 1]. Note that in the experimental setup of [26], tropical polynomials are fit to the data where the monomials are chosen via k-means clustering. In our setup, exponents for the numerator and denominator polynomials are $\{0, 1, \ldots, d_{max}\}^2$. It appears that the approach in [26] yields a lower error in the low-parameter setting, while the rational regression leads to lower training error in the high-parameter setting despite the data-independent monomial selection.

Nonconvex Functions Here, we test the performance of Algorithm 1 on nonconvex functions tested by [19, 31]. Specifically, we consider the functions

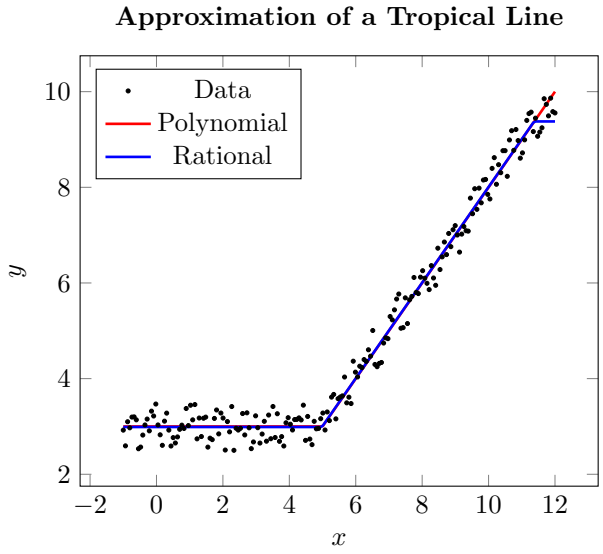


Figure 7: Approximation of a tropical line using tropical polynomial regression and Algorithm 1. Note that the two fits diverge near $x = 11$, where the tropical rational approximation becomes nonconvex.

$$g_1(x_1, x_2) = x_1^2 - x_2^2 \qquad \text{for } (x_1, x_2) \in [0.5, 7.5] \times [0.5, 3.5]$$
$$g_2(x_1, x_2) = x_2^2 \frac{\sin(x_1)}{x_1} \qquad \text{for } (x_1, x_2) \in [1, 3] \times [1, 2]$$
$$g_3(x_1, x_2) = \exp(-10(x_1^2 - x_2^2)^2) \qquad \text{for } (x_1, x_2) \in [1, 2] \times [1, 2].$$

For each function, we study the effect of degree of tropical rational function and number of data points used on the error and solution time. Specifically, for $N \in \{10^2, 20^2, 50^2, 100^2\}$, we generate a dataset of N equally spaced gridpoints $(x_1^{(i)}, x_2^{(i)})$ of the function domain and their evaluations $g_j(x_1^{(i)}, x_2^{(i)})$ and use Algorithm 1 to fit tropical rational functions of degrees $1, 2, \ldots, 25$. The results are displayed in Figure 9. Note that the training errors are comparable to those tested in [19].

4.5 ReLU Neural Network Initialization

Here we investigate the use of Algorithm 1 to initialize the weights of a ReLU neural network. The motivation for this approach is that the output of Algorithm 1 carries information about the training data. So, networks with weights initialized from the output of the tropical regression heuristic should start from a lower loss than those with weights drawn from a distribution that does not depend on the training data. Additionally, the computational cost of performing an iteration of Algorithm 1 is $\mathcal{O}(ND)$, which is comparable to the cost of an epoch of stochastic gradient descent for a network with D parameters. As noted in Section 2.3, the networks that we are able to initialize have significantly more than D parameters. Despite the potential advantages of a tropical initialization, we find that such a scheme does not always lead to faster convergence or lower training or validation error. Moreover, the network architectures which we are able to initialize using a tropical rational function appear to have unstable training, even when initialized using well-known strategies.

In our experiments, we apply Algorithm 1 on data from the noisy sine curve and `peaks` datasets to generate approximations of the data, then use the output tropical rational function to initialize the weights of ReLU networks. The architecture of the initialized network is determined by the number of monomials in the tropical rational function f used to initialize the network. For each dataset, we compare networks of the same architecture using the following initialization strategies:

- Repeated applications of (6) to the terms of the tropical rational function f output by Algorithm 1

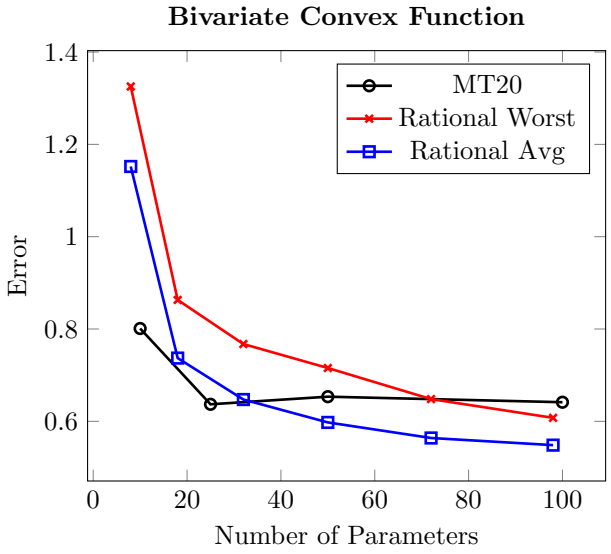


Figure 8: Errors in tropical approximations to a bivariate dataset generated from a convex function. Note that despite a data-blind choice of exponents, in the large parameter setting, tropical rational regression using Algorithm 1 produces approximations with lower training error than that reported in [26, Table 1], which used tropical polynomials with exponents chosen via k-means clustering.

- He initialization [16]
- Weights and biases drawn uniformly at random from $[-k, k]$[1], where $k = \sqrt{\frac{1}{\text{number of inputs}}}$ for each layer

All neural network parameter optimization is done in PyTorch version 1.11.0 using the Adam optimizer [20] to minimize the MSE loss.

4.5.1 Univariate Data

We use a degree 15 tropical rational function to initialize a neural network to fit the noisy sin curve from above. The test data consists of 200 pairs $(x^{(i)}, y^{(i)})$, where $x^{(i)}$ is randomly drawn points on the interval $[-1, 12]$ and $y^{(i)} = \sin(x^{(i)})$. The networks are trained for 1000 epochs with batches of size 64 and a learning rate of 5×10^{-6} for the tropical initialized network and 10^{-2} for the He-initialized and uniformly-initialized networks. We found choosing a smaller learning rate for the tropical initialization important to prevent the optimization from reducing the accuracy of the model. Training and validation errors are shown in Figure 10. The network initialized from a tropical rational function has lower training and validation error than the network initialized using the other methods.

4.5.2 Bivariate Data

We use a degree 31 tropical rational function to initialize the `peaks` dataset using Algorithm 1 as the initialization. The networks are trained for 1000 epochs with a batch size of 64 and a learning rate of 10^{-4} for He-initialized and uniformly-initialized networks and 10^{-7} for the tropically initialized network. Results are shown in Figure 11. The networks initialized with He initialization and with uniform initialization reach lower training and validation errors than the tropically initialized network.

[1]This is the default initialization for linear layers in PyTorch version 1.11.0

5 Discussion

We investigated the solution of regression with tropical rational functions by presenting an alternating heuristic. The proposed heuristic leverages known algebraic structure in tropical polynomial regression to iteratively fit numerator and denominator polynomials. Each iteration involves only (tropical) matrix-vector products and vector addition. The error at each iterate is nonincreasing, and each iterate is located in the nondifferentiability locus of the ℓ^∞ loss function. Computational experiments demonstrate that our method can produce a qualitatively reasonable approximation of the input data. However, the optimal error and optimality conditions are unknown in general, preventing a quantitative evaluation of the heuristic. On datasets generated from tropical rational functions of low degrees where the true optimal error is known to be zero, the heuristic produces an approximation with very low training error. However, the validation error is large in our experiments, suggesting a need for future work in developing regularization strategies

One potential application domain is in ReLU network initialization. In this work, we successfully initialized a ReLU network using a tropical rational function for a univariate regression task, while the tropical initialization was outperformed by existing initialization strategies for a bivariate regression task. This indicates the potential for future work to develop a better understanding of network initialization. In particular, the network architectures used in our experiments are limited, and a full understanding of correspondences between network architectures and tropical functions is currently an open problem.

Future work could help to develop a better theoretical understanding of the convergence behavior of Algorithm 1. In most numerical experiments in Section 4, the method reaches the stopping criterion of sufficiently small update steps; however, the number of iterations required to reach this criterion appears highly sensitive to the structure of the underlying data as well as the degree of the approximating function. For example, when making approximations with degree 3 tropical rational functions of 6 variables, the update steps converged to 0 for datasets generated from tropical rational functions, while the update steps failed to converge to 0 before a maximum iteration count on data generated from an arbitrary nonconvex function.

Additionally, future work could augment the polynomial regression steps using the ideas in [17, 40, 41] to develop variants of Algorithm 1 for use with different norms or which enforce sparsity patterns or a regularization term. More generally, the development of a procedure for monomial selection remains open.

6 Acknowledgments

We would like to thank two anonymous referees for their insightful feedback which improved the paper. Additionally, we would like to thank Georg Loho for bringing the reference [7] to our attention. This work was supported in part by NSF awards DMS 1751636, DMS 2038118, AFOSR grant FA9550- 20-1-0372, and US DOE Office of Advanced Scientific Computing Research Field Work Proposal 20-023231.

References

[1] Alekh Agarwal et al. "Learning Sparsely Used Overcomplete Dictionaries via Alternating Minimization". In: *SIAM Journal on Optimization* 26.4 (2016), pp. 2775–2799.

[2] Marianne Akian et al. "Best approximation in max-plus semimodules". In: *Linear Algebra and its Applications* 435.12 (2011), pp. 3261–3296.

[3] Raman Arora et al. "Understanding Deep Neural Networks with Rectified Linear Units". In: *International Conference on Learning Representations (ICLR)*. 2018.

[4] Amir Beck. "First-order methods in optimization". In: vol. 25. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA, 2017, pp. xii+475.

[5] Erwan Brugallé et al. *Brief introduction to tropical geometry*. 2015.

[6] Vasileios Charisopoulos and Petros Maragos. *A Tropical Approach to Neural Networks with Piecewise Linear Activations*. 2019.

[7] R.A. Cuninghame-Green and P. Butkovic. "The equation $A \otimes x = B \otimes y$ over (max,+)". In: *Theoretical Computer Science* 293.1 (2003). Max-PLus Algebras, pp. 3–12.

[8] Raymond Cuninghame-Green. *Minimax Algebra*. Ed. by M. Beckmann and H. P. Künzi. Vol. 166. Lecture Notes in Economics and Mathematical Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 1979.

[9] I. Daubechies et al. "Nonlinear Approximation and (Deep) ReLU Networks". en. In: *Constructive Approximation* 55.1 (Feb. 2022), pp. 127–172.

[10] Bernd Gärtner and Martin Jaggi. *Tropical support vector machines*. Tech. rep. ACS Technical Report No.: ACS-TR-362502-01. 2008.

[11] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256.

[12] J. Elisenda Grigsby and Kathryn Lindsey. "On Transversality of Bent Hyperplane Arrangements and the Topological Expressiveness of ReLU Neural Networks". In: *SIAM Journal on Applied Algebra and Geometry* 6.2 (2022), pp. 216–242.

[13] Boris Hanin and David Rolnick. "Deep ReLU Networks Have Surprisingly Few Activation Patterns". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[14] Boris Hanin and David Rolnick. "How to Start Training: The Effect of Initialization and Architecture". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[15] Philip Hartman. "On functions representable as a difference of convex functions". In: *Pacific J. Math.* 9 (1959), pp. 707–713.

[16] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015.

[17] James Hook. "Max-plus linear inverse problems: 2-norm regression and system identification of max-plus linear dynamical systems with Gaussian noise". In: *Linear Algebra and its Applications* 579 (2019), pp. 1–31.

[18] Michael Joswig and Georg Loho. "Monomial Tropical Cones for Multicriteria Optimization". en. In: *SIAM Journal on Discrete Mathematics* 34.2 (Jan. 2020), pp. 1172–1191.

[19] Kody Kazda and Xiang Li. "Nonconvex multivariate piecewise-linear fitting using the difference-of-convex representation". en. In: *Computers & Chemical Engineering* 150 (July 2021), p. 107310.

[20] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014.

[21] Nikolai Krivulin. "Algebraic Solution of Tropical Best Approximation Problems". In: *Mathematics* 11.18 (2023).

[22] Nikolai K Krivulin. "On the Solution of a Two-Sided Vector Equation in Tropical Algebra". In: *Vestnik St. Petersburg University, Mathematics* 56.2 (2023), pp. 172–181.

[23] Diane Maclagan and Bernd Sturmfels. *Introduction to Tropical Geometry*. Vol. 161. Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2015, pp. vii+359.

[24] Alessandro Magnani and Stephen P. Boyd. "Convex piecewise-linear fitting". en. In: *Optimization and Engineering* 10.1 (Mar. 2009), pp. 1–17.

[25] Petros Maragos, Vasileios Charisopoulos, and Emmanouil Theodosis. "Tropical Geometry and Machine Learning". In: *Proceedings of the IEEE* 109.5 (2021), pp. 728–755.

[26] Petros Maragos and Emmanouil Theodosis. "Multivariate tropical regression and piecewise-linear surface fitting". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 3822–3826.

[27] Petros Maragos and Emmanouil Theodosis. *Tropical Geometry and Piecewise-Linear Approximation of Curves and Surfaces on Weighted Lattices*. 2019.

[28] Smitha Milli et al. "Model Reconstruction from Model Explanations". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. FAT* '19. Atlanta, GA, USA: Association for Computing Machinery, 2019, pp. 1–9.

[29] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.

[30] Lior Pachter and Bernd Sturmfels. "Tropical geometry of statistical models". en. In: *Proceedings of the National Academy of Sciences* 101.46 (Nov. 2004), pp. 16132–16137.

[31] Steffen Rebennack and Josef Kallrath. "Continuous piecewise linear delta-approximations for bivariate and multivariate functions". In: *Journal of Optimization Theory and Applications* 167.1 (2015), pp. 102–117.

[32] David Rolnick and Konrad Kording. "Reverse-engineering deep ReLU networks". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 8178–8187.

[33] Georgios Smyrnis and Petros Maragos. "Multiclass neural network minimization via tropical newton polytope approximation". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9068–9077.

[34] Georgios Smyrnis and Petros Maragos. "Tropical Polynomial Division and Neural Networks". In: *CoRR* abs/1911.12922 (2019).

[35] Xiaoxian Tang, Houjie Wang, and Ruriko Yoshida. *Tropical Support Vector Machine and its Applications to Phylogenomics*. 2020.

[36] Alejandro Toriello and Juan Pablo Vielma. "Fitting piecewise linear continuous functions". en. In: *European Journal of Operational Research* 219.1 (May 2012), pp. 86–95.

[37] Ngoc Mai Tran and Jidong Wang. *Minimal Representations of Tropical Rational Signomials*. 2022.

[38] Martin Trimmel, Henning Petzka, and Cristian Sminchisescu. "TropEx: An Algorithm for Extracting Linear Terms in Deep Neural Networks". In: *International Conference on Learning Representations*. 2021.

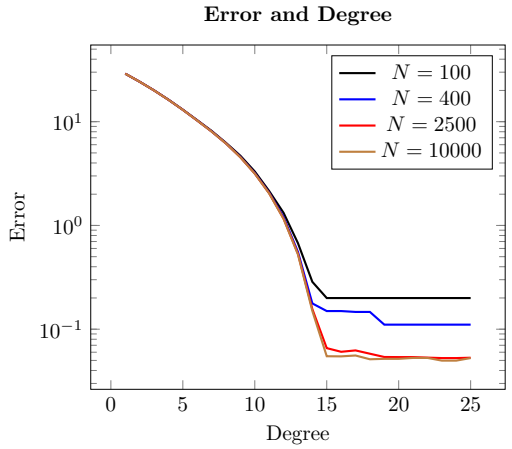
[39] Paul Tseng. "Applications of a Splitting Algorithm to Decomposition in Convex Programming and Variational Inequalities". In: *SIAM Journal on Control and Optimization* 29.1 (1991), pp. 119–138.

[40] Anastasios Tsiamis and Petros Maragos. "Sparsity in max-plus algebra and systems". In: *Discrete Event Dynamic Systems* 29.2 (2019), pp. 163–189.

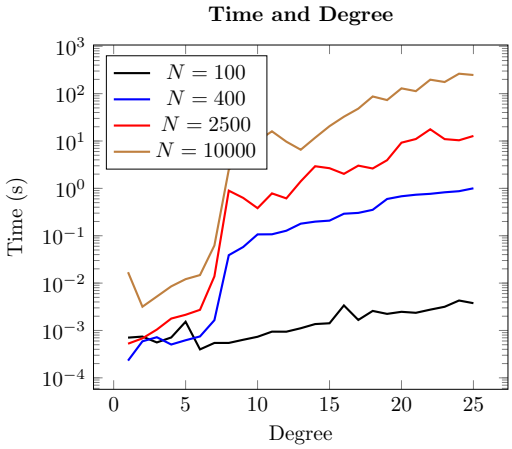
[41] Nikolaos Tsilivis, Anastasios Tsiamis, and Petros Maragos. "Toward a Sparsity Theory on Weighted Lattices". In: *Journal of Mathematical Imaging and Vision* (2022), pp. 1–13.

[42] Ruriko Yoshida et al. "Tropical Support Vector Machines: Evaluations and Extension to Function Spaces". In: *CoRR* abs/2101.11531 (2021).

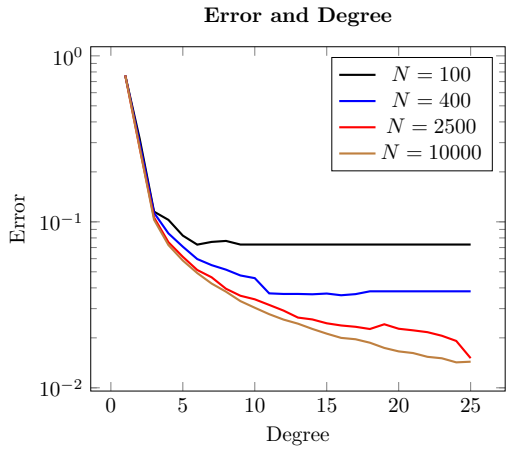
[43] Liwen Zhang, Gregory Naitzat, and Lek-Heng Lim. "Tropical geometry of deep neural networks". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5824–5832.



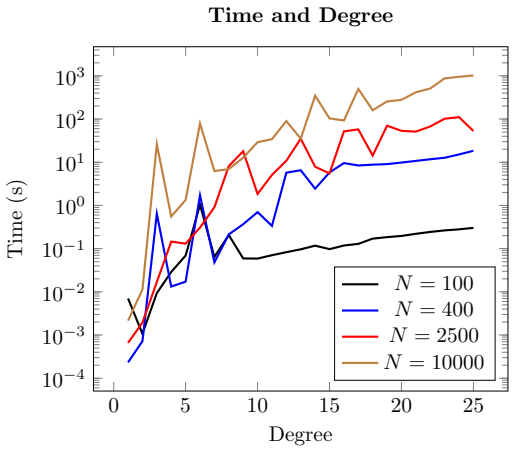
(a) $g_1(x_1, x_2) = x_1^2 - x_2^2$



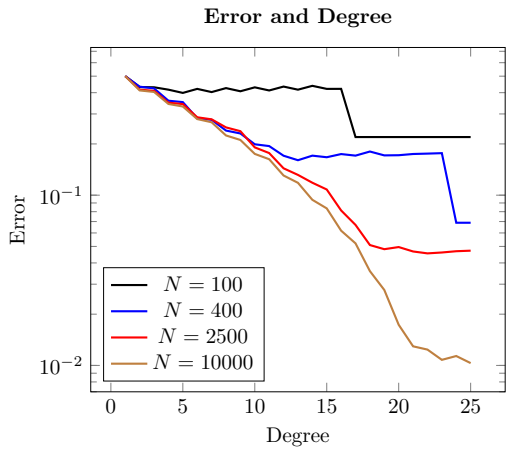
(b) $g_1(x_1, x_2) = x_1^2 - x_2^2$



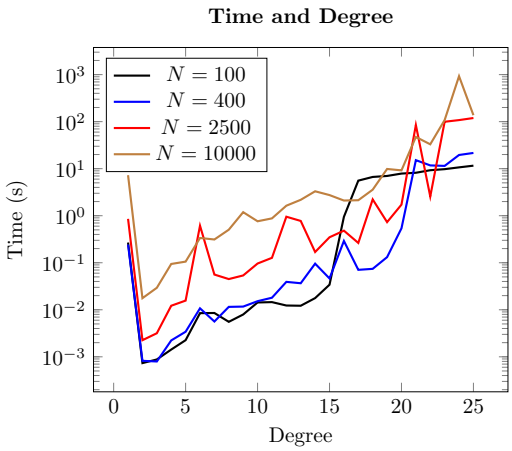
(c) $g_2(x_1, x_2) = x_2^2 \sin(x_1)/x_1$



(d) $g_2(x_1, x_2) = x_2^2 \sin(x_1)/x_1$



(e) $g_3(x_1, x_2) = \exp(-10(x_1^2 - x_2^2)^2)$



(f) $g_3(x_1, x_2) = \exp(-10(x_1^2 - x_2^2)^2)$

Figure 9: Performance of Algorithm 1 on datasets generated from nonconvex bivariate functions. The plots on the left display the relationship between error, degree, and number of sample points, while the figures on the right show the dependence of computation time on degree and number and sample points.

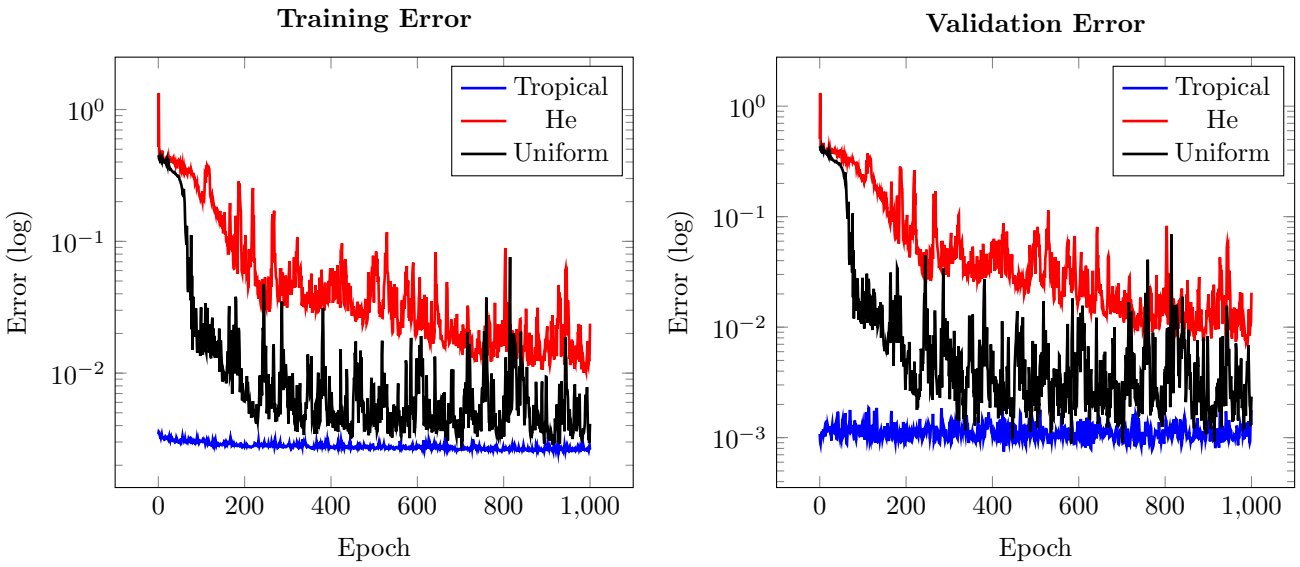


Figure 10: Training and validation errors for neural network fit to noisy sin data. The network initialized from a tropical rational approximation to the dataset starts and remains at lower training and validation losses than the networks initialized with the He Initialization [16]. and with uniform initialization

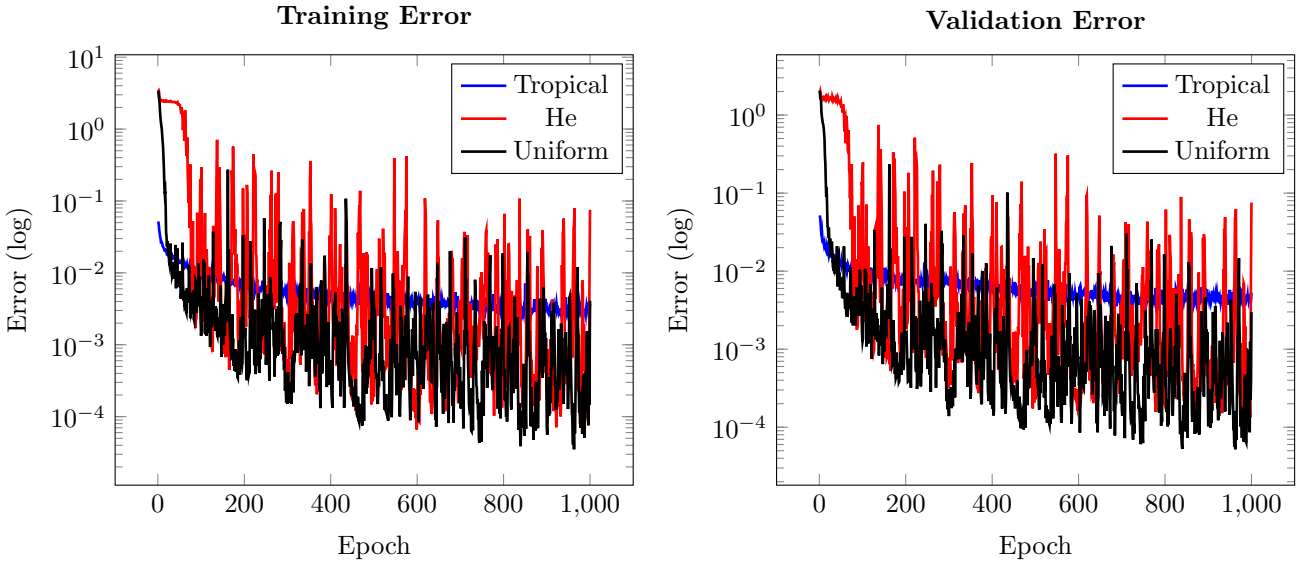


Figure 11: Training and validation errors for neural network fits to `peaks` data. The networks initialized the He Initialization [16] and uniform initialization reach lower training and validation errors than the tropically initialized network.