Predicting Polytope Areas with One-Hidden-Layer ReLU Neural Networks: An Analysis of Accuracy

Abstract

We study the accuracy of a neural network (NN) with one hidden layer and ReLU activation in predicting the area of polytopes after a change of coordinates. Specifically, we train a NN with respect to a given database and then we apply a change of coordinates to create a transformed dataset of polytopes with different vertices but the same area. Then, we calculate the mean absolute error (MAE) of our trained neural network in estimating the area of the polytopes in the transformed datasets.

1 Preliminaries

In this section, we introduce some definitions and notation that will be used throughout the paper.

1.1 Labeled Databases of Polytopes

Given a finite collection of points $q_1, \cdots, q_n$ in the plane with integer coefficients, the lattice polytope defined by these points is given by

$$P := \left\{ \sum_{i=1}^{n} \lambda_i q_i \;\middle|\; 1 \geq \lambda_i \geq 0,\ \sum \lambda_i = 1 \right\}$$

Every lattice polytope P has a non-negative area, which can be computed with an equation known as Gauss' formula:

$$\text{Area}(P) = \frac{1}{2} \left(\sum_{i=1}^{n} \det \begin{pmatrix} q_{i1} & q_{(i+1)1} \\ q_{i2} & q_{(i+1)2} \end{pmatrix} \right).$$

where $q_i = (q_{i1}, q_{i2})$.

We will construct a labeled database, we construct a vector of coordinates from the vertices of a polytope. Then, to each vector, we can associate the area of the corresponding polytope. For example, given the pentagon in Figure 1, its area is 5/2. This database carries also the information of the vertices ordering. Indeed, there are 5! = 120 possible ways to order the entries. Therefore, we can obtain a coordinate vector such as

$$(0, 1, 1, 0, 0, -1, -1, -1, -1, 0) \longrightarrow 5/2,$$

or

$$(1, 0, -1, 0, 0, 1, 0, -1, -1, -1) \longrightarrow 5/2.$$

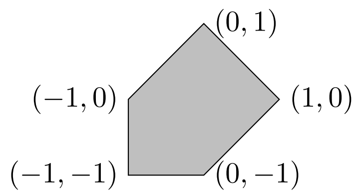


Figure 1: Pentagon and its vertices

For the same pentagon. The set of vertices of polytopes and their areas define a labeled database denoted as D and given by

$$D := \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \text{ vertices of } P_i,\ y_i = \text{Area}(P_i)\}.$$

1.2 Neural Networks

A neural network is certain function $F : \mathbb{R}^n \to \mathbb{R}^s$, where x is mapped to $\mathbf{x} = F(\mathbf{x})$, constructed by iterating linear maps and a type of nonlinear function known as an activation function. There are many types of neural networks with various complex features. In this work, we focus on neural networks with one hidden layer of m nodes:

$$F_m(\mathbf{x}) = \sum_{i=1}^{m} a_i \max\{0, \mathbf{w}_i \cdot \mathbf{x} + b_i\} \quad \mathbf{w}_i \in (\mathbb{R}^2)^n.$$

In our case, n is the number of vertices of the polytope. The coefficients $\mathbf{w}_i$ and a_i of the linear maps are called the weights of the neural network, and the b_i are the bias terms.

Given a labeled database D, to "train" a neural network means finding the values of the weights $\mathbf{w}_i$, a_i, and b_i that minimize the error between $F_m(\mathbf{x})$ and y. The final error is known as the training error:

$$R_T(F_m, D) = \frac{1}{|D|} \sum_{(\mathbf{x}_i, y_i) \in D} |F_m(\mathbf{x}_i) - y_i|^2.$$

The goal of training, which can be done via gradient descent, is to minimize this error and achieve a high degree of accuracy when predicting the areas of polytopes in a new dataset. We remark that such idea of using neural network in mathematical objects has been used in works such as (Berman et al., 2022), (Bernal et al., 2023), (He, 2022) and (Bao et al., 2021).

2 Our Mathematical Experiment

We start with the observation: We can change the coordinate system without altering the area of a polytope. For example, the change of coordinates $x \to x + y$ and $y \to x + 2y$ will transform the pentagon, see Figure 2, but keeping the area constant.

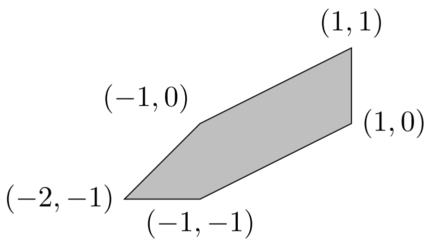


Figure 2: Pentagon after the change of coordinates $x \mapsto x + y$ and $y \mapsto y$

Given a labeled database of polytopes D and a change of coordinates induced by a matrix B with $\det(B) = 1$ and integer entries, we transform each polytope P as follows:

$$B \cdot P := \left\{ \sum_{i=1}^{n} \lambda_i (Bq_i) \;\middle|\; 1 \geq \lambda_i \geq 0, \sum \lambda_i = 1 \right\}$$

Notice that $\text{Area}(P) = \text{Area}(B \cdot P)$, then we can generate a new labeled database of polytopes:

$$B \cdot D := \{(\text{vertices}(B \cdot P), Area(P)) \mid P \text{ in } D\}.$$

To study the accuracy of our Neural Network, we complete the following steps:

1. We start with a database D of polytopes with a fixed number of vertices n. Each polytope has three vectors of coordinates (chosen randomly) so that the neural network does not learn the ordering of the vertices.

2. For a fixed number of nodes m, we train a ReLU neural network $F_m(x)$ with our original database D.

3. Given a change of coordinates induced by the matrix

$$B^k = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^k, \quad x \mapsto x + ky, \quad y \mapsto y,$$

we construct a database of polytopes $B^k \cdot D$. The matrix B is mathematically meaningful. It is a generator of the group $\mathrm{SL}(2, \mathbb{Z})$.

4. We calculate the error of using $F_m(\mathbf{x})$ to predict the areas of the new polytopes $B^k \cdot D$.

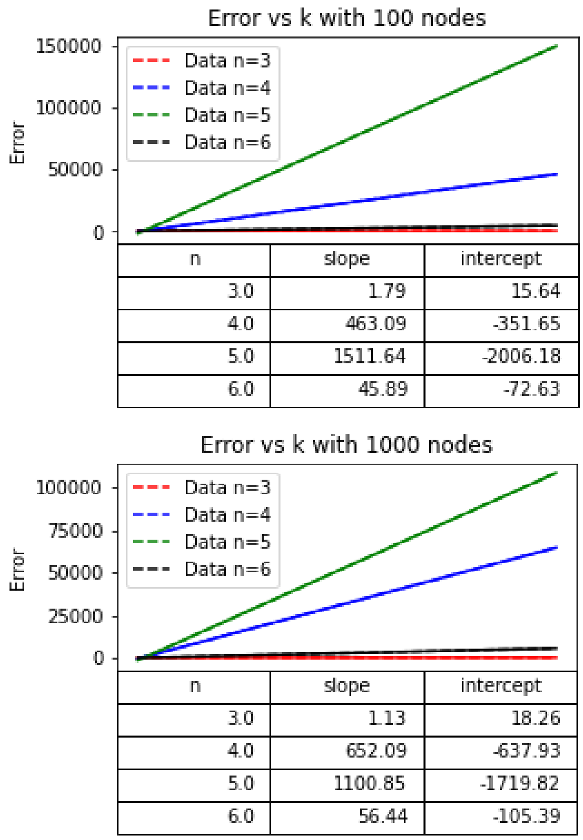
5. We plot the change in the error with respect to the exponent k.

These steps allow us to explore how accurately the neural network can predict the areas of transformed polytopes and how sensitive the prediction is to changes in the coordinate system.

3 Computational results

We completed the above steps for polytopes with $3 \leq n \leq 7$ vertices. In each case, we varied the number of nodes m used in the neural network. The following are our main results.

1. The error grows linearly. In all cases, we observed that the error changes linearly with respect to the parameter k.



n	slope	intercept
3.0	1.79	15.64
4.0	463.09	-351.65
5.0	1511.64	-2006.18
6.0	45.89	-72.63

n	slope	intercept
3.0	1.13	18.26
4.0	652.09	-637.93
5.0	1100.85	-1719.82
6.0	56.44	-105.39

Figure 3: Behavior of the error under change of coordinates $x \mapsto x + ky$ and $y \mapsto y$.

2. The impact of the number of nodes The slope and intercept of the linear equation that describe the error depend on the number of nodes in the hidden layer of the neural network. However, the relationship does not seem straightforward.

Next, we illustrate the cases for $n = 4, 5$ and $n = 7$.

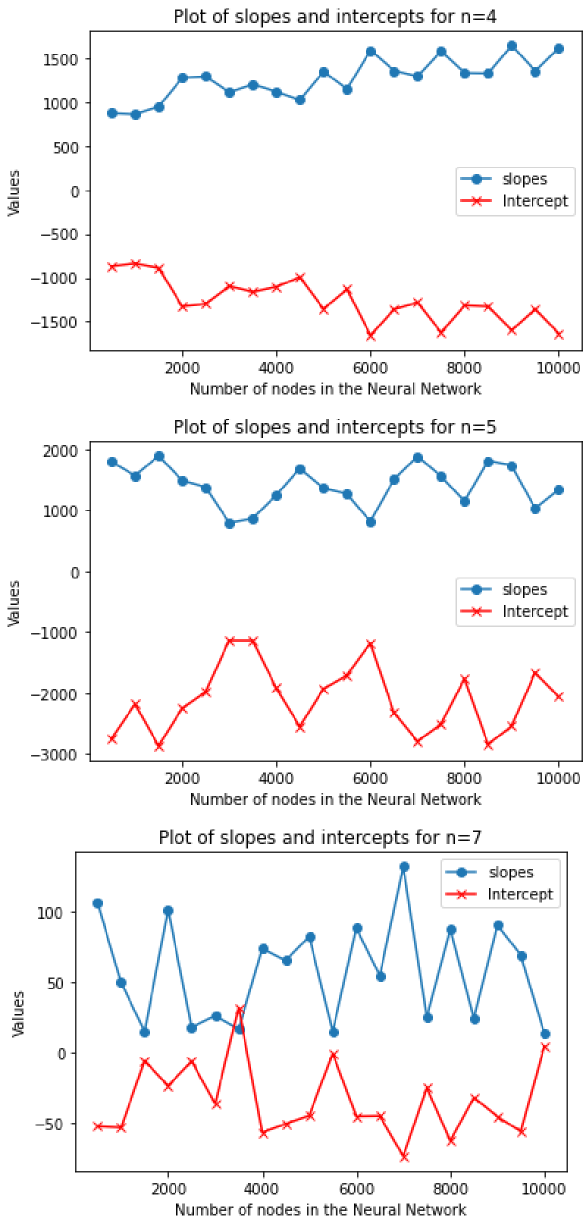


Figure 4: Behavior of the slopes (blue) and the intercets (red) with respect to the number of nodes.

4 Team

The results involved the work of the students Damla Erdogan, Andrew Krikorian, Rahul Kulkarni, Connor Stewart, with a support team of Madison Juliana Oliva and Mina Toufani.

References

Jiakang Bao, Yang-Hui He, Edward Hirst, Johannes Hofscheier, Alexander Kasprzyk, and Suvajit Majumder. 2021. Polytopes and machine learning. *arXiv preprint arXiv:2109.09602*.

David S Berman, Yang-Hui He, and Edward Hirst. 2022. Machine learning calabi-yau hypersurfaces. *Physical Review D*, 105(6):066002.

Edgar A Bernal, Jonathan D Hauenstein, Dhagash Mehta, Margaret H Regan, and Tingting Tang. 2023. Machine learning the real discriminant locus. *Journal of Symbolic Computation*, 115:409–426.

Yang-Hui He. 2022. Machine-learning mathematical structures. *International Journal of Data Science in the Mathematical Sciences*, pages 1–25.