Infinite-Width 1-Layer ReLU Networks with L2 Regularization on 2D Data

Juhyun Park

a senior thesis submitted in partial fulfillment
of the requirements for the degree of
Bachelor of Arts in Mathematics at
Princeton University

Adviser: Boris Hanin

May 1, 2023

Abstract

Given a dataset $\mathcal{D} = \{((x_1^{(i)}, x_2^{(i)}), y^{(i)})\}$ of two-dimensional input and one-dimensional output, we investigate the set of 1-layer ReLU networks $f(\mathbf{x}; \theta)$ that interpolate the dataset and, among such interpolants, minimize the ℓ_2 norm of the weights. In Section 3.1, we consider a dataset $\mathcal{D}$ where the points $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)})$ form the vertices of a regular polygon. We present the optimal network that assigns a non-zero value to one or two consecutive points on the polygon. Using this as a basic component, we present a heuristic of constructing an interpolant f of the dataset $\mathcal{D}$ which we believe to be near-optimal. In Section 3.2, we consider $\mathcal{D}$ to be symmetric with respect to a line ℓ. We show that if the dataset is effectively 1-dimensional, then any optimal f should also be 1-dimensional.

Acknowledgements

First and foremost, I would like to thank my adviser Prof. Hanin. This work would not have existed without his insight and guidance. I would also like to thank my second reader Prof. Sly who agreed to the position on such a short notice. I thank Hyungjun Choi and everyone else who provided useful insight and feedback.

This work also marks the conclusion of my undergraduate education. I express my gratitude to my parents who supported my education and to all of my friends who assisted me throughout this journey. And last but not least, I would like to thank my girlfriend April for providing the emotional support that I needed.

Declaration

I declare that I have not violated the Honor Code during the composition of this work. This paper represents my own work in accordance with University regulations.

I authorize Princeton University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purposes of scholarly research.

Contents

Chapter 1

Introduction

1.1 Background

In statistical modeling, we are given a set $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{n_d}$ of training data, where each data point represents a pair of input variables [1] $\mathbf{x}^{(i)} \in \mathbb{R}^{n_{in}}$ and output variables [2] $\mathbf{y}^{(i)} \in \mathbb{R}^{n_{out}}$. The main assumption is that there is an unknown relationship $f : \mathbb{R}^{in} \to \mathbb{R}^{out}$ such that $f(\mathbf{x}^{(i)}) = \mathbf{y}^{(i)}$. By using only the data points in the training dataset $\mathcal{D}$, the goal is to learn a function $\hat{f}$ that approximates the underlying relationship f such that the learned function $\hat{f}$

1. fits the training data points: for each $\mathbf{x}^{(i)} \in \mathcal{D}$, we want $\hat{f}(\mathbf{x}^{(i)}) \approx \mathbf{y}^{(i)}$;
2. generalizes to unseen data points: when we observe the input variables $\mathbf{x}$ of a new data point, we wish $\hat{f}(\mathbf{x}) \approx f(\mathbf{x})$ to be true

where the suitability of $\hat{f}$ in describing the dataset is often measured with the **mean squared loss**:

$$\mathcal{L}_{MSE}(\hat{f}, \mathcal{D}) := \frac{1}{n_d} \sum_{i=1}^{n_d} \left(\hat{f}(\mathbf{x}^{(i)}) - \mathbf{y}^{(i)} \right)^2 \tag{1.1}$$

The MSE loss measures the average deviation of our prediction from the real output, and we would like to minimize this value. When choosing the function $\hat{f}$, instead of considering all possible functions from $\mathbb{R}^{n_{in}}$ to $\mathbb{R}^{n_{out}}$, we generally restrict our attention to a class of functions $\hat{f}(*; \theta)$ that can be parameterized with a parameter vector $\theta \in \mathbb{R}^{n_p}$. This function space, along with the choice of parameterization, is referred to as the **model**.

[1] also known as explanatory variables or features
[2] also known as response variables or labels

The choice of parameterization represents our assumption about the complexity of the underlying function f. The conventional wisdom is that when n_p is small, the model is not specific enough to represent f, and the minimum loss we can achieve with the model is large on both training and test data. On the other hand, when we add parameters to the model, we end up searching for $\hat{f}$ in a richer, more complex class of functions, and we expect the model to fit the training data better.

When the number of parameters n_p exceeds the number of data points n_d, the model is overparameterized, and in general, there are infinitely many functions $\hat{f}$ in the function space that perfectly fit the training data. But not all of these functions generalize well to data points unseen during training.

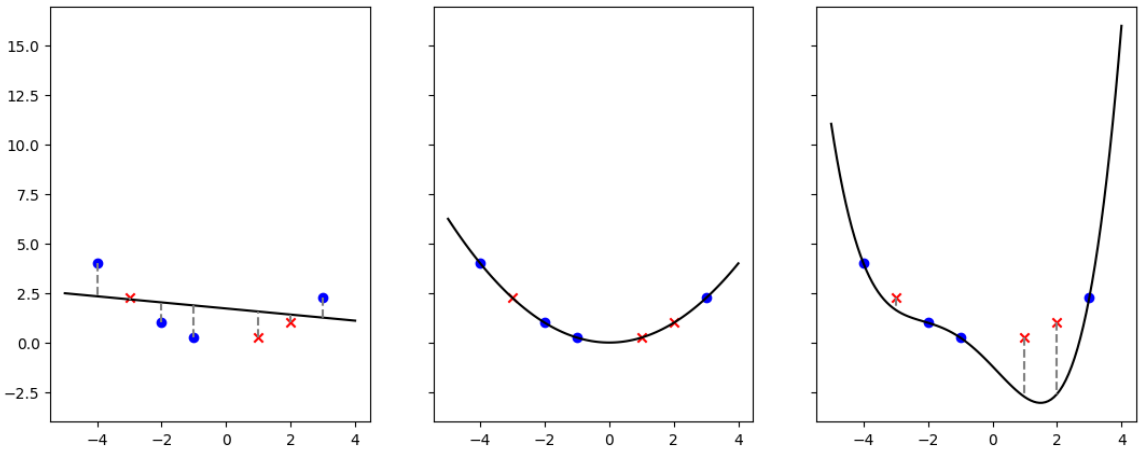

Figure 1.1: Given the training data (blue circle) and test data (red cross), the model without enough parameters (left) exhibits large training and test loss. When parameters are added to the model, it is possible to perfectly fit the training data (middle), but when there are too many parameters to the model (right), it may overfit and generalize poorly to test data.

In classical learning theory, different methods have been proposed for explicit capacity control that restricts the function space for better generalization [11, 1]. However, in the field of machine learning, overparameterized neural networks have been shown to exhibit good generalization abilities without an explicit capacity control on various tasks from computer vision [5, 12] and natural language processing [2, 9].

One explanation is the common use of ℓ_2 regularization in machine learning models. Even though its effect on the parameters during a single step of training is easily explained, the existence of non-linear activation function components (e.g., ReLU) in the model makes it difficult to understand how ℓ_2 regularization affects the solution space. Following previous works [7, 10, 8, 4], this paper examines the geometric properties that ℓ_2 regularization induces on the function space learned by shallow neural networks with a non-linear ReLU component.

1.2 Feedforward Neural Networks

Feedforward neural networks are a class of functions that is commonly used in modern machine learning. In this section, we introduce how these functions are parameterized.

1.2.1 Artificial Neuron

An **artificial neuron** is the basic building block of a neural network. Provided a parameter vector $\theta = (\mathbf{w}, b) \in \mathbb{R}^{n_{in}+1}$ consisting of a weight vector $\mathbf{w} \in \mathbb{R}^{n_{in}}$ and a bias term $b \in \mathbb{R}$, a neuron represents the function $f(\mathbf{x}; \theta) : \mathbb{R}^{n_{in}} \to \mathbb{R}$ defined as

$$f(\mathbf{x}; \theta) := \sigma\left(\mathbf{w} \cdot \mathbf{x} + b\right) \tag{1.2}$$

for some choice of non-linear function $\sigma : \mathbb{R} \to \mathbb{R}$, referred to as the **activation function**.

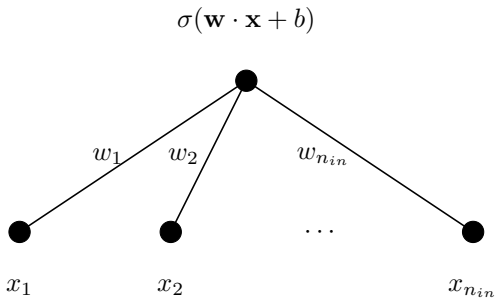

Figure 1.2: Diagram of an artificial neuron.

One of the most commonly used activation function is the **Rectified Linear Unit (ReLU)** defined as follows:

$$[x]_+ := \max(x, 0) \tag{1.3}$$

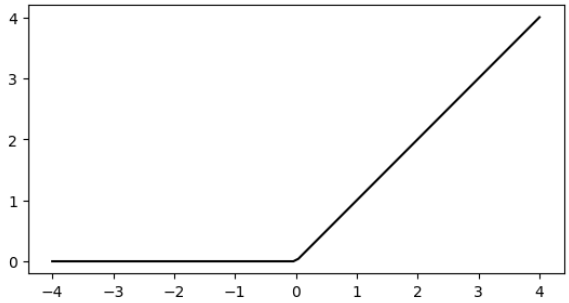

Figure 1.3: Graph of $y = [x]_+$.

When ReLU is chosen as the activation function σ for a neuron, it is customary to say that the neuron is **activated** or **on** when $f(\mathbf{x};\theta) \geq 0$ and **deactivated** or **off** otherwise.

1.2.2 Hidden Layers

In feedforward neural networks, a large number of neurons are used as intermediate variables. In particular, the neurons are arranged in a layered structure such that output values of the neurons of one layer are used as the input of the neurons of the next layer. Any neuron that is used as an intermediate variable is referred to as the **hidden node** and the collection of these hidden nodes are referred to as the **hidden layers** of the network.

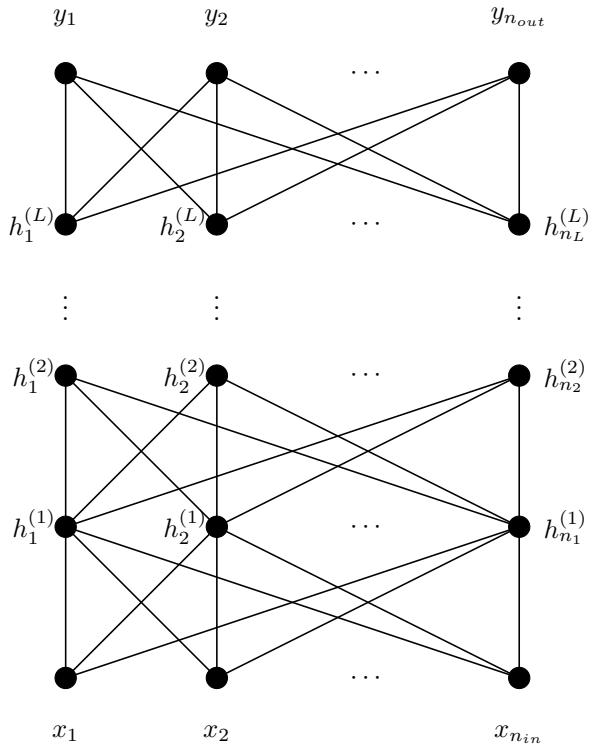

Figure 1.4: Diagram of a deep neural network with L hidden layers.

If we let $\mathbf{h^{(j)}} = \left(h_1^{(j)}, h_2^{(j)}, \cdots, h_{n_j}^{(j)}\right)$ denote the values of n_j nodes of the j-th layer, then each layer of nodes can be understood as the affine transformation of the previous layer of nodes, followed by an element-wise application of a non-linear activation function σ. Then a neural network with L

hidden layers can be mathematically formulated as:

$$\begin{cases} \mathbf{h}^{(1)} = \sigma\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) \\ \mathbf{h}^{(j)} = \sigma\left(\mathbf{W}^{(j)}\mathbf{h}^{(j-1)} + \mathbf{b}^{(j)}\right) \quad \forall j = 2, 3, \cdots, L \\ \mathbf{y} = \mathbf{W}^{(L+1)}\mathbf{h}^{(L)} + \mathbf{b}^{(L)} \end{cases} \tag{1.4}$$

Without the non-linear activations, the entire function will collapse as a single affine transformation. However, by adding an element-wise non-linearity in between each layer, the function class that the model represents becomes very rich. [3] When the network contains $L > 1$ layers, we also say that the network is **deep**, and if $L = 1$, we say the network is **shallow**.

1.3 Problem Setting

1.3.1 Dataset

We consider data with $n_{out} = 1$. That is, we are given a set $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{n_d}$ of data points, where $\mathbf{x}^{(i)} \in \mathbb{R}^{n_{in}}$ and $y^{(i)} \in \mathbb{R}$.

1.3.2 Model

We consider **1-layer ReLU networks** of size (or width) n_h defined as:

$$f(\mathbf{x}; \theta) := \mathbf{W}^{(2)} \cdot \left[\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right]_+ + \mathbf{a} \cdot \mathbf{x} + b^{(2)} \tag{1.5}$$

$$:= \left(\sum_{i=1}^{n_h} W_i^{(2)} \left[\mathbf{W}_i^{(1)} \cdot \mathbf{x} + b_i^{(1)}\right]_+\right) + \mathbf{a} \cdot \mathbf{x} + b^{(2)} \tag{1.6}$$

where the parameter vector θ refers to all entries in

1. $\mathbf{W}^{(1)} \in \mathbb{R}^{n_h \times n_{in}}$: the weight matrix for the hidden layer;
2. $\mathbf{b}^{(1)} \in \mathbb{R}^{n_h}$: the bias terms for the hidden layer;
3. $\mathbf{W}^{(2)} \in \mathbb{R}^{n_h}$: the weight vector for the output;
4. $\mathbf{a} \in \mathbb{R}^{n_{in}}$ and $b \in \mathbb{R}$: the bias terms for the output

[3] [3] shows that the set of neural networks of $L = 1$ with a sigmoidal activation is dense in the set of all continuous functions on $[0, 1]^{n_{in}}$.

Note that the vector **a** represents a **residual connection** from the input to the output that bypasses the hidden layer. A residual connection defined in this particular way is not common in practice, but it has been included for the sake of cleaner mathematical analysis, in alignment with previous works [10, 8, 4].

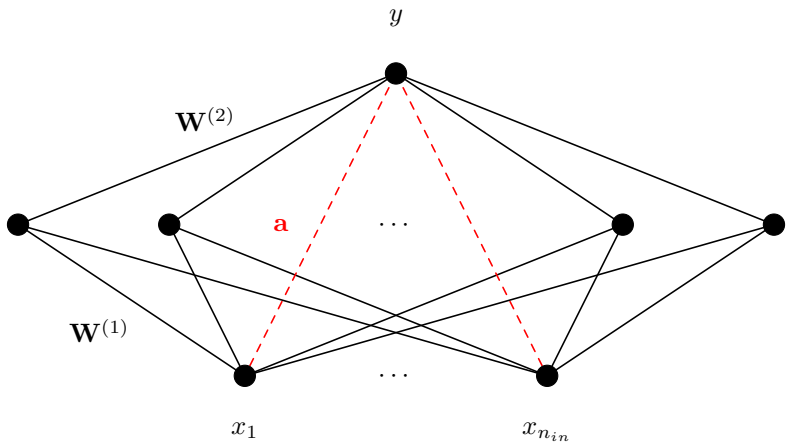

Figure 1.5: Diagram of a 1-layer ReLU network.

1.3.3 Weight Cost

We denote the set of all 1-layer ReLU networks that perfectly fit $\mathcal{D}$ as $ReLU(\mathcal{D})$, with no restriction on the width n_h of the network. That is,

$$ReLU(\mathcal{D}) := \{f(*;\theta) \mid f \text{ is a 1-layer ReLU network and } f(\mathbf{x};\theta) = y \quad \forall (\mathbf{x}, y) \in \mathcal{D}\} \tag{1.7}$$

Since we have no upper limit on the width n_h of the network, generally $ReLU(\mathcal{D})$ contains an infinite number of overparameterized networks that interpolate the dataset. However, many of these functions are expected to overfit the training data and generalize poorly to test data. A widely used technique that mitigates the overfitting problem is the ℓ_2 regularization, which seeks to minimize the ℓ_2 **weight cost** $C_2(\theta)$ out of all interpolants. In particular, we define

$$RidgelessReLU(\mathcal{D}) := \underset{f(\mathbf{x};\theta) \in ReLU(\mathcal{D})}{\arg\min} \; C_2(\theta) \tag{1.8}$$

where the weight cost is defined as

$$C_2(\theta) := \sum_{i=1}^{n_h} \left(\left\| \mathbf{W}_i^{(1)} \right\|_2^2 + \left| \mathbf{W}_i^{(2)} \right|^2 \right) \tag{1.9}$$

Following the conventional definition, the bias terms $\mathbf{b}^{(1)}$ and $b^{(2)}$ have been omitted from the weight cost. Additionally, the weights $\mathbf{a}$ for the residual affine layer has been omitted, following the analysis of previous works [10, 8, 4].

1.4 Related Work

In [7] and [10], it was shown that controlling the ℓ_2 weight cost of a 1-layer ReLU network is equivalent to controlling the ℓ_1 weight of the output layer, when the weights of the hidden layer are restricted to unit norm. We present the formal statement without proof.

Proposition 1.4.1 (Theorem 1 of [7], Lemma A.1 of [10])**.**

$$RidgelessReLU(\mathcal{D}) = \underset{\substack{f(\mathbf{x};\theta)\in ReLU(\mathcal{D}) \\ \left\|\mathbf{W}_i^{(1)}\right\|_2=1 \quad \forall i}}{\arg\min} C_1(\theta) \tag{1.10}$$

where $C_1(\theta)$ is defined as

$$C_1(\theta) := \sum_{i=1}^{n_h} \left|\mathbf{W}_i^{(2)}\right| \tag{1.11}$$

In the case of $n_{in} = 1$, [10] shows that minimizing the weight cost $C_1(\theta)$ is equivalent to minimizing the total variation norm of the function. In particular, the function $f_{\mathcal{D}}$ that linearly interpolates the dataset satisfies $f_{\mathcal{D}} \in RidgelessReLU(\mathcal{D})$. Furthermore, [4] completely describes $RidgelessReLU(\mathcal{D})$ as the set of functions that perform nearest neighbor curvature extrapolation — any function $f \in RidgelessReLU(\mathcal{D})$ must coincide with $f_{\mathcal{D}}$ on segments where the convexity/concavity is ambiguous, but otherwise can be convex/concave within the boundaries set by the neighboring segments.

In the general case where $n_{in} > 1$, the analysis is not so simple. [8] finds that minimizing $C_1(\theta)$ is equivalent to minimizing the semi-norm $\|f\|_{\mathcal{R}}$ on $ReLU(\mathcal{D})$ defined as

$$\|f\|_{\mathcal{R}} := \sup\left\{ -\frac{1}{2(2\pi)^{n_{in}-1}} \left\langle f, (-\Delta)^{(n_{in}+1)/2}\mathcal{R}^*\{\psi\}\right\rangle \;\middle|\; \psi \in \mathcal{S}(\mathbb{S}^{n_{in}-1}\times\mathbb{R}), \psi \text{ even}, \|\psi\|_\infty \le 1\right\}$$

where $\mathbb{S}^{n_{in}-1}$ is the unit sphere and $\mathcal{S}(\mathbb{S}^{n_{in}-1}\times\mathbb{R})$ is the set of Schwartz functions on $\mathbb{S}^{n_{in}-1}\times\mathbb{R}$ and $\mathcal{R}^*$ represents the dual Radon transform. But the question of "What are the examples or properties of the functions that minimize this semi-norm?" is still not well understood.

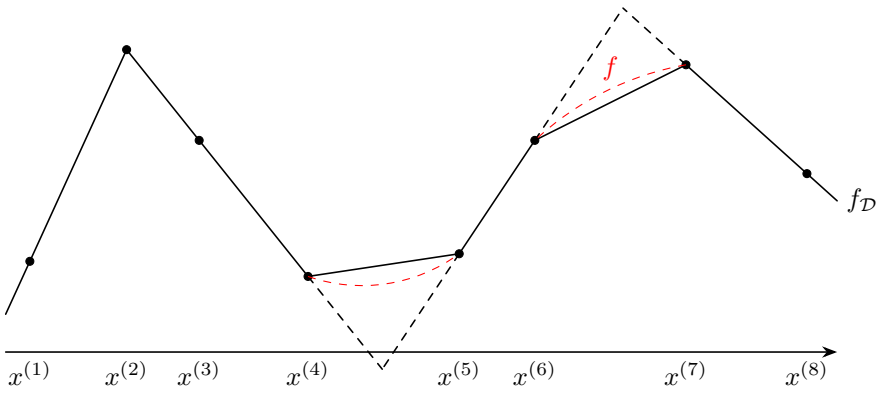

Figure 1.6: When $n_{in} = 1$, any $f \in \mathit{RidgelessReLU}(\mathcal{D})$ (red dashed line) coincides with the linear interpolant $f_{\mathcal{D}}$ (black solid line) on segments where curvature is ambiguous and is concave/convex within the dashed boundary.

1.5 Summary of Results

In this paper, we consider two different classes of datasets where $n_{in} = 2$. In Section 3.1, we consider a dataset $\mathcal{D}$ where the points $\mathbf{x}^{(i)}$ form the vertices of a regular polygon. For this class of dataset, we propose a heuristic to directly compute an interpolant $f \in \mathit{ReLU}(\mathcal{D})$ that we believe to be near-optimal. In Section 3.2, we consider $\mathcal{D}$ to be symmetric with respect to a line ℓ. We show that if the dataset is effectively 1-dimensional, then any optimal $f \in \mathit{RidgelessReLU}(\mathcal{D})$ should also be 1-dimensional.

Chapter 2

Preliminaries

In this chapter, we present some preliminary definitions and observations that will be relevant for the main discussion.

2.1 Geometry of ReLU Networks

2.1.1 Decomposing into ReLU Gates

Say we have a 1-layer ReLU network

$$f(\mathbf{x};\theta) = \left(\sum_{i=1}^{n_h} W_i^{(2)} \left[\mathbf{W}_i^{(1)} \cdot \mathbf{x} + b_i^{(1)}\right]_+\right) + \mathbf{a} \cdot \mathbf{x} + b^{(2)}$$

When we ignore the residual connection $\mathbf{a} \cdot \mathbf{x} + b^{(2)}$, then f is the sum of n_h individual components

$$f_i(\mathbf{x};\theta) = W_i^{(2)} \left[\mathbf{W}_i^{(1)} \cdot \mathbf{x} + b_i^{(1)}\right]_+$$

that each correspond to one hidden node. We refer to this function component as a **ReLU gate**.

Example 2.1.1. *Consider a 1-layer ReLU network*

$$f(\mathbf{x};\theta) = 2[(1,2) \cdot \mathbf{x} - 1]_+ + 0.3 \cdot [(-1,-3) \cdot \mathbf{x} - 4]_+ + 0.5 \cdot [(-2,1) \cdot \mathbf{x} - 3]_+ \tag{2.1}$$

with three ReLU gates. Figure 2.1 shows the graph of f alongside the graph for its three ReLU gates. Notice that each ReLU gate is a continuous piecewise linear function with a single hyperplane

[1] *where the ReLU changes behavior.*

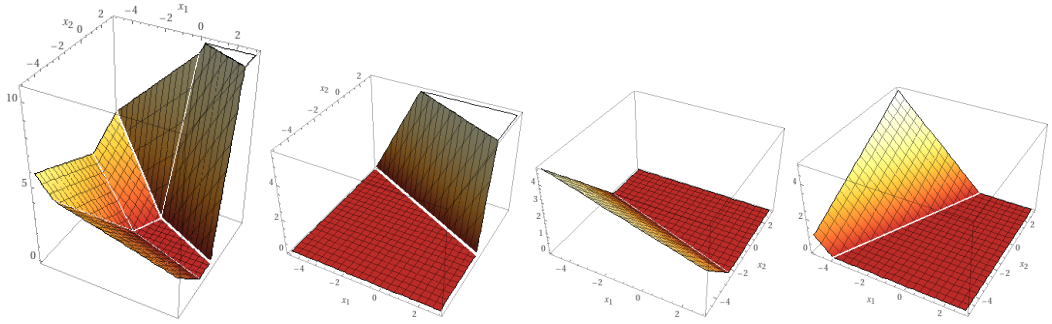

Figure 2.1: The graph of $f(\mathbf{x};\theta) = 2[(1,2)\cdot\mathbf{x}-1]_+ + 0.3\cdot[(-1,-3)\cdot\mathbf{x}-4]_+ + 0.5\cdot[(-2,1)\cdot\mathbf{x}-3]_+$ and its three ReLU gates.

From Example 2.1.1, we see that a ReLU gate is characterized by the location of its defining hyperplane, the direction from the hyperplane where it is activated, and its "slope" when activated.

2.1.2 Normalizing a ReLU Gate

Given an arbitrary ReLU gate

$$f_i(\mathbf{x};\theta) = W_i^{(2)}\left[\mathbf{W}_i^{(1)}\cdot\mathbf{x} + b_i^{(1)}\right]_+$$

we can define its **normalization**

$$\tilde{f}_i(\mathbf{x};\tilde{\theta}) = \widetilde{W}_i^{(2)}\left[\widetilde{\mathbf{W}}_i^{(1)}\cdot\mathbf{x} + \tilde{b}_i^{(1)}\right]_+ \tag{2.2}$$

where

$$\begin{cases} \widetilde{W}_i^{(2)} = \left\|\mathbf{W}_i^{(1)}\right\| W_i^{(2)} \\ \widetilde{\mathbf{W}}_i^{(1)} = \dfrac{\mathbf{w}_i^{(1)}}{\left\|\mathbf{w}_i^{(1)}\right\|} \\ \tilde{b}_i^{(1)} = \dfrac{b_i^{(1)}}{\left\|\mathbf{w}_i^{(1)}\right\|} \end{cases} \tag{2.3}$$

such that the decision boundary of the ReLU gate is maintained, but the hyperplane is now defined by a unit normal vector $\widetilde{\mathbf{W}}_i^{(1)}$. Lemma 2.1.3 will show that the two ReLU gates are actually equivalent, by using the fact that ReLU is a 1-homogeneous function.

[1] In the case of our 2-dimensional input, this is a line, but for generalizability to higher dimensional input, we refer to the decision boundary as a hyperplane.

Lemma 2.1.2. *For any $c \geq 0$ and for any $x \in \mathbb{R}$, we have $[cx]_+ = c \cdot [x]_+$*

Proof. When $c = 0$, both sides of the equation are 0, and the conclusion is obvious. If $c > 0$, then $sgn(cx) = sgn(x)$, and therefore

$$[cx]_+ = \begin{cases} cx & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Notice that the function on the right hand side is precisely $c \cdot [x]_+$. □

Lemma 2.1.3. *For any ReLU gate $f_i(\mathbf{x}; \theta) = W_i^{(2)} \left[\mathbf{W}_i^{(1)} \cdot \mathbf{x} + b_i^{(1)}\right]_+$, its normalized version $\tilde{f}_i(\mathbf{x}; \tilde{\theta})$ defined as in (2.2) and (2.3) satisfies $f_i(\mathbf{x}; \theta) = \tilde{f}_i(\mathbf{x}; \tilde{\theta})$ for any $\mathbf{x} \in \mathbb{R}^2$.*

Proof. By Lemma 2.1.2, for any $\mathbf{x} \in \mathbb{R}^2$, we have

$$\begin{aligned} \tilde{f}_i(\mathbf{x}; \tilde{\theta}) &= \widetilde{W}_i^{(2)} \left[\widetilde{\mathbf{W}}_i^{(1)} \cdot \mathbf{x} + \tilde{b}_i^{(1)}\right]_+ \\ &= \widetilde{W}_i^{(2)} \left[\frac{1}{\left\|\mathbf{W}_i^{(1)}\right\|} \left(\mathbf{W}_i^{(1)} \cdot \mathbf{x} + b_i^{(1)}\right)\right]_+ \\ &= \frac{\widetilde{W}_i^{(2)}}{\left\|\mathbf{W}_i^{(1)}\right\|} \left[\mathbf{W}_i^{(1)} \cdot \mathbf{x} + b_i^{(1)}\right]_+ \\ &= W_i^{(2)} \left[\mathbf{W}_i^{(1)} \cdot \mathbf{x} + b_i^{(1)}\right]_+ \\ &= f_i(\mathbf{x}; \theta) \end{aligned}$$

□

Corollary 2.1.4. *For any 1-layer ReLU network $f(\mathbf{x}; \theta)$, there exists another network $\tilde{f}(\mathbf{x}; \tilde{\theta})$ whose ReLU gates are all normalized and $f(\mathbf{x}; \theta) = \tilde{f}(\mathbf{x}; \tilde{\theta})$.*

By the result of Corollary 2.1.4, we can define an equivalence relation on $ReLU(\mathcal{D})$ based on the normalized version of each network. After factoring out $ReLU(\mathcal{D})$ by the equivalence classes, we can assume that each ReLU gate of a ReLU network is defined by a unit normal vector $\mathbf{W}_i^{(1)}$ without loss of generality. In view of Proposition 1.4.1, we can now focus our attention on minimizing $C_1(\theta) = \sum_{i=1}^{n_h} \left|W_i^{(2)}\right|$.

2.1.3 Signed Distance From the Hyperplane

As observed earlier, a ReLU gate $f_i(\mathbf{x}; \theta) = W_i^{(2)} \left[\mathbf{W}_i^{(1)} \cdot \mathbf{x} + b_i^{(1)}\right]_+$ is characterized by the decision boundary $\mathcal{H}_i := \left\{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{W}_i^{(1)} \cdot \mathbf{x} + b_i^{(1)} = 0\right\}$. Once $\mathbf{W}_i^{(1)}$ has been normalized to have unit

norm, then $\left|\mathbf{W}_i^{(1)} \cdot \mathbf{x} + b_i^{(1)}\right|$ computes the Euclidean distance from an arbitrary point $\mathbf{x} \in \mathbb{R}^2$ to the hyperplane $\mathcal{H}_i$. But simply computing the Euclidean distance is not sufficient for our analysis. We additionally need to know which side of $\mathcal{H}_i$ that f_i is on. So we consider the **signed distance**

$$d(\mathbf{x}, \mathcal{H}_i) := \mathbf{W}_i^{(1)} \cdot \mathbf{x} + b_i^{(1)} \tag{2.4}$$

which will be positive on the side of $\mathcal{H}_i$ that the normal vector is pointing towards, and negative on the other side.

The value of a single ReLU gate $f_i(\mathbf{x}; \theta)$ on a given point $\mathbf{x} \in \mathbb{R}^2$ can now be computed as $f_i(\mathbf{x}; \theta) = W_i^{(2)}[d(\mathbf{x}, \mathcal{H}_i)]_+$, the product of the weight $W_i^{(2)}$ of the ReLU gate and the signed distance between $\mathbf{x}$ and the hyperplane $\mathcal{H}_i$. On the other hand, if we know that we want to assign a value of y to the given point $\mathbf{x}$, we should set the weight of the ReLU gate to be $W_i^{(2)} = \frac{y}{[d(\mathbf{x}, \mathcal{H}_i)]_+}$.

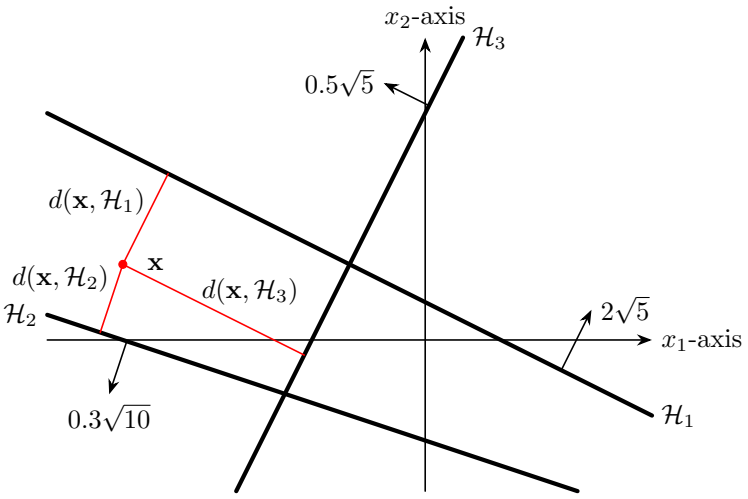

Figure 2.2: Visualization of the ReLU network in (2.1) using normalized ReLU gates.

Example 2.1.5 (Example 2.1.1 revisited)**.** *After normalization, each of ReLU gates in* (2.1) *can be rewritten as*

$$\begin{aligned}
f_1(\mathbf{x}; \theta) &= 2\sqrt{5}\left[\left(\frac{1}{\sqrt{5}}, \frac{2}{\sqrt{5}}\right) \cdot \mathbf{x} - \frac{1}{\sqrt{5}}\right]_+ \\
f_2(\mathbf{x}; \theta) &= 0.3\sqrt{10} \cdot \left[\left(-\frac{1}{\sqrt{10}}, -\frac{3}{\sqrt{10}}\right) \cdot \mathbf{x} - \frac{4}{\sqrt{10}}\right]_+ \\
f_3(\mathbf{x}; \theta) &= 0.5\sqrt{5} \cdot \left[\left(-\frac{2}{\sqrt{5}}, \frac{1}{\sqrt{5}}\right) \cdot \mathbf{x} - \frac{3}{\sqrt{5}}\right]_+
\end{aligned}$$

Figure 2.1.3 shows an alternate visualization of the ReLU network, where each ReLU gate is rep-

resented with its hyperplane (with orientation) and weight. Given a data point $\mathbf{x} = (-4, 1)$*, its function value can be computed from its signed distance to the hyperplanes:*

$$\begin{aligned} f(\mathbf{x};\theta) &= 2\sqrt{5}[d(\mathbf{x},\mathcal{H}_1)]_+ + 0.3\sqrt{10}[d(\mathbf{x},\mathcal{H}_2)]_+ + 0.5\sqrt{5}[d(\mathbf{x},\mathcal{H}_3)]_+ \\ &= 2\sqrt{5}\cdot 0 + 0.3\sqrt{10}\cdot 0 + 0.5\sqrt{5}\cdot\frac{6}{\sqrt{5}} = 3 \end{aligned}$$

In view of the observations made in this section, we now denote a 1-layer ReLU network as

$$f(\mathbf{x};\theta) = \left(\sum_{i=1}^{n_h} W_i^{(2)}\,[d(\mathbf{x},\mathcal{H}_i)]_+\right) + \mathbf{a}\cdot\mathbf{x} + b^{(2)} \tag{2.5}$$

where the $\mathcal{H}_i$ are the hyperplanes that define each of the ReLU gates of the network.

2.1.4 Connection to Continuous Piecewise Linear Functions

Every ReLU gate is continuous and piecewise linear. Hence, a 1-layer ReLU network, a linear combination of these components, is also continuous and linear on each cell of the hyperplane arrangement. For the case of $n_{in} = 1$, the reverse direction is also true — any continuous piecewise linear function can be represented with a 1-layer ReLU network. However, for $n_{in} \geq 2$, that is not necessarily true.

Theorem 2.1.6 (Theorem 4.1 of [6])**.** *If* $n_{in} \geq 2$*, then any continuous piecewise linear function with compact support on* $\mathbb{R}^{n_{in}}$ *cannot be represented by a 1-layer ReLU network.* [2]

The theorem above illustrates one of the reasons why it is difficult to analyze the case for $n_{in} \geq 2$, compared to the case $n_{in} = 1$. Since not all continuous piecewise linear functions can be attained with a 1-layer ReLU network, we cannot directly try to linearly interpolate the data points. Instead, a geometric understanding of ReLU networks is necessary to clearly understand the shape of the function space $ReLU(\mathcal{D})$.

2.2 Regular Polygons

In Section 3.1, we will consider dataset $\mathcal{D}$ that consists of data points $\mathbf{x}$ which form the vertices of a regular polygon. To prepare for the discussion, we review relevant definitions and make preliminary observations.

[2] The definition of a 1-layer ReLU network in [6] does not include a residual connection, but the proof can be easily adapted by rewriting the residual connection as the sum of two ReLU gates that share the same hyperplane and face opposite directions.

Definition 2.2.1. *A set of points* $\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(n)}\}$ *form the vertices of a* ***regular n-gon*** *with side length* $r > 0$ *if for each* $1 \leq i \leq n$ [3]

1. $\overline{\mathbf{x}^{(i)}\mathbf{x}^{(i+1)}} = r$
2. $\angle \mathbf{x}^{(i-1)}\mathbf{x}^{(i)}\mathbf{x}^{(i+1)} = \pi - \gamma_n$

where $\gamma_n = \frac{2\pi}{n}$ *denotes the common external angle.*

The following fact is well known and will be presented without proof.

Proposition 2.2.2. *A regular n-gon can be inscribed in a circle.*

One property of circles is that any chord splits a circle into two continuous arcs. Therefore, given a ReLU gate, its hyperplane will separate a circle into two continuous pieces.

Corollary 2.2.3. *Let* $\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(n)}\}$ *form the vertices of a regular polygon. Then given a ReLU gate, the set of points that activate the ReLU gate is either* $\emptyset$ *or* $\{\mathbf{x}^{(i)}, \cdots, \mathbf{x}^{(i+k)}\}$ *for some* $1 \leq i \leq n$ *and* $0 \leq k \leq n-1$*. On the other hand, given a set of the form* $\{\mathbf{x}^{(i)}, \cdots, \mathbf{x}^{(i+k)}\}$*, we can find a ReLU gate that is activated only on the prescribed set of points.*

2.3 Reflection and Symmetry

In Section 3.2, we will consider dataset $\mathcal{D}$ that is symmetric with respect to a line ℓ. To prepare for the discussion, we review relevant definitions and make preliminary observations.

2.3.1 Reflection of Points

Given a point $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and a line $\ell = \left\{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{a} \cdot \mathbf{x} + b = 0\right\}$, the **reflection** of $\mathbf{x}$ across ℓ is defined as the unique point $\tilde{\mathbf{x}} \in \mathbb{R}^2$ such that $d(\mathbf{x}, \ell) = -d(\tilde{\mathbf{x}}, \ell)$ and the segment between $\mathbf{x}$ and $\tilde{\mathbf{x}}$ is perpendicular to ℓ. In other words, it is the point with the same distance away from ℓ but in the "opposite direction." The formula for the reflection operation $R_\ell : \mathbf{x} \mapsto \tilde{\mathbf{x}}$ can be explicitly given as

$$R_\ell(\mathbf{x}) := \mathbf{x} - 2d(\mathbf{x}, \ell)\mathbf{a} = \mathbf{x} - \frac{2(\mathbf{a} \cdot \mathbf{x} + b)}{\mathbf{a} \cdot \mathbf{a}}\mathbf{a} \tag{2.6}$$

In particular, if $b = 0$ (i.e., ℓ goes through the origin), then the reflection operation R_ℓ is orthogonal. For any general $b \neq 0$, the reflection across ℓ can be decomposed into a pair of translations and an orthogonal reflection.

[3] To avoid having to consider edge cases at the boundary, we understand the indices i to be an element of $\mathbb{Z}_n$. For example, $\mathbf{x}^{(0)} := \mathbf{x}^{(n)}$ and $\mathbf{x}^{(n+1)} := \mathbf{x}^{(1)}$.

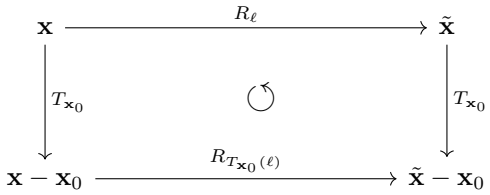

Figure 2.3: Diagram representing Proposition 2.3.1.

Proposition 2.3.1. *Let $\ell = \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{a}\cdot\mathbf{x} + b = 0\}$ be a line and choose any point $\mathbf{x}_0 \in \ell$. Then $R_\ell = T_{\mathbf{x}_0}^{-1} R_{T_{\mathbf{x}_0}(\ell)} T_{\mathbf{x}_0}$ where $T_{\mathbf{x}_0} : \mathbf{x} \mapsto \mathbf{x} - \mathbf{x}_0$ denotes the translation that maps $\mathbf{x}_0$ to the origin.*

Proof. First notice that $T_{\mathbf{x}_0}(\ell) = \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{a}\cdot\mathbf{x} + \mathbf{a}\cdot\mathbf{x}_0 + b = 0\}$. Therefore, by the formula (2.6),

$$\begin{aligned}
(T_{\mathbf{x}_0}^{-1} R_{T_{\mathbf{x}_0}(\ell)} T_{\mathbf{x}_0})(\mathbf{x}) &= (T_{\mathbf{x}_0}^{-1} R_{T_{\mathbf{x}_0}(\ell)})(\mathbf{x} - \mathbf{x}_0) \\
&= T_{\mathbf{x}_0}^{-1}\left((\mathbf{x} - \mathbf{x}_0) - \frac{2(\mathbf{a}\cdot(\mathbf{x}-\mathbf{x}_0) + \mathbf{a}\cdot\mathbf{x}_0 + b)}{\mathbf{a}\cdot\mathbf{a}}\mathbf{a}\right) \\
&= T_{\mathbf{x}_0}^{-1}\left(R_\ell(\mathbf{x}) - \mathbf{x}_0\right) \\
&= R_\ell(\mathbf{x})
\end{aligned}$$

□

When $\tilde{\mathbf{x}}$ is a reflection of $\mathbf{x}$ across ℓ, we say that $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are **symmetric** with respect to ℓ.

2.3.2 Reflection of Hyperplanes

Given a hyperplane $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{w}\cdot\mathbf{x} + b_{\mathcal{H}} = 0\}$ and a line $\ell = \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{a}\cdot\mathbf{x} + b_\ell = 0\}$, let us define $\mathbf{x}_0$ to be the intersection of $\mathcal{H}$ and ℓ. If $\mathcal{H}$ and ℓ are parallel, define $\mathbf{x}_0$ to be any point on ℓ instead. Now let

$$\widetilde{\mathbf{w}} = R_{T_{\mathbf{x}_0}(\ell)}(\mathbf{w}) \tag{2.7}$$

denote the reflection across the line ℓ after it has been translated such that $\mathbf{x}_0$ is the origin. The **reflection** of $\mathcal{H}$ across ℓ is defined as

$$\widetilde{\mathcal{H}} := \{\mathbf{x} \in \mathbb{R}^2 \mid \widetilde{\mathbf{w}}\cdot(\mathbf{x} - \mathbf{x}_0) = 0\} \tag{2.8}$$

Similarly to points, if $\widetilde{\mathcal{H}}$ is the reflection of $\mathcal{H}$ across ℓ, we say that $\mathcal{H}$ and $\widetilde{\mathcal{H}}$ are **symmetric** with respect to ℓ. We now present a trivial observation.

Proposition 2.3.2. *If $\mathbf{x}, \tilde{\mathbf{x}}$ and $\mathcal{H}, \widetilde{\mathcal{H}}$ are respectively symmetric with respect to ℓ, then $d(\mathbf{x}, \mathcal{H}) = d(\tilde{\mathbf{x}}, \widetilde{\mathcal{H}})$.*

Proof. First, by Proposition 2.3.1, we have

$$\tilde{\mathbf{x}} - \mathbf{x}_0 = T_{\mathbf{x}_0}(R_\ell(\mathbf{x}) - \mathbf{x}_0) = R_{T_{\mathbf{x}_0}(\ell)} T_{\mathbf{x}_0}(\mathbf{x}) = R_{T_{\mathbf{x}_0}(\ell)}(\mathbf{x} - \mathbf{x}_0)$$

In particular, $R_{T_{\mathbf{x}_0}(\ell)} : \mathbf{w} \mapsto \widetilde{\mathbf{w}}$ is orthogonal since $T_{\mathbf{x}_0}(\ell)$ passes through the origin.

$$d(\tilde{\mathbf{x}}, \widetilde{\mathcal{H}}) = \left\langle R_{T_{\mathbf{x}_0}(\ell)}(\mathbf{w}), \tilde{\mathbf{x}} - \mathbf{x}_0 \right\rangle = \left\langle \mathbf{w}, R^{\intercal}_{T_{\mathbf{x}_0}(\ell)} R_{T_{\mathbf{x}_0}(\ell)}(\mathbf{x} - \mathbf{x}_0) \right\rangle = \langle \mathbf{w}, \mathbf{x} - \mathbf{x}_0 \rangle = d(\mathbf{x}, \mathcal{H})$$

□

2.3.3 Reflection of ReLU Gates

Given a ReLU gate $f(\mathbf{x}; \theta) = W^{(2)} \left[d(\mathbf{x}, \mathcal{H}) \right]_+$ and a line ℓ, we can define the **reflection** of f across ℓ as

$$\tilde{f}(\mathbf{x}; \tilde{\theta}) := W^{(2)} \left[d(\mathbf{x}, \widetilde{\mathcal{H}}) \right]_+ \tag{2.9}$$

where we use the same weight $W^{(2)}$ but the defining hyperplane $\mathcal{H}$ has been reflected across ℓ.

Chapter 3

Main Results

3.1 Regular Polygons

In this section, we assume that the data points $\mathbf{x}^{(i)}$ of the dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{n_d}$ form the vertices of a regular n_d-gon where $n_d \geq 4$. [1]

Definition 3.1.1. *A dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{n_d}$ where $n_d \geq 4$ is **sampled from a regular polygon with side length r** if $\{\mathbf{x}^{(i)}\}_{i=1}^{n_d}$ forms the vertices of a regular n_d-gon with side length r.*

When $\mathcal{D}$ is sampled from a regular polygon, the set of points that can be linearly separable from the remaining points can be precisely described as the set of consecutive points on the polygon. Hence, given a ReLU gate f_i, the points that are activated are always consecutive; on the other hand, given a set of consecutive points, it is possible to propose a ReLU gate f_i that is only activated only on the selected points.

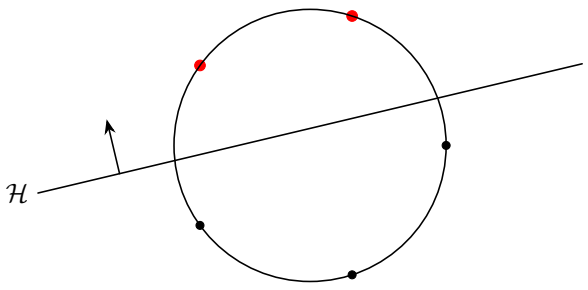

Figure 3.1: When $\mathcal{D}$ is sampled from a regular polygon, any $\mathcal{H}$ separates a set of consecutive points from the remaining points.

[1] The assumption that $n_d \geq 4$ is critical for many of the arguments.

Based on this understanding, we devise a heuristic that will construct a ReLU network $f \in ReLU(\mathcal{D})$ that interpolates the dataset. The main idea is to

1. select a point $\mathbf{x}^{(i)}$ or a pair of consecutive points $\mathbf{x}^{(i)}, \mathbf{x}^{(i+1)}$;
2. fix a value y_i or a pair of values y_i, y_{i+1} that we would like to assign to each point; [2]
3. and find a ReLU gate that assigns the prescribed values with minimal weight cost.

We will refer to this process as **relaxing** the selected point $(\mathbf{x}^{(i)}, y_i)$ or the pair of points $(\mathbf{x}^{(i)}, y_i)$, $(\mathbf{x}^{(i+1)}, y_{i+1})$. We propose using this process as a subroutine to interpolate the full dataset.

3.1.1 Relaxing One Point

As a warm up, assume we wish to relax a single point $(\mathbf{x}^{(i)}, y_i)$; that is, we want to assign a value y_i to the point $\mathbf{x}^{(i)}$ but 0 to all other points $\mathbf{x}^{(j)}$ where $j \neq i$. Recall from Section 2.1.3 that the weight cost of a ReLU gate is inversely proportional to the distance $d(\mathbf{x}^{(i)}, \mathcal{H})$. Therefore, to minimize the weight cost, the ReLU gate has to be as far way from $\mathbf{x}^{(i)}$ as possible, while also maintaining the fact that it has to be deactivated at all other points. Intuitively, the hyperplane $\mathcal{H}_i$ that passes through $\mathbf{x}^{(i-1)}$ and $\mathbf{x}^{(i+1)}$ achieves this goal. Proposition 3.1.2 verifies this intuition for an arbitrary choice of a point and a value.

Proposition 3.1.2. *Let $\mathcal{D}$ be sampled from a regular polygon with side length r. To relax $(\mathbf{x}^{(i)}, y_i)$ with a single ReLU gate, we need at least a weight cost of $C_1(\theta_i) = \frac{|y^{(i)}|}{\sin\frac{\gamma_{n_d}}{2} \cdot r}$, and it is uniquely achieved by*

$$f_i(\mathbf{x}; \theta_i) = \frac{y_i}{\sin\frac{\gamma_{n_d}}{2} \cdot r} \left[d(\mathbf{x}, \mathcal{H}_i)\right]_+$$

where $\mathcal{H}_i$ is the hyperplane that goes through $\mathbf{x}^{(i-1)}$ and $\mathbf{x}^{(i+1)}$ such that $d(\mathbf{x}^{(i)}, \mathcal{H}_i) > 0$.

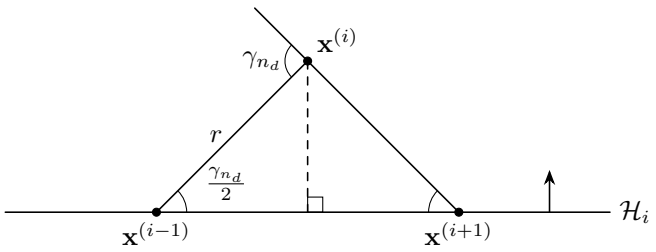

Figure 3.2: Diagram for computing the optimal weight cost in Proposition 3.1.2.

[2] Since multiple ReLU gates may be used to fit a single data point, y_i does not necessarily equal $y^{(i)}$.

Proof. First let us compute the weight cost of f_i. Notice that

$$\angle \mathbf{x}^{(i)}\mathbf{x}^{(i-1)}\mathbf{x}^{(i+1)} = \angle \mathbf{x}^{(i)}\mathbf{x}^{(i+1)}\mathbf{x}^{(i-1)} = \frac{\gamma_{n_d}}{2}$$

Then the distance between $\mathbf{x}^{(i)}$ and $\mathcal{H}_i$ can be explicitly calculated as

$$d(\mathbf{x}^{(i)}, \mathcal{H}_i) = \sin \frac{\gamma_{n_d}}{2} \cdot r \tag{3.1}$$

Now let $\mathcal{H}$ be any other hyperplane that linearly separates $\mathbf{x}^{(i)}$ from the rest of the points; that is, $d(\mathbf{x}^{(i)}, \mathcal{H}) > 0$ and $d(\mathbf{x}^{(i-1)}, \mathcal{H}), d(\mathbf{x}^{(i+1)}, \mathcal{H}) \leq 0$. It suffices to show that $d(\mathbf{x}^{(i)}, \mathcal{H}) < d(\mathbf{x}^{(i)}, \mathcal{H}_i)$.

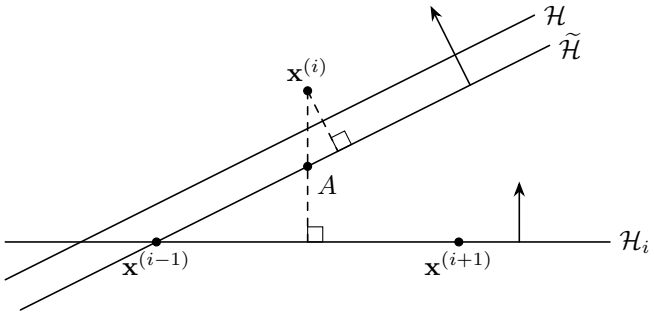

Figure 3.3: Diagram of the proof for Proposition 3.1.2.

Case 1 First consider the case where $\mathcal{H}$ passes through precisely one of $\mathbf{x}^{(i-1)}$ and $\mathbf{x}^{(i+1)}$. Without loss of generality, assume $d(\mathbf{x}^{(i-1)}, \mathcal{H}) = 0$, $d(\mathbf{x}^{(i+1)}, \mathcal{H}) < 0$. Notice that the line segment between $\mathbf{x}^{(i)}$ and its projection on $\mathcal{H}_i$ intersects with $\mathcal{H}$. Let A denote this point. Then

$$d(\mathbf{x}, \mathcal{H}_i) > d(\mathbf{x}, A) > d(\mathbf{x}, \mathcal{H})$$

Case 2 Next consider the case where $\mathcal{H}$ does not pass through $\mathbf{x}^{(i-1)}$ nor $\mathbf{x}^{(i+1)}$. Notice that we can translate $\mathcal{H}$ away from $\mathbf{x}^{(i)}$ until it touches one of $\mathbf{x}^{(i-1)}$ or $\mathbf{x}^{(i+1)}$. Let $\widetilde{\mathcal{H}}$ be the translated hyperplane. We apply the analysis of Case 1 above to get

$$d(\mathbf{x}, \mathcal{H}_i) > d(\mathbf{x}, \widetilde{\mathcal{H}}) > d(\mathbf{x}, \mathcal{H})$$

□

3.1.2 Relaxing Two Points

Now assume we wish to relax a pair of points $(\mathbf{x}^{(i)}, y_i), (\mathbf{x}^{(i+1)}, y_{i+1})$ simultaneously with a single ReLU gate. This immediately adds a restriction to the values we can assign — we require $sgn(y_i) = sgn(y_{i+1})$. Also, since we need to relax the two points simultaneously, we additionally require the hyperplane $\mathcal{H}$ to satisfy $d(\mathbf{x}^{(i)}, \mathcal{H}) : d(\mathbf{x}^{(i+1)}, \mathcal{H}) = y_i : y_{i+1}$. Fortunately, Lemma 3.1.3 guarantees that as long as $sgn(y_i) = sgn(y_{i+1})$, we can find a single ReLU gate that will relax $(\mathbf{x}^{(i)}, y_i), (\mathbf{x}^{(i+1)}, y_{i+1})$. Proposition 3.1.4 will compute the exact weight cost of the optimal choice of the hyperplane.

Lemma 3.1.3. *If $\mathcal{D}$ is sampled from a regular polygon and if $sgn(y_i) = sgn(y_{i+1}) \neq 0$, then it is possible to relax $(\mathbf{x}^{(i)}, y_i), (\mathbf{x}^{(i+1)}, y_{i+1})$ with a single ReLU gate.*

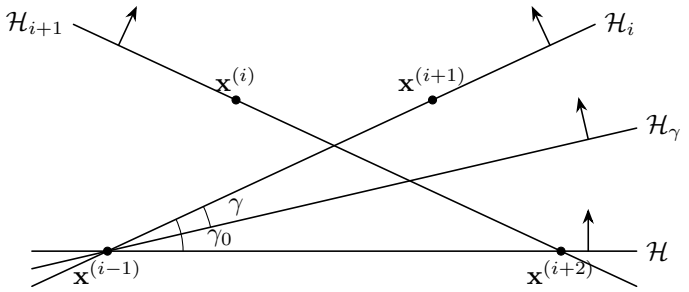

Figure 3.4: Diagram of the proof for Lemma 3.1.3.

Proof. Following the notation in the previous section, let $\mathcal{H}_i$ be the hyperplane that goes through $\mathbf{x}^{(i-1)}$ and $\mathbf{x}^{(i+1)}$ and is activated at $\mathbf{x}^{(i)}$ and let $\mathcal{H}_{i+1}$ be the one that passes through $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(i+2)}$ and is activated at $\mathbf{x}^{(i+1)}$. Additionally, let $\mathcal{H}$ be the hyperplane that passes through $\mathbf{x}^{(i-1)}$ and $\mathbf{x}^{(i+2)}$ and is activated at $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(i+1)}$. Let $\gamma_0 > 0$ denote the angle between $\mathcal{H}_i$ and $\mathcal{H}$.

For $\gamma \in (0, \gamma_0]$, consider the hyperplane $\mathcal{H}_\gamma$ that passes through $\mathbf{x}^{(i-1)}$ and forms an angle γ with $\mathcal{H}_i$. Then the ratio

$$r(\gamma) := \frac{d(\mathbf{x}^{(i)}, \mathcal{H}_\gamma)}{d(\mathbf{x}^{(i+1)}, \mathcal{H}_\gamma)} \tag{3.2}$$

is a continuous function of γ. In particular, $r(\gamma) \to \infty$ when $\gamma \to 0$ and $r(\gamma_0) = 1$. Similarly, for $\gamma \in [\gamma_0, 2\gamma_0)$, consider the hyperplane $\mathcal{H}_\gamma$ that passes through $\mathbf{x}^{(i+2)}$ and forms an angle $2\gamma_0 - \gamma$ with $\mathcal{H}$. Then the ratio $r(\gamma)$ is a continuous function of γ, where $r(\gamma) \to 0$ when $\gamma \to 2\gamma_0$. By the Intermediate Value Theorem, there exists a γ^* such that $r(\gamma) = \frac{y_i}{y_{i+1}}$. □

Now we compute the weight cost of the hyperplane we found.

Proposition 3.1.4. *If $\mathcal{D}$ is sampled from a regular polygon with side length r and if $sgn(y_i) = sgn(y_{i+1}) \neq 0$, then the minimum weight cost of relaxing a pair of points $(\mathbf{x}^{(i)}, y_i), (\mathbf{x}^{(i+1)}, y_{i+1})$ simultaneously using a single ReLU gate is*

$$\frac{\sqrt{(y_M - y_m)^2 + 2\cos\gamma_{n_d} y_M (y_M - y_m) + y_M^2}}{\sin\gamma_{n_d} \cdot r} \tag{3.3}$$

where $y_M = \max(|y_i|, |y_{i+1}|)$ and $y_m = \min(|y_i|, |y_{i+1}|)$

Proof. By Lemma 3.1.3, we are guaranteed a hyperplane $\mathcal{H}$ that can relax $(\mathbf{x}^{(i)}, y_i), (\mathbf{x}^{(i+1)}, y_{i+1})$. Also, by the construction of the proof, we are guaranteed one of the following:

1. $|y_i| > |y_{i+1}| \iff \mathcal{H}$ passes through $\mathbf{x}^{(i-1)}$ but not $\mathbf{x}^{(i+2)}$
2. $|y_i| < |y_{i+1}| \iff \mathcal{H}$ passes through $\mathbf{x}^{(i+2)}$ but not $\mathbf{x}^{(i-1)}$
3. $|y_i| = |y_{i+1}| \iff \mathcal{H}$ passes through both $\mathbf{x}^{(i-1)}$ and $\mathbf{x}^{(i+2)}$

For each case, it suffices to prove that

$$d(\mathbf{x}^{(i)}, \mathcal{H}) = \frac{\sin\gamma_{n_d} \cdot r \cdot |y_i|}{\sqrt{(y_M - y_m)^2 + 2\cos\gamma_{n_d} y_M (y_M - y_m) + y_M^2}} \tag{3.4}$$

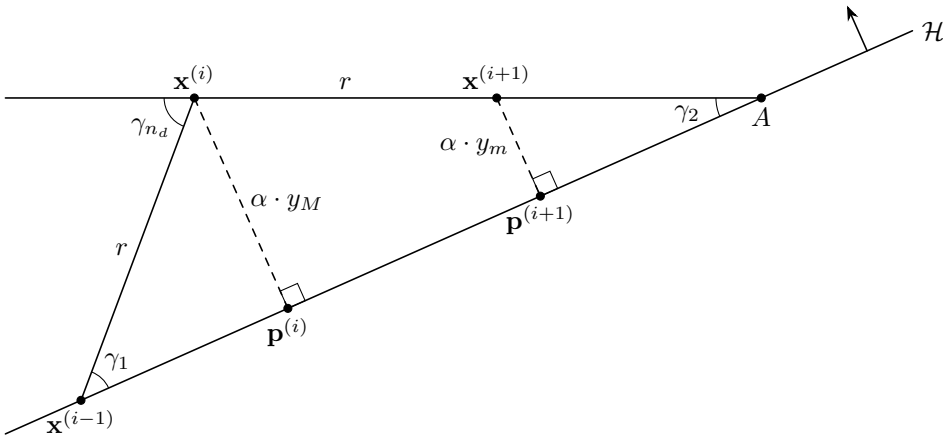

Figure 3.5: Diagram of Case 1 & 2 in the proof for Proposition 3.1.4.

Case 1 & 2 Relabel the indices if necessary to have $|y_i| > |y_{i+1}|$. That is, $y_M = |y_i|$ and $y_m = |y_{i+1}|$. Let $\mathbf{p}^{(i)}$ and $\mathbf{p}^{(i+1)}$ respectively denote the projection of $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(i+1)}$ onto $\mathcal{H}$. Let A denote the intersection between the line $\overline{\mathbf{x}^{(i)}\mathbf{x}^{(i+1)}}$ and $\mathcal{H}$. Then by symmetry of the triangles

$\triangle A\mathbf{x}^{(i)}\mathbf{p}^{(i)}$ and $\triangle A\mathbf{x}^{(i+1)}\mathbf{p}^{(i+1)}$, we see that

$$\overline{\mathbf{x}^{(i)}A} = r \cdot \frac{y_M}{y_M - y_m}$$

Next, by the choice of $\mathcal{H}$, we know that

$$\overline{\mathbf{x}^{(i)}\mathbf{p}^{(i)}} : \overline{\mathbf{x}^{(i+1)}\mathbf{p}^{(i+1)}} = y_M : y_m$$

Let $\overline{\mathbf{x}^{(i)}\mathbf{p}^{(i)}} = \alpha \cdot y_M$ and $\overline{\mathbf{x}^{(i+1)}\mathbf{p}^{(i+1)}} = \alpha \cdot y_m$. Also, let γ_1 denote the angle $\angle\mathbf{x}^{(i)}\mathbf{x}^{(i-1)}A$ and γ_2 denote $\angle\mathbf{x}^{(i)}A\mathbf{x}^{(i-1)}$. Then we notice that

$$\sin\gamma_1 = \frac{\alpha \cdot y_M}{r_{n_d}} \qquad \sin\gamma_2 = \frac{\alpha \cdot y_m}{r \cdot \frac{y_M}{y_M - y_m}} \tag{3.5}$$

At the same time, since $\gamma_{n_d} = \gamma_1 + \gamma_2$, we have $\sin\gamma_2 = \sin\gamma_{n_d}\cos\gamma_1 - \cos\gamma_{n_d}\sin\gamma_1$. We plug in the values from (3.5) to get

$$\alpha = \frac{\sin\gamma_{n_d} \cdot r}{\sqrt{(y_M - y_m)^2 + 2\cos\gamma_{n_d} y_M(y_M - y_m) + y_M^2}} \tag{3.6}$$

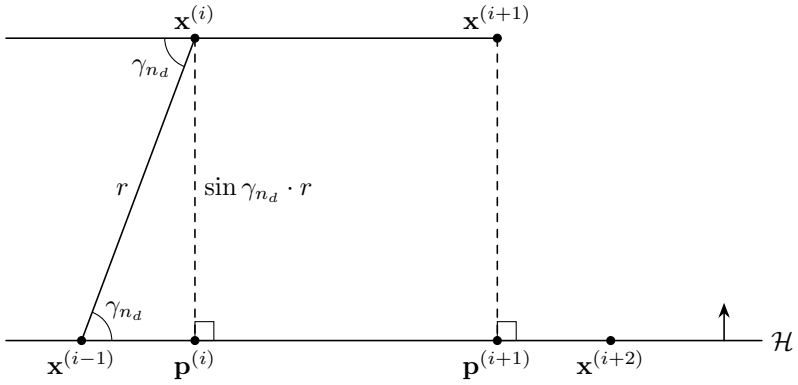

Figure 3.6: Diagram of Case 3 in the proof for Proposition 3.1.4.

Case 3 In this case, $y_M = y_m = |y_i| = |y_{i+1}|$. Let $\mathbf{p}^{(i)}$ and $\mathbf{p}^{(i+1)}$ respectively denote the projection of $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(i+1)}$ onto $\mathcal{H}$. Then we have $\angle\mathbf{x}^{(i)}\mathbf{x}^{(i-1)}\mathbf{x}^{(i+2)} = \gamma_{n_d}$ and therefore

$$d(\mathbf{x}^{(i)}, \mathcal{H}) = \overline{\mathbf{x}^{(i)}, \mathbf{p}^{(i)}} = \sin\gamma_{n_d} \cdot r \tag{3.7}$$

□

3.1.3 Relaxing Three or More Points

Now we wish to relax a number of points $(\mathbf{x}^{(i)}, y_i)$, $(\mathbf{x}^{(i+1)}, y_{i+1})$, $\cdots$, $(\mathbf{x}^{(i+k)}, y_{i+k})$. However, in general we cannot find a single ReLU gate to perform the task. To relax all points simultaneously, we require the distance between each point and the ReLU gate to be proportional to the value we want to assign. In the case of two points, the ReLU gate had a large degree of freedom to rotate to allow an arbitrary ratio between the two distances, but for a general number of points, it is extremely difficult to describe the set of ratios that are achievable. It likely also depends on n_d, the number of data points in the dataset, which does not allow for a general analysis. Instead, in this subsection, we investigate the process of using one ReLU gate to relax each pair of consecutive points. A clean mathematical formulation of the optimal choice of hyperplanes suggests that there is some underlying theory in play.

When we relax three consecutive points $(\mathbf{x}^{(i-1)}, y_{i-1})$, $(\mathbf{x}^{(i)}, y_i)$, $(\mathbf{x}^{(i+1)}, y_{i+1})$ by using ReLU gates that can only relax two points at a time, both ReLU gates will try to assign some values to the middle point $\mathbf{x}^{(i)}$. Therefore, we need to split y_i into $y_{i,1}$ and $y_{i,2}$ and allocate each of them to the corresponding ReLU gate. Theorem 3.1.5 states that the optimal allocation is proportional to the values of the surrounding points. That is, $y_{i,1} : y_{i,2} = y_{i-1} : y_{i+1}$.

Theorem 3.1.5. *Let $\mathcal{D}$ be sampled from a regular polygon with a side length r. If we wish to relax three points $(\mathbf{x}^{(i-1)}, y_{i-1})$, $(\mathbf{x}^{(i)}, y_i)$, $(\mathbf{x}^{(i+1)}, y_{i+1})$ simultaneously using exactly two ReLU gates, where $sgn(y_{i-1}) = sgn(y_i) = sgn(y_{i+1}) \neq 0$, then it is optimal to relax $(\mathbf{x}^{(i-1)}, y_{i-1}), (\mathbf{x}^{(i)}, \frac{y_{i-1}}{y_{i-1}+y_{i+1}} y_i)$ with one and $(\mathbf{x}^{(i)}, \frac{y_{i+1}}{y_{i-1}+y_{i+1}} y_i), (\mathbf{x}^{(i+1)}, y_{i+1})$ with the other.*

Proof. Without loss of generality, assume $y_{i-1}, y_i, y_{i+1} > 0$. For $y \in [0, y_i]$, let $c_1(y)$ be the weight cost of the first ReLU gate which relaxes the pair of points $(\mathbf{x}^{(i-1)}, y_{i-1}), (\mathbf{x}^{(i)}, y)$ and let $c_2(y)$ denote the weight cost for the other ReLU gate relaxing $(\mathbf{x}^{(i)}, y_i - y), (\mathbf{x}^{(i+1)}, y_{i+1})$. Then the total weight cost is $c(y) = c_1(y) + c_2(y)$, where c_1, c_2 is explicitly given by Proposition 3.1.4:

$$c_1(y) = \begin{cases} \frac{\sqrt{(y-y_{i-1})^2 + 2\cos\gamma_{n_d} y_{i-1}(y_{i-1}-y) + y_{i-1}^2}}{\sin\gamma_{n_d} \cdot r} & y \leq y_{i-1} \\ \frac{\sqrt{(y-y_{i-1})^2 + 2\cos\gamma_{n_d} y(y-y_{i-1}) + y^2}}{\sin\gamma_{n_d} \cdot r} & y \geq y_{i-1} \end{cases}$$

$$c_2(y) = \begin{cases} \frac{\sqrt{(y-y_i+y_{i+1})^2 + 2\cos\gamma_{n_d} (y-y_i)(y-y_i+y_{i+1}) + (y-y_i)^2}}{\sin\gamma_{n_d} \cdot r} & y \leq y_i - y_{i+1} \\ \frac{\sqrt{(y-y_i+y_{i+1})^2 + 2\cos\gamma_{n_d} y_{i+1}(y-y_i+y_{i+1}) + y_{i+1}^2}}{\sin\gamma_{n_d} \cdot r} & y \geq y_i - y_{i+1} \end{cases}$$

Notice that each of the cost function is piecewise twice-differentiable. The derivative is given as

$$c_1'(y) = \begin{cases} \frac{y-(1+\cos\gamma_{n_d})y_{i-1}}{\sin\gamma_{n_d}\cdot r\sqrt{(y-y_{i-1})^2+2\cos\gamma_{n_d}y_{i-1}(y_{i-1}-y)+y_{i-1}^2}} & <0 \quad y<y_{i-1} \\ \frac{2(1+\cos\gamma_{n_d})y-(1+\cos\gamma_{n_d})y_{i-1}}{\sin\gamma_{n_d}\cdot r\sqrt{(y-y_{i-1})^2+2\cos\gamma_{n_d}y(y-y_{i-1})+y^2}} & >0 \quad y>y_{i-1} \end{cases}$$

$$c_2'(y) = \begin{cases} \frac{2(1+\cos\gamma_{n_d})y-2(1+\cos\gamma_{n_d})(2y_i+y_{i+1})}{\sin\gamma_{n_d}\cdot r\sqrt{(y-y_i+y_{i+1})^2+2\cos\gamma_{n_d}(y-y_i)(y-y_i+y_{i+1})+(y-y_i)^2}} & <0 \quad y<y_i-y_{i+1} \\ \frac{y+(1+\cos\gamma_{n_d})y_{i+1}-y_i}{\sin\gamma_{n_d}\cdot r\sqrt{(y-y_i+y_{i+1})^2+2\cos\gamma_{n_d}y_{i+1}(y-y_i+y_{i+1})+y_{i+1}^2}} & >0 \quad y>y_i-y_{i+1} \end{cases} \tag{3.8}$$

By explicit calculation, we observe that $c_1''(y), c_2''(y) > 0$ except at the points $y = y_{i-1}$ or $y = y_i - y_{i+1}$ (if they exist in the domain of the functions), where the derivative may not be defined. Note that c_1', c_2' are negative on the left and positive on the right of their respective non-differentiable points. Since c_1, c_2 are continuous throughout the domain, we conclude that c_1, c_2 are strictly convex, which implies that $c = c_1 + c_2$ is also strictly convex. Therefore, it suffices to examine the first order conditions. Set $y^* = \frac{y_{i-1}}{y_{i-1}+y_{i+1}}y_i$. We prove that y^* is optimal in the following three cases.

Case 1 First consider the case where $y_i = y_{i-1} + y_{i+1}$. Then we have $y^* = y_{i-1} = y_i - y_{i+1}$, so the derivative is undefined for both c_1, c_2. But since the derivatives are negative on the left of y^* and positive on the right of y^*, we know that c obtains a unique minimum at y^*.

Case 2 Next consider the case $y_i < y_{i-1} + y_{i+1}$. Then $y^* < y_{i-1}$ and $y^* < y_i - y_{i+1}$. We plug in y^* to the first case of each formula in (3.8) to conclude $c_1'(y^*) + c_2'(y^*) = 0$.

Case 3 Finally consider the case $y_i > y_{i-1}+y_{i+1}$. Then $y^* > y_{i-1}$ and $y^* > y_i - y_{i+1}$. Similarly to Case 2, we plug in y^* to the second case of each formula in (3.8) to conclude $c_1'(y^*)+c_2'(y^*) = 0$. □

Now assume we relax $k+1$ consecutive points $(\mathbf{x}^{(i)}, y_i)$, $(\mathbf{x}^{(i+1)}, y_{i+1})$, $\cdots$, $(\mathbf{x}^{(i+k)}, y_{i+k})$ by using k ReLU gates each of which can only relax two points at a time. Then for each $j = 1, \cdots, k-1$, we need to split the value y_{i+j} into $y_{i+j,1}$ and $y_{i+j,2}$ and allocate each of them to the corresponding ReLU gate. Corollary 3.1.6 shows that the optimal allocation is proportional to the allocated values of the surrounding points (if such allocation exists).

Corollary 3.1.6. *Let $\mathcal{D}$ be sampled from a regular polygon with a side length r. If we wish to relax $k+1$ points $(\mathbf{x}^{(i)}, y_i)$, $(\mathbf{x}^{(i+1)}, y_{i+1})$, $\cdots$, $(\mathbf{x}^{(i+k)}, y_{i+k})$ simultaneously using at most k ReLU gates, where $sgn(y_i) = \cdots = sgn(y_{i+k}) \neq 0$, then for each $j = 1, \cdots, k$, it is optimal to relax $(\mathbf{x}^{(i+j-1)}, y_{i+j-1,2}), (\mathbf{x}^{(i+j)}, y_{i+j,1})$ if there exist allocation $y_{i+j,1}, y_{i+j,2} \in [0, y_{i+j}]$ (if $y_{i+j} > 0$) or*

$y_{i+j,1}, y_{i+j,2} \in [y_{i+j}, 0]$ *(if* $y_{i+j} < 0$*) such that*

1. $y_{i+j,1} + y_{i+j,2} = y_{i+j}$
2. $y_{i,1} = y_{i+k,2} = 0$
3. $y_{i+j,1} : y_{i+j,2} = y_{i+j-1,2} : y_{i+j+1,1}$

Proof. Let $\mathcal{H}_j$ be the ReLU gate that relaxes $(\mathbf{x}^{(i+j-1)}, y_{i+j-1,2}), (\mathbf{x}^{(i+j)}, y_{i+j,1})$ simultaneously. Similarly to the proof of Theorem 3.1.5, we can define a cost function c_j for $\mathcal{H}_j$. We can similarly show that the total cost of the ReLU gates is strictly convex. Also, note that the total cost depends on a particular variable $y_{i+j,1}$ or $y_{i+j,2}$ only through two cost functions c_j and c_{j+1}. Therefore, if there is a point with zero gradient, each variable has to satisfy the relationship prescribed by Theorem 3.1.5. □

Unfortunately, there are values $y_i, \cdots, y_{i+k}$ where no such allocation exists.

Example 3.1.7. *Let* $y_1 = 1$*,* $y_2 = 100$*,* $y_3 = 1$*,* $y_4 = 1$*,* $y_5 = 100$*. Assume that for each* $j = 2, 3, 4$*, we want to find values* $y_{j,1}, y_{j,2} \in [0, y_j]$ *such that* $y_{j,1} + y_{j,2} = y_j$ *and*

1. $y_{2,1} : y_{2,1} = y_1 : y_{3,1}$
2. $y_{3,1} : y_{3,2} = y_{2,2} : y_{4,1}$
3. $y_{4,1} : y_{4,2} = y_{3,2} : y_5$

Solving the system of linear equations gives $y_{3,2} = -\frac{4999}{51}$*, which does not have the same sign as* y_3 *and cannot be used as a solution.*

Based on the result of a few numerical experiments, we propose a few conjectures about how we can apply the result of Corollary 3.1.6.

Definition 3.1.8. *We say that a sequence of numbers* $\{y_1, \cdots, y_k\}$ *is an* ***increasing-decreasing sequence*** *if there exists* i *such that*

1. $sgn(y_1) = \cdots = sgn(y_k) \neq 0$
2. $|y_1| \leq \cdots \leq |y_i|$
3. $|y_i| \geq \cdots \geq |y_k|$

Conjecture 3.1.9. *If we have an increasing-decreasing sequence of values* $\{y_1, \cdots, y_k\}$*, then there exists an allocation* $y_{j,1}, y_{j,2}$ *that satisfy the condition of Corollary 3.1.6.*

Note that any sequence of length 1 is an increasing-decreasing sequence. Therefore, given an arbitrary sequence of values, we can decompose them into a concatenation of increasing-decreasing subsequences. Note that the values need not be of the same sign.

Lemma 3.1.10. *Given a sequence of values $\{y_1, \cdots, y_k\}$, it is possible to find $1 \le j \le k+1$ and indices $i_0 = 1 \le i_1 \le i_2 \le \cdots \le i_j = k$ such that for each $j' = 0, \cdots, j-1$, the subsequence $\{y_{i_{j'}}, \cdots, y_{i_{j'+1}-1}\}$ is an increasing-decreasing subsequence. We refer to the indices $\{i_0, \cdots, i_j\}$ as the* ***decomposition into increasing-decreasing subsequences****.*

We conjecture that given an arbitrary dataset, the optimal method of choosing ReLU gates that will each relax two points is to decompose the sequence into increasing-decreasing subsequences and relaxing each subsequence.

Conjecture 3.1.11. *Let $\mathcal{D}$ be sampled from a regular polygon with a side length r. If we wish to relax all points $(x^{(1)}, y^{n_d}), \cdots, (\mathbf{x}^{(n_d)}, y^{n_d})$ simultaneously, using ReLU gates that each relax at most two points, then there exists a decomposition of $\{y^{(1)}, \cdots, y^{(n_d)}\}$ into increasing-decreasing subsequences such that the optimal choice of ReLU gates is the union of optimal ReLU gates that relax each increasing-decreasing subsequence as prescribed by Corollary 3.1.6 and Conjecture 3.1.9.*

Informally, we also expect this solution to have a weight cost equal to or nearly equal to any $f \in RidgelessReLU(\mathcal{D})$.

3.2 Symmetric Dataset

In this section, we assume that the dataset is symmetric with respect to a line in the following sense:

Definition 3.2.1. *A dataset $\mathcal{D}$ is* ***symmetric*** *with respect to a line $\ell = \{\mathbf{x} \mid \mathbf{a} \cdot \mathbf{x} + b = 0\}$ if for any $(\mathbf{x}, y) \in \mathcal{D}$, we also have $(\tilde{\mathbf{x}}, y) \in \mathcal{D}$ where $\tilde{\mathbf{x}}$ is the reflection of $\mathbf{x}$ across ℓ.*

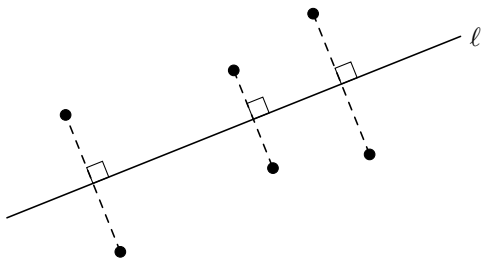

Figure 3.7: Example of a symmetric dataset $\mathcal{D}$.

The key idea is that a symmetric dataset is roughly 1-dimensional. Indeed, when the data points are projected to the line of symmetry ℓ, the dataset is still consistent. Then the natural question to

ask is whether the 1-dimensional solution from the projected space (that is, using only ReLU gates perpendicular to ℓ) will also fit the 2-dimensional dataset, and if so, whether it will minimize the weight cost.

In this section, we will first show that for any network that fits a dataset, which is symmetric with respect to ℓ, we can construct another network that fits the data with the same weight cost but is symmetric with respect to ℓ. Then we will show that replacing each pair of symmetric ReLU gates with ones that are perpendicular to ℓ (if possible to do so) uses strictly less weight cost. We will discuss the sufficient conditions of the dataset that allows all pairs of the ReLU gates to be replaced, which will rule out any solution that is not 1-dimensional.

First let us define what it means for a ReLU network to be symmetric:

Definition 3.2.2. *A 1-layer ReLU network $f(\mathbf{x};\theta)$ is **symmetric** with respect to a line $\ell = \{\mathbf{x} \mid \mathbf{a}\cdot\mathbf{x}+b=0\}$ if for any ReLU gate $W^{(2)}\left[d(\mathbf{x},\mathcal{H})\right]_+$ that defines f, there exists another ReLU gate $W^{(2)}\left[d(\mathbf{x},\widetilde{\mathcal{H}})\right]_+$ that defines f with the same weight $W^{(2)}$, where $\widetilde{\mathcal{H}}$ is the reflection of $\mathcal{H}$ across ℓ.*

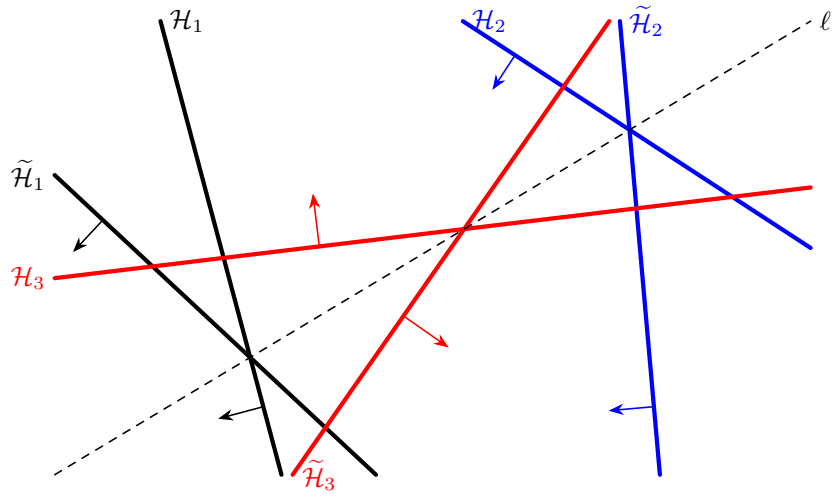

Figure 3.8: Example of a symmetric 1-layer ReLU network.

Theorem 3.2.3. *If $\mathcal{D}$ is symmetric with respect to a line ℓ and $f(\mathbf{x};\theta) \in ReLU(\mathcal{D})$, then there exists $\tilde{f}(\mathbf{x};\tilde{\theta}) \in ReLU(\mathcal{D})$ such that $\tilde{f}$ is symmetric with respect to ℓ and $C(\theta) = C(\tilde{\theta})$*

Proof. Let us decompose $f(\mathbf{x};\theta)$ into the sum of its ReLU gates:

$$f(\mathbf{x};\theta) = \sum_{i=1}^{n_h} W_i^{(2)}\left[d(\mathbf{x},\mathcal{H}_i)\right]_+ + \mathbf{a}\cdot\mathbf{x} + b^{(2)}$$

For each i, let $\widetilde{\mathcal{H}}_i$ be the reflection of hyperplane $\mathcal{H}_i$ for the i-th ReLU gate and let $\widetilde{\mathcal{A}} := \left\{\mathbf{x} \mid \tilde{\mathbf{a}}\cdot\mathbf{x}+\tilde{b}^{(2)}=0\right\}$

be the reflection of the residual hyperplane $\mathcal{A} := \left\{\mathbf{x} \mid \mathbf{a} \cdot \mathbf{x} + b^{(2)} = 0\right\}$. Now define a new network

$$\tilde{f}(\mathbf{x}; \tilde{\theta}) = \sum_{i=1}^{n_h} \frac{W_i^{(2)}}{2} \left[d(\mathbf{x}, \mathcal{H}_i)\right]_+ + \sum_{i=1}^{n_h} \frac{W_i^{(2)}}{2} \left[d(\mathbf{x}, \tilde{\mathcal{H}}_i)\right]_+ + \frac{\mathbf{a} + \tilde{\mathbf{a}}}{2} \cdot \mathbf{x} + \frac{b^{(2)} + \tilde{b}^{(2)}}{2}$$

Notice first that $\tilde{f}$ is a 1-layer ReLU network, with $2n_h$ ReLU gates, that is symmetric with respect to the line ℓ. Also for any $(\mathbf{x}, y) \in \mathcal{D}$, we have

$$\tilde{f}(\mathbf{x}; \tilde{\theta}) = \frac{f(\mathbf{x}; \theta) + f(\tilde{\mathbf{x}}; \tilde{\theta})}{2} = \frac{y + y}{2} = y$$

which shows that $\tilde{f}(\mathbf{x}; \tilde{\theta}) \in ReLU(\mathcal{D})$. □

3.2.1 One Pair of Parallel Lines

In this part, in addition to the assumption that the dataset $\mathcal{D}$ is symmetric with respect to a line ℓ, we further assume that they lie on a pair of parallel lines. That is, for any data point $(\mathbf{x}, y) \in \mathcal{D}$, it is on one of the two lines ℓ_1, ℓ_2 where the two lines are parallel to ℓ.

Theorem 3.2.4. *Assume $\mathcal{D}$ is symmetric with respect to ℓ and for any $(\mathbf{x}, y) \in \mathcal{D}$, we have $\mathbf{x} \in \ell_1 \cup \ell_2$ where ℓ_1, ℓ_2 are parallel to ℓ. If a ReLU network $f(\mathbf{x}; \theta) \in ReLU(\mathcal{D})$ consists only of a pair of distinct ReLU gates $\mathcal{H}_1, \mathcal{H}_2$ that are symmetric to each other with respect to ℓ such that*

$$f(\mathbf{x}; \theta) = W^{(2)} \left[d(\mathbf{x}, \widetilde{\mathcal{H}}_1)\right]_+ + W^{(2)} \left[d(\mathbf{x}, \widetilde{\mathcal{H}}_2)\right]_+ + \mathbf{a} \cdot \mathbf{x} + b^{(2)}$$

then there exists another ReLU network $\tilde{f}(\mathbf{x}; \tilde{\theta})$ that consists of at most two ReLU gates that are perpendicular to ℓ such that $\tilde{f}(\mathbf{x}; \tilde{\theta}) \in ReLU(\mathcal{D})$ and $C(\tilde{\theta}) < C(\theta)$.

Proof. We prove the theorem in the following two cases:

Case 1 Consider the case where $\mathcal{H}_1, \mathcal{H}_2$ are also parallel to ℓ. In particular, $\mathcal{H}_1, \mathcal{H}_2, \ell_1, \ell_2$ are all parallel to each other. Then for any $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}) \in \mathcal{D}$ such that $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in \ell_1$, we have

$$d(\mathbf{x}^{(1)}, \mathcal{H}_1) = d(\mathbf{x}^{(2)}, \mathcal{H}_1)$$

Let $d(\ell_1, \mathcal{H}_1)$ denote this common value. Similarly define $d(\ell_1, \mathcal{H}_2), d(\ell_2, \mathcal{H}_1), d(\ell_2, \mathcal{H}_2)$ to be the common distance from an appropriate data point to the corresponding hyperplane. Since ℓ_2 and $\mathcal{H}_2$

are respectively the reflections of ℓ_1 and $\mathcal{H}_1$ across ℓ,

$$d(\ell_1, \mathcal{H}_1) = d(\ell_2, \mathcal{H}_2) \quad \text{and} \quad d(\ell_1, \mathcal{H}_2) = d(\ell_2, \mathcal{H}_1)$$

This shows that for any $(\mathbf{x}, y) \in \mathcal{D}$, we have

$$f(\mathbf{x}; \theta) = W^{(2)} \left([d(\ell_1, \mathcal{H}_1)]_+ + [d(\ell_2, \mathcal{H}_1)]_+ \right) + \mathbf{a} \cdot \mathbf{x} + b^{(2)}$$

Notice that the first term is a constant, so we can absorb it into the bias. Then if we define

$$\tilde{f}(\mathbf{x}; \tilde{\theta}) := \mathbf{a} \cdot \mathbf{x} + \left(b^{(2)} + W^{(2)} \left([d(\ell_1, \mathcal{H}_1)]_+ + [d(\ell_2, \mathcal{H}_1)]_+ \right) \right) \tag{3.9}$$

then $\tilde{f} \in ReLU(\mathcal{D})$ and it does not use any ReLU gate. In particular, $C(\tilde{\theta}) = 0 < C(\theta)$

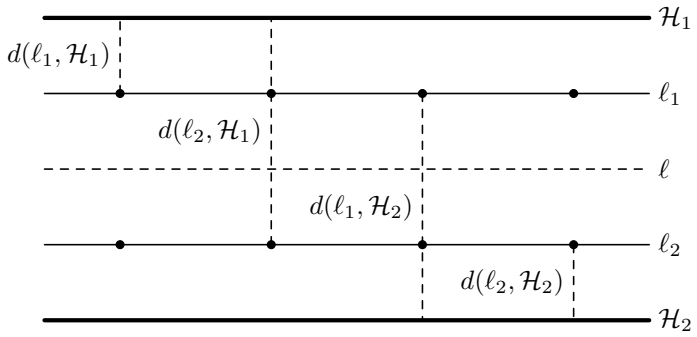

Figure 3.9: Diagram of Case 1 in the proof of Theorem 3.2.4.

Case 2 Consider the general case where $\mathcal{H}_1, \mathcal{H}_2$ are not parallel to ℓ. First notice that $\mathcal{H}_1, \mathcal{H}_2$ cannot be perpendicular to ℓ. If they were, $\mathcal{H}_1 = \mathcal{H}_2$ and the two ReLU gates would not have been distinct. For $i, j \in \{1, 2\}$, let A_{ij} denote the intersection of ℓ_i and $\mathcal{H}_j$. Define a ReLU gate $\widetilde{\mathcal{H}}_1$ such that it passes through A_{11} and A_{22} and $\widetilde{\mathcal{H}}_2$ that passes through A_{12} and A_{21}. Set the orientation of these ReLU gates such that they make an acute angle $\gamma \in \left(0, \frac{\pi}{2}\right)$ with both $\mathcal{H}_1, \mathcal{H}_2$. Alternatively, the sum of their normal vectors should have the same direction as the sum of the normal vectors for $\mathcal{H}_1, \mathcal{H}_2$.

Let $(\mathbf{x}, y) \in \mathcal{D}$. If $\mathbf{x} \in \ell_1$, then

$$d(\mathbf{x}, \widetilde{\mathcal{H}}_1) = \frac{d(\mathbf{x}, \mathcal{H}_1)}{\cos \gamma} \quad \text{and} \quad d(\mathbf{x}, \widetilde{\mathcal{H}}_2) = \frac{d(\mathbf{x}, \mathcal{H}_2)}{\cos \gamma}$$

Similarly, if $\mathbf{x} \in \ell_2$, then

$$d(\mathbf{x}, \widetilde{\mathcal{H}}_1) = \frac{d(\mathbf{x}, \mathcal{H}_2)}{\cos\gamma} \quad \text{and} \quad d(\mathbf{x}, \widetilde{\mathcal{H}}_2) = \frac{d(\mathbf{x}, \mathcal{H}_1)}{\cos\gamma}$$

Therefore, if we define

$$\tilde{f}(\mathbf{x}; \tilde{\theta}) = \cos\gamma \cdot W^{(2)} \left[d(\mathbf{x}, \widetilde{\mathcal{H}}_1) \right]_+ + \cos\gamma \cdot W^{(2)} \left[d(\mathbf{x}, \widetilde{\mathcal{H}}_2) \right]_+ + \mathbf{a} \cdot \mathbf{x} + b^{(2)} \tag{3.10}$$

then $\tilde{f} \in ReLU(\mathcal{D})$ and $C(\tilde{\theta}) = \cos\gamma \cdot C(\theta) < C(\theta)$.

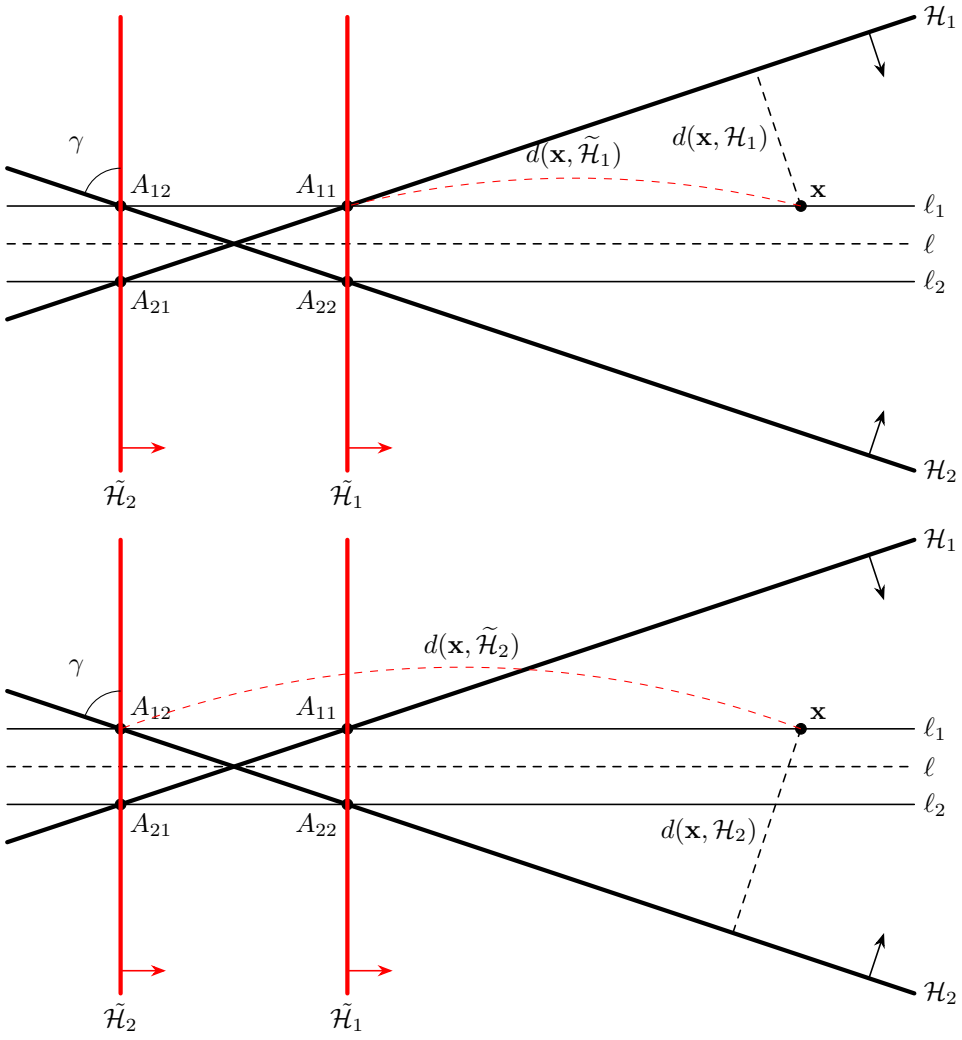

Figure 3.10: Diagram of Case 2 in the proof of Theorem 3.2.4.

□

Let us now interpret Theorem 3.2.4 once we project our input space $\mathbb{R}^2$ onto ℓ. The main observation in the proof was that the original ReLU network f, when restricted to the domain of ℓ_1, ℓ_2, is precisely equivalent to a continuous piecewise linear function in the projected space, where the breakpoints of the function are defined by A_{ij}. Then we can decompose this function as a sum of two ReLU components f_i in R. The next main observation is that for each f_i there is a unique ReLU gate $\widetilde{\mathcal{H}}_i$ in $\mathbb{R}^2$ such that its image under the projection is equal to f_i — in particular, $\widetilde{\mathcal{H}}_i$ is perpendicular to ℓ. Theorem 3.2.4 states that using this pair of ReLU gates uses a strictly less weight cost than the original pair of symmetric ReLU gates.

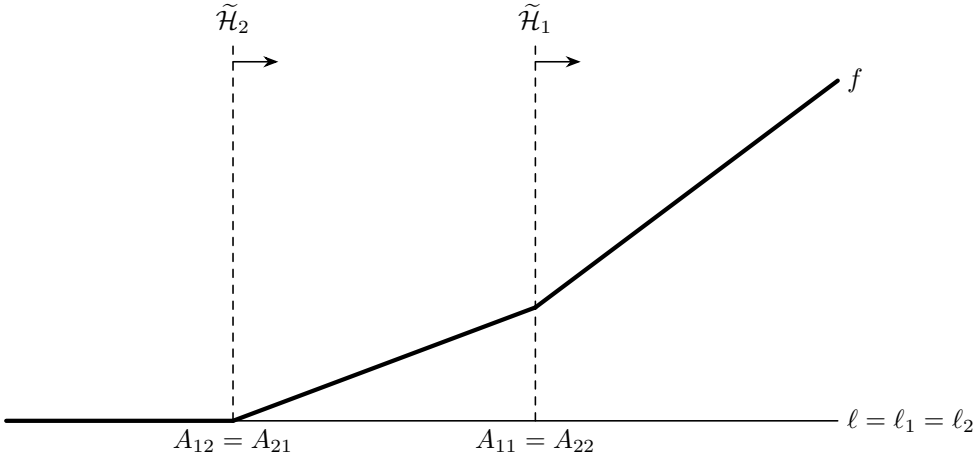

Figure 3.11: In Theorem 3.2.4, when the domain of f is restricted to ℓ_1, ℓ_2 and is projected onto ℓ, it is equivalent to a continuous piecewise linear function with 2 breakpoints. The slope of the function is respectively $0, W^{(2)}, 2W^{(2)}$ from left to right.

Notice that this construction does not take into account how the dataset $\mathcal{D}$ is actually distributed in the projected space. That is, it is the most general choice of ReLU gates that will work for an arbitrary dataset $\mathcal{D}$. However, consider a specific case where there are no points on ℓ_1 between A_{12} and A_{11} (equivalently, no points on ℓ_2 between A_{21} and A_{22}). Then Figure 3.12 shows that a single ReLU gate also interpolates the dataset in the projected space. The ReLU gate $\widetilde{\mathcal{H}}$ in the original space is perpendicular to ℓ and passes through A, the intersection between $\mathcal{H}$ and ℓ.

We formally introduce the result in the following theorem.

Theorem 3.2.5. *Assume $\mathcal{D}$ and $f(\mathbf{x};\theta) \in ReLU(\mathcal{D})$ satisfy the conditions of Theorem 3.2.4. Additionally, if $d(\mathbf{x}, \mathcal{H}_1) \cdot d(\mathbf{x}, \mathcal{H}_2) \geq 0$ [3] for any $(\mathbf{x}, y) \in \mathcal{D}$, then there exists another ReLU network $\tilde{f}(\mathbf{x};\tilde{\theta})$ that consists of at most one ReLU gate that is perpendicular to ℓ such that $\tilde{f}(\mathbf{x};\tilde{\theta}) \in ReLU(\mathcal{D})$ and $C(\tilde{\theta}) < C(\theta)$.*

[3] This is equivalent to saying $sgn(d(\mathbf{x}, \mathcal{H}_1)) = sgn(d(\mathbf{x}, \mathcal{H}_2))$ or $d(\mathbf{x}, \mathcal{H}_1) = 0$ or $d(\mathbf{x}, \mathcal{H}_2) = 0$. It allows $\mathbf{x}$ to be on one of the hyperplanes.

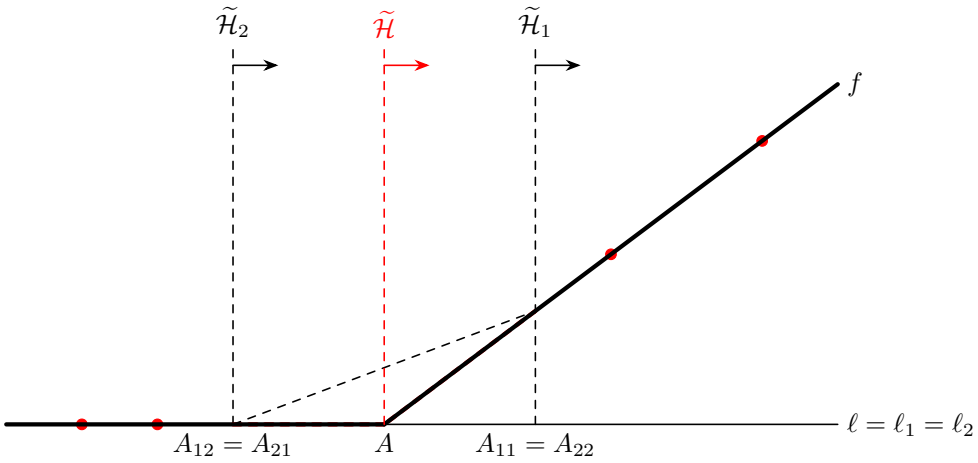

Figure 3.12: If no point existed on ℓ_1 between A_{12} and A_{11} (equivalently, no points on ℓ_2 between A_{21} and A_{22}), then the dataset can be interpolated by a single ReLU gate.

Proof. Let A be the intersection between $\mathcal{H}$ and ℓ, and let $\widetilde{\mathcal{H}}$ be the hyperplane that is perpendicular to ℓ and goes through A. For any $\mathbf{x}$, we have

$$d(\mathbf{x}, \widetilde{\mathcal{H}}_1) + d(\mathbf{x}, \widetilde{\mathcal{H}}_2) = 2d(\mathbf{x}, \widetilde{\mathcal{H}}) \tag{3.11}$$

At the same time,

$$\begin{aligned} d(\mathbf{x}, \mathcal{H}_1), d(\mathbf{x}, \mathcal{H}_2) \geq 0 \implies d(\mathbf{x}, \widetilde{\mathcal{H}}_1), d(\mathbf{x}, \widetilde{\mathcal{H}}_2) \geq 0 \implies d(\mathbf{x}, \widetilde{\mathcal{H}}) \geq 0 \\ d(\mathbf{x}, \mathcal{H}_1), d(\mathbf{x}, \mathcal{H}_2) \leq 0 \implies d(\mathbf{x}, \widetilde{\mathcal{H}}_1), d(\mathbf{x}, \widetilde{\mathcal{H}}_2) \leq 0 \implies d(\mathbf{x}, \widetilde{\mathcal{H}}) \leq 0 \end{aligned} \tag{3.12}$$

By (3.11) and (3.12), we can rewrite (3.10) as

$$\tilde{f}(\mathbf{x}; \tilde{\theta}) = 2\cos\gamma \cdot W^{(2)} \left[d(\mathbf{x}, \widetilde{\mathcal{H}}) \right]_+ + \mathbf{a} \cdot \mathbf{x} + b^{(2)} \tag{3.13}$$

□

3.2.2 General Case

In this part, we now consider the general case where $\mathcal{D}$ is symmetric, without the additional assumption that all data points lie on a single pair of parallel lines.

For each $(\mathbf{x}, y) \in \mathcal{D}$, we can draw a line $\ell_{\mathbf{x}}$ that goes through $\mathbf{x}$ and is parallel to ℓ. By the assumption that $\mathcal{D}$ is symmetric, we know that the collection of these lines will be form a set of symmetric pairs of lines $\{(\ell_{n,1}, \ell_{n,2})\}$. For each of these pair of symmetric lines, we can apply a

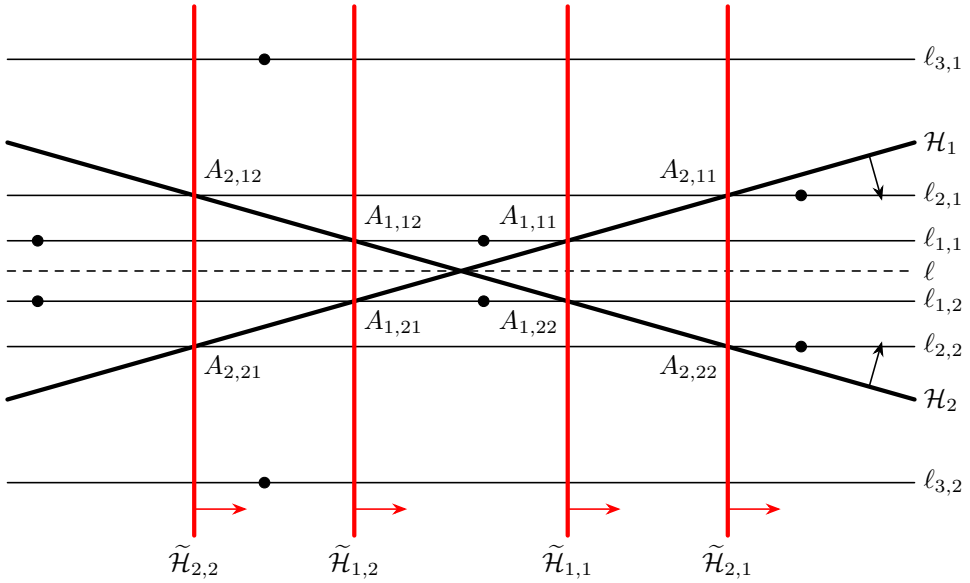

Figure 3.13: Diagram of $\mathcal{D}$ that is symmetric with respect to ℓ. Data points (bold black) lie on one of the pairs of symmetric lines $(\ell_{n,1}, \ell_{n,2})$.

similar analysis as in the previous section. In particular, let $A_{n,ij}$ be the intersection between $\ell_{n,i}$ and $\mathcal{H}_j$. Let $\widetilde{\mathcal{H}}_{n,1}$ be the hyperplane that goes through $A_{n,11}$ and $A_{n,22}$ and let $\widetilde{\mathcal{H}}_{n,2}$ be the one that goes through $A_{n,12}$ and $A_{n,21}$. When we consider the projected space, the function f on the projected space behaves differently, when the domain is restricted to a different pair $(\ell_{n,1}, \ell_{n,2})$ of symmetric lines. Specifically, the function should have two breakpoints, respectively at $A_{n,12} = A_{n,21}$ and $A_{n,11} = A_{n,22}$, but the projection of these points are different from each choice of n.

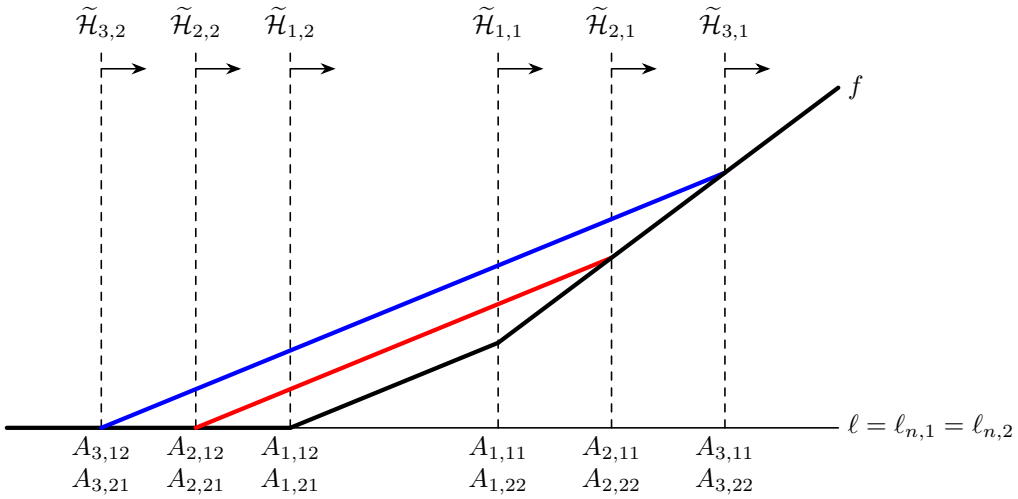

Figure 3.14: When the domain of f is restricted to $\ell_{n,1}, \ell_{n,2}$ and is projected onto ℓ, it is equivalent to a continuous piecewise linear function with 2 breakpoints at $A_{n,12} = A_{n,21}$ and $A_{n,11} = A_{n,22}$.

Figure 3.14 shows the general version of Figure 3.11. When the domain of f is restricted to each pair $(\ell_{n,1}, \ell_{n,2})$, it is equivalent to a continuous piecewise linear function with 2 breakpoints at $A_{n,12} = A_{n,21}$ and $A_{n,11} = A_{n,22}$. But each restriction results in a different function on the projected space. This hints that if we have data points between $A_{n,12} = A_{n,21}$ and $A_{n,11} = A_{n,22}$, then it is generally not possible to replace the ReLU gates $\mathcal{H}_1, \mathcal{H}_2$ with ones that are perpendicular to ℓ.

Proposition 3.2.6. *Given a line ℓ, there exist $\mathcal{D}$ and $f \in ReLU(\mathcal{D})$ that are symmetric with respect to ℓ such that there is no $\tilde{f} \in ReLU(\mathcal{D})$ that consists only of ReLU gates perpendicular to ℓ.*

Proof. Let ℓ be the x_1-axis. That is, $\ell = \{\mathbf{x} \mid (0,1) \cdot \mathbf{x} = 0\}$. If we define $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^3$ where the three data points are given as

$$\begin{cases} \mathbf{x}^{(1)} = (0,0), & y^{(1)} = 0 \\ \mathbf{x}^{(2)} = (0,1), & y^{(2)} = 1 \\ \mathbf{x}^{(3)} = (0,-1), & y^{(3)} = 1 \end{cases}$$

Then if we define $f : \mathbb{R}^2 \to \mathbb{R}$ as follows

$$f(\mathbf{x}; \theta) = \sqrt{2} \left[\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right) \cdot \mathbf{x} \right]_+ + \sqrt{2} \left[\left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right) \cdot \mathbf{x} \right]_+$$

we have $f \in ReLU(\mathcal{D})$. Now assume to the contrary that there is another network $\tilde{f} \in ReLU(\mathcal{D})$

$$\tilde{f}(\mathbf{x}; \tilde{\theta}) = \sum_{i=1}^{n_h} W_i^{(2)} \left[d(\mathbf{x}, \widetilde{\mathcal{H}}_i) \right]_+ + \mathbf{a} \cdot \mathbf{x} + b^{(2)}$$

where each $\widetilde{\mathcal{H}}_i$ are all perpendicular to ℓ. However, since $\mathbf{x}^{(i)}$ are colinear and the line through the three points is perpendicular to ℓ, we have for any i

$$d(\mathbf{x}^{(1)}, \widetilde{\mathcal{H}}_i) = d(\mathbf{x}^{(2)}, \widetilde{\mathcal{H}}_i) = d(\mathbf{x}^{(3)}, \widetilde{\mathcal{H}}_i)$$

At the same time, the residual connection $\mathbf{x} \mapsto \mathbf{a} \cdot \mathbf{x} + b^{(2)}$ is linear and $2\mathbf{x}^{(1)} = \mathbf{x}^{(2)} + \mathbf{x}^{(3)}$ so

$$2(\mathbf{a} \cdot \mathbf{x}^{(1)} + b^{(2)}) = (\mathbf{a} \cdot \mathbf{x}^{(2)} + b^{(2)}) + (\mathbf{a} \cdot \mathbf{x}^{(3)} + b^{(2)})$$

which suggests that we should have $\tilde{f}(\mathbf{x}^{(1)}; \tilde{\theta}) = \tilde{f}(\mathbf{x}^{(2)}; \tilde{\theta}) = \tilde{f}(\mathbf{x}^{(3)}; \tilde{\theta})$, but this is against our assumption that $\tilde{f} \in ReLU(\mathcal{D})$. This is the desired contradiction. □

The main bottleneck in Proposition 3.2.6 was that in the region between $\mathcal{H}_1, \mathcal{H}_2$, f is inconsistent when projected to ℓ. Therefore, if we had any data point in the region, we cannot perfectly reconstruct the function using only ReLU gates perpendicular to ℓ. However, it is possible if we remove all such data points.

Theorem 3.2.7. *Assume $\mathcal{D}$ is symmetric with respect to ℓ. If $f \in ReLU(\mathcal{D})$ consists only of a pair of distinct ReLU gates $\mathcal{H}_1, \mathcal{H}_2$ that are symmetric to each other with respect to ℓ and if $d(\mathbf{x}, \mathcal{H}_1) \cdot d(\mathbf{x}, \mathcal{H}_2) \geq 0$ for any $(\mathbf{x}, y) \in \mathcal{D}$, then there exists another ReLU network $\tilde{f}(\mathbf{x}; \tilde{\theta})$ that consists of at most one ReLU gate that is perpendicular to ℓ such that $\tilde{f}(\mathbf{x}; \tilde{\theta}) \in ReLU(\mathcal{D})$ and $C(\tilde{\theta}) < C(\theta)$.*

Proof. The choice of ReLU network $\tilde{f}$ as in (3.13) suffices for the proof. □

3.2.3 Main Theorem

We are now ready to present the main theorem of this section.

Theorem 3.2.8. *Let $\mathcal{D}$ be symmetric to ℓ and $f \in ReLU(\mathcal{D})$. If f contains at least one ReLU gate $\mathcal{H}$ that is not perpendicular to ℓ and if one of the following conditions is satisfied:*

1. *There exist two lines ℓ_1, ℓ_2 parallel to ℓ such that for any $(\mathbf{x}, y) \in \mathcal{D}$, $\mathbf{x} \in \ell_1 \cup \ell_2$;*
2. *For any $(\mathbf{x}, y) \in \mathcal{D}$, $d(\mathbf{x}, \mathcal{H})) \cdot d(\mathbf{x}, \widetilde{\mathcal{H}}) \geq 0$, where $\widetilde{\mathcal{H}}$ is the reflection of $\mathcal{H}$ across ℓ,*

then $f \notin RidgelessReLU(\mathcal{D})$.

Proof. By Theorem 3.2.3, we can construct a symmetric network $\tilde{f}(\mathbf{x}; \tilde{\theta}) \in ReLU(\mathcal{D})$ with the same weight cost as f. In particular, $\tilde{f}$ contains both $\mathcal{H}$ and $\widetilde{\mathcal{H}}$. Let g denote the ReLU network, consisting only of $\mathcal{H}$ and $\widetilde{\mathcal{H}}$. Then Theorem 3.2.4 and Theorem 3.2.7 guarantees that there is another pair $\tilde{g}$ of ReLU gates that use strictly less weight cost such that $g = \tilde{g}$ on $\mathcal{D}$. Replacing g with $\tilde{g}$ in $\tilde{f}$ will return another interpolant of $\mathcal{D}$ but will use strictly less weight cost. Therefore, $\tilde{f}$ does not have the minimum weight cost among the ReLU networks in $ReLU(\mathcal{D})$ and nor does f. □

Bibliography

[1] P. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. In *The Journal of Machine Learning Research*, volume 3, pages 224–240, 06 2001.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

[3] G. V. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[4] B. Hanin. Ridgeless interpolation with shallow relu networks in $1d$ is nearest neighbor curvature extrapolation and provably generalizes on lipschitz functions, 2021.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 06 2016.

[6] J. X. Juncai He, Lin Li and C. Zheng. Relu deep neural networks and linear finite elements. *Journal of Computational Mathematics*, 38(3):502–527, 2020.

[7] B. Neyshabur, R. Tomioka, and N. Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning, 2015.

[8] G. Ongie, R. Willett, D. Soudry, and N. Srebro. A function space view of bounded norm infinite width relu nets: The multivariate case, 2019.

[9] OpenAI. Gpt-4 technical report, 2023.

[10] P. Savarese, I. Evron, D. Soudry, and N. Srebro. How do infinite width bounded norm networks look in function space?, 2019.

[11] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of Complexity*, pages 11–30. Springer International Publishing, 1971.

[12] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12104–12113, June 2022.