Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Chair for Mathematical Information Science
Prof. Dr. H. Bölcskei

Student Project / Master Thesis in Information Technology and Electrical Engineering

Fall/Spring Semester 2024/2025

Philipp Nazari

Why do Neural Networks Generalize?

Supervisors: Clemens Hutter

Oktober 2025

Abstract

Acknowledgments

Contents

Notation

Chapter 1

Introduction

While conventional learning theory would predict generalization error to increase in the highly overparameterized regime, empirical experiments show that deep neural network (DNN) training exhibits the so-called “double-descent“ phenomenon [1]. As parameter-count increases, test-error first gets worse before it gets better again. A reoccurring argument attributes the good performance of DNNs in the overparameterized regime to implicit regularization introduced by first order optimization techniques like SGD [2, 3, 4].

A related phenomenon we are lacking a fundamental understanding of is the so-called *simplicity bias* [5, 6], which describes the observation that machine learning algorithms often converge towards simple models even though their expressivity would allow them to overfit the data [7].

[8] connect these two phenomena by arguing that our physical universe is “simple“ and therefore a training algorithm should favor simple hypotheses over complex ones in order to accurately capture the true rule of the universe, i.e. to generalize. It nevertheless remains an open question why optimizers are able to find simple, well generalizing minima even in networks powerful enough to overfit excessively. Using a Guess & Check algorithm, [9] show empirically that this simplicity bias also exists under zeroth order optimizers, leading them to attribute it solely to the nature of the loss landscape. They conjecture that the crucial selective bias comes from the geometry of the loss landscape and that “good hypotheses occupy larger volumes“, coining the claim the **volume hypothesis**. In this work, we take a Bayesian perspective on neural networks to study the volume hypothesis in greater detail.

Chapter 2

Mathematical Background

2.1 Neural Networks

In this section we first introduce the general concept of a fully connected, feed forward artificial neural networks before focusing on ReLU networks for binary classification. We will then give a brief overview over different training algorithms for neural networks.

2.1.1 Linear, Fully Connected Feed Forward Networks

A linear, fully connected feed forward network f with L layers can be thought of as a concatenation of L functions $f = f_1 \circ \ldots \circ f_L \colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$, where each f_i acts like $f_i \colon \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i}$ for some natural numbers $n_0, \ldots, n_L$. Each of the f_i can be thought of as consisting of n_i computational units – its coordinate functions, also called *neurons* – which form the *i'th layer* of the network. Every such neuron is a map $\mathbb{R}^{n_{i-1}} \to \mathbb{R}$. A network is called deep if $L \gg 1$ and wide if $n_i \gg 1$.

The first layer f_1 is the *input-layer* and acts like the identity. The last layer is called the *output layer*. All other layers are called *hidden layers*. We will study them in the following few paragraphs. Fix $i > 1$. The j'th neuron in the i'th layer is assigned a *weight*-vector $\mathbf{w}_{i,j} \in \mathbb{R}^{n_{i-1}}$ and a *bias* $b_{i,j} \in \mathbb{R}$. Given an an input $\mathbf{x} \in \mathbb{R}^{n_{i-1}}$, the neuron computes $a_{i,j} := \langle \mathbf{x}, \mathbf{w}_{i,j} \rangle + b_{i,j} \in \mathbb{R}$. The value $a_{i,j}$ is called the *pre-activation* of neuron j in layer i.

In order to obtain a simpler formalism and parallelize computations, we group together all of the n_i neurons of layer i to obtain a *weight-matrix* $\mathbf{W}_i \in \mathbb{R}^{n_i, n_{i-1}}$, which contains $\mathbf{w}_{i,j}$ in its j'th row. The biases $b_{i,j}$ are grouped in a vector $\mathbf{b}_i \in \mathbb{R}^{n_i}$. The pre-activation of layer i thus becomes

$$f_i(\mathbf{x}) = \mathbf{W}_i \mathbf{x} + \mathbf{b}_i \in \mathbb{R}^{n_i}. \tag{2.1}$$

Given that the neural network is the concatenation of the layer functions f_i, the recursive definition of our network is

$$\mathbf{a}_1 \coloneqq \mathbf{x} \tag{2.2}$$

$$f_{i+1}(\mathbf{a}_i) \coloneqq \mathbf{a}_{i+1} \coloneqq \mathbf{W}_{i+1}\mathbf{a}_i + \mathbf{b}_{i+1},\ i = 1, \ldots, L-1, \tag{2.3}$$

where $\mathbf{x}$ is the *input* and $\mathbf{y} \coloneqq \mathbf{a}_L$ the *output* or *prediction* of the neural network. Equation (2.2) explains why we think of $\mathbf{x}$ as propagating forward through the network. The whole process is called the *forward pass*.

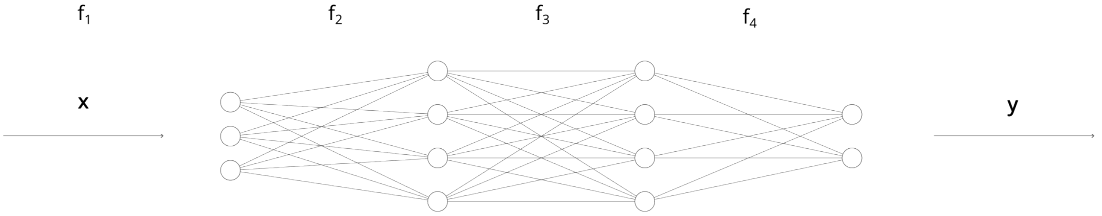


Figure 2.1: The computational graph of a neural network with $L = 4$ layers. The first layer f_1 acts like the identity and feeds the input into the network. The consecutive layers f_2, f_3, f_4 propagate the signal forward through the network until it outputs a value $\mathbf{y} \coloneqq \mathbf{a}_L$.

For ease of notation we will from now on summarize all of the p parameters of a neural network – that is, its weights and biases – in a *parameter-vector* $\theta \in \mathbb{R}^p$, denoting the neural network with parameters θ as f_θ. The map $\mathcal{M}\colon \theta \mapsto f_\theta$ assigning a parameter vector to the corresponding neural network is called the *parameter map*.

2.1.2 Architecture of ReLU Networks

The neural network described in Section 2.1.1 is a concatenation of linear functions and thus itself linear. In practice one adds an additional step to every iteration of the feed forward process, which increases the expressivity of the neural network (that is, the class of functions one can express with a given network architecture). [10] show that the following architecture creates universal approximators of continuous functions.

The additional step requires a non-linear *activation function* $\rho\colon \mathbb{R} \to \mathbb{R}$, which is applied to the pre-activations at any given layer. A complete set of defining equation is thus given by

$$\mathbf{a}_1 \coloneqq \mathbf{x} \tag{2.4}$$

$$f_{l+1}(\mathbf{a}_l) \coloneqq \mathbf{a}_{l+1} \coloneqq \rho(\mathbf{W}_{l+1}\mathbf{a}_l + \mathbf{b}_{l+1}),\ l = 1, \ldots, L-1, \tag{2.5}$$

where ρ is applied element-wise.

In this work we focus on non-linearities ρ_{t_l} of the form

$$\begin{aligned}\rho_{t_l} \colon \mathbb{R} &\to \mathbb{R} \\ x &\mapsto \max(x, t_l)\end{aligned}$$

where $t_l \in \mathbb{R} \cup \{-\infty\}$ is the *threshold*. We want to mention two special cases:

1. ρ_0 is the ReLU ("rectified linear unit")
2. $\rho_{-\infty}$ is the identity.

A more complete comparison of different activation functions used in deep learning can be found in [11].

In the following we give a definition of the neural networks we consider.

Definition 2.1.1 (Neural Network)**.** A network $f \colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ for binary classification takes as an input a vector $\mathbf{x} \in \mathbb{R}^{n_0}$ and returns an output $\mathbf{y} := \mathbf{a}_L$. It is defined inductively by

$$\begin{cases} \mathbf{a}_1 := \mathbf{x} \\ \mathbf{a}_{l+1} = \rho_{t_{l+1}} \left(\mathbf{W}_{l+1} \mathbf{a}_l + \mathbf{b}_{l+1} \right), \ l = 1, \ldots, L-1 \end{cases}$$

where $\mathbf{W}_l \in \mathbb{R}^{n_l, n_{l-1}}$ and $\mathbf{b}_l \in \mathbb{R}^{n_l}$ are the *weight matrix* and *bias vector* at layer l. The scalar t_l is the *threshold* at *layer l*. The number L is called the *depth* of the network, while n_l is the *width* of layer l. The *parameter vector $\theta \in \mathbb{R}^p$* summarizes all of the weights and biases. The network is *deep* if $L \gg 1$.

Remark 2.1.2. We will think of the *activations* $\mathbf{a}_l$ at layer l as a function of the input $\mathbf{x} \in \mathbb{R}^{n_0}$, i.e. $\mathbf{a}_l \colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_l}$.

2.1.3 Classification vs. Regression

A fundamental objective of machine learning is learning unknown functions from samples (x_i, y_i) of input-output pairs. If y_i can take continuous values we speak of regression. An example would be matching economic information to stock price. If y_i can only take discrete values we speak of classification and call the y_i the *class* or *label* of the sample. An example would be classifying images of cats and dogs.

While for regression a typical architecture can look like the one introduced in Definition 2.1.1, the classification settings needs a small adjustment.

In this workwe are specifically interested in the case of binary classification, i.e. $n_L = 1$.

Definition 2.1.3 (Binary-Classification Network)**.** A *binary classification network* is a neural network in the sense of Definition 2.1.1 where the last layer has width $n_L = 1$. Furthermore, the threshold at the last layer is $t_L = -\infty$. In this paper, we will use $t_l = 0$ for $l < L$ and thus speak of a *ReLU (binary) classification network*.

The output of the binary classification network is interpreted as a vote for the class label.

Definition 2.1.4. A *scoring function* takes the output returned by a binary classification network and classifies the corresponding sample. In our setting, it will take the form

$$\begin{aligned} s\colon \mathbb{R} &\to \{-1, 1\} \\ \tilde{y} &\mapsto \begin{cases} 1 & \tilde{y} \geq 0 \\ -1 & \tilde{y} < 0. \end{cases} \end{aligned}$$

In order to measure the complexity of a ReLU binary classification network, we need the following definitions:

Definition 2.1.5. Let f_θ be a ReLU binary classification network in the sense of Definition 2.1.3. Then the *decision boundary* of f_θ is the set

$$\mathcal{B}_\theta := f_\theta^{-1}(0). \tag{2.6}$$

2.1.4 Training Neural Networks

One can argue that the power of neural networks lies in their many degrees of freedom and the ability to choose them automatically. In order to do so, one defines a *loss $\mathcal{L}$* which the network should minimize. In the setting of regression this loss might, for example, measure how good a prediction a_L describes a true output y using the mean squared error:

$$\mathcal{L}(a_L, y) = \|a_L - y\|_2^2.$$

We usually assume to have a *training set* of input-output pairs (x, y) of the true function. Using those samples, the goal is to learn the data generating process matching x to y. A common way for the network to automatically adjusting the network parameters towards a minimum of $\mathcal{L}$ is *gradient descent*, which iteratively updates the weights of the network like

$$\theta \mapsto \theta - \alpha \nabla_\theta \mathcal{L}, \tag{2.7}$$

where $\alpha > 0$ is a *learning rate*. Intuitively, gradient descent adjusts the best guess for the networks parameters by iteratively walking down the loss landscape in the direction of steepest descent. Since it uses the gradient of the loss to do so, gradient descent is also called a *first order* optimization technique.

Since in general we do not have access to the gradient of the loss function, it is estimated empirically using the training data. The resulting algorithm is called *stochastic gradient descent (SGD)*. We want to point out that in practice one uses refined versions of SGD, for example ADAM [12].

In this work we will mainly study another optimization algorithm called *Guess & Check (G&C)* [9]. The optimizer gets by without any gradient and thus falls in the category of *zero'th order* optimization techniques. Instead, it randomly generates a solution from parameter space until it finds one with low training error (see Algorithm 1).

Algorithm 1 Guess and Check Algorithm for Sampling Parameters θ

```
Initialize $\mathcal{L}_{\text{train}} \leftarrow \infty$
while $\mathcal{L}_{\text{train}} \geq \varepsilon$ do
    Uniformly sample a random parameter vector $\theta$
    Compute the training error $\mathcal{L}_{\text{train}}(\theta)$
    if $\mathcal{L}_{\text{train}}(\theta) < \varepsilon$ then
        Return $\theta$
    end if
end while
```

2.2 Polyhedral Complexes

In this section we introduce some basic knowledge about polyhedral complexes, which will be useful for our studies of affine and tropical geometry. Throughout this section we fix a dimension $d \in \mathbb{N}$.

Definition 2.2.1 (Polyhedron & Polytope)**.** A *polyhedron P* is the intersection of finitely many closed half-spaces in $\mathbb{R}^d$:

$$P = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$$

for some $\mathbf{A} \in \mathbb{R}^{m,d}$ and $\mathbf{b} \in \mathbb{R}^m$. A *polytope* is a bounded polyhedron.

Lemma 2.2.2 ([13] (Section 2.3))**.** *The convex hull of finitely many vertices is a polytope.*

Definition 2.2.3 (Face of a Polytope)**.** Let $P \subseteq \mathbb{R}^d$ be a polytope. Then a *face* of P is a subset $F \subseteq P$ s.t.

$$F = P \cap \{\mathbf{A}'\mathbf{x} = \mathbf{b}'\} \tag{2.8}$$

where $\mathbf{A}'$ arises from $\mathbf{A}$ and $\mathbf{b}'$ arises from $\mathbf{b}$ by deleting rows with the same indices (that is, some of the inequalities defining P are satisfied as equalities). The normal vector related to F pointing into F is called the *inner normal vector* of F.

Definition 2.2.4 (Dimension of a Face)**.** The *dimension* of a face is the dimension of the smallest affine subspace containing it. Zero-dimensional faces are called *vertices*.

Definition 2.2.5 (Upper Convex Hull)**.** Consider the polytope P formed by the convex hull $\mathcal{C}(S)$ of a finite set of points $S \subseteq \mathbb{R}^d$. Then a *upper face* of P is a face whose inner normal vector $\mathbf{n}$ has negative last coordinate. We call the union of all upper faces the *upper convex hull* of S, denoted by $\mathcal{U}(S)$. The union of all k-faces in $\mathcal{U}(S)$ is denoted by $\mathcal{U}_k(S)$. In the specific case where $k = 0$, we write $\mathcal{U}^*(S) := \mathcal{U}_0(S)$.

Definition 2.2.6 (Polyhedral Complex)**.** A *polyhedral complex* is a collection Σ of polyhedra P satisfying two conditions:

i) if P is a polyhedron contained in Σ, then any face of P is also contained in Σ

ii) if P and Q are both polyhedra contained in Σ, then $P \cap Q$ is either empty or a face of both P and Q.

Definition 2.2.7 (Support). The *support* of a polyhedral complex Σ in $\mathbb{R}^d$ is

$$|\Sigma| := \{\mathbf{x} \in \mathbb{R}^d \,|\, \mathbf{x} \in P \text{ for some polyhedron} P \in \Sigma\}.$$

Definition 2.2.8 (Cells). The polyhedra in a polyhedral complex Σ are called *cells*. The *dimension* of a cell σ is the dimension of the smallest affine subspace containing it.

Definition 2.2.9 (K-Skeleton). The k-skeleton of a polyhedral complex Σ is the polyhedral complex consisting of all cells $\sigma \in \Sigma$ with dimension $\dim(\sigma) = k$.

2.3 On Sets and Functions

In this section we collect a number of useful definitions and basic statements. Throughout this section, fix a dimension $d \in \mathbb{N}$.

Definition 2.3.1 (Point-Function).

i) given a function $f\colon \mathbb{R}^d \to \mathbb{R}$ and a point $\mathbf{y} \in \mathbb{R}^{d+1}$, we write $\mathbf{y} \in f$ if $\mathbf{y}$ lies in the graph of f, i.e. there exists an $\mathbf{x} \in \mathbb{R}^d$ such that $\mathbf{y} = (\mathbf{x}, f(\mathbf{x}))$

ii) given a function $f\colon \mathbb{R}^d \to \mathbb{R}$ and a point $(\mathbf{x}, y) \in \mathbb{R}^{d+1}$, we say that $\mathbf{x}$ lies *above* f if $y > f(\mathbf{x})$. In this case, we write $\mathbf{x} \succ f$. If $\mathbf{x}$ lies above or on f, i.e. $y \geq f(\mathbf{x})$, we write $\mathbf{x} \succeq f$. Similarly, we write $\mathbf{x} \prec f$ if $\mathbf{x}$ lies *below* f and $\mathbf{x} \preceq f$ if $\mathbf{x}$ lies below or on f.

Definition 2.3.2 (Set-Function). Given a function $f\colon \mathbb{R}^d \to \mathbb{R}$ and a set $X \subseteq \mathbb{R}^{d+1}$, we write $f \succ X$ if $f \succ x$ for all $x \in X$. We analogously define $f \succeq X$, $f \prec X$ and $f \preceq X$.

Definition 2.3.3 (Set-Point). Given a point $(\mathbf{x}, y) \in \mathbb{R}^{d+1}$ and a subset $U \subseteq \mathbb{R}^{d+1}$, we say that $(\mathbf{x}, y)$ lies below U if $y < u$ for all $(\mathbf{t}, u) \in U$.

Definition 2.3.4 (Linear Independence of Functions). We say that a family $(f_{\mathbf{a}_i,b_i})_{i=1,\dots,n}$ of affine function is *linearly independent* if they are linearly independent as functions, i.e. there exist no family of scalars $(\alpha_i)_{i=1,\dots,n} \subseteq \mathbb{R}$, not all zero, s.t. $\sum_i^n \alpha_i f_{\mathbf{a}_i,b_i}(x) = 0$ for all $x \in \mathbb{R}^d$.

Definition 2.3.5 (Affine Independence). A finite set $X \subset \mathbb{R}^d$ of size n is called affinely independent if there does not exist a family of scalars $\alpha_i \in \mathbb{R}$, not all zero, such that

$$\sum_{i=1}^n \alpha_i x_i = 0, \quad \sum_{i=1}^n \alpha_i = 0.$$

Lemma 2.3.6. *Let $X \subset \mathbb{R}^d$ be a finite set of points. Then the extreme points S of $\mathcal{C}(X)$ are affinely independent.*

Proof. By contradiction. Assume the extreme points $S = \{s_1, \dots, s_n\}$ where affinely dependent, i.e. there existed a family of scalars $(\alpha_i)_{i=1,\dots,n}$ such that $\sum_{i=1}^n \alpha_i s_i = 0$ and $\sum_{i=1}^n \alpha_i = 0$. Assume w.l.o.g. that $\alpha_n \neq 0$. Then

$$\sum_{i=1}^{n-1} \beta_i s_i = s_n, \quad \sum_{i=1}^{n-1} \beta_i = 1$$

where $\beta_i := \frac{\alpha_i}{\alpha_n}$.

Let $I^+ := \{i \in \{1, \ldots, n-1\} \,|\, \beta_i > 0\}$ and $I^+ := \{i \in \{1, \ldots, n-1\} \,|\, \beta_i < 0\}$. Then

$$s_n = \sum_{i \in I^+} \beta_i s_i + \sum_{i \in I^-} \beta_i s_i.$$

Define furthermore

$$L := \sum_{i \in I^+} \beta_i.$$

Assume first that $0 < L < 1$. Then

$$s_n = L \left(\sum_{i \in I^+} \frac{\beta_i}{L} s_i \right) + (1 - L) \left(\sum_{i \in I^-} \frac{\beta_i}{L} s_i \right).$$

We arrive at a contradiction if we can show that the two sums lie in $\mathcal{C}(S)$, since then we could write an extreme point of $\mathcal{C}(S)$ as a strict convex combination of the other extreme points. Indeed, $\sum_{i \in I^+} \frac{\beta_i}{1-L} = 1$ and $\sum_{i \in I^-} \frac{\beta_i}{1-L} = 1$ by construction.

The case $L = 0$ is not possible since $\sum_{i=1}^{n-1} \beta_i = 1$. If $L = 1$, then s_n is already a strict convex combination of the other extreme-points, which is a contradiction. □

Lemma 2.3.7. *Let $X = (x_1, \ldots, x_n) \subseteq \mathbb{R}^d$ be a set of n linearly independent points in $\mathbb{R}^d$. Then the affine span of X (i.e. the convex hull $\mathcal{C}(X)$) has dimension $n - 1$.*

Proof. Let $\mathbf{v}_i := x_1 - x_n$ for $i = 1, \ldots, n-1$. Then the family $(\mathbf{v}_i)_{i=1,\ldots,n-1}$ is linearly independent and spans an $n-1$-dimensional subspace of $\mathbb{R}^d$. A smallest affine subspace containing $\mathcal{C}(X)$ is then given by this affine subspace translated by x_n. □

Chapter 3

The Volume Hypothesis

3.1 Introduction

It remains an open research question why even heavily overparameterized networks can generalize well in practice, even though their capacity would allow them to overfit the training data [7]. The question why common optimization algorithms reliably find well generalizing minima in the loss landscape remains a mostly open research question. It was long believed that this is due to implicit regularization introduced by first order optimization techniques like SGD [2, 3, 4], which bias the network towards good minima. However, [9] show using a zeroth order optimization technique – a Guess & Check Algorithm – that this regularization is not necessary to find well generalizing minima. By randomly sampling networks until finding one with 100% training accuracy, they show in the setting of binary classification that the majority of those networks still be simple and generalize well. This leads them to attribute the simplicity bias solely to the geometry of the loss landscape, independent of the first order regularization of SGD. They call their conjecture the **Volume Hypothesis**:

> "good hypotheses occupy larger volume in parameter space than bad hypotheses."

In this work we quantify the volume hypothesis by describing $\mathbb{P}(b_\theta | 100\% \text{ training accuracy})$, the number of linear segments in the decision boundary of network given that it achieves 100% training accuracy. Our results confirm that indeed simpler networks have larger posterior density, which explains the simplicity bias and thus the good performance of deep neural networks even in the overparameterized regime.

3.2 Mathematical Formulation

Taking the Bayesian perspective, we assume θ to be drawn i.i.d. from a probability distribution, f.e. $\theta \sim \mathcal{U}\left([0,1]^p\right)$. Given a dataset $D \subset \mathbb{R}^{n_0}$ of size d, let E_θ be the event that all training samples are classified correctly by f_θ.

In order to study the volume hypotheses, we are interested in the posterior density $\mathbb{P}(b_\theta | E_\theta = 1)$ of the linear pieces in the decision boundary given that it achieve 100% training accuracy. Using Bayes' rule, this density can be re-written as

$$\begin{aligned}\mathbb{P}(b_\theta | E_\theta = 1) &= \frac{\mathbb{P}(E_\theta = 1 | b_\theta)\mathbb{P}(b_\theta)}{\mathbb{P}(E_\theta)} \\ &\sim \mathbb{P}(b_\theta) I_{E_\theta}.\end{aligned}$$

In words, the posterior density is proportional to the volume of the network in parameter space (the function prior).

Using this notation, we conjecture the following behaviour of the number b_θ of linear pieces in the decision boundary:

Conjecture 3.2.1. The number b_θ of linear segments in the decision boundary of a random network f_θ satisfies

$$\mathbb{P}(b_\theta = n) \lesssim 2^{C \cdot n}, \tag{3.1}$$

where C is a constant independent of n.

Chapter 4

Affine Geometry

The following three chapters are devoted characterizing the decision boundary of a ReLU classification network. There are two ways to get to the result, once using affine geometry and the other time using tropical geometry. Both approaches are equivalent and one can switch from one to the other via a number of identifications.

In this chapter we will explain the approach using affine geometry. In the next chapter we will focus on tropical geometry and show how to switch from one perspective to the other.

Throughout this chapter, we fix an integer $d \in \mathbb{N}$.

4.1 Basic Definitions

We begin by defining a number of fundamental concepts in affine geometry. The most fundamental definition is that of an affine function.

Definition 4.1.1 (Affine Functions)**.** Given a vector $\mathbf{a} \in \mathbb{R}^d$ and a scalar $b \in \mathbb{R}$, we define the affine function with parameters $\mathbf{a}$ and b as

$$\begin{aligned} \mathrm{f}_{\mathbf{a},\mathrm{b}} \colon \mathbb{R}^d &\to \mathbb{R} \\ \mathbf{x} &\mapsto \langle \mathbf{a}, \mathbf{x} \rangle + b, \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product on $\mathbb{R}^d$.

Definition 4.1.2 (CPA Functions)**.** We say that a function $f \colon \mathbb{R}^d \to \mathbb{R}$ is cpa if it is convex and piece-wise affine. We denote by $\mathrm{CPA(d)}$ that space of cpa functions $\mathbb{R}^d \to \mathbb{R}$.

Definition 4.1.3 (DCPA Functions)**.** We say that a function $f \colon \mathbb{R}^d \to \mathbb{R}$ is dcpa if it can be written as the difference of two cpa functions. We denote by $\mathrm{DCPA(d)}$ the space of dcpa function $\mathbb{R}^d \to \mathbb{R}$.

Proposition 4.1.4 (Characterizing CPA Functions (Proposition 2 in [14]))**.** *Any function $F \colon \mathbb{R}^d \to \mathbb{R}$ of the form*

$$F(x) = \max\{f_1(x), \dots, f_n(x)\}$$

with affine functions $f_i \colon \mathbb{R}^d \to \mathbb{R}$ *is* cpa. *Also every CPA function with a finite number of linear pieces is of this form.*

4.2 Affine Dualities

In this section we mostly follow the argument introduced by [14]. It turns out that affine function $f \colon \mathbb{R}^d \to \mathbb{R}$ can be translated to a *dual space*. Studying this transformation helps us understand that a ReLU network can be understood as a dcpa function.

The graph of an affine function $\mathbb{R}^d \to \mathbb{R}$ defines a hyperplane in $\mathbb{R}^d \times \mathbb{R} = \mathbb{R}^{d+1}$. The latter space is of great importance in our work, leading us to the following definition:

Definition 4.2.1 (Real and Dual Space). We define *real space* $\mathfrak{R} := \mathbb{R}^{d+1}$ and *dual space* $\mathfrak{D} := \mathbb{R}^{d+1}$ to be two *distinct* copies of $\mathbb{R}^{d+1}$.

Definition 4.2.2 (Real and Dual Affine Space). For the space of all affine functions $\mathbb{R}^d \to \mathbb{R}$ with graph in $\mathfrak{R}$, we write

$$\mathrm{Aff}_{\mathfrak{R}}(\mathrm{d}) := \{\mathrm{f}_{\mathbf{a},\mathrm{b}} \,|(\mathbf{a}, b) \in \mathfrak{R}\}.$$

For the space of all affine functions $\mathbb{R}^d \to \mathbb{R}$ with graph in $\mathfrak{D}$, we write

$$\mathrm{Aff}_{\mathfrak{D}}(\mathrm{d}) := \{\mathrm{f}_{\mathbf{a},\mathrm{b}} \,|(\mathbf{a}, b) \in \mathfrak{D}\}.$$

The following lemma allows us to translate between affine and dual space (see Figure 4.1).

Lemma 4.2.3. *For any fixed dimension* d*, there exists a bijection between real affine space and dual space. It is given by*

$$\begin{aligned} \mathcal{R} \colon \ \mathfrak{D} &\to \mathrm{Aff}_{\mathfrak{R}}(\mathrm{d}) \\ (\mathbf{x}, y) &\mapsto f_{\mathbf{x},y} \end{aligned}$$

with inverse

$$\begin{aligned} \mathcal{R}^{-1} \colon \ \mathrm{Aff}_{\mathfrak{R}}(\mathrm{d}) &\to \mathfrak{D} \\ \mathrm{f}_{\mathbf{a},\mathrm{b}} &\mapsto (\mathbf{a}, b). \end{aligned}$$

The map $\mathcal{R}$ has the following properties.

Proposition 4.2.4. *Let* $(\mathbf{x}_i, y_i)_{i=1,\dots,n}$ *be a family of dual points.*

i) $\mathcal{R}$ *is a linear map, i.e. for any family of scalars* $\alpha_i \in \mathbb{R}$, $i = 1, \dots, n$,

$$\mathcal{R}(\sum_{i=1}^{n} \alpha_i(\mathbf{x}_i, y_i)) = \sum_{i=1}^{n} \alpha_i \mathcal{R}((\mathbf{x}_i, y_i)).$$

ii) the family of dual points is linearly independent if and only if the corresponding family of affine functions is linearly independent.

$$\begin{tikzcd}
\mathrm{Aff}_{\mathfrak{R}}(d) \arrow[r, "\mathrm{graph}"] \arrow[d, "\mathcal{R}"', "\wr"] & \mathfrak{R} \\
\mathfrak{D} & \mathrm{Aff}_{\mathfrak{D}}(d) \arrow[l, "\mathrm{graph}"] \arrow[u, "\wr", "\check{\mathcal{R}}"']
\end{tikzcd}$$

Figure 4.1: Diagram indicating the relationship between affine and dual space.

Proof. $i)$ can be confirmed by an easy calculation. $ii)$ follows from $i)$. □

Since both $\mathfrak{R}$ and $\mathfrak{D}$ are copies of $\mathbb{R}^{d+1}$, it seems natural to ask whether we can reverse the roles of $\mathfrak{R}$ and $\mathfrak{D}$ in the above constructions. The answer to this question is "yes", although we need to make some slight modifications. To do so, we start with the following definition:

Definition 4.2.5 (Dual Affine Space)**.** Analogous to real affine space, we define the *dual affine space* $\mathrm{Aff}_{\mathfrak{D}}(\mathrm{d})$ as the space of affine functions $\mathbb{R}^d \to \mathbb{R}$ with graph in $\mathfrak{D}$.

With this definition at hand, we can make a statement analogous to Lemma 4.2.3:

Lemma 4.2.6. *For any fixed dimension* d*, there exists a bijection between dual affine space and real space. It is given by*

$$\begin{aligned} \check{\mathcal{R}}\colon\ \mathrm{Aff}_{\mathfrak{D}}(\mathrm{d}) &\to \mathfrak{R} \\ f_{\mathbf{a},b} &\mapsto (-\mathbf{a}, b). \end{aligned}$$

with inverse

$$\begin{aligned} \check{\mathcal{R}}^{-1}\colon\ \mathfrak{R} &\to \mathrm{Aff}_{\mathfrak{D}}(\mathrm{d}) \\ (\mathbf{x}, y) &\mapsto f_{-\mathbf{x},y} \end{aligned}$$

Note that, compared to $\mathcal{R}$ and $\mathcal{R}^{-1}$, the functions $\check{\mathcal{R}}$ and $\check{\mathcal{R}}^{-1}$ have an additional minus. This is necessary for the duality results stated in the following Proposition.

Proposition 4.2.7 (Duality Properties (Proposition 7 in [14]))**.** *The maps $\mathcal{R}$ and $\check{\mathcal{R}}$ have the following properties (note that we are using notation from Definition 2.3.2):*

1. *A dual point $\mathbf{c} \in \mathfrak{D}$ lies on the graph of a dual affine function $\mathrm{f}_{\mathbf{a},\mathrm{b}} \in \mathrm{Aff}_{\mathfrak{D}}(\mathrm{d})$ if and only if the graph of the corresponding real affine function $\mathcal{R}(\mathbf{c})$ contains the corresponding real point $\check{\mathcal{R}}(\mathrm{f}_{\mathbf{a},\mathrm{b}})$:*
$$\mathbf{c} \in \mathrm{f}_{\mathbf{a},\mathrm{b}} \iff \check{\mathcal{R}}(\mathrm{f}_{\mathbf{a},\mathrm{b}}) \in \mathcal{R}(\mathbf{c})$$

2. *A dual point $\mathbf{c} \in \mathfrak{D}$ lies above the graph of a dual affine function $\mathrm{f}_{\mathbf{a},\mathrm{b}} \in \mathrm{Aff}_{\mathfrak{D}}(\mathrm{d})$ if and only if the real point $\check{\mathcal{R}}(\mathrm{f}_{\mathbf{a},\mathrm{b}})$ lies below the graph of $\mathcal{R}(c)$:*
$$\mathbf{c} \succ \mathrm{f}_{\mathbf{a},\mathrm{b}} \iff \mathcal{R}(\mathbf{c}) \succ \check{\mathcal{R}}(\mathrm{f}_{\mathbf{a},\mathrm{b}})$$

4.3 Neural Networks and Affine Geometry

It turns out that we can study ReLU networks in the context of affine geometry. To do so, we again start with some basic definitions.

Definition 4.3.1 (Minkowski Sum)**.** Given two sets of points $X, Y \subseteq \mathbb{R}^{d+1}$, we define

$$X + Y \coloneqq \{\mathbf{x} + \mathbf{y} \,|\, \mathbf{x} \in X, \mathbf{y} \in Y\}$$

to be the *Minkowski sum* of X and Y.

On vectors of sets of dual points, we define $+$ to act component-wise.

Definition 4.3.2. On the set $\mathcal{P}_{\text{fin}}(\mathfrak{D})$ of finite subsets of $\mathfrak{D}$, the function

$$\begin{aligned} \mathcal{Q}\colon \mathcal{P}_{\text{fin}}(\mathfrak{D}) &\to \mathrm{CPA(d)} \\ S &\mapsto \mathcal{Q}(S) \coloneqq \max_{\mathbf{s} \in S} \mathcal{R}(\mathbf{s}) \end{aligned}$$

assigns to a set of dual points the associated cpa function

$$\begin{aligned} \max_{\mathbf{s} \in S} \mathcal{R}(\mathbf{s})\colon \mathbb{R}^d &\to \mathbb{R} \\ \mathbf{x} &\mapsto \max_{(\mathbf{a}, b) \in S} \langle \mathbf{x}, \mathbf{a} \rangle + b \end{aligned}$$

We define $\mathcal{Q}(\emptyset) \coloneqq 0$.

For a *vector X of sets* of dual points, we define Q to act component-wise.

Remark 4.3.3. If follows directly from Propposition 4.1.4 that $\mathcal{Q}$ is a cpa function.

The map $\mathcal{Q}$ has the following properties:

Lemma 4.3.4 (Properties of $\mathcal{Q}$ (Proposition 12 in [14]))**.** *For any two sets of points $X, Y \subseteq \mathfrak{D}$ and every positive scalar $\alpha \geq 0$, we have*

i) $\mathcal{Q}(X \cup Y) = \max\{\mathcal{Q}(X), \mathcal{Q}(Y)\}$

ii) $\mathcal{Q}(X + Y) = \mathcal{Q}(X) + \mathcal{Q}(Y)$

iii) $\alpha \cdot \mathcal{Q} = \mathcal{Q}(a \cdot X)$*, where the last multiplication is the natural multiplication of a set with a real number.*

We furthermore need to define what it means to multiply a matrix with a vector of sets of points.

Definition 4.3.5. We define the multiplication of an $m \times n$ matrix A with a size-n vector X of sets of dual points in the following way:

$$\begin{aligned} \cdot\colon \mathbb{R}^{m,n} \times (P_{\text{fin}}(\mathfrak{D}))^n &\to (P_{\text{fin}}(\mathfrak{D}))^m \\ (\mathbf{A}, X) &\mapsto \mathbf{A} \cdot X \end{aligned}$$

where

$$(\mathbf{A} \cdot X)_i \coloneqq \sum_{j=1}^{n} \mathbf{A}_{ij} \cdot X_j.$$

In the notation above, $(P_{\text{fin}}(\mathfrak{D}))^n$ denotes the n-fold Cartesian product of $P_{\text{fin}}(\mathfrak{D})$ with itself and $\bigplus_{i=1}^{n}$ denotes the Minkowski-sum over all sets indexed with j.

Lemma 4.3.6 (Matrix Multiplication [14])**.** *Let $X \in \big(P_{\mathit{fin}}(\mathfrak{D})\big)^n$ be a vector of finite sets of dual points and $\mathbf{A} \in \mathbb{R}_{+}^{m,n}$ a matrix with non-negative entries. Then*

$$\mathbf{A}\mathcal{Q}(X) = \mathcal{Q}(\mathbf{A} \cdot X).$$

Proof. We understand the multiplication $\mathbf{A}Q(X)$ as a matrix-vector multiplication

$$[\mathbf{A}\mathcal{Q}(X)]_i = \sum_{j=1}^{n} \mathbf{A}_{ij}[\mathcal{Q}(X)]_j.$$

This allows us to deduce

$$\begin{aligned}
[\mathbf{A}\mathcal{Q}(X)]_i &= \sum_{j=1}^{n} \mathbf{A}_{ij}[\mathcal{Q}(X)]_j \\
&\overset{(*)}{=} \sum_{j=1}^{n} \mathcal{Q}(\mathbf{A}_{ij}X_j) \\
&\overset{(**)}{=} \mathcal{Q}(\bigplus_{j=1}^{n} \mathbf{A}_{ij}X_j) \\
&= \mathcal{Q}([\mathbf{A} \cdot X]_i) \\
&= [\mathcal{Q}(\mathbf{A} \cdot X)]_i.
\end{aligned}$$

In $(*)$ and $(**)$ we used Lemma 4.3.4 $iii)$ and $ii)$ respectively. □

In order to account for biases in our network (as an extension of the argument put forward by [14]), we introduce in the following what it means to add a scalar to a set of dual points.

Definition 4.3.7. We define the sum of a set of dual points and a scalar to act as follows:

$$\begin{aligned}
\boxplus \colon P_{\text{fin}}(\mathfrak{D}) \times \mathbb{R} &\to P_{\text{fin}}(\mathfrak{D}) \\
(X, \alpha) &\mapsto X \boxplus \alpha,
\end{aligned}$$

where $X \boxplus \alpha$ is the set points

$$X \boxplus \alpha := \{(\mathbf{x}, y + \alpha) \,|(\mathbf{x}, y) \in X\}.$$

It turns out that $\mathcal{Q}$ is well behaved with respect to $\boxplus$:

Lemma 4.3.8. *For any finite set of dual points $X \subseteq \mathfrak{D}$ and scalar $\alpha \in \mathbb{R}$, it holds that*

$$\mathcal{Q}(X) + \alpha = \mathcal{Q}(X \boxplus \alpha).$$

Proof. For every $\mathbf{z} \in \mathbb{R}^d$, it holds that

$$\begin{aligned}(Q(X)+\alpha)(\mathbf{z}) &= \mathcal{Q}(X)(\mathbf{z})+\alpha \\ &= \max_{(\mathbf{a},b)\in X} \mathrm{f}_{\mathbf{a},\mathrm{b}}(\mathbf{z})+\alpha \\ &= \max_{(\mathbf{a},b)\in X} \langle \mathbf{a},\mathbf{z}\rangle + b + \alpha \\ &= \max_{(\mathbf{a},b)\in X\boxplus\alpha} \langle \mathbf{a},\mathbf{z}\rangle + b \\ &= \mathcal{Q}(X\boxplus\alpha)(\mathbf{z}).\end{aligned}$$

□

With this machinery at hand, we can deduce a fundamental proposition on the connection between neural networks and affine duality. We will start with a quick reminder. The Details can be found in Definition 2.1.1.

Proposition 4.3.9 (NNs as DCPAs (Proposition 16 in [14])). *Assume that a neural network in the sense of Definition 2.1.1 can be written up to layer $l-1$ as a* dcpa *function $\mathbf{a}_{l-1} = \mathcal{Q}(P_{l-1}) - \mathcal{Q}(N_{l-1})$ for some vectors of sets of dual points P_{l-1}, N_{l-1}. Then, after writing $\mathbf{W}_l = \mathbf{W}_l^+ - \mathbf{W}_l^-$ using matrices $\mathbf{W}_l^+$ and $\mathbf{W}_l^-$ with non-negative entries, also the network up to the l'th layer can be written as a* dcpa *function*

$$\mathbf{a}_l = \mathcal{Q}(P_l) - \mathcal{Q}(N_l)$$

with

$$\begin{aligned} N_l &= (\mathbf{W}_l^- \cdot P_{l-1}) + (\mathbf{W}_l^+ \cdot N_{l-1}) \\ P_l &= \big((\mathbf{W}_l^+ \cdot P_{l-1}) + (\mathbf{W}_l^- \cdot N_{l-1}) \boxplus \mathbf{b}_l\big) \cup \begin{cases} N_l \boxplus t_l, & t_l \neq -\infty \\ \emptyset, & t_l = -\infty. \end{cases}\end{aligned}$$

(The operations $+$ and $\cup$ are to be understood coordinate-wise.)

Proof. First, note that

$$\begin{aligned}\mathbf{W}_l a_{l-1} &= \left(\mathbf{W}_l^+ - \mathbf{W}_l^-\right)\left(\mathcal{Q}(P_{l-1}) - \mathcal{Q}(N_{l-1})\right) \\ &= (\mathbf{W}_l^+\mathcal{Q}(P_{l-1}) + \mathbf{W}_l^-\mathcal{Q}(N_{l-1})) - (\mathbf{W}_l^-\mathcal{Q}(P_{l-1}) + \mathbf{W}_l^+\mathcal{Q}(N_{l-1})) \\ &\overset{(*)}{=} \mathcal{Q}\left((\mathbf{W}_l^+ \cdot P_{l-1}) + (\mathbf{W}_l^- \cdot N_{l-1})\right) - \mathcal{Q}((\mathbf{W}_l^- \cdot P_{l-1}) + (\mathbf{W}_l^+ \cdot N_{l-1})) \\ &= \mathcal{Q}\left((\mathbf{W}_l^+ \cdot P_{l-1}) + (\mathbf{W}_l^- \cdot N_{l-1})\right) - \mathcal{Q}(N_l),\end{aligned}$$

where in $(*)$ we used Lemma 4.3.4 $ii)$ and Lemma 4.3.6.

First, assume that $t_l \neq -\infty$. Using the identity $\max\{x-y,t\} = \max\{x, y+t\} - y$ and the above reformulation of $\mathbf{W}_l a_{l-1}$, we can write, using the definitions of N_l and P_f of the statement,

$$
\begin{aligned}
\mathbf{a}_l &= \rho(\mathbf{W}_l a_{l-1} + \mathbf{b}_l) \\
&= \max\{\mathcal{Q}\left(\left(\mathbf{W}_l^+ \cdot P_{l-1}\right) + \left(\mathbf{W}_l^- \cdot N_{l-1}\right)\right) - \mathcal{Q}\left(N_l\right) + \mathbf{b}_l, t_l\} \\
&\overset{(**)}{=} \max\{\mathcal{Q}\left(\left(\left(\mathbf{W}_l^+ \cdot P_{l-1}\right) + \left(\mathbf{W}_l^- \cdot N_{l-1}\right)\right) \boxplus \mathbf{b}_l\right) - \mathcal{Q}(N_l), t_l\} \\
&= \max\{\mathcal{Q}\left(\left(\left(\mathbf{W}_l^+ \cdot P_{l-1}\right) + \left(\mathbf{W}_l^- \cdot N_{l-1}\right)\right) \boxplus \mathbf{b}_l\right), \mathcal{Q}(N_l) + t_l\} - \mathcal{Q}(N_l) \\
&\overset{(***)}{=} \mathcal{Q}\left(\left(\left(\left(\mathbf{W}_l^+ \cdot P_{l-1}\right) + \left(\mathbf{W}_l^- \cdot N_{l-1}\right)\right) \boxplus \mathbf{b}_l\right) \cup (N_l \boxplus t_l)\right) - \mathcal{Q}(N_l) \\
&= \mathcal{Q}(P_l) - \mathcal{Q}(N_l).
\end{aligned}
$$

In $(**)$ we used a vectorized version of Lemma 4.3.8 and in $(***)$ we used Lemma 4.3.4 $i)$ and Lemma 4.3.8.

Now, assume $t_l = -\infty$. Then

$$
\begin{aligned}
\mathbf{a}_l &= \rho(\mathbf{W}_l a_{l-1} + \mathbf{b}_l) \\
&= \max\{\mathcal{Q}\left(\left(\mathbf{W}_l^+ \cdot P_{l-1}\right) + \left(\mathbf{W}_l^- \cdot N_{l-1}\right)\right) - \mathcal{Q}\left(N_l\right) + \mathbf{b}_l, -\infty\} \\
&= \mathcal{Q}\left(\left(\left(\mathbf{W}_l^+ \cdot P_{l-1}\right) + \left(\mathbf{W}_l^- \cdot N_{l-1}\right)\right) \boxplus \mathbf{b}_l\right) - \mathcal{Q}(N_l) \\
&= \mathcal{Q}(P_l) - \mathcal{Q}(N_l).
\end{aligned}
$$

This finishes the proof. □

Corollary 4.3.10. *Every ReLU binary classification network f_θ in the sense of Definition 2.1.3 can be written as a $DCPA$ function*

$$f_\theta = \mathcal{Q}(P) - \mathcal{Q}(N)$$

for some sets of dual points $P, N \subseteq \mathfrak{D}$.

Proof. We use induction on the number of layers L of the network. For $L = 1$, the network just consists of the input-function $f_1 = \mathrm{id}_{\mathbb{R}^{n_0}}$, whose i'th coordinate function can be represented as

$$
\begin{aligned}
f_{1,i}(x) &= \mathbf{x} \cdot \mathbf{e}_i + 0 \\
&= \mathcal{Q}(\{(\mathbf{e}_i, 0)\}) - \mathcal{Q}(\emptyset) \\
&= \mathcal{Q}(\{(\mathbf{e}_i, 0)\}),
\end{aligned}
$$

where $\mathbf{e}_i \in \mathbb{R}^{n_0}$ is the i'th unit vector with entries $(\mathbf{e}_i)_j = \delta_{ij}$. Hence, we can write

$$f_1 = \mathcal{Q}\left((\{(\mathbf{e}_1, 0)\}, \ldots, \{(\mathbf{e}_{n_0}, 0)\})\right).$$

We may thus choose $P_1 = (\{(\mathbf{e}_1, 0)\}, \ldots, \{(\mathbf{e}_{n_0}, 0)\})$ and $N_1 = (\emptyset)$. This shows the claim for $L = 1$. For the induction step, use Proposition 4.3.9. Finally, since the output-dimension is 1, the final vectors of sets of dual points can be identified with just sets of dual points. □

4.3.1 Understanding The Representation

Corollary 4.3.10 shows that any neural network $f\colon \mathbb{R}^d \to \mathbb{R}$ in the sense of Definition 2.1.3 can be written as a dcpa function $f = \mathcal{Q}(P) - \mathcal{Q}(N)$ for two sets of dual points $P, N \subset \mathfrak{D}$. In this subsection we use the recursive formulation in Proposition 4.3.9 work towards a better understanding of the two sets P and Q. In particular, we are interested in the size of both P and Q. We start with some basic definitions.

Definition 4.3.11. Given a vector of finite sets of dual points $X \in \mathcal{P}_{\text{fin}}(\mathfrak{D})^n$, we define $s(X) := \max_{i=1,\ldots,n} |X_i|$ to be the size of the maximum set in X.

Lemma 4.3.12. *Let $\mathbf{A} \in \mathbb{R}^{m,n}$ be a matrix, $b \in \mathbb{R}$ a scalar and $X \in \left(\mathcal{P}_{\mathit{fin}}(\mathfrak{D}) \setminus \emptyset\right)^n$, $Y \in \left(\mathcal{P}_{\mathit{fin}}(\mathfrak{D}) \setminus \emptyset\right)^k$ be vectors of finite non-empty sets of dual points. Then the following are true:*

i) For any $1 \le i \le m$,

$$|(\mathbf{A} \cdot X)_i| \le \prod_{j=1}^{n} |X_j|.$$

In particular $\mathbf{A} \cdot X$ has uniform size. If this was already true for X, then

$$s(\mathbf{A} \cdot X) \le s(X)^n.$$

ii)

$$|X_i + Y_j| \le |X_i||Y_j|, \quad 1 \le i \le n,\ 1 \le j \le k.$$

iii)

$$|X_i \boxplus b| = |X_i|, \quad 1 \le i \le n.$$

Proof. $ii)$ and $iii)$ are clear. For $i)$ we compute

$$\begin{aligned} |(\mathbf{A} \cdot X)_i| &= |\mathop{\Large +}_{j=1}^{n} \mathbf{A}_{ij} X_j| \\ &\overset{ii)}{\le} \prod_{j=1}^{n} |\mathbf{A}_{ij} X_j| \\ &= \prod_{j=1}^{n} |X_j| \\ &\le s(X)^n. \end{aligned}$$

This shows the claim. □

Lemma 4.3.13. *In the setting of Proposition 4.3.9, let $\xi_l := s(P_l)s(N_l)$ for $l = 2, \ldots, L$. Then, $s(N_l)$ and $s(P_l)$ can be upper-bounded in terms of ξ_{l-1} via*

$$s(N_l) \le \xi_{l-1}^{n_{l-1}} \tag{4.1}$$

and

$$s(P_l) \leq \begin{cases} 2\xi_{l-1}^{n_{l-1}}, & t_k \neq -\infty \\ \xi_{l-1}^{n_{l-1}}, & t_k = -\infty. \end{cases} \tag{4.2}$$

for $l = 3, \ldots, L$. Furthermore, ξ_l satisfies the recursive relation

$$\xi_l \leq \begin{cases} 2\xi_{l-1}^{2n_{l-1}}, & t_k \neq -\infty \\ \xi_{l-1}^{2n_{l-1}}, & t_k = -\infty \end{cases} \tag{4.3}$$

for $l = 3, \ldots, L$.

Proof. We start by showing Equation (4.1). Fix $l \in \{3, \ldots, L\}$. Then

$$\begin{aligned} |(N_l)_i| &= |(\mathbf{W}_l^- \cdot P_{l-1})_i + (\mathbf{W}_l^+ \cdot N_{l-1})_i| \\ &\overset{4.3.12}{\leq} |(\mathbf{W}_l^- \cdot P_{l-1})_i||(\mathbf{W}_l^+ \cdot N_{l-1})_i| \\ &\overset{4.3.12}{\leq} s(P_{l-1})^{n_{l-1}} s(N_{l-1})^{n_{l-1}} \\ &= \xi_{l-1}^{n_{l-1}} \end{aligned}$$

This shows Equation (4.1).

Similarly, one can show that

$$s(P_{l-1}) \leq s(P_{l-1})^{n_{l-1}} s(N_{l-1})^{n_{l-1}} + \begin{cases} s(P_{l-1})^{n_{l-1}} s(N_{l-1})^{n_{l-1}}, & t_{l-1} \neq \infty \\ 0, & t_{l-1} = \infty. \end{cases}$$

which shows Equation (4.2). Finally, Equation (4.3) follows from the other two.. □

We finally resolve the recursion to find a close-form expression for $s(N_l)$.

Proposition 4.3.14. *Consider the setting of Proposition 4.3.9 with initial definitions of N_1 and P_1 as in Corollary 4.3.10. Then, if the threshold $t_l \neq -\infty$ for all $l \leq k$, the product ξ_k is upper bounded by*

$$\xi_k \leq 2^{1+\sum_{j=2}^{k-1} \prod_{i=j}^{k-1} 2n_j}, \quad k = 3, \ldots, L. \tag{4.4}$$

(For completeness: $\xi_2 \leq 2$.)

Proof. We prove the claim by induction on k.

First note that $N_1 = (\emptyset)$ and $P_1 = (\{(\mathbf{e}_1, 0)\}, \ldots, \{(\mathbf{e}_d, 0)\})$, which immediately implies that $s(P_1) = 1$. It follows directly from the recursion in Proposition 4.3.9 and Lemma 4.3.12.i) that

$$\begin{aligned} s(N_2) &= s(\mathbf{W}_2^+ P_1) \\ &\leq 1^d \\ &\leq 1 \end{aligned}$$

and

$$s(P_2) \leq s(P_1) + s(N_2) \leq 2$$

which implies

$$\xi_2 = s(P_2)s(N_2) \leq 2. \tag{4.5}$$

For $k = 3$, we can use Lemma 4.3.13 and Equation (4.5) to write

$$\begin{aligned} \xi_3 &\leq 2\xi_2^{2n_2} \\ &\leq 2^{1+2n_2}, \end{aligned}$$

which starts the induction. Next, assume Equation (4.4) holds for any layer $3 \leq l < k$. We show that it also holds for $l = k$. Indeed,

$$\begin{aligned} \xi_k &\leq 2\xi_{k-1}^{2n_{k-1}} \\ &\overset{IH}{\leq} 2\left(2^{1+\sum_{j=2}^{k-2}\prod_{i=j}^{k-2} 2n_j}\right)^{2n_{k-1}} \\ &= 2 \cdot 2^{2n_{k-1}+\sum_{j=2}^{k-2}\prod_{i=j}^{k-1} 2n_j} \\ &= 2^{1+\sum_{j=2}^{k-1}\prod_{i=j}^{k-1} 2n_j}. \end{aligned}$$

This finishes the proof. □

In the special case of binary classification, N_L and P_L are vectors containing just one set of dual points each. Thus, we can identify them with sets. The following Corollary describes their size.

Corollary 4.3.15. *Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be a binary classification network in the sense of Definition 2.1.3 with $L \geq 3$ layers. Then we can write*

$$f = \mathcal{Q}(P) - \mathcal{Q}(N)$$

where $P, Q \subset \mathfrak{D}$ are finite sets of dual points whose size is upper bounded by

$$|P|, |N| \leq 2^{n_L\left(1+\sum_{j=2}^{L-2}\prod_{i=j}^{L-2} 2n_j\right)}.$$

Proof. Since $t_L = -\infty$ (i.e., the last layer is linear), Lemma 4.3.13 implies that

$$|P|, |N| \leq \xi_{L-1}^{n_{L-1}}.$$

The claim then follows from Proposition 4.3.14 since $t_l = 0$ for all $l < L$. □

In the special case where all layers have the same width, the following special case holds:

Corollary 4.3.16. *Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be a binary classification network in the sense of Definition 2.1.3 with $L \geq 3$ layers of width $n_l = w \in \mathbb{N}$ for all $l = 1, \ldots, L$. Then we can write*

$$f = \mathcal{Q}(P) - \mathcal{Q}(N)$$

where $P, Q \subset \mathfrak{D}$ are finite sets of dual points whose size is upper bounded by

$$|P|, |N| \leq 2^{w\frac{1-(2w)^{L-2}}{1-2w}}.$$

Proof. The bound follows from Corollary 4.3.15 if we can show that

$$w\left(1+\sum_{j=2}^{L-2}\prod_{i=j}^{L-2}2w\right)=w\frac{1-(2w)^{L-2}}{1-2w}.$$

Indeed,

$$\begin{aligned}
w\left(1+\sum_{j=2}^{L-2}\prod_{i=j}^{L-2}2w\right) &= w\left(1+\sum_{j=2}^{L-2}(2w)^{L-2-j+1}\right)\\
&\overset{(*)}{=} w\left(1+\sum_{j=1}^{L-3}(2w)^{j}\right)\\
&= w\sum_{j=0}^{L-3}(2w)^{j}\\
&= w\frac{1-(2w)^{L-2}}{1-2w}.
\end{aligned}$$

$(*)$ follows from inserting the upper and lower bound. The last equality follows from identifying the geometric sum. □

Need a probability space etc. to make more precise. But roughly I want to say the following:

In practice, the weights $\mathbf{W}_l$ are often initialized *i.i.d.* following a Gaussian distribution. Since the sets P_l and N_l are finite, we conjecture that

$$|(\mathbf{W}_l^- \cdot P_{l-1})_i + (\mathbf{W}_l^+ \cdot N_{l-1})_i| \overset{a.s.}{=} |(\mathbf{W}_l^- \cdot P_{l-1})_i||(\mathbf{W}_l^+ \cdot N_{l-1})_i|$$

Similarly,

$$\begin{aligned}
&|\left(((\mathbf{W}_l^+ \cdot P_{l-1}) + (\mathbf{W}_l^- \cdot N_{l-1}))\boxplus \mathbf{b}_l\right)_i \cup (N_l \boxplus t_l)_i\,| =\\
&\qquad = |\left(((\mathbf{W}_l^+ \cdot P_{l-1}) + (\mathbf{W}_l^- \cdot N_{l-1}))\boxplus \mathbf{b}_l\right)_i| \cup |(N_l \boxplus t_l)_i\,|
\end{aligned}$$

Thus, we can replace all of the "$\leq$" in the calculations of this subsection by $\overset{a.s.}{=}$:

Conjecture 4.3.17. Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be a binary classification network in the sense of Definition 2.1.3 with $L \geq 3$ layers. Then we can write

$$f = \mathcal{Q}(P) - \mathcal{Q}(N)$$

where $P, Q \subset \mathfrak{D}$ are finite sets of dual points whose size is given by

$$|P| = |N| \overset{a.s.}{=} 2^{n_L\left(1+\sum_{j=2}^{L-2}\prod_{i=j}^{L-2}2n_j\right)}.$$

Chapter 5

Tropical Geometry

It turns out that all of the constructions in Chapter 4 can be translated to a branch of algebraic geometry called *Tropical Geometry*. The following discussion is more abstract than the one using affine functions. We nevertheless provide it here since it provides valuable insights. Note that this chapter should be seen as a new perspective to affine geometry. Besides Corollary 5.3.5, it does not provide a lot of deep revelations. However, it allows to indirectly equip the space of affine functions (or rather an extension using an element called $-\infty$ thereof) with the structure of a semi-ring). In this chapter we largely follow [15] and [13].

5.1 Basic Definitions

5.1.1 Concepts from Tropical Algebra

Tropical Geometry takes place in the *tropical semiring*.

Definition 5.1.1. The *tropical semiring* consists of the set $\mathbb{T} := \mathbb{R} \cup \{-\infty\}$ together with the operations $\oplus$ and $\odot$, where $\oplus$ is called *tropical addition* and $\odot$ is called *tropical multiplication*. These operations are defined as

$$\begin{aligned} \oplus \colon & \mathbb{T} \times \mathbb{T} \to \mathbb{T} \\ & (x, y) \mapsto x \oplus y := \max\{x, y\} \end{aligned}$$

$$\begin{aligned} \odot \colon & \mathbb{T} \times \mathbb{T} \to \mathbb{T} \\ & (x, y) \mapsto x \odot y := x + y \end{aligned}$$

where $+$ is the usual addition on $\mathbb{R}$ and

$$\begin{aligned} -\infty \oplus x &:= x \\ -\infty \odot x &:= -\infty. \end{aligned}$$

We furthermore define the *tropical quotient* as $x \oslash y := x - y$.

Remark 5.1.2. As the name suggests, $(\mathbb{T}, \oplus, \odot)$ is a semi-ring. In the following we repeat the defining properties for the sake of completeness:

i) $(\mathbb{R}, \odot)$ is a monoid under tropical multiplication, i.e. $\odot$ is associative and has a multiplicative identity: 0

ii) $(\mathbb{T}, \oplus)$ is an abelian groups *except for the existence of a tropical additive inverse*, i.e. $\oplus$ is associative, commutative and has an additive identity: $-\infty$

iii) tropical multiplication is distributive with respect to tropical addition.

As a next step we make sense of tropical powers.

Definition 5.1.3. Given $x \in \mathbb{T}$ and $a \in \mathbb{N}$, we can raise x to the power of a in the following way:

$$x^{\odot a} := \begin{cases} x \odot ... \odot x = a \cdot x & x \in \mathbb{R} \\ -\infty & x = -\infty \text{ and } a > 0 \\ 0 & x = -\infty \text{ and } a = 0 \end{cases}.$$

Similar to the real case, we can define polynomials over $\mathbb{T}$:

Definition 5.1.4 (Tropical Monomials)**.** A *tropical monomial* in d variables is an expression of the form

$$b \odot x_1^{\odot a_1} \odot ... \odot x_d^{\odot a_d}$$

where $b \in \mathbb{T}$ and $a_1, ..., a_d \in \mathbb{N}$. As a shorthand we will use multiindex notation, writing $b \odot x^{\odot \alpha}$ where $\alpha \in \mathbb{N}^d$. We denote the space of tropical monomials in d variables by $\mathbb{T}\{x_1, \dots, x_d\}$.

Definition 5.1.5 (Tropical Polynomials)**.** A *tropical polynomial* $f(x) = f(x_1, ..., x_d)$ is a finite tropical sum of tropical monomials

$$f(x) = b_1 \odot x^{\odot \alpha_1} \oplus ... \oplus b_r \odot x^{\odot \alpha_r}$$

where $\alpha_i \in \mathbb{N}^d$ and $b_i \in \mathbb{T}$ for all $i = 1, ..., r$. We will assume that $\alpha_i \neq \alpha_j$ for $i \neq j$, i.e. that the polynomial is in some sense reduced maximally. We denote the space of tropical polynomials in d variables by $\mathbb{T}[x_1, \dots, x_d]$

A notion which will appear often in the context of neural networks is that of a *tropical rational function*:

Definition 5.1.6 (Tropical Rational Functions)**.** A *tropical rational function* is the tropical quotient of two tropical polynomials $f(x)$ and $g(x)$:

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

We will denote the space of tropical rational functions in d variables by $\mathbb{T}(x_1, \dots, x_d)$.

Lemma 5.1.7. *A d-variate tropical polynomial $f(x)$ defines a function $f \colon \mathbb{R}^d \to \mathbb{R}$ that is convex.*

Proof. Taking the max of and summing over convex functions is convex. □

Definition 5.1.8 (Higher Dimensions). A function $F\colon \mathbb{R}^d \to \mathbb{R}^p$, $x = (x_1, \dots, x_d) \mapsto (f_1(x), \dots, f_p(x))$ is called a *tropical polynomial map* if each $f_i\colon \mathbb{R}^d \to \mathbb{R}$ is a tropical polynomial. It is called a *tropical rational map* if each f_i is a tropical rational function.

We will denote the set of tropical polynomial maps by $\mathrm{Pol}(d, p)$ and the set of tropical rational maps by $\mathrm{Rat}(d, p)$.

5.1.2 Signomials

So far we have only discussed concepts required to study tropical algebra. In the following we introduce concepts that go a bit further but allow us to draw closer parallels to affine geometry.

We extend Definition 5.1.3 in the obvious way to make sense of what it means to raise an expression to a real-valued tropical power, i.e. $x^{\odot\alpha}$ for $\alpha \in \mathbb{R}$. This allows us to make the following definition:

Definition 5.1.9 (Tropical Simple Signomial). A *tropical simple signomial* in d variables is an expression of the form

$$b \odot x_1^{\odot a_1} \odot \dots \odot x_d^{\odot a_d}$$

where $b \in \mathbb{T}$ and $a_1, \dots, a_d \in \mathbb{R}$. In particular, the exponent is real and not a natural number as for tropical monomials. As a shorthand we will use multiindex notation, writing $b \odot x^{\odot\alpha}$ with $\alpha \in \mathbb{R}^d$. We denote the space of tropical simple signomials by $\mathbb{T}_{\mathbb{R}}\{x_1, \dots, x_d\}$.

Definition 5.1.10 (Tropical Signomial). A *tropical signomial* $\varphi(x) = \varphi(x_1, ..., x_d)$ is a finite tropical sum of tropical monomials

$$\varphi(x) = b_1 \odot x^{\odot\alpha_1} \oplus ... \oplus b_r \odot x^{\odot\alpha_r}$$

where $\alpha_i \in \mathbb{R}^d$ and $b_i \in \mathbb{T}$ for all $i = 1, ..., r$. We denote the space of tropical signomials in d variables by $\mathbb{T}_{\mathbb{R}}[x_1, \dots, x_d]$.

Definition 5.1.11 (Tropical Rational Signomial). A *tropical rational signomial* is the tropical quotient of two tropical signomials $\varphi(x)$ and $\psi(x)$:

$$\varphi(x) \oslash \psi(x) = \varphi(x) - \psi(x).$$

We denote the space of tropical rational signomials in d variables by $\mathbb{T}_{\mathbb{R}}(x_1, \dots, x_d)$

Definition 5.1.12 (Tropical Signomial Map). A *tropical rational signomial map* is a function $\mathbb{R}^d \to \mathbb{R}^p$ which is a tropical rational sigmoidal in every coordinate. We denote the space of tropical signomial maps by $\mathrm{Sig}(d, p)$.

5.2 Relation to Basic Concepts from Affine Geometry

There is a close relation between tropical and affine geometry which we show in this secion.

Definition 5.2.1. We denote by

$$\mathbb{T}_{\mathbb{R}}^{\text{fin}}\{x_1,\dots,x_d\} \coloneqq \{b \odot x^{\odot\alpha} \in \mathbb{T}_{\mathbb{R}}\{x_1,\dots,x_d\} \mid b \in \mathbb{R}\}$$

the space of *tropical simple signomials* with finite coefficients $b \in \mathbb{R}$. Analogously we denote by $\mathbb{T}_{\mathbb{R}}^{\text{fin}}[x_1,\dots,x_d]$ the space of tropical sums of tropical simple signomials, called *tropical signomials*. We furthermore denote by $\mathbb{T}_{\mathbb{R}}^{\text{fin}}(x_1,\dots,x_d)$ the space of tropical quotients of tropical signomials, called *tropical rational signomials*.

Note that in general $\mathbb{T}_{\mathbb{R}}^{\text{fin}}\{x_1,\dots,x_d\} \subsetneq \mathbb{T}_{\mathbb{R}}\{x_1,\dots,x_d\}$, since the space on the right hand side also allows for coefficients $b = -\infty$.

This allows us to make the following identifications:

Proposition 5.2.2. *The following maps are bijections:*

1. *affine functions can be identified with tropical monomials,*

$$\begin{aligned} \mathrm{Aff} &\to \mathbb{T}_{\mathbb{R}}^{\text{fin}}\{x_1,\dots,x_d\} \\ \mathrm{f}_{\mathbf{a},\mathrm{b}} &\mapsto b \odot x^{\odot\mathbf{a}} \end{aligned}$$

2. cpa *functions can be identified with tropical signomials,*

$$\begin{aligned} \mathrm{CPA(d)} &\to \mathbb{T}_{\mathbb{R}}^{\text{fin}}[x_1,\dots,x_n] \\ \max_{i=1,\dots,n} f_{\mathbf{a}_i,b_i} &\mapsto b_1 \odot x^{\odot\mathbf{a}_1} \oplus \ldots \oplus b_n \odot x^{\odot\mathbf{a}_n} \end{aligned}$$

3. dcpa *functions can be identified with tropical rational signomials,*

$$\begin{aligned} \mathrm{DCPA(d)} &\to \mathbb{T}_{\mathbb{R}}^{\text{fin}}(x_1,\dots,x_n) \\ \max_{i=1,\dots,n} f_{\mathbf{a}_i,b_i} - \max_{i=1,\dots,m} f_{\mathbf{c}_i,d_i} &\mapsto \bigoplus_{i=1}^{n} b_i \odot x^{\odot\mathbf{a}_i} - \bigoplus_{i=1}^{m} d_i \odot x^{\odot\mathbf{c}_i}. \end{aligned}$$

5.3 Neural Networks and Tropical Geometry

In the previous section we saw how we can identify concepts from tropical algebra with affine algebra. Consequently, the results from Chapter 4, concerning f.e. affine hypersurfaces, affine duality and the identification of ReLU networks with dcpa functions, can be re-formulated in the realm of affine geometry.

We will re-formulate some known concepts in the language of affine geometry. The proof of most of the statements can be copied from Chapter 4 after considering the identifications in Section 5.2.

Proposition 5.3.1 (NN as Tropical Signomial Rational Functions (Proposition 5.1 in [15])). *Assume that a neural network in the sense of Definition 2.1.1 can be written up to layer $l-1$ as a tropical rational signomial map $\mathbf{a}_l(\mathbf{x}) = F_l(\mathbf{x}) \oslash G_l(\mathbf{x})$ for some tropical signomial maps F_l and G_l*[1]*. Then, after writing $\mathbf{W}_l = \mathbf{W}_l^+ - \mathbf{W}_l^-$ using matrices $\mathbf{W}_l^+$ and*

[1] Remember that think of $\mathbf{a}_l$ as a function of the network input $\mathbf{x}$

$\mathbf{W}_l^-$ *with non-negative entries, also the network up to l'th layer can be written as a tropical rational signomial map*

$$\mathbf{a}_{l+1} = F_{l+1} \oslash G_{l+1}.$$

The tropical signomial maps are given by

$$\begin{aligned} G_{l+1} &= \mathbf{W}_{l+1}^+ G_l + \mathbf{W}_{l+1}^- F_l \\ F_{l+1} &= \max\{\mathbf{W}_{l+1}^+ F_l + \mathbf{W}_{l+1}^- G_l + b, G_{l+1} + t\}. \end{aligned}$$

Writing $f_i^{(l)}$ and $g_i^{(l)}$ for the ith coordinate of F_l and G_l, the recurrence takes the form

$$\begin{aligned} g_i^{(l+1)} &= \left[\bigodot_{j=1}^{n_l} \left(f_j^{(l)}\right)^{\odot w_{ij}^-}\right] \odot \left[\left(g_j^{(l)}\right)^{\odot w_{ij}^+}\right] \\ f_i^{(l+1)} &= \left\{\left[\bigodot_{j=1}^{n_l} \left(f_j^{(l)}\right)^{\odot w_{ij}^+}\right] \odot \left[\left(g_j^{(l)}\right)^{\odot w_{ij}^-}\right] \odot b_i\right\} \oplus \left(g_i^{(l+1)} \odot t_i\right) \end{aligned}$$

where we write $(\mathbf{W}_{l+1}^+)_{ij} = w_{ij}^+$ and analogously for $\mathbf{W}_{l+1}^-$.

Proof. One can quickly check that this recursion corresponds to the recursion in Proposition 4.3.9 after identifying

$$\begin{aligned} F_l &= \mathcal{Q}(P_{l-1}) \\ G_l &= \mathcal{Q}(N_{l-1}). \end{aligned}$$

□

Theorem 5.3.2 (Tropical Characterization of Neural Networks). *A ReLU binary classification network in the sense of Definition 2.1.3 is a tropical rational signomial map of its input. That is,*

$$f_\theta(x) = F(x) \oslash G(x)$$

where F and G are tropical signomial maps.

Corollary 5.3.3. *A ReLU classification network in the sense of Definition 2.1.3 can be written a tropical rational signomial function.*

One can show an even stronger result, namely that we have already found all binary classification neural networks once we understand tropical rational signomial maps:

Theorem 5.3.4 (Tropical Equivalence (Theorem 5.4.i in [15])). *Let $f \colon \mathbb{R}^d \to \mathbb{R}$. Then f is a tropical rational signomial with finite coefficients if and only if it is a neural network in the sense of Definition 2.1.1.*

Proof. The "if" follows from Corollary 5.3.3. We thus only need to show the "only if". In this proof we identify finite coefficient tropical signomials with CPA functions in the sense of Proposition 5.2.2.

We first claim that every tropical signomial $\bigoplus_{i=1}^{n} b_i \odot x^{\odot \alpha_i} \in \mathbb{T}_{\mathbb{R}}^{\text{fin}}[x_1, \dots, x_d]$ with n tropical summands can be written as a neural network with n layers, i.e.

$$\bigoplus_{i=1}^{n} b_i \odot x^{\odot \alpha_i} = \rho_{-\infty} \circ f_n \circ \rho_0 \circ \ldots \circ \rho_0 \circ f_1(\mathbf{x}) \tag{5.1}$$

with affine functions $f_l \colon \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$.

We show Equation (5.1) by induction on n. The base-case follows readily:

$$\begin{aligned} b_1 x^{\odot \alpha_1} &= \langle \alpha_1, \mathbf{x} \rangle + b_1 \\ &= \rho_{-\infty} \left(\langle \alpha_1, \mathbf{x} \rangle + b_1 \right) \\ &= \rho_{-\infty} \circ f_1(\mathbf{x}) \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product on $\mathbb{R}^d$ and $f_1 = f_{\alpha_1, b}$.

Assume that Equation (5.1) holds for all finite coefficient tropical signomials with less than n tropical summands. We now show that in this case it also holds for n tropical summands.

Let $\bigoplus_{i=1}^{n} b_i \odot x^{\odot \alpha_i} \in \mathbb{T}_{\mathbb{R}}^{\text{fin}}[x_1, \dots, x_d]$ be a finite coefficient tropical signomial n tropical summands.

By the induction hypothesis we can find two ReLU network representations

$$\begin{aligned} p(\mathbf{x}) &= \bigoplus_{i=1}^{n-1} b_i \odot x^{\odot \alpha_i} \\ q(\mathbf{x}) &= b_n \odot x^{\odot \alpha_n}. \end{aligned}$$

Define $y(\mathbf{x}) := (p(\mathbf{x}), q(\mathbf{x}))$. Then y can also be expressed by a $(n-1)$-layer ReLU network by extending $q(\mathbf{x})$ using linear layers and isolating the subnetworks using zero-weights. Write

$$y(\mathbf{x}) = \rho_{-\infty} \circ h_{n-1} \circ \rho_0 \circ \ldots \circ \rho_0 \circ h_1(\mathbf{x})$$

for affine functions h_i.

Next, note that

$$\begin{aligned} \bigoplus_{i=1}^{n} b_i \odot x^{\odot \alpha_i} &= \max\{p(\mathbf{x}), q(\mathbf{x})\} \\ &= \max\{p(\mathbf{x}) - q(\mathbf{x}), 0\} + q(\mathbf{x}) \\ &= \rho_{-\infty} \left(\rho_0(p(\mathbf{x}) - q(\mathbf{x})) + \rho_0(q(\mathbf{x})) - \rho_0(-q(\mathbf{x})) \right) \\ &= \rho_{-\infty} \circ e_n \circ \rho_0 \circ g_n(y(\mathbf{x})) \end{aligned}$$

where e_n is the linear function

$$e_n(\mathbf{x}) = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} \mathbf{x}$$

and g_n is the linear function

$$g_n(\mathbf{x}) = \begin{pmatrix} 1 & -1 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} \mathbf{x}$$

We obtain Equation 5.1 by defining

$$f_i = \begin{cases} h_i & i = 1, \ldots, n-2 \\ g_n \circ h_{i-1} & i = n-1 \\ e_n & i = n \end{cases}$$

It remains to show that we can also express a signomial tropical quotient $\varphi(x) \oslash \psi(x)$ as a ReLU network where φ and ψ are tropical signomials with n and m tropical summands respectively. By Equation (5.1), both $\varphi(x)$ and $\psi(x)$ can be represented as ReLU networks. Again, fuse these two representation into a single ReLU network representation of $z(\mathbf{x}) = (\varphi(x), \psi(x))$ using $\max(m, n)$ layers.

Next, note that $\varphi \oslash \psi$ can be written

$$\begin{aligned} (\varphi \oslash \psi)(\mathbf{x}) &= \rho_{-\infty}\left(\rho_0(p(\mathbf{x})) - \rho_0(-p(\mathbf{x}) + \rho_0(-q(\mathbf{x})) - \rho_0(q(\mathbf{x}))\right) \\ &= \rho_{-\infty} \circ j_n \circ \sigma_0 \circ k_n(z(\mathbf{x})) \end{aligned}$$

where j_n is the linear function

$$j_n(\mathbf{x}) = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \mathbf{x}$$

and k_n is the linear function

$$k_n(\mathbf{x}) = \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} \mathbf{x}$$

Analogous to the proof of Equation 5.1, this gives a ReLU representation of $\varphi(x) \oslash \psi(x)$ using $\max(m, n) + 1$ layers.

□

Translated to the setting of affine geometry, this result reads:

Corollary 5.3.5. *Let $f\colon \mathbb{R}^d \to \mathbb{R}$. Then f is a* dcpa *function if and only if it is a neural network in the sense of Definition 2.1.1.*

Chapter 6

Characterizing the Decision Boundary

We have seen in Chapter 4 how we can indentify neural networks (in the sense of Definition 2.1.1) as dcpa functions using affine geometry. In Chapter 5 we have translated this result to tropical geometry, where it reads that every neural network can be written as a tropical rational map.

In this chapter, we use these results to derive a characterization of a ReLU binary classification network which allows us to count decision boundary pieces. Going forward, we make the decision to formulate successive results in the affine setting. This is because it is arguably slightly more intuitive than the tropical setting, even though one can easily switch from one to the other.

6.1 Affine Hypersurfaces

In this section we study affine hypersurfaces which arise from cpa functions. We again begin with a number of basic definitions.

Throughout this section, let $S = \{s_1, \ldots, s_n\} \subseteq \mathfrak{D}$ be a set of n dual points.

Definition 6.1.1. Given a set of indices $I \subseteq \{1, \ldots, n\}$, we write

$$S_I := \{s_i \,|\, i \in I\}.$$

I think there is a mistake in the original paper. What I think they mean to say is $U^*(S)$, not $U(S)$. Also this is what they do with their example. And if they would actually mean $\mathcal{U}(S)$, then in their proof they could just stop after the first step. There would be no need to do steps two and three. At least thats what I think. Can you confirm? If yes, how should I cite. Because what I say here is different to what they say in their Proposition 9? The following lemma relates upper convex hulls to CPAs.

Proposition 6.1.2 (Maximality of Upper Convex Hull (Proposition 9 in [14]))**.** *Let $S \subseteq \mathfrak{D}$ be a finite set of points. Then for every point $w \in \mathfrak{D}$ lying below or on $\mathcal{U}(S)$ (in the sense of Definition 2.3.3), the affine function dual to w lies fully below the maximum of the affine* *functions whose duals lie in $\mathcal{U}^*(S)$. That is,*

$$\mathcal{R}(w) \leq \max\{\mathcal{R}(s) \mid s \in \mathcal{U}^*(S)\} = \mathcal{Q}(\mathcal{U}^*(S)).$$

Proof. Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathfrak{D}$, $n \geq 3$, be dual points. We start with the following two observations:

i) if $(\mathbf{x}_1, y_1)$ lies directly below $(\mathbf{x}_2, y_2)$, i.e. $\mathbf{x}_1 = \mathbf{x}_2$ and $y_1 < y_2$, then the dual plane related to $(\mathbf{x}_1, y_1)$ lies below $(\mathbf{x}_2, y_2)$, i.e. $\mathcal{R}((\mathbf{x}_1, y_1))(\mathbf{x}) < \mathcal{R}((\mathbf{x}_2, y_2))(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^d$

ii) if $(\mathbf{x}_n, y_n)$ lies on a face of $\mathcal{U}(S)$ convexly spanned by $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n-1}, y_{n-1}) \in \mathcal{U}^*(S)$, then $\mathcal{R}(\mathbf{x}_n, y_n) \leq \max\{\mathcal{R}((\mathbf{x}_i, y_i) \mid i = 1, \ldots, n-1\}$.

Claim $i)$ is trivial. For claim $ii)$, assume there exist $\alpha_i \in [0, 1]$, $\sum_{i=1}^{n}$, s.t.

$$(\mathbf{x}_n, y_n) = \sum_{i=1}^{n-1} \alpha_i (\mathbf{x}_i, y_i).$$

Then

$$\mathcal{R}((\mathbf{x}_n, y_n))(\mathbf{x}) = \sum_{i=1}^{n-1} \alpha_i \mathcal{R}((\mathbf{x}_i, y_i))(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbb{R}^d$$

by linearity of $\mathcal{R}$ (see Proposition 4.2.4). In particular,

$$\mathcal{R}(\mathbf{x}_n, y_n)(\mathbf{x}) \leq \max\{\mathcal{R}((\mathbf{x}_i, y_i))(\mathbf{x}) \mid i = 1, \ldots, n-1\} \quad \forall \mathbf{x} \in \mathbb{R}^d.$$

This shows claim $ii)$.

The Proposition then follows from the following observation. Assume that the point $(\mathbf{x}_1, y_1)$ lies below or on $\mathcal{U}(S)$. Let $(\mathbf{x}_2, y_1)$ be the point exactly above $(\mathbf{x}_1, y_1)$ lying on $\mathcal{U}(S)$. Then, by $i)$, $\mathcal{R}((\mathbf{x}_1, y_1)) \leq \mathcal{R}((\mathbf{x}_2, y_2))$. Furthermore, by $ii)$, $\mathcal{R}((\mathbf{x}_2, y_2)) \leq \max\{\mathcal{R}(s) \mid s \in \mathcal{U}^*(S)\}$. This shows the claim □

The following Corollary is stated without proof in [14]:

Corollary 6.1.3 (CPAs as Upper Convex Hulls). *Every CPA function $\mathcal{Q}(S)$ can be uniquely represented as an upper convex hull in dual space. That is, $\mathcal{Q}(S) = \mathcal{Q}(\mathcal{U}^*(S))$*

Proof. Let $\mathcal{Q}(S)$ be a CPA function. Then for any $\mathbf{x} \in \mathbb{R}^d$,

$$\begin{aligned}
\mathcal{Q}(S)(\mathbf{x}) &= \max_{s \in S} \mathcal{R}(s)(\mathbf{x}) \\
&= \max\{\max_{s \in \mathcal{U}^*(S)} R(s)(\mathbf{x}), \max_{s \in S \setminus \mathcal{U}^*(S)} \mathcal{R}(s)(\mathbf{x})\} \\
&\overset{6.1.2}{=} \max_{s \in \mathcal{U}^*(S)} R(s)(\mathbf{x}) \\
&= \mathcal{Q}(\mathcal{U}^*(S))(\mathbf{x}).
\end{aligned}$$

This shows the claim. □

The following two Propositions reveal valuable insights into the nature of dual points related to maximal functions.

Proposition 6.1.4. *Let $w \in \mathcal{C}(S)$. If there exists an $\mathbf{x} \in \mathbb{R}^d$ s.t. $\mathcal{R}(w)(\mathbf{x}) = \mathcal{Q}(S)(\mathbf{x})$, then $w \in \mathcal{U}(S)$.*

Proof. Since $\mathcal{R}(w)(\mathbf{x}) \leq \mathcal{Q}(S)(\mathbf{x})$ is always true, we may assume that $\mathcal{R}(w)(\mathbf{x}) \geq \mathcal{Q}(S)(\mathbf{x})$.

The proof goes by contraposition. Assume $w \notin \mathcal{U}(S)$. We argue analogously to the proof of Proposition 6.1.2. First, there exists a $v \in \mathcal{U}(S)$ directly above w. As in claim $i)$ of the proof, this implies that $\mathcal{R}(w) < \mathcal{R}(v)$. Second, the point v is contained in a face of $\mathcal{U}(S)$. We conclude that $\mathcal{R}(v) \leq \mathcal{Q}(S)$. Stitching these two observations together, we conclude that $\mathcal{R}(w) < \mathcal{Q}(S)$. But since there has to be a $\mathbf{x} \in \mathbb{R}^d$ where this is not true, we arrive at the contradiction. This shows the proof. □

Proposition 6.1.4 argued that dual points corresponding to maximal functions have to lie in the upper convex hull. The following Proposition shows that dual points lying in the upper convex hull which correspond to maximal functions provide information about other maximal functions.

Proposition 6.1.5. *Assume $w \in \mathbb{R}^{d+1}$ lies on a face ζ of $\mathcal{U}(S)$ given by $\zeta = \mathcal{C}(S_I)$ for some $I \subseteq \{1, \ldots, n\}$. Assume furthermore that $\mathcal{R}(w)(\mathbf{x}) = \mathcal{Q}(S)(\mathbf{x})$ for some $\mathbf{x} \in \mathbb{R}^d$. Then*

$$\mathcal{R}(s_i)(\mathbf{x}) = \mathcal{Q}(S)(\mathbf{x}) \quad \forall i \in I.$$

Proof. By assumption there exists a family of non-negative scalars $(\alpha_i)_{i \in I}$ which sum to one such that

$$w = \sum_{i \in I} \alpha_i s_i.$$

By linearity of $\mathcal{R}$ (see Proposition 4.2.4), this implies that

$$\begin{aligned} \mathcal{Q}(S)(\mathbf{x}) &= \mathcal{R}(w)(\mathbf{x}) \\ &= \sum_{i \in I} \alpha_i \mathcal{R}(s_i)(\mathbf{x}). \end{aligned}$$

Assume by contradiction that $\mathcal{R}(s_1)(\mathbf{x}) < \mathcal{Q}(S)(\mathbf{x})$. Then, since $\mathcal{R}(s_i)(\mathbf{x}) \leq \mathcal{Q}(S)(\mathbf{x})$ for all $i \in I$ and since $\sum_{i \in I} \alpha_i = 1$, this implies that

$$\mathcal{Q}(S)(\mathbf{x}) < \mathcal{Q}(S)(\mathbf{x}),$$

which is a contradiction. □

The following Lemma provides a sufficient condition for a number of vertices in the upper convex hull to be linearly independent. It turns out to be useful for proving Proposition 6.1.12.

Lemma 6.1.6. *Let $S_I \subseteq \mathcal{U}^*(S)$ be a size-n-set of vertices of the upper convex hull and assume that $\mathcal{R}(s_i)(\mathbf{x}) = \mathcal{Q}(S)(\mathbf{x}) \neq 0$ for all $i \in I$ and some $\mathbf{x} \in \mathbb{R}^d$. Then the points in S_I are linearly independent.*

Proof. Assume by contradiction that the points in S_I where linearly dependent. This means that there exists a family $(\alpha_i)_{i=1,\dots,n}$, not all zero, such that

$$\sum_{i=1}^{n} \alpha_i s_i = 0.$$

By linearity of $\mathcal{R}$ (see Proposition 4.2.4), it follows that

$$\begin{aligned} 0 &= \sum_{i=1}^{n} \alpha_i \mathcal{R}(s_i)(\mathbf{x}) \\ &= \sum_{i=1}^{n} \alpha_i \mathcal{Q}(S)(\mathbf{x}). \end{aligned}$$

Since $\mathcal{Q}(S)(\mathbf{x}) \neq 0$, we conclude that

$$\sum_{i=1}^{n} \alpha_i = 0.$$

Thus the set S_I is linearly dependent, which is a contradiction to Lemma 2.3.6 since it contains only extreme points of $\mathcal{C}(S)$. □

Next we define what it means for a cpa function to induce a *tessellation* of $\mathbb{R}^d$. This definition plays an important role towards characterizing decision boundaries of ReLU classification networks.

Definition 6.1.7 (Tessellation)**.** Given a cpa function $F(x) := \max\{f_1(x), \dots, f_n(x)\}$, an *affine region* of F is

$$\{\mathbf{x} \in \mathbb{R}^d \,|\, f_i(x) = f_{i'}(x) \geq f_j(x) \text{ for all } i, i' \in I, j \in J\}$$

where I, J are disjoint sets whose union is $\{1, ..., n\}$. The set of all affine regions of F is called a *tessellation* of F and denoted by $\mathcal{T}(F)$.

By a slight abuse of notation, we will write $\mathcal{T}(S)$ for the tessellation induced by $\mathcal{Q}(S)$.

Lemma 6.1.8. *The tessellation induced by a* cpa *function F forms a polyhedral complex.*

Proof. Every affine region of F is a polyhedron since it is defined by a set of linear inequalities. It is left to show that the following two properties hold (see Definition 2.2.6):

i) any face of an affine region is also an affine region

ii) the intersection of two affine regions is either empty or a face of both intersecting affine regions.

But this follows directly from the Definition of the tessellation. Indeed, let σ be an affine region defined via two sets I and J as in Definition 6.1.7. Then a face of σ is an affine region associated with two sets $I' \supseteq I$, $J' \subseteq J$ obtained from moving indices from J to I. This shows $i)$. To see that $ii)$ holds, observe that the intersection of two affine regions associated with sets I, J and I', J' respectively, is the affine region associated with the sets $I \cap I'$, $J \cup J' \cup I \setminus I' \cup I' \setminus I$. □

We can thus think of a tessellation as a polyhedral complex and the following definition makes sense:

Definition 6.1.9. Let F be a dcpa function. Then the affine regions of $\mathcal{T}(F)$ are cells in a polyhedral complex. We denote by $\mathcal{T}_k(f)$ the union of all cells of dimension k.

Definition 6.1.10 (Affine Hypersurface)**.** The *affine hypersurface* of a cpa function $F(x) = \max_i f_i$ is the set

$$\mathcal{H}(F) = \{x \in \mathbb{R}^d \mid f_i(x) = f_j(x) = F(x) \text{ for some } i \neq j\}.$$

Remark 6.1.11. For once we translate Definition 6.1.10 from the affine to the tropical setting. This should be an example for how the concepts studied in this chapter have a counterpart in tropical geometry.

The *tropical hypersurface* of a tropical polynomial $f(x) = c_1 x^{\odot\alpha_1} \oplus \ldots \oplus c_r x^{\odot\alpha_r}$ is the set

$$\mathcal{H}(f) \coloneqq \{x \in \mathbb{R}^d \mid c_i x^{\odot\alpha_i} = c_j x^{\odot\alpha_j} = f(x) \text{ for some } i \neq j\}.$$

After this short excursion, we come back to the affine setting.

Proposition 6.1.12 (Dual Cell Duality)**.** *Let $S \subseteq \mathfrak{D}$ be a set of n dual points and $\sigma \subseteq \mathbb{R}^d$ a cell of $\mathcal{T}(S)$. Then σ has dimension k if and only if it can be written as the region defined by a system*

$$\begin{cases} f_i = f_{i'} & \forall i, i' \in I \\ f_i \geq f_j & \forall i \in I,\ j \in J \end{cases}$$

for some disjoint partition $I \sqcup J = \{1, ..., n\}$ of size $|I| = d - k + 1$, $|J| = n - (d - k + 1)$ and affine functions $f_i \colon \mathbb{R}^d \to \mathbb{R}$ s.t. the family $(f_i)_{i \in I}$ is maximal and linearly independent in the sense of Definition 2.3.4 and so that $S_I \subseteq \mathcal{U}^*(S)$. By maximal we mean that it is not possible to move indices from J to I s.t. the solution-space stays the same and the family $(f_i)_{i \in I}$ stays linearly independent.

Proof. "$\Rightarrow$" First assume that σ has dimension k. We start by enumerating $S = \{s_1, \ldots, s_n\}$ and introduce the short-hand notation $f_i \coloneqq \mathcal{R}(s_i)$, writing $f_i(\mathbf{x}) = \mathbf{a}_i\mathbf{x} + b_i$ for some $\mathbf{a}_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$.

W.l.o.g. assume that σ can then be written as the region defined by a system

$$\begin{cases} f_i = f_{i'} & \forall i, i' \in I' \\ f_i > f_j & \forall i \in I,\ j \in J' \end{cases} \tag{6.1}$$

for some disjoint partition $I' \sqcup J' = \{1, ..., n\}$. We may furthermore assume that I' contains the first $|I'|$ indices, i.e. $I' = \{1, ..., |I'|\}$. (Note that, compared to Definition 6.1.7, the second row of System (6.1) contains a strict "$>$" instead of a "$\geq$". This can be achieved by moving indices from J' to I').

We will show how to reformulate System (6.1) in such a way that $I = I' \cap \mathcal{U}^*(S)$ has the desired properties.

Our first step is to show the following inequality:

$$d - k + 1 \leq |I'|. \tag{6.2}$$

To see that this is indeed true, we introduce matrices $\mathbf{A} \in \mathbb{R}^{|I'|-1,d}$ and $\overline{\mathbf{A}} \in \mathbb{R}^{|J'|,d}$, where $\mathbf{A}_{ij} = (\mathbf{a}_i - \mathbf{a}_{|I'|})_j$ and $\overline{\mathbf{A}}_{ij} = (\mathbf{a}_{|I'|} - \mathbf{a}_{|I'|+i})_j$. We furthermore introduce the vectors $\mathbf{b} \in \mathbb{R}^{|I'|-1}$ and $\overline{\mathbf{b}} \in \mathbb{R}^{|J'|}$ where $\mathbf{b}_i = b_i - b_{|I'|}$ and $\overline{\mathbf{b}}_i = b_{|I'|} - b_{|I'|+i}$.

This allows us to re-write System (6.1) as

$$\begin{cases} \mathbf{A}\mathbf{x} = \mathbf{b} \\ \overline{\mathbf{A}}\mathbf{x} > \overline{\mathbf{b}} \end{cases} \tag{6.3}$$

Introducing the sets

$$\begin{aligned} G &:= \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{A}\mathbf{x} = \mathbf{b}\} \\ H &:= \{\mathbf{x} \in \mathbb{R}^d \mid \overline{\mathbf{A}}\mathbf{x} > \overline{\mathbf{b}}\}, \end{aligned}$$

allows to re-write

$$\sigma = G \cap H.$$

Next, we claim that

$$\dim \sigma = \dim \ker \mathbf{A}. \tag{6.4}$$

Indeed, since the cell σ is non-empty, we know that $G \neq \emptyset$ and thus $\dim G = \dim \mathbf{A}$. Furthermore, H is the intersection of *open* half-spaces in $\mathbb{R}^d$ and since σ is non-empty, it is an open polytope of full dimension d. This shows equation (6.4)

We come back to proving Equation (6.2). The following clearly holds by construction:

$$\operatorname{rank} \mathbf{A} \leq |I'| - 1,$$

which, using the rank-nullity-theorem, is equivalent to

$$\begin{aligned} d - |I'| + 1 &\leq d - \operatorname{rank} \mathbf{A} \\ &= \dim \ker \mathbf{A} \\ &\overset{6.4}{=} \dim \sigma \\ &= k \end{aligned}$$

This shows Equation 6.2 and finishes the first step.

Let

$$K := \{i \in I \mid s_i \notin \mathcal{U}^*(S)\}$$

and move the set K from I' to J':

$$\begin{aligned} I &:= I' \setminus K \\ J &:= J' \cup K. \end{aligned}$$

We claim that Equation (6.2) still holds for the new set I, i.e. that

$$d - k + 1 \leq |I|. \tag{6.5}$$

Indeed, for any $k \in K$, s_k lies on a face ζ of $\mathcal{U}(S)$. By Proposition 6.1.5, the vertices of ζ are also contained in S_I. We conclude that s_k is a linear combination of some vertices in S_I and thus

$$\operatorname{rank} \mathbf{A} \leq |I| - 1.$$

Equation (6.5) then follows analogously to Equation (6.2).

Next, we claim that actually equality holds in Equation (6.5). Assume by contradiction that the inequality in Equation (6.5) was strict, i.e.

$$d - k + 1 < |I|.$$

As in the above calculation, we can then use the rank-nullity-theorem to deduce

$$\operatorname{rank} \mathbf{A} = d - k < |I| - 1.$$

Consequently the rows of $\mathbf{A}$ are linearly dependent. Since the cell σ is non-empty, $\operatorname{rank} \mathbf{A} = \operatorname{rank}[\mathbf{A}|\mathbf{b}]$, which implies that the S_I are linearly dependent. But that contradicts Proposition 6.1.6 since $\mathcal{S}_I \subseteq \mathcal{U}^*(S)$. We conclude that

$$d - k + 1 = |I|$$

It follows from Proposition 4.2.4.ii) that the $(f_i)_{i \in I}$ are also linearly independent. Furthermore, $I \subset \mathcal{U}^*(S)$ by construction. Finally, the family $(f_i)_{i \in I}$ is maximal by construction (moving functions from J to I either alters the solution-space or adds a linearly dependent function). This shows one implication.

"$\Leftarrow$" Now assume that σ is the solution of a system as in the Proposition. Move indices from I to J without altering the solution space until the cell is given be the solution of a system of the form (6.1) with a partition $\{1, \ldots, n\} = I' \sqcup J'$(in particular, all functions with index in I' are strictly larger than those with index in J' when restricted to σ). Repeat the construction of $\mathbf{A}, \overline{\mathbf{A}}, \mathbf{b}, \overline{\mathbf{b}}, H$ and G as above. We claim that

$$\operatorname{rank} \mathbf{A} = d - k. \tag{6.6}$$

By assumption, we know that the family $(f_i)_{i \in I}$ is linearly independent. By maximality, so is the family $(f_{i'})_{i' \in I'}$. Equation (6.6) then follows from Proposition 4.2.4.ii) and from the fact that there are exactly $|I| - 1 = d - k$ linearly independent rows in $\mathbf{A}$ by construction. Applying the Rank-nullity-theorem then yields

$$\dim \sigma = \dim \ker \mathbf{A} = d - \operatorname{rank} \mathbf{A} = k.$$

This finishes the proof.

□

Proposition 6.1.13. *Let I and J form a bipartition of $\{1, \ldots, n\}$ related to a k-cell $\sigma \in \mathcal{T}_k(S)$ as in Proposition 6.1.12. Then S_I forms a $(d-k)$-face in $\mathcal{U}(S)$, i.e. $\mathcal{C}(S_I) \in \mathcal{U}_{d-k}(S)$*

Proof. Write

$$\zeta := \mathcal{C}(S_I)$$

for the convex hull of S_I. We know that $S_I \subseteq \mathcal{U}^*(S)$ by construction of I (see Proposition 6.1.12) and it is left to show that

i) $\zeta \subseteq \mathcal{U}(S)$

ii) $\dim \zeta = d - k$.

We start with $i)$. For any family of positive scalars $\alpha_i \geq 0$, $\sum_{i=1}^n \alpha_i = 1$, the following is true for $i \in I$ and any $\mathbf{x} \in \sigma$:

$$\begin{aligned}
\mathcal{Q}(S)(\mathbf{x}) &= \mathcal{R}(s_i)(\mathbf{x}) \\
&= \sum_{j \in I} \alpha_j \mathcal{R}(s_i)(\mathbf{x}) \\
&= \sum_{j \in I} \alpha_j \mathcal{R}(s_j)(\mathbf{x}) \\
&\overset{4.2.4}{=} \mathcal{R}(\sum_{j \in I} \alpha_j s_j)(\mathbf{x}).
\end{aligned}$$

It follows from Proposition 6.1.4 that $\sum_{j \in I} \alpha_j s_j \in \mathcal{U}(S)$ for any family $(\alpha_i)_{i \in I}$ of non-negative scalars summing to one. We conclude $i)$.

For $ii)$, note that $|I| = d - k + 1$ and $(f_i)_{i \in I}$ is a linearly independent family by assumption. It follows from Proposition 4.2.4 that S_I is a linearly independent family of $d - k + 1$ points. The claim then follows from Lemma 2.3.7. □

Proposition 6.1.14. *The k-cells of $\mathcal{T}(S)$ are in one-to-one correspondence with the $(d-k)$-faces in $\mathcal{U}(S)$. Specifically, there exists a bijection*

$$\begin{aligned}
\mathcal{T}_k(S) &\to \mathcal{U}_{d-k}(S) \\
\sigma &\mapsto \mathcal{C}(S_{I(\sigma)})
\end{aligned}$$

where $I(\sigma)$ is the set I of Proposition 6.1.12 related to σ.

Proof. We first have to show that $I(\sigma)$ is well-defined, i.e. that the set I in Proposition 6.1.12 is unique. But this follows directly from the construction of $I = I' \cap \mathcal{U}^*(S)$ and from the uniqueness of I'. Next, it follows from Proposition 6.1.13 that the map is well-defined, i.e. that it maps to the space of $(d-k)$-dimensional faces in $\mathcal{U}(S)$. Finally, the fact that it defines a bijection is also clear from the construction. □

6.1.1 Refinements

In this section we characterize positive and negative samples of a ReLU binary classification network.

We start by defining introducing what it means for a dcpa function to induce a tesselation of $\mathbb{R}^d$.

Definition 6.1.15. Let $F = \mathcal{Q}(P) - \mathcal{Q}(N)$ be a dcpa function. We then define the tessellation $\mathcal{T}(P, N)$ induced by F to consist of all pairwise non-empty intersections of cells induced by P and N, i.e.

$$\mathcal{T}(P, N) := \{\sigma \cap \sigma' \,|\, \sigma \in \mathcal{T}(P),\ \sigma' \in \mathcal{T}(N)\} \setminus \emptyset.$$

Definition 6.1.16 (Refinements)**.** Let F and G be cpa functions. Then we say that $\mathcal{T}(F)$ is a *refinement* $\mathcal{T}(G)$ if every cell of $\mathcal{T}(F)$ is contained in a cell of $\mathcal{T}(G)$. In this case, we write $\mathcal{T}(F) \ll \mathcal{T}(G)$.

Lemma 6.1.17. *Given two sets of dual points $P, N \subseteq \mathfrak{D}$, it holds that*

$$\mathcal{T}(P, N) \ll \mathcal{T}(P \cup N) \ll \mathcal{T}(P), \mathcal{T}(N).$$

Proof. For ease of notation, enumerate $N = \{n_1, \ldots, n_m\}$ and $P = \{p_1, \ldots, p_k\}$ with $m, k \in \mathbb{N}$. Furthermore write $f_i := \mathcal{R}(n_i)$ as well as $g_i := \mathcal{R}(p_i)$.

A cell of $\mathcal{T}(N)$ is given by the solution of a system

$$\begin{cases} f_i = f_j \;\; \forall i, j \in I \\ f_i \geq f_j \;\; \forall i \in I,\ j \in J. \end{cases} \tag{6.7}$$

for some disjoint partition $I \sqcup J = \{1, \ldots, m\}$.

A cell of $\mathcal{T}(P \cup N)$ is given by the solution of a system

$$\begin{cases} f_i = f_j \;\; \forall i, j \in I \\ g_{i'} = g_{j'} \;\; \forall i, j \in I' \\ f_i = g_{j'} \;\; \forall i \in I,\ j' \in I' \\ f_i \geq f_j \;\; \forall i \in I,\ j \in J \\ g_{i'} \geq g_{j'} \;\; \forall i' \in I',\ j' \in J' \\ f_i \geq g_{j'} \;\; \forall i \in I,\ j' \in J' \\ g_{i'} \geq f_j \;\; \forall i' \in I',\ j \in J \end{cases} \tag{6.8}$$

for some disjoint partitions $I \sqcup J = \{1, \ldots, m\}$ and $I' \sqcup J' = \{1, \ldots, k\}$.

A cell of $\mathcal{T}(P, N)$ is given by the solution of the system.

$$\begin{cases} f_i = f_j \;\; \forall i, j \in I \\ g_{i'} = g_{j'} \;\; \forall i', j' \in I' \\ f_i \geq f_j \;\; \forall i \in I,\ j \in J \\ g_{i'} \geq g_{j'} \;\; \forall i' \in I',\ j' \in J' \end{cases} \tag{6.9}$$

for some disjoint partitions $I \sqcup J = \{1, \ldots, m\}$ and $I' \sqcup J' = \{1, \ldots, k\}$.

It follows immediately that any set solving System (6.9) also solves System (6.8) and any set solving System (6.8) also solves System (6.7). This implies the claim.

□

The following Proposition characterizes positive and negative regions of a network using the above refinement.

Lemma 6.1.18. *Let $P, Q \subseteq \mathbb{R}^{d+1}$ be finite sets of points. Then the following are true for $x \in \mathbb{R}^d$:*

$$\mathcal{Q}(P \cup N)(x) = \mathcal{Q}(P)(x) \iff \mathcal{Q}(P)(x) \geq \mathcal{Q}(N)(x) \tag{6.10}$$

$$\mathcal{P}(P \cup N)(x) = \mathcal{Q}(N)(x) \iff \mathcal{Q}(P)(x) \leq \mathcal{Q}(N)(x). \tag{6.11}$$

Proof. Follows from Lemma 4.3.4 i). □

Proposition 6.1.19. *Let $f_\theta = \mathcal{Q}(P) - \mathcal{Q}(N)$ be a ReLU binary classification network. Then the following are true for an input $x \in \mathbb{R}^d$:*

$$f_\theta(x) \geq 0 \iff \mathcal{Q}(P \cup N)(x) = \mathcal{Q}(P)(x) \tag{6.12}$$

$$f_\theta(x) \leq 0 \iff \mathcal{Q}(P \cup N)(x) = \mathcal{Q}(N)(x). \tag{6.13}$$

Proof. Follows from Lemma 6.1.17 and Lemma 6.1.18. □

Proposition 6.1.19 tells us that positively labeled points are characterized by the property that the maximum $\mathcal{Q}(P \cup N)$ is attained by $\mathcal{Q}(P)$. Similarly, negatively labeled points are characterized by the property that the maximum is attained by $\mathcal{Q}(N)$.

6.2 Decision Boundary

As a final step, we will use the constructions from the previous sections to characterize decision boundaries of ReLU networks.

In the following, let $f_\theta \colon \mathbb{R}^d \to \mathbb{R}$ be a ReLU binary classification network in the sense of Definition 2.1.3. By Corollary 4.3.10, we can find two sets of dual points $P, N \subseteq \mathfrak{D}$ such that f_θ is the dcpa function

$$f_\theta = \mathcal{Q}(P) - \mathcal{Q}(N).$$

According to Definition 2.1.5, the networks decision boundary is thus given by

$$\mathcal{B}_\theta = \left(\mathcal{Q}(P) - \mathcal{Q}(N)\right)^{-1}(0).$$

As a consequence we are interested in studying the zero-set D of a dcpa function. As it turns out, D is closely related to cells of a certain tessellation.

We start with a special case.

Proposition 6.2.1 (Decision Boundary I (Proposition 19 in [14])). *Let $F = \mathcal{Q}(P)$ and $G = \mathcal{Q}(N)$ be cpa functions $\mathbb{R}^d \to \mathbb{R}$ for some finite sets of dual points $P, N \subseteq \mathfrak{D}$. Assume that no points of P lie on $\mathcal{U}(N)$ and vice versa. Let D be the zero-set of $F - G$. Then D is the union of precisely those $(d-1)$-cells of $\mathcal{T}(P \cup N)$ which (in the sense of proposition 6.1.14) correspond to edges (i.e. 1-faces) of $\mathcal{U}(P \cup N)$ with one end in P and the other end in N.*

Proof. For ease of notation, enumerate $N = \{n_1, \dots, n_m\}$ and $P = \{p_1, \dots, p_k\}$ with $m, k \in \mathbb{N}$. Furthermore write $f_i := \mathcal{R}(n_i)$ as well as $g_i := \mathcal{R}(p_i)$.

Fix $\mathbf{x} \in \mathbb{R}^d$. Then $\mathbf{x}$ lies in D if and only if $\mathcal{Q}(P)(\mathbf{x}) = \mathcal{Q}(N)(\mathbf{x})$, which by definition means that $\max_{1 \le i \le m} f_i(\mathbf{x}) = \max_{1 \le j \le k} g_j(\mathbf{x})$. Let f_i and g_j be respective maximizers of the two sides of the equation. Then we can re-write the equality condition as

$$\begin{cases} f_i(\mathbf{x}) = g_j(\mathbf{x}) \\ f_i(\mathbf{x}) \ge f_k(\mathbf{x}) & \forall k \in \{1, \dots, m\} \setminus \{i\} \\ f_i(\mathbf{x}) \ge g_k(\mathbf{x}) & \forall k \in \{1, \dots, k\} \setminus \{j\}. \end{cases}$$

The point $\mathbf{x}$ by Definition solves this system of equations if and only if it lies inside of a $(d-1)$-cell of $\mathcal{T}(P \cup N)$ with one maximizer coming from $\mathcal{Q}(P)$ and one coming from $\mathcal{Q}(N)$.

By the duality result in Proposition 6.1.14, such a cell corresponds to a dual 1-cell (i.e. an edge) of $\mathcal{U}(P \cup N)$ with some vertex $\mathcal{R}^{-1}(f_i) \in N$ and some vertex $\mathcal{R}^{-1}(g_j) \in P$. Indeed, in the setting of Proposition 6.1.14, the set S_I would contain both n_i and p_j. Thus, $\eta = \mathcal{C}(S_I)$ would contain both $\mathcal{R}^{-1}(f_i) = n_i \in N$ and $\mathcal{R}^{-1}(g_j) = p_j \in P$.

Finally, since $\mathcal{P} \cap \mathcal{U}(N) = N \cap \mathcal{U}(P) = \emptyset$, such a cell has to have one end in P and one end in N. □

Even though Proposition 6.2.1 handles the in practice most likely case that $P \cap \mathcal{U}(N) = N \cap \mathcal{U}(P) = \emptyset$, this extra condition need not always be satisfied. The following Proposition handles the general case.

Proposition 6.2.2 (Decision Boundary II (Proposition 20 in [14])). *Let $F = \mathcal{Q}(P)$ and $G = \mathcal{R}(N)$ be* cpa *functions $\mathbb{R}^d \to \mathbb{R}$ for some finite sets of dual points $P, N \subseteq \mathfrak{D}$. Let D be the zero-set of $F - G$. Then D is the union of precisely those $(d-1)$-cells of $\mathcal{T}(P \cup N)$ which (in the sense of proposition 6.1.14) correspond to edges of $\mathcal{U}(P \cup N)$ containing points from both P and N. Thus, we have to require that the edge contains points from both P and N.*

Proof. We start the proof as the one for Proposition 6.2.1 and stop before identifying the dual cell. Since $P \cap \mathcal{U}(N)$ and $N \cap \mathcal{U}(P)$ need not be empty, we can not conclude that edges in $\mathcal{U}(P \cup N)$ containing both points from P and N have to start in P and end in N. It could, for example, also be the case that the edge starts and ends in N but contains a points in P. □

Proposition 6.2.2 gives us the following Corollary:

Corollary 6.2.3. *The decision boundary of a ReLU network is piece-wise linear in the sense that each piece is a $(d-1)$-cell.*

Definition 6.2.4. We define the *boundary complexity* $\mathfrak{C}(f_\theta)$ of a ReLU binary classification network f_θ in the sense of Definition 2.1.3 as the number of linear pieces in its decision boundary.

Chapter 7

Theoretical Decision Boundary Complexity

We have seen in Corollary 5.3.5 that ReLU networks $f\colon \mathbb{R}^d \to \mathbb{R}$ can be identified with dcpa functions and thus be written as $f = \mathcal{Q}(P) - \mathcal{Q}(N)$. By Proposition 6.2.2, estimating network complexity $\mathfrak{C}(F)$ thus comes down to counting the possibilities of creating edges in $\mathcal{U}(P \cup N)$ which contain points from both P and N. In this chapter we develop a mathematical framework to do exactly that.

7.1 Basic Concepts

We start by giving some basic definitions and lemmas from statistics.

Definition 7.1.1 (Binomial Coefficient)**.** For an integer $k \in \mathbb{Z}$ and a positive integer $n \in \mathbb{N}$, we define the *binomial coefficient* as

$$\binom{n}{k} = \begin{cases} \frac{n!}{k!(n-k)!}, & k \geq 0 \\ 0, & k < 0 \text{ or } k > n \end{cases}$$

In particular, $\binom{n}{0} = 1$.

We will furthermore need the following basic lemmas from combinatorics.

Lemma 7.1.2 (Stars and Bars)**.** *For any pair of positive integers $n, k \in \mathbb{N}$, the number of k-tuples whose sum is n is equal to $\binom{n-1}{k-1}$.*

Lemma 7.1.3 (Van der Monde Identity (Table 169 in [16]))**.** *Let r be an integer and $m, n \in \mathbb{N}_0$ be non-negative integers. Then*

$$\sum_{k\in\mathbb{Z}} \binom{m}{k}\binom{n}{r-k} = \binom{m+n}{r}.$$

Lemma 7.1.4 ([16]). *Let $m, n \in \mathbb{Z}$ be integers and $r, s \in \mathbb{N}_0$ be non-negative integers. Then*

$$\sum_{k\in\mathbb{Z}} \binom{r}{m+k}\binom{s}{n+k} = \binom{r+s}{r-m+n}.$$

Proof.

$$\begin{aligned}\sum_{k\in\mathbb{Z}} \binom{r}{m+k}\binom{s}{n+k} &= \sum_{k\in\mathbb{Z}} \binom{r}{r-m-k}\binom{s}{s-n-k} \\ &= \sum_{k\in\mathbb{Z}} \binom{r}{k}\binom{s}{s-n-(r-m-k)} \\ &= \sum_{k\in\mathbb{Z}} \binom{r}{k}\binom{s}{r-m-k+n} \\ &= \sum_{k\in\mathbb{Z}} \binom{r}{k}\binom{s}{(r-m+n)-k} \\ &= \binom{r+s}{r-m+n}\end{aligned}$$

In step one and three we used symmetry of the binomial coefficient, in step two we did an index shift. In the last step we used Lemma 7.1.3. □

Lemma 7.1.5. *For any non-negative integer $n \in \mathbb{N}_0$ and any integer $k \in \mathbb{Z}$, the following recursive formula holds:*

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}.$$

7.2 The Two-Dimensional Case

In this section develop the theory for the case that the dual space has dimension 2. We begin in Section 7.2.1 with a more general perspective, concentrating on what happens in dual space. In Section 7.2.2 we apply our findings to neural networks without biases.

7.2.1 Abstract Considerations

We start with some basic definitions. Throughout this section fix two integers $\alpha, \beta \in \mathbb{N}$. Let $n := \alpha + \beta$.

Definition 7.2.1 (Probability Space). Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with sample-space

$$\Omega := \bigoplus_{i=1}^{\alpha} \mathbb{R}^2 \oplus \bigoplus_{i=1}^{\beta} \mathbb{R}^2,$$

σ-algebra

$$\mathcal{F} := \bigotimes_{i=1}^{\alpha} \mathcal{B}(\mathbb{R}^2) \otimes \bigotimes_{i=1}^{\beta} \mathcal{B}(\mathbb{R}^2)$$

and gaussian measure

$$d\mathbb{P} := \bigotimes_{i=1}^{\alpha} \mathcal{N}(0, \Sigma) d\lambda^2 \otimes \bigotimes_{i=1}^{\beta} \mathcal{N}(0, \Sigma) d\lambda^2,$$

where $\Sigma = \operatorname{diag}(\sigma^2, \sigma^2)$, $\sigma^2 > 0$ and λ^2 is the 2-dimensional Lebesgue-measure.

Remark 7.2.2 (Notation). We denote elements in Ω by $(\mathcal{R}, \mathcal{B})$ where $\mathcal{R} \in \bigoplus_{i=1}^{\alpha} \mathbb{R}^2$, $\mathcal{B} \in \bigoplus_{i=1}^{\beta} \mathbb{R}^2$ or, alternatively, by $(x_1, \ldots, x_n)$ with $x_1, \ldots, x_n \in \mathbb{R}^2$.

Definition 7.2.3. Given a vector $(x_1, \ldots, x_m) \in (\mathbb{R}^2)^m$ for some $m \in \mathbb{N}$, we define $\operatorname{set}((x_1, \ldots, x_m)) := \{x_1, \ldots, x_m\}$. That is, set extracts the entries of a vector and puts them into a set.

The function set translates a vector $(\mathcal{R}, \mathcal{B}) \in \Omega$ to a set containing n points. We call the points coming from $\mathcal{R}$ *red* and those coming from $\mathcal{B}$ *blue*. The function set is required since, once we switch to the setting of neural networks, the order of the dual points does not matter.

In accordance with Proposition 6.2.2, we are interested in counting *good* edges in $\mathcal{U}(\operatorname{set}(\mathcal{R}, \mathcal{B}))$. We call an edge *good* if it contains both a red and a blue point.

Definition 7.2.4. We define the random variable $e \colon \Omega \to \mathbb{N}$ as counting the number of good edges in $\mathcal{U}(\operatorname{set}(\mathcal{R}, \mathcal{B}))$.

The goal of this section is to determine the distribution of e. We achieve this goal by first assessing the probability that any edge is in the boundary of the convex hull before counting good edges in the upper convex hull.

Definition 7.2.5. Let $(x_1, \ldots, x_n) \in \Omega$ be a vector of points (i.e. $x_i \in \mathbb{R}^2$). We denote by E_{ij} the event that edge (x_i, x_j) is present in the boundary of the convex hull $\mathcal{C}(\{x_1, \ldots, x_n\})$.

Proposition 7.2.6 (Boundary Edges (Satz 4 in [17])). *The probability that an edge between any pair of points is in the boundary of the convex hull $\mathcal{C}(\operatorname{set}(\mathcal{R}, \mathcal{B}))$ is given by*

$$\mathbb{P}(E_{ij} = 1) = \frac{2}{\sqrt{\pi}} \int_{0}^{\infty} \left(\Phi(p)^{n-2} + (1 - \Phi(p))^{n-2}\right) e^{-p^2} dp$$

where

$$\Phi(p) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{p} e^{-\frac{u^2}{2}} du.$$

Now that we know the probability that any edge is in the boundary of the convex hull, the more specific case concerning the upper convex hull follows readily:

Definition 7.2.7. Let $(x_1, \ldots, x_n) \in \Omega$ be a vector of points. We denote by U_{ij} the event that edge (x_i, x_j) is present in the upper convex hull $\mathcal{U}(\{x_1, \ldots, x_n\})$.

Lemma 7.2.8. *The probability of any edge being not only in the boundary of the convex hull but even in the upper convex hull can be computed as*

$$\mathbb{P}(U_{12} = 1) = \mathbb{P}(U_{ij} = 1) = \frac{1}{2}\mathbb{P}(E_{ij} = 1).$$

Proof. Indeed,

$$\begin{aligned}
\mathbb{P}(U_{ij} = 1) &= \mathbb{P}(U_{ij} = 1, E_{ij} = 1) \\
&= \mathbb{P}(U_{ij} = 1 \mid E_{ij} = 1)\mathbb{P}(E_{ij} = 1) \\
&= \frac{1}{2}\mathbb{P}(E_{ij} = 1).
\end{aligned}$$

In the last step we used the symmetry of the distribution. If we know that an edge is in the boundary of the convex hull, then it can be either in the upper or lower convex hull, both with equal probability. □

Proposition 7.2.6 and Lemma 7.2.8 provide the probability that any edge is in the upper convex hull. The next step allows counting the total number of vertices in the upper convex hull:

Definition 7.2.9. Let U be the random variable counting the number of *vertices* in the upper convex hull $\mathcal{U}^*(\mathrm{set}(\mathcal{R}, \mathcal{B}))$.

The following lemma specifically requires the case $d = 2$:

Lemma 7.2.10. *The probability of having u vertices in the upper convex hull is given by*

$$\mathbb{P}(U = u) = \binom{\binom{n}{2}}{u-1}\mathbb{P}(U_{12} = 1)^{u-1}(1 - \mathbb{P}(U_{12} = 1))^{\binom{n}{2}-(u-1)}$$

Proof. Note that there are u vertices if and only if there are $u - 1$ edges in the upper convex hull. We can write U as a sum over $\binom{n}{2}$ Bernoulli random variables. □

The following random variable counts the number of red vertices in the upper convex hull, bringing us a step further towards counting the number of good edges.

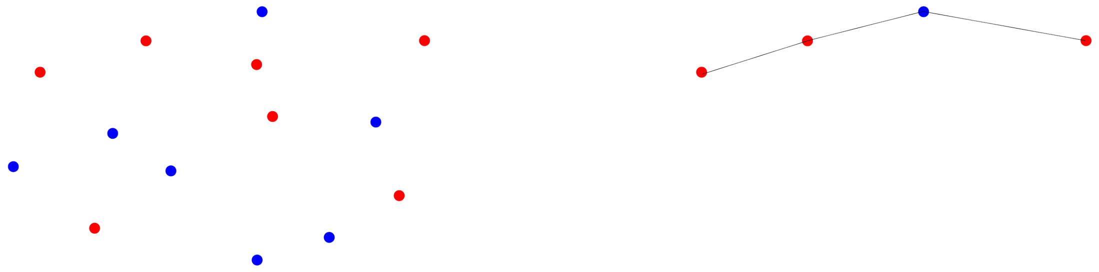
Definition 7.2.11. Let R be the random variable counting the number of red vertices in $\mathcal{U}^*(\mathrm{set}(\mathcal{R}, \mathcal{B}))$.

Lemma 7.2.12. *The probability of having r red points in an upper convex hull of size u is given by*

$$\mathbb{P}(R = r \mid U = u) = \frac{\binom{\alpha}{r}\binom{\beta}{u-r}}{\binom{n}{u}}$$

We now have all of the tools to wrap up our argument for counting the number of good edges.

Definition 7.2.13. Let E be the random variable counting the number of good edges in $\mathcal{U}(\operatorname{set}(\mathcal{R}, \mathcal{B}))$.



(a) Caption for the first subfigure

(b) Caption for the second subfigure

Figure 7.1: An example of red and blue points in $\mathbb{R}^2$ with $\alpha = 7$ and $\beta = 7$. Their upper convex hull can be thought of as a chain of red and blue nobs.

First assume that the total number of vertices as well as the number of red vertices in the upper convex hull is known. Then the number of good edges has the following distribution.

Proposition 7.2.14 (Counting Good Edges)**.** *The probability of having k good edges in the upper convex hull, given that the hull contains u vertices r of which are red, can be computed as follows.*

If $r \in \{0, u\}$, i.e. there are either only red or only blue points in the upper convex hull, then

$$\mathbb{P}(E = k \,|\, U = u, R = r) = \begin{cases} 1, & k = 0 \\ 0, & \textit{otherwise}. \end{cases}$$

If $r \notin \{0, u\}$, i.e. if there are blue and red points in the upper convex hull, then

$$\mathbb{P}(E = k \,|\, U = u, R = r) = \binom{u}{r}^{-1} T_{u,r}(k) \tag{7.1}$$

where

$$T_{u,r}(k) = \binom{r-1}{\lceil \frac{k+1}{2} \rceil - 1} \cdot \binom{u-r-1}{\lfloor \frac{k+1}{2} \rfloor - 1} + \binom{r-1}{\lfloor \frac{k+1}{2} \rfloor - 1} \cdot \binom{u-r-1}{\lceil \frac{k+1}{2} \rceil - 1}.$$

Proof. We think of the upper convex hull as a chain with of u nobs colored red and blue (see Figure 7.1b). The total number of red nobs is r while the total number of blue nobs is $u - r$.

We first compute the un-normalized probabilities, which comes down to counting how many possibilities there are to pick k good edges from a chain with u nobs, r of which are red.

If $r \in \{0, u\}$, then there are only nobs of one color. There can thus only be $k = 0$ good edges.

In the following we may assume that $r \notin \{0, u\}$, i.e. that there are both red and blue nods in the chain.

Let C be a chain as described above. The crucial idea is that there are k good edges in C if and only if C can be subdivided into $k+1$ monochromatic groups (i.e. consecutive nobs) of alternating color.

We start by assuming that the first group is red. Then the number g of red groups is given by $g = \lceil \frac{k+1}{2} \rceil$ (this is because there are $k+1$ groups in total and colors alternate). By Lemma 7.1.2 there are $\binom{r-1}{\lceil \frac{k+1}{2} \rceil - 1}$ possibilities of forming $\lceil \frac{k+1}{2} \rceil$ groups from r red nobs (order of groups matters).

We apply the same argument for the left-over blue balls. Since C starts with a red group, we have to count how many ways there are to form $\lfloor \frac{k+1}{2} \rfloor$ groups from $u - r$ blue nobs (order of groups matters).

This shows the first summand of Equation (7.1).

Next, we assume that the first group is blue. In this case $g = \lfloor \frac{k+1}{2} \rfloor$ and we repeat the argument above. This concludes the first part.

The inverse of the normalization-constant is

$$C^{-1} = \sum_{k \in \mathbb{Z}} \binom{r-1}{\lceil \frac{k+1}{2} \rceil - 1} \cdot \binom{u-r-1}{\lfloor \frac{k+1}{2} \rfloor - 1} + \binom{r-1}{\lfloor \frac{k+1}{2} \rfloor - 1} \cdot \binom{u-r-1}{\lceil \frac{k+1}{2} \rceil - 1}$$

We split the computation of C^{-1} into even and odd $k \in \mathbb{Z}$, $C^{-1} = (D_{\text{even}} + D_{\text{odd}})^{-1}$.

Then, using Lemma 7.1.4:

$$\begin{aligned}
D_{\text{odd}} &= \sum_{k \in 2\mathbb{Z}+1} \binom{r-1}{\frac{k+1}{2} - 1} \cdot \binom{u-r-1}{\frac{k+1}{2} - 1} + \binom{r-1}{\frac{k+1}{2} - 1} \cdot \binom{u-r-1}{\frac{k+1}{2} - 1} \\
&= \sum_{k \in \mathbb{Z}} 2\binom{r-1}{k} \cdot \binom{u-r-1}{k} \\
&\overset{7.1.4}{=} 2\binom{u-r-1+r-1}{r-1} \\
&= 2\binom{u-2}{r-1}.
\end{aligned}$$

Furthermore

$$\begin{aligned}
D_{\text{even}} &= \sum_{k\in 2\mathbb{Z}} \binom{r-1}{\lceil\frac{k+1}{2}\rceil - 1}\cdot\binom{u-r-1}{\lfloor\frac{k+1}{2}\rfloor - 1} + \binom{r-1}{\lfloor\frac{k+1}{2}\rfloor - 1}\cdot\binom{u-r-1}{\lceil\frac{k+1}{2}\rceil - 1}\\
&= \sum_{k\in\mathbb{Z}} \binom{r-1}{k}\cdot\binom{u-r-1}{k-1} + \binom{r-1}{k-1}\cdot\binom{u-r-1}{k}\\
&\overset{7.1.4}{=} \binom{u-2}{u-r-1+1} + \binom{u-2}{r-1+1}\\
&= \binom{u-2}{u-r} + \binom{u-2}{r}\\
&= \binom{u-2}{u-2-(u-r)} + \binom{u-2}{r}\\
&= \binom{u-2}{r-2} + \binom{u-2}{r}.
\end{aligned}$$

Put together, we obtain

$$\begin{aligned}
C^{-1} &= (D_{\text{even}} + D_{\text{uneven}})^{-1}\\
&= \binom{u-2}{r-1} + \binom{u-2}{r-2} + \binom{u-2}{r-1} + \binom{u-2}{r}\\
&\overset{7.1.5}{=} \binom{u-1}{r-1} + \binom{u-1}{r}\\
&\overset{7.1.5}{=} \binom{u}{r}.
\end{aligned}$$

This concludes the proof. □

Putting everything together we arrive at the following equation for the probability distribution of the number of good edges:

Theorem 7.2.15 (Good Edge Distribution). *The number of good edges has the following probability distribution:*

$$\mathbb{P}(E=k) = \binom{n}{\alpha}^{-1}\left[\sum_{u=1}^{n}\mathbb{P}(U=u)\left\{\sum_{r=1}^{u-1}\binom{n-u}{\alpha-r}T_{u,r}(k) + \frac{\binom{\alpha}{u}+\binom{\beta}{u}}{\binom{n}{u}}\chi_{k=0}\right\}\right] \tag{7.2}$$

where $\chi_{k=0}$ is 1 if $k=0$ and 0 otherwise and

$$\mathbb{P}(U=u) = \binom{\binom{n}{2}}{u-1}\mathbb{P}(U_{12}=1)^{u-1}(1-\mathbb{P}(U_{12}=1))^{\binom{n}{2}-(u-1)}$$

with

$$\mathbb{P}(U_{12}=1) = \frac{1}{\sqrt{\pi}}\int_0^\infty \left(\Phi(p)^{n-2} + (1-\Phi(p))^{n-2}\right)e^{-p^2}\,dp.$$

Proof. It follows from basic probability theory that

$$\mathbb{P}(E=k)=\sum_{u=1}^{n}\sum_{r=0}^{u}\mathbb{P}(E=k\,|\,U=u,R=r)\mathbb{P}(R=r\,|\,U=u)\mathbb{P}(U=u).$$

Lemma 7.2.10 together with Lemma 7.2.8 and Proposition 7.2.6 provide the expression for $\mathbb{P}(U=u)$.

It is left to show that $\mathbb{P}(E=k\,|\,U=u,R=r)\mathbb{P}(R=r\,|\,U=u)$ takes the described form. But this follows from Proposition 7.2.14 and Lemma 7.2.12 after differentiating the cases $r\in\{0,u\}$ and $r\notin\{0,u\}$ since

$$\frac{\binom{\alpha}{r}\binom{\beta}{u-r}}{\binom{n}{u}\binom{u}{r}}=\frac{\binom{n-u}{\alpha-r}}{\binom{n}{\alpha}},$$

which can be confirmed by expanding the binomial coefficients on the left hand side and using elementary calculations. □

The most problematic term in Equation (7.2) is $\mathbb{P}(U=u)$, as it contains the integral. However, in the limit of $n\to\infty$, we can make approximate simplifications.

Definition 7.2.16. For two $\mathbb{R}$-valued functions f_n, g_n depending on a parameter $n\in\mathbb{N}$, we write $f_n\sim g_n$ if $\lim_{n\to\infty}\frac{f_n}{b_n}=1$.

Proposition 7.2.17 (Asymptotic Behavior of the Integral (Page 10 in [17]))**.** *In the limit of $n\to\infty$, the probability of edge (i,j) being in the upper convex hull behaves like*

$$\mathbb{P}(U_{ij}=1)\sim\frac{2\sqrt{2\pi\log n}}{n^2}$$

We obtain the following corollary:

Corollary 7.2.18. *The number of good edges has the following asymptotic distribution:*

$$\mathbb{P}(E=k)\sim\binom{n}{\alpha}^{-1}\left[\sum_{u=1}^{n}H(u)\left\{\sum_{r=1}^{u-1}\binom{n-u}{\alpha-r}T_{u,r}(k)+\frac{\binom{\alpha}{u}+\binom{\beta}{u}}{\binom{n}{u}}\chi_{k=0}\right\}\right]\tag{7.3}$$

where $\chi_{k=0}$ is 1 if $k=0$ and 0 otherwise and

$$H(u):=\binom{\binom{n}{2}}{u-1}\left(\frac{2\sqrt{2\pi\log n}}{n^2}\right)^{u-1}\left(1-\frac{2\sqrt{2\pi\log n}}{n^2}\right)^{\binom{n}{2}-(u-1)}$$

Not sure this corollary actually holds. Limit behavior also depends on the other terms depending on n, I'm pretty sure.

7.2.2 Application to Neural Networks

In Section 7.2.1 we have approached the problem of counting good edges in a two-dimensional dual-space, given a fixed number of red and blue points. To come back to the setting of neural networks, we will use the developed theory to assess the boundary complexity of a $2d$ ReLU binary classification network $f_\theta\colon \mathbb{R}^2 \to \mathbb{R}$ *without biases*. The latter restriction allows us to think of dual-space as being two- instead of three-dimensional.

In practice, we want to approach the problem from the other side. That is, given a neural network, we aim to compute its dcpa-representation and count the corresponding number of good edges. There is, however, a problem with this approach. While we do have a conjecture for the number of dual points in P and N (Conjecture 4.3.17), the assumptions that the red and blue points are drawn $i.i.d.$ is a simplification. One usually initializes the network weights using a Gaussian, which results in a more complex distribution of the sets P and N due to the complex recursive relations in Proposition 4.3.9.

Chapter 8

Empirical Decision Boundary Complexity

Algorithm 2 shows how we can empirically compute the boundary complexity of a ReLU binary classification network.

We compare the empirical complexity with our theoretical expectation derived above.

Algorithm 2 Empirical Boundary Complexity $\mathfrak{C}(f_\theta)$

```
Input: Neural network $f_\theta\colon \mathbb{R}^d \to \mathbb{R}$, sample $\mathbf{x} \in \mathbb{R}^d$
Output: Boundary complexity $\mathfrak{C}(f_\theta)$

Computing $\mathrm{dcpa}$ Representation
Initialize $P_0 = (\{(\mathbf{e}_1, 0), \ldots, (\mathbf{e}_d, 0)\})$ and $N_0 = (\emptyset)$
for layer $l = 1$ to $L$ do
    Decompose $\mathbf{W}_l$ into positive part $\mathbf{W}_l^+$ and negative part $\mathbf{W}_l^-$
    Compute $N_L = (\mathbf{W}_l^- \times P_{l-1}) + (\mathbf{W}_l^+ \times N_{l-1})$
    Compute $P_L = \big(((\mathbf{W}_l^+ \times P_{l-1}) + (\mathbf{W}_l^- \times N_{l-1})\big) \boxplus \mathbf{b}_l\big) \cup (N_l \boxplus t_l)$
end for

Comuting Boundary Complexity
Find the upper convex hull $\mathcal{U}(P_L \cup N_L)$
for each 1-cell (edge) in $\mathcal{U}(P_l \cup N_l)$ do
    if the 1-cell joins a point in $P_L$ to a point in $N_L$ then
        Mark as a boundary piece
    end if
end for
Return: Total boundary piece count
```

Bibliography

[1] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep double descent: Where bigger models and more data hurt," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2021, no. 12, p. 124003, 2021.

[2] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro, "The implicit bias of gradient descent on separable data," *Journal of Machine Learning Research*, vol. 19, no. 70, pp. 1–57, 2018.

[3] C. Zhang, Q. Liao, A. Rakhlin, K. Sridharan, B. Miranda, N. Golowich, and T. Poggio, "Musings on deep learning: Properties of sgd," Center for Brains, Minds and Machines (CBMM), Tech. Rep., 2017.

[4] Y. Zhang, A. M. Saxe, M. S. Advani, and A. A. Lee, "Energy–entropy competition and the effectiveness of stochastic gradient descent in machine learning," *Molecular Physics*, vol. 116, no. 21-22, pp. 3214–3223, 2018.

[5] K. Dingle, C. Q. Camargo, and A. A. Louis, "Input–output maps are strongly biased towards simple outputs," *Nature communications*, vol. 9, no. 1, p. 761, 2018.

[6] C. Mingard, H. Rees, G. Valle-Pérez, and A. A. Louis, "Do deep neural networks have an inbuilt occam's razor?" *arXiv preprint arXiv:2304.06670*, 2023.

[7] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[8] G. Valle-Perez, C. Q. Camargo, and A. A. Louis, "Deep learning generalizes because the parameter-function map is biased towards simple functions," *arXiv preprint arXiv:1805.08522*, 2018.

[9] P.-y. Chiang, R. Ni, D. Y. Miller, A. Bansal, J. Geiping, M. Goldblum, and T. Goldstein, "Loss landscapes are all you need: Neural network generalization can be explained without the implicit bias of gradient descent," in *The Eleventh International Conference on Learning Representations*, 2022.

[10] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[11] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.

[12] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations*, 2015, pp. 1–15.

[13] D. Maclagan and B. Sturmfels, *Introduction to tropical geometry*. American Mathematical Society, 2021, vol. 161.

[14] P. Piwek, A. Klukowski, and T. Hu, "Exact count of boundary pieces of relu classifiers: Towards the proper complexity measure for classification," in *Uncertainty in Artificial Intelligence*. PMLR, 2023, pp. 1673–1683.

[15] L. Zhang, G. Naitzat, and L.-H. Lim, "Tropical geometry of deep neural networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5824–5832.

[16] R. L. Graham, *Concrete mathematics: a foundation for computer science*. Pearson Education India, 1994.

[17] A. Rényi and R. Sulanke, "Über die konvexe hülle von n zufällig gewählten punkten," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 2, pp. 75–84, 1963.

Appendix A

Understanding the Tessellation

A.1 Counting Linear Pieces

Using the dual space and tessellations, one can also derive results on lower-order cells. Even though the statements in this section are not directly related to our main subject of study, we still want to provide them for completeness and because we think they are interesting.

It turns out to be useful to not only derive a tessellation from a cpa function, but to also use dcpa functions to endow $\mathbb{R}^d$ with a cell-structure.

Definition A.1.1. Given two CPA functions $\mathcal{Q}(P)$ and $\mathcal{Q}(N)$, we define $\mathcal{T}\left(\mathcal{Q}(P), \mathcal{Q}(N)\right)$ to consist of all pairwise non-empty intersections of cells from $\mathcal{Q}(P)$ and $\mathcal{Q}(N)$. That is,

$$\mathcal{T}\left(P, N\right) := \{C \cap D \,|\, C \in \mathcal{Q}(P), \;\; D \in \mathcal{Q}(N)\} \setminus \emptyset.$$

We still call elements of $\mathcal{T}\left(P, N\right)$ *cells*.

Proposition A.1.2. *Let $P, N \subseteq \mathfrak{D}$ be finite sets of dual points. Then the k-cells of $\mathcal{T}(P, N)$ are in one-to-one correspondence with the $(d-k)$-cells of $\mathcal{U}(P+N)$. Specifically, each k-cell C of $\mathcal{T}(P, N)$ is of the form*

$$\bigcup_{f \in \mathfrak{F}} \check{\mathcal{R}}(f)$$

where

$$\mathfrak{F} = \{f \in \mathrm{Aff}_{\mathfrak{D}}(\mathrm{d}) \,|\, f \| \mathcal{U}(P+N) \textit{ and } f \supseteq C' \textit{ for some } (d-k)-\textit{cell } C' \textit{ of } \mathcal{U}(P+N)\}.$$

As a consequence, we can easily count the number of affine pieces of a ReLU network:

Corollary A.1.3. *The number of affine pieces (d-cells) of a ReLU network $\mathcal{R}(P) - \mathcal{R}(N)$ is equal to the number of vertices of $\mathcal{U}(P+N)$.*