The Expressivity of Artificial Neural Networks

Candidate No. 1011879

University of Oxford

A thesis submitted for the degree of

Master of Science in
Mathematics and the Foundations of Computer Science

Trinity 2017

Abstract

Loosely inspired by the ultra-parallel architectures of animal brains, artificial neural networks are models of computation which take real input vectors and propagate them through networks of simple computational units. In recent years many-layer (or *deep*) artificial neural networks have achieved a high level of proficiency in solving a number of machine-learning problems, notably computer vision tasks. However, building and training these networks is still more art than science.

One major component of this art is the choice of what network architecture (in particular, what depth) to pick for training. Importantly, in this choice we must be careful to avoid underfitting, which in turn requires an understanding of the relationship between a network architecture and its expressivity; that is, the set of functions it is capable of representing. In this thesis we survey the fields' theoretical understanding of this relationship and extend some of these results with a view towards understanding depth. Along the way we will encounter larger questions about function composition and complexity measures that are of interest to a general audience.

Acknowledgements

I would first and foremost like to thank my supervisor Professor Varun Kanade for his ideas, excitement, and support. I would not be near my present level of understanding if weren't for his insight and enduring patience.

I am also very grateful to Arne Gouwy for crucial discussions regarding functional analysis, as well as to all my peers in our MFoCS cohort for their acuity, energy, and kindness.

And finally I thank my family, to whom I owe it all.

Contents

Introduction

Loosely inspired by the ultra-parallel architectures of animal brains, artificial neural networks (ANNs) are models of computation which take input vectors and propagate them through layers of affine transformations and simple nonlinearities. This thesis is concerned with understanding the relationship between the structure of a neural network and the functions it can implement. Motivations to study this relationship come from ANNs' connection to long-established research topics such as boolean circuits and approximation theory, but also (and perhaps more urgently) to recent developments in machine learning. In this chapter we discuss these motivations in detail and prepare the stage for the mathematics to follow.

What is an ANN?

An ANN consists of a network of computational units whose parameters are selected to determine a specific function. It will be useful for us to differentiate between the network *architecture*, an *instantiation* of the network where parameters have been selected, and the *implemented function* determined by this instantiation. The definitions we present here are based on those from early theoretical literature on the subject e.g., [17], [18], but have been simplified slightly to reflect practice at the time of this writing.

Definition 1. An *ANN architecture* $\mathcal{A}$ consists of

i. A directed acyclic graph (V, E), where we call a vertex a *network input* if it has no predecessors and a *network output* if it has no successors,

ii. For each non-input vertex, or *unit*, $v \in V$, an ordering of its predecessors $(w_1, \ldots, w_n)$,

iii. An ordering of input vertices $S = (s_1, s_2, \ldots, s_r)$ and an ordering of output vertices $K = (k_1, k_2, \ldots, k_t)$,

iv. An assignment to each unit v an *activation function* $\sigma_v : \mathbb{R} \to \mathbb{R}$.

In keeping with boolean circuit literature (e.g., [30]), we refer to the number of units of $\mathcal{A}$ as the *size* of $\mathcal{A}$. Architectures are also described in terms of *depth*—the length of the longest path from an input vertex to an output vertex—and *width*, a somewhat subtle measurement that corresponds roughly to the minimal memory required to compute the output of the neural network. (We defer the formal definition till Chapter 1).

Definition 2. An *(instantiated) artificial neural network* $\mathcal{N}$ consists of a neural architecture $\mathcal{A}$ and an assignment to each unit v in $\mathcal{A}$ a *weight vector* $\mathbf{w}_v \in \mathbb{R}^{i(v)}$ (where $i(v)$ is the indegree of v) and a *bias* $b_v \in \mathbb{R}$. Taken together over all units, these weights and biases are called the *programmable parameters* of $\mathcal{A}$.

Definition 3. Given a network $\mathcal{N}$ instantiated on an architecture with r inputs and m outputs, we say the network *implements* the function

$$\begin{aligned} \mathcal{N} : \mathbb{R}^r &\to \mathbb{R}^m \\ \mathbf{x} &\mapsto (y_{k_1}, y_{k_2}, \ldots, y_{k_m}), \end{aligned}$$

where for all units v,

$$y_v = \begin{cases} x_i & \text{if } v = s_i, \text{the } i^{\text{th}} \text{ input node,} \\ \sigma_v\big(\mathbf{w}_v \cdot (y_{u_1}, y_{u_2}, \ldots, y_{u_n}) + b_v\big) & \text{otherwise,} \end{cases}$$

where $\{u_i\}_{i=1}^n$ are v's predecessors and '$\cdot$' is the dot product.

In practice activation functions are typically simple nonlinearities like the rectifier $\max\{0, x\}$ (giving rise to *rectified linear units*, or ReLUs) or the sigmoid $1/(1+e^{-x})$. Neural network architectures also typically contain only a small number of different activation functions, often two or three. For example, most networks discussed here have the same nonlinear σ assigned to each non-output, or *hidden* unit, and then have the identity activation function assigned to the output units. For overviews of neural networks as used in practice, we refer the reader to [10] and [11].

This set of definitions describes the majority of popular network architectures, but it is worth noting that there are other paradigms not captured by it. In particular, [17] and others consider arbitrary n-variable polynomials within each σ_v, rather than affine transformations; Poon & Domingos [25] and others have investigates networks formed by units which compute sums or products of their inputs; and we will see in Chapter 3 a result that also holds for units which compute the max or min of their inputs.

Training Neural Networks

As computational networks ANNs can be considered an extension of traditional boolean circuits, and some of our investigations here can be seen as analogues of traditional questions asked about boolean circuits. However there is currently a far greater motivation for this work, as in comparison to boolean circuits ANNs possess the particular advantage of trainability: the parameters of a network architecture may be tuned to minimize the distance between the implemented function and a target function we desire to learn. Let us discuss this in greater formality.

Definition 4. Let K be a measurable subset of $\mathbb{R}^r$ and let μ be a probability measure defined on K. Suppose also that we have $f, f' : K \to \mathbb{R}^m$ with coordinates $(f_1, \ldots, f_m), (f'_1, \ldots, f'_m)$ respectively. Then we say f' (ε, L_p, μ)-approximates f if the quantity

$$\|f - f'\|_{L_p(\mu)} = \begin{cases} \Big(\sum_{i=1}^{m} \int |f_i - f'_i|^p \, \mathrm{d}\mu \Big)^{1/p} & \text{if } 1 \leq p < \infty \\ \max\limits_{1 \leq i \leq m} \sup\limits_{\mathbf{x} \in K} |f_i(\mathbf{x}) - f'_i(\mathbf{x})| & \text{if } p = \infty \end{cases}$$

is strictly less than ε. In the case that μ is the uniform distribution over a known K, we will omit μ from the notation.

The general setting of neural network training is as follows. Provided samples drawn from $f : \mathbb{R}^r \to \mathbb{R}^m$ according to some probability distribution μ, we first select an architecture $\mathcal{A}$. Next, we search for instantiation $\mathcal{N}$ thereof which (ε, L_p, μ)-approximates f. The search for such an instantiation happens over the following search space (or similar).

Definition 5 (Neural Function Class)**.** Given a neural network architecture $\mathcal{A}$, select a bound $B \in (0, \infty]$. Then we say the *ANN function class* $\mathcal{F}(\mathcal{A}, B)$ is the set of neural network functions derived by all possible assignments of weights $\mathbf{w}_i$ and biases b_i to each unit v_i such that $\|\mathbf{w}_i\|_{L_\infty} < B$ and $|b_i| < B$.

Note that when B and p are clear or we are discussing generally, we will denote a neural function class simply as $\mathcal{F}(\mathcal{A})$. We will also informally refer to the set $\mathcal{F}(\mathcal{A})$ as the *expressivity* of $\mathcal{A}$. For instance if $\mathcal{F}(\mathcal{A})$ is larger than $\mathcal{F}(\mathcal{B})$ by some measurement (e.g., it is a superset, or it has greater VC dimension—to be discussed in Chapter 2), we might say the former architecture is more expressive.

For a given neural network $\mathcal{N}$, we may concatenate all the vectors representing weights and biases into a single vector $\mathbf{W}$, say of length t. Doing this for all possible

assignments of weights and biases to $\mathcal{A}$ that meet the requirements in Definition 5, we may write the resulting collection $\{\mathbf{W}\}$ as a subset S of $\mathbb{R}^t$. We therefore have a map $F : S \to \mathcal{F}(\mathcal{A})$ which parametrizes the neural function class. Hence for $\mathbf{W} \in S$, the task of neural network training takes the form of minimizing the loss function $L(\mathbf{W}) = \|F(\mathbf{W}) - f\|_{L_p(\mu)}$. As long as the activation functions in $\mathcal{A}$ are differentiable, so is F (and by extension L) and this optimization problem is amenable to iterative optimization techniques such as stochastic gradient descent. Again, we refer the interested reader to [10] and [11] for more thorough treatments of neural network training.

Thanks to their trainability, ANNs find great success in the present day as machine learning algorithms, comfortably sitting in first place for computer vision tasks (e.g., [28]) and finding use in data compression (e.g., [19]) and medical diagnosis [1]. That being said, the design of ANNs and their associated training algorithms remains more of an art than a science. For any nontrivial ANN architecture $\mathcal{A}$, the loss surface defined by $L(\mathbf{W})$ is highly non-convex and as a result $\mathcal{A}$ may fail to learn an approximation to the target function f within a reasonable number of samples. This results in bad predictions on unseen data, characterized by a high *generalization error*. Assuming enough data is available for learning to be possible, this failure is due to one of two problems:

> **Problem 1.** $\mathcal{F}(\mathcal{A})$ does not contain an (ε, L_p)-approximation to f (a problem known generally as *underfitting*).
>
> **Problem 2.** $\mathcal{F}(\mathcal{A})$ does contain an (ε, L_p)-approximation to f, but the probability that the learning algorithm finds this approximation in a reasonable amount of time is too low. Here the optimization algorithm may repeatedly get stuck in local minima, or 'bad wells' (and thus must try optimizing from a different starting point), or *overfit* by introducing noise into certain programmable parameters.

Observe that the behavior of the training algorithm can only ever cause problem 2. On the other hand, the relationship between $\mathcal{F}(\mathcal{A})$ and the structure of $\mathcal{A}$ is connected to both: if a network is too small it may not contain an appropriate approximation of f; whereas if it is too large or yields a particularly unfavorable loss surface, convergence to an approximation of f with high probability is impossible. Therefore understanding the map $\mathcal{A} \mapsto \mathcal{F}(\mathcal{A})$ is of great practical interest.

While it is often unclear which the problems above plague a given ANN, we note that the task of understanding how the structure of $\mathcal{A}$ affects $L(\mathbf{W})$ appears to be

slightly out of reach at the time of this writing. Indeed, to address the architecture's role in Problem 2, researchers have turned to approximations by ensemble methods from statistical physics as well as direct experimentation to say anything of use (e.g., [5]) Thus while inadequate network expressivity is the simpler of the two ways network architecture can obstruct the learning process, we will focus on it in the present work, appreciating that results are still hard-won.

Underfitting in the context of deep learning

Suppose we wish to select a network architecture to achieve an approximation of a target function f and additionally we have the following information:

- A set of activation functions we intend to use in our network (known as our *gate set* or *basis*),
- Properties of f such as its smoothness, its Lipschitz coefficient, or number of extrema.

We would like to find 'rules of thumb' which take this information and output recommendations for the structure (bounds on size, depth, width, or similar properties) of $\mathcal{A}$ so that the membership in $\mathcal{F}(\mathcal{A})$ of an approximation to f is guaranteed. Let us call these sorts of results *underfitting avoidance strategies*, or *UAS*s.

A starting point for a UAS might be inferred from the following theorem, proved for many different sorts of activation functions and network types (we will see two such proofs in Chapter 1) around the year 1990:

Theorem (Universal Approximation Theorem)**.** *Let K be a compact subset of $\mathbb{R}^r$. Then for any continuous function $f : K \to \mathbb{R}$ and any $\varepsilon > 0$, there exists a depth-two neural network which (ε, L_∞)-approximates f.*

This result suggests one way to avoid underfitting: start by trying to train a small neural network of depth two, and if that doesn't work, add some units to the hidden layer; repeat until success is achieved. In Chapter 2 we'll even see bounds on how wide one must go to achieve this success for certain restricted classes of functions. In summary, one need not consider networks of depth three or greater.

The picture today looks quite different, however: thanks to the growth of 'Big Data', sophisticated training algorithms for many-layer networks, and better GPUs for putting it all together, many of the successful ANNs used in practice today are of far greater depth than pure reliance on the universal approximation theorem would

Year	Name	Error	Depth
2011	—	25.8%	2
2012	AlexNet	16.4%	8
2013	—	11.7%	8
2014	GoogleNet	6.7%	22
2015	ResNet	3.6%	152

Table 1: A table of recent depths for successful image classification networks in the ILSVRC competitions [12]. Error here is the "Top 5 Error," meaning the percentage of images in the test set whose correct classification was not in the top five most likely candidates proposed by the network.

engender. (Indeed, the buzzword "deep learning" refers specifically to neural networks of depth at least three.)

For example, consider Table 1, a list of depths of successful networks entered in the ImageNet Large Scale Visual Recognition Competition in recent years. The relationship between the usefulness of a neural network and its structure is clearly more subtle than the Universal Approximation Theorem. But does this turn towards deep learning have anything to do with the desire to avoid underfitting?

Perhaps depth doesn't really improve expressivity, but rather makes the curvature of the loss surface more amenable to training. This would mean that our search for UASs isn't very relevant to the choice of deep learning architecture because they merely give lower bounds. But there is evidence this isn't the case, and that part of the reason to turn towards deeper networks may be to gain in expressivity. In particular, as we'll see in Chapter 3, there are certain functions that benefit greatly from depth in the following sense. Any ε-approximation to f of depth less than d requires exponentially-many units (in some property of the ANN such as input dimension, depth, or similar), while there exist certain networks of depth greater than d requiring only polynomially-many units to ε-approximate f.

Thus we have at least some cases in which expressivity is provably improved by increasing depth. But there's another reason our rules of thumb might be irrelevant: maybe deep networks, with their millions of parameters, brute-force their way through Problem 1. That is, perhaps deep networks are *so* deep that they can *always* guarantee f is in the ε-neighborhood of $\mathcal{F}(\mathcal{A})$. It is hard to know whether this is true in practice, but even if it is there is still reason to seek our rules of thumb.

As an illustration, Zhang et al. [36] found that some networks which are very successful at image classification can also happily memorize random noise, implying

that the networks are capable of drastic overfitting. The reason they don't do this for image classification is an interesting mystery in its own right, but this result serves to suggest that there may be other learning problems in which depth is not a catch-all solution. Thus even in the context of deep learning, it's useful to have UASs because they stand as approximate lower bounds for the size of the network to use, and thereby may also guard against overfitting.

This thesis

We've just seen that the development of UASs, and in particular those which recognize the importance of depth, is a worthy research direction even in the context of deep ANNs. Yet we must be cautious about what goals we set: in order to generate such recommendations, we must first understand with greater precision the relationship between $\mathcal{F}(\mathcal{A})$ and the structure of $\mathcal{A}$, a task that sits at the very edge of the our mathematical abilities.

Moreover, there are computational results that set limits on the power of the UASs we might eventually develop. In particular, in [14] (Appendix B) Judd proves the NP-completeness of the following decision problem:

> Given an ANN architecture $\mathcal{A}$ with activation functions that are nonlinear and bounded, a sample of inputs $S \subseteq \{0,1\}^r$, and a function $f : \{0,1\}^R \to \{0,1\}^m$, is there an $\mathcal{N} \in \mathcal{F}(\mathcal{A}, \infty)$ which $(0.1, L_\infty)$-approximates any $g : \{0,1\}^r \to \{0,1\}^m$ with $g|_S = f|_S$?

Similarly, [4] prove that determining exact membership (the answer to the question, "Is f in $\mathcal{F}(\mathcal{A})$?") is NP-complete for threshold networks (that is, using activation functions $\sigma(x) = \text{sgn}(x)/2 + 1/2$) that implement functions on the boolean cube. Hence a 'Holy Grail' UAS which could tell us with certainty the approximability of f by $\mathcal{F}(\mathcal{A})$ is out of reach, or at least it seems any such algorithm would be little better than brute force guessing-and-checking.

However, these results do not imply that useful bounds cannot be gleaned for more restricted classes of functions, or that approximate, probabilistic UASs are impossible. And not all hope is lost; we'll see at that end that if we already know something about the structure of the function we're attempting to learn, then we can use this knowledge as a "structural prior" to gain information about required network size.

With this in mind, we set ourselves to the task of surveying the present understanding of expressivity. In particular, we will explore the following questions:

- What classes $\mathcal{C}$ of networks architectures are universal approximators of continuous functions over compact subspaces of $\mathbb{R}^r$? That is, under what conditions is the closure of a collection $\mathcal{C}$ of ANN architectures equal to $C[K]$ with K a compact subset of $\mathbb{R}^r$?
- If some information is known about f, what upper and lower bounds can we put on the size of $\mathcal{A}$ such that $\mathcal{F}(\mathcal{A})$ contains an ε-approximation to f?
- Are there function classes which obtain a substantial improvement in representation efficiency (minimum size of network) from depth? From width?
- Can information about a decomposition of f lead to an efficient ANN approximation thereof?
- And how much do all of these depend on the particular activation function being used?

We opt to give proofs for a number of results in their full because the techniques used therein are not only interesting in their own right, but also demonstrate the present limitations of our mathematical ability to understand these complex systems. We will also present a number of extensions of these results inspired by the questions above, and with a view towards gaining insight into the relationship between depth and expressivity.

Organization of this thesis

Chapter 1: Universality. We give two proofs of the universal approximation theorem that highlight different proof techniques. We then extend the result to networks with fixed width rather than depth.

Chapter 2: Approximation Bounds. We present a result of Barron that gives explicit network size upper bounds for a certain restricted class of functions, including polynomials. We then present lower bounds for network size inspired in one case by an early Shannon result, and in the other case based on VC-dimension bounds.

Chapter 3: Exponential separations. We examine function classes that admit efficient deep representations yet cannot be efficiently approximated by shallow networks. We also examine the role a particular choice of activation function plays.

Conclusion. Here we take a conceptual overview of the results discussed, concluding with a list of open problems.

Chapter 1

Universality Results

There are a number of versions of the Universal Approximation Theorem (UAT) in the literature (see [31] for a survey); we will see two proofs which use different techniques at different levels of abstraction.

In this chapter we consider neural networks with r inputs and one output, where the hidden units all use identical activation functions and the output unit uses the identity activation function. The universality theorems we show here don't require a bias term on the output unit, so we will leave it out of our notation. Thus the depth-two ANNs of this kind have the explicit form

$$\sum_{i=1}^{N} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$$

for some N, weights $\{\mathbf{w}_i\}$ and $(a_1, \ldots, a_N)$, and biases $\{b_i\}$. Define

$$\mathbf{N}^r_{\bullet \times 2} = \bigcup_{N=1}^{\infty} \left\{ \sum_{i=1}^{N} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i) \,\middle|\, a_i, b_i \in \mathbb{R}, \mathbf{w}_i \in \mathbb{R}^r \right\}.$$

As we will see shortly, the relative simplicity of shallow networks' functions allow us to use standard technique from functional analysis to analyze them.

1.1 Cybenko's Universality Theorem

Cybenko [6] proves the UAT for networks with activation functions from the following class:

Definition 6 (Sigmoidal functions)**.** A continuous function $\sigma : \mathbb{R} \to \mathbb{R}$ is *sigmoidal* if

$$\lim_{x \to -\infty} \sigma(x) = 0 \text{ and } \lim_{x \to \infty} \sigma(x) = 1.$$

He proves the UAT for target functions $I^r \to \mathbb{R}$ where $I = [0, 1]$; generalization to arbitrary compact subsets of $\mathbb{R}^r$ is straightforward. The basic structure of Cybenko's proof is this: first we prove the UAT for a more general but somewhat opaque class of activation functions, then secondly we show that sigmoidal functions belong to this class. Here $M(X)$ denotes the set of all finite, signed regular Borel measures on a space X. For notational convenience we will also define $\sigma_{\mathbf{w},b}(\mathbf{x}) := \sigma(\mathbf{w} \cdot \mathbf{x} + b)$.

Definition 7 (Discriminatory Function)**.** A function $\sigma : \mathbb{R} \to \mathbb{R}$ is *discriminatory* if for all $\mu \in M(I^n)$,

$$\left(\begin{array}{c} \displaystyle\int_{I^n} \sigma_{\mathbf{w},b} \, \mathrm{d}\mu = 0 \\ \text{for all } \mathbf{w} \in \mathbb{R}^n \text{ and } b \in \mathbb{R} \end{array} \right) \quad \Longrightarrow \quad \begin{array}{c} \mu = \mathbf{0}. \\ \text{(the trivial measure).} \end{array}$$

For a descriminatory activation function σ, Cybenko proves the density of $\mathbf{N}^r_{\bullet \times 2}$ in $C(\mathbb{R})$ by showing any functional annihilated on its closure $\overline{\mathbf{N}^r_{\bullet \times 2}}$ is annihilated everywhere.

Theorem 1.1 (G. Cybenko, 1989)**.** *Let σ be a continuous discriminatory function. Then for discriminatory activation functions, $\mathbf{N}^r_{\bullet \times 2}$ is dense in $C[I^n]$.*

Proof. Clearly $\mathbf{N}^r_{\bullet \times 2}$ is a linear subspace of $C[I^n]$. Let $\overline{\mathbf{N}^r_{\bullet \times 2}}$ be the closure of $\mathbf{N}^r_{\bullet \times 2}$. We claim that any bounded linear functional that is zero on $\overline{\mathbf{N}^r_{\bullet \times 2}}$ must be zero on all of $C[I^n]$; this proves the result by the Hahn-Banach theorem (see e.g., [27], Theorem 5.19.)

To that end, suppose L is a functional on $C[I^n]$ such that $L(\overline{\mathbf{N}^r_{\bullet \times 2}}) = \{0\}$. By the Riesz Representation Theorem there is some $\mu \in M(I^n)$ such that for all $h \in C[I^n]$,

$$L(h) = \int_{I^n} h \, \mathrm{d}\mu.$$

In particular, because $L(\overline{\mathbf{N}^r_{\bullet \times 2}}) = \{0\}$ we have

$$L(\sigma_{\mathbf{w},b}) = \int_{I^n} \sigma_{\mathbf{w},b} \, \mathrm{d}\mu = 0$$

for all $\mathbf{w}$ and b. Because σ is discriminatory $\mu = \mathbf{0}$, and so $L = \mathbf{0}$. □

This proof is curiously simple. We appear to gain a lot of power from the definition of "discriminatory" so just how realistic is this condition? As it turns out, the answer is "quite realistic!" Here's where the rest of the difficulty of this universality result is hidden:

Lemma 1.2. *Bounded, measurable sigmoidal functions are discriminatory.*

Proof. Suppose σ is a bounded measurable sigmoidal function and μ is a measure on I^n such that for all $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$,

$$\int_{I^n} \sigma_{\mathbf{w},b} \, \mathrm{d}\mu = 0.$$

We will show μ is the trivial measure by first showing $\mu(H) = 0$ for every halfspace in I^n, and then showing that this condition implies μ is identically 0.

To begin, fix φ and define $\sigma_{\lambda,\mathbf{w},b}(\mathbf{x}) = \sigma(\lambda(\mathbf{w} \cdot \mathbf{x} + b) + \varphi)$. Then for all $\mathbf{w}, b$ we have

$$\int_{I^n} \sigma_{\lambda,\mathbf{w},b} \, \mathrm{d}\mu = \int_{I^n} \sigma\big((\lambda\mathbf{w}) \cdot \mathbf{x} + (\lambda b + \varphi)\big) \, \mathrm{d}\mu(\mathbf{x}) = 0. \tag{1.1}$$

Observe that for any $\mathbf{x}, \mathbf{w}, b$ we have the pointwise (along $\mathbf{x}$) bounded convergence

$$\lim_{\lambda \to +\infty} \sigma_{\lambda,\mathbf{w},b}(\mathbf{x}) = \left\{ \begin{array}{ll} 1 & \text{for } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0 & \text{for } \mathbf{w} \cdot \mathbf{x} + b < 0 \\ \sigma(\varphi) & \text{for } \mathbf{w} \cdot \mathbf{x} + b = 0 \end{array} \right\} =: \gamma_{\mathbf{w},b}(\mathbf{x}).$$

Then by the Lesbegue Bounded Convergence Theorem, we have for all φ, b, and $\mathbf{w}$,

$$\begin{aligned} \int_{I^n} \gamma_{\mathbf{w},b} \, \mathrm{d}\mu &= \lim_{\lambda \to +\infty} \int_{I^n} \sigma_{\lambda,\mathbf{w},b}(x) \, \mathrm{d}\mu(\mathbf{x}) \\ &= 0 \text{ by (1.1).} \end{aligned} \tag{1.2}$$

Let $\Pi_{\mathbf{w},b}$ be the hyperplane defined by $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = 0\} = \gamma^{-1}(\sigma(\varphi))$ and let $H_{\mathbf{w},b}$ be the open half-space defined by $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b > 0\} = \gamma^{-1}(1)$. Then by (1.2) we also have for all $\mathbf{w}$ and b

$$\int_{I^n} \gamma_{\mathbf{w},b} \, \mathrm{d}\mu = \sigma(\varphi)\mu(\Pi_{\mathbf{w},b}) + \mu(H_{\mathbf{w},b}) = 0.$$

So far φ has been arbitrary, so the above implies:

$$\begin{aligned} &\lim_{\varphi \to -\infty} \sigma(\varphi)\mu(\Pi_{\mathbf{w},b}) + \mu(H_{\mathbf{w},b}) = 0 \\ &\qquad \implies \mu(H_{\mathbf{w},b}) = 0 \text{ by definition of } \sigma. \end{aligned}$$

But this is true for all $\mathbf{w}, b$, so indeed $\mu(H) = 0$ for every half-space H. It remains to show a finite signed measure with this property is identically zero. This is a general fact which we leave for Lemma 1.3. $\square$

A word about the strategy behind the definition of $\sigma_{\lambda,\mathbf{w},b}(\mathbf{x})$: multiplying the input to σ by λ and then taking it to ∞ has the effect of 'squeezing' σ all the way to look like a Heaviside step function with singular value $\sigma(\varphi)$. Because σ is continuous, the intermediate value theorem says that we can set $\sigma(\varphi)$ to any value between 0 and 1 we want, depending on our choice of φ. In particular, sending $\varphi \to -\infty$ simplifies our integral even further. Seen this way, Lemma 1.2 proves the expressivity of sigmoidal functions by reducing them to threshold functions.

Lemma 1.3. *Suppose μ is a finite signed measure on I^n. Then if $\mu(H) = 0$ for all half-spaces H in I^n, $\mu = \mathbf{0}$.*

Proof. Fix $\mathbf{w} \in I^n$. For a bounded measurable function $h : \mathbb{R} \to \mathbb{R}$, define the linear functional $F(h)$ by

$$F(h) = \int_{I^n} h(\mathbf{w} \cdot \mathbf{x}) \, \mathrm{d}\mu(\mathbf{x}).$$

Observe that F is a bounded functional on $L^\infty(\mathbb{R})$ because μ is finite signed. Now set $h_b = \mathbb{1}_{[b,\infty)}$, so $h_b(\mathbf{w} \cdot \mathbf{x})$ ranges over all halfspaces as $b, \mathbf{w}$ change. Then

$$F(h_b) = \int_{I^n} h_b(\mathbf{w} \cdot \mathbf{x}) \, \mathrm{d}\mu = \mu\big(h_b^{-1}(1)\big) = 0$$

by hypothesis. The same is true if $h_b = \mathbf{1}_{(b,\infty)}$. By linearity, this means $F(h) = 0$ for the indicator of any interval and hence by Fubini's Theorem $F(h) = 0$ for any simple function. Simple functions are dense in $L^\infty(\mathbb{R})$, so $F = \mathbf{0}$.

In particular, using the bounded measurable functions $s(\mathbf{x}) = \sin(\mathbf{m} \cdot \mathbf{x})$ and $c(\mathbf{x}) = \cos(\mathbf{m} \cdot \mathbf{x})$, we have

$$F(s + ic) = \int_{I^n} \cos(\mathbf{m} \cdot \mathbf{x}) + i \sin(\mathbf{m} \cdot \mathbf{x}) \, \mathrm{d}\mu(\mathbf{x}) = \int_{I^n} e^{i\,\mathbf{m}\cdot\mathbf{x}} \, \mathrm{d}\mu(\mathbf{x}) = 0$$

for all $\mathbf{m}$. Hence the Fourier transform of μ is 0 and so μ is itself 0, as desired. □

Techniques used here are highly nonconstructive, so Cybenko's proof does not easily lend itself to exploring approximation bounds or other extensions. The next proof we'll see is much more explicit, but loses the elegance afforded by the Hahn-Banach and Riesz Representation theorems.

Before we move on, note that we may easily generalize this theorem to networks built with ReLUs:

Corollary 1.3.1. $\mathbf{N}^r_{\bullet\times 2}$ *with ReLUs in the hidden layer is also a universally approximating class.*

Proof. Let $\sigma(x) = \max\{0, x\}$ and observe that $\delta(x) = \sigma(x) - \sigma(x-1)$ is sigmoidal. Apply Cybenko's UAT for networks with activation function δ. Thus for every f and any measure μ we have an ε-approximation $\bar{f}$ to f by a network of the following form:

$$\bar{f} = \sum_{i=1}^{N} a_i \delta(\mathbf{w}_i \cdot \mathbf{x} + b_i)$$

for some N, weights $\{\mathbf{w}_i\}$ and $(a_1, \ldots, a_N)$, and biases $\{b_i\}$. Substituting for δ, we have

$$\begin{aligned}\bar{f} &= \sum_{i=1}^{N} a_i\big(\sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i) - \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i - 1)\big) \\ &= \sum_{i=1}^{N} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i) + \sum_{i=1}^{N} (-a_i)\sigma\big(\mathbf{w}_i \cdot \mathbf{x} + (b_i - 1)\big).\end{aligned}$$

This last line defines a new neural network with ReLUs in the hidden layer that exactly implements $\bar{f}$, as desired. □

1.2 Funahashi's Universality Theorem

Funahashi's work [9] is based on a result of Irie-Miyake [13] which proves the representability of continuous functions by a network with a continuum of hidden nodes. Funahashi shows the approximability of this continuum-network by networks of finite size, achieving an approximate representation that is significantly more constructive than Cybenko's. Their combined work proceeds as follows: we begin with a Fourier integral representation of the target function, replace the exponential with a Fourier representation of a function derived from sigmoidal functions, and then take the multivariate Riemann sum to produce a convergent series of approximations to the target function.

Funahashi also works with a different class of activation functions, though this class does contain certain sigmoidal functions including the traditional sigmoid.

Theorem 1.4 (Funahashi, 1989)**.** *Let K be a compact subset of $\mathbb{R}^r$ on which a continuous real function f is defined and suppose $\sigma : \mathbb{R} \to \mathbb{R}$ is an bounded, increasing, and non-constant function. Then there exists a two layer ANN using activation function σ which (ε, L_∞)-approximates f.*

Proof. Note that any continuous f can be extended to a continuous function f' on $\mathbb{R}^r$ with compact support, and that the set of C^∞ functions with compact support

is dense in the set of continuous functions on $\mathbb{R}^r$ with compact support. Hence we assume without loss of generality that f is C^∞ on $\mathbb{R}^r$ with compact support. By the Paley-Weiner theorem, the Fourier transform F of f is real analytic and for any integer N, there is a constant C_N such that

$$|F(\boldsymbol{\omega})| \leq C_N(1+|\boldsymbol{\omega}|)^{-N}.$$

Define as our first approximation of f the function

$$f_A = \frac{1}{(2\pi)^r}\int_{[-A,A]^r} F(\boldsymbol{\omega})e^{i\boldsymbol{\omega}\cdot\mathbf{x}}\,\mathrm{d}\boldsymbol{\omega}$$

and note that the modulus of the error is bounded as

$$|f_A(\mathbf{x}) - f(\mathbf{x})| \leq \frac{1}{(2\pi)^r}\int_{\mathbb{R}^r\setminus[-A,A]^r} |F(\boldsymbol{\omega})|\,\mathrm{d}\boldsymbol{\omega},$$

which is independent of $\mathbf{x}$ and goes to zero as $A\to\infty$ by (1.2). Hence f is uniformly approximated by f_A as as $A\to\infty$. Let $\varepsilon>0$ be given and fix A such that $\|f-f_A\|_\infty < \varepsilon/2$.

Now let ψ be a function in $L^1(\mathbb{R})$ with Fourier transform Ψ such that $\Psi(1)\neq 0$. Define g as follows and observe the equality:

$$\begin{aligned} g(\mathbf{x}) = \int_{-\infty}^{\infty}\psi(\mathbf{x}\cdot\mathbf{w}-\omega_0)e^{i\omega_0}\,\mathrm{d}\omega_0 &= \int_{-\infty}^{\infty}\psi(\omega)e^{i(\boldsymbol{\omega}\cdot\mathbf{x}-\omega)}\,\mathrm{d}\omega \\ &= e^{i(\boldsymbol{\omega}\cdot\mathbf{x})}\int_{-\infty}^{\infty}\psi(\omega)e^{-i\omega}\,\mathrm{d}\omega \\ &= \Psi(1)e^{i(\boldsymbol{\omega}\cdot\mathbf{x})}. \end{aligned}$$

We now aim to replace the exponential in f_A with an approximation of $g(\mathbf{x})/\Psi(1)$. Define

$$\begin{aligned} g_B(\mathbf{x}) &= \int_{-B}^{B}\psi(\mathbf{x}\cdot\mathbf{w}-\omega_0)e^{i\omega_0}\,\mathrm{d}\omega_0 \\ &= e^{i\mathbf{x}\cdot\mathbf{w}}\int_{\mathbf{x}\cdot\mathbf{w}-B}^{\mathbf{x}\cdot\mathbf{w}+B}\psi(t)e^{it}\,\mathrm{d}t \end{aligned}$$

and observe that

$$|g_B(\mathbf{x}) - g(\mathbf{x})| \leq \int_{-\infty}^{\mathbf{x}\cdot\mathbf{w}-B}|\psi(t)|\,\mathrm{d}t + \int_{\mathbf{x}\cdot\mathbf{w}+B}^{\infty}|\psi(t)|\,\mathrm{d}t.$$

This bound goes to zero as $B \to \infty$, so g_B converges uniformly to g as $B \to \infty$. Fix B such that $\|g - g_B\| \leq \varepsilon\Psi(1)/2$. Now define

$$\begin{aligned} f_{A,B} &= \frac{1}{\Psi(1)(2\pi)^r} \int_{[-A,A]^r} F(\boldsymbol{\omega}) g_A(\mathbf{x}) \, \mathrm{d}\boldsymbol{\omega} \\ &= \frac{1}{\Psi(1)(2\pi)^r} \int_{[-A,A]^r} \int_{-B}^{B} F(\boldsymbol{\omega}) e^{i\omega_0} \psi(\mathbf{x} \cdot \mathbf{w} - \omega_0) \, \mathrm{d}\omega_0 \, \mathrm{d}\boldsymbol{\omega} \end{aligned}$$

and observe that for all $\mathbf{x}$ we have $|f_{A,B}(\mathbf{x}) - f_A(\mathbf{x})| \leq \varepsilon/2$. Hence by the triangle inequality, $\|f - f_{A,B}\|_\infty < \varepsilon$ for appropriate choices of A and B.

Now replace the integral in $f_{A,B}$ with a Riemann sum approximation. The function being integrated is continuous over a compact set and is thus uniformly continuous (e.g., [26] Thm. 4.19). Hence with sufficiently small intervals, a Riemann sum approximation converges uniformly to $f_{A,B}$. We have therefore shown that for any continuous $f : K \to \mathbb{R}$, there exist $\{a_i\}, \{\mathbf{w}_i\}, \{b_i\}, n$ such that

$$\left\| f(\mathbf{x}) - \sum_{i=1}^{n} a_i \psi(\mathbf{w}_i \cdot \mathbf{x} + b_i) \right\|_\infty \leq \varepsilon.$$

Finally, by Lemma 1.5 below, we may substitute each ψ for a linear combination of σs, as desired. □

Lemma 1.5. *Let σ be an bounded, increasing, and nonconstant function. Then for all $\alpha > 0$ there exists a $\delta > 0$ such that*

$$\psi(x) = \sigma(x/\delta + \alpha) - \sigma(x/\delta - \alpha)$$

is in $L^1(\mathbb{R})$ and $\Psi(1) \neq 0$.

Proof. σ is bounded so there is an M which bounds $|\psi|$. By definition ψ is positive, so for all $L > M, \alpha, \delta > 0$,

$$\begin{aligned} \int_{-L}^{L} |\psi| \, \mathrm{d}x &= \int_{-L}^{L} \psi \\ &= \int_{\delta(-L+\alpha)}^{\delta(L+\alpha)} \sigma(x) \, \mathrm{d}x - \int_{\delta(-L-\alpha)}^{\delta(L-\alpha)} \sigma(x) \, \mathrm{d}x \\ &= \int_{\delta(L-\alpha)}^{\delta(L+\alpha)} \sigma(x) \, \mathrm{d}x - \int_{\delta(-L-\alpha)}^{\delta(-L+\alpha)} \sigma(x) \, \mathrm{d}x \leq \frac{4\alpha M}{\delta}. \end{aligned}$$

And, taking the limit, we see $\psi \in L^1(\mathbb{R})$.

Now suppose there does not exist a $\delta > 0$ for which $\Psi(1) \neq 0$. Then for all δ

$$\begin{aligned} 0 &= \int_{-\infty}^{\infty} (\sigma(x/\delta - \alpha) - \sigma(x/\delta + \alpha))e^{-ix}\,\mathrm{d}x \\ &= \int_{-\infty}^{\infty} (\sigma(x - \alpha) - \sigma(x + \alpha))e^{-ix\delta}\,\mathrm{d}x \\ &= \int_{-\infty}^{\infty} (\sigma(x - \alpha) - \sigma(x + \alpha))e^{ix\delta}\,\mathrm{d}x. \end{aligned}$$

The fact that $\psi \in L^1(\mathbb{R})$ implies Ψ is continuous, so by the above it must also be identically 0. Hence $\sigma(x + \alpha) - \sigma(x - \alpha) = 0$ for all x, α. But this contradicts the given that σ is not constant. □

1.3 Linear-width Universality

Observe how specific the techniques used above are to the depth-two regime. Analyzing function composition is very difficult, and while depth-two ANNs do have some function composition in their expressions (*i.e.,* a linear combination of nonlinear mappings of affine transformations), we were able to cope with them because these few compositions have special properties. In Cybenko's case, sigmoidal functions could be squeezed into indicators on arbitrary halfspaces, while in Funahashi's case we matched the affine transformation inside the activation function with the exponential in the Fourier transform of f. It is unclear how these techniques might generalize to shed light on deeper networks. The difficulty of analyzing deep networks is certainly a recurring theme in this thesis, though we now present a rare opportunity to say something about the approximation capabilities of deep networks.

In particular we may ask a dual question address above: if we allow unrestricted depth, what is the minimum width required of a network class to be a universal approximator? The answer is a width asymptotically equal to the input dimension r, it it comes fairly quickly as an extension of the depth-two UAT.

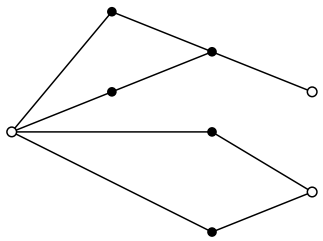
To prove this result we must first define width—a task which is surprisingly tedious because naive attempts at a definition lack monotonicity. That is, the property that $\mathcal{N} \subseteq \mathcal{W} \implies w(\mathcal{N}) \leq w(\mathcal{W})$, where $\subseteq$ is defined as follows.

Definition 8 (Subnetwork)**.** Suppose architecture $\mathcal{A}$ has graph (V, E) with input vertices S and output units K and architecture $\mathcal{B}$ has graph (V', E') with input vertices S' and output units K'. If $V' \subseteq V, E' \subseteq E, S' \subseteq S$, and $K' \subseteq K$, we say $\mathcal{B}$ is a *subnetwork* of $\mathcal{A}$ and write $\mathcal{B} \subseteq \mathcal{A}$.

A definition of width from boolean circuit theory—in particular from Pippenger [22]—is our starting point.

Definition (Width′)**.** Let the *level* of vertex v be the maximum length of a path from a source node to v. Let the *thickness* of a level ℓ be the number of vertices at levels not exceeding ℓ which share an edge with a vertex in a level exceeding ℓ. Then the *width′* of the network is the maximum thickness over all levels in the graph.

However, this definition fails to accurately capture our intuition of resource use. Consider the network

$\mathcal{N} =$ 

The width′ of $\mathcal{N}$ is four. Yet by delaying the computation of the bottom two black vertices till the second "layer" as suggested by the diagram, we can get away with holding at most three values in memory at any particular time. Motivated by this example, we present the definition of width that we will use here.

Definition 9 (Width)**.** We say an ordered partition $\mathcal{P} = (\ell_0 < \ell_1 < \ldots < \ell_m)$ of $V(\mathcal{N})$ is a *level partition* if, viewing $\mathcal{N}$ as a poset,

1. $S(\mathcal{N}) = \ell_0$ and $K(\mathcal{N}) = \ell_m$,
2. For all $\ell \in \mathcal{P}$, for all $v_1, v_2 \in \ell$, $v_1 \not\leq v_2$ and $v_2 \not\leq v_1$,
3. For all $i < j$, for all $v \in \ell_i, v' \in \ell_j$, $v \not\geq v'$.

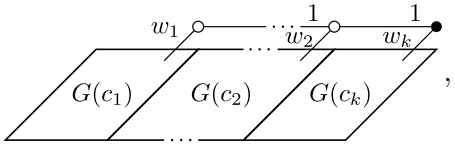
We say the $\mathcal{P}$-*width* of $\mathcal{N}$ is the maximum thickness (in the Pippenger sense) over all $\ell \in \mathcal{P}$, and define the *width* of $\mathcal{N}$ to be the minimum $\mathcal{P}$-width over all possible level partitions $\mathcal{P}$ of $\mathcal{N}$.

Our result about small-width universal classes follows directly from the depth-two universality theorem and the following method for "massaging" wide networks into deep and narrow ones. The construction is given for networks of with one output unit, though it may be easily generalized. We note also that a similar construction for boolean circuits is given in [35].

Theorem 1.6 (Width Reduction)**.** *Suppose a network $\mathcal{N} : \mathbb{R}^r \to \mathbb{R}$ has depth d and admits a level partition with levels $\{\ell_i\}_{i=1}^d$. Then, if there exists another network that computes the same function with width at at most $r+d$, depth at most $d+\prod_{i=1}^d |\ell_i|$, and uses activation functions from the original network as well as the identity function.*

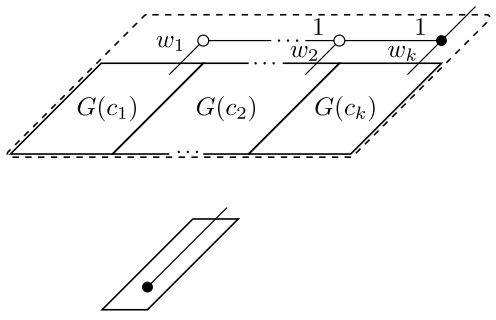
Proof. Let us call a network a *tree network* when the subgraph induced by the non-source nodes is a tree. Observe than we may convert any network into a tree network in the following way. Moving backwards through the levels, for each vertex v, replace v with a number of copies equal to its outdegree and then attach each of these duplicates to a distinct one of its descendents. Let $\mathcal{W}$ be this new network and let T be the tree induced by non-source nodes $\mathcal{W}$ with the sink node s of $\mathcal{W}$ as its root.

We now produce a network $\mathcal{U}$ based on $\mathcal{W}$. First we define a subnetwork G of $\mathcal{U}$ based on the T. Suppose $f_s(\mathbf{x}) = \sigma(\mathbf{w}\cdot\mathbf{x}+b)$ with $\mathbf{w} = (w_1, \ldots, w_k)$. Then recursively define G as

$G =$ 

,

where the black node has activation function σ and bias b, the white nodes have identity activation functions, and weight vectors are defined the values assigned to incoming edges. $G(c_i)$ corresponds to the subgraph assigned to the i^{th} tree-child of s, defined as follows

$G(v) =$ 

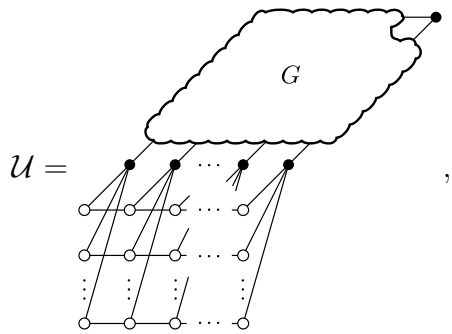
if v is an interior node in T

if v is a leaf,

where again c_i refers to the i^{th} tree-child of v and weights are marked on edges.

Now that G is defined, we attach an 'input bus' to the exposed leaves of T as

follows:

$\mathcal{U} =$ 

,

In this diagram, the leftmost column of white nodes are the source nodes from $\mathcal{W}$, horizontal edges bear weight one, and the edges going to each leaf from T are assigned the weights they received in $\mathcal{W}$.

Observe that $\mathcal{U}$ has depth at most

$$\#\text{ leaves in } T + \#\text{ nodes in longest path in } T \leq \prod_{i=1}^{d} |\ell_i| + d$$

and width

$$\#\text{ nodes in input bus} + \#\text{ nodes in longest path in } T = r + d. \qquad \square$$

Corollary 1.6.1. *Width $r+2$ ANNs using sigmoidal and identity activation functions are universal approximators.*

Corollary 1.6.2. *Width $r + 3$ ReLU ANNs are universal approximators.*

Proof. We can simulate identity units with ReLUs by replacing the unit $1 \cdot (\mathbf{w} \cdot \mathbf{x})$ with $\sigma(\mathbf{w} \cdot \mathbf{x}) - \sigma(-\mathbf{w} \cdot \mathbf{x})$. This increases the width of the top rail in G by one. $\square$

Recall that in practice, locating a neural network that implements a (ε, L_∞)-approximation of a target function involves a search in both the size of the network and the space of programmable parameters. However, the universality results just presented make no mention of bounds on any these quantities for a given function. This suggests an explanation for why universality results are a bad starting points for an underfitting avoidance strategy: the most efficient representations of functions may be spread out across different universality classes so far discovered. In the next section we will pursue some bounds which do tell us what function are *efficiently* approximated by depth-two networks, as well as explore some extensions to deeper regimes.

Chapter 2

Approximation Bounds

Given a continuous target function, how do network size requirements scale with approximation error or input dimension?

2.1 Upper Bounds: a result of Barron

A general bound does not explicitly appear in the literature to the best of the author's knowledge, but for certain restricted classes of functions we are able to prove facts about approximation rates. In particular, we present a classic result from Barron [3] which shows that functions whose Fourier distributions are concentrated toward zero require only linear growth in the number of nodes of a depth-two approximating network.

Formally, for a bounded set B and a constant $C > 0$, Barron defines the function class of $\Gamma_{B,C}$ as follows. Suppose f admits a Fourier representation

$$f(\mathbf{x}) = f(0) + \int (e^{i\boldsymbol{\omega}\cdot\mathbf{x}} - 1)\tilde{F}(\mathrm{d}\boldsymbol{\omega}) \tag{2.1}$$

on B, where $\tilde{F}$ is the Fourier distribution of f (see e.g., [27] for an overview). $\tilde{F}$ is complex-valued, so we may write $\tilde{F}(\boldsymbol{\omega}) = e^{i\theta(\boldsymbol{\omega})}F(\mathrm{d}\boldsymbol{\omega})$, where F is the modulus of $\tilde{F}$ and $\theta(\boldsymbol{\omega})$ denotes the phase of $\tilde{F}$ at $\boldsymbol{\omega}$. Define

$$C_{f,B} = \int |\boldsymbol{\omega}|_B F(\mathrm{d}\boldsymbol{\omega}),$$

where $|\boldsymbol{\omega}|_B = \sup_{\mathbf{x}\in B} |\mathbf{x}\cdot\boldsymbol{\omega}|$. Then let $\Gamma_{B,C}$ to be those f admitting representations (2.1) with $C_{f,B} \leq C$.

Theorem 2.1 (Barron 1993)**.** *For every $f \in \Gamma_{B,C}$, every probability measure μ and every $n \geq 1$, there exists a linear combination of sigmoidal functions of the form*

$$f_n = f(0) + \sum_{i=1}^{n} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x}_i + b_i)$$

such that

$$\int_B \big(f(\mathbf{x}) - f_n(\mathbf{x})\big)^2 \,\mathrm{d}\mu(\mathbf{x}) \leq \frac{4C^2}{n}.$$

Moreover the coefficients a_i may be restricted to satisfy $\sum_{i=1}^N |a_i| \leq 2C$.

Note that Barron allows a bias term on the output node (in particular, this bias is f_0). Barron's proof is motivated by the following lemma, credited to Bernhard Maurey in [23].

Lemma 2.2. *Suppose G is a set bounded by the ball of radius b in a Hilbert space and let $\bar{f} \in \overline{\mathrm{Conv}(G)}$. Then for all $n \geq 1$, every $\varepsilon > 0$ there exists a function f_n in the convex hull of n points in G with*

$$\|\bar{f} - f_n\| \leq \frac{b^2 - \|\bar{f}\|^2}{n} + \varepsilon.$$

Proof. Given $n \geq 1$ and fixing a $\delta > 0$, let $f^* \in \mathrm{Conv}(G)$ with $\|\bar{f} - f^*\| \leq \delta/n$. By definition $f^* = \sum_{k=1}^m \gamma_k g_k^*$ with $g_k^* \in G, \gamma_k \geq 0, \sum_{k=1}^m \gamma_k = 1$ for some m. Let $g, g_1, g_2, \ldots, g_n$ be drawn independently from $\{g_j^*\}$ according to $P(g_i = g_j^*) = \gamma_j$. Set $f_n = \frac{1}{n}\sum_{i=1}^n g_i$, the sample average, and note that $\mathbb{E}[f_n] = f^*$ with expected error

$$\begin{aligned}
\mathbb{E}\left[\|f_n - f^*\|^2\right] &= \mathbb{E}\left[\left\|\tfrac{1}{m}\textstyle\sum_{i=1}^n (g_i - f^*)\right\|^2\right] \\
&= \frac{1}{n^2}\left(\sum_{i=1}^n \mathbb{E}\left[\langle g_i - f^*, g_i - f^*\rangle\right] + \sum_{1\leq i\neq j\leq n} \mathbb{E}\left[\langle g_i - f^*, g_j - f^*\rangle\right]\right) \\
&= \frac{1}{n}\mathbb{E}\left[\|g - f^*\|^2\right] + \left(1 + \frac{1}{n}\right)\sum_{1\leq k,\ell\leq m} \gamma_k\gamma_\ell \left\langle g_k^* - f^*, g_\ell^* - f^*\right\rangle \\
&= \frac{1}{n}\mathbb{E}[\|g - f^*\|^2] \\
&= \frac{1}{n}(\mathbb{E}[\|g\|^2] - \|f^*\|^2) \\
&\leq \frac{1}{n}(b^2 - \|f^*\|^2).
\end{aligned}$$

This implies there exist specific $g_1, \ldots . g_n$ for which $\|f_n - f^*\|^2 \leq (1/n)(b^2 - \|f^*\|^2)$. Noting that $\|f^*\|^2 > \|\bar{f}\|^2 - \frac{2\delta}{n}\|\bar{f}\|$ we have by the triangle inequality

$$\begin{aligned}
\|\bar{f} - f_n\|^2 &\leq \frac{1}{n}(b^2 - \|f^*\|^2) + \frac{2\delta}{n}\sqrt{\frac{1}{n}(b^2 - \|f^*\|^2)} + \frac{\delta^2}{n^2} \\
&\leq \frac{1}{n}(b^2 - \|\bar{f}\|^2 + \frac{2\delta}{n}\|\bar{f}\|) + \frac{2\delta}{n}\sqrt{\frac{1}{n}(b^2 - \|f^*\|^2)} + \frac{\delta^2}{n^2} \\
&= \frac{b^2 - \|\bar{f}\|^2}{n} + \delta\left(\frac{2}{n}\|\bar{f}\| + \frac{2}{n}\sqrt{\frac{1}{n}(b^2 - \|f^*\|^2)} + \frac{\delta}{n^2}\right),
\end{aligned}$$

which can be upper-bounded as desired by the appropriate choice of δ. □

Thus, using the L_2 norm over μ as $\|\cdot\|$, this lemma asserts that if we can show $\bar{f}(\mathbf{x}) = f(\mathbf{x}) - f(0)$ is in the closure of the convex hull of

$$\mathcal{F}_\sigma = \{a\sigma(\mathbf{w}\cdot\mathbf{x}+b) : |a| \le 2C, w \in \mathbb{R}^d, b \in \mathbb{R}\},$$

we can prove our result by sampling a convex combination that ε-approximates f. We will demonstrate this membership via the sequence of inclusions

$$\bar{f} \in \overline{\operatorname{Conv}\mathcal{F}_{\cos}} \subseteq \overline{\operatorname{Conv}\mathcal{F}_{\text{step}}} \subseteq \overline{\operatorname{Conv}\mathcal{F}_\sigma},$$

with $\mathcal{F}_{\text{step}}$ and $\mathcal{F}_{\cos}$ to be defined shortly.

Lemma 2.3. *For each $f \in \Gamma_{B,C}$, $\bar{f}$ is in the closure of the convex hull of*

$$\mathcal{F}_{\cos} = \left\{ \frac{\gamma}{|\boldsymbol{\omega}|_B}\big(\cos(\boldsymbol{\omega}\cdot\mathbf{x}+b) - \cos(b)\big) : \boldsymbol{\omega} \neq 0, |\gamma| \le C, b \in \mathbb{R} \right\}.$$

Proof. Let $\Omega = \mathbb{R}^r - 0$. By definition of $\Gamma_{B,C}$ and the fact that f is real-valued,

$$\begin{aligned}\bar{f} &= \operatorname{Re}\int_\Omega (e^{i\boldsymbol{\omega}\cdot\mathbf{x}} - 1)e^{i\theta(\boldsymbol{\omega})}\,\mathrm{d}F(\boldsymbol{\omega}) \\ &= \int_\Omega \Big(\cos\big(\boldsymbol{\omega}\cdot\mathbf{x}+\theta(\boldsymbol{\omega})\big) - \cos\big(\theta(\boldsymbol{\omega})\big)\Big)\,\mathrm{d}F(\boldsymbol{\omega})\end{aligned}$$

Defining $\Lambda(\mathrm{d}\boldsymbol{\omega})$ as the probability distribution $|\boldsymbol{\omega}|_B F(\mathrm{d}\boldsymbol{\omega})/C_{f,B}$, we then have

$$\bar{f} = \int_\Omega \frac{C_{f,B}}{|\boldsymbol{\omega}|_B}\Big(\cos\big(\boldsymbol{\omega}\cdot\mathbf{x}+\theta(\boldsymbol{\omega})\big) - \cos\big(\theta(\boldsymbol{\omega})\big)\Big)\,\mathrm{d}\Lambda(\boldsymbol{\omega}), \tag{2.2}$$

and $\bar{f}$ is now expressed as an infinite convex combination of elements from $\mathcal{F}_{\cos}$. We claim this entails the result for the L_2 norm according to any distribution μ. Supose $\{\boldsymbol{\omega}_i\}_{i=1}^m$ is a random sample of n points from Ω drawn according to Λ. Then, writing the integrand in (2.2) as $g(\mathbf{x},\boldsymbol{\omega})$, the expected square of the $L_2(\mu,B)$ error between $\bar{f}$ and the sample average is

$$\begin{aligned}\mathbb{E}\left[\int_{B_r}\left(\bar{f} - \frac{1}{n}\sum_{i=1}^n g(\mathbf{x},\boldsymbol{\omega}_i)\right)^2 \mathrm{d}\mu(\mathbf{x})\right] &= \int_{B_r}\mathbb{E}\left[\left(\bar{f} - \frac{1}{n}\sum_{i=1}^n g(\mathbf{x},\boldsymbol{\omega}_i)\right)^2\right]\mathrm{d}\mu(\mathbf{x}) \\ &= \frac{1}{n}\int_{B_r}\operatorname{var}\big(g(\mathbf{x},\boldsymbol{\omega})\big)\,\mathrm{d}\mu(\mathbf{x}) \\ &\le \frac{1}{n}\int_{B_r}\mathbb{E}[g(\mathbf{x},\boldsymbol{\omega})^2]\,\mathrm{d}\mu(\mathbf{x}) \\ \text{and, by some careful trigonometry,}\quad &\le \frac{1}{n}\int_{B_r}\mathbb{E}\left[C^2\frac{|\mathbf{x}\cdot\boldsymbol{\omega}|^2}{|\boldsymbol{\omega}|_B^2}\right]\mathrm{d}\mu(\mathbf{x}) \\ &\le \frac{C^2}{n}.\end{aligned}$$

Thus, by sending $n \to \infty$, we have a convergent sequence of functions to $\bar{f}$ as desired. □

Lemma 2.4. *Define*

$$\mathcal{F}_{\text{step}} = \{\gamma \,\text{step}(\boldsymbol{\omega} \cdot \mathbf{x} - b) : |\gamma| \leq 2C, |\boldsymbol{\omega}|_B = 1, |b| \leq 1\},$$

where $\text{step}(x) = 1$ *if* $x \geq 0$ *and* 0 *otherwise. Then* $\overline{\text{Conv}\,\mathcal{F}_{\cos}} \subseteq \overline{\text{Conv}\,\mathcal{F}_{\text{step}}}$.

Proof. We may write any function in $\mathcal{F}_{\cos}$ as $g \circ h(\mathbf{x})$ with

$$g(x) = \frac{\gamma}{|\boldsymbol{\omega}|_B}\big(\cos(|\boldsymbol{\omega}|_B\, x + b) - \cos(b)\big) \qquad \text{and} \qquad h(\mathbf{x}) = \frac{\boldsymbol{\omega} \cdot \mathbf{x}}{|\boldsymbol{\omega}|_B}.$$

If we can approximate any g with a linear combination of step functions, then we are done. Note that $h(\mathbf{x}) \in [-1, 1]$ for $\mathbf{x} \in B$, so we concern ourselves with approximating g only over $[-1, 1]$. The derivative of g is bounded by $|\gamma| < C$, implying g is uniformly continuous and thus uniformly approximated by linear combinations of step functions. In particular, there exists a sequence of partitions $\{(-1 = b_0 < b_1 < \cdots < b_k = 1)\}_{k=1}^{\infty}$ for which

$$g_k(x) = g(b_0)\,\text{step}(x - b_0) + \sum_{i=1}^{k} \big(g(b_i) - g(b_{i-1})\big)\,\text{step}(x - b_i) \xrightarrow{unif} g(x).$$

Because $g(0) = 0$ and its derivative is bounded by $|\gamma|$, we have $\max_{-1 \leq x \leq 1} |g(x)| \leq qC$, implying the coefficients in $g_k(x)$ are each bounded by $2C$. Thus the function $g_k \circ h(\mathbf{x})$ constitutes the desired approximation. □

Lemma 2.5. $\overline{\text{Conv}\,\mathcal{F}_{step}} \subseteq \overline{\text{Conv}\,\mathcal{F}_{\sigma}}$.

Proof. Assuming the distribution of $\boldsymbol{\omega} \cdot \mathbf{x}$ induced by μ is continuous for every α, we obtain a pointwise approximation of $\text{step}(\boldsymbol{\omega} \cdot \mathbf{x} - b)$ by taking α arbitrarily large in $\sigma(\alpha(\boldsymbol{\omega} \cdot \mathbf{x} - b))$. By the dominated convergence theorem, the limit also holds in $L_2(\mu, B)$.

The case where the distribution of $\boldsymbol{\omega} \cdot \mathbf{x}$ is not continuous requires a small technical modification; we refer the reader to the original paper [3] for more details. □

Proof of Theorem 2.1. The result follows from Lemma 2.2 and the inclusions just demonstrated. There are three cases.

If $\|\bar{f}\| = 0$ then f is equal to a constant μ-almost everywhere on B and the approximation bound is trivially true.

Now suppose $\|\bar{f}\| > 0$. If $|\sigma| \leq 1$, then by definition elements of $\mathcal{F}_{\sigma}$ are bounded by $2C$. Therefore we may apply Lemma 2.2 with $b = 2C$.

If $|\sigma| > 1$, then slightly more work is required. Using Lemma 2.2 and the membership of $\bar{f}$ in the closure of $\mathcal{F}_{\text{step}}$, obtain a convex combination of n functions which approximates $\bar{f}$ within

$$\frac{(2C)^2 - \frac{1}{2}\|\bar{f}\|^2}{n}$$

according to the $L_2(\mu, B)$ norm. We may then replace each step function in the convex combination with a sufficiently-sharp approximation from $\mathcal{F}_\sigma$, thanks to the inclusion $\mathcal{F}_{\text{step}} \subseteq \mathcal{F}_\sigma$ to obtain a total $L_2(\mu, B)$ error bounded by $4C^2/n$. □

So what functions are in $\Gamma_{B,C}$? Barron provides a thorough set of examples in Section 9 of [3], but of special interest to us is the membership of polynomials in $\Gamma_{B,C}$. By the Stone-Weierstrass theorem, polynomials are dense in $C[K]$ for K a compact subset of $\mathbb{R}^r$. Thus Barron's result also constitutes a universality theorem, if in a somewhat round-about fashion.

For a reader interested in deep networks, however Barron's result is not particularly satisfying. Again our mathematical techniques are restricted to operations in linear spaces, in this case convex combinations, made possible by the especially simple structure of depth-two ANNs. Barron has demonstrated a nice class of functions which admit a tractable approximation rate by depth-two networks, but this class is fairly opaque and restrictive, and moreover Γ tells us nothing about classes of functions which admit linear or polynomial approximation rates when using deep network architectures. We would expect, for instance, that deeper networks have good rates (as width increases) for larger classes of functions. Fortunately, we can without too much work generalize Barron's result for larger classes of functions that admit efficient approximation rates by deep networks.

We will require a strong form of continuity to describe these functions.

Definition 10. A function $f : K \to \mathbb{R}$ is (K, L_p)-Lipschitz if for all $\mathbf{x}, \mathbf{y} \in K$, we have

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq K\|\mathbf{x} - \mathbf{y}\|_{L_p}.$$

Letting A and B be bounded subsets of $\mathbb{R}^r$ and $\mathbb{R}^m$ respectively, begin by defining $\Gamma_{A,B}(C, K)$ as the class of functions $f : A \to B$ for which $f_i \in \Gamma_{A,C}$ and f_i is (K, L_∞)-Lipschitz for each i, where f_i is the i^{th} component of f. Then we have the following:

Corollary 2.5.1. *Suppose A, B are bounded subsets of $\mathbb{R}^r, \mathbb{R}^m$ respectively and $f : A \to B_d$ can be decomposed as*

$$A \xrightarrow{f} B_d \qquad = \qquad (A = B_0) \xrightarrow{g_1} B_1 \xrightarrow{g_2} B_2 \cdots B_{d-1} \xrightarrow{g_d} (B_d = B),$$

where each B_i is a bounded subset of $\mathbb{R}^m$ and $g_i \in \Gamma_{B_{i-1},B_i}(C,K)$ for each $1 \leq i \leq d$. Then the L_∞ approximation rate of f by sigmoidal networks of depth d is

$$O\left(\frac{K^d C}{\sqrt{n}}\right),$$

where n is the width of the network.

Proof. The bound we will prove is

$$\|f - \bar{f}\|_\infty \leq \frac{2C\sum_{i=1}^{d} K^i}{\sqrt{n}}$$

For each i, let $\bar{g}_i = (\bar{g}_{i1}, \ldots, \bar{g}_{im})$, where $\bar{g}_{ij}$ is the n^{th} approximation of g_{ij} in the sense of Barron. Then the L_∞ error of $\bar{g}_i$ is bounded by

$$\begin{aligned}
\|g_i - \bar{g}_i\|_\infty &= \sup_{\mathbf{x}\in B_{i-1}} \max_{1\leq j\leq m} |g_{ij}(\mathbf{x}) - \bar{g}_{ij}(\mathbf{x})| \\
&= \max_{1\leq j\leq m} \sup_{\mathbf{x}\in B_{i-1}} |g_{ij}(\mathbf{x}) - \bar{g}ij(\mathbf{x})| \\
&\leq \max_{1\leq j\leq m} \|g_{ij} - \bar{g}_{ij}\|_2 \\
&\leq \frac{2C}{\sqrt{n}} \text{ by Barron.}
\end{aligned}$$

We now prove the bound inductively. The case for $d = 1$ is already done, so let $d > 1$. Then

$$\begin{aligned}
\|f - \bar{f}\|_\infty &\leq \sup_{\mathbf{x}\in A} \max_{1\leq i\leq w} \big|g_d\big((g_{d-1}\circ\cdots\circ g_1)(\mathbf{x})\big) - \bar{g}_d\big((\bar{g}_{d-1}\circ\cdots\circ\bar{g}_1)(\mathbf{x})\big)\big| \\
&\leq \max_{1\leq i\leq w} \sup_{\mathbf{x}\in A} \Big(\big|g_d\big((g_{d-1}\circ\cdots\circ g_1)(\mathbf{x})\big) - g_d\big((\bar{g}_{d-1}\circ\cdots\circ\bar{g}_1)(\mathbf{x})\big)\big| \\
&\qquad\qquad + \big|g_d\big((\bar{g}_{d-1}\circ\cdots\circ\bar{g}_1)(\mathbf{x})\big) - \bar{g}_d\big((\bar{g}_{d-1}\circ\cdots\circ\bar{g}_1)(\mathbf{x})\big)\big|\Big) \\
&\leq K\|(g_{d-1}\circ\cdots\circ g_1)(\mathbf{x})\big) - (\bar{g}_{d-1}\circ\cdots\circ\bar{g}_1)(\mathbf{x})\|_\infty + \frac{2C}{\sqrt{n}} \\
&\leq K\left(\frac{2C}{\sqrt{n}}\sum_{i=0}^{d-2} K^i\right) + \frac{2C}{\sqrt{n}} \qquad \text{by induction} \\
&= \frac{2C}{\sqrt{n}}\sum_{i=0}^{d-1} K^i.
\end{aligned}$$

□

Therefore, if we have reason to suspect the the target function we are trying to learn can be written as a finite composition of functions in Γ, this corollary implies that we can achieve acceptable error by working with networks of depth approximately d and looking at increasing widths.

However: let $\Gamma^d_{A,B}(C,K)$ be the set of functions f equal to d compositions of functions in $\Gamma_{A,B}(C,K)$. It is not known how much larger $\Gamma^d_{A,B}(C,K)$ is than $\Gamma_{A,B}(C,K)$. In particular, is $\Gamma^d_{A,B}(C,K) \subseteq \Gamma_{A,B}(C',K')$ for some C', K'? If this is the case, then this corollary is much less interesting, because it doesn't represent any gain in representational power by deep neural networks, and instead suggests Barron's class Γ is too restrictive to benefit from depth. We leave this as an open question, and transition now to calculations of lower bounds on the size of networks required to approximate certain classes of functions.

2.2 Lower Bounds

Here we present two lower bounds for classes of functions, one inspired by a proof technique from a classic result of Shannon in the boolean circuit literature, and the other based on VC-dimension bounds for ANNs.

2.2.1 A lower bound for Lipschitz functions

One of the earliest results in boolean circuit theory is from Shannon's seminal paper [33]. Therein he shows that there aren't enough small boolean circuits to accommodate the 2^{2^n} different functions $\{0,1\}^n \to \{0,1\}$. Here we show a similar argument goes through for neural networks and Lipschitz functions, though a different technique must be used to count the number of sufficiently different functions implemented by small ANNs. This process takes the form of bounding the size of a minimal ε-net, and it is likely the methods used therein could be substantially improved. Even so, we obtain the analogous result we seek.

Theorem 2.6. *Let $B, J, K > 0$ be fixed. Then, working in the L_∞ norm, for sufficiently large r there exist (K, L_∞)-Lipschitz functions $\{f\}$ from $[0,1]^r \to \mathbb{R}$ such that for all $\varepsilon \in (0, K)$, each f cannot be ε-approximated by neural networks with J-Lipschitz activation functions, weights and biases bounded by B, and*

$$\frac{2^{r/3}}{\sqrt[3]{2\log(7JB/\varepsilon)}} \text{ nodes.} \tag{2.3}$$

Proof. Consider the set of K-Lipschitz functions defined as follows: For each $\beta : \{0,1\}^n \to \{0,1\}$ define f_β such that $f_\beta(\mathbf{x}) = K/2$ if $\beta(\mathbf{x}) = 1$ and $-K/2$ if $\beta(\mathbf{x}) = 0$. These functions may be extended to K-Lipschitz functions over $[0,1]^r$. Thus we have at least 2^{2^r} K-Lipschitz functions such that for any pair f, f' we have $\|f - f'\| \geq K$.

For a fixed J-Lipschitz activation function σ, let N be the set of neural network architectures of size n with internal units assigned activation function σ and sink units assigned the identity. Define

$$\mathsf{F} = \bigcup_{\mathcal{N} \in \mathsf{N}} \mathcal{F}(\mathcal{N}, B).$$

We now bound the size of a minimal ε-net of F.

To that end, we first consider a specific $\mathcal{N} \in \mathsf{N}$ and bound the size of minimal ε-net of $\mathcal{F}(\mathcal{N}, B)$. Taking the programmable parameters of $\mathcal{N}$ as the space $[-B, B]^t$ for some t, we will do this by defining a lattice

$$L_\delta = (\delta\mathbb{Z})^t \cap [-B, B]^t$$

such that $F(L_\delta)$ (the set of network functions defined by parameter vectors in L_δ) is an ε-net of $\mathcal{F}(\mathcal{N}, B)$.

Consider a unit connected to a source node in $\mathcal{N}$. By a similar argument to that used to bound the error in the proof of Corollary 2.5.1, changing $\mathbf{w}$ or b by δ creates at most a $\delta(J+1)^d \leq \delta(J+1)^n$ change in the output of $\mathcal{N}$, where d is the depth of the network. We want $\delta(J+1)^n < \varepsilon$, so setting $\delta < \varepsilon/(J+1)^n$ ensures $F(L_\delta)$ is an ε-net.

What is $|L_\delta|$? Each dimension contributes a point every δ from $-B$ to B, so in total L has

$$\frac{2B}{\varepsilon/(J^n)} = \frac{2BJ^n}{\varepsilon}$$

points in each of t dimensions. An upper bound on the number of programmable parameters on each vertex is $n+1$ (one for the bias), so $t \leq n^2 + n$. Hence there is an ε-net of $\mathcal{N}$ with size

$$\left(\frac{2S^2MJ^S}{\varepsilon}\right)^{n^2+n}.$$

There are at most $3^{\binom{n}{2}}$ networks on n nodes, so as a grand total the size of an ε-net of F is upper-bounded by

$$3^{n^2}\left(\frac{2MJ^n}{\varepsilon}\right)^{n^2+n} \leq \left(\frac{2M}{\varepsilon}\right)^{n^2+n}(3J)^{n^3+n^2} \leq \left(\frac{6BJ}{\varepsilon}\right)^{2n^3}$$

for sufficiently large n.

Substituting the choice of network size (2.3) for n we see there are at most

$$2^{2^n log(6JM/\varepsilon)/log(7JM/\varepsilon)}$$

functions in an ε-net of F for sufficiently large n. This number is smaller than 2^{2^n}, implying the result. □

2.2.2 A lower bound for polynomials

We now present a clever construction by Schmitt [32] to give lower bounds on the sigmoidal network size required to represent polynomials. The analysis of this construction depends on upper bounds for the VC dimension of a given neural network architecture $\mathcal{A}$, which is a measurement of the richness of $\mathcal{F}(\mathcal{A})$.

Definition 11 (VC Dimension)**.** Let $S \subseteq \mathbb{R}^m$ and let $\mathcal{F}$ be a family of functions $f : \mathbb{R}^r \to \{0,1\}$. We say $\mathcal{F}$ *shatters* S if for every boolean function $\beta : S \to \{0,1\}$, there exists an $f_\beta \in \mathcal{F}$ such that $f_\beta|_S = \beta$. The *Vapnik-Chervonenkis dimension* of $\mathcal{F}$ is the greatest cardinality among all sets shattered by $\mathcal{F}$.

In the case of a neural network $\mathcal{N}$ we define its VC dimension to be that of the function $\text{step}(\mathcal{N} + 1/2) : \mathbb{R}^r \to \{0,1\}$.

The VC dimension bounds we use are from Karpinski & Macintyre [15]

Lemma 2.7. *(Karpinski & Macintyre 1997) Sigmoidal neural networks have VC dimension at most $O(n\ell)$, where n is the number of nodes and ℓ is the number of parameters.*

Their proof of this fact uses techniques from model theory and algebraic geometry and is outside the scope of this thesis.

Schmitt's construction is motivated by the following Lemma, given in [16].

Lemma 2.8 (Koiran & Sontag 1997)**.** *Let p_n be defined by*

$$p_n(x) = \begin{cases} 4x(1-x) & n = 0 \\ p_1(p_{n-1}(x)) & n \geq 1. \end{cases}$$

Fix $n \in \mathbb{N}$ and define $[n] = \{0, 1, 2, \ldots, n-1\}$. Then for all $\beta : [n] \to \{0,1\}$, there exists a $w_\beta \in \mathbb{R}$ such that for all $i \in [n]$

$$p_i(w_\beta) > 1/2 \text{ if } \beta(i) = 1 \text{ and}$$
$$p_i(w_\beta) < 1/2 \text{ if } \beta(i) = 0.$$

Proof. Let $\beta : [n] \to \{0,1\}$ be given. We define a sequence $w_n, w_{n-1} \ldots, w_0, w_\beta$ such that for all $i \in [n]$, $w_i = p_i(w)$ and w_i is bounded as desired.

If $\beta(n-1) = 0$, choose $w_{n-1} = 1/4$ and if $\beta(n-1) = 1$, choose $w_{n-1} = 3/4$. Note that all for all $x \in (0,1)$, x has two preimages in p_0, one in $(0, 1/2)$ and one in $(1/2, 1)$. So for each $i = n-2, n-3, \ldots, 0$ we may choose $w_i \in p_0^{-1}(w_{i+1})$ such that

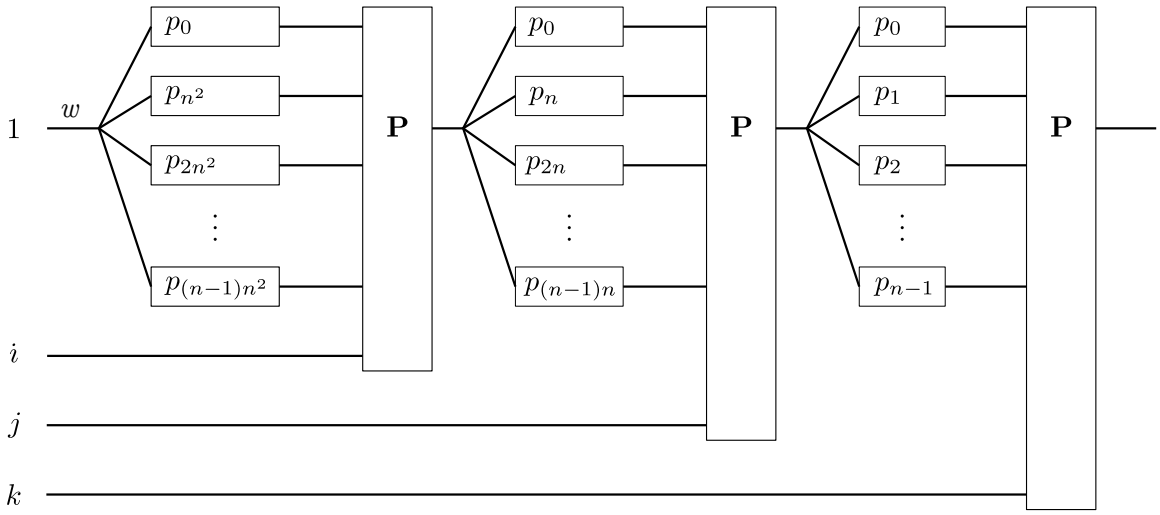
$$w_i > 1/2 \text{ if } \beta(i) = 1 \text{ and}$$
$$w_i < 1/2 \text{ if } \beta(i) = 0.$$

Finally, choose w_β arbitrarily in $p_0^{-1}(w_0)$. Observe that by construction, for each $i \in [n], w_i = p_i(w_\beta)$ is bounded as desired. □

This lemma means that if we can construct a network architecture $\mathcal{A}$ that allows us to range over the is in $p_i(w)$ by changing inputs and allows us to range over ws by adjusting weights, then $\mathcal{F}(\mathcal{A})$ can shatter some $[n]$.

Theorem 2.9 (Schmitt 1999)**.** *Sigmoidal networks approximating $(p_n)_{n>1}$ on $[0,1]$ with L_∞ error at most $O(2^{-n})$ require at least $\Omega(n^{1/4})$ nodes.*

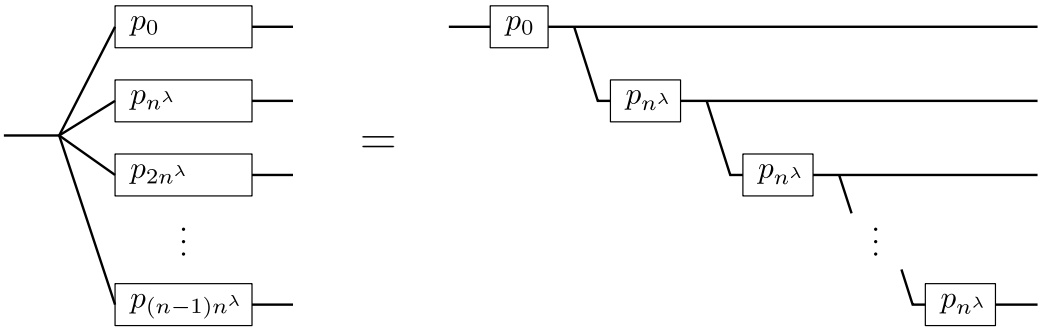
Proof. We construct a network architecture with four inputs. The last three inputs will be used to range over $i \in [n^3]$ as written in base n (and thus requiring three digits) and the first will be kept at 1. Define $\mathcal{N}_w(1, i, j, k)$ as



where each box represents a sub-network implementing the labeled function; here $\mathbf{P} : \mathbb{R}^{n+1} \to \mathbb{R}$ is a function that has

$$\mathbf{P}(x_0, \ldots, x_{n-1}, y) = x_y$$

for $y \in [n]$. Observe that for $(i, j, k) \in [n]^3$, $\mathcal{N}_w(1, i, j, k)$ computes $p_k(p_{jn}(p_{in^2}(w))) = p_{in^2+jn+k}(w)$ as desired. Further, note that for $\lambda = 0, 1, 2$ each column of p_{in^λ}s may also be implemented as



Hence in order to derive an implementation of $\mathcal{N}_w(1, i, j, k)$, we need only implement p_0, p_n, p_{n^2}, and $\mathbf{P}$.

By Lemma 1 in [16] (reproduced below as Lemma 2.10), there exists a network architecture comprised of $O(n)$ linear, multiplication, and division gates which admits network that (ε, L_∞)-approximates $\mathbf{P}$. Now suppose that we replace each p_{n^λ} with a sigmoidal network approximation with error bounded by $O(2^{-n})$. Straightforward analysis shows this network still shatters $[n]^3$.

By the Theorem 7 from [15] (given above as Lemma 2.7), the VC dimension of $\mathcal{N}_w$ is $O(m^2)$ where m is the number of vertices. But $\mathcal{N}_w$ shatters a set of size n^3, so for some $\lambda \in \{0, 1, 2\}$ the number of nodes in a sigmoidal network implementing p_{n^λ} must be lower-bounded by $\Omega(\sqrt{n})$. This implies a lower bound of $\Omega(n^{1/4})$ for sigmoidal networks implementing p_n. □

Lemma 2.10. *(Koiran & Sontag, 1997) For all $n > 1$, there exists a network architecture $\mathcal{N}$ with $O(n)$ units which admits a sequence of networks $\mathcal{N}_1, \mathcal{N}_2, \ldots$ such that $\lim_{i\to\infty} \mathcal{N}_i = \mathbf{P}$.*

Proof. Consider the sequence of functions

$$f_k(x_0, \ldots, x_{n-1}, y) = \left(\prod_{i=0}^{n-1}(y - i - 1/k)\right)\left(\sum_{i=0}^{n-1}\frac{a_i x_i}{y - i - 1/k}\right),$$

where $a_i = 1/\prod_{j\neq i}(i - j)$. In this form each f_k can be implemented by a network with a single multiplication unit, n division units, and $n + 1$ linear units, for a total of $O(n)$. Observe that we may rearrange terms to obtain

$$f_k = \sum_{i=1}^{n-1} x_i \prod_{i\neq j}\frac{x - j - \varepsilon}{i - j},$$

which converges pointwise to $\mathbf{P}$ on the relevant domain. □

Chapter 3

Exponential Separations

3.1 What is an exponential separation?

Inspired by separation results in the boolean circuit literature (e.g., [30]), as well as a desire to understand expressivity's role in deep learning, researchers have found a number of functions that are easy to approximate by deep networks (size requirements are polynomial in depth, input dimension, or error) but hard to approximate by shallow neural networks (size requirements are exp in the same quantities).

This is important for architecture design because while it may be intractable to learn such functions with a shallow network (networks of tractable size wouldn't be able to offer an approximations), it is easy to learn them with deeper networks. If we could characterize the sorts of functions amenable to this efficiency gain with depth, we would be well on our way to avoiding underfitting.

Before we describe these functions, we will take a moment to understand separations in general, and look for some justification as to why we desire *exponential* separations, rather than polynomial or somewhere in-between. This justification is based on the concept of mutual simulation, borrowed from computational ideas like Kolmogorov complexity. In particular, we aim to show that if two classes of networks, each with their own assignments of activation functions, can efficiently simulate the units of the other, then any constant-depth separation result that holds for one network class will automatically hold for the other.

If a neural architecture $\mathcal{A}$ is assigned activation functions from some basis $\mathcal{G}$, then we call $\mathcal{A}$ a *$\mathcal{G}$-network architecture* and we call an $\mathcal{N} \in \mathcal{F}(\mathcal{A})$ a *$\mathcal{G}$-network.* Let us formalize what a separation result looks like:

Definition 12. (Network Separations) Suppose $F = \{f_i\}_{i=1}^{\infty}$ is a sequence of functions. Fix a gateset $\mathcal{G}$ and suppose $B = \{B_i\}_{i=1}^{\infty}$ and $G = \{G_i\}_{i=1}^{\infty}$, where each B_i, G_i

is a set of $\mathcal{G}$-networks. Further, let $g : \mathbb{N} \times (0, \infty) \to \mathbb{N}$ and $b : \mathbb{N} \to \mathbb{N}$ be functions such that for each i and ε, $b(i) \geq g(i, 1/\varepsilon)$.

Then we say F (g, b)-separates B from G along i if there exists an N such that for all $i > N$, for all $\varepsilon > 0$, the following two conditions hold:

i. There exists an $\mathcal{N} \in G_i$ with at most $g(i, 1/\varepsilon)$ nodes for which $\|\mathcal{N} - f_i\|_\infty < \varepsilon$,
ii. There exists an $\varepsilon' > 0$ such that for all networks $\mathcal{W} \in B_i$ that have $\|\mathcal{W} - f\|_\infty < \varepsilon'$, $|\mathcal{W}| \geq b(i)$.

We call the separation *exponential* if g is a polynomial in i and $1/\varepsilon$ and b is an exponential in i.

As an example, here is an exponential separation result we will see later on in this chapter:

Theorem. *Let $F_r : \mathbb{S}^{r-1} \times \mathbb{S}^{r-1} \to \mathbb{R}$ be given by $F_r(\mathbf{x}, \mathbf{x}') = \sin(\pi r^3 \langle \mathbf{x}, \mathbf{x}' \rangle)$. Then for all $\varepsilon > 0, r \geq 2$,*

1. *There exists an ReLU network of depth 3 and width at most $16\pi d^5/\varepsilon$ which (ε, L_2)-approximates F_r,*
2. *To obtain a $(1/(50e^2\pi^2), L_2)$-approximation of F_r with a 2-layer ReLU network with weights bounded by 2^r, we require at least $2^{\Omega(r \log r)}$ units.*

Here $\{F_r\}$ is exponentially separating depth-two ReLU-networks from depth-three ReLU-networks along input dimension r.

Definition 13 (Gateset Simulation)**.** Let $\mathcal{G}$ and $\mathcal{H}$ be activation function bases. We then say $\mathcal{H}$ *simulates* $\mathcal{G}$ if for all $g \in \mathcal{G}$ and all $\varepsilon > 0$ there exists an $\mathcal{H}$-network $\mathcal{N}$ such that $\|g - \mathcal{N}\|_\infty < \varepsilon$. If there exists such an $\mathcal{N}$ with $|\mathcal{N}| \in \text{poly}(1/\varepsilon)$ we say $\mathcal{H}$ *efficiently* simulates $\mathcal{G}$.

If $\mathcal{G}$ and $\mathcal{H}$ efficiently simulate each other using depth-two networks, we say they are *mutually depth-two-simulating*.

Remark 3.1. Sigmoids and ReLUs are mutually depth-two-simulating.

Lemma 3.2 (Simulation error)**.** *Suppose $\mathcal{N}$ is a network with gateset $\mathcal{G}$ and gateset $\mathcal{H}$ simulates $\mathcal{G}$. Let the network $\mathcal{N}'$ be obtained by replacing each unit $\sigma(\mathbf{w} \cdot \mathbf{x})$ in $\mathcal{N}$ with its simulation with gates in $\mathcal{H}$. Then if $\mathcal{N}$ has depth d and all functions in $\mathcal{G}$ are (K, L_∞)-Lipschitz, we have*

$$\|\mathcal{N} - \mathcal{N}'\|_\infty \leq \varepsilon \sum_{i=0}^{d-1} K^i,$$
$$\left(\leq \varepsilon (K+1)^{d-1} \right).$$

The proof technique of this lemma is nearly identical to that of Corollary 2.5.1 and so we will omit the proof.

Theorem 3.3. *Suppose $\mathcal{G}$ and $\mathcal{H}$ are depth-2 mutually simulating (K, L_∞)-Lipschitz neural gatesets and that F (poly,exp)-separates depth-d_b networks from depth-d_g networks on $\mathcal{G}$ Then F also (poly,exp)-separates depth-d_b networks from depth-d_g networks on $\mathcal{H}$.*

Proof. Let $\varepsilon > 0$ be given. Then there's a $\mathcal{G}$-network $\mathcal{N} \in G_i$ for which $\|\mathcal{N} - f_i\|_\infty < \varepsilon/2$ and $|\mathcal{N}| = \text{poly}(i)$. By Lemma we have a network $\mathcal{W}$ in $\mathcal{H}$ with the same depth in $\text{poly}(K^{d_g}/\varepsilon) \cdot \text{poly}(i) = \text{poly}(1/\varepsilon, i)$ nodes.

Now suppose by contradiction that for all ε, there's a d_b-layer network $\mathcal{W}'$ in $\mathcal{H}$ for which $\|f_i - \mathcal{W}'\|_\infty \leq \varepsilon$ and $|\mathcal{W}'|$ is sub-exponential in i and $1/\varepsilon$. Then again by Lemma we have a subexponential-size $\mathcal{G}$-network also approximating f_i of depth d_b. This cannot be. □

This theorem motivates to an extent why we search for exponential separations: as long as they separate fixed-depth network classes, they are agnostic of choice of network basis among those which are equivalent in approximation power up to a polynomial.

We present two sorts of exponential separations for neural networks: some which separate depth-two networks from depth-three networks along input dimension; and one which separates depth d networks from depth d^3 networks along depth d itself.

3.2 Separating depth-two and depth-three networks

There exist a number of results of varying generality which separate depth-two networks from those of depth three. Arguably the "nicest" of these is an extension of a result of Eldan & Shamir [8] to a more natural setting by Safran & Shamir [29]:

Theorem 3.4. *(Safran & Shamir, 2017) For any continuous probability distribution μ, the indicator of the Euclidean unit ball in $\mathbb{R}^r$ can be (ε, L_2, μ)-approximated to any accuracy ε using a 3-layer network with $O(d/\varepsilon)$ units. On the other hand, there exists a continuous probability distribution γ such that any 2-layer network requires $\exp(r)$ units to provide an approximation of accuracy better than $O(1/d^4)$.*

This result is "nice" because it shows depth separation on a natural function, or one that a learning algorithm may encounter in application. This is an improvement over the result upon which it is based, which separates depth-two from depth-three

networks via a function whose Fourier transform is an irregular sequence of nested shells.

We forgo proofs of either of these theorems as they are long, technical, and share most important conceptual elements with the following result of Daniely [7], the proof of which is somewhat easier-going.

Theorem 3.5. *(Daniely 2017) Let $F_r : \mathbb{S}^{r-1} \times \mathbb{S}^{r-1} \to \mathbb{R}$ be given by $F_r(\mathbf{x}, \mathbf{x}') = \sin(\pi r^3 \langle \mathbf{x}, \mathbf{x}' \rangle)$. Then for all $\varepsilon > 0, r \geq 2$,*

1. *There exists an ReLU network of depth 3 which (ε, L_2)-approximates F_r,*
2. *To obtain a $(1/(50e^2\pi^2), L_2)$-approximation of F_r with a 2-layer ReLU network with weights bounded by 2^r, we require at least $2^{\Omega(r \log r)}$ units.*

We prove this theorem via a series of lemmas. The proofs themselves have been left largely as-is (save for a few notational changes and expansion of some steps), but they have completely reorganized for clarity, concision, and to better suit the surrounding discussion. We address parts 1 and 2 of Theorem 3.5 separately.

Nonexistence of an efficient shallow approximation:

Let *inner product functions* denote those functions $f : \mathbb{S}^{r-1} \times \mathbb{S}^{r-1} \to \mathbb{R}$ which take the form of $\phi(\langle \mathbf{x}, \mathbf{x}' \rangle)$ for some $\phi : [-1, 1] \to \mathbb{R}$. Also, let say a function $f : \mathbb{S}^{r-1} \times \mathbb{S}^{r-1} \to \mathbb{R}$ is $\mathbf{v}, \mathbf{v}'$-*separable* if it can be written as $\psi(\langle \mathbf{v}, \mathbf{x} \rangle, \langle \mathbf{v}', \mathbf{x}' \rangle)$ for some $\psi : [-1, 1]^2 \to \mathbb{R}$. Observe that separable functions contain all functions representable by depth-two ANNs.

We show the first claim of Theorem 3.5 in two steps. First we prove a general fact about the relationship between the approximability of an inner product function f by low-degree polynomials and approximability of f by separable functions. We then show the sine function is poorly approximated by low-degree polynomials. To prepare, we must collect some notation for and facts about harmonic analysis on the sphere. See e.g., [2] for a thorough treatment.

The surface area of $\mathbb{S}^{r-1}$, denoted $|\mathbb{S}^{r-1}|$ may be calculated by integrating $(r-2)$-dimensional slices of the $(r-1)$-sphere along a single axis in $\mathbb{R}^r$. In particular,

$$|\mathbb{S}^{r-1}| = |\mathbb{S}^{r-2}| \int_{-1}^{1} (1-x^2)^{(r-3)/2} \, \mathrm{d}x \qquad \text{with} \qquad \int_{-1}^{1} (1-x^2)^{(r-3)/2} \, \mathrm{d}x = \frac{\sqrt{\pi}\Gamma(\frac{r-1}{2})}{\Gamma(\frac{r}{2})},$$

where Γ is Euler's Gamma function. Thus dividing the first representation by the

second we have that

$$1 = \frac{\Gamma(\frac{r}{2})}{\sqrt{\pi}\Gamma(\frac{r-1}{2})} \int_{-1}^{1} (1-x^2)^{(r-3)/2}\,\mathrm{d}x,$$

$$\text{and so} \qquad \mu_r(x) = \frac{\Gamma(\frac{r}{2})}{\sqrt{\pi}\Gamma(\frac{r-1}{2})}(1-x^2)^{(r-3)/2}$$

denotes a univariate probability distribution with support on $[-1,1]$ that is a projection of the uniform distribution over $\mathbb{S}^{r-1}$ onto a single line through the origin, say along the first component x_1 of a vector $\mathbf{x} \in \mathbb{R}^r$. This constitutes *Fact 1.* Note that we will assume in the lemmas that follow that the functions begin considered are $L_2(\mu)$-integrable.

We now make the following definitions

$$P_n(x) = \frac{2n+r-4}{n+r-3} x P_{n-1}(x) - \frac{n-1}{n+r-3} P_{n-2}(x) \qquad \text{with } P_0(x) = 1, P_1(x) = x,$$

$$N_{r,n} = \binom{r+n-1}{r-1} - \binom{r+n-3}{r-1},$$

$$h_n(\mathbf{x}, \mathbf{x}') = \sqrt{N_{r,n}} P_n(\langle \mathbf{x}, \mathbf{x}' \rangle) \qquad \text{for } \mathbf{x}, \mathbf{x}' \in \mathbb{S}^{r-1} \times \mathbb{S}^{r-1},$$

$$L_n^{\mathbf{x}} = h_n(\mathbf{x}, \mathbf{x}').$$

P_n is the n^{th} r-dimensional Legendre Polynomial. In [2] the following are shown:

Fact 2. For $r \geq 2$, the sequence $\{\sqrt{N_{r,b}} P_n\}_{n=0}^{\infty}$ is an orthonormal basis of the Hilbert space $L_2(\mu_r)$.

Fact 3. For every n, $\|P_n\|_{\ell_\infty} = 1$ and $P_n(1) = 1$.

Fact 4. $\left\langle L_i^{\mathbf{x}}, L_j^{\mathbf{x}'} \right\rangle = \delta_{ij} P_i(\langle \mathbf{x}, \mathbf{x}' \rangle)$, where δ_{ij} is the Dirac delta function. (Note this follows quickly from Fact 2).

We will also be interested in the approximability of a function f by polynomials of bounded degree. To that end, let $\mathcal{P}_{n,r} \subset L_2(\mu_r)$ denote the subspace of polynomials of degree at most $n-1$ and define

$$A_{n,r}(f) = \min_{p \in \mathcal{P}} \|f - p\|_{L_2(\mu_r)},$$

noting that such a minimum exists by the Hilbert projection theorem. As defined, $A_{n,r}(f)$ is also the norm of the projection $\mathbf{P}_{n,r}(f)$ of f on the orthogonal complement of $\mathcal{P}$; *i.e.,* $A_{n,r} = \|\mathbf{P}_{n,r}(f)\|_{L_2(\mu_r)}$. We will call this *Fact 5.*

Lemma 3.6. *Let f be an inner product function anr $g_1, \ldots, g_k$ be separable functions. Then*

$$\|f - \textstyle\sum_{i=1}^k g_i\|^2 \geq A_{n,r}(f) \left(A_{n,r}(f) - \frac{2\sum_{i=1}^k \|g_i\|}{\sqrt{N_{r,n}}} \right).$$

Proof. First we observe two Hilbert space isomorphisms. Let $\mathcal{H}_r \subset L_2(\mathbb{S}^{r-1} \times \mathbb{S}^{r-1})$ be the space of inner product functions. Then for $f = \phi(\langle \cdot, \cdot \rangle) \in \mathcal{H}_r$

$$\begin{aligned}
\|f\|^2 &= \int_{\mathbf{x} \in \mathbb{S}^{r-1}} \int_{\mathbf{x}' \in \mathbb{S}^{r-1}} \phi(\langle \mathbf{x}, \mathbf{x}' \rangle) \, \mathrm{d}\mathbf{x}' \, \mathrm{d}\mathbf{x} \\
&= \int_{\mathbf{x} \in \mathbb{S}^{r-1}} \phi(\langle \mathbf{x}, 1 \rangle) \, \mathrm{d}\mathbf{x} && \text{by symmetry} \\
&= \int_{x_1 \in [-1,1]} \phi(x_1) \, \mathrm{d}\mu(x_1) && \text{by Fact 1.} \\
&= \|\phi\|^2_{L_2(\mu)}
\end{aligned}$$

Therefore we have an isomorphism of the Hilbert spaces $L_2(\mu_r)$ and $\mathcal{H}_r$ given by $f \leftrightarrow \phi$. This implies that the projection operator $\mathbf{P}_{n,r}$ truncates Legendre polynomial representations of inner product functions. That is,

$$\mathbf{P}_{n,r} \left(\sum_{i=0}^{\infty} \alpha_i h_i \right) = \sum_{i=n}^{\infty} \alpha_i h_i.$$

Let $\mathbf{v}, \mathbf{v}' \in \mathbb{S}^{r-1}$ and denote the space of $(\mathbf{v}, \mathbf{v}')$-separable functions as $\mathcal{H}_{\mathbf{v},\mathbf{v}'} \subset L_2(\mathbb{S}^{r-1} \times \mathbb{S}^{r-1})$. Our second isomorphism is between $L_2(\mu_r \times \mu_r)$ and $\mathcal{H}_{\mathbf{v},\mathbf{v}'}$ via the mapping $f \leftrightarrow \psi$ and is derived similarly. Note that in particular, this isomorphism sends the orthonormal basis $\{\sqrt{N_{r,n}} P_n \otimes \sqrt{N_{r,m}} P_m\}_{n,m=0}^{\infty}$ to $\{L_n^{\mathbf{v}} \otimes L_m^{\mathbf{v}'}\}_{n,m=0}^{\infty}$, where $\otimes$ is the product in the respective Hilbert spaces.

Further, observe that

$$\begin{aligned}
\int_{\mathbf{x}} \int_{\mathbf{x}'} h_n(\mathbf{x}, \mathbf{x}') L_i^{\mathbf{v}}(\mathbf{x}) L_j^{\mathbf{v}'}(\mathbf{x}') \, \mathrm{d}\mathbf{x}' \, \mathrm{d}\mathbf{x} &= \int_{\mathbf{x}} L_i^{\mathbf{v}}(\mathbf{x}) \int_{\mathbf{x}'} h_n(\mathbf{x}, \mathbf{x}') L_j^{\mathbf{v}'}(\mathbf{x}') \, \mathrm{d}\mathbf{x}' \, \mathrm{d}\mathbf{x} \\
&= \int_{\mathbf{x}} L_i^{\mathbf{v}}(\mathbf{x}) \left\langle L_n^{\mathbf{x}}, L_j^{\mathbf{v}'} \right\rangle \mathrm{d}\mathbf{x} \\
&= \delta_{nj} \int_{\mathbf{x}} L_i^{\mathbf{v}}(\mathbf{x}) P_n(\langle \mathbf{x}, \mathbf{v}' \rangle) \, \mathrm{d}\mathbf{x} && \text{by Fact 4} \\
&= \frac{\delta_{nj}}{\sqrt{N_{r,n}}} \left\langle L_i^{\mathbf{v}}, L_n^{\mathbf{v}'} \right\rangle && \text{def'n of } L_n^{\mathbf{v}'} \\
&= \frac{\delta_{in} \delta_{jn} P_n(\langle \mathbf{v}, \mathbf{v}' \rangle)}{\sqrt{N_{r,n}}}. && (3.1)
\end{aligned}$$

By Fact 2, we may write f as $f = \sum_{i=0}^{\infty} \alpha_i h_i$. Also, let $g = \sum_{j=1}^{k} g_j$ where, by the given, g_j depends on $\langle \mathbf{v}, \mathbf{x} \rangle, \langle \mathbf{v}', \mathbf{x}' \rangle$ for $\mathbf{v}, \mathbf{v}' \in \mathbb{S}^{r-1}$. Again from Fact 2, we may write each g_j as $g_j(\mathbf{x}, \mathbf{x}') = \sum_{\ell,q=0}^{\infty} \beta_{\ell,q}^j L_\ell^{\mathbf{v}_j}(\mathbf{x}) L_q^{\mathbf{v}_j'}(\mathbf{x}')$.

Now certainly $\|f - g\|^2 \geq \|f\|^2 - 2\langle f, g \rangle$. But by (3.1) we see that f is orthogonal to $L_\ell^{\mathbf{v}_j} \otimes L_q^{\mathbf{v}_j'}$ whenever ℓ and q differ. So we may replace each g_j with $\sum_{\ell=0}^{\infty} \beta_\ell^j L_\ell^{\mathbf{v}_j}(\mathbf{x}) L_\ell^{\mathbf{v}_j'}(\mathbf{x}')$ to obtain

$$\begin{aligned}
\|f - g\|^2 &= \sum_{i=0}^{\infty} \left\| \alpha_i h_i - \sum_{j=1}^{k} \beta_i^j \left(L_i^{\mathbf{v}_j} \otimes L_i^{\mathbf{v}_j'} \right) \right\|^2 \\
&\geq \sum_{i=n}^{\infty} \left\| \alpha_i h_i - \sum_{j=1}^{k} \beta_i^j \left(L_i^{\mathbf{v}_j} \otimes L_i^{\mathbf{v}_j'} \right) \right\|^2 \\
&\geq \sum_{i=n}^{\infty} \alpha_i^2 - 2 \sum_{i=n}^{\infty} \sum_{j=1}^{k} \left\langle \alpha_i h_i,\ \beta_i^j \left(L_i^{\mathbf{v}_j} \otimes L_i^{\mathbf{v}_j'} \right) \right\rangle \\
&= \|\mathbf{P}_{n,r}(f)\|^2 - 2 \sum_{i=n}^{\infty} \sum_{j=1}^{k} \frac{\alpha_i \beta_i^j P_i(\langle \mathbf{v}_j, \mathbf{v}_j' \rangle)}{\sqrt{N_{r,k}}} \qquad \text{by the isometries} \\
&\geq \|\mathbf{P}_{n,r}(f)\|^2 - 2 \sum_{j=1}^{k} \sum_{i=n}^{\infty} \frac{|\alpha_i||\beta_i^j|}{\sqrt{N_{r,k}}} \qquad \text{by Fact 2} \\
&\geq \|\mathbf{P}_{n,r}(f)\|^2 - \frac{2}{\sqrt{N_{d,n}}} \sum_{j=1}^{k} \left(\sum_{i=n}^{\infty} |\alpha_i|^2 \right)^{\frac{1}{2}} \left(\sum_{i=n}^{\infty} |\beta_i^j|^2 \right)^{\frac{1}{2}} \\
&\geq \|\mathbf{P}_{n,r}(f)\|^2 - \frac{2\|\mathbf{P}_{n,r}(f)\| \sum_{j=1}^{k} \|g_j\|}{\sqrt{N_{r,n}}} \text{ by the triangle inequality,}
\end{aligned}$$

and using Fact 5 to replace $\|\mathbf{P}_{n,r}(f)\|$ with $A_{n,r}(f)$ we have our result. □

Lemma 3.7. *Let $g_{r,m}(x) = \sin(\pi \sqrt{r} m x)$. Then for sufficiently large r and any $n \geq 0$*

$$A_{n,r}(g_{r,m}(x)) \geq \sqrt{\frac{m-n}{4e\pi m}}$$

Refer to the original paper for [7] a proof of this lemma; the work amounts to counting intervals between in $[-1, 1]$ in which g and p differ in sign and summing their areas.

Proof of 3.5, Part 1. Write F as $f(\langle \cdot, \cdot \rangle)$ for some $f : [-1, 1] \to \mathbb{R}$. Observe that any neural unit with weights bounded by B implements a separable function with norm at most $B \max_{|x| \leq \sqrt{4r}B} |\sigma(x)|$ and that the output bias is also (vacuously) separable

with norm at most B. Thus by Lemma 3.6 we have for all depth-two networks $\mathcal{N}$ with k hidden units,

$$\|f - \mathcal{N}\|^2 \geq A_{n,r}(f) \left(A_{n,r}(f) - \frac{2kB \max_{|x| \leq \sqrt{4r}B} |\sigma(x)| + 2B}{\sqrt{N_{r,n}}} \right).$$

Set $m = r^3/\sqrt{r}$ so $f = g_{r,m} = \sin(\pi r^3 x)$, and set $n = r^2$. Then

$$A_{n,r}(f) \geq \sqrt{\frac{r^{5/2} - r^2}{4e\pi r^{5/2}}}.$$

The limit of the RHS is $1/(2\sqrt{e\pi})$ so for sufficiently large r we have $A_{n,r}(f) \geq 1/(5e\pi)$. Hence to have a $1/(50e^2\pi^2)$-approximation of F, the number of hidden neurons k must be at least

$$\frac{\sqrt{N_{r,r^2}}}{20e\pi 2^{2r}(1 + \sqrt{4r}) + 2^{r+1}} = 2^{\Omega(r \log r)}. \qquad \square$$

Existence of an efficient depth-three approximation:

This part of Theorem 3.5 relies on a construction in [8] to square the input in a single hidden layer. This construction is then modified to compute the inner product, and another hidden layer is added to compute the sin function. We direct the reader to the original paper [7] for a more thorough treatment.

Observe that in order to obtain a strong separation, Daniely had to again exploit the special structure of depth-two neural networks. In particular, these techniques don't suggest an easy generalization to finding a depth 3-4 separation or similar. It appears that when depth is any greater than 2, it is drastically easier to construct a specific network and analyze it, in comparison to making claims that a certain approximation is impossible. The next result gives separations at any depth, but must sacrifice the sharpness of the separation in return for generality.

3.3 Loose exponential separation at arbitrary depth

In [34] Telgarsky proves a general exponential depth separation using any units of the following sort:

Definition 14. A function $f : \mathbb{R}^r \to \mathbb{R}$ is (t, α, β)-semi-algebraic if for some m there are polynomials $\{p_i\}_{i=1}^m$ each of maximum total degree β, polynomials $\{q_i\}_{i=1}^t$ of total degree at most α, and for each $i \in [m]$, subsets L_i, U_i of $[t]$ such that

$$f(\mathbf{x}) = \sum_{i=1}^{m} \begin{cases} p_i(\mathbf{x}) & \text{if } q_j(\mathbf{x}) < 1/2 \text{ for all } j \in L_i \text{ and } q_j(\mathbf{x}) \geq 1/2 \text{ for all } j \in U_i, \\ 0 & \text{otherwise}^1. \end{cases}$$

Seen another way, the q_j define regions of $\mathbb{R}^r$ in which we may select, via the definitions of L_i, U_i, which p_i are included in the sum. Notes the absence of m in the name (t, α, β)-algebraic. Telgarsky gives somewhat vague reasons for this, but here's a more direct argument: we claim m is never more than 3^t. Indeed, there are at most 3^t ways to restrict a polynomial p_i according to the q_js. So if $m > 3^t$, then two polynomials, say p, p' have identical restrictions. We can thus replace them both with the single polynomial $p + p'$, and if we repeat this process for all duplicates, we now have $m \leq 3^t$. Hence a bound on the size of m is implicit in the inclusion of t in '(t, α, β)-semi-algebraic'.

The definition of semi-algebraic is quite general. We are particularly interested in the fact that ReLUs are $(1, 1, 1)$-semi-algebraic, though the class (for appropriate (t, α, β) also contains max and min gates, as well as decision trees. (See the original paper [34] for more).

Further, Telgarsky proves his result for *fully connected* architectures, defined as follows:

Definition 15. If the vertices of $\mathcal{A}$ can be placed in an ordered partition $\mathcal{P} = (\ell_1, \ldots, \ell_d)$ such that the subgraph induced by $\ell_i \cup \ell_{i+1}$ for all i is a fully-connected bipartite graph and there are no edges between vertices in non-adjacent ℓ_i, ℓ_j, we say $\mathcal{A}$ is a *fully-connected* architecture.

This is not a major restriction, however, as it's easy to see that non-fully-connected architectures can be embedded in fully-connected architectures without much overhead.

Theorem 3.8. *Let $d, r \geq 1$ and let $\mathcal{C}$ denote the set of functions computed by depth-d networks with at most $2^d/(t\alpha\beta)$ (t, α, β)-semi-algebraic units. Then there exists $f : \mathbb{R}^r \to \mathbb{R}$ computed exactly by an ReLU neural network of depth $d^3 + 5$, size $2d^3 + 4$, and $4 + r$ distinct programmable parameters such that*

$$\inf_{g \in \mathcal{C}} \int_{[0,1]^r} |f(\mathbf{x}) - g(\mathbf{x})| \, \mathrm{d}\mathbf{x} \geq \frac{1}{32}.$$

We will prove this in a series of lemmas, following Telgarsky's outline: first we show that functions with few oscillations are bad approximators of those with many; then we show shallow semi-algebraic networks exhibit few oscillations; finally we construct

[1]Note that this is a slightly different, but equivalent, definition to that which Telgarsky presents; in particular, in his paper there are zeros where there are presently 1/2s. This change was made to deal with a slight inconsistency in the definition of *crossing number* (to be defined on the next page) and its use in Lemma 3.6 of the original paper.

a deep ReLU network with many oscillations. In comparison to Telgarsky's original version we condense the exposition of each lemma at the expense of generality, though the actual theorem statement here reflects a slight improvement of his bounds.

First, a couple definitions. Let $f : \mathbb{R} \to \mathbb{R}$ be piecewise-continuous and define $\bar{f} = \mathbb{1}_{\{x : f(x) \geq 1/2\}}$. Then the set $\{x : \bar{f} = 1\}$ is the union of a set U of disjoint intervals. This means $\mathbb{R} - U$ is also the union of a set L of disjoint intervals. Define $\mathcal{I}_f = U \cup L$ and say the *crossing number* of f is $\mathrm{Cr}(f) = |\mathcal{I}_f|$. In general the crossing number may not be finite, but it will in our case because we are working with compositions of piecewise-polynomial functions.

Lemma 3.9. *Let f and g be piecewise-continuous maps $\mathbb{R} \to \mathbb{R}$ with finite crossing numbers. For an interval $U \in \mathcal{I}_f$, say f is* badly approximated by g in U *if for all $x \in I$, $\bar{f} \neq \bar{g}$. Let $B(\mathcal{I}_f)$ be the number of intervals in $\mathcal{I}_f$ in which f is badly approximated by g. Then*

$$\frac{B(\mathcal{I}_f)}{\mathrm{Cr}(f)} \geq \frac{1}{2} - \frac{\mathrm{Cr}(g)}{\mathrm{Cr}(f)}.$$

Proof. For each $J \in \mathcal{I}_g$, define $X_J = \{I \in \mathcal{I}_f : I \subseteq J\}$. Note that $\bar{g}$ is fixed on a given J, whereas $\bar{f}$ alternates, so the number of badly approximated intervals in X_J is least when $|X_J|$ is odd and the leftmost interval in X_j is not badly approximated, giving the bound $B(X_J) \geq (|X_J| - 1)/2$. Thus

$$\begin{aligned} \frac{B(\mathcal{I}_f)}{\mathrm{Cr}(f)} \geq \frac{1}{\mathrm{Cr}(f)} \sum_{J \in \mathcal{I}_g} B(X_J) &\geq \frac{1}{\mathrm{Cr}(f)} \sum_{J \in \mathcal{I}_g} \frac{|X_J| - 1}{2} \\ &= \frac{\sum_{J \in \mathcal{I}_g} |X_J| - \mathrm{Cr}(g)}{2\,\mathrm{Cr}(f)} \end{aligned} \tag{3.2}$$

Observe that $\mathcal{I}_f - \cup_{J \in \mathcal{I}_g} X_J$ is the set of intervals which straddle a boundary between intervals in $\mathcal{I}_g$. There are at most $\mathrm{Cr}(g) - 1$ such boundaries, so $\mathrm{Cr}(f) \leq \mathrm{Cr}(g) + \sum_{J \in \mathcal{I}_g} |X_J|$, implying

$$\sum_{J \in \mathcal{I}_g} |X_J| \geq \mathrm{Cr}(f) - \mathrm{Cr}(g).$$

Substitution into (3.2) gives the result. □

Given a neural network of semialgebraic units implementing function f, we now give a bound on the crossing number of the restriction of f to any line through its domain. That is, using (t, γ)-poly to denote piecewise polynomials with t polynomial intervals of maximum total degree γ,

Lemma 3.10. *Suppose $f : \mathbb{R}^r \to \mathbb{R}$ is implemented by a fully-connected network of depth d with m (t, α, β)-semi-algebraic gates, $\alpha, \beta \geq 1$. Let $h : \mathbb{R} \to \mathbb{R}^r$ be an affine map. Then $\mathrm{Cr}(f \circ h) \leq 2(2tm\alpha/d)^d \beta^{d^2}$.*

This is proved by induction, aided by the following result.

Technical Lemma. *Suppose $f : \mathbb{R}^k \to \mathbb{R}$ is (s, α, β)-semi-algebraic and $(g_1, \ldots, g_k)$ are (t, γ)-poly. Then $f(g_1, \ldots, g_k(x))$ is $(stk(1 + \alpha\gamma), \beta\gamma)$-poly.*

Proof. Let $\{q_i\}_{i=1}^s$ be the set of polynomials defining the regions for f and let $Q_i = q_i(g_1, \ldots, g_k)$.

First, we claim that for all i, Q_i is $(tk, \alpha\gamma)$-poly: observe that for each i there exists a partition of $\mathbb{R}$ into at most tk intervals which is a refinement of each of the g's partitions, and that Q_i is polynomial over each of these intervals.

Now for each i consider $\mathcal{I}_{Q_i}$. By definition, each polynomial segment of the function $Q_i + 1/2$ has at most $\alpha\gamma$ roots, so each segment contributes at most $\alpha\gamma + 1$ intervals to $\mathrm{Cr}(Q_i)$, for a total bound of $tk(1 + \alpha\gamma)$.

Now observe that the tk-sized partitions of $\mathbb{R}$ associated with each Q_i have a mutual refinement of size $stk(1 + \alpha\gamma)$. Thus $f(g_1, \ldots, g_k)$ is a fixed polynomial of degree at most $\beta\gamma$ over any one of these intervals, as desired. □

Proof of Lemma 3.10. Suppose $g : \mathbb{R}^r \to \mathbb{R}$ is the (t, α, β)-semi-algebraic function computed by some unit in the first layer of the given network. Then because h is linear in each of its coordinates, $g \circ h : \mathbb{R} \to \mathbb{R}$ is still (t, α, β)-semi-algebraic. Thus the network obtained by replacing each unit g in the first layer with the unit $g \circ h$ yields a network implementing $f \circ h$ with units satisfying the same constraints as the original network.

Let m_i be the number of units in each layer of this network (so $\sum_{i=1}^d m_i = m$) and for notational convenience let $B_i = \beta^{\sum_{j=1}^{i-1} j} = \beta^{i(i-1)/2}$. We will first show by induction on i that all units in layer $i = 1, 2, \ldots, d$ are $\big((2t\alpha)^i B_i (\prod_{j=1}^{i-1} m_i), \beta^i\big)$-poly We will then relax this bound to prove the lemma.

As a convenient base case, suppose $d = 0$. Then the network consists of a single (t, α, β)-semi-algebraic unit implementing a function $g : \mathbb{R} \to \mathbb{R}$. Treating the input vertex as the identity function (which is $(1, 1)$-poly), we may apply the technical lemma to get that g is $(t(1 + \alpha), \beta)$-poly, and is thus $(2t\alpha, \beta)$-poly because $\alpha \geq 1$.

Now consider a unit v in layer i, which by the given is (t, α, β)-semi-algebraic and connected to at most m_{i-1} preceding nodes. Then if each of its preceding units

are (ℓ, γ)-poly, by the technical lemma, v is $(t\ell m_{i-1}(1+\alpha\gamma),\, \beta\gamma)$-poly. Now by the inductive hypothesis each of the m_{i-1} units in layer $i-1$ are

$$\big((2t\alpha)^{i-1} B_{i-1}(\textstyle\prod_{j=1}^{i-2} m_i),\, \beta^{i-1}\big)\text{-poly},$$

and so substituting we have that v is

$$\big(2^{i-1} t^i \alpha^{i-1} (\textstyle\prod_{j=1}^{i-1} m_j) B_{i-1}(1+\alpha\beta^{i-1}),\, \beta^i\big)\text{-poly}.$$

This entails the desired bound since $1+\alpha\beta^{i-1} \le 2\alpha\beta^{i-1}$. In particular, the output unit (and thus $f \circ h$) is

$$\big((2t\alpha)^d \beta^{d(d-1)/2} \textstyle\prod_{i=1}^{d-1} m_i,\, \beta^d\big)\text{-poly}$$

We now relax this bound to prove the lemma. Noting that there is one unit in layer d, by Jensen's inequality we have

$$\begin{aligned}\ln\left(\prod_{i=1}^{d-1} m_i\right) &= \ln\left(\prod_{i=1}^{d} m_i\right) \\ &\le d\ln(m/d) = \ln(m/d)^d,\end{aligned}$$

and so $\prod_{i=1}^{d-1} m_i \le (m/d)^d$. Further, as we saw in the proof of the technical lemma, an (ℓ, γ)-poly function has crossing number upper-bounded by $\ell(1+\gamma)$. Thus $\mathrm{Cr}(f \circ h) \le (2tm\alpha/d)^d \beta^{d^2}(1+\beta^d) \le 2(2tm\alpha/d)^d \beta^{d^2}$ because m is at least d and $t, \alpha, \beta \ge 1$. □

And the last lemma, an explicit construction of a highly oscillatory function. We are now working in the restricted domain $[0,1]$, and so below $\mathcal{I}_f$ is the collection of intervals as before, but restricted to $[0,1]$.

Lemma 3.11. *Define $F(x) = 2\sigma(x) - 4\sigma(x-1/2)$ for $\sigma = \max\{0, x\}$ and $d \ge 1$. Then $\mathrm{Cr}(F^d) = 2^d + 1$, for each $U \in \mathcal{I}_{F^d}$,*

$$\int_U |F^d(x) - 1/2|\,\mathrm{d}x \ge 2^{-d-3}$$

and there exists a network of ReLUs of depth $d+1$ and at most 2 units per layer that implements F^d.

Proof. The crossing number of F^d is easy to check as F^d is a triangle wave with period 2^{-d}. Observe that for each $U \in \mathcal{I}_{F^d}$, $(F^d - 1/2)|_U$ makes a triangle with the x-axis of height $1/2$ and base 2^{-d-1} if $0 \in U$ or $1 \in U$, or 2^{-d} otherwise. Thus $\int_U |F^d(x) - 1/2|\,\mathrm{d}x \ge 2^{-d-1}/4 = 2^{-d-3}$.

It is easy to check an ReLU network of depth $d+1$ implementing F^d may be constructed as follows Layers $1,\dots,d$ each have two units. In layer one, the upper unit implements $\sigma(x)$ and the lower implements $\sigma(x-1/2)$. Each unit in layers $\ell=2,\dots,d$ has two inputs, say x from the upper unit of layer $\ell-1$ and y from the lower unit of layer $\ell-1$. Let the upper unit in layer ℓ therefore implement $\sigma(2x-4y)$ and the lower unit implement $\sigma(2x-4y-1/2)$. Finally, let the single unit of layer $d+1$ implement $\sigma(2x-4y)$. □

Proof of Theorem 3.8. Let $f_0(x)=F^{d^3+4}(x)$ and define $f:\mathbb{R}^r\to\mathbb{R}$ as $f(\mathbf{x})=f_0(x_1)$ for $\mathbf{x}=(x_1,\dots,x_r)$. Then by Lemma 3.11, we have an ReLU network implementing f with d^3+5 layers, $2d^3+9$ units, and $4+r$ distinct programmable parameters. Also by Lemma 3.11, $\operatorname{Cr}(f_0)=2^{d^3+4}+1$ and so for all $U\in\mathcal{I}_{f_0}$,

$$\begin{aligned}\int_U |f_0(x)-1/2|\,\mathrm{d}x &\geq \frac{1}{2^{d^3+7}}\\ &\geq \frac{1}{2^3(2^{d^3+4}+1)}=\frac{1}{8\operatorname{Cr}(f_0)}.\end{aligned}\tag{3.3}$$

Now suppose g is implemented by a network of depth d with at most m units, all (t,α,β)-semi-algebraic. For $\mathbf{y}=(y_2,\dots,y_r)\in\mathbb{R}^{r-1}$, define $h_{\mathbf{y}}:\mathbb{R}\to\mathbb{R}^r$ by $h_{\mathbf{y}}(x)=(x,y_2,\dots,y_r)$. Noting that h is affine, we have by Lemma 3.10 that for all $\mathbf{y}\in\mathbb{R}^{r-1}$,

$$\operatorname{Cr}(g\circ h_{\mathbf{y}})\leq 2(2tm\alpha\beta d)^d\beta^{d^2}\leq 4(tm\alpha\beta)^{d^2}\leq 2^{d^3+2}.\tag{3.4}$$

We now bound $\|f\circ h_{\mathbf{y}}-g\circ h_{\mathbf{y}}\|_1$ from below. For each $U\in\mathcal{I}_f$, let $B_U=1$ if $f\circ h_{\mathbf{y}}$ is badly approximated by $g\circ h_{\mathbf{y}}$ in U, and otherwise let $B_U=0$. Then for any $\mathbf{y}\in\mathbb{R}^{r-1}$,

$$\begin{aligned}\int_{[0,1]} |(f\circ h_{\mathbf{y}})(x)-(g\circ h_{\mathbf{y}})(x)|\,\mathrm{d}x &= \sum_{U\in\mathcal{I}_f}\int_U |(f\circ h_{\mathbf{y}})(x)-(g\circ h_{\mathbf{y}})(x)|\,\mathrm{d}x\\ &\geq \sum_{U\in\mathcal{I}_f}\int_U B_U|(f\circ h_{\mathbf{y}})(x)-1/2|\,\mathrm{d}x\\ &= \sum_{U\in\mathcal{I}_f} B_U\int_U |(f_0)(x)-1/2|\,\mathrm{d}x\\ &\geq \frac{1}{8\operatorname{Cr}(f_0)}\sum_{U\in\mathcal{I}_f}B_U=\frac{B(\mathcal{I}_{f_0})}{8\operatorname{Cr}(f_0)} && \text{by (3.3)}\\ &\geq \frac{1}{16}-\frac{\operatorname{Cr}(g\circ h_{\mathbf{y}})}{8\operatorname{Cr}(f_0)} && \text{by Lemma 3.9}\\ &\geq \frac{1}{16}\left(1-\frac{2(2^{d^3+2})}{2^{d^3+4}}\right)=\frac{1}{32}. && \text{by (3.4)}\end{aligned}$$

According to this bound we therefore have

$$\int_{[0,1]^d} |f(\mathbf{x}) - g(\mathbf{x})| \, \mathrm{d}\mathbf{x} = \int_{[0,1]^{d-1}} \int_{[0,1]} |(f \circ h_{\mathbf{y}})(x) - (g \circ h_{\mathbf{y}})(x)| \, \mathrm{d}x \, \mathrm{d}\mathbf{y} \geq \frac{1}{32}. \quad \square$$

In closing, we note that both Daniely and Telgarsky ended up counting oscillations, though tighter separations (like Daniely's) appear to require more careful analysis.

Further, if we could characterize entire classes of functions that are amenable to these sorts of separations, that would go a long way towards determining some recommendations for network designs to avoid underfitting. The Daniely result does some of that—the full result in the original paper does characterize classes of functions that give such a separation. But even still these classes are restricted to the somewhat strange domain of $\mathbb{S}^{r-1} \times \mathbb{S}^{r-1}$.

One general pattern to note among all of these separating functions is that of their *compositionality*. Safran & Shamir's is the composition of the norm and a step function, Daniely's is the composition of the inner product and a sine function, and Telgarsky's is a collection of compositions. If we were to search for a way to separate depth 3 networks from depth 4, we may want to consider other function compositions that cannot be easily "flattened."

Conclusion

Understanding the expressivity of neural networks is a difficult problem for a number of reasons. Some of these are practical, in that it is a challenge to settle on a definition of a neural network that contains all ANN-like objects used in practice, and it's hard to guarantee that any result will be future proof as we're still unsure what the essential properties of a given network model are.

But as this thesis attempts to illustrate, there are deep mathematical challenges embedded in ANN architectures as well. These find their way in through ANNs' similarity with boolean circuits, but also from approximation theory. In particular, as we observed at least once in each chapter, many of the extant techniques are tied tightly to the special structure of depth-two networks. It appears that the theoretical community still awaits a better general understanding of function composition, though the study of fractals (e.g., [20], [21]) may engender this in the future.

At the same time, there appears to be hope for the development of underfitting avoidance strategies: understanding the relationship between "compositionality" and separation results may be a fruitful research avenue, one which Poggio (e.g., [24]) and others have already started down. That is, if it is known that a target function decomposes into a number of simpler functions, efficient deep representation may be relatively easy to find.

The question, then, becomes how one might detect the "compositionality" in samples from f. Certainly "compositionality" only has meaning when placed in the context of a set of functions that one wishes to compose together to construct f. In the ANNs here, that set has been $\{\sigma(\mathbf{w}\cdot\mathbf{x}+b)\}$ for appropriate weights and biases, and as the separation results have demonstrated, depth lends these simple nonlinearities efficient means by which to oscillate.

Indeed, each separation result made use of oscillations in some way. In particular, consider Telgarsky's crossing number, $\mathrm{Cr}(f)$. Supposing we had direct access to f, by measuring its crossing number and using the upper bound estimates of crossing numbers of networks proved in Lemma 3.10, along with the error estimate proved

in Lemma 3.9, we may discover lower bounds on the size, depth, or width that is required for learning close approximations to f.

Once we remove direct access to f, and must infer its crossing number from a sample, even more questions arise. How many samples are required to get a good approximation of $\mathrm{Cr}(f)$? What is the computational complexity of this task? Can we create better, finer measurements that are tailored for specific activation functions?

Another research direction would be to characterize the relationship between different classes of compositions of Barron's Γ functions (as explored in Corollary 2.5.1). Can we prove 'no-flattening' theorems that guarantee compositions thereof are most efficiently represented as deep networks? Might it help us discover a depth 3-4 separation?

Whatever the particular answers to these questions might be, it is clear that the study of neural network expressivity stands as an example of the interconnectedness of applied investigation and theoretical exploration. May it encourage great discoveries in the years ahead.

Bibliography

[1] Filippo Amato, Alberto López, Eladia María Peña-Méndez, Petr Vaňhara, Aleš Hampl, and Josef Havel. Artificial neural networks in medical diagnosis. *Journal of Applied Biomedicine*, 11:47–58, 2013.

[2] Kendall Atkinson and Weimin Han. *Spherical harmonics and approximations on the unit sphere: An introduction*, volume 2044. Springer, Heidelberg, 2012.

[3] A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, may 1993.

[4] Avrim Blum and Ronald L Rivest. Training a 3-Node Neural Network is NP-Complete. In D S Touretzky, editor, *Advances in Neural Information Processing Systems 1*, pages 494–501. Morgan-Kaufmann, 1989.

[5] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The Loss Surfaces of Multilayer Networks. nov 2014.

[6] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, dec 1989.

[7] Amit Daniely. Depth Separation for Neural Networks. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 690–696, Amsterdam, Netherlands, 2017. PMLR.

[8] Ronen Eldan and Ohad Shamir. The Power of Depth for Feedforward Neural Networks. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 907–940, Columbia University, New York, New York, USA, 2016. PMLR.

[9] Ken Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183–192, jan 1989.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[11] Simon Haykin. *Neural Networks and Learning Machines*. Prentice Hall, New York, 3 edition, 2009.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. dec 2015.

[13] Irie and Miyake. Capabilities of three-layered perceptrons. In *IEEE International Conference on Neural Networks*, pages 641–648 vol.1. IEEE, 1988.

[14] Stephen J. Judd. *Neural Network Design and the Complexity of Learning*. The MIT Press, Cambridge, Massachusetts, 1990.

[15] Marek Karpinski and Angus Macintyre. Polynomial Bounds for VC Dimension of Sigmoidal and General Pfaffian Neural Networks. *Journal of Computer and System Sciences*, 54(1):169–176, feb 1997.

[16] Pascal Koiran and Eduardo D. Sontag. Neural Networks with Quadratic VC Dimension. *Journal of Computer and System Sciences*, 54:190–198, 1997.

[17] Wolfgang Maass. Perspectives of Current Research about the Complexity of Learning on Neural Nets. In *Theoretical Advances in Neural Computation and Learning*, pages 295–336. Springer US, Boston, MA, 1994.

[18] Angus Macintyre and Eduardo D. Sontag. Finiteness results for sigmoidal "neural" networks. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing - STOC '93*, pages 325–334, New York, New York, USA, 1993. ACM Press.

[19] Matthew V Mahoney. Fast Text Compression with Neural Networks. In *Proc. AAAI FLAIRS*, Orlando, 2000. The AAAI Press.

[20] Peter Massopust. *Interpolation and Approximation with Splines and Fractals*. Oxford University Press, 2010.

[21] M. A. Navascués. Fractal Approximation. *Complex Analysis and Operator Theory*, 4(4):953–974, nov 2010.

[22] Nicholas Pippenger. On simultaneous resource bounds. In *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, pages 307–311. IEEE, oct 1979.

[23] G. Pisier. Remarques sur un résultat non publié de B. Maurey. In *Séminaire Analyse fonctionnelle (dit "Maurey-Schwartz")*, Ecole Polytechnique, Centre de Mathematiques, Palaiseau, 1980.

[24] Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and When Can Deep – but Not Shallow – Networks Avoid the Curse of Dimensionality: a Review. 2017.

[25] Hoifung Poon and Pedro Domingos. Sum-Product Networks: A New Deep Architecture. feb 2012.

[26] Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, 3rd edition, 1976.

[27] Walter Rudin. *Real and complex analysis*. McGraw-Hill, 1987.

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[29] Itay Safran and Ohad Shamir. Depth-Width Tradeoffs in Approximating Natural Functions with Neural Networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2979–2987, International Convention Centre, Sydney, Australia, 2017. PMLR.

[30] John E Savage. *Models of Computation: Exploring the Power of Computing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1997.

[31] Franco Scarselli and Ah Chung Tsoi. Universal Approximation Using Feedforward Neural Networks: A Survey of Some Existing Methods, and Some New Results. *Neural Networks*, 11(1):15–37, jan 1998.

[32] Schmitt and Michael. Lower bounds on the complexity of approximating continuous functions by sigmoidal neural networks, 1999.

[33] Claude. E Shannon. The Synthesis of Two-Terminal Switching Circuits. *Bell System Technical Journal*, 28(1):59–98, 1949.

[34] Matus Telgarsky. Benefits of depth in neural networks. In *JMLR: Workshop and Conference Proceedings*, volume 49, pages 1–23, 2016.

[35] Hiroto Yasuura. Width and depth of combinational logic circuits. *Information Processing Letters*, 13(4-5):191–194, jan 1981.

[36] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference for Learning Representations*, nov 2017.