Deep ReLU Networks Have Surprisingly Few Activation Patterns

Boris Hanin
Facebook AI Research
Texas A&M University
bhanin@math.tamu.edu

David Rolnick
University of Pennsylvania
Philadelphia, PA USA
drolnick@seas.upenn.edu

Abstract

The success of deep networks has been attributed in part to their expressivity: per parameter, deep networks can approximate a richer class of functions than shallow networks. In ReLU networks, the number of activation patterns is one measure of expressivity; and the maximum number of patterns grows exponentially with the depth. However, recent work has showed that the practical expressivity of deep networks – the functions they can learn rather than express – is often far from the theoretical maximum. In this paper, we show that the average number of activation patterns for ReLU networks at initialization is bounded by the total number of neurons raised to the input dimension. We show empirically that this bound, which is independent of the depth, is tight both at initialization and during training, even on memorization tasks that should maximize the number of activation patterns. Our work suggests that realizing the full expressivity of deep networks may not be possible in practice, at least with current methods.

1 Introduction

A fundamental question in the theory of deep learning is why deeper networks often work better in practice than shallow ones. One proposed explanation is that, while even shallow neural networks are universal approximators [3, 9, 12, 20, 27], there are functions for which increased depth allows exponentially more efficient representations. This phenomenon has been quantified for various complexity measures [6, 7, 8, 13, 24, 25, 28, 29, 30, 31]. However, authors such as Ba and Caruana have called into question this point of view [2], observing that shallow networks can often be trained to imitate deep networks and thus that functions learned in practice by deep networks may not achieve the full expressive power of depth.

In this article, we attempt to capture the difference between the maximum complexity of deep networks and the complexity of functions that are actually learned (see Figure 1). We provide theoretical and empirical analyses of the typical complexity of the function computed by a ReLU network $\mathcal{N}$. Given a vector θ of its trainable parameters, $\mathcal{N}$ computes a continuous and piecewise linear function $x \mapsto \mathcal{N}(x;\theta)$. Each θ thus is associated with a partition of input space $\mathbb{R}^{n_{\text{in}}}$ into activation regions, polytopes on which $\mathcal{N}(x;\theta)$ computes a single linear function corresponding to a fixed activation pattern in the neurons of $\mathcal{N}$.

We aim to count the number of such activation regions. This number has been the subject of previous work (see §1.1), with the majority concerning large *lower bounds* on the *maximum* over all θ of the number of regions for a given network architecture. In contrast, we are interested in the typical behavior of ReLU nets as they are used in practice. We therefore focus on small *upper bounds* for the *average* number of activation regions present for a typical value of θ. Our main contributions are:

- We give precise definitions and prove several fundamental properties of both linear and activation regions, two concepts that are often conflated in the literature (see §2).
- We prove in Theorem 5 an upper bound for the expected number of activation regions in a ReLU net $\mathcal{N}$. Roughly, we show that if n_{in} is the input dimension and $\mathcal{C}$ is a cube in input

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

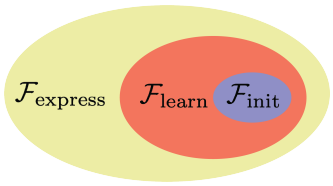

Figure 1: Schematic illustration of the space of functions $f : \mathbb{R}^{n_{\text{in}}} \to \mathbb{R}^{n_{\text{out}}}$. For a given neural network architecture, there is a set $\mathcal{F}_{\text{express}}$ of functions expressible by that architecture. Within this set, the functions corresponding to networks at initialization are concentrated within a set $\mathcal{F}_{\text{init}}$. Intermediate between $\mathcal{F}_{\text{init}}$ and $\mathcal{F}_{\text{express}}$ is a set $\mathcal{F}_{\text{learn}}$ containing the functions which the network has non-vanishing probability of learning using gradient descent. (These are, of course, not formal definitions and are intended to provide intuition for our results.) This paper seeks to demonstrate the gap between $\mathcal{F}_{\text{express}}$ and $\mathcal{F}_{\text{learn}}$ and that, at least for certain measures of complexity, there is a surprisingly small gap between $\mathcal{F}_{\text{init}}$ and $\mathcal{F}_{\text{learn}}$.

space $\mathbb{R}^{n_{\text{in}}}$, then, under reasonable assumptions on network gradients and biases,

$$\frac{\#\text{activation regions of } \mathcal{N} \text{ that intersect } \mathcal{C}}{\text{vol}(\mathcal{C})} \leq \frac{(T\#\text{neurons})^{n_{\text{in}}}}{n_{\text{in}}!}, \quad T \text{ constant.} \quad (1)$$

- This bound holds in particular for deep ReLU nets at initialization, and is in sharp contrast to the maximum possible number of activation patterns, which is exponential in depth [29, 34].
- Theorem 5 also strongly suggests that the bounds on number of activation regions continue to hold approximately throughout training. We empirically verify that this behavior holds, even for networks trained on memorization-based tasks (see §4 and Figures 3-6).

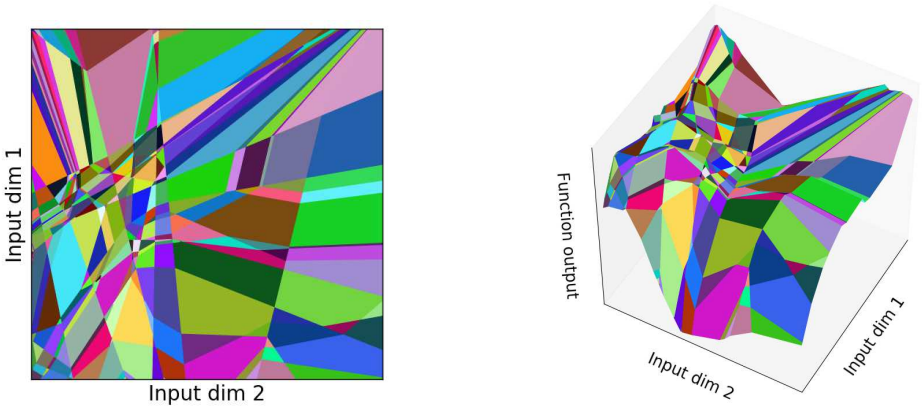

Figure 2: Function defined by a ReLU network of depth 5 and width 8 at initialization. Left: Partition of the input space into regions, on each of which the activation pattern of neurons is constant. Right: the function computed by the network, which is linear on each activation region.

Our results show both theoretically and empirically not only that the number of activation patterns can be small for a ReLU network but that it typically *is* small both at initialization and throughout training, effectively capped far below its theoretical maximum even for tasks where a higher number of regions would be advantageous (see §4). We provide in §3.2-3.3 two intuitive explanations for this phenomenon. The essence of both is that many activation patterns can be created only when a typical neuron z in $\mathcal{N}$ turns on/off repeatedly, forcing the value of its pre-activation $z(x)$ to cross the level of its bias b_z many times. This requires (i) significant overlap between the range of $z(x)$ on the different activation regions of $x \mapsto z(x)$ and (ii) the bias b_z to be picked within this overlap. Intuitively, (i) and (ii) require either large or highly coordinated gradients. In the former case, $z(x)$ oscillates over a large range of outputs and b_z can be random, while in the latter $z(x)$ may oscillate only over a small range of outputs and b_z is carefully chosen. Neither is likely to happen with a proper initialization. Moreover, both appear to be difficult to learn with gradient-based optimization.

The rest of this article is structured as follows. Section 2 gives formal definitions and some important properties of both activation regions and the closely related notion of linear regions (see Definitions 1 and 2).Section 3 contains our main technical result, Theorem 5, stated in §3.1. Sections 3.2 and

3.3 provide heuristics for understanding Theorem 5 and its implications. Finally, §4 is devoted to experiments that push the limits of how many activation regions a ReLU network can learn in practice.

1.1 Relation to Prior Work

We consider the typical number of activation regions in ReLU nets. In contrast, interesting bounds on the maximum number of regions are given in [6, 8, 25, 28, 29, 31, 32, 34]. Such worst case bounds are used to control the VC-dimension of ReLU networks in [4, 5, 18] but differ dramatically from our average case analysis.

Our main theoretical result, Theorem 5, is related to [17], which conjectured that our Theorem 5 should hold and proved bounds for other notions of average complexity of activation regions. Theorem 5 is also related in spirit to [10], which uses a mean field analysis of wide ReLU nets to show that they are biased towards simple functions. Our empirical work (e.g. §4) is related both to the experiments of [26] and to those of [1, 36]. The last two observe that neural networks are capable of fitting noisy or completely random data. Theorem 5 and experiments in §4 give a counterpoint, suggesting limitations on the complexity of random functions that ReLU nets can fit in practice (see Figures 4-6).

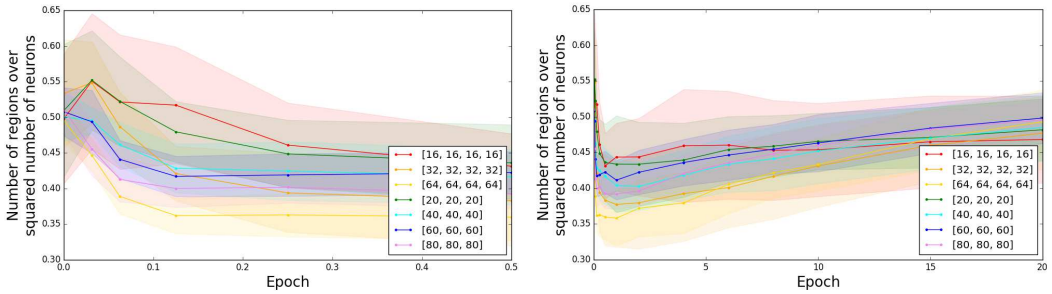

Figure 3: The average number of activation regions in a 2D cross-section of input space, for fully connected networks of various architectures training on MNIST. Left: a closeup of 0.5 epochs of training. Right: 20 epochs of training. The notation $[20, 20, 20]$ indicates a network with three layers, each of width 20. The number of activation regions starts at approximately $(\#\text{neurons})^2/2$, as predicted by Theorem 5 (see Remark 1). This value changes little during training, first decreasing slightly and then rebounding, but never increasing exponentially. Each curve is averaged over 10 independent training runs, and for each run the number of regions is averaged over 5 different 2D cross-sections, where for each cross-section we count the number of regions in the (infinite) plane passing through the origin and two random training examples. Standard deviations between different runs are shown for each curve. See Appendix A for more details.

2 How to Think about Activation Regions

Before stating our main results on counting activation regions in §3, we provide a formal definition and contrast them with linear regions in §2.1. We also note in §2.1 some simple properties of activation regions that are useful both for understanding how they are built up layer by layer in a deep ReLU net and for visualizing them. Then, in §2.2, we explain the relationship between activation regions and arrangements of bent hyperplanes (see Lemma 4).

2.1 Activation Regions vs. Linear Regions

Our main objects of study in this article are activation regions, which we now define.

Definition 1 (Activation Patterns/Regions)**.** *Let $\mathcal{N}$ be a ReLU net with input dimension n_{in}. An activation pattern for $\mathcal{N}$ is an assignment to each neuron of a sign:*

$$\mathcal{A} \;:=\; \{a_z,\ z \text{ a neuron in } \mathcal{N}\} \in \{-1, 1\}^{\#\text{neurons}}.$$

Fix θ, a vector of trainable parameters in $\mathcal{N}$, and an activation pattern $\mathcal{A}$. The activation region corresponding to $\mathcal{A}, \theta$ is

$$\mathcal{R}(\mathcal{A}; \theta) := \{x \in \mathbb{R}^{n_{\text{in}}} \quad | \quad (-1)^{a_z}\,(z(x;\theta) - b_z) > 0, \quad z \text{ a neuron in } \mathcal{N}\},$$

where neuron z has pre-activation $z(x;\theta)$, bias b_z, and post-activation $\max\{0, z(x;\theta) - b_z\}$. We say the activation regions of $\mathcal{N}$ at θ are the non-empty activation regions $\mathcal{R}(\mathcal{A}, \theta)$.

When specifying an activation pattern $\mathcal{A}$, the sign a_z assigned to a neuron z determines whether z is on or off for inputs in the activation region $\mathcal{R}(\mathcal{A};\theta)$ since the pre-activation $z(x;\theta) - b_z$ of neuron z is positive (resp. negative) when $a_z = 1$ (resp. $a_z = -1$). Perhaps the most fundamental property of activation regions is their convexity.

Lemma 1 (Convexity of Activation Regions)**.** *Let $\mathcal{N}$ be a ReLU net. Then for every activation pattern $\mathcal{A}$ and any vector θ of trainable parameters for $\mathcal{N}$ each activation region $\mathcal{R}(\mathcal{A};\theta)$ is convex.*

We note that Lemma 1 has been observed before (e.g. Theorem 2 in [29]), but in much of the literature the difference between linear regions (defined below), which are not necessarily convex, and activation regions, which are, is ignored. It turns out that Lemma 1 holds for *any* piecewise linear activation, such as leaky ReLU and hard hyperbolic tangent/sigmoid. This fact seems to be less well-known (see Appendix B.1 for a proof). To provide a useful alternative description of activation regions for a ReLU net $\mathcal{N}$, a fixed vector θ of trainable parameters and neuron z of $\mathcal{N}$, define

$$H_z(\theta) \;:=\; \{x \in \mathbb{R}^{n_{\text{in}}} \mid z(x;\theta) = b_z\}. \tag{2}$$

The sets $H_z(\theta)$ can be thought of as "bent hyperplanes" (see Lemma 4). The non-empty activation regions of $\mathcal{N}$ at θ are the connected components of $\mathbb{R}^{n_{\text{in}}}$ with all the bent hyperplanes $H_z(\theta)$ removed:

Lemma 2 (Activation Regions as Connected Components)**.** *For any ReLU net $\mathcal{N}$ and any vector θ of trainable parameters*

$$\{\textit{non-empty activation regions } \mathcal{R}(\mathcal{A};\theta)\} \;=\; \text{connected components}\big(\mathbb{R}^{n_{\text{in}}} \setminus \bigcup_{\text{neurons } z} H_z(\theta)\big). \tag{3}$$

We prove Lemma 2 in Appendix B.2. We may compare activation regions with *linear regions*, which are the regions of input space on which the network defines different linear functions.

Definition 2 (Linear Regions)**.** *Let $\mathcal{N}$ be a ReLU net with input dimension n_{in}, and fix θ, a vector of trainable parameters for $\mathcal{N}$. Define*

$$\mathcal{B}_{\mathcal{N}}(\theta) \;:=\; \{x \in \mathbb{R}^{n_{\text{in}}} \mid \nabla\mathcal{N}(\cdot\,;\theta) \textit{ is discontinuous at } x\}.$$

The linear regions of $\mathcal{N}$ at θ are the connected components of input space with $\mathcal{B}_{\mathcal{N}}$ removed:

$$\text{linear regions}\,(\mathcal{N},\theta) \;=\; \text{connected components}\,(\mathbb{R}^{n_{\text{in}}} \setminus \mathcal{B}_{\mathcal{N}}(\theta))\,.$$

Linear regions have often been conflated with activation regions, but in some cases they are different. This can, for example, happen when an entire layer of the network is zeroed out by ReLUs, leading many distinct activation regions to coalesce into a single linear region. It can also occur for non-generic configurations of weights and biases where the linear functions computed in two neighboring activation regions happen to coincide. However, the number of activation regions is always at least as large as the number of linear regions.

Lemma 3 (More Activation Regions than Linear Regions)**.** *Let $\mathcal{N}$ be a ReLU net. For any parameter vector θ for $\mathcal{N}$, the number of linear regions in $\mathcal{N}$ at θ is always bounded above by the number of activation regions in $\mathcal{N}$ at θ. In fact, the closure of every linear region is the closure of the union of some number of activation regions.*

Lemma 3 is proved in Appendix B.3. We prove moreover in Appendix B.4 that generically, the gradient of $\nabla\mathcal{N}$ is different in the interior of most activation regions and hence that most activation regions lie in different linear regions. In particular, this means that the number of linear regions is generically very similar to the number of activation regions.

2.2 Activation Regions and Hyperplane Arrangements

Activation regions in depth 1 ReLU nets are given by hyperplane arrangements in $\mathbb{R}^{n_{\text{in}}}$ (see [33]). Indeed, if $\mathcal{N}$ is a ReLU net with one hidden layer, then the sets $H_z(\theta)$ from (2) are simply hyperplanes, giving the well-known observation that the activation regions in a depth 1 ReLU net are the connected components of $\mathbb{R}^{n_{\text{in}}}$ with the hyperplanes $H_z(\theta)$ removed. The study of regions induced by hyperplane arrangements in $\mathbb{R}^n$ is a classical subject in combinatorics [33]. A basic result is that for hyperplanes in general position (e.g. chosen at random), the total number of connected components coming from an arrangement of m hyperplanes in $\mathbb{R}^n$ is constant:

$$\#\text{connected components} \;=\; \sum_{i=0}^{n} \binom{m}{i} \;\simeq\; \begin{cases} \frac{m^n}{n!}, & m \gg n \\ 2^m, & m \leq n \end{cases}. \tag{4}$$

Hence, for random w_j, b_j drawn from any reasonable distributions the number of activation regions in a ReLU net with input dimension n_{in} and one hidden layer of size m is given by (4). The situation is more subtle for deeper networks. By Lemma 2, activation regions are connected components for an arrangement of "bent" hyperplanes $H_z(\theta)$ from (2), which are only locally described by hyperplanes. To understand their structure more carefully, fix a ReLU net $\mathcal{N}$ with d hidden layers and a vector θ of trainable parameters for $\mathcal{N}$. Write $\mathcal{N}_j$ for the network obtained by keeping only the first j layers of $\mathcal{N}$ and θ_j for the corresponding parameter vector. The following lemma makes precise the observation that the hyperplane $H_z(\theta)$ can only bend only when it meets a bent hyperplane $H_{\widehat{z}}(\theta)$ corresponding to some neuron $\widehat{z}$ in an earlier layer.

Lemma 4 ($H_z(\theta)$ as Bent Hyperplanes)**.** *Except on a set of $\theta \in \mathbb{R}^{\#\text{params}}$ of measure 0 with respect to Lebesgue measure, the sets $H_z(\theta_1)$ corresponding to neurons from the first hidden layer are hyperplanes in $\mathbb{R}^{n_{\text{in}}}$. Moreover, fix $2 \leq j \leq d$. Then, for each neuron z in layer j, the set $H_z(\theta_j)$ coincides with a single hyperplane in the interior of each activation region of $\mathcal{N}_{j-1}$.*

Lemma 4, which follows immediately from the proof of Lemma 7 in Appendix B.1, ensures that in a small ball near any point that does not belong to $\bigcup_z H_z(\theta)$, the collection of bent hyperplanes $H_z(\theta)$ look like an ordinary hyperplane arrangement. Globally, however, $H_z(\theta)$ can define many more regions than ordinary hyperplane arrangements. This reflects the fact that deep ReLU nets may have many more activation regions than shallow networks with the same number of neurons.

Despite their different extremal behaviors, we show in Theorem 5 that the average number of activation regions in a random ReLU net enjoys depth-independent upper bounds at initialization. We show experimentally that this holds throughout training as well (see §4). On the other hand, although we do not prove this here, we believe that the effect of depth can be seen through the *fluctuations* (e.g. the variance), rather than the mean, of the number of activation regions. For instance, for depth 1 ReLU nets, the variance is 0 since for a generic configuration of weights/biases, the number of activation regions is constant (see (4)). The variance is strictly positive, however, for deeper networks.

3 Main Result

3.1 Formal Statement

Theorem 5 gives upper bounds on the average number of activation regions per unit volume of input space for a feed-forward ReLU net with random weights/biases. Note that it applies even to highly correlated weight/bias distributions and hence holds throughout training. Also note that although we require no tied weights, there are no further constraints on the connectivity between adjacent layers.

Theorem 5 (Counting Activation Regions)**.** *Let $\mathcal{N}$ be a feed-forward ReLU network with no tied weights, input dimension n_{in}, output dimension 1, and random weights/biases satisfying:*

1. *The vector of weights is a continuous random variable (i.e. has a density).*
2. *The conditional distribution of any collection of biases given all weights and other biases is a continuous random variable (for identically zero biases, see Appendix D).*
3. *There exists $C_{\text{grad}} > 0$ so that for every neuron z and each $m \geq 1$, we have*

$$\sup_{x \in \mathbb{R}^{n_{\text{in}}}} \mathbb{E}\left[\|\nabla z(x)\|^m\right] \;\leq\; C_{\text{grad}}^m.$$

4. *There exists $C_{\text{bias}} > 0$ so that for any neurons $z_1, \ldots, z_k$, the conditional distribution of the biases $\rho_{b_{z_1},\ldots,b_{z_k}}$ of these neurons given all the other weights and biases in $\mathcal{N}$ satisfies*

$$\sup_{b_1,\ldots,b_k \in \mathbb{R}} \rho_{b_{z_1},\ldots,b_{z_k}}(b_1,\ldots,b_k) \;\leq\; C_{\text{bias}}^k.$$

Then, there exists $\delta_0, T > 0$ depending on $C_{\text{grad}}, C_{\text{bias}}$ with the following property. Suppose that $\delta > \delta_0$. Then, for all cubes $\mathcal{C}$ with side length δ, we have

$$\frac{\mathbb{E}\left[\#\text{non-empty activation regions of } \mathcal{N} \text{ in } \mathcal{C}\right]}{\text{vol}(\mathcal{C})} \;\leq\; \begin{cases} (T\#\text{neurons})^{n_{\text{in}}}/n_{\text{in}}! & \#\text{neurons} \geq n_{\text{in}} \\ 2^{\#\text{neurons}} & \#\text{neurons} \leq n_{\text{in}} \end{cases}. \tag{5}$$

Here, the average is with respect to the distribution of weights and biases in $\mathcal{N}$.

Remark 1. The heuristic of §3.3 suggests the average number of activation patterns in $\mathcal{N}$ over all of $\mathbb{R}^{n_{\text{in}}}$ is at most $(\#\text{neurons})^{n_{\text{in}}}/n_{\text{in}}!$, its value for depth 1 networks (see (4)). This is confirmed in our experiments (see Figures 3-6).

We state and prove a generalization of Theorem 5 in Appendix C.

Two constants C_{grad} and C_{bias} appear in Theorem 5. At initialization when all biases are independent, the constant C_{bias} can be taken simply to be the maximum of the density of the bias distribution. In sufficiently wide fully connected ReLU nets at initialization, C_{grad} is bounded. Indeed, by Theorem 1 (and Proposition 2) in [15], in a depth d fully connected ReLU net $\mathcal{N}$ with independent weights and biases whose marginals are symmetric around 0 and satisfy $\text{Var}[\text{weights}] = 2/\text{fan-in}$, the constant C_{grad} has an upper bound depending only the sum $\sum_{j=1}^{d} 1/n_j$ of the reciprocals of the hidden layer widths of $\mathcal{N}$. For example, if the layers of $\mathcal{N}$ have constant width n, then C_{grad} depends on the depth and width only via the aspect ratio d/n of $\mathcal{N}$, which is small for wide networks. Moreover, its has been shown that both the backward pass [14, 15] and the forward pass [16] in a fully connected ReLU net display significant numerical instability unless $\sum_{j=1}^{d} 1/n_j$ is small. Hence, at least in this simple setting, the constant C_{grad} is bounded as soon as the network is stable enough to begin training. However, there is no *a priori* reason that the constants C_{grad} and C_{bias} must remain bounded during training. Our numerical experiments (see §4 and Figures 4, 5, 2) indicate that even when training on corrupted or random data, however, the number of activation regions during and after training is typically comparable to its value at initialization. We believe, but have not verified all the details, that for very wide ReLU nets in which the neural tangent kernel governs the training dynamics [11, 21, 23, 35] the constant C_{grad} stays bounded throughout training.

Below we present two kinds of intuition motivating the Theorem. First, in §3.2 we derive the upper bound (5) via an intuitive geometric argument. Then in §3.3, we explain why, at initialization, we expect the upper bounds (5) to have matching, depth-independent, lower bounds (to leading order in the number of neurons). This suggests that the average *total* number of activation regions at initialization should be the same for any two ReLU nets with the same number of neurons (see (4) and Figure 3).

3.2 Geometric Intuition

We give an intuitive explanation for the upper bounds in Theorem 5, beginning with the simplest case of a ReLU net $\mathcal{N}$ with $n_{\text{in}} = 1$. Activation regions for $\mathcal{N}$ are intervals, and at an endpoint x of such an interval the pre-activation of some neuron z in $\mathcal{N}$ equals its bias: i.e. $z(x) = b_z$. Thus,

$$\#\text{activation regions of } \mathcal{N} \text{ in } [a,b] \;\leq\; 1 \;+\; \sum_{\text{neurons z}} \#\{x \in [a,b] \mid z(x) = b_z\}.$$

Geometrically, the number of solutions to $z(x) = b_z$ for inputs $x \in I$ is the number of times the horizontal line $y = b_z$ intersects the graph $y = z(x)$ over $x \in I$. A large number of intersections at a given bias b_z may only occur if the graph of $z(x)$ has many oscillations around that level. Hence, since b_z is random, the graph of $z(x)$ must oscillate many times over a large range on the y axis. This can happen only if the total variation $\int_{x\in I} |z'(x)|$ of $z(x)$ over I is large. Thus, if $|z'(x)|$ is typically of moderate size, we expect only $O(1)$ solutions to $z(x) = b_z$ per unit input length, suggesting

$$\mathbb{E}\left[\#\text{activation regions of } \mathcal{N} \text{ in } [a,b]\right] \;=\; O\left((b-a)\cdot \#\text{neurons}\right),$$

in accordance with Theorem 5 (cf. Theorems 1,3 in [17]). When $n_{\text{in}} > 1$, the preceding argument, shows that density of 1-dimensional regions per unit length along any 1-dimensional line segment in input space is bounded above by the number of neurons in $\mathcal{N}$. A unit-counting argument therefore suggests that the density of n_{in}-dimensional regions per unit n_{in}-dimensional volume is bounded above by $\#\text{neurons}$ raised to the input dimension, which is precisely the upper bound in Theorem 5 in the non-trivial regime where $\#\text{neurons} \gg n_{\text{in}}$.

3.3 Is Theorem 5 Sharp?

Theorem 5 shows that, on average, depth does not increase the *local density* of activation regions. We give here an intuitive explanation of why this should be the case in wide networks on any fixed subset of input space $\mathbb{R}^{n_{\text{in}}}$. Consider a ReLU net $\mathcal{N}$ with random weights/biases, and fix a layer index $\ell \geq 1$. Note that the map $x \mapsto x^{(\ell-1)}$ from inputs x to the post-activations of layer $\ell - 1$ is itself a ReLU net. Note also that in wide networks, the gradients $\nabla z(x)$ for different neurons in the same layer are only weakly correlated (cf. e.g. [22]). Hence, for the purpose of this heuristic,

we will assume that the bent hyperplanes $H_z(\theta)$ for neurons z in layer ℓ are independent. Consider an activation region $\mathcal{R}$ for $x^{(\ell-1)}(x)$. By definition, in the interior of $\mathcal{R}$, the gradient $\nabla z(x)$ for neurons z in layer ℓ are constant and hence the corresponding bent hyperplane from (2) inside $\mathcal{R}$ is the hyperplane $\{x \in \mathcal{R} \mid \langle \nabla z, x \rangle = b_z\}$. This in keeping with Lemma 4. The $2/\text{fan-in}$ weight normalization ensures that for each x

$$\mathbb{E}\left[\partial_{x_i}\partial_{x_j} z(x)\right] \;=\; 2 \cdot \delta_{i,j} \quad \Rightarrow \quad \text{Cov}[\nabla z(x)] \;=\; 2\,\text{Id}.$$

See, for example, equation (17) in [14]. Thus, the covariance matrix of the normal vectors ∇z of the hyperplanes $H_z(\theta) \cap \mathcal{R}$ for neurons in layer ℓ are *independent of* ℓ! This suggests that, per neuron, the average contribution to the number of activation regions is the same in every layer. In particular, deep and shallow ReLU nets with the same number of neurons should have the same average number of activation regions (see (4), Remark 1, and Figures 3-6).

4 Maximizing the Number of Activation Regions

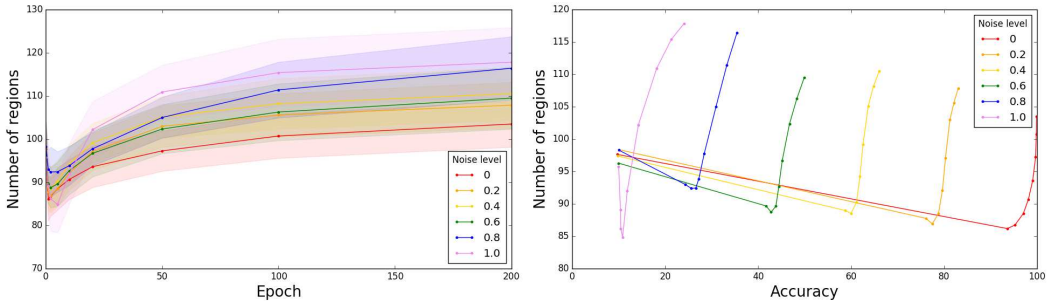

Figure 4: Depth 3, width 32 network trained on MNIST with varying levels of label corruption. Activation regions are counted along lines through input space (lines are selected to pass through both the origin and randomly selected MNIST examples), with counts averaged across 100 such lines. Theorem 5 and [17] predict the expected number of regions should be approximately the number of neurons (in this case, 96). Left: average number of regions plotted against epoch. Curves are averaged over 40 independent training runs, with standard deviations shown. Right: average number of regions plotted against average training accuracy. Throughout training the number of regions is well-predicted by our result. There are slightly, but not exponentially, more regions when memorizing more datapoints. See Appendix A for more details.

While we have seen in Figure 3 that the number of regions does not strongly increase during training on a simple task, such experiments leave open the possibility that the number of regions would go up markedly if the task were more complicated. Will the number of regions grow to achieve the theoretical upper bound (exponential in the depth) if the task is designed so that having more regions is advantageous? We now investigate this possibility. See Appendix A for experimental details.

4.1 Memorization

Memorization tasks on large datasets require learning highly oscillatory functions with large numbers of activation regions. Inspired by the work of Arpit et. al. in [1], we train on several tasks interpolating between memorization and generalization (see Figure 4) in a certain fraction of MNIST labels have been randomized. We find that the maximum number of activation regions learned does increase with the amount of noise to be memorized, but only slightly. In no case does the number of activation regions change by more than a small constant factor from its initial value. Next, we train a network to memorize binary labels for random 2D points (see Figure 5). Again, the number of activation regions after training increases slightly with increasing memorization, until the task becomes too hard for the network and training fails altogether. Varying the learning rate yields similar results (see Figure 6(a)), suggesting the small increase in activation regions is probably not a result of hyperparameter choice.

4.2 The Effect of Initialization

We explore here whether varying the scale of biases and weights at initialization affects the number of activation regions in a ReLU net. Note that scaling the biases changes the maximum density of the bias, and thus affects the upper bound on the density of activation regions given in Theorem 5 by increasing T_{bias}. Larger, more diffuse biases reduce the upper bound, while smaller, more tightly concentrated biases increase it. However, Theorem 5 counts only the *local* rather than *global* number of regions. The latter are independent of scaling the biases:

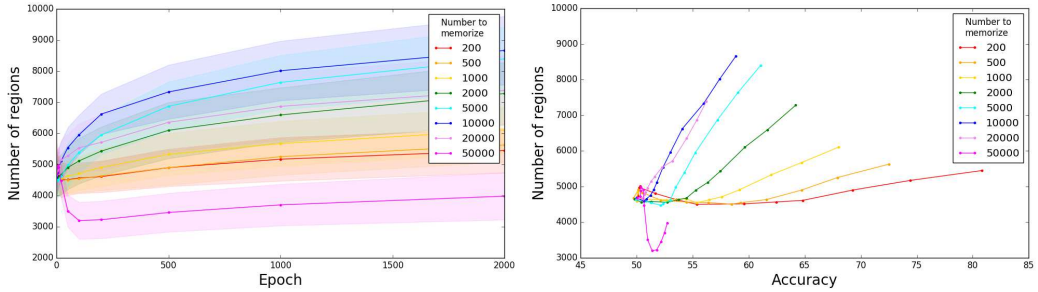

Figure 5: Depth 3, width 32 fully connected ReLU net trained for 2000 epochs to memorize random 2D points with binary labels. The number of regions predicted by Theorem 5 for such a network is $96^2/2! = 4608$. Left: number of regions plotted against epoch. Curves are averaged over 40 independent training runs, with standard deviations shown. Right: $\#\text{regions}$ plotted against training accuracy. The number of regions increased during training, and increased more for greater amounts of memorization. The exception was for the maximum amount of memorization, where the network essentially failed to learn, perhaps because of insufficient capacity. See Appendix A for more details.

Lemma 6. *Let $\mathcal{N}$ be a deep ReLU network, and for $c > 0$ let $\mathcal{N}_c^{bias}$ be the network obtained by multiplying all biases in $\mathcal{N}$ by c. Then, $\mathcal{N}(x) = \mathcal{N}_c^{bias}(cx)/c$. Rescaling all biases by the same constant therefore does not change the total number of activation regions.*

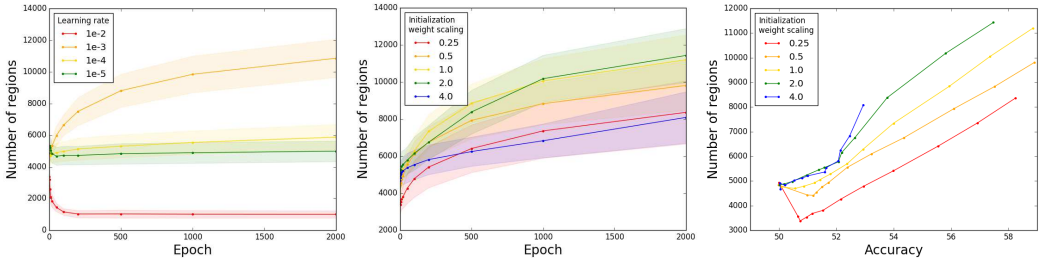

Figure 6: Depth 3, width 32 network trained to memorize 5000 random 2D points with independent binary labels, for various learning rates and weight scales at initialization. All networks start with ≈ 4608 regions, as predicted by Theorem 5. Left: None of the learning rates gives a number of regions larger than a small constant times the initial value. Learning rate 10^{-3}, which gives the maximum number of regions, is the learning rate in all other experiments, while 10^{-2} is too large and causes learning to fail. Center: Different weight scales at initialization do not strongly affect the number of regions. All weight scales are given relative to variance $2/\text{fan-in}$. Right: For a given accuracy, the number of regions learned grows with the weight scale at initialization. However, poor initialization impedes high accuracy. See Appendix A for details.

In the extreme case of biases initialized to zero, Theorem 5 does not apply. However, as we explain in Appendix D, zero biases only create fewer activation regions (see Figure 7). We now consider changing the scale of weights at initialization. In [29], it was suggested that initializing the weights of a network with greater variance should increase the number of activation regions. Likewise, the upper bound in Theorem 5 on the density of activation regions increases as gradient norms increase, and it has been shown that increased weight variance increases gradient norms [15]. However, this is again a property of the local, rather than global, number of regions.

Indeed, for a network $\mathcal{N}$ of depth d, write $\mathcal{N}_c^{\text{weight}}$ for the network obtained from $\mathcal{N}$ by multiplying all its weights by c, and let $\mathcal{N}_{1/c*}^{\text{bias}}$ be obtained from $\mathcal{N}$ by dividing the biases in the kth layer by c^k. A scaling argument shows that $\mathcal{N}_c^{\text{weight}}(x) = c^d \mathcal{N}_{1/c*}^{\text{bias}}(x)$. We therefore conclude that the activation regions of $\mathcal{N}_c^{\text{weight}}$ and $\mathcal{N}_{1/c*}^{\text{bias}}$ are the same. Thus, scaling the weights uniformly is equivalent to scaling the biases differently for every layer. We have seen from Lemma 6 that scaling the biases uniformly by any amount does not affect the global number of activation regions. We conjecture that scaling the weights uniformly should approximately preserve the global number of activation regions. We test this intuition empirically by attempting to memorize points randomly drawn from a 2D input space with arbitrary binary labels for various initializations (see Figure 6). We find that neither at

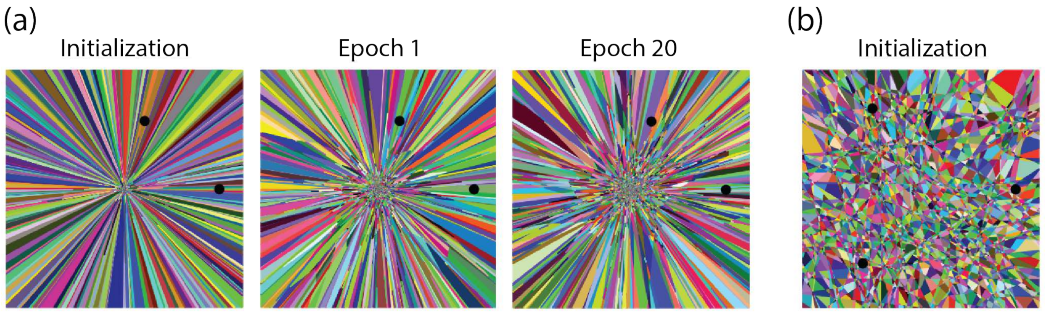

Figure 7: Activation regions within input space, for a network of depth 3 and width 64 training on MNIST. (a) Cross-section through the origin, shown at initialization, after one epoch, and after twenty epochs. The plane is chosen to pass through two sample points from MNIST, shown as black dots. (b) Cross-section not through the origin, shown at initialization. The plane is chosen to pass through three sample points from MNIST. For discussion of activation regions at zero bias, see Appendix D.

initialization nor during training is the number of activation regions strongly dependent on the weight scaling used for initialization.

5 Conclusion

We have presented theoretical and empirical evidence that the number of activation regions learned in practice by a ReLU network is far from the maximum possible and depends mainly on the number of neurons in the network, rather than its depth. This surprising result implies that, at least when network gradients and biases are well-behaved (see conditions 3,4 in the statement of Theorem 5), the partition of input space learned by a deep ReLU network is not significantly more complex than that of a shallow network with the same number of neurons. We found that this is true even after training on memorization-based tasks, in which we expect a large number of regions to be advantageous for fitting many randomly labeled inputs. Our results are stated for ReLU nets with no tied weights and biases (and arbitrary connectivity). We believe that analogous results and proofs hold for residual and convolutional networks but have not verified the technical details. While our results do not directly influence architecture selection for deep ReLU nets, they present the practical takeaway that the utility of depth may lie more in its effect on optimization than on expressivity.

References

[1] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *ICML*, 2017.

[2] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NeurIPS*, 2014.

[3] Andrew R Barron. Approximation and estimation bounds for artificial neural networks. *Machine learning*, 14(1):115–133, 1994.

[4] Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.

[5] Peter L Bartlett, Vitaly Maiorov, and Ron Meir. Almost linear VC dimension bounds for piecewise polynomial networks. In *NeurIPS*, 1999.

[6] Monica Bianchini and Franco Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE Transactions on Neural Networks and Learning Systems*, 25(8):1553–1565, 2014.

[7] Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. In *COLT*, pages 698–728, 2016.

[8] Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of ReLU networks via maximization of linear regions. In *AISTATS*, 2018.

[9] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[10] Giacomo De Palma, Bobak Toussi Kiani, and Seth Lloyd. Deep neural networks are biased towards simple functions. *Preprint arXiv:1812.10156*, 2018.

[11] Ethan Dyer and Guy Gur-Ari. Asymptotics of wide networks from Feynman diagrams. *Preprint arXiv:1909.11304*, 2019.

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

[13] Boris Hanin. Universal function approximation by deep neural nets with bounded width and ReLU activations. *Preprint arXiv:1708.02691*, 2017.

[14] Boris Hanin. Which neural net architectures give rise to exploding and vanishing gradients? In *NeurIPS*, 2018.

[15] Boris Hanin and Mihai Nica. Products of many large random matrices and gradients in deep neural networks. *Preprint arXiv:1812.05994*, 2018.

[16] Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. In *NeurIPS*, 2018.

[17] Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In *ICML*, 2019.

[18] Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension bounds for piecewise linear neural networks. In *COLT*, 2017.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *ICCV*, 2015.

[20] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[21] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018.

[22] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian processes. In *ICLR*, 2018.

[23] Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Preprint arXiv:1902.06720*, 2019.

[24] Henry W Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247, 2017.

[25] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *NeurIPS*, 2014.

[26] Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. In *ICLR*, 2018.

[27] Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta numerica*, 8:143–195, 1999.

[28] Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *NeurIPS*, 2016.

[29] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *ICML*, pages 2847–2854, 2017.

[30] David Rolnick and Max Tegmark. The power of deeper networks for expressing natural functions. In *ICLR*, 2018.

[31] Thiago Serra and Srikumar Ramalingam. Empirical bounds on linear regions of deep rectifier networks. *Preprint arXiv:1810.03370*, 2018.

[32] Thiago Serra, Christian Tjandraatmadja, and Srikumar Ramalingam. Bounding and counting linear regions of deep neural networks. In *ICML*, 2018.

[33] Richard P Stanley et al. An introduction to hyperplane arrangements. *Geometric combinatorics*, 13:389–496, 2004.

[34] Matus Telgarsky. Representation benefits of deep feedforward networks. *Preprint arXiv:1509.08101*, 2015.

[35] Greg Yang. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *Preprint arXiv:1902.04760*, 2019.

[36] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

A Experimental Design

We run several experiments that involve calculating the activation regions intersecting a 1D or 2D subset of input space. In order to compute these regions, we add neurons of the network one by one from the first to last hidden layer, observing how each neuron cuts existing regions. Determining whether a region is cut by a neuron involves identifying whether the corresponding linear function on that region has zeros within the region. This can be solved easily by identifying whether all vertices of the region have the same sign (region is not cut) or some two of the vertices have different signs (region is cut). Thus, our procedure is to maintain a list of regions and the linear functions defined on them, then for each newly added neuron identify on which regions its preactivation vanishes and replace these regions by the resulting split regions. Note that this procedure also works in three dimensions and higher, but becomes slower as the number of regions in higher dimensions grows like the number of neurons to the dimension, as shown in Theorem 5.

Unless otherwise specified, all experiments involving training a network were performed using an Adam optimizer, with learning rate 10^{-3} and batch size 128. Networks are, unless otherwise specified, initialized with i.i.d. normal weights with variance $2/\text{fan-in}$ (for justification of this initialization with ReLU networks, see [16, 19]) and i.i.d. normal biases with variance 10^{-6}.

B Proofs of Various Lemmas

B.1 Statement and Proof of Lemma 1 for General Piecewise Linear Activations

We begin by formulating Lemma 1 for a general continuous piecewise linear function $\varphi : \mathbb{R} \to \mathbb{R}$. For such a ϕ, there exists a non-negative integer $T \geq 0$, as well as

$$-\infty = \xi_0 \ < \ \xi_1 \ < \ \cdots \ < \ \xi_T, \qquad p_0, q_0, \ldots, p_T, q_T \in \mathbb{R}, \quad \text{with } p_i \neq p_{i+1} \text{ for } i = 0, \ldots, T-1$$

so that

$$\varphi(t) \ = \ p_i t + q_i, \quad t \in (\xi_i, \xi_{i+1}].$$

Definition 3. *Let $\mathcal{N}$ be a network with input dimension n_{in} and non-linearity ϕ. An activation pattern for $\mathcal{N}$ assigns to each neuron an element of the alphabet $\{0, \ldots, T\}$:*

$$\mathcal{A} \ := \ \{a_z, \ z \text{ a neuron in } \mathcal{N}\} \in \{0, \ldots, T\}^{\#\text{neurons}}.$$

Fix θ, a vector of trainable parameters in $\mathcal{N}$, and an activation pattern $\mathcal{A}$. The activation region corresponding to $\mathcal{A}, \theta$ is

$$\mathcal{R}(\mathcal{A}; \theta) := \{x \in \mathbb{R}^{n_{\text{in}}} \quad | \quad z(x) - b_z \in (\xi_{a_z}, \xi_{a_z+1}), \quad z \text{ a neuron in } \mathcal{N}\},$$

where the pre-activation of a neuron z is $z(x; \theta)$, its bias is b_z, and its post-activation is therefore $\varphi(z(x; \theta) - b_z)$. Finally, the activation regions of $\mathcal{N}$ at θ is the collection of all non-empty activation regions $\mathcal{R}(\mathcal{A}, \theta)$.

We will prove the following generalization of Lemma 8.

Lemma 7 (Activation Regions are Convex)**.** *Let $\mathcal{N}$ be a network with non-linearity φ, and let*

$$\mathcal{A} = \{a_{j,z}, \ z \text{ a neuron in } \mathcal{N}\} \in \{0, \ldots, T\}^{\#\text{neurons}}$$

be any activation pattern. Then, for any vector θ of trainable parameters for $\mathcal{N}$, the region $\mathcal{R}(\mathcal{A}; \theta)$ is convex.

Proof. Write d for the depth of $\mathcal{N}$ and note that, by definition,

$$\mathcal{R}\left(\mathcal{A}; \theta\right) \ = \ \bigcap_{\ell=1}^{d} \bigcap_{\substack{\text{neurons} \\ \text{in layer } \ell}}^{z} \mathcal{R}_z(\mathcal{A}; \theta), \quad \mathcal{R}_z(\mathcal{A}; \theta) \ := \ \{x \in \mathbb{R}^{n_{\text{in}}} \mid z(x) - b_z \in (-\xi_{a_z}, \xi_{a_z+1})\}.$$

We will show that $\mathcal{R}(\mathcal{A}, \theta)$ is convex by induction on d. For the base case, note that given $s < t \in [-\infty, \infty]$, for every $w \in \mathbb{R}^{n_{\text{in}}}$, $b \in \mathbb{R}$

$$\{x \in \mathbb{R}^{n_{\text{in}}} \mid w \cdot x - b \in (s, t)\}$$

is convex. Hence, the intersection of any number of such sets is convex as well. Note that when $d = 1$, $R_z(\mathcal{A}; \theta)$ is of this form every z proves the base case. For the inductive case, suppose we

have shown the claim for some $d \geq 1$. For inputs x in the $\bigcap_{\ell=1}^{d} \bigcap_{\substack{\text{neurons } z \\ \text{in layer } \ell}} \mathcal{R}_z(\mathcal{A};\theta)$, there exists, for every neuron z in layer $d+1$ a vector $w_z \in \mathbb{R}^{n_{\text{in}}}$ and a scalar $b_z \in \mathbb{R}$ so that

$$z(x) = w_z \cdot x - b.$$

Therefore,

$$\bigcap_{\ell=1}^{d} \bigcap_{\substack{\text{neurons } z \\ \text{in layer } \ell}} \mathcal{R}_z(\mathcal{A};\theta) \,\cap\, R_z(\mathcal{A};\theta) \;=\; \{x \in \mathcal{R}_d \mid w_z \cdot x - b \in (\xi_{a_z}, \xi_{a_z+1})\}$$

is the intersection of two convex sets and is therefore convex. Taking the intersection over all z in layer $d+1$ completes the proof. □

B.2 Proof of Lemma 2

We claim the following general fact. Suppose X is a topological space and $f_j : X \to \mathbb{R}$ are continuous, $j = 1, \ldots, J$. Then, on every connected component of $X \setminus \bigcup_j f_j^{-1}(0)$, the sign of f is constant. Indeed, consider a connected component X_α. Since f_j are never 0 on X_α by construction, we have $f_j(X_\alpha) \subseteq (-\infty, 0) \cup (0, \infty)$. But the image under a continuous map of a connected set is connected. Hence, for each j $f_j(X_\alpha) \subseteq (-\infty, 0)$ or $f_j(X_\alpha) \subseteq (0, \infty)$, and the claim follows.

Turning to Lemma 2, let $\mathcal{N}$ be a ReLU net with input dimension n_{in}, and fix a vector of trainable parameters θ for $\mathcal{N}$. The claim above shows that on every connected component of $\mathbb{R}^{n_{\text{in}}} \setminus \bigcup_{\text{neurons } z} H_z(\theta)$ the functions $z(x) - b_z$ have a definite sign for all neurons z. Thus, every connected component is contained in some activation region $\mathcal{R}(\mathcal{A};\theta)$. Finally, by construction,

$$\mathcal{R}\left(\mathcal{A};\theta\right) \;\subset\; \mathbb{R}^{n_{\text{in}}} \setminus \bigcup_{\text{neurons } z} H_z(\theta)$$

and, by Lemma 1, $\mathcal{R}\left(\mathcal{A};\theta\right)$ is convex and hence connected. Therefore it is equal to the connected component we started with. □

B.3 Proof of Lemma 3

Let $\mathcal{N}$ be a ReLU net, and fix a vector θ of its trainable parameters. Let us first check that

$$\#\text{linear regions}\left(\mathcal{N},\theta\right) \;\leq\; \#\text{ non-empty activation regions } \mathcal{R}(\mathcal{A};\theta). \tag{6}$$

We will use the following simple fact: if X, Y are subsets of a topological space T, then $X \subset Y$ implies that every connected component of X is the subset of some connected component of Y. Indeed, if X_α is a connected component of X, then it is a connected subjset of Y and hence is contained in a unique connected component of Y.

This fact shows that the cardinality of the set of connected components of Y is bounded above by the cardinality of the set of connected components for X. Using Lemma 2, the inequality (6) therefore reduces to showing that

$$\mathcal{B}_{\mathcal{N}}(\theta) \;\subseteq \bigcup_{\text{neurons } z} H_z(\theta), \quad H_z(\theta) = \{x \in \mathbb{R}^{n_{\text{in}}} \mid z(x) = b_z\}. \tag{7}$$

Fix any x in the complement of the right hand side. By definition, we have

$$|z(x) - b_z| > 0, \qquad \forall \text{ neurons } z.$$

The functions $z(x)$ are continuous. Hence, in a small neighborhood of x, the estimate in the previous line holds in a small neighborhood of x. Thus, in an open neighborhood of x, the collection of neurons that are on and off are constant. The inclusion (7) now follows by observing that if for all y in an open neighborhood U of $x \in \mathbb{R}^{n_{\text{in}}}$, the sets

$$\mathcal{A}_{y,\pm}(\theta) \;:=\; \{\text{neurons } z \mid \pm\left(z(y;\theta) - b_z\right) > 0\}$$

are constant and

$$\mathcal{A}_{y,0}(\theta) \;:=\; \{\text{neurons } z \mid z(y;\theta) = b_z\} \;=\; \emptyset,$$

then $\mathcal{N}$ restricted to U is given by a single linear function and hence has a continuous gradient $\nabla\mathcal{N}\left(\cdot\,;\theta\right)$ on U.

It remains to check that the closure of every linear region of $\mathcal{N}$ at θ is the closure of the union of some activation regions of $\mathcal{N}$ at θ. Note that, except on a set of $\theta \in \mathbb{R}^{\text{params}}$ with Lebesgue measure 0, both $\bigcup_z H_z(\theta)$ and $\mathcal{B}_{\mathcal{N}}(\theta)$ are co-dimension 1 piecewise linear submanifolds of $\mathbb{R}^{n_{\text{in}}}$ with finitely many pieces. Thus, $\mathbb{R}^{n_{\text{in}}} \setminus \bigcup_z H_z(\theta)$ is open and dense in $\mathbb{R}^{n_{\text{in}}} \setminus \mathcal{B}_{\mathcal{N}}(\theta)$. And now we appeal to a general topological fact: if $X \subset Y$ are subsets of a topological space T and X is open and dense in Y, then the closure in T of every connected component of Y is the closure of the union of some connected components of X. Indeed, consider a connected component Y_β of Y. Then $X \cap Y_\beta$ is open and dense in Y_β. On the other hand, $X \cap Y_\beta$ is the union of connected components of X_α of X. Thus, the closure of this union is the closure of $X \cap Y_\beta$, namely the closure of Y_β. □

B.4 Distinguishability of Activation Regions

In addition to being convex, activation regions for a ReLU net $\mathcal{N}$ generically correspond to different linear functions:

Lemma 8 (Activation Regions are Distinguishable)**.** *Let $\mathcal{N}$ be a ReLU net, and let*

$$\mathcal{A}_j = \{a_{j,z},\ z \text{ a neuron in } \mathcal{N}\} \in \{-1,1\}^{\#\text{neurons}} \quad j = 1,2$$

be two activation patterns with $\mathcal{R}(\mathcal{A}_j;\theta) \neq \emptyset$. Suppose also that, for every layer and each $j = 1,2$ there exists a neuron z_j with $a_{j,z_j} = 1$. Then, except on a measure zero set of θ with respect to Lebesgue measure in the parameter space $\mathbb{R}^{\#\text{params}}$, the gradient $\nabla\mathcal{N}(x;\theta)$ is different for x in the interiors of $\mathcal{R}(\mathcal{A}_j;\theta)$.

Proof. Fix an activation pattern $\mathcal{A}$ for a depth d ReLU network $\mathcal{N}$ with $\mathcal{R}(\mathcal{A};\theta) \neq \emptyset$. Fix $x \in \mathcal{R}(\mathcal{A};\theta)$. We will use the following well-known formula:

$$\nabla\mathcal{N}(x;\theta) \ = \ \sum_{\text{paths } \gamma} \mathbf{1}_{\{\gamma \text{ open at } x\}} \prod_{i=1}^{d} w_\gamma^{(j)},$$

where the sum is over all paths in the computational graph of $\mathcal{N}$, a path is open at x if every neuron z in γ satisfies $z(x) - b_z > 0$, and $w_\gamma^{(j)}$ is the weight on the edge of γ between layers $j-1$ and j. If there exist two different, non-empty activation regions corresponding to activation patterns $\mathcal{A}$ for which there is at least one open path through the network on which $\nabla N(\cdot\,;\theta)$ has the same value on , then there exists $a_\gamma \in \{\pm 1\}$ and a non-empty collection Γ of paths so that

$$\sum_{\gamma\in\Gamma} a_\gamma \prod_{i=1}^{d} w_\gamma^{(j)} \ = \ 0. \tag{8}$$

The zero set of any such polynomial (since $\Gamma \neq \emptyset$) is a co-dimension 1 variety in $\mathbb{R}^{\#\text{weights}}$. Since there are only finitely many (in fact $2^{\#\text{paths}}$) such polynomials, the set of θ for which (8) can occur has measure 0 with respect to the Lebesgue measure on $\mathbb{R}^{\#\text{weights}}$, as claimed. □

C Statement and Proof of a Generalization of Theorem 5

We begin by stating a generalization of Theorem 5 to what we term partial activation regions. Let us write

$$\text{sgn}(t) := \begin{cases} 1, & t > 0 \\ 0, & t = 0 \\ -1, & t < 0 \end{cases}.$$

Definition 4 (Partial Activation Regions)**.** *Let $\mathcal{N}$ be a ReLU net with input dimension n_{in}. Fix a non-negative integer k. A k-partial activation pattern for $\mathcal{N}$ is an assignment to each neuron of a sign of $-1, 0, 1$, with exactly k neurons being assigned a 0:*

$$\mathcal{A} \ := \ \{a_z,\ z \text{ a neuron in } \mathcal{N}\} \in \{-1,0,1\}^{\#\text{neurons}}, \quad \#\{z \mid a_z = 0\} = k.$$

Fix θ, a vector of trainable parameters in $\mathcal{N}$, and a k-partial activation pattern $\mathcal{A}$. The k-partial activation region corresponding to $\mathcal{A}, \theta$ is

$$\mathcal{R}(\mathcal{A};\theta) := \{x \in \mathbb{R}^{n_{\text{in}}} \quad | \quad \text{sgn}\left(z(x;\theta) - b_z\right) = a_z, \quad z \text{ a neuron in } \mathcal{N}\},$$

where the pre-activation of a neuron z is $z(x;\theta)$, its bias is b_z, its post-activation is therefore $\max\{0, z(x;\theta) - b_z\}$. Finally, the activation regions of $\mathcal{N}$ at θ is the collection of all non-empty activation regions $\mathcal{R}(\mathcal{A},\theta)$.

The same argument as the proof of Lemma 7 yields the following result:

Lemma 9. *Let $\mathcal{N}$ be a ReLU net. Fix a non-negative integer r and let $\mathcal{A}$ be a r-partial activation pattern for $\mathcal{N}$. For every vector θ of trainable parameters for $\mathcal{N}$, the r-partial activation region $\mathcal{R}(\mathcal{A};\theta)$ is convex.*

We will prove the following generalization of Theorem 5.

Theorem 10. *Let $\mathcal{N}$ be a feed-forward ReLU network with no tied weights, input dimension n_{in} and output dimension 1. Suppose that the weights and biases $\mathcal{N}$ is random and satisfies:*

1. *The distribution of all weights has a density with respect to Lebesgue measure on $\mathbb{R}^{\#\text{weights}}$.*
2. *Every collection of biases has a density with respect to Lebesgue measure conditional on the values of all weights and other biases (for identically zero biases, see Appendix D).*
3. *There exists $C_{\text{grad}} > 0$ so that for every neuron z and each $m \geq 1$, we have*
$$\sup_{x \in \mathbb{R}^{n_{\text{in}}}} \mathbb{E}\left[\|\nabla z(x)\|^m\right] \;\leq\; C_{\text{grad}}^m.$$
4. *There exists $C_{\text{bias}} > 0$ so that for any neurons $z_1, \ldots, z_k$, the conditional distribution of the biases $\rho_{b_{z_1},\ldots,b_{z_k}}$ of these neurons given all the other weights and biases in $\mathcal{N}$ satisfies*
$$\sup_{b_1,\ldots,b_k \in \mathbb{R}} \rho_{b_{z_1},\ldots,b_{z_k}}(b_1,\ldots,b_k) \;\leq\; C_{\text{bias}}^k.$$

Fix $r \in \{0,\ldots,n_{\text{in}}\}$. Then, there exists $\delta_0, T > 0$ depending on $C_{\text{grad}}, C_{\text{bias}}$ with the following property. Suppose that $\delta > \delta_0$. Then, for all cubes $\mathcal{C}$ with side length δ, we have

$$\frac{\mathbb{E}\left[\#\text{r} - \text{partial activation regions of } \mathcal{N} \text{ in } \mathcal{C}\right]}{\text{vol}(\mathcal{C})} \;\leq\; \begin{cases} \frac{(T\#\text{neurons})^{n_{\text{in}}}}{2^r n_{\text{in}}!}, & \#\text{neurons} \geq n_{\text{in}} \\ 2^{\#\text{neurons}}, & \#\text{neurons} \leq n_{\text{in}} \end{cases}. \quad (9)$$

Here, the average is with respect to the distribution of weights and biases in $\mathcal{N}$.

Proof. Fix a ReLU net $\mathcal{N}$ with input dimension n_{in} and a non-negative integer r. Since the number of distinct r-partial activation patterns in $\mathcal{N}$ is at most $\binom{\#\text{neurons}}{r} 2^{\#\text{neurons}-r}$, Theorem 5 only requires proof when $\#\text{neurons} \geq n_{\text{in}}$. For this, for any vector of trainable parameters θ define

$$X_{\mathcal{N},k}(\theta) := \bigcup_{\substack{\text{collections } I \text{ of } k \\ \text{distinct neurons in } \mathcal{N}}} \left[\bigcap_{z \in I} H_z(\theta) \cap \bigcap_{z \notin I} H_z(\theta)^c\right].$$

In words, $X_{\mathcal{N},k}$ is the collection of inputs $x \in \mathbb{R}^{n_{\text{in}}}$ for which there exist exactly k neurons z so that x solves $z(x) = b_z$. We record for later use the following fact.

Lemma 11. *With probability 1 over the space of θ's, for any $x \in X_{\mathcal{N},k}$ there exists $\varepsilon > 0$ (depending on x and θ) so that the set $X_{\mathcal{N},k}$ intersected with the ε ball $B_\varepsilon(x)$ coincides with a hyperplane of dimension $n_{\text{in}} - k$.*

Proof. We begin with the following observation. Let $\mathcal{N}$ be a ReLU net, fix a vector θ of trainable parameters, and consider a neuron z in $\mathcal{N}$. The function $x \mapsto z(x;\theta)$ is continuous and piecewise linear with a finite number of regions $\mathcal{R}$ on which $z(x,\theta)$ is some fixed linear function. On each such $\mathcal{R}$, the gradient ∇z is constant. If that constant is non-zero, then $H_z(\theta) \cap \mathcal{R}$ is either empty if b_z does not belong to the range of z on $\mathcal{R}$ or is a co-dimension 1 hyperplane if it does. In contrast, if $\nabla z = 0$ on $\mathcal{R}$, then z is constant and $H_z(\theta) \cap \mathcal{R}$ is empty unless b_z is precisely equal to the value of z on $\mathcal{R}$. Thus, given any choice of weights in $\mathcal{N}$ and for all but a finite number of biases, the set $H_z(\theta)$ coincides with co-dimension one hyperplane in each linear region $\mathcal{R}$ for the function $z(\cdot;\theta)$ that it intersects.

Now let us fix all the weights (but not biases) in $\mathcal{N}$ and a collection $z_1, \ldots, z_k$ of k distinct neurons in $\mathcal{N}$ arranged so that $\ell(z_1) \leq \cdots \leq \ell(z_k)$ where $\ell(z)$ denotes the layer of $\mathcal{N}$ to which z belongs. Suppose x belongs to $H_{z_j}(\theta)$ for $j = 1,\ldots,k$ but not to $H_z(\theta)$ for any neuron z not in $\{z_1,\ldots,z_k\}$.

By construction, the function $z_1(\cdot\,;\theta)$ is linear in a neighborhood of x. Therefore, by the observation above, near x, for all but a finite collection of choices of bias b_{z_1} for z_1, $H_{z_1}(\theta)$ coincides with a co-dimension 1 hyperplane. Let us define a new network obtained by restricting $\mathcal{N}$ to this hyperplane (and keeping all the weights from $\mathcal{N}$ fixed). Repeating the preceding argument applied to the neuron z_2 in this new network, we find again that, except for a finite number of values for the bias b_{z_2}, near x the set $H_{z_2}(\theta) \cap H_{z_1}(\theta)$ is also a co-dimension 1 hyperplane inside H_{z_1}. Proceeding in this way shows that for any fixed collection of weights for $\mathcal{N}$, there are only finitely many choices of biases for which the conclusion the present Lemma fails to hold. Thus, in particular, the conclusion holds on a set of probability 1 with respect to any distribution on $\mathbb{R}^{\#\text{params}}$ that has a density relative to Lebesgue measure. □

Repeating the proof Theorem 6 in [17] with the sets $\widetilde{S}_z$ replaced by S_z (which only makes the proof simpler since Proposition 9 in [17] is not needed) shows that, under the assumptions of Theorem 5, there exists $T > 0$ so that for any bounded $S \subset \mathbb{R}^{n_{\text{in}}}$

$$\frac{\mathbb{E}\left[\text{vol}_{n_{\text{in}}-k}(X_{\mathcal{N},k} \cap S)\right]}{\text{vol}_{n_{\text{in}}}(S)} \;\leq\; T^k \binom{\#\text{neurons}}{k}. \tag{10}$$

We will use the volume bounds in (10) to prove Theorem 5 by an essentially combinatorial argument, which we now explain. Fix a closed cube $\mathcal{C} \subseteq \mathbb{R}^{n_{\text{in}}}$ with sidelength $\delta > 0$ and define

$$\mathcal{C}_k := \text{dimension } k \text{ skeleton of } \mathcal{C}, \qquad k = 0, \ldots, n_{\text{in}}.$$

So for example, $\mathcal{C}_0$ is the $2^{n_{\text{in}}}$ vertices of $\mathcal{C}$, $\mathcal{C}_{n_{\text{in}}}$ is $\mathcal{C}$ itself, and $\mathcal{C}_{n_{\text{in}}-1}$ is the set of $2n_{\text{in}}$ co-dimension 1 faces of $\mathcal{C}$. In general, $\mathcal{C}_k$ consists of $\binom{n_{\text{in}}}{k}2^{n_{\text{in}}-k}$ linear pieces with dimension k, each with volume δ^k. Hence,

$$\text{vol}_k\left(\mathcal{C}_k\right) \;=\; \binom{n_{\text{in}}}{k} 2^{n_{\text{in}}-k}\delta^k \tag{11}$$

For any vector θ of trainable parameters for $\mathcal{N}$ define

$$\mathcal{V}_k(\theta) \;:=\; X_{\mathcal{N},k}(\theta) \cap \mathcal{C}_k.$$

The collections $\mathcal{V}_k(\theta)$ are useful for the following reason.

Lemma 12. *With probability* 1 *with respect to* θ*, for every* k*, the set* $\mathcal{V}_k(\theta)$ *has dimension* 0 *(i.e. is a collection of discrete points) and*

$$\#\{\text{r}-\text{partial activation regions } \mathcal{R}\left(\mathcal{A};\theta\right) \text{ with } \mathcal{R}\left(\mathcal{A};\theta\right) \cap \mathcal{C} \neq \emptyset\} \;\leq\; \sum_{k=r}^{n_{\text{in}}} \binom{k}{r} 2^{k-r} \#\mathcal{V}_k, \tag{12}$$

where $\#\mathcal{V}_k$ *is the number of points in* $\mathcal{V}_k$*.*

Proof. The statement that generically $\mathcal{V}_k(\theta)$ has dimension 0 follows since, by construction, $\mathcal{C}_k$ has dimension k and, with probability 1, $X_{\mathcal{N},k}(\theta)$ coincides locally with a co-dimension k hyperplane (see Lemma 11) in general position. To prove (12) note that any non-empty, bounded, convex polytope has at least one vertex on its boundary. Thus, with probability 1,

$$\mathcal{R}\left(\mathcal{A};\theta\right) \cap \mathcal{C} \neq \emptyset \quad \Rightarrow \quad \exists\, k = 0, \ldots, n_{\text{in}} \quad \text{s.t.} \quad \partial\mathcal{R}\left(\mathcal{A};\theta\right) \cap V_k \neq \emptyset,$$

where $\partial\mathcal{R}\left(\mathcal{A};\theta\right)$ is the boundary of the r-partial activation region $\mathcal{R}\left(\mathcal{A};\theta\right)$. Indeed, if an r-partial activation region $\mathcal{R}\left(\mathcal{A};\theta\right)$ intersects $\mathcal{C}$, then $\mathcal{R}\left(\mathcal{A};\theta\right) \cap \mathcal{C}$ is a non-empty, bounded, convex polytope. It must therefore have a vertex on its boundary. This vertex is either in the interior of $\mathcal{C}$ (and hence belongs to $\mathcal{V}_{n_{\text{in}}}$) or is the intersection of some co-dimension k part of the boundary of $\mathcal{R}\left(\mathcal{A};\theta\right)$, which belongs with probability 1 to $X_{\mathcal{N},r+k}$, with some $r+k-$dimensional face of the boundary of $\mathcal{C}$ (and hence belongs to $\mathcal{V}_{k+r}$). Thus, with probability 1,

$$\#\{\text{r-partial activation regions } \mathcal{R}\left(\mathcal{A};\theta\right) \text{ with } \mathcal{R}\left(\mathcal{A};\theta\right) \cap \mathcal{C} \neq \emptyset\} \;\leq\; \sum_{k=r}^{n_{\text{in}}} T_k \#\mathcal{V}_k,$$

where T_k is the maximum over all $v \in \mathcal{V}_k$ of the number of r-partial activation regions whose boundary contains v. To complete the proof, it remains to check that, with probability 1 over θ,

$$\forall v \in \mathcal{V}_k \qquad \#\{\text{r}-\text{partial regions } \mathcal{R}\left(\mathcal{A};\theta\right) \mid v \in \partial\mathcal{R}\left(\mathcal{A};\theta\right)\} \;\leq\; \binom{k}{r} 2^{k-r}.$$

To see this, note that, by definition of $X_{\mathcal{N},k}$, in a sufficiently small neighborhood U of any $v \in \mathcal{V}_k$, all but k neurons have pre-activations satisfying either $z(x) > b_z$ or $z(x) < b_z$ for all $x \in U$. Thus, there are at most $\binom{k}{r}2^{k-r}$ different r-partial activation patterns $\mathcal{A}$ whose r-partial activation region $\mathcal{R}(\mathcal{A};\theta)$ can intersect U. □

To proceed with the proof of Theorem 5, we need the following observation.

Lemma 13. *Fix $k = 0, \dots, n_{\text{in}}$. For all but a measure 0 set of vectors θ of trainable parameters for $\mathcal{N}$ there exists $\varepsilon > 0$ (depending on θ) so that the balls $B_\varepsilon(v)$ of radius ε centered at $v \in \mathcal{V}_k$ are disjoint and*

$$\text{vol}_{n_{\text{in}}-k}\left(X_{\mathcal{N},k} \cap B_\varepsilon(v)\right) \;=\; \varepsilon^{n_{\text{in}}-k}\omega_{n_{\text{in}}-k},$$

where ω_k is the volume of unit ball in $\mathbb{R}^k$.

Proof. With probability 1 over θ, each $\mathcal{V}_k$, by Lemma 12, is a discrete collection of points. Hence, we may choose $\varepsilon > 0$ sufficiently small so that the balls $B_\varepsilon(v)$ are disjoint. Moreover, by Lemma 11, in a sufficiently small neighborhood of $v \in \mathcal{V}_k$, the set $X_{\mathcal{N},k}$ coincides with a $(n_{\text{in}}-k)$-dimensional hyperplane passing through v. The $(n_{\text{in}}-k)-$dimensional volume of this hyperplane in $B_\varepsilon(v)$ is the volume of an $(n_{\text{in}}-k)-$dimensional ball of radius ε, which equals $\varepsilon^{n_{\text{in}}-k}\omega_{n_{\text{in}}-k}$, completing the proof. □

For each $k = 0, \dots, n_{\text{in}}$, denote by

$$\mathcal{C}_{k,\varepsilon} \;:=\; \{x \in \mathbb{R}^{n_{\text{in}}} \mid \text{dist}(x, \mathcal{C}_k) \le \varepsilon\}$$

the ε- thickening of $\mathcal{C}_k$. Then Lemma 13 implies that for all but a measure 0 set of $\theta \in \mathbb{R}^{\#\text{params}}$, there exists $\varepsilon > 0$ so that

$$\frac{\text{vol}_{n_{\text{in}}-k}\left(X_{\mathcal{N},k} \cap \mathcal{C}_{k,\varepsilon}\right)}{\varepsilon^{n_{\text{in}}-k}\omega_{n_{\text{in}}-k}} \;\ge\; \#\mathcal{V}_k.$$

Indeed, for any θ in a full measure set, Lemma 13 guarantees the existence of an $\varepsilon > 0$ so that the ε balls $B_\varepsilon(v)$ are contained in $\mathcal{C}_{k,\varepsilon}$ and are disjoint. Hence,

$$\text{vol}_{n_{\text{in}}-k}\left(X_{\mathcal{N},k} \cap \mathcal{C}_{k,\varepsilon}\right) \;\ge\; \sum_{v \in \mathcal{V}_k} \varepsilon^{n_{\text{in}}-k}\omega_{n_{\text{in}}-k} \;=\; \#\mathcal{V}_k \cdot \varepsilon^{n_{\text{in}}-k}\omega_{n_{\text{in}}-k}.$$

Note that

$$\lim_{\varepsilon \to 0} \frac{\text{vol}_{n_{\text{in}}}\left(\mathcal{C}_{k,\varepsilon}\right)}{\varepsilon^{n_{\text{in}}-k}\omega_{n_{\text{in}}-k}} \;=\; \text{vol}_k\left(\mathcal{C}_k\right).$$

Thus, taking $\varepsilon \to 0$, we have

$$\mathbb{E}\left[\#\mathcal{V}_k\right] \;\le\; \mathbb{E}\left[\lim_{\varepsilon \to 0} \frac{\text{vol}_k\left(X_{\mathcal{N},k} \cap \mathcal{C}_{k,\varepsilon}\right)}{\varepsilon^{n_{\text{in}}-k}\omega_{n_{\text{in}}-k}}\right].$$

By Fatou's Lemma and (10) there exists $T > 0$ such that

$$\mathbb{E}\left[\#\mathcal{V}_k\right] \;\le\; \lim_{\varepsilon \to 0} \mathbb{E}\left[\frac{\text{vol}_{n_{\text{in}}-k}\left(X_{\mathcal{N},k} \cap \mathcal{C}_{k,\varepsilon}\right)}{\varepsilon^{n_{\text{in}}-k}\omega_{n_{\text{in}}-k}}\right] \;\le\; T^k \binom{\#\text{neurons}}{k} \text{vol}_k(\mathcal{C}_k).$$

Combining this with (11) and (12), we find

$$\mathbb{E}\left[\#\{\text{r} - \text{partial activation regions } \mathcal{R}(\mathcal{A};\theta) \text{ with } \mathcal{R}(\mathcal{A};\theta) \cap \mathcal{C} \neq \emptyset\}\right]$$

is bounded above by

$$\begin{aligned}\sum_{k=r}^{n_{\text{in}}} \binom{n_{\text{in}}}{k}\binom{\#\text{neurons}}{k}\binom{k}{r}2^{k-r}2^{n_{\text{in}}-k}(\delta T)^k \;&\le\; 2^{2n_{\text{in}}-r}\sum_{k=r}^{n_{\text{in}}} \binom{n_{\text{in}}}{k}\binom{\#\text{neurons}}{k}(\delta T)^k \\ &\le\; 2^{-r}(4T\delta)^{n_{\text{in}}}\sum_{k=r}^{n_{\text{in}}} \binom{n_{\text{in}}}{k}^2 \frac{\binom{\#\text{neurons}}{k}}{\binom{n_{\text{in}}}{k}},\end{aligned}$$

where in the last line we've assumed $\delta > 1/T$ and in the first inequality we used that

$$\binom{k}{r} \leq \sum_{r=0}^{k} \binom{k}{r} = 2^k \leq 2^{n_{\text{in}}}.$$

Observe that

$$\frac{\binom{\#\text{neurons}}{k}}{\binom{n_{\text{in}}}{k}} = \frac{(\#\text{neurons})! \cdot (n-k)!}{n_{\text{in}}! \cdot (\#\text{neurons} - k)!} \leq \frac{(\#\text{neurons})^{n_{\text{in}}}}{n_{\text{in}}!} \cdot \frac{(n_{\text{in}} - k)!}{(\#\text{neurons})^{n_{\text{in}}-k}} \leq \frac{(\#\text{neurons})^{n_{\text{in}}}}{n_{\text{in}}!}.$$

Hence, using that

$$\sum_{k=0}^{n_{\text{in}}} \binom{n_{\text{in}}}{k}^2 = \binom{2n_{\text{in}}}{n_{\text{in}}} \leq 4^{n_{\text{in}}},$$

$\mathbb{E}\left[\#\{\text{activation regions } \mathcal{R}(\mathcal{A};\theta) \text{ with } \mathcal{R}(\mathcal{A};\theta) \cap \mathcal{C} \neq \emptyset\}\right]$ is bounded above by

$$\frac{(T\#\text{neurons})^{n_{\text{in}}}}{2^r n_{\text{in}}!} \text{vol}(\mathcal{C}).$$

□

D Zero Bias

The reader may notice that, as stated, Theorem 5 does not apply for networks with zero biases, as is commonly the case at initialization. While zero biases clearly do not persist during normal training past initialization, it is interesting to consider how many activation regions occur in this case. No finite value of constant C_{bias} in Theorem 5 satisfies Condition 4; does this mean that the number of activation regions goes to infinity? The answer is no:

Proposition 14. *Suppose that $\mathcal{N}$ is a deep ReLU net with any bias distribution, and let $\mathcal{N}_0^{\text{bias}}$ be the same network with all biases set to 0. Then, the total number of activation regions (over all of input space) for $\mathcal{N}_0^{\text{bias}}$ is no more than that for $\mathcal{N}$.*

A key point in the proof is the scale equivariance of ReLU networks with zero bias.

Lemma 15. *Let $\mathcal{N}$ be a ReLU network with biases set identically to zero. Then,*

(a) $\mathcal{N}$ is equivariant under positive constant multiplication: $\mathcal{N}(cx) = c\mathcal{N}(x)$ for each $c > 0$.

(b) For every activation region R of $\mathcal{N}$, and every point x in R, all points cx are also in R for $c > 0$ (this implies that R is a convex cone).

Proof. Note that each neuron of the network computes a function of the form $z(x_1, \ldots, x_m) = \text{ReLU}(\sum_{i=1}^m w_i x_i)$. Note that:

$$z(cx_1, \ldots, cx_m) = \text{ReLU}\left(c \sum_{i=1}^m w_i x_i\right) = c \cdot \text{ReLU}\left(\sum_{i=1}^m w_i x_i\right) = cz(x_1, \ldots, x_m).$$

Thus, each neuron is equivariant under multiplication by positive constants c, and thus the overall network must be as well, proving (a). Note also that the activation patterns of ReLUs for x and cx must also be identical, implying that x and cx lie in the same linear region. This proves (b).

Wee now turn to the proof of Proposition 14. We proceed by defining an injective mapping from regions of $\mathcal{N}_0^{\text{bias}}$ to regions of $\mathcal{N}$. For each linear region R of $\mathcal{N}_0^{\text{bias}}$, pick a point $x_R \in R$. By the Lemma, $cx_R \in R$ for each $c > 0$. Let $\mathcal{N}_{1/c}^{\text{bias}}$ be the network obtained from $\mathcal{N}$ by dividing all biases by c, and observe that $\mathcal{N}(cx) = c\mathcal{N}_{1/c}^{\text{bias}}(x)$, with the same activation pattern between the two networks. But by picking c sufficiently large, $\mathcal{N}_{1/c}^{\text{bias}}$ becomes arbitrarily close to $\mathcal{N}_0^{\text{bias}}$. We conclude that for some sufficiently large c_R, $\mathcal{N}_0^{\text{bias}}(c_R x_R)$ and $\mathcal{N}(c_R x_R)$ have the same pattern of activations, so the regions of $\mathcal{N}$ in which $c_R x_R$ lies must be distinct for all distinct R. Thus, the number of regions of $\mathcal{N}$ must be at least as large as the number of regions of $\mathcal{N}_0^{\text{bias}}$, as desired. □

An informal argument suggests that Proposition 14 is relatively weak. As a consequence of Lemma 15(b), for any zero-bias net $\mathcal{N}_0^{\text{bias}}$ and any closed surface S containing the origin, S must intersect every linear region of $\mathcal{N}_0^{\text{bias}}$ (since such regions are convex cones with apices at the origin). Taking S to be the unit hypercube in n_{in} dimensions, the total number of activation regions of $\mathcal{N}_0^{\text{bias}}$ is no more than the sum of the numbers of activation regions intersecting each of its $2n_{\text{in}}$ facets, each of dimension $(n_{\text{in}} - 1)$. We expect from Theorem 5 that the number of activation regions intersecting each facet is like $(\#\text{neurons})^{n_{\text{in}}-1}$. Thus, the total number of activation regions of $\mathcal{N}_0^{\text{bias}}$ should grow no faster than $n_{\text{in}}\,(\#\text{neurons})^{n_{\text{in}}-1}$, instead of the $n_{\text{in}}\,(\#\text{neurons})^{n_{\text{in}}}$ directly implied by Proposition 14.

Lemma 15 implies that networks with small bias are *almost* scale equivariant. Figure 7 shows the activation regions for a network initialized with very small biases (i.i.d. normal with variance 10^{-6}). Part (a) displays regions intersecting a plane through the origin, indicating that, outside of a small radius around the origin, all regions are infinite and are approximated by cones. Thus, ReLU networks near initialization are almost scale equivariant except for sample points very close to the origin, a simple but potentially useful property that has not been widely recognized. During training, biases grow and the radius grows within which finite regions occur. Note that, as shown in part (b) of the figure, a plane not containing the origin does not reveal this structure, as such a plane can have finite intersection with many regions that are in fact infinite.