On the Number of Linear Regions of Convolutional Neural Networks

Huan Xiong [1] **Lei Huang** [2] **Mengyang Yu** [2] **Li Liu** [2] **Fan Zhu** [2] **Ling Shao** [1 2]

Abstract

One fundamental problem in deep learning is understanding the outstanding performance of deep Neural Networks (NNs) in practice. One explanation for the superiority of NNs is that they can realize a large class of complicated functions, i.e., they have powerful expressivity. The expressivity of a ReLU NN can be quantified by the maximal number of linear regions it can separate its input space into. In this paper, we provide several mathematical results needed for studying the linear regions of CNNs, and use them to derive the maximal and average numbers of linear regions for one-layer ReLU CNNs. Furthermore, we obtain upper and lower bounds for the number of linear regions of multi-layer ReLU CNNs. Our results suggest that deeper CNNs have more powerful expressivity than their shallow counterparts, while CNNs have more expressivity than fully-connected NNs per parameter.

1. Introduction

Over the past decade, deep Neural Networks (NNs), especially deep Convolutional Neural Networks (CNNs), have attracted much attention and achieved state-of-the-art results in many machine learning tasks, such as speech recognition, image classification, and video games (Hinton et al., 2012; Goodfellow et al., 2013; Sainath et al., 2013; Abdel-Hamid et al., 2014; Silver et al., 2016). Various popular and powerful CNNs, such as AlexNet (Krizhevsky et al., 2012), VGGNet (Simonyan & Zisserman, 2014), GoogleNet (Szegedy et al., 2015) and ResNet (He et al., 2016), have empirically shown that applying deeper networks can significantly improve the performance of various network architectures. A key problem in the study of deep learning is to understand why neural networks, especially very deep neural networks, perform well in practice.

[1]Mohamed bin Zayed University of Artificial Intelligence, UAE [2]Inception Institute of Artificial Intelligence, Abu Dhabi, UAE. Correspondence to: Huan Xiong <huan.xiong@mbzuai.ac.ae>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

One explanation for the superiority of NNs is their powerful expressivity, i.e., they can represent a large classes of functions arisen in practice. It has been shown that an NN with only one hidden layer can adequately approximate any given continuous function if its width is large enough (Cybenko, 1989; Funahashi, 1989; Hornik, 1991; Barron, 1994). However, normally the width of such a hidden layer has to be exponentially large in order to approximate a given function to arbitrary precision. In contrast, if multiple layers are involved, (Hanin, 2017; Hanin & Sellke, 2017; Lu et al., 2017) proved that to approximate any given Lebesgue-integrable function from $\mathbb{R}^n$ to $\mathbb{R}$ to arbitrary precision, one only needs to apply some multi-layer NN with width at most $n+1$, while the depth depends on the given function and may be very large. Although these approximate results show that NNs can represent a large class of functions, there are few hints on how to determine the suitable architectures needed to realise a given function or which architectures are more efficient. Recently, several theoretical studies have been conducted to compare the efficiency of distinct architectures. It was proved in (Telgarsky, 2015; 2016; Arora et al., 2016) that certain functions realized by some deep architectures will require a shallow network with exponentially more parameters to represent. For example, (Telgarsky, 2016) showed that, for any positive integer n, there exist some networks with depth $\Theta(n^3)$, width $\Theta(1)$, and $\Theta(1)$ parameters, that cannot be approximated by an $\mathcal{O}(n)$-layer network unless it has a width of $\Omega(2^n)$. Their results reveal that deeper networks usually have more powerful expressivity of functions, which provides an explanation for why deeper networks outperform shallow networks with the same number of parameters in many practical tasks.

A natural measure for characterizing the expressivity of NNs is the maximal number of distinct linear regions (Pascanu et al., 2013) in the domain of functions that can be computed by NNs. Among this direction, people mainly focus on NNs whose activation functions are Rectified Linear Units (ReLUs), which were first introduced in 2000 (Hahnloser et al., 2000; Hahnloser & Seung, 2001) and have been widely adopted in various architectures since 2011 (Glorot et al., 2011). It is known that the composition of piecewise linear[1] functions is still piecewise linear; thus, every feed-forward

[1]Although "piecewise affine" would be more accurate, we use "piecewise linear" here since it is a conventional concept.

ReLU NN (neural network with only ReLU activations and linear hidden layers) with certain parameters can be seen as a piecewise linear function. This means that the input space of a ReLU network can be divided into several distinct pieces (we call them linear regions), such that the function represented by the network is affine when restricted to each piece. Then, the expressivity of a ReLU network can be quantified by the maximal number of linear regions it can separate its input space into. Pascanu et al. (2013) first considered a one-layer fully-connected ReLU network with n_0 inputs and n_1 hidden neurons, and showed that its maximal number of linear regions equals $\sum_{i=0}^{n_0}\binom{n_1}{i}$ by translating this problem to a counting problem of regions of hyperplane arrangements in general position (the definition of "general position" is given in the Section 1 of the Supplementary Material), then directly applying Zaslavsky's Theorem (Zaslavsky, 1975; Stanley, 2004). Furthermore, using the idea of identifying distinct linear regions, they derived a lower bound $\left(\prod_{l=0}^{L-1}\left\lfloor\frac{n_l}{n_0}\right\rfloor\right)\sum_{i=0}^{n_0}\binom{n_L}{i}$ for the maximal number of linear regions of a fully-connected ReLU network with n_0 inputs and L hidden layers of widths $n_1, n_2, \ldots, n_L$. Based on these results, they concluded that deep fully-connected ReLU NNs have exponentially more maximal linear regions than their shallow counterparts with the same number of parameters. Later, the lower bound was improved to $\left(\prod_{l=0}^{L-1}\left\lfloor\frac{n_l}{n_0}\right\rfloor^{n_0}\right)\sum_{i=0}^{n_0}\binom{n_L}{i}$ by Montúfar et al. (2014). Following their work, various results on the lower and upper bounds for the maximal number of linear regions of fully-connected ReLU NNs have been obtained (Bianchini & Scarselli, 2014; Telgarsky, 2015; Poole et al., 2016; Montúfar, 2017; Raghu et al., 2017; Serra et al., 2018; Croce et al., 2018; Hu & Zhang, 2018; Serra & Ramalingam, 2018; Hanin & Rolnick, 2019a;b). For example, Arora et al. (2016) obtained a lower bound $2\sum_{i=0}^{n_0-1}\binom{m-1}{i}n^{L-1}$ for $n_1 = 2m$ and $n_2 = n_3 = \ldots = n_L = n$. Raghu et al. (2017) derived an upper bound $\mathcal{O}(n^{Ln_0})$ when $n_1 = n_2 = \ldots = n_L = n$. Montúfar et al. (2017) proved an upper bound of $\prod_{l=1}^{L}\sum_{i=0}^{m_l}\binom{n_l}{i}$ where $m_l = \min\{n_0, n_1, n_2, \ldots, n_{l-1}\}$. Later, these lower and upper bounds were improved by (Serra et al., 2018). Recently, Hanin et al. (Hanin & Rolnick, 2019a;b) studied the average number of linear regions when the weights range over $\mathbb{R}^{\#weights}$ and derived an upper bound for the expectation of the number of linear regions of ReLU NNs under several mild assumptions. Other studies have replaced the ReLU activation with the maxout activation or piecewise linear functions, and derived several bounds for the number of linear regions in these cases (Montufar et al., 2014; Hu & Zhang, 2018).

Most studies on the number of linear regions of ReLU NNs assume that the networks are fully-connected. Under this assumption, the problem is equivalent to counting regions of hyperplane arrangements in general position. Thus, one can use a well-established mathematical tool on hyperplane arrangements, Zaslavsky's Theorem (Zaslavsky, 1975), to directly obtain the maximal numbers of linear regions for one-layer fully-connected ReLU NNs, then derive the upper and lower bounds for multi-layer NNs by induction. Since CNNs are very popular in practice, it is natural to study an analogous problem on the number of linear regions for ReLU CNNs. However, as far as we know, there are no specific results for CNNs so far. The difficulty is that, although the problem for CNNs can also be translated to counting regions of hyperplane arrangements, usually the corresponding hyperplane arrangements are not in general position for CNNs, as discussed in Section 3 and the Supplementary Material. Therefore, mathematical tools like Zaslavsky's Theorem cannot be directly applied.

Our Contributions. In this paper, we establish new mathematical tools needed to study hyperplane arrangements (which usually are not in general position) arisen in CNN case, and use them to derive results on the number of linear regions for ReLU CNNs. *To the best of our knowledge, our paper is the first work on calculating the number of linear regions for CNNs.* The main contributions of this work are:

- We translate the problem of counting the linear regions of CNNs to a problem on counting the regions of some class of hyperplane arrangements which usually are not in general position, and develop suitable mathematical tools to solve this problem. Through this we provide the exact formula for the maximal number of linear regions of a one-layer ReLU CNN $\mathcal{N}$ and show that it actually equals the expectation of the number of linear regions when the weights of $\mathcal{N}$ range over $\mathbb{R}^{\#weights}$. The asymptotic formula for this number is also derived.
- Furthermore, we derive upper and lower bounds for the number of linear regions of multi-layer ReLU CNNs by induction and the idea of identifying distinct linear regions.
- Based on these bounds, we show that deep ReLU CNNs have exponentially more linear regions per parameter than their shallow counterparts under some mild assumptions on the architectures. This means that deep CNNs have more powerful expressivity than shallow ones and thus provides some hints on why CNNs normally perform better as they get deeper. We also show that ReLU CNNs have much more expressivity than the fully-connected ReLU NNs with asymptotically the same number of parameters, input dimension and number of layers.

This paper is organized as follows. We provide a detailed description of the CNN architectures that will be considered throughout the paper, and then introduce the definition

of activation patterns and linear regions in Section 2. In Section 3, we obtain the maximal and average numbers of linear regions of one-layer ReLU CNNs in Theorem 2. In Section 4, we derive results on multi-layer ReLU CNNs. A comparison on the expressivity of distinct architectures is given in Section 5. We briefly explain the experimental settings for verifying our results by sampling methods in Section 6. In Section 7, we provide the conclusion and propose future directions. The preliminary knowledge on hyperplane arrangements and the proofs of Theorems are given in the Supplementary Material.

2. Preliminary

In this section, we fix some notations and introduce the CNN architecture which will be considered in this paper. Let $\mathbb{N}$, $\mathbb{N}^+$ and $\mathbb{R}$ be the sets of nonnegative integers, positive integers and real numbers, respectively. For a set S, let $\#S$ denote the number of elements in S. In this paper, we consider ReLU CNNs $\mathcal{N}$ with L hidden convolutional layers (we exclude pooling layers and fully-connected layers, and do not use zero-padding for simplicity). Let the dimension of input neurons of $\mathcal{N}$ be $n_0^{(1)} \times n_0^{(2)} \times d_0$, where $n_0^{(1)}, n_0^{(2)}, d_0$ are the height, the width and the depth of the input space (we also call the input space the 0-th layer), respectively. Assume that there are $n_l^{(1)} \times n_l^{(2)} \times d_l$ neurons (i.e., d_l feature maps with the dimension $n_l^{(1)} \times n_l^{(2)}$) in the l-th hidden layer for $1 \le l \le L$. The Rectified Linear Unit (ReLU) is adopted as the activation function for each neuron in the hidden layers. There are d_l filters with dimension $f_l^{(1)} \times f_l^{(2)} \times d_{l-1}$ between neurons in the $(l-1)$-th and the l-th hidden layers. Such filters slide from left to right and from top to bottom across feature maps as far as possible with a stride s_l. We assume that the output layer has only one unit, which is a linear combination of the outputs in the L-th hidden layer. As explained in Lemma 2 from (Pascanu et al., 2013), the number of (linear) output units of a ReLU NN does not affect the number of linear regions that it can realize since the composition of affine functions is still affine. By the same argument this claim is also true for a ReLU CNN. *Therefore, in this paper, we take one unit in the output layer for simplicity and ignore the output layer in the statements of results.* Let $X^0 = (X^0_{a,b,c})_{n_0^{(1)} \times n_0^{(2)} \times d_0} \in \mathbb{R}^{n_0^{(1)} \times n_0^{(2)} \times d_0}$ be the inputs of $\mathcal{N}$ and $X^l = (X^l_{a,b,c})_{n_l^{(1)} \times n_l^{(2)} \times d_l} \in \mathbb{R}^{n_l^{(1)} \times n_l^{(2)} \times d_l}$ be the outputs of the l-th hidden layer. The weights $W = (W^1, W^2, \ldots, W^L)$ and biases $B = (B^1, B^2, \ldots, B^L)$ are drawn from a fixed distribution μ which has densities with respect to Lebesgue measure in $\mathbb{R}^{\#weights+\#bias}$, where $W^l = (W^{l,1}, W^{l,2}, \ldots, W^{l,d_l})$ such that $W^{l,k} = (W^{l,k}_{a,b,c})_{f_l^{(1)} \times f_l^{(2)} \times d_{l-1}} \in \mathbb{R}^{f_l^{(1)} \times f_l^{(2)} \times d_{l-1}}$ is the weight matrix of the k-th filter between neurons in the $(l-1)$-th and the l-th hidden layers; and $B^l = (B^{l,1}, B^{l,2}, \ldots, B^{l,d_l}) \in \mathbb{R}^{d_l}$, such that $B^{l,k} \in \mathbb{R}$ is the bias for the k-th filter between neurons in the $(l-1)$-th and the l-th hidden layers. Therefore, for any given weights W and biases B, this CNN can been seen as a piece-wise linear function $\mathcal{F}_{\mathcal{N},W,B} : \mathbb{R}^{n_0^{(1)} \times n_0^{(2)} \times d_0} \to \mathbb{R}$ given by

$$\mathcal{F}_{\mathcal{N},W,B}(X^0) = g_{L+1} \circ h_L \circ g_L \circ \cdots \circ h_1 \circ g_1(X^0),$$

where g_l is an affine function and h_l is a ReLU activation function. More specifically, let $Z^l(X^0;\theta) = (Z^l_{i,j,k}(X^0;\theta))_{n_l^{(1)} \times n_l^{(2)} \times d_l} \in \mathbb{R}^{n_l^{(1)} \times n_l^{(2)} \times d_l}$ be the pre-activations of the l-th layer, where $\theta := \{W, B\}$ is a fixed set of parameters (weights and biases) in the CNN $\mathcal{N}$. For $1 \le l \le L$, we have

$$\begin{aligned} Z^l_{i,j,k}(X^0;\theta) &= g_l(X^{l-1}) \\ &= \sum_{a=1}^{f_l^{(1)}} \sum_{b=1}^{f_l^{(2)}} \sum_{c=1}^{d_{l-1}} W^{l,k}_{a,b,c} X^{l-1}_{a+(i-1)s_l, b+(j-1)s_l, c} + B^{l,k} \end{aligned} \quad (1)$$

and

$$X^l_{i,j,k} = h_l(Z^l_{i,j,k}(X^0;\theta)) = \max(Z^l_{i,j,k}(X^0;\theta), 0). \quad (2)$$

The following relation between the number of neurons in the $(l-1)$-th and the l-th layers are easy to derive.

Lemma 1 ((Dumoulin & Visin, 2016)). *For $1 \le l \le L$, we have $n_l^{(1)} = \lfloor \frac{n_{l-1}^{(1)} - f_l^{(1)}}{s_l} \rfloor + 1$ and $n_l^{(2)} = \lfloor \frac{n_{l-1}^{(2)} - f_l^{(2)}}{s_l} \rfloor + 1$, where $\lfloor x \rfloor$ is the greatest integer less than or equal to x.*

Remark 1. *When the stride $s_l = 1$, we have $n_l^{(1)} = n_{l-1}^{(1)} - f_l^{(1)} + 1$ and $n_l^{(2)} = n_{l-1}^{(2)} - f_l^{(2)} + 1$. In this case, when a filter slides, all neurons in the $(l-1)$-th hidden layer are involved in the convolutional calculation.*

By analogy with the ReLU NN case (Pascanu et al., 2013; Montufar et al., 2014; Serra et al., 2018; Brandfonbrener, 2018; Hanin & Rolnick, 2019a;b), we introduce the following definition of activation patterns and linear regions for ReLU CNNs.

Definition 1 (Activation Patterns and Linear Regions). *Let $\mathcal{N}$ be a ReLU CNN with L hidden convolutional layers given above. An activation pattern of $\mathcal{N}$ is a function $\mathcal{P}$ from the set of neurons to $\{1, -1\}$, i.e., for each neuron z in $\mathcal{N}$, we have $\mathcal{P}(z) \in \{1, -1\}$. Let θ be a fixed set of parameters (weights and biases) in $\mathcal{N}$, and $\mathcal{P}$ be an activation pattern. The region corresponding to $\mathcal{P}$ and θ is*

$$\begin{aligned} \mathcal{R}(\mathcal{P};\theta) := \{X^0 \in \mathbb{R}^{n_0^{(1)} \times n_0^{(2)} \times d_0} : \\ z(X^0;\theta) \cdot \mathcal{P}(z) > 0, \quad \forall z \text{ a neuron in } \mathcal{N}\}, \end{aligned}$$

where $z(X^0;\theta)$ is the pre-activation of a neuron z. A linear region of $\mathcal{N}$ at θ is a non-empty set $\mathcal{R}(\mathcal{P},\theta) \neq \emptyset$ for

some activation pattern $\mathcal{P}$. Let $R_{\mathcal{N},\theta}$ denote the number of linear regions of $\mathcal{N}$ at θ, i.e., $R_{\mathcal{N},\theta} := \#\{\mathcal{R}(\mathcal{P};\theta) : \mathcal{R}(\mathcal{P};\theta) \neq \emptyset$ for some activation pattern $\mathcal{P}\}$. Moreover, let $R_{\mathcal{N}} := \max_\theta R_{\mathcal{N},\theta}$ denote the maximal number of linear regions of $\mathcal{N}$ when θ ranges over $\mathbb{R}^{\#weights+\#bias}$.

Remark 2. *By the above definition it is easy to check that each non-empty $\mathcal{R}(\mathcal{P};\theta)$ is a convex set. Furthermore, $\mathcal{F}_{\mathcal{N},W,B}$ becomes an affine function when restricted to each nonempty linear region $\mathcal{R}(\mathcal{P};\theta)$ of $\mathcal{N}$. Thus $\mathcal{F}_{\mathcal{N},W,B}$ can represent a piecewise linear function with $R_{\mathcal{N},\theta}$ linear pieces. Therefore, the number $R_{\mathcal{N},\theta}$ of linear regions can be seen as a measure on the expressivity of a CNN. The more linear regions a CNN has, the more complicated functions it can represent. The aim of this paper is to provide a characterization of the number of linear regions for CNNs, and use it to compare the expressivity of different CNNs.*

By Definition 1, each activation pattern of a CNN $\mathcal{N}$ is a function $\mathcal{P}$ from the set of its neurons to $\{1,-1\}$. It is obvious that there are at most $2^{\#neurons}$ such functions. Therefore, the number of activation patterns is also at most $2^{\#neurons}$. Then, we derive the following trivial upper bound for the number of linear regions of a ReLU CNN. Actually, a similar result for a fully-connected ReLU NN is given in Proposition 3 from (Montufar et al., 2014).

Lemma 2. *Let $\mathcal{N}$ be a ReLU CNN with n hidden neurons. Then, the number $R_{\mathcal{N}}$ of linear regions of $\mathcal{N}$ is at most 2^n.*

3. The Number of Linear Regions for One-Layer CNNs

In this section, we obtain the exact formula for the maximal number and the average number of linear regions of one-layer CNNs, and derive their asymptotic formulas when the number of filters tends to infinity.

3.1. Exact Formulas.

First, we recall the following result on the maximal number of linear regions of one-layer fully-connected ReLU NNs.

Theorem 1 (Proposition 2 from (Pascanu et al., 2013))**.** *Let $\mathcal{N}$ be a one-layer ReLU NN with n_0 input neurons and n_1 hidden neurons. Then, the maximal number of linear regions of $\mathcal{N}$ is equal to $\sum_{i=0}^{n_0}\binom{n_1}{i}$.*

Theorem 1 was derived by translating this problem to a study on the number of regions of hyperplane arrangements in general position, then directly applying a pure mathematical result, Zaslavsky's Theorem (Zaslavsky, 1975; Stanley, 2004), which states that, when an arrangement with n_1 hyperplanes is in general position, $\mathbb{R}^{n_0}$ can be divided into $\sum_{i=0}^{n_0}\binom{n_1}{i}$ distinct regions. Basic background on hyperplane arrangements and general position is given in Section 1 of the Supplementary Material.

Since the set of ReLU CNNs can be seen as a subset of ReLU NNs, Theorem 1 also gives an upper bound for $R_{\mathcal{N}}$ where $\mathcal{N}$ is a one-layer ReLU CNN. However, for the CNN case, usually this upper bound is not equal to the exact number since the corresponding hyperplane arrangement are not in general position normally. In this paper, we develop new tools to study the number of regions of corresponding hyperplane arrangements (which are not in general position usually) for ReLU CNNs. More precisely, we translate the problem for ReLU CNNs to a tractable integer programming problem by techniques and results from combinatorics and linear algebra. (see Eqs. (3), (4) and Section 2 of the Supplementary Material). Our first main result is stated as follows, which shows that the exact number of $R_{\mathcal{N}}$ is much smaller than the upper bound given by Theorem 1 for a one-layer ReLU CNN $\mathcal{N}$.

Theorem 2. *Assume that $\mathcal{N}$ is a one-layer ReLU CNN with input dimension $n_0^{(1)} \times n_0^{(2)} \times d_0$ and hidden layer dimension $n_1^{(1)} \times n_1^{(2)} \times d_1$. The d_1 filters have the dimension $f_1^{(1)} \times f_1^{(2)} \times d_0$ and the stride s_1. Suppose that the parameters $\theta = \{W, B\}$ are drawn from a fixed distribution μ which has densities with respect to Lebesgue measure in $\mathbb{R}^{\#weights+\#bias}$. Define $I_{\mathcal{N}} = \{(i,j) : 1 \le i \le n_1^{(1)},\ 1 \le j \le n_1^{(2)}\}$ and $S_{\mathcal{N}} = (S_{i,j})_{n_1^{(1)} \times n_1^{(2)}}$ where*

$$S_{i,j} = \{(a+(i-1)s_1,\ b+(j-1)s_1, c) : 1 \le a \le f_1^{(1)},\ 1 \le b \le f_1^{(2)},\ 1 \le c \le d_0\}$$

for each $(i,j) \in I_{\mathcal{N}}$. Therefore, $S_{i,j}$ is the set of indexes of input neurons involved in the calculation of the pre-activation $Z^1_{i,j,k}(X^0;\theta)$. Furthermore, $\cup_{(i,j)\in I_{\mathcal{N}}} S_{i,j}$ is the set of indexes of input neurons involved in the convolutional calculation of $\mathcal{F}_{\mathcal{N},W,B}$. Let

$$K_{\mathcal{N}} := \{(t_{i,j})_{(i,j)\in I_{\mathcal{N}}} : t_{i,j} \in \mathbb{N},\ \sum_{(i,j)\in J} t_{i,j} \le \#\cup_{(i,j)\in J} S_{i,j}\ \ \forall J \subseteq I_{\mathcal{N}}\}. \tag{3}$$

Then, we obtain the following two results.

(i) The maximal number $R_{\mathcal{N}}$ of linear regions of $\mathcal{N}$ equals

$$R_{\mathcal{N}} = \sum_{(t_{i,j})_{(i,j)\in I_{\mathcal{N}}} \in K_{\mathcal{N}}} \prod_{(i,j)\in I_{\mathcal{N}}} \binom{d_1}{t_{i,j}}. \tag{4}$$

(ii) Moreover, Eq. (4) also equals the expectation of the number $R_{\mathcal{N},\theta}$ of linear regions of $\mathcal{N}$:

$$\mathbb{E}_{\theta\sim\mu}[R_{\mathcal{N},\theta}] = \sum_{(t_{i,j})_{(i,j)\in I_{\mathcal{N}}} \in K_{\mathcal{N}}} \prod_{(i,j)\in I_{\mathcal{N}}} \binom{d_1}{t_{i,j}}. \tag{5}$$

The detailed proof of Theorem 2 is given in the Supplementary Material. We briefly explain the idea below.

Outline of the Proof of Theorem 2. First, by Definition 1, we translate the problem to the calculation of the number of regions of some specific hyperplane arrangements which are not in general position usually. Next, in Proposition 2 of the Supplementary Material, we derive a generalization of Zaslavsky's Theorem, which can be used to handle a large class of hyperplane arrangements that are not in general position. More specifically, we obtain an upper bound for the number of regions of such hyperplane arrangements and show that if a hyperplane arrangement satisfies the two conditions (i) and (ii) in Proposition 2, then this upper bound equals the exact number of its regions. The rest of Section 2 (Lemmas 3 – 7) in the Supplementary Material is devoted to showing that, actually, the specific hyperplane arrangements corresponding to a one-layer ReLU CNN $\mathcal{N}$ satisfy the conditions for hyperplane arrangements in Proposition 2. Thus, finally we can apply Proposition 2 of the Supplementary Material to derive the maximal and average numbers of linear regions for $\mathcal{N}$. □

Next, we provide several examples to explain Theorem 2.

Example 1. *Let $n_0^{(1)} = d_0 = f_1^{(1)} = s_1 = 1$, $n_0^{(2)} = 3$ and $f_1^{(2)} = 2$ in Theorem 2. Then by Lemma 1 we have $n_1^{(1)} = 1$ and $n_1^{(2)} = 2$. Furthermore, we obtain $I_{\mathcal{N}} = \{(1,1),(1,2)\}$, $S_{1,1} = \{(1,1,1),(1,2,1)\}$, $S_{2,1} = \{(1,2,1),(1,3,1)\}$ and $K_{\mathcal{N}} = \{(t_{1,1},t_{1,2}) \in \mathbb{N}^2 : t_{1,1} \leq 2,\ t_{1,2} \leq 2,\ t_{1,1}+t_{1,2} \leq 3\} = \{(0,0),(0,1),(0,2),(1,0),(1,1),(1,2),(2,0),(2,1)\}$. Finally, by Eq. (4) we derive*

$$R_{\mathcal{N}} = \sum_{(t_{1,1},t_{1,2})\in K_{\mathcal{N}}} \binom{d_1}{t_{1,1}}\binom{d_1}{t_{1,2}} = d_1^3 + d_1^2 + d_1 + 1,$$

which is also verified by our experiments for $1 \leq d_1 \leq 8$ (the experimental settings are given in Section 6). On the other hand, by Lemma 2, we have $R_{\mathcal{N}} \leq 2^{2d_1}$; by Theorem 1, we obtain $R_{\mathcal{N}} \leq \sum_{i=0}^{3}\binom{2d_1}{i}$ (since the CNN in Example 1 can be seen as a one-layer NN with 3 input neurons and $2d_1$ hidden neurons). When $1 \leq d_1 \leq 8$, the above bounds for $R_{\mathcal{N}}$ are given in Table 1 (more examples are given in Section 5 of the Supplementary Material). As can be seen, the exact number of $R_{\mathcal{N}}$ we obtained in Theorem 2 is smaller than the upper bounds obtained by previous methods.

Example 2. *Let $s_1 = 1$, $f_1^{(1)} = n_0^{(1)}$ and $f_1^{(2)} = n_0^{(2)}$. Then $n_1^{(1)} = n_1^{(2)} = 1$. Therefore, the CNN becomes a one-layer fully-connected ReLU NN with d input neurons and d_1 hidden neurons where $d = n_0^{(1)} \times n_0^{(2)} \times d_0$. Under these assumptions, by Theorem 2 we have $I_{\mathcal{N}} = \{(1,1)\}$, $K_{\mathcal{N}} = \{k \in \mathbb{Z} : 0 \leq k \leq d\}$. Thus, by (4) the maximal number of linear regions of a one-layer fully-connected ReLU NN $\mathcal{N}$ with input dimension d and output dimension d_1 equals $R_{\mathcal{N}} = \sum_{k=0}^{d}\binom{d_1}{k}$, which implies the well-known result in Theorem 1 (see (Pascanu et al., 2013; Montufar et al., 2014; Serra et al., 2018; Hanin & Rolnick, 2019a;b)). This means that Theorem 2 is a more general result than Theorem 1.*

Example 3. *Let $s_1 = f_1^{(1)} = f_1^{(2)} = 1$, which means that each filter has the dimension $1 \times 1 \times d_0$. Then, $S_{i,j} = \{(i,j,c) : 1 \leq c \leq d_0)\}$ is a d_0-element set and thus $\#\cup_{(i,j)\in J} S_{i,j} = d_0 \times \#J$ for each $J \subseteq I_{\mathcal{N}}$. Therefore, $K_{\mathcal{N}} = \{(t_{i,j})_{n_1^{(1)}\times n_1^{(2)}} : 0 \leq t_{i,j} \leq d_0\} = \{0,1,2,\ldots,d_0\}^{n_1^{(1)}\times n_1^{(2)}}$ and*

$$R_{\mathcal{N}} = \sum_{(t_{i,j})_{n_1^{(1)}\times n_1^{(2)}} \in \{0,1,2,\ldots,d_0\}^{n_1^{(1)}\times n_1^{(2)}}} \ \prod_{(i,j)\in I_{\mathcal{N}}} \binom{d_1}{t_{i,j}}.$$

When d_1 tends to infinity, we obtain

$$R_{\mathcal{N}} = \left(\frac{{d_1}^{d_0}}{d_0!}\right)^{n_1^{(1)}\times n_1^{(2)}} + \mathcal{O}(d_1^{n_0^{(1)}\times n_0^{(2)}\times d_0 - 1}). \quad (6)$$

We can see that $R_{\mathcal{N}} = \Theta(d_1^{n_0^{(1)}\times n_0^{(2)}\times d_0})$ in this example. In the following subsection, we will show that this also holds for general cases.

3.2. Asymptotic Analysis.

In this subsection, we study the asymptotic behavior of $R_{\mathcal{N}}$.

For two functions $f(n)$ and $g(n)$, we write $f(n) = \mathcal{O}(g(n))$ if there exists some positive constant $c > 0$ such that $f(n) \leq cg(n)$ for all n larger than some constant; $f(n) = \Omega(g(n))$ if there exists some positive constant c such that $f(n) \geq cg(n)$ for all n large enough; and $f(n) = \Theta(g(n))$ if there exists some positive constants c_1, c_2 such that $c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all n large enough.

We need the following lemma in the asymptotic analysis.

Lemma 3. *Let $\mathcal{N}, I_{\mathcal{N}}, K_{\mathcal{N}}, S_{i,j}$ be the same as defined in Theorem 2. Then, there always exists some $(t_{i,j})_{(i,j)\in I_{\mathcal{N}}} \in K_{\mathcal{N}}$ such that*

$$\sum_{(i,j)\in I_{\mathcal{N}}} t_{i,j} = \#\cup_{(i,j)\in I_{\mathcal{N}}} S_{i,j}.$$

We derive the following asymptotic formula for $R_{\mathcal{N}}$.

Theorem 3 (Asymptotic Analysis)**.** *Let $\mathcal{N}$ be the one-layer ReLU CNN defined in Theorem 2. Suppose that $n_0^{(1)}, n_0^{(2)}, d_0, f_1^{(1)}, f_1^{(2)}, s_1$ are some fixed integers. When d_1 tends to infinity, the asymptotic formula for the maximal number of linear regions of $\mathcal{N}$ behaves as $R_{\mathcal{N}} = \Theta(d_1^{\#\cup_{(i,j)\in I_{\mathcal{N}}} S_{i,j}})$ asymptotically. Furthermore, if all input neurons have been involved in the convolutional calculation,*

Table 1. The results for the maximal number of linear regions for a one-layer ReLU CNN $\mathcal{N}$ with input dimension $1 \times 3 \times 1$, hidden layer dimension $1 \times 2 \times d_1$, d_1 filters with dimension $1 \times 2 \times 1$, and stride $s_1 = 1$. More precisely, we have $R_{\mathcal{N}} = d_1^3 + d_1^2 + d_1 + 1$.

	$d_1 = 1$	$d_1 = 2$	$d_1 = 3$	$d_1 = 4$	$d_1 = 5$	$d_1 = 6$	$d_1 = 7$	$d_1 = 8$
$R_{\mathcal{N}}$ by Theorem 2	4	15	40	85	156	259	400	585
Upper bounds by Theorem 1	4	15	42	93	176	299	470	697
Upper bounds by Lemma 2	4	16	64	256	1024	4096	16384	65536

i.e., $\cup_{(i,j)\in I_{\mathcal{N}}} S_{i,j} = \{(a,b,c) : 1 \le a \le n_0^{(1)},\ 1 \le b \le n_0^{(2)},\ 1 \le c \le d_0\}$, *we have*

$$R_{\mathcal{N}} = \Theta(d_1^{n_0^{(1)} \times n_0^{(2)} \times d_0}). \tag{7}$$

Remark 3. *Note that, by Theorem 2, $R_{\mathcal{N}}$ grows at most as a polynomial of the number of neurons in the hidden layers, instead of growing exponentially fast, as suggested by Lemma 2. This implies that the upper bound in Lemma 2 is too loose and may not be achieved in practice.*

4. Bounds for the Number of Linear Regions for Multi-layer CNNs

In this section, we consider multi-layer CNNs and derive the lower and upper bounds for their maximal numbers of linear regions. First, we prove a lemma on the composition of two consecutive convolutional layers without an activation function layer between them. It is easy to see that such a composition is equivalent to a single convolutional layer. However, we could not find a precise description in the literature of this phenomenon concerning the relation of filter sizes and strides between these three convolutional layers. Therefore, we precisely describe this phenomenon and prove it in the following theorem.

Theorem 4. *Let $\mathcal{N}$ be a two-layer CNN without activation layers. For $l = 1, 2$, there are d_l filters with dimension $f_l^{(1)} \times f_l^{(2)} \times d_{l-1}$ and stride s_l between neurons in the $(l-1)$-th and the l-th hidden layers. Then $\mathcal{N}$ can be realized as a CNN with only one hidden convolutional layer such that its d_2 filters have size $f^{(1)} \times f^{(2)} \times d_0 = (f_1^{(1)} + (f_2^{(1)} - 1)s_1) \times (f_1^{(2)} + (f_2^{(2)} - 1)s_1) \times d_0$ and stride $s = s_1 s_2$. In particular, if $f_1^{(1)} = f_1^{(2)} = s_1 = 1$, we have $f^{(1)} \times f^{(2)} \times d_0 = f_2^{(1)} \times f_2^{(2)} \times d_0$ and stride $s = s_2$. That is, if the first convolutional layer has filter size $1 \times 1 \times d_0$ and stride 1, then the composition of the two convolutional layers has the same filter size and stride as the second convolutional layer.*

Now we are ready to derive the lower and upper bounds for the maximal numbers of linear regions of multi-layer CNNs using induction and the idea of identifying distinct linear regions motivated by (Pascanu et al., 2013; Montufar et al., 2014).

Theorem 5. *Suppose that $\mathcal{N}$ is a ReLU CNN with L hidden convolutional layers. The input dimension is $n_0^{(1)} \times n_0^{(2)} \times d_0$; the l-th hidden layer has dimension $n_l^{(1)} \times n_l^{(2)} \times d_l$ for $1 \le l \le L$; and there are d_l filters with dimension $f_l^{(1)} \times f_l^{(2)} \times d_{l-1}$ and stride s_l in the l-th layer. Assume that $d_l \ge d_0$ for each $1 \le l \le L$. Then, we have*

(i) The maximal number $R_{\mathcal{N}}$ of linear regions of $\mathcal{N}$ is at least (lower bound)

$$R_{\mathcal{N}} \ge R_{\mathcal{N}'} \prod_{l=1}^{L-1} \left\lfloor \frac{d_l}{d_0} \right\rfloor^{n_l^{(1)} \times n_l^{(2)} \times d_0}, \tag{8}$$

where $\mathcal{N}'$ is a one-layer ReLU CNN which has input dimension $n_{L-1}^{(1)} \times n_{L-1}^{(2)} \times d_0$ (the third dimension is d_0, not d_{L-1}), hidden layer dimension $n_L^{(1)} \times n_L^{(2)} \times d_L$, and d_L filters with dimension $f_L^{(1)} \times f_L^{(2)} \times d_0$ and stride s_L. Note that the exact formula of $R_{\mathcal{N}'}$ can be calculated by Eq. (4).

(ii) The maximal number $R_{\mathcal{N}}$ of linear regions of $\mathcal{N}$ is at most (upper bound)

$$R_{\mathcal{N}} \le R_{\mathcal{N}''} \prod_{l=2}^{L} \sum_{i=0}^{n_0^{(1)} n_0^{(2)} d_0} \binom{n_l^{(1)} n_l^{(2)} d_l}{i}, \tag{9}$$

where $\mathcal{N}''$ is a one-layer ReLU CNN which has input dimension $n_0^{(1)} \times n_0^{(2)} \times d_0$, hidden layer dimension $n_1^{(1)} \times n_1^{(2)} \times d_1$, and d_1 filters with dimension $f_1^{(1)} \times f_1^{(2)} \times d_0$ and stride s_1.

Example 4. *Let $\mathcal{N}$ be a two-layer CNN such that the input dimension is $1 \times 4 \times 1$, there are 2 filters with dimension $1 \times 2 \times 1$ and stride 1 in the first hidden layer; and d_2 filters with dimension $1 \times 2 \times 2$ and stride 1 in the second hidden layer. The dimensions of neurons in the first and second hidden layer are $1 \times 3 \times 2$ and $1 \times 2 \times d_2$ respectively. Theorem 5 yields the upper and lower bounds for $R_{\mathcal{N}}$ as shown in Table 2, which is compatible with the estimation of $R_{\mathcal{N}}$ by sampling methods in our experiment.*

Example 5 (Reduce to fully-connected ReLU NN case)**.** *Let $n_0^{(1)} = n_0^{(2)} = 1$ and $s_l = f_l^{(1)} = f_l^{(2)} = n_l^{(1)} = n_l^{(2)} = 1$ for each $1 \le l \le L$. Then the CNN becomes a fully-connected ReLU NN. Under these assumptions, by Eq.*

Table 2. The upper and lower bounds for $R_{\mathcal{N}}$ in Example 4.

	$d_2=1$	$d_2=2$	$d_2=3$	$d_2=4$	$d_2=5$	$d_2=6$	$d_2=7$	$d_2=8$
Upper bounds by Theorem 5	220	880	3520	13585	46640	138050	356180	819115
Estimation of $R_{\mathcal{N}}$ by sampling methods	170	261	685	1186	1796	2725	3398	4822
Lower bounds by Theorem 5	32	120	320	680	1248	2072	3200	4680

(4) *and Theorem 5 we have*

$$\sum_{k=0}^{d_0}\binom{d_L}{k}\times\prod_{l=1}^{L-1}\left\lfloor\frac{d_l}{d_0}\right\rfloor^{d_0}\leq R_{\mathcal{N}}\leq\prod_{l=1}^{L}\sum_{i=0}^{d_0}\binom{d_l}{i}, \quad (10)$$

which is a well-known result for fully-connected ReLU NNs (see Theorem 4 from (Montufar et al., 2014) for the first inequality; see Proposition 3 from (Montúfar, 2017), Theorem 1 from (Raghu et al., 2017) and Theorem 1 from (Serra et al., 2018) for the second inequality). Note that (10) implies that $R_{\mathcal{N}}=\Theta(d^{Ld_0})$ when d_0 is fixed and $d_1=d_2=\ldots=d_L=d\to+\infty$.

5. Expressivity Comparison of Different Network Architectures

In this section, we compare the expressivity of different network architectures in terms of the maximal number of linear regions based on the explicit formulas and bounds derived in Sections 3 and 4. The first conclusion is that deep CNNs usually have more expressivity than their shallow counterparts with the same number of parameters. Furthermore, we compare ReLU CNNs with the fully-connected ReLU NNs with asymptotically the same number of parameters, input dimension and number of layers. We show that CNNs have more expressivity than fully-connected NNs in this setting.

5.1. Deep CNNs v.s. Shallow CNNs

First, we calculate the number of parameters for CNNs.

Lemma 4. *Let $\mathcal{N}$ be an L-layer ReLU CNN in Theorem 5 (ignoring the output layer for simplicity). Then, the number of parameters in $\mathcal{N}$ is $\sum_{l=1}^{L}\left(f_l^{(1)}\times f_l^{(2)}\times d_{l-1}\times d_l+d_l\right)$.*

Now we can derive the number of linear regions per parameter for deep and shallow CNNs. The next result follows directly from Theorem 3, Theorem 5 and Lemma 4.

Theorem 6. *Let $\mathcal{N}_1$ be an L-layer ReLU CNN in Theorem 5 where $f_l^{(1)}$, $f_l^{(2)}=\mathcal{O}(1)$ for $1\leq l\leq L$, and $d_0=\mathcal{O}(1)$. When $d_1=d_2=\cdots=d_L=d$ tends to infinity, we obtain that $\mathcal{N}_1$ has $\Theta(Ld^2)$ parameters, and the ratio of $R_{\mathcal{N}_1}$ to the number of parameters of $\mathcal{N}_1$ is*

$$\frac{R_{\mathcal{N}_1}}{\#\text{ parameters of }\mathcal{N}_1}=\Omega\Big(\frac{1}{L}\cdot\left\lfloor\frac{d}{d_0}\right\rfloor^{d_0\sum_{l=1}^{L-1}n_l^{(1)}n_l^{(2)}-2}\Big).$$

For a one-layer ReLU CNN $\mathcal{N}_2$ with input dimension $n_0^{(1)}\times n_0^{(2)}\times d_0$ and hidden layer dimension $n_1^{(1)}\times n_1^{(2)}\times Ld^2$, when Ld^2 tends to infinity, $\mathcal{N}_2$ has $\Theta(Ld^2)$ parameters, and the ratio for $\mathcal{N}_2$ is

$$\frac{R_{\mathcal{N}_2}}{\#\text{ parameters of }\mathcal{N}_2}=\mathcal{O}\left((Ld^2)^{d_0n_0^{(1)}n_0^{(2)}-1}\right).$$

By Theorem 6 we will show that, with asymptotically the same number $\Theta(Ld^2)$ of parameters and the same number of input dimensions n^2d_0, deep CNNs can represent functions that have more number of linear regions than shallow CNNs. For simplicity, we set the stride $s_l=1$ for each layer, $n_0^{(1)}=n_0^{(2)}=n$ and $f_l^{(1)}=f_l^{(2)}=1$ in Theorem 6 (in practice, filters with small sizes such as 3×3, 5×5 and 7×7 are often adopted; for such cases, the conclusion is similar to the case $f_l^{(1)}=f_l^{(2)}=1$) in Theorem 6. Therefore, by Lemma 1 we have $n_l^{(1)}=n_l^{(2)}=n$ for each $1\leq l\leq L$. Then, the first ratio in Theorem 6 is

$$\frac{R_{\mathcal{N}_1}}{\#\text{ parameters of }\mathcal{N}_1}=\Omega\left(\frac{1}{L}\cdot\left\lfloor\frac{d}{d_0}\right\rfloor^{d_0(L-1)n^2-2}\right),$$

which grows at least exponentially fast with the number L of hidden layers and polynomially fast with the depth d of each hidden layer.

In contrast, the second ratio in Theorem 6 grows at most polynomially fast with L and d:

$$\frac{R_{\mathcal{N}_2}}{\#\text{ parameters of }\mathcal{N}_2}=\mathcal{O}\left((Ld^2)^{d_0n^2-1}\right).$$

Therefore, we obtain that $R_{\mathcal{N}_1}$ is far larger than $R_{\mathcal{N}_2}$ when L and d are large enough. By this we conclude that *ReLU deep CNNs have much more expressivity than their shallow counterparts with asymptotically the same number of parameters and the same number of input dimensions.*

5.2. CNNs v.s. Fully-connected NNs

In this subsection, we compare the expressivity of ReLU CNNs and fully-connected ReLU NNs with asymptotically the same number of parameters, input dimension and number of hidden layers. The settings for the L-layer ReLU CNN $\mathcal{N}_1$ is the same as in Subsection 5.1. For an L-layer fully-connected ReLU NN $\mathcal{N}_3$, we assume that the input dimension equals n^2d_0, and the number of neurons in each

of the L hidden layers equals d_0. Then $\mathcal{N}_1$ and $\mathcal{N}_3$ have asymptotically the same number of parameters $\mathcal{O}(Ld^2)$, input dimension $n^2 d_0$ and number L of layers. However, the maximal number of linear regions for $\mathcal{N}_1$ is

$$R_{\mathcal{N}_1} = \Omega\left(\left\lfloor \frac{d}{d_0} \right\rfloor^{Ld_0n^2}\right)$$

by (6) and Theorem 5. On the other hand, for $\mathcal{N}_3$ we obtain

$$\begin{aligned} R_{\mathcal{N}_3} &= \mathcal{O}\left(\binom{d}{n^2 d_0}^L\right) = \mathcal{O}\left(\frac{d^{Ld_0n^2}}{(n^2d_0)!^L}\right) \\ &= \mathcal{O}\left(\frac{d^{Ld_0n^2}}{(\sqrt{2\pi n^2 d_0})^L (n^2 d_0/e)^{Ld_0n^2}}\right) \end{aligned} \tag{11}$$

by (10) and the Stirling's formula (Flajolet & Sedgewick, 2009). Therefore,

$$\frac{R_{\mathcal{N}_1}}{R_{\mathcal{N}_3}} \geq \Omega\left((\sqrt{2\pi n^2 d_0})^L (n^2/e)^{Ld_0n^2}\right).$$

When n tends to infinity, the ratio $\frac{R_{\mathcal{N}_1}}{R_{\mathcal{N}_3}}$ also tends to infinity. Thus $R_{\mathcal{N}_1}$ is much larger than $R_{\mathcal{N}_3}$, and we conclude that *ReLU CNNs have much more expressivity than the fully-connected ReLU NNs with asymptotically the same number of parameters, input dimension and number of layers.*

6. Experimental Settings

We empirically validate our results by randomly sampling data points from the input space and determining which linear regions they belong to by Definition 1. For a given CNN architecture, we initialize the parameters (weights and biases) based on the He initialization (He et al., 2015). Given the sampled weight, each data point in the input space is sampled from a normal distribution with mean 0 and standard deviation v. We use v ranging from $\{3, 5, 7, 9, 11, 13\}$ and report the maximal number of linear regions from such v. We sample 2×10^9 data points in total, and for each data point, we determine which region it belongs to based on Definition 1 (for a new data point X^0, we simply calculate the sign of $z(X^0, \theta)$ for each neuron z and use it to determine whether X^0 belongs to a new region). This sampling method may skip some regions. Thus, the number of linear regions obtained by sampling is usually smaller than the exact number. However, when the number of sampling points is large enough, we can usually find almost all the linear regions. For example, we use this sampling method to find all $R_{\mathcal{N}}$ linear regions for one-layer CNNs $\mathcal{N}$ in Table 1, and find the number of regions between the lower and upper bounds for two-layer CNNs in Table 2. By these, we validate the correctness of our results. *We provide the codes for the experiments in the Supplementary Material.*

7. Conclusion and Future Work

In this paper, we obtained exact formulas for the maximal and average number of linear regions of one-layer ReLU CNNs, and derived lower and upper bounds for multi-layer CNNs. By these results, we concluded that deep ReLU CNNs have more expressivity than their shallow counterparts, while ReLU CNNs have more expressivity than fully-connected ReLU NNs per parameter.

To the best of our knowledge, our paper is the first work investigating the number of linear regions for CNNs. We plan to explore more aspects in the future based on this work. Possible future directions are summarized below.

(1) In this paper, we only consider ReLU CNNs without pooling layers, fully-connected layers, and zero-padding for simplicity. After adding pooling layers, the functions represented by ReLU CNNs are still piecewise linear, thus the definition of linear regions still applies. It would be interesting to study the number of linear regions for ReLU CNNs with pooling layers, fully-connected layers, and zero-padding in the future.

(2) In Theorem 2 we showed that the expectation of $R_{\mathcal{N},\theta}$ is equal to the maximal number $R_{\mathcal{N}}$ for a one-layer ReLU CNN $\mathcal{N}$. This result is consistent with the one-layer fully-connected NN case in (Hanin & Rolnick, 2019b) (see the last two sentences of Section 2 and the first sentence in Remark 1 of (Hanin & Rolnick, 2019b)). When the number of layers of the fully-connected NN is at least two, it is proved in (Hanin & Rolnick, 2019b) that the expectation of the number of linear regions is much smaller than the maximal number. It would be interesting to explore similar formulas for the expectation of $R_{\mathcal{N},\theta}$ for multi-layer ReLU CNNs. We believe this will be a more challenging topic than the fully-connected NN case due to the correlated weights and bias for CNNs.

(3) In Theorem 5 we derive lower and upper bounds for multi-layer CNNs. By Table 2 we can see that these bounds are not very close to each other. We would like to derive tighter bounds, or further exact formulas in the future.

(4) We would like to extend our method to study the changing number of linear regions for CNNs when the parameters are updated (for example, by backpropagation) with a small perturbation. When the parameters θ are replaced by some $\theta + \Delta\theta$, what is the relation between $R_{\mathcal{N},\theta}$ and $R_{\mathcal{N},\theta+\Delta\theta}$? For which parameters θ_1 and θ_2, the numbers $R_{\mathcal{N},\theta_1}$ and $R_{\mathcal{N},\theta_2}$ are equal to each other? During the training process, the parameters θ changes to some $\theta + \Delta\theta$. Thus, the answer to the above question may help us have a better understanding of the training process and optimization for CNNs. In fact, recently, Hanin and Rolnick (Hanin & Rolnick, 2019a;b) have already done some research on the changing number of linear regions during the training

process for fully-connected NNs.

(5) In (Hu & Zhang, 2018), the ReLU activation was generalized to piecewise linear (PWL) functions. The results on exact formulas and bounds for the number of linear regions for fully-connected PWL NNs were presented. In the future, we plan to replace the ReLU activation with PWL functions for CNNs and study their numbers of linear regions.

Acknowledgements

We really appreciate the valuable suggestions given by reviewers for improving the overall quality of this paper. We also would like to thank Dr. Jingtao Zang and Dr. Keqian Yan for helpful discussions.

References

Abdel-Hamid, O., Mohamed, A.-r., Jiang, H., Deng, L., Penn, G., and Yu, D. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.

Arora, R., Basu, A., Mianjy, P., and Mukherjee, A. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*, 2016.

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

Bianchini, M. and Scarselli, F. 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.

Brandfonbrener, D. The expressive power of neural networks. *CPSC 490*, 2018.

Croce, F., Andriushchenko, M., and Hein, M. Provable robustness of relu networks via maximization of linear regions. *arXiv preprint arXiv:1810.07481*, 2018.

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

Dumoulin, V. and Visin, F. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

Flajolet, P. and Sedgewick, R. *Analytic combinatorics*. cambridge University press, 2009.

Funahashi, K.-I. On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3): 183–192, 1989.

Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, 2011.

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

Hahnloser, R. H. and Seung, H. S. Permitted and forbidden sets in symmetric threshold-linear networks. In *Advances in neural information processing systems*, pp. 217–223, 2001.

Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., and Seung, H. S. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947, 2000.

Hanin, B. Universal function approximation by deep neural nets with bounded width and relu activations. *arXiv preprint arXiv:1708.02691*, 2017.

Hanin, B. and Rolnick, D. Complexity of linear regions in deep networks. In *International Conference on Machine Learning*, pp. 2596–2604, 2019a.

Hanin, B. and Rolnick, D. Deep relu networks have surprisingly few activation patterns. In *Advances in Neural Information Processing Systems*, pp. 359–368, 2019b.

Hanin, B. and Sellke, M. Approximating continuous functions by relu nets of minimal width. *arXiv preprint arXiv:1710.11278*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

Hu, Q. and Zhang, H. Nearly-tight bounds on linear regions of piecewise linear neural networks. *arXiv preprint arXiv:1810.13192*, 2018.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. In *Advances in neural information processing systems*, pp. 6231–6239, 2017.

Montúfar, G. Notes on the number of linear regions of deep neural networks. *Sampling Theory Appl., Tallinn, Estonia, Tech. Rep*, 2017.

Montufar, G. F., Pascanu, R., Cho, K., and Bengio, Y. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pp. 2924–2932, 2014.

Pascanu, R., Montufar, G., and Bengio, Y. On the number of response regions of deep feed forward networks with piece-wise linear activations. *arXiv preprint arXiv:1312.6098*, 2013.

Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. Exponential expressivity in deep neural networks through transient chaos. In *Advances in neural information processing systems*, pp. 3360–3368, 2016.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Dickstein, J. S. On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2847–2854. JMLR. org, 2017.

Sainath, T. N., Mohamed, A.-r., Kingsbury, B., and Ramabhadran, B. Deep convolutional neural networks for lvcsr. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 8614–8618. IEEE, 2013.

Serra, T. and Ramalingam, S. Empirical bounds on linear regions of deep rectifier networks. *arXiv preprint arXiv:1810.03370*, 2018.

Serra, T., Tjandraatmadja, C., and Ramalingam, S. Bounding and counting linear regions of deep neural networks. In *International Conference on Machine Learning*, pp. 4565–4573, 2018.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Stanley, R. P. An introduction to hyperplane arrangements. In *Lecture notes, IAS/Park City Mathematics Institute*, 2004.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Telgarsky, M. Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101*, 2015.

Telgarsky, M. Benefits of depth in neural networks. *arXiv preprint arXiv:1602.04485*, 2016.

Zaslavsky, T. *Facing Up to Arrangements: Face-Count Formulas for Partitions of Space by Hyperplanes*. Number 154 in Memoirs of the American Mathematical Society. American Mathematical Society, 1975. URL https://books.google.ae/books?id=2DUZAQAAIAAJ.