Neural Networks Should Be Wide Enough to Learn Disconnected Decision Regions

Quynh Nguyen [1] **Mahesh Chandra Mukkamala** [1] **Matthias Hein** [2]

Abstract

In the recent literature the important role of depth in deep learning has been emphasized. In this paper we argue that sufficient width of a feed-forward network is equally important by answering the simple question under which conditions the decision regions of a neural network are connected. It turns out that for a class of activation functions including leaky ReLU, neural networks having a pyramidal structure, that is no layer has more hidden units than the input dimension, produce necessarily connected decision regions. This implies that a sufficiently wide hidden layer is necessary to guarantee that the network can produce disconnected decision regions. We discuss the implications of this result for the construction of neural networks, in particular the relation to the problem of adversarial manipulation of classifiers.

1. Introduction

While deep learning has become state of the art in many application domains such as computer vision and natural language processing and speech recognition, the theoretical understanding of this success is steadily growing but there are still plenty of questions where there is little or no understanding. In particular, for the question how one should construct the network e.g. choice of activation function, number of layers, number of hidden units per layer etc., there is little guidance and only limited understanding on the implications of the choice e.g. "The design of hidden units is an extremely active area of research and does not yet have many definitive guiding theoretical principles." is a quote from the recent book on deep learning (Goodfellow et al., 2016, p. 191). Nevertheless there is recently progress in the understanding of these choices.

[1]Department of Mathematics and Computer Science, Saarland University, Germany [2]University of Tübingen, Germany. Correspondence to: Quynh Nguyen <quynh@cs.uni-saarland.de>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

The first important results are the universal approximation theorems (Cybenko, 1989; Hornik et al., 1989) which show that even a single hidden layer network with standard non-polynomial activation function (Leshno et al., 1993), like the sigmoid, can approximate arbitrarily well every continuous function over a compact domain of $\mathbb{R}^d$. In order to explain the success of deep learning, much of the recent effort has been spent on analyzing the representation power of neural networks from the perspective of depth (Delalleau & Bengio, 2011; Telgarsky, 2016; 2015; Eldan & Shamir, 2016; Safran & Shamir, 2017; Yarotsky, 2016; Poggio et al., 2016; Liang & Srikant, 2017; Mhaskar & Poggio, 2016). Basically, they show that there exist functions that can be computed efficiently by deep networks of linear or polynomial size but require exponential size for shallow networks. To further highlight the power of depth, (Montufar et al., 2014; Pascanu et al., 2014) show that the number of linear regions that a ReLU network can form in the input space grows exponentially with depth. Tighter bounds on the number of linear regions are later on developed by (Arora et al., 2018; Serra et al., 2018; Charisopoulos & Maragos, 2018). Another measure of expressivity so-called trajectory length is proposed by (Raghu et al., 2017). They show that the complexity of functions computed by the network along a one-dimensional curve in the input space also grows exponentially with depth.

While most of previous work can only show the existence of depth efficiency (*i.e.* there exist certain functions that can be efficiently represented by deep networks but not effectively represented or even approximated by shallow networks) but cannot show how often this holds for all functions of interest, (Cohen et al., 2016) have taken the first step to address this problem. In particular, by studying a special type of networks called convolutional arithmetic circuits – also known as Sum-Product networks (Poon & Domingos, 2011), the authors show that besides a set of measure zero, all functions that can be realized by a deep network of polynomial size require exponential size in order to be realized, or even approximated by a shallow network. Later, (Cohen & Shashua, 2016) show that this property however no longer holds for convolutional rectifier networks, which represents so far the empirically most successful deep learning architecture in practice.

Unlike most of previous work which focuses on the power of depth, (Lu et al., 2017; Hanin & Sellke, 2017) have recently shown that neural networks with ReLU activation function have to be wide enough in order to have the universal approximation property as depth increases. In particular, the authors show that the class of continuous functions on a compact set cannot be arbitrarily well approximated by an arbitrarily deep network if the maximum width of the network is not larger than the input dimension d. Moreover, it has been shown recently, that the loss surface of fully connected networks (Nguyen & Hein, 2017) and for convolutional neural networks (Nguyen & Hein, 2018) is well behaved, in the sense that almost all local minima are global minima, if there exists a layer which has more hidden units than the number of training points.

In this paper we study the question under which conditions on the network the decision regions of a neural network are connected respectively can potentially be disconnected. The decision region of a class is the subset of $\mathbb{R}^d$, where the network predicts this class. A similar study has been in (Makhoul et al., 1989; 1990) for feedforward networks with threshold activation functions, where they show that the initial layer has to have width $d+1$ in order that one can get disconnected decision regions. On an empirical level it has recently been argued (Fawzi et al., 2017) that the decision regions of the Caffe Network (Jia et al., 2014) on ImageNet are connected. In this paper we analyze feedforward networks with continuous activation functions as currently used in practice. We show in line with previous work that almost all networks which have a pyramidal structure up to the last hidden layer, that is the width of all hidden layers is smaller than the input dimension d, can only produce connected decision regions. We show that the result is tight by providing explicit counterexamples for the case $d+1$. We conclude that a guiding principle for the construction of neural networks should be that there is a layer which is wider than the input dimension as it would be a strong assumption that the Bayes optimal classifier must have connected decision regions. Interestingly, our result holds for leaky ReLU, that is $\sigma(t) = \max\{t, \alpha t\}$ for $0 < \alpha < 1$, whereas the result of (Hanin & Sellke, 2017) is for ReLU, that is $\sigma(t) = \max\{t, 0\}$, but "the generalization is not straightforward, even for activations of the form $\sigma(t) = \max\{l_1(t), l_2(t)\}$, where l_1, l_2 are affine functions with different slopes." We discuss also the implications of connected decision regions regarding the generation of adversarial samples, which will provide another argument in favor of larger width for neural network architectures.

2. Feedforward Neural Networks

We consider in this paper feedforward neural networks for multi-class classification. Let d be the input dimension and m the number of classes. Let L be the number of layers where the layers are indexed from $k = 0, 1, \ldots, L$ which respectively corresponds to the input layer, 1st hidden layer, ..., and the output layer L. Let n_k be the width of layer k. For consistency, we assume that $n_0 = d$ and $n_L = m$. Let $\sigma_k : \mathbb{R} \to \mathbb{R}$ be the activation function of every hidden layer $1 \leq k \leq L-1$. In the following, all functions are applied componentwise. We define $f_k : \mathbb{R}^d \to \mathbb{R}^{n_k}$ as the feature map of layer k, which computes for every input $x \in \mathbb{R}^d$ a feature vector at layer k defined as

$$f_k(x) = \begin{cases} x & k = 0 \\ \sigma_k\big(W_k^T f_{k-1}(x) + b_k\big) & 1 \leq k \leq L-1 \\ W_L^T f_{L-1}(x) + b_L & k = L \end{cases}$$

where $W_k \in \mathbb{R}^{n_{k-1} \times n_k}$ is the weight matrix at layer k. Please note that the output layer is linear as it is usually done in practice. We consider in the following activation functions $\sigma : \mathbb{R} \to \mathbb{R}$ which are continuous and strictly monotonically increasing. This is true for most of proposed activation functions, but does not hold for ReLU, $\sigma(t) = \max\{t, 0\}$. On the other hand, it has been argued in the recent literature, that the following variants are to be preferred over ReLU as they deal better with the vanishing gradient problem and outperform ReLU in prediction performance (He et al., 2015; Clevert et al., 2016). This is leaky ReLU (Maas et al., 2013):

$$\sigma(t) = \max\{t, \alpha t\} \quad \text{for} \quad 0 < \alpha < 1,$$

where typically α is fixed but it has also been optimized together with the network weights (He et al., 2015) and ELU (exponential linear unit) (Clevert et al., 2016):

$$\sigma(t) = \begin{cases} e^t - 1 & t < 0 \\ t & t \geq 0. \end{cases}.$$

Note that image of the activation function σ, $\sigma(\mathbb{R}) = \{\sigma(t) \,|\, t \in \mathbb{R}\}$, is equal to $\mathbb{R}$ for leaky ReLU and $(-1, \infty)$ for the exponential linear unit.

3. Connectivity of Decision Regions

In this section, we prove two results on the connectivity of the decision regions of a classifier. Both require that the activation function is continuous and strictly monotonically increasing. Our main Theorem 3.10 holds for feedforward networks of arbitrary depth and requires additionally $\sigma(\mathbb{R}) = \mathbb{R}$, the second Theorem 3.11 holds just for one hidden layer networks but has no further requirements on the activation function. Both show that in general pyramidal feedforward neural networks where the width of all the hidden layers is smaller than or equal to the input dimension can only produce connected decision regions.

3.1. Preliminary technical results

We first introduce the definitions and terminologies used in the following, before we prove or recall some simple results about continuous mappings from $\mathbb{R}^m$ to $\mathbb{R}^n$. For a function $f : U \to V$, where $\mathrm{dom}(f) = U \subseteq \mathbb{R}^m$ and $V \subseteq \mathbb{R}^n$, we denote for every subset $A \subseteq U$, the image $f(A)$ as $f(A) := \{f(x) \mid x \in A\} = \bigcup_{x\in A} f(x)$. Let $\mathrm{range}(f) := f(U)$.

Definition 3.1 (Decision region) *The decision region of a given class $1 \leq j \leq m$, denoted by C_j, is defined as*

$$C_j = \left\{x \in \mathbb{R}^d \;\middle|\; (f_L)_j(x) > (f_L)_k(x),\ \forall k \neq j\right\}.$$

Definition 3.2 (Connected set) *A subset $S \subseteq \mathbb{R}^d$ is called connected if for every $x, y \in S$, there exists a continuous curve $r : [0, 1] \to S$ such that $r(0) = x$ and $r(1) = y$.*

To prove our key Lemma 3.9, the following properties of connected sets and continuous functions are useful. All proofs are moved to the appendix due to limited space.

Proposition 3.3 *Let $f : U \to V$ be a continuous function. If $A \subseteq U$ is a connected set then $f(A) \subseteq V$ is also a connected set.*

Proposition 3.4 *The Minkowski sum of two connected subsets $U, V \subseteq \mathbb{R}^n$, defined as $U + V = \{u + v \mid u \in U, v \in V\}$, is a connected set.*

As our main idea is to transfer the connectedness of a set from the output layer back to the input layer, we require the notion of pre-image and inverse mapping.

Definition 3.5 (Pre-Image) *The pre-image of a function $f : U \to V$ is the set-valued function $f^{-1} : V \to U$ defined for every $y \in V$ as*

$$f^{-1}(y) = \{x \in U \mid f(x) = y\}.$$

Similarly, for every subset $A \subseteq V$, let

$$f^{-1}(A) = \bigcup_{y\in A} f^{-1}(y) = \{x \in U \mid f(x) \in A\}.$$

By definition, it holds for every subset $A \subseteq V$ that $f(x) \in A$ if and only if $x \in f^{-1}(A)$. Moreover, for every $A \subseteq V$

$$\begin{aligned} f^{-1}(A) &= f^{-1}(A \cap \mathrm{range}(f)) \cup f^{-1}(A \setminus \mathrm{range}(f)) \\ &= f^{-1}(A \cap \mathrm{range}(f)) \cup \emptyset \\ &= f^{-1}(A \cap \mathrm{range}(f)). \end{aligned}$$

As a deep feedforward network is a composition of the individual layer functions, we need the following property.

Proposition 3.6 *Let $f : U \to V$ and $g : V \to Q$ be two functions. Then it holds that $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$.*

Apart from the property of connectivity, we can also show the openness of a set when considering the pre-image of a given network. We recall the following standard result from topology (see *e.g.* Apostol, 1974, Theorem 4.23, p. 82).

Proposition 3.7 *Let $f : \mathbb{R}^m \to \mathbb{R}^n$ be a continuous function. If $U \subseteq \mathbb{R}^n$ is an open set then $f^{-1}(U)$ is also open.*

We now recall a standard result from calculus showing that under certain, restricted conditions the inverse of a continuous mapping exists and is as well continuous.

Proposition 3.8 *Let $f : \mathbb{R} \to f(\mathbb{R})$ be continuous and strictly monotonically increasing. Then the inverse mapping $f^{-1} : f(\mathbb{R}) \to \mathbb{R}$ exists and is continuous.*

The following lemma is a key ingredient in the following proofs. It allows us to show that the pre-image of an open and connected set by a one hidden layer network is again open and connected. Using the fact that deep networks can be seen as a composition of such individual layers, this will later on allow us to transfer the result to deep networks.

Lemma 3.9 *Let $m \geq n$ and $f : \mathbb{R}^m \to \mathbb{R}^n$ be a function defined as $f = \hat{\sigma} \circ h$ where $\hat{\sigma} : \mathbb{R}^n \to \mathbb{R}^n$ is defined as*

$$\hat{\sigma}(x) = \begin{pmatrix} \sigma(x_1) \\ \vdots \\ \sigma(x_n) \end{pmatrix}, \tag{1}$$

and $\sigma : \mathbb{R} \to \mathbb{R}$ is bijective, continuous and strictly monotonically increasing, $h : \mathbb{R}^m \to \mathbb{R}^n$ is a linear map defined as $h(x) = W^T x + b$ where $W \in \mathbb{R}^{m\times n}$ has full rank and $b \in \mathbb{R}^n$. If $V \subseteq \mathbb{R}^n$ is an open connected set then $f^{-1}(V) \subseteq \mathbb{R}^m$ is also an open connected set.

Proof: By Proposition 3.6, it holds that $f^{-1}(V) = h^{-1}(\hat{\sigma}^{-1}(V))$. As $\hat{\sigma}$ is a componentwise function, the inverse mapping $\hat{\sigma}^{-1}$ is given by the inverse mappings of the components

$$\hat{\sigma}^{-1} : \mathbb{R}^n \to \mathbb{R}^n,\ \hat{\sigma}^{-1}(x) = \begin{pmatrix} \sigma^{-1}(x_1) \\ \vdots \\ \sigma^{-1}(x_n) \end{pmatrix},$$

where under the stated assumptions the inverse mapping $\sigma^{-1} : \mathbb{R} \to \mathbb{R}$ exists by Lemma 3.8 and is continuous. Since $V \subseteq \mathbb{R}^n = \mathrm{dom}(\hat{\sigma}^{-1})$, $\hat{\sigma}^{-1}(V)$ is the image of the connected set V under the continuous map $\hat{\sigma}^{-1}$. Thus by Proposition 3.3, $\hat{\sigma}^{-1}(V)$ is connected. Moreover, $\hat{\sigma}^{-1}(V)$ is an open set by Proposition 3.7.

It holds for every $y \in \mathbb{R}^n$ that

$$h^{-1}(y) = \begin{cases} \emptyset & y \notin \mathrm{range}(h) \\ W(W^TW)^{-1}(y - b) + \ker(W^T) & y \in \mathrm{range}(h), \end{cases}$$

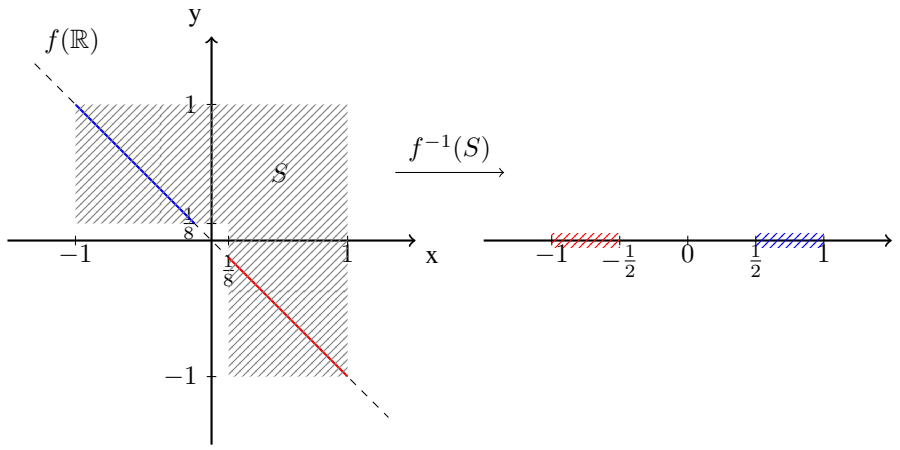


Figure 1. Left: illustration of the image of $\mathbb{R}$ under the mapping f, denoted as $f(\mathbb{R}) \subset \mathbb{R}^2$ for the toy example from (2) which maps into a lower-dimensional subspace (the diagonal line). Right: The pre-image $f^{-1}(S) \subset \mathbb{R}$ of the connected S becomes disconnected.

where the inverse of W^TW exists as W has full rank n (note that we assume $n \leq m$). As W has full rank and $m \geq n$, it holds that $\text{range}(h) = \mathbb{R}^n$ and thus

$$h^{-1}(y) = W(W^TW)^{-1}(y-b) + \ker(W^T), \quad \forall\, y \in \mathbb{R}^n.$$

Therefore it holds for $\hat{\sigma}^{-1}(V) \subseteq \mathbb{R}^n$ that

$$h^{-1}\big(\hat{\sigma}^{-1}(V)\big) = W(W^TW)^{-1}\big(\hat{\sigma}^{-1}(V) - b\big) + \ker(W^T),$$

where the first term is the image of the connected set $\hat{\sigma}^{-1}(V)$ under an affine mapping and thus is again connected by Proposition 3.3, the second term $\ker(W^T)$ is a linear subspace which is also connected. By Proposition 3.4, the Minkowski sum of two connected sets is connected. Thus $f^{-1}(V) = h^{-1}(\hat{\sigma}^{-1}(V))$ is a connected set. Moreover, as $f^{-1}(V)$ is the pre-image of the open set V under the continuous function f, it must be also an open set by Proposition 3.7. Thus $f^{-1}(V)$ is an open and connected set. □

Note that in Lemma 3.9, if $m < n$ and W has full rank then $\text{range}(h) \subsetneq \mathbb{R}^n$ and the linear equation $h(x) = y$ has a unique solution $x = (WW^T)^{-1}W(y-b)$ for every $y \in \text{range}(h)$ and thus

$$\begin{aligned} f^{-1}(V) &= h^{-1}\big(\sigma^{-1}(V)\big) \\ &= h^{-1}\big(\sigma^{-1}(V) \cap \text{range}(h)\big) \\ &= (WW^T)^{-1}W\big((\sigma^{-1}(V) \cap \text{range}(h)) - b\big). \end{aligned}$$

In this case, even though $\sigma^{-1}(V)$ is a connected set, the intersection $\sigma^{-1}(V) \cap \text{range}(h)$ can be disconnected which can imply that $f^{-1}(V)$ is disconnected and thus the decision region becomes disconnected.

We illustrate this with a simple example, where $m = 1$ and $n = 2$ with $\sigma(x) = x^3$ and $W^T = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ and $b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. In this case it holds that

$$f(x) = \hat{\sigma}(W^Tx + b) = \begin{pmatrix} \sigma(-x) \\ \sigma(x) \end{pmatrix} = \begin{pmatrix} -x^3 \\ x^3 \end{pmatrix}. \tag{2}$$

Figure 1 shows that $f(\mathbb{R})$ is a one-dimensional submanifold (in this case subspace) of $\mathbb{R}^2$ and provides an example of a set $S \subset \mathbb{R}^2$ where the pre-image $f^{-1}(S)$ is disconnected.

3.2. Main results

We show in the following that the decisions regions of feedforward networks which are pyramidal and have maximal width at most the input dimension d can only produce connected decision regions. We assume for the activation functions that $\sigma(\mathbb{R}) = \mathbb{R}$, which is fulfilled by leaky ReLU.

Theorem 3.10 *Let the width of the layers of the feedforward network network satisfy $d = n_0 \geq n_1 \geq \ldots \geq n_{L-1}$ and let $\sigma_l : \mathbb{R} \to \mathbb{R}$ be a continuous, strictly monotonically increasing function with $\sigma_l(\mathbb{R}) = \mathbb{R}$ for every layer $1 \leq l \leq L-1$ and all the weight matrices $(W_l)_{l=1}^{L-1}$ have full rank. Then every decision region C_j is an open connected subset of $\mathbb{R}^d$ for every $1 \leq j \leq m$.*

Proof: From Definition 3.1, it holds for every $1 \leq j \leq m$

$$C_j = \big\{x \in \mathbb{R}^d \;\big|\; f_{Lj}(x) - f_{Lk}(x) > 0, \forall k \neq j\big\}$$

where

$$\begin{aligned} &f_{Lj}(x) - f_{Lk}(x) \\ &= \langle (W_L)_{:j} - (W_L)_{:k}, f_{L-1}(x)\rangle + (b_L)_j - (b_L)_k. \end{aligned}$$

Let us define the set

$$V_j = \Big\{y \;\Big|\; \langle (W_L)_{:j} - (W_L)_{:k}, y\rangle > (b_L)_k - (b_L)_j, \forall k \neq j\Big\}$$

then it holds $C_j = \{x \in \mathbb{R}^d \mid f_{L-1}(x) \in V_j\} = f_{L-1}^{-1}(V_j)$. If V_j is an empty set then we are done, otherwise one observes that V_j is the intersection of a finite number of open half-spaces (or the whole space), which is thus an open and connected set. Moreover, it holds $V_j \cap \hat{\sigma}_{L-1}(\mathbb{R}) = V_j$, where $\hat{\sigma}_{L-1}$ is defined as in (1). It follows from Proposition 3.7 that C_j must be an open set as it is the pre-image of the open set V_j under the continuous mapping f_{L-1}. To show that C_j is a connected set, one first observes that

$$f_{L-1} = \hat{\sigma}_{L-1} \circ h_{L-1} \circ \hat{\sigma}_{L-2} \circ h_{L-2} \ldots \circ \hat{\sigma}_1 \circ h_1$$

where $h_k : \mathbb{R}^{n_{k-1}} \times \mathbb{R}^{n_k}$ is an affine mapping between layer $k-1$ and layer k defined as $h_k(x) = W_k^T x + b_k$ for every $1 \leq k \leq L-1, x \in \mathbb{R}^{n_{k-1}}$, and $\hat{\sigma}_k : \mathbb{R}^{n_k} \to \mathbb{R}^{n_k}$ is the activation mapping of layer k defined as in (1). By Proposition 3.6 it holds that

$$f_{L-1}^{-1}(V_j) = (h_1^{-1} \circ \hat{\sigma}_1^{-1} \circ \ldots \circ h_{L-1}^{-1} \circ \hat{\sigma}_{L-1}^{-1})(V_j)$$

Since $\sigma_k : \mathbb{R} \to \mathbb{R}$ is a continuous bijection by our assumption, it follows that $\hat{\sigma}_k : \mathbb{R}^{n_k} \to \mathbb{R}^{n_k}$ is also a continuous bijection. Moreover, it holds that W_k has full rank and $n_{k-1} \geq n_k$ for every $1 \leq k \leq L-1$ and V_j is a connected set. Thus one can apply Lemma 3.9 subsequently for the composed functions $(\hat{\sigma}_k \circ h_k)$ for every $k = L-1, L-2, \ldots, 1$ and obtains that $C_j = f_{L-1}^{-1}(V_j)$ is a connected set. Thus C_j is an open and connected set for every $1 \leq j \leq m$. □

The next theorem holds just for networks with one hidden layer but allows general activation functions which are continuous and strictly monotonically increasing, that is leaky ReLU, ELU, softplus or sigmoid activation functions. Again the decision regions are connected if the hidden layer has maximal width smaller than $d+1$.

Theorem 3.11 *Let the one hidden layer network satisfy $d = n_0 \geq n_1$ and let $\sigma_1 : \mathbb{R} \to \mathbb{R}$ be a continuous, strictly monotonically increasing function and the hidden layer's weight matrix W_1 has full rank. Then every decision region C_j is an open connected subset of $\mathbb{R}^d$ for every $1 \leq j \leq m$.*

Proof: We note that in the proof of Theorem 3.10 the V_j is a finite intersection of open half-spaces and thus a convex set. Moreover, $\hat{\sigma}_1(\mathbb{R}^{n_1})$ is an open convex set (it is just an axis-aligned open box), as σ_1 is strictly monotonically increasing. Thus

$$\begin{aligned} C_j &= \{x \in \mathbb{R}^d \mid f_1(x) \in V_j \cap \hat{\sigma}_1(\mathbb{R}^{n_1})\} \\ &= f_1^{-1}\big(V_j \cap \hat{\sigma}_1(\mathbb{R}^{n_1})\big). \end{aligned}$$

As both sets are open convex sets, the intersection $V_j \cap \hat{\sigma}_1(\mathbb{R}^{n_1})$ is again convex and open as well. Thus $V_j \cap \hat{\sigma}_1(\mathbb{R}^{n_1})$ is a connected set. The rest of the argument follows then by using Lemma 3.9, noting that by Proposition 3.8 $\hat{\sigma}_1^{-1} : \hat{\sigma}_1(\mathbb{R}^{n_1}) \to \mathbb{R}^{n_1}$ is a continuous mapping. □

Note that Theorem 3.10 and Theorem 3.11 make no assumption on the structure of all layers in the network. Thus they can be applied to neural networks with both fully connected layers and convolutional layers. Moreover, the results hold regardless of how the parameters of the network $(W_l, b_l)_{l=1}^L$ have been attained, trained or otherwise, as long as all the weight matrices of hidden layers have full rank. This is a quite weak condition in practice as the set of low rank matrices has just Lebesgue measure zero. Even if the optimal weight parameters for the data generating distribution would be low rank (we discuss such an example below), then it is very unlikely that the trained weight parameters are low rank, as one has statistical noise by the training sample, "optimization noise" from the usage of stochastic gradient descent (SGD) and its variants and finally in practice one often uses early stopping and thus even if the optimal solution for the training set is low rank, one will not find it.

Theorem 3.10 covers activation functions like leaky ReLU but not sigmoid, ELU or softplus. At the moment it is unclear for us if the result might hold also for the more general class of activation functions treated in Theorem 3.11. The problem is that then in Lemma 3.9 one has to compute the pre-image of $V \cap \hat{\sigma}(\mathbb{R}^n)$. Even though both sets are connected, the intersection of connected sets need not be connected. This is avoided in Theorem 3.11 by using that the initial set V_j and $\hat{\sigma}(\mathbb{R}^{n_{L-1}})$ are both convex and the intersection of convex sets is convex and thus connected.

We show below that the result is tight in the sense that we give an empirical example of a neural network with a single hidden layer of $d+1$ hidden units which produces disconnected regions. Note that our result complements the result of (Hanin & Sellke, 2017), where they show the universal approximation property (for ReLU) only if one considers networks of width at least $d+1$ for arbitrary depth. Theorem 3.10 and Theorem 3.11 indicate that this result could also hold for leaky ReLU as approximation of arbitrary functions implies approximation of arbitrary decisions regions, which clearly requires that one is able to get disconnected decision regions. Taking both results together, it seems rather obvious that as a general guiding principle for the construction of hidden layers in neural networks one should use, at least for the first hidden layer, more units than the input dimension, as it is rather unlikely that the Bayes optimal decision regions are connected. Indeed, if the true decision regions are disconnected then using a network of smaller width than $d+1$ might still perfectly fit the finite training data but since the learned decision regions are connected there exists a path between the true decision regions which then can be used for potential adversarial manipulation. This is discussed in the next section where we show empirical evidence for the existence of such adversarial examples.

4. Illustration and Discussion

In this section we discuss with analytical examples as well as trained networks that the result is tight and the conditions of the theorem cannot be further relaxed. Moreover, we argue that connected decision regions can be problematic as they open up the possibility to generate adversarial examples.

4.1. Why pyramidal structure of the network is necessary to get connected decision regions?

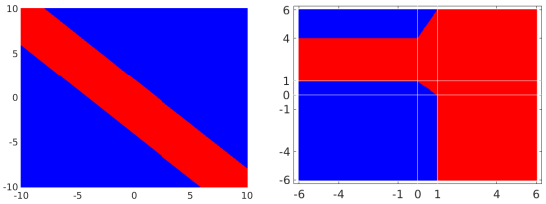


Figure 2. Decision region of the network in (3)(left) and (5)(right).

In Theorem 3.10, if the network does not have pyramidal structure up to the last hidden layer, *i.e.* the condition $d_1 \geq \ldots \geq d_{L-1}$ is not fulfilled, then the statement of the theorem might not hold as the decision regions can be disconnected. We illustrate this via a counter-example below. Let us consider a non-pyramidal network 2-1-2-2 defined as

$$W_3^T \hat{\sigma}_2(W_2^T \hat{\sigma}_1(W_1^T x + b_1) + b_2) + b_3 \tag{3}$$

where $\sigma_1(t) = \sigma_2(t) = \max\{0.5\,t, t\}$, and $W_1 = \begin{bmatrix}1\\1\end{bmatrix}, b_1 = 0, W_2 = \begin{bmatrix}1 & -1\end{bmatrix}, b_2 = \begin{bmatrix}0\\0\end{bmatrix}, W_3 = \begin{bmatrix}2 & 1\\3 & 2\end{bmatrix}, b_3 = \begin{bmatrix}0\\1\end{bmatrix}$. Then one can check that this network has (see appendix for the full derivation) $C_1 = \{x \in \mathbb{R}^2 \mid x_1 + x_2 - 2 > 0 \text{ and } x_1 + x_2 + 4 < 0\}$, which is a disconnected set as illustrated in Figure 2.

4.2. Why full rank of the weight matrices is necessary to get connected decision regions?

Similar to Section 4.1, we show that if the weight matrices of hidden layers are not full rank while the other conditions are still satisfied, then the decision regions can be disconnected. The reason is simply that low rank matrices, in particular in the first layer, reduce the effective dimension of the input. We illustrate this effect with a small analytical example and then argue that nevertheless in practice it is extremely difficult to get low rank weight matrices.

Suppose one has a two-class classification problem on $\mathbb{R}^2$ (see Figure 3) with equal class probabilities $P(\text{red}) = P(\text{blue})$, and the conditional distribution is given as

$$p(x_1, x_2|\text{blue}) = \frac{1}{2},\ \forall\, x_1 \in [-2,-1] \cup [1,2], x_2 \in [-\frac{1}{2}, \frac{1}{2}]$$
$$p(x_1, x_2|\text{red}) = 1,\ \forall\, x_1 \in [-1,1], x_2 \in [-\frac{1}{2}, \frac{1}{2}]. \tag{4}$$

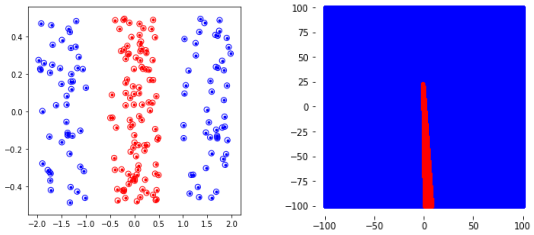


Figure 3. Left: the training set corresponding to the distribution in (4). Right: the decision regions of a trained classifier, which are connected as the learned weight matrix W_1 has full rank.

Note that the Bayes optimal decision region for class blue is disconnected. Moreover, it is easy to verify that a one hidden layer network with leaky ReLU $\sigma(t) = \max\{t, \alpha t\}$ for $0 < \alpha < 1$ can perfectly fit the data with

$$W_1^T = \begin{pmatrix}1 & 0\\-1 & 0\end{pmatrix}, b_1 = \begin{pmatrix}-1\\-1\end{pmatrix}, W_2^T = \begin{pmatrix}1 & 1\\0 & 0\end{pmatrix}, b_2 = \begin{pmatrix}0\\-2\alpha\end{pmatrix}$$

Note that W_1 has low rank. Suppose that the first output unit corresponds to the blue class and second output unit corresponds to the red class. Then it holds $(f_2)_{\text{red}}(x_1, x_2) = -2\alpha$, $(f_2)_{\text{blue}}(x_1, x_2) = \max\{x_1 - 1, \alpha(x_1 - 1)\} + \max\{-(x_1 + 1), -\alpha(x_1 + 1)\}$ and thus

$$(f_2)_{\text{blue}}(x_1, x_2) = \begin{cases}(1-\alpha)x_1 - (1+\alpha) & x_1 \geq 1\\ -2\alpha & -1 \leq x_1 \leq 1\\ -(1-\alpha)x_1 - (1+\alpha) & x_1 \leq -1\end{cases}$$

which implies that $(f_2)_{\text{blue}}(x_1, x_2) > (f_2)_{\text{red}}(x_1, x_2)$ for every $x_1 \in (-\infty, -1) \cup (1, +\infty)$ and thus the decision region for class blue has two disconnected decision regions. This implies that Theorems 3.10 and 3.11 do indeed not hold if the weight matrices do not have full rank. Nevertheless in practice, it is unlikely that one will get such low rank weight matrices, which we illustrate in Figure 3 that the decision regions of the trained classifier has indeed connected decision regions. This is due to statistical noise in the training set as well as through the noise in the optimization procedure (SGD) and the common practice of early stopping in training of neural networks.

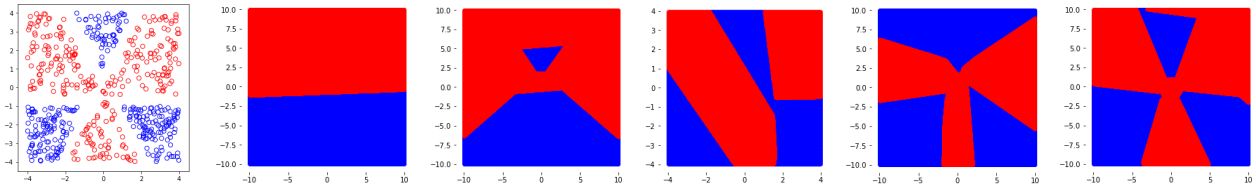
4.3. Does the result hold for ReLU activation function?

As the conditions of Theorem 3.10 are not fulfilled for ReLU, one might ask whether the decision regions of a ReLU network with pyramidal structure and full rank weight matrices can be potentially disconnected. We show that this is indeed possible via the following example. Let a two hidden layer network (2-2-2-2) be defined as

$$W_3^T \hat{\sigma}_2(W_2^T \hat{\sigma}_1(W_1^T x + b_1) + b_2) + b_3 \tag{5}$$

where $\sigma_1(t) = \sigma_2(t) = \max\{t, 0\}$ and

$$W_1^T = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}, W_2^T = \frac{\sqrt{2}}{2}\begin{bmatrix}1 & 1\\-1 & 1\end{bmatrix}, W_3^T = \begin{bmatrix}-1 & 0\\0 & -3\end{bmatrix},$$



(a) Training data (b) $n_1 = 2(122)$ (c) $n_1 = 3(72)$ (d) $n_1 = 4(45)$ (e) $n_1 = 7(12)$ (f) $n_1 = 50(0)$

Figure 4. Decision region of a one hidden layer network trained with SGD for varying number of hidden units for the toy example given in (a). As shown by Theorem 3.10 the decision region for $n_1 = d = 2$ is connected, however already for $n_1 > d = 2$ one gets disconnected decision regions which shows that Theorem 3.10 is tight. The numbers in bracket show the number of misclassified training points.

and $b_1 = [0,0]^T, b_2 = \frac{1}{\sqrt{2}}[\sqrt{2}-1,-3]^T, b_3 = [1,0]^T$. Then one can derive the decision region for the first class as (see appendix for the full derivation)

$$C_1 = \left\{x \in \mathbb{R}^2 \;\middle|\; x_1 < 1,\ x_2 < 1,\ x_1 + x_2 < 1\right\} \cup \left\{x \in \mathbb{R}^2 \;\middle|\; x_2 > 4,\ 2x_1 - x_2 + 4 < 0\right\}$$

which is a disconnected set as illustrated in Figure 2.

Finally, one notes in this example that except for the activation function, all the other conditions of Theorem 3.10 are still satisfied, that is, the network has pyramidal structure (2-2-2-2) and all the weight matrices $(W_l)_{l=1}^2$ have full rank by our construction. Thus the statement of Theorem 3.10, at least under current form, does not hold for ReLU.

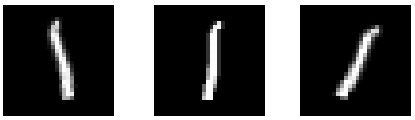
4.4. The theorems are tight: disconnected decision regions for width $d+1$

We consider a binary classification task in $\mathbb{R}^2$ where the data points are generated so that the blue class has disconnected components on the square $[-4,4] \times [-4,4]$, see Figure 4 (a) for an illustration. We use a one hidden layer network with varying number of hidden units, two output units, leaky ReLU activation function and cross-entropy loss. We then train this network by using SGD with momentum for 1000 epochs and learning rate 0.1 and reduce the it by a factor of 2 after every 50 epochs. For all the attempts with different starting points that we have done in our experiment, the resulting weight matrices always have full rank.

We show the training error and the decision regions of trained network in Figure 4. The grid size in each case of Figure 4 has been manually chosen so that one can see clearly the connected/disconnected components in the decision regions. First, we observe that for two hidden units ($n_1 = 2$), the network satisfies the condition of Theorem 3.10 and thus can only learn connected regions, which one can also clearly see in the figure, where one basically gets a linear separator. However, for three hidden units ($n_1 = 3$), one can see that the network can produce disconnected decision regions, which shows that both our Theorems 3.10 and 3.11 are tight, in the sense that width $d+1$ is already sufficient to produce disconnected components, whereas the results say that for width less than $d+1$ the decision regions have to be connected. As the number of hidden units increases, we observe that the network produces more easily disconnected decision regions as expected.

4.5. Relation to adversarial manipulation

We use a single image of digit 1 from the MNIST dataset to create a new artificial dataset where the underlying data generation probability measure has a similar one-dimensional structure as in (4) but now embedded in the pixel space $\mathbb{R}^{28\times 28}$. This is achieved by using rotation as the one-dimensional degree of freedom. We generate 2000 training images for each red/blue class by rotating the chosen digit 1 with angles ranging from $[-5°, 5°]$ for the read class, and $[-20°, -15°] \cup [15°, 20°]$ for the blue class, see Figure 5.



$[-20°, -15°]$ $[-5°, 5°]$ $[15°, 20°]$

Figure 5. Training examples for our binary digit-1 dataset. The color (red/blue) denotes the class of corresponding example.

Note that this is a binary classification task where the dataset has just one effective degree of freedom and the Bayes optimal decision regions are disconnected. We train a one hidden layer network with 784 hidden units which is equal to the input dimension and leaky ReLU as activation function with $\alpha = 0.1$. The training error is zero and the resulting weight matrices have full rank, thus the conditions of Theorem 3.10 are satisfied and the decision region of class blue should be connected even though the Bayes optimal decision region is disconnected. This can only happen by establishing a connection around the other red class. We test this by sampling a source image from the $[-20°, -15°]$ part of the blue class and a target image from the other part $[15°, 20°]$. Next, we generate an adversarial image [1] from the red class using the one step target class method (Kurakin

[1] This is essentially a small perturbation of an image from the red class which is classified as blue class

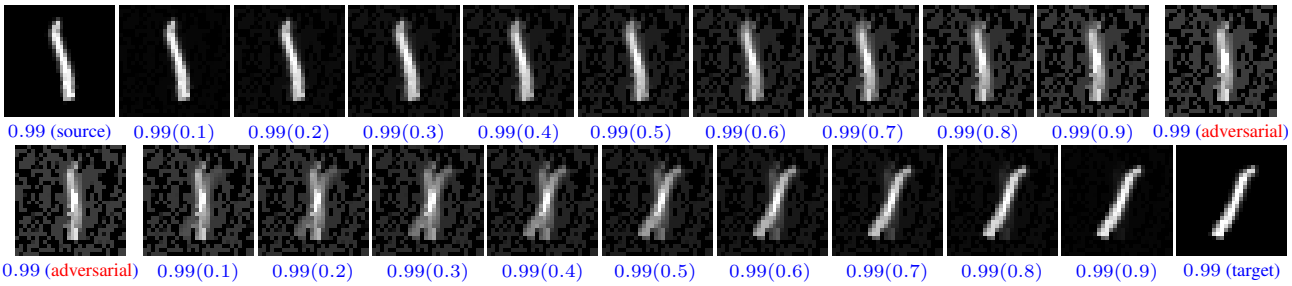


Figure 6. Digit-1 dataset (2 output classes): The trajectory from source image to adversarial image (top row) parameterized by λ (numbers inside brackets), and from adversarial image to target image (second row). Each number outside bracket shows the confidence that the corresponding image was predicted to be in blue class. The image with red caption can be seen as an adversarial image of the red class.

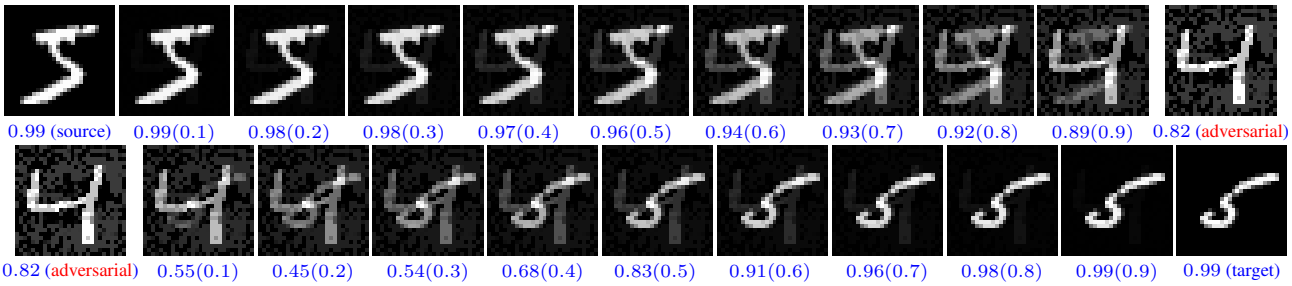


Figure 7. MNIST dataset (10 output classes): The trajectory from source image to adversarial image (top row) parameterized by λ (numbers inside brackets), and from adversarial image to target image (second row). Each number outside bracket shows the confidence that the corresponding image was predicted to be in blue class (digit 5) out of 10 classes.

et al., 2016; 2017) and consider the path between the source image to the adversarial image and subsequently from the adversarial image to the target one. For each path, we simply consider the line segment $\lambda s + (1-\lambda)t$ for $\lambda \in [0,1]$ between the two endpoint images s and t and sample it very densely by dividing $[0,1]$ into 10^4 equidistant parts. Figure 6 shows the complete path from the source image to the target image where the color indicates that all the intermediate images are classified as blue with high confidence (note that we turned the output of the network into probabilities by using the softmax function). Moreover, the intermediate images from Figure 6 look very much like images from the red class thus could be seen as adversarial samples for the red class. The point we want to make here is that one might think that in order to avoid adversarial manipulation the solution is to use a simple classifier of low capacity. We think that rather the opposite is true in the sense that only if the classifier is rich enough to model the true underlying data generating distribution it will be able to model the true decision boundaries. In particular, the classifier should be able to realize disconnected decision regions in order to avoid paths through the input space which connect different disconnected regions of the Bayes optimal classifier. Now one could argue that the problem of our synthetic example is that the corresponding digits obviously do not fill the whole image space, nevertheless the classifier has to do a prediction for all possible images. This could be handled by introducing a background class, but then it would be even more important that the classifier can produce disconnected decision regions which naturally requires a minimal width of $d+1$ of the network.

In Figure 7, we show another similar experiment on MNIST dataset, but now for all the 10 image classes. We train a network with 200 hidden units, leaky ReLU and softmax cross-entropy loss to zero training error. Once again, one can see that there exists a continuous path that connects two different-looking images of digit 5 (blue class) where every image along this path is classified as blue class with high confidence. Moreover this path goes through a pre-constructed adversarial image of the red class (digit 4).

5. Conclusion

We have shown that deep neural networks (with a certain class of activation functions) need to have in general width larger than the input dimension in order to learn disconnected decision regions. It remains an open problem if our current requirement $\sigma(\mathbb{R}) = \mathbb{R}$ can be removed. While our result does not resolve the question how to choose the network architecture in practice, it provides at least a guideline how to choose the width of the network. Moreover, our result and experiments show that too narrow networks produce high confidence predictions on a path connecting the true disconnected decision regions which could be used to attack these networks using adversarial manipulation.

Acknowledgements

The authors would like to thank the reviewers for their helpful comments on the paper and Francesco Croce for bringing up a counter-example for the ReLU activation function.

References

Apostol, T. M. *Mathematical analysis*. Addison Wesley, Reading, Massachusetts, 1974.

Arora, R., Basu, A., Mianjy, P., and Mukherjee, A. Understanding deep neural networks with rectified linear units. In *ICLR*, 2018.

Charisopoulos, V. and Maragos, P. A tropical approach to neural networks with piecewise linear activations, 2018. arXiv:1805.08749.

Clevert, D., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). In *ICLR*, 2016.

Cohen, N. and Shashua, A. Convolutional rectifier networks as generalized tensor decompositions. In *ICML*, 2016.

Cohen, N., Sharir, O., and Shashua, A. On the expressive power of deep learning: A tensor analysis. In *COLT*, 2016.

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

Delalleau, O. and Bengio, Y. Shallow vs. deep sum-product networks. In *NIPS*, 2011.

Eldan, R. and Shamir, O. The power of depth for feedforward neural networks. In *COLT*, 2016.

Fawzi, A., Dezfooli, S. M. M., Frossard, P., and Soatto, S. Classification regions of deep neural networks, 2017. arXiv:1705.09552.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

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

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Grishick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In *ACM Int. Conference on Multimedia*, 2014.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial machine learning at scale. In *ICLR*, 2016.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. In *ICLR Workshop*, 2017.

Leshno, M., Lin, Y., Pinkus, A., and Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6:861–867, 1993.

Liang, S. and Srikant, R. Why deep neural networks for function approximation? In *ICLR*, 2017.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. In *NIPS*, 2017.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013.

Makhoul, J., El-Jaroudi, A., and Schwartz, R. Formation of disconnected decision regions with a single hidden layer. *IJCNN*, pp. 455–460, 1989.

Makhoul, J., El-Jaroudi, A., and Schwartz, R. Partitioning capbabilities of two-layer neural networks. *IEEE Trans. Signal Processing*, 39(6):1435–1440, 1990.

Mhaskar, H. and Poggio, T. Deep vs. shallow networks : An approximation theory perspective, 2016. arXiv:1608.03287.

Montufar, G., Pascanu, R., Cho, K., and Bengio, Y. On the number of linear regions of deep neural networks. In *NIPS*, 2014.

Nguyen, Q. and Hein, M. The loss surface of deep and wide neural networks. In *ICML*, 2017.

Nguyen, Q. and Hein, M. Optimization landscape and expressivity of deep cnns. In *ICML*, 2018.

Pascanu, R., Montufar, G., and Bengio, Y. On the number of response regions of deep feedforward networks with piecewise linear activations. In *ICLR*, 2014.

Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., and Liao, Q. Why and when can deep – but not shallow – networks avoid the curse of dimensionality: a review, 2016. arXiv:1611.00740.

Poon, H. and Domingos, P. Sum-product networks: A new deep architecture. In *ICCV Workshop*, 2011.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Sohl-Dickstein, J. On the expressive power of deep neural networks. In *ICML*, 2017.

Safran, I. and Shamir, O. Depth-width tradeoffs in approximating natural functions with neural networks. In *ICML*, 2017.

Serra, T., Tjandraatmadja, C., and Ramalingam, S. Bounding and counting linear regions of deep neural networks, 2018. arXiv:1711.02114.

Telgarsky, M. Representation benefits of deep feedforward networks, 2015. arXiv:1509.08101v2.

Telgarsky, M. Benefits of depth in neural networks. In *COLT*, 2016.

Yarotsky, D. Error bounds for approximations with deep relu networks, 2016. arXiv:1610.01145.