Quantifying the generalization error in deep learning in terms of data distribution and neural network smoothness

Pengzhan Jin[a,b,1], Lu Lu[c,1], Yifa Tang[a,b], George Em Karniadakis[c,*]

[a] *LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China*
[b] *School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China*
[c] *Division of Applied Mathematics, Brown University, Providence, RI 02912, USA*

Abstract

The accuracy of deep learning, i.e., deep neural networks, can be characterized by dividing the total error into three main types: approximation error, optimization error, and generalization error. Whereas there are some satisfactory answers to the problems of approximation and optimization, much less is known about the theory of generalization. Most existing theoretical works for generalization fail to explain the performance of neural networks in practice. To derive a meaningful bound, we study the generalization error of neural networks for classification problems in terms of data distribution and neural network smoothness. We introduce the *cover complexity* (CC) to measure the difficulty of learning a data set and the *inverse of modules of continuity* to quantify neural network smoothness. A quantitative bound for expected accuracy/error is derived by considering both the CC and neural network smoothness. We validate our theoretical results by several data sets of images. The numerical results verify that the expected error of trained networks scaled with the square root of the number of classes has a linear relationship with respect to the CC. In addition, we observe a clear consistency between test loss and neural network smoothness during the training process.

Keywords: Neural networks, Generalization error, Learnability, Data distribution, Cover complexity, Neural network smoothness

1. Introduction

In the last 15 years, deep learning, i.e., deep neural networks (NNs), has been used very effectively in diverse applications, such as image classification (Krizhevsky et al., 2012), natural language processes (Maas et al., 2013), and game playing (Silver et al., 2016). Despite this remarkable success, our theoretical understanding of deep learning is lagging behind. The accuracy of NNs can be characterized by dividing the expected error into three main types: approximation (or called expressivity), optimization, and generalization (Bottou and Bousquet, 2008; Bottou, 2010), see Fig. 1. The well-known approximation result was obtained by Cybenko (1989) and Hornik et al. (1989) almost three decades ago stating that feed-forward neural nets can approximate essentially any function. In the past several years, there have been numerous studies that analyze the landscape of the non-convex objective functions, and the optimization process by stochastic gradient descent (SGD) (Lee et al., 2016; Liao and Poggio, 2017; Allen-Zhu et al., 2018b; Du et al., 2018; Lu et al., 2019). Whereas there are some satisfactory answers to the problems of approximation and optimization, much less is known about the theory of generalization, which is the focus of this study.

The classical analysis of generalization is based on controlling the complexity of the function class, i.e., model complex-

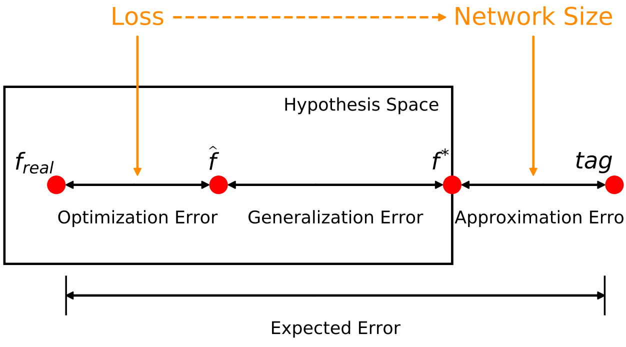


Figure 1: Illustration of approximation error, optimization error, and generalization error. The total error consists of these three errors. *tag* is the target true function, f^* is the best function close to *tag* in the hypothesis space, $\hat{f}$ is a neural network whose loss is at a global minimum, and f_{real} is the function by training a neural network. Thus, the optimization error is correlated with the loss value, while the approximation error depends on the network size. In addition, a small loss requires a large network size, which in turn leads to a small approximation error. Assuming a sufficient small loss, the expected error mainly depends on the generalization error.

[*] Corresponding author at: 182 George Street, Providence, RI 02912, USA.
Email address: george_karniadakis@brown.edu (George Em Karniadakis)

[1] Pengzhan Jin and Lu Lu contributed equally to this work.

ity, by managing the bias-variance trade-off (Friedman et al., 2001). However, this type of analysis is not able to explain the small generalization gap between training and test performance of neural networks learned by SGD in practice, considering the fact that deep neural networks often have far more model parameters than the number of samples they are trained on, and have sufficient capacity to memorize random labels (Neyshabur et al., 2014; Zhang et al., 2016). To explain this phenomenon, several approaches have been recently developed by many researchers. The first approach is characterizing neural networks with some other low "complexity" instead of the traditional Vapnik-Chervonenkis (VC) dimension (Bartlett et al., 2017b) or Rademacher complexity (Bartlett and Mendelson, 2002), such as path-norm (Neyshabur et al., 2015), margin-based bounds (Sokolić et al., 2017; Bartlett et al., 2017a; Neyshabur et al., 2017b), Fisher-Rao norm (Liang et al., 2017), and more (Neyshabur et al., 2019; Wei and Ma, 2019). The second approach is to analyze some good properties of SGD or its variants, including its stability (Hardt et al., 2015; Kuzborskij and Lampert, 2017; Gonen and Shalev-Shwartz, 2017; Chen et al., 2018), robustness (Sokolic et al., 2016; Sokolić et al., 2017), implicit biases/regularization (Poggio et al., 2017; Soudry et al., 2018; Gunasekar et al., 2018; Nagarajan and Kolter, 2019b), and the structural properties (e.g., sharpness) of the obtained minimizers (Keskar et al., 2016; Dinh et al., 2017; Zhang et al., 2018). The third approach relies on overparameterization, e.g., sufficiently overparameterized networks can learn the ground truth with a small generalization error using SGD from random initialization (Li and Liang, 2018; Allen-Zhu et al., 2018a; Arora et al., 2019; Cao and Gu, 2019). There are also other approaches, such as compression (Arora et al., 2018; Baykal et al., 2018; Zhou et al., 2018; Cheng et al., 2018), Fourier analysis (Rahaman et al., 2018; Xu et al., 2019), "double descent" risk curve (Belkin et al., 2018), and PAC-Bayesian framework (Neyshabur et al., 2017b; Nagarajan and Kolter, 2019a).

However, most theoretical bounds fail to explain the performance of neural networks in practice (Neyshabur et al., 2017a; Arora et al., 2018). To get non-vacuous and tight enough bounds to be practically meaningful, some problem-specific factors should be taken into consideration, such as the easiness of the data (i.e., data-dependent analysis) (Dziugaite and Roy, 2017; Kawaguchi et al., 2017), or properties of the trained neural networks (Sokolić et al., 2017; Arora et al., 2018; Wei and Ma, 2019). In this study, to achieve a practically meaningful bound, our analysis relies on the data distribution and the smoothness of the trained neural network. The analysis proposed in this study provides guarantees on the generalization error, and theoretical insights to guide the practical application.

As shown in Fig. 1, the optimization error is correlated with the loss value, while the approximation error depends on the network size. In addition, a small loss requires a sufficient approximation ability, i.e., a large network size, which in turn leads to a small approximation error. If we assume a sufficient small loss, which is indeed true in practice, then the expected error mainly depends on the generalization error. Hence, we study the expected error/accuracy directly. In particular, we propose a mathematical framework to analyze the expected accuracy of neural networks for classification problems. We introduce the concepts of *total cover (TC)*, *self cover (SC)*, *mutual cover (MC)* and *cover difference (CD)* to represent the data distribution, and then we use the concept of *cover complexity (CC)* as a measure of the complexity of classification problems. On the other hand, the smoothness of a neural network f is characterized by the *inverse of the modulus of continuity* δ_f. Because δ_f is not tractable in general, we propose an estimation using the 2-norm of weight matrices of the neural network. The main terminologies are illustrated in Fig 2. By combining the properties of data distribution and the smoothness of neural networks, we derive a lower bound for the expected accuracy, i.e., an upper bound for the expected classification error.

Subsequently, we test our theoretical bounds on several data sets, including MNIST (LeCun et al., 1998), CIFAR-10 (Krizhevsky and Hinton, 2009), CIFAR-100 (Krizhevsky and Hinton, 2009), COIL-20 (Nene et al., 1996b), COIL-100 (Nene et al., 1996a), and SVHN (Netzer et al., 2011). Our numerical results not only confirm our theoretical bounds, but also provide insights into the optimization process and the learnability of neural networks. In particular, we find that:

- The best accuracy that can be achieved by fully-connected networks is approximately linear with respect to the cover complexity of the data set.
- The trend of the expected accuracy is consistent with the smoothness of the neural network, which provides a new "early stopping" strategy by monitoring the smoothness of the neural network.

The paper is organized as follows. After setting up notation and terminology in Section 2, we present the main theoretical results of the accuracy based on data distribution and neural network smoothness in Section 3. In Section 4, we provide the numerical results for several data sets. In Section 5 we include a discussion, and in Section 6 we summarize our findings.

2. Preliminaries

Before giving the main results, we introduce the necessary notation and terminology. Without loss of generality, we assume that the space we need to classify is

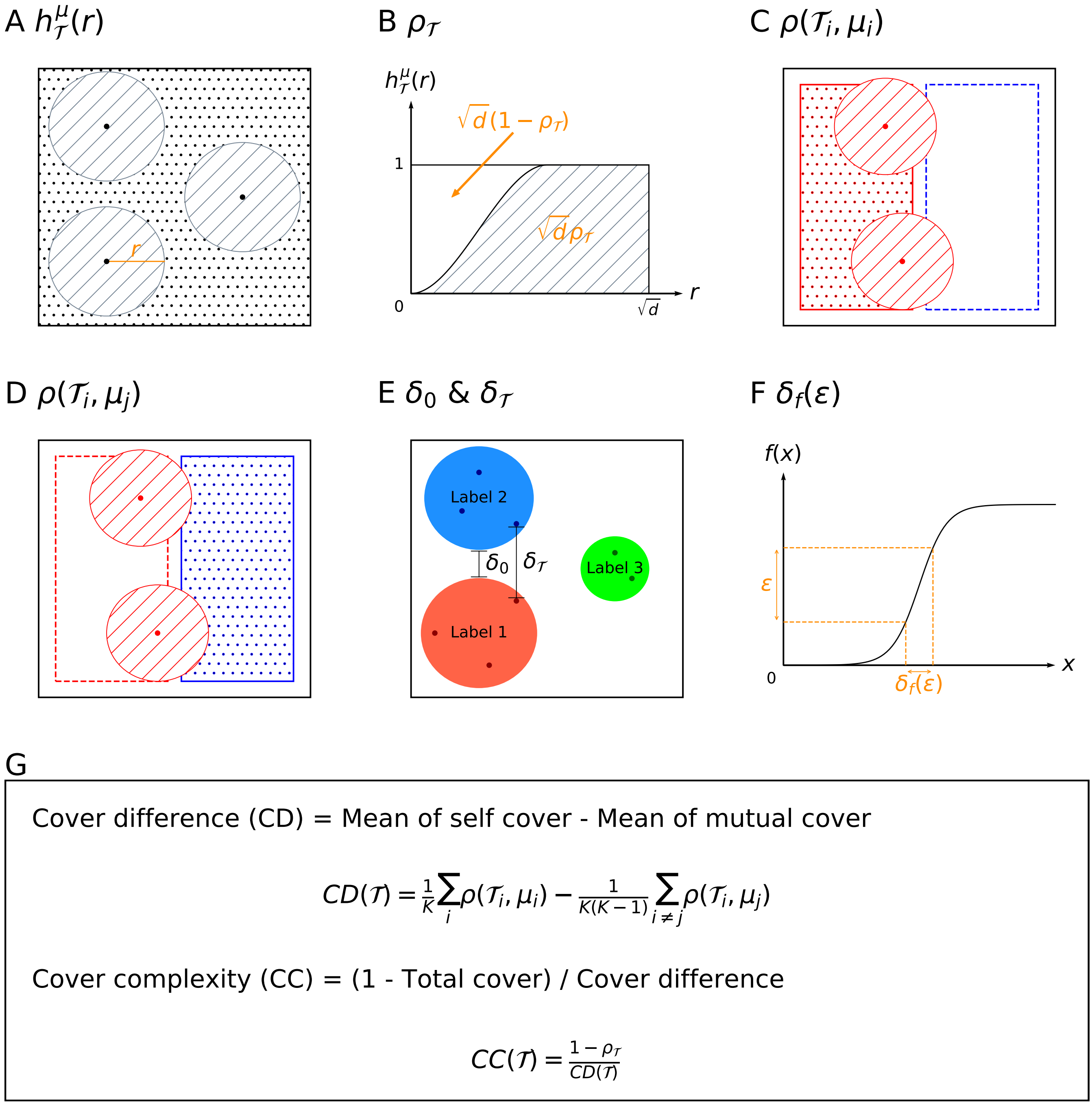
$$D = [0, 1]^d,$$

where d is the dimensionality, and the points in this space are classified into K categories, i.e., there are K labels $\{1, 2, \cdots, K\}$. Let P be the probability density function by which samples are drawn from D, and we have

$$\int_D P(x)dx = 1.$$

2.1. Ideal label function

For the problem setup, we assume that every sample has at least one true label, and one sample may have multiple true labels. Taking image classification as an example, each image has



A $h_{\mathcal{T}}^{\mu}(r)$
r
B $\rho_{\mathcal{T}}$
$h_{\mathcal{T}}^{\mu}(r)$
$\sqrt{d}(1-\rho_{\mathcal{T}})$
1
$\sqrt{d}\rho_{\mathcal{T}}$
0
$\sqrt{d}$
r
C $\rho(\mathcal{T}_i, \mu_i)$
D $\rho(\mathcal{T}_i, \mu_j)$
E δ_0 & $\delta_{\mathcal{T}}$
Label 2
δ_0
$\delta_{\mathcal{T}}$
Label 3
Label 1
F $\delta_f(\varepsilon)$
$f(x)$
ε
0
$\delta_f(\varepsilon)$
x
G

Cover difference (CD) = Mean of self cover - Mean of mutual cover

$$CD(\mathcal{T}) = \frac{1}{K}\sum_i \rho(\mathcal{T}_i, \mu_i) - \frac{1}{K(K-1)}\sum_{i \neq j} \rho(\mathcal{T}_i, \mu_j)$$

Cover complexity (CC) = (1 - Total cover) / Cover difference

$$CC(\mathcal{T}) = \frac{1-\rho_{\mathcal{T}}}{CD(\mathcal{T})}$$

Figure 2: Illustrations of the main definitions and terminologies. (**A**) $h_{\mathcal{T}}^{\mu}(r)$ in Eq. (2): the probability of the neighborhood of the training set with radius of r. (**B**) $\rho_{\mathcal{T}}$ in Definition 2.4: total cover of training set $\mathcal{T}$. (**C**) $\rho(\mathcal{T}_i, \mu_i)$ in Definition 2.4: self cover of training set $\mathcal{T}$. (**D**) $\rho(\mathcal{T}_i, \mu_j)$ in Definition 2.4: mutual cover of training set $\mathcal{T}$. (**E**) δ_0 in Proposition 2.1: separation gap; $\delta_{\mathcal{T}}$ in Theorem 3.2: empirical separation gap. (**F**) δ_f in Definition 2.9: the inverse of modulus of continuity. (**G**): $CD(\mathcal{T})$ and $CC(\mathcal{T})$ in Definition 2.4: cover difference and cover complexity, respectively. Note that d is the dimensionality of the sample space D and K is the number of categories.

at least one correct label. A fuzzy image or an image with more than one object it may have multiple possible correct labels, and as long as the prediction is one of these labels, we consider the prediction to be correct.

It is intuitive that when two samples are close enough, they should have similar labels, which means that the ideal label function should be continuous. Continuity of a mapping depends on the topology of both domain and image space. For the domain of the ideal label function, we choose the standard topology induced by the Euclidean-metric. As for the topology of the image space, we define it as follows. We first define the label set and the topology on it.

Definition 2.1 (Topology)**.** *Let*

$$T = 2^{\{1,2,\cdots,K\}} \backslash \{\varnothing\}$$

be the label set. Define the topology on T to be

$$\tau_T = \{U \in 2^T | U = \bigcup_{k \in I} U_k,\ U_k = \{V \in T | k \subseteq V\},\ I \subseteq T\}$$

and thus (T, τ_T) constitutes a topological space.

In this definition, k is an element of T, and U_k is a set comprised of all elements containing k from T. All sets like U_k constitute a topological base for T, and then τ_T is the topology generated by this base, see Appendix A for an example. Next we give the definition of the ideal label function according to this topological space.

Definition 2.2 (Ideal label function)**.** *Define an ideal label function, i.e., an ideal classifier, as*

$$\begin{array}{cccc} tag: & D & \rightarrow & T, \\ & (D, \tau_0) & & (T, \tau_T) \end{array}$$

where τ_0 is the Euclidean-metric topology. Then tag is a continuous function (i.e., $\forall\ U \in \tau_T,\ tag^{-1}(U) \in \tau_0$). Moreover, continuity holds if and only if

$$\forall k \in tag(x),\ \exists \delta > 0,\ s.t.\ k \in tag(y) \text{ when } \|y - x\| < \delta. \qquad (1)$$

Eq. (1) means that two neighboring points would have some common labels. Based on the topological space defined above, it is easy to show that Eq. (1) is equivalent to continuity. The reason why we consider a multi-label setup for classification problems is that it induces the continuity property in Eq. (1), which does not exist in the setup of a single label set. In addition, the multi-label setup introduces a smooth transition, i.e., a buffer domain, between two domains of different labels, while the transition is sharp in the single label setup. In the following proposition, we show that if two samples are close enough, they must share at least one common label.

Proposition 2.1 (Separation gap)**.** *$\exists \delta > 0,\ s.t.\ tag(x) \cap tag(y) \neq \varnothing$ when $\|x - y\| < \delta$. We denote the supremum of δ as the separation gap δ_0, which is used in the sequel.*

Proof. The proof can be found in Appendix B. □

To understand the geometric interpretation of δ_0, we consider the following special case: the label of each sample is either a single label set, such as $\{1\}$, or the full label set $\{1, 2, \cdots, K\}$ if it is not uniquely identifiable.

Proposition 2.2 (Geometric interpretation of separation gap)**.** *If the label of each sample is either a single label set or the full label set $\{1, 2, \cdots, K\}$, then δ_0 is the smallest distance between two different single label points, i.e.,*

$$\delta_0 = \inf\{\ \|x - y\|\ |\ tag(x) \neq tag(y), and$$
$$tag(x), tag(y) \in \{\{1\}, \{2\}, \cdots, \{K\}\}\ \}.$$

Proof. The proof can be found in Appendix C. □

2.2. Cover complexity of data set

In this subsection, we introduce a quantity to measure the difficulty of learning a training data set

$$\mathcal{T} = \{x_1, x_2, \ldots, x_n\} \subseteq D.$$

First, we give some notations and propositions.

Definition 2.3. *Denote the probability measure on D as μ, that is, for a measurable set $A \subseteq D$, we have*

$$\mu(A) := \int_A P(x) dx.$$

$\mu(A)$ is the probability of a random sample falling in A. Then the probability of the neighborhood of the training set with radius of r is

$$h^{\mu}_{\mathcal{T}}(r) := \mu\left(\bigcup_{x_i \in \mathcal{T}} B(x_i, r) \bigcap D\right), \qquad (2)$$

where $B(x_i, r)$ is the open ball centered at x_i with radius of r, see Fig. 2A. Obviously, $h^{\mu}_{\mathcal{T}}(r)$ is a monotone non-decreasing continuous function, $h^{\mu}_{\mathcal{T}}(0) = 0$, and $h^{\mu}_{\mathcal{T}}(r) = 1$ when $r \geq \sqrt{d}$, see Fig. 2B. To represent the global behavior of $h^{\mu}_{\mathcal{T}}(r)$, we use the integral of $h^{\mu}_{\mathcal{T}}(r)$ with respect to r:

$$\rho(\mathcal{T}, \mu) := \frac{1}{\sqrt{d}} \int_0^{\sqrt{d}} h^{\mu}_{\mathcal{T}}(r) dr.$$

$\rho(\mathcal{T}, \mu)$ considers both the number of data points, but also the probability distribution of the space. A larger $\rho(\mathcal{T}, \mu)$ means a larger number of data points and also that the probability distribution is more concentrated around $\mathcal{T}$, which we call the "coverability" of $\mathcal{T}$. We can increase $\rho(\mathcal{T}, \mu)$ by adding more data points or redistribute their locations. Next, we show the formal definition for the "coverability".

Definition 2.4 (Coverability)**.** *Let $\mathcal{T}$ be a data set from a domain D with probability measure μ. We define the following for the coverability of $\mathcal{T}$.*

(i) The total cover (TC) is

$$\rho_{\mathcal{T}} := \rho(\mathcal{T}, \mu).$$

Thus, $0 < \rho_{\mathcal{T}} < 1$.

(ii) The cover difference (CD) is

$$CD(\mathcal{T}) := \frac{1}{K}\sum_i \rho(\mathcal{T}_i,\mu_i) - \frac{1}{K(K-1)}\sum_{i\neq j}\rho(\mathcal{T}_i,\mu_j),$$

where K is the number of categories, and $\mathcal{T}_i \subset \mathcal{T}$ and μ_i represent the subset and probability measure of the label i, respectively. Here, $\rho(\mathcal{T}_i,\mu_i)$ is called self cover (SC), and $\rho(\mathcal{T}_i,\mu_j)$ is called mutual cover (MC).

(iii) The cover complexity (CC) is

$$CC(\mathcal{T}) := \frac{1-\rho_{\mathcal{T}}}{CD(\mathcal{T})}.$$

Remark 2.1. *The CD is defined as the difference between the mean of SC and the mean of MC, since each category occurs with the same probability ($\sim 1/K$) in the data sets mostly used in practice. If there are some categories occurring more frequently than others, then it is straightforward to extend this definition by using the mean weighted by the probability of each category.*

In image classification, the dimension of the image space is very high, and thus the data points are quite sparse. However, due to the fact that images actually live on a manifold of low dimension, the probability density around $\mathcal{T}$ is actually high, which makes the TC to be meaningful. Then, we derive a lower bound of $h^{\mu}_{\mathcal{T}}(r)$ by $\rho_{\mathcal{T}}$.

Proposition 2.3. *Let $\mathcal{T}$ be a data set. $h^{\mu}_{\mathcal{T}}(r)$ and $\rho_{\mathcal{T}}$ are defined as above. Then we have*

$$h^{\mu}_{\mathcal{T}}(r) > 1 - \frac{\sqrt{d}}{r}(1-\rho_{\mathcal{T}}), \quad 0 < r < \sqrt{d}.$$

Proof. The proof can be found in Appendix D. □

From this proposition, we know that for a fixed r, $h^{\mu}_{\mathcal{T}}(r)$ can be close to 1 when $\rho_{\mathcal{T}}$ is large enough. However, the probability distribution is usually given in practice, and what we can control is the number of samples. The following theorem shows $\rho_{\mathcal{T}}$ can be arbitrary close to 1 when enough samples are available.

Theorem 2.1. *Let $\mathcal{T}$ be a data set of size n drawn from D by P. Then there exists a non-increasing function $\varrho(\epsilon)$ satisfying $\lim\limits_{\epsilon\to 0}\varrho(\epsilon) = 1$, and for any $0 < \eta, \epsilon \leq \frac{1}{2}$, there exists an*

$$m = O\left(\frac{d+1}{\epsilon}\ln\frac{d+1}{\epsilon} + \frac{1}{\epsilon}\ln\frac{1}{\eta}\right),$$

such that

$$\rho_{\mathcal{T}} \geq \varrho(\epsilon)$$

holds with probability at least $1-\eta$ when $n \geq m$.

Proof. The proof and some other results of TC can be found in Appendix E. □

The reason why CD is introduced is that TC does not consider the labels of each data points. However, data points of the same label should be clustered in a good data set. $CD(\mathcal{T})$ is the difference of self cover and mutual cover, which considers the distributions of each label. By normalizing TC with CD, cover complexity $CC(\mathcal{T})$ is able to measure the difficulty of learning a data set. The difficulty of a problem should be translation-independent and scale-independent. It is easy to see that $CC(\mathcal{T})$ is independence of translation, and the following proposition shows that it is also scale-independent.

Proposition 2.4 (Scale independence)**.** *$CC(\mathcal{T})$ is scale-independent, i.e., if all the data points are scaled with same rate, then $CC(\mathcal{T})$ is unchanged.*

Proof. The proof can be found in Appendix F. □

2.3. Setup for accuracy analysis

The setup for accuracy analysis is as follows.

Definition 2.5. *If $f : D \to \mathbb{R}^K$ is a continuous mapping, then the mapping*

$$\hat{f} : x \mapsto \frac{e^{f(x)}}{\sum_i e^{f_i(x)}}$$

is still continuous, where $f_i(x)$ represents the i-th component of $f(x)$. We have $\hat{f}_i(x) > 0$, and $\sum_i \hat{f}_i(x) = 1$. For convenience, we directly consider the case that $f_i(x) > 0$ and $\sum f_i(x) = 1$, and we call such mapping the normalized continuous positive mapping.

Remark 2.2. *A neural network with softmax nonlinear is a normalized continuous positive mapping.*

Different from the accuracy usually used in classification problems, we define a stronger accuracy called c-accuracy as follows.

Definition 2.6 (c-accuracy at x)**.** *Let f be a normalized continuous positive mapping. For $0.5 \leq c < 1$, we state that f is c-accurate at point x if*

$$\exists 1 \leq i_{max} \leq K, \ s.t. \ f_{i_{max}}(x) > c, \ i_{max} \in tag(x).$$

Definition 2.7 (c-accuracy on D)**.** *Let f be a normalized continuous positive mapping. The c-accuracy of f on a sample space D is defined as*

$$p_c := \frac{\mu(H^c_f)}{\mu(D)} = \mu(H^c_f),$$

where $H^c_f := \{x \in D | f \text{ is c-accurate at } x\}$.

Definition 2.8 (c-accuracy on $\mathcal{T}$)**.** *Let f be a normalized continuous positive mapping. The c-accuracy of f on a data set $\mathcal{T}$ is defined as*

$$p^{\mathcal{T}}_c := \frac{\rho_{\widetilde{\mathcal{T}}}}{\rho_{\mathcal{T}}},$$

where $\widetilde{\mathcal{T}} := \{x \in \mathcal{T} | f \text{ is c-accurate at } x\}$, and $\rho_{\widetilde{\mathcal{T}}}$ and $\rho_{\mathcal{T}}$ are the TC of $\widetilde{\mathcal{T}}$ and $\mathcal{T}$, respectively.

We note that the c-accuracy of f on D represents the expected c-accuracy, and the c-accuracy of f on $\mathcal{T}$ represents the empirical c-accuracy.

Finally, we define a non-decreasing function δ_f to describe the smoothness of f.

Definition 2.9 (Smoothness)**.** *$f : D \to \mathbb{R}^K$ is a continuous mapping, then f is uniformly continuous due to the compactness of D, i.e.*

$$\forall \epsilon > 0,\ \exists \delta > 0,\ s.t.\ \|f(x) - f(y)\|_\infty < \epsilon \text{ when } \|x - y\| < \delta.$$

We denote the supremum of δ satisfying the above requirement by $\delta_f(\epsilon)$. It is easy to see that $\delta_f(\epsilon)$ is equal to the inverse of modulus of continuity of f.

For low dimensional problems, we can directly compute δ_f by brute force. However, for high dimensional problems, it is hard to compute δ_f, and thus we give the following lower bound of δ_f for a neural network f:

$$\delta_f(\epsilon) \geq \frac{\epsilon}{Lip(f)} \geq \frac{\epsilon}{\|W_1\|_2 \cdots \|W_l\|_2 \cdot Lip(softmax)}, \tag{3}$$

where $\|W_i\|_2$ is the 2-norm of the weight matrix W_i of the layer i in the neural network f, and $Lip(f)$ and $Lip(softmax)$ represent the Lipschitz coefficients of f and $softmax$, respectively. $Lip(softmax)$ is a constant, and thus is ignored in our numerical examples. We note that although the lower bound of $\delta_f(\epsilon)$ depends exponentially on the neural net depth, $\delta_f(\epsilon)$ itself does not necessarily scale exponentially in the network depth.

3. Lower bound of expected accuracy

In this section, we present a theoretical analysis of the lower bound of expected accuracy as well as an upper bound of expected error.

Proposition 3.1. *Let f be a normalized continuous positive mapping. Suppose that $\mathcal{T}$ is a single label training set, i.e. $tag(\mathcal{T}) \subseteq \{\{1\}, \{2\}, \cdots, \{K\}\}$. For any $0.5 \leq c_2 < c_1 \leq 1$, we have*

$$p_{c_2} > 1 - \frac{\sqrt{d}}{\delta}(1 - p_{c_1}^{\mathcal{T}} \rho_{\mathcal{T}}),$$

where $\delta = \min(\delta_0, \delta_f(c_1 - c_2))$.

Proof. The proof can be found in Appendix G. □

Proposition 3.1 shows that the expected c-accuracy of f can be bounded by the empirical c-accuracy and the TC of training set. We can see that p_{c_2} tends to 1 when $\rho_{\mathcal{T}}$ and $p_{c_1}^{\mathcal{T}}$ tend to 1. Next we derive a bound of the accuracy by taking into account the loss function.

Theorem 3.1 (Lower bound of c-accuracy)**.** *Let f be a normalized continuous positive mapping. Suppose that $\mathcal{T} = \{x_1, \cdots, x_n\}$ is a single label training set, and $tag(x_i) = \{k_i\}$. For any $0.5 \leq c < 1$, if the maximum cross entropy loss*

$$L_f^{max} := \max_{1\leq i\leq n} \ell(f(x_i), k_i) = -\min_{1\leq i\leq n} \ln(f_{k_i}(x_i)) < -\ln c,$$

then we have

$$p_c > 1 - \frac{\sqrt{d}}{\delta}(1 - \rho_{\mathcal{T}}),$$

where ℓ is the cross entropy loss, $\delta = \min(\delta_0, \delta_f(e^{-L_f^{max}} - c))$, and δ_0 is defined in Proposition 2.1.

Proof. The proof can be found in Appendix H. □

Corollary 3.1. *Let f be a normalized continuous positive mapping. Suppose that $\mathcal{T} = \{x_1, \cdots, x_n\}$ is a single label training set, and $tag(x_i) = \{k_i\}$. For any $0.5 \leq c < 1$, if the loss function*

$$L_f := \frac{1}{n} \sum_{1\leq i\leq n} \ell(f(x_i), k_i) = -\frac{1}{n} \sum_{1\leq i\leq n} \ln(f_{k_i}(x_i)) < -\frac{1}{n} \ln c,$$

then we have

$$p_c > 1 - \frac{\sqrt{d}}{\delta}(1 - \rho_{\mathcal{T}}),$$

where ℓ is the cross entropy loss, and $\delta = \min(\delta_0, \delta_f(e^{-nL_f} - c))$.

Proof. The corollary can be obtained from Theorem 3.1 based on the fact $L_f^{max} \leq nL_f < -\ln c$. □

Theorem 3.1 reveals that the expected accuracy is correlated with the total cover $\rho_{\mathcal{T}}$, separation gap δ_0, neural network smoothness δ_f, and loss value L_f^{max}. We will show numerically in Section 4 that $\delta_f(e^{-L_f^{max}} - c)$ increases first and then decreases during the training of neural networks. The following theorem states that the maximum value of $\delta_f(e^{-L_f^{max}} - c)$ is bounded by the empirical separation gap.

Theorem 3.2 (Empirical separation gap)**.** *Let f be a normalized continuous positive mapping. Suppose that $\mathcal{T} = \{x_1, \cdots, x_n\}$ is a single label training set. For any $0.5 \leq c < 1$, when $L_f^{max} < -\ln c$, then we have*

$$\delta_f(e^{-L_f^{max}} - c) \leq \delta_{\mathcal{T}}/2,$$

where

$$\delta_{\mathcal{T}} := \min_{tag(x_i)\neq tag(x_j)} \|x_i - x_j\| \geq \delta_0$$

is called the empirical separation gap, i.e., the smallest distance between two different labeled training points. Furthermore, when $c = 0.5$, the upper bound $\delta_{\mathcal{T}}/2$ is tight.

Proof. The proof can be found in Appendix I. □

Besides the upper bound, the lower bound of δ_f is also important to the accuracy. We have observed that in practice NNs always have satisfactory smoothness. Based on this observation, we have the following theorem for the accuracy.

Theorem 3.3 (Lower bound of accuracy)**.** *Assume that there exists a constant $\kappa > 0$, such that*

$$\kappa\delta_0 \leq \kappa\delta_{\mathcal{T}} \leq \delta_f(e^{-L_f^{max}} - 0.5) \leq \delta_{\mathcal{T}}/2$$

holds for any single label training set $\mathcal{T}$ and any trained network f on $\mathcal{T}$, then we have the following conclusions for the expected accuracy p and the expected error $\mathcal{E} = 1 - p$:

(i) with the same condition of Theorem 3.1,
$p \geq 1 - \frac{\sqrt{d}}{\delta}(1-\rho_{\mathcal{T}})$,

(ii) $\mathcal{E} \leq \alpha(\mathcal{T}) \cdot CC(\mathcal{T})$,

(iii) $\lim_{\rho_{\mathcal{T}} \to 1} p = 1$,

where $\delta = \min(\delta_0, \delta_f(e^{-L_f^{max}} - c))$, *and* $\alpha(\mathcal{T}) = \frac{\sqrt{d} \cdot CD(\mathcal{T})}{\min(\delta_0, \kappa\delta_{\mathcal{T}})}$.

Proof. (i) is the conclusion of Theorem 3.1. The proof of (ii) and (iii) can be found in Appendix J. □

Here, the cover complexity $CC(\mathcal{T})$ consists of two parts, one represents the richness of the whole training set while the other part describes the degree of separation between different labeled subsets. As for $\alpha(\mathcal{T})$, both the denominator and numerator seem to have a forward correlation with respect to separation level. What we wish is that $\alpha(\mathcal{T})$ is almost close to a constant with high probability and the expected error $\mathcal{E}$ is mainly determined by $CC(\mathcal{T})$, which approximately represents the complexity level of data set. We will provide more information in detail in the section of numerical results.

4. Numerical results

In this section, we use numerical simulations to test the accuracy of neural networks in terms of the data distribution (cover complexity), and neural network smoothness.

4.1. Data distribution

In this subsection, we explore how $CC(\mathcal{T})$ affects the expected error $\mathcal{E}$. In our experiments, we test several data sets, including MNIST (LeCun et al., 1998), CIFAR-10 (Krizhevsky and Hinton, 2009), CIFAR-100 (Krizhevsky and Hinton, 2009), COIL-20 (Nene et al., 1996b), COIL-100 (Nene et al., 1996a), SVHN (Netzer et al., 2011). In addition to the original data set, we also create some variants: (1) the images of grey color, (2) the images extracted from a convolutional layer after training the original data set using a convolutional neural network (CNN), (3) combine several categories into one category to reduce the number of total categories, see Table 1 and details in Appendix K.

For a training data set $\mathcal{T}$, we estimate $h^{\mu}_{\mathcal{T}}(r)$ by the proportion of the test data points within the balls with radius r centered at training data points, i.e.,

$$h^{\mu}_{\mathcal{T}}(r) \approx \frac{\#\text{ Test points within radius-}r\text{ balls of training points}}{\#\text{ Test data points}},$$

and then $\rho_{\mathcal{T}}$ is obtained by Definition 2.4. Similarly, we estimate $CD(\mathcal{T})$ and then compute $CC(\mathcal{T})$. Next for each data set, we train fully-connected neural networks with different hyperparameters, and record the best error we observed, see the details in Appendix K. The cover complexity and best error of each data set is shown in Table 1.

These data sets are divided into three groups according to their output dimensions. For each group of the same output dimension, the error is linearly correlated with $CC(\mathcal{T})$, see Fig. 3A, regardless of the input dimension. In addition, we find that all the cases collapse into a single line by normalized the error $\mathcal{E}$ with $\sqrt{K}$, see Fig. 3B.

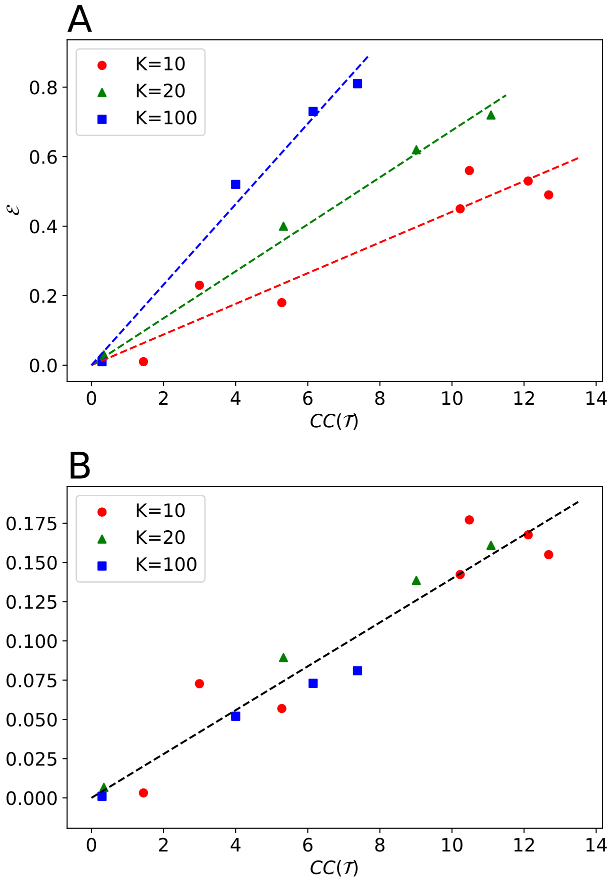


Figure 3: Relationship between cover complexity and error achieved by fully-connected neural networks. (**A**) Linear relationship between cover complexity $CC(\mathcal{T})$ and error $\mathcal{E}$ for different data sets with different category number K. (**B**) Linear relationship between cover complexity $CC(\mathcal{T})$ and normalized error $\frac{\mathcal{E}}{\sqrt{K}}$ for different data sets. Red, green and blue points represent data sets with output dimension of 10, 20 and 100, respectively. The line is fitted as $\frac{\mathcal{E}}{\sqrt{K}} \approx 0.014CC(\mathcal{T})$.

It is noteworthy that the $CC(\mathcal{T})$ of convolutional variants of data sets is much smaller than the original data sets, and hence the expected accuracy increases. The results confirm the importance of data distribution.

Next, we consider the most difficult data set, i.e., data with random labels. We choose MNIST and then assign each image a random label. We repeat this process 50 times, and compute each $CC(\mathcal{T})$. The distribution of $|CC(\mathcal{T})|$ is shown in Fig. 4. The smallest $|CC(\mathcal{T})|$ is ~300, which is much larger than the normal data sets with $CC(\mathcal{T}) < 20$. This extreme example again confirms that $CC(\mathcal{T})$ is a proper measure of the difficulty of classifying a data set.

4.2. Neural network smoothness

In this subsection, we will investigate the relationship between the neural network smoothness $\delta_f(e^{-L_f^{max}} - c)$ and the

Data Set	Variants	Input dim (d)	Output dim (K)	$\rho_{\mathcal{T}}$	$CD(\mathcal{T})$	$CC(\mathcal{T})$	$\mathcal{E}$	$\frac{\mathcal{E}}{\sqrt{K}}$
MNIST	Original	784	10	.8480	.1053	1.442	.01	.0032
CIFAR-10	Original	3072	10	.8332	.0163	10.23	.45	.1423
CIFAR-10	Grey	1024	10	.8486	.0125	12.11	.53	.1676
CIFAR-10	Conv	1024	10	.9505	.0094	5.280	.18	.0569
SVHN	Original	3072	10	.9034	.0076	12.68	.49	.1550
SVHN	Grey	1024	10	.9117	.0084	10.48	.56	.1771
SVHN	Conv	1024	10	.9632	.0123	2.995	.23	.0727
CIFAR-100	Original (coarse)	3072	20	.8337	.0185	9.012	.62	.1386
CIFAR-100	Grey (coarse)	1024	20	.8541	.0132	11.08	.72	.1610
CIFAR-100	Conv (coarse)	1024	20	.9626	.0070	5.326	.40	.0894
COIL-20	Original	16384	20	.9176	.2385	.3453	.03	.0067
CIFAR-100	Original (fine)	3072	100	.8337	.0270	6.149	.73	.0730
CIFAR-100	Grey (fine)	1024	100	.8541	.0198	7.380	.81	.0810
CIFAR-100	Conv (fine)	1024	100	.9457	.0136	4.000	.52	.0520
COIL-100	Original	49152	100	.9430	.1944	.2930	.01	.0010

Table 1: Cover complexity $CC(\mathcal{T})$, best error $\mathcal{E}$, and normalized error $\frac{\mathcal{E}}{\sqrt{K}}$ of different data sets. Different variants of data sets are used, including the original RGB or grey images (Original), grey images (Grey), and images extracted from a CNN (Conv). Images in CIFAR-100 have two variants: 100 categories (fine) and 20 categories (coarse). See Appendix K for details.

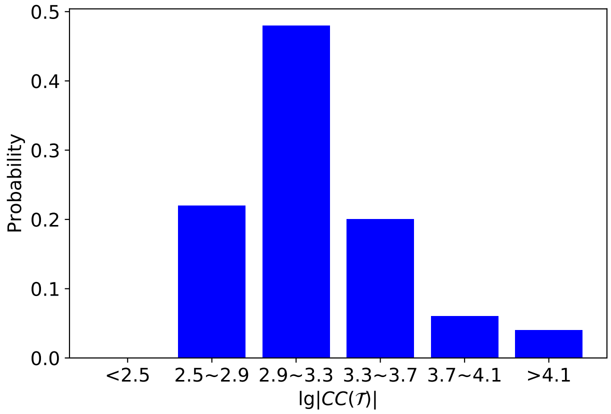


Figure 4: Distribution of $CC(\mathcal{T})$ of randomly label MNIST data set. We assign each image in MNIST a random label and compute its $CC(\mathcal{T})$. The smallest $|CC(\mathcal{T})|$ is ~300. The distribution is obtained for 50 random data sets.

accuracy. We first show results for one- and two-dimensional problems, where $\delta_f(e^{-L_f^{max}} - c)$ can be computed accurately by brute force. Subsequently, we show the high dimensional problem of MNIST data set, and $\delta_f(e^{-L_f^{max}} - c)$ is estimated by Eq. (3).

4.2.1. One- and two-dimensional problems

We first consider a one-dimensional case and a two-dimensional case. For the one-dimensional case, we choose the sample space $D = [0, 1]$, $K = 2$, and the ideal label function as

$$tag(x) = \begin{cases} \{1\} & \text{if } x \in [0, 0.5 - \frac{\delta_0}{2}] \\ \{1, 2\} & \text{if } x \in (0.5 - \frac{\delta_0}{2}, 0.5 + \frac{\delta_0}{2}) \\ \{2\} & \text{if } x \in [0.5 + \frac{\delta_0}{2}, 1] \end{cases},$$

with separation gap $\delta_0 = 0.1$. We use n equispaced points ($n \geq 4$ is an even number) on $D \setminus (0.5 - \frac{\delta_0}{2}, 0.5 + \frac{\delta_0}{2})$ as the training set, i.e., $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$, where

$$\mathcal{T}_1 = \{0, \frac{1 - \delta_0}{n - 2}, \frac{1 - \delta_0}{n - 2} \cdot 2, \ldots, \frac{1}{2} - \frac{\delta_0}{2}\},$$
$$\mathcal{T}_2 = \{\frac{1}{2} + \frac{\delta_0}{2}, \ldots, 1 - \frac{1 - \delta_0}{n - 2}, 1\}.$$

For the two-dimensional case, we choose the sample space $D = [0, 1]^2$, $K = 2$, and the ideal label function as

$$tag(\mathbf{x}) = \begin{cases} \{1\} & \text{if } \|\mathbf{x} - (0.5, 0.5)\| \leq 0.4 - \frac{\delta_0}{2} \\ \{2\} & \text{if } \|\mathbf{x} - (0.5, 0.5)\| \geq 0.4 + \frac{\delta_0}{2} \\ \{1, 2\} & \text{otherwise} \end{cases},$$

with $\delta_0 = 0.1$. For the training set, we first choose $n = m^2$ equispaced points, i.e., $\mathcal{T} = \{0, \frac{1}{m-1}, \frac{2}{m-1}, \ldots, \frac{m-2}{m-1}, 1\}^2$, and then remove the points with label $\{1, 2\}$ to ensure that all samples are of single label.

In our experiments, we use a 3-layer fully-connected NN with ReLU activation and 30 neurons per layer. The neural network is trained for 1000 iterations by Adam optimizer (Kingma and Ba, 2015) for the one-dimensional problem, and 2000 iterations for the two-dimensional problem. For the one-dimensional problem, the c-accuracy p_c with $c = 0.5$ and lower bounds for different number of training set are listed in Table 2. We can see that the bounds become tighter when n is larger.

n	L_f^{max}	$\rho_\mathcal{T}$	δ	p_c	$h_\mathcal{T}(\delta)$	$1 - \frac{\sqrt{d}}{\delta}(1-\rho_\mathcal{T})$
10	.285	.972	.045	1.0	0.80	0.38
20	.246	.988	.041	1.0	1.00	0.69
40	.182	.994	.041	1.0	1.00	0.85
80	.127	.997	.038	1.0	1.00	0.92

Table 2: Comparison between c-accuracy p_c with $c = 0.5$ and the lower bounds for different training set in the one-dimensional problem. The neural network is trained for 1000 iterations by the Adam optimizer.

During the training process of the neural network, the test loss first decreases and then increases, while δ_f first increases and then decreases, see Fig. 5A for the one-dimensional problem ($n = 20$) and Fig. 5B for the two-dimensional problem ($n = 400$). δ_f is bounded by $\delta_\mathcal{T}/2$, as proved in Theorem 3.2. We also observe that the trends of test loss and δ_f coincide, and thus we should stop the training when δ_f begins to decrease to prevent overfitting.

4.2.2. *High-dimensional problem*

In the high-dimensional problem of MNIST, we consider the average loss L_f instead of the maximum loss L_f^{max}, which is very sensitive to extreme points. As shown in Eq. (3), we use the following quantity to bound $\delta_f(e^{-L_f} - c)$:

$$\Delta_f = \frac{e^{-L_f} - c}{\|W_1\|_2 \cdots \|W_l\|_2}.$$

Because we use c-accuracy to approximate the true accuracy, for the classification problems with two categories, these two values are equivalent. However, they are not equal for problems with more than two categories, where the best c depends on the properties of the data set, such as the easiness. If the data set is easy to classify, such as MNIST, the best c should be close to 1. In our example, we choose $c = 0.9$. We train MNIST using a 3-layer fully-connected NN with ReLU activation and 100 neurons per layer for 100 epochs. In Fig. 6, we can also see the consistency between the test loss and neural network smoothness, as we observed in the low-dimensional problems.

5. Discussion

When neural networks are used to solve classification problems, we expect that the accuracy is dependent on some properties of the data set. However, it is still quite surprising as we have seen in Section 4.1 that the accuracy and error are approximately linearly dependent on the cover complexity of

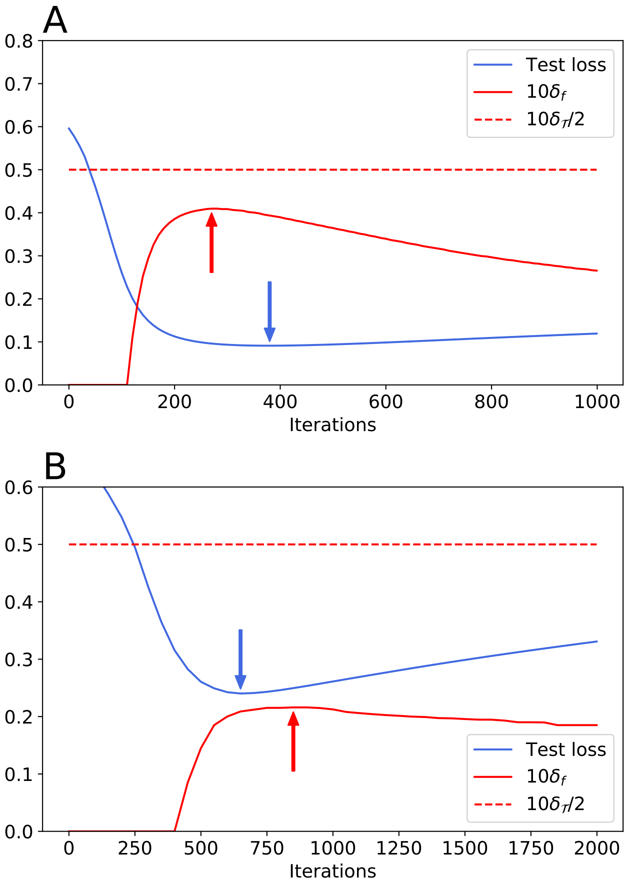


Figure 5: Consistency between the test loss and neural network smoothness δ_f during the training process of the neural network. (**A**) One-dimensional problem of $n = 20$. (**B**) Two-dimensional problem of $n = 400$. δ_f is bounded by $\delta_\mathcal{T}/2$. The arrows indicate the minimum of the test loss and the maximum of δ_f.

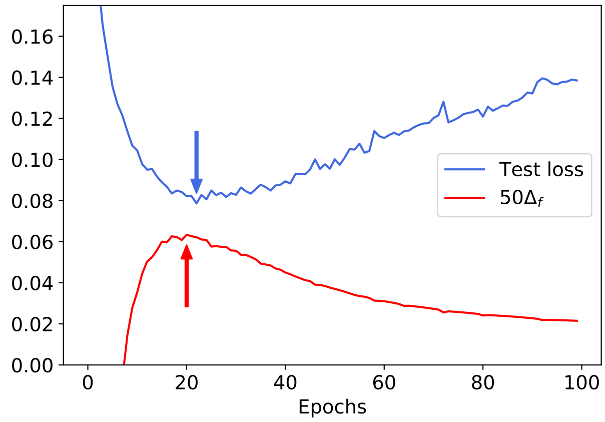


Figure 6: Consistency between test loss and neural network smoothness during the training of the neural network for MNIST. The arrows indicate the minimum of the test loss and the maximum of Δ_f.

data sets. Theorem 3.3(ii) provides an upper bound of the error, but a lower upper is missing. To fully explain this observation, two conjectures of the learnability of fully-connected neural networks are proposed:

- For a data set $\mathcal{T}$,
$$\mathcal{E} \approx c(K) \cdot CC(\mathcal{T}),$$
where $c(K)$ is a constant depending only on K.
- For a data set $\mathcal{T}$,
$$\frac{\mathcal{E}}{\sqrt{K}} \approx c \cdot CC(\mathcal{T}),$$
where $c \approx 0.014$ is a constant.

On the other hand, the theoretical and numerical results provide us a better understanding of the generalization of neural network from the training procedure. The smoothness $\delta_f(e^{-L} - c)$ of neural networks plays a key role, where L is the maximum loss L_f^{max} or the average loss L_f. We can see that:

- $\delta_f(e^{-L} - c)$ depends on both the regularity of f and the loss value L (which is also depends on f). Large $\delta_f(e^{-L} - c)$ requires good regularity and large $e^{-L} - c$, i.e., small L. However, small L could correspond to bad regularity of f. Thus, there is a trade-off between the loss value L and the regularity of f.
- Due to this trade-off, $\delta_f(e^{-L} - c)$ increases first and then decrease during training process. Hence, we should not optimize neural networks excessively. Instead, we should stop the training early when $\delta_f(e^{-L} - c)$ begins to decrease, which leads to another "early stopping" strategy to prevent overfitting.

We also note that the lower bound of $\delta_f(e^{-L} - c)$ in Eq. (3) relates to the norm of weight matrices of neural networks:

$$\frac{e^{-L} - c}{\|W_1\|_2 \cdots \|W_l\|_2}.$$

There have been some works to study the norm-based complexity of neural networks (see the Introduction), and these bounds typically scale with the product of the norms of the weight matrices, e.g., (Neyshabur et al., 2017a)

$$\frac{1}{\gamma_{margin}^2} \prod_{i=1}^{l} h_i \|W_i\|_2^2,$$

where h_i and W_i are the number of nodes and weight matrix in layer i of a l-layers network, and γ_{margin} is the margin quantity, which describes the goodness of fit of the trained network on the data. The product of the matrix norms depends exponentially on depth, while some recent works show that the generalization bound could scale polynomially in depth under some assumptions (Nagarajan and Kolter, 2019a; Wei and Ma, 2019). Clearly our neural net smoothness $\delta_f(e^{-L} - c)$ has a much weaker dependence on depth than exponent, and the detailed analysis of this dependence is left for future work.

6. Conclusion

In this paper, we study the generalization error of neural networks for classification problems in terms of data distribution and neural network smoothness. We first establish a new framework for classification problems. We introduce the *cover complexity* (CC) to measure the difficulty of learning a data set, an accuracy measure called *c-accuracy* which is stronger than the standard classification accuracy, and *the inverse of modules of continuity* to quantify neural network smoothness. Subsequently, we derive a quantitative bound for the expected accuracy/error in Theorem 3.3, which considers both the cover complexity and neural network smoothness.

We validate our theoretical results by several data sets of images. Our numerical results verify that the expected error of trained network has a linear relationship with respect to the CC. In addition, we find that the most difficult case, i.e., random labeled data, leads to quite large CC. Hence, CC is a reliable measure for the difficulty of a data set. On the other hand, we observe a clear consistency between test loss and neural network smoothness during the training process.

Acknowledgements

This work is supported by the DOE PhILMs project (No. de-sc0019453), the AFOSR grant FA9550-17-1-0013, and the DARPA AIRA grant HR00111990025. The work of P. Jin and Y. Tang is partially supported by the National Natural Science Foundation of China (Grant No. 11771438).

Appendix A. Example of topology

Example Appendix A.1. *Given*

$$T = 2^{\{1,2,3\}} \backslash \{\varnothing\} = \{\ \{1\},\ \{2\},\ \{3\},\ \{1,2\},\ \{1,3\},\ \{2,3\},\ \{1,2,3\}\ \},$$

$$\begin{aligned} U = \{\ &\{\{1\},\{1,2\},\{1,3\},\{1,2,3\}\},\ \{\{2\},\{2,1\},\{2,3\},\{1,2,3\}\},\\ &\{\{3\},\{3,1\},\{3,2\},\{1,2,3\}\},\ \{\{1,2\},\{1,2,3\}\},\\ &\{\{1,3\},\{1,2,3\}\},\ \{\{2,3\},\{1,2,3\}\},\ \{\{1,2,3\}\}\ \}, \end{aligned}$$

then τ_T is the topology generated by U.

In this example, $\{\{1\},\{1,2\},\{1,3\},\{1,2,3\}\}$ is an open set, since it consists of all elements containing label $\{1\}$, and $\{\{2,3\},\{1,2,3\}\}$ is also an open set with common part $\{2,3\}$. Besides open sets from base U, $\{\{1\},\{1,2\},\{1,3\},\{2,3\},\{1,2,3\}\}$ is still an open set as the union of the two shown above.

Appendix B. Proof of Proposition 2.1

Proof. We use the proof by contradiction. Assume that the result does not hold, then

$$\exists\{x_n\},\{y_n\} \subseteq D,\ s.t.\ \|x_n - y_n\| \to 0 (n \to \infty), \text{ and}$$
$$tag(x_n) \cap tag(y_n) = \varnothing.$$

As we know that D is compact, then there exists $x^* \in D$ and the subsequence $\{x_{k_n}\}$ of $\{x_n\}$ such that $x_{k_n} \to x^*$. As $\|x_n - y_n\| \to 0$, thus $y_{k_n} \to x^*$. Choose any $i_0 \in tag(x^*)$, then there exists a sufficient large k_{n_0} such that $i_0 \in tag(x_{k_{n_0}})$, $i_0 \in tag(y_{k_{n_0}})$. Therefore $tag(x_{k_{n_0}}) \cap tag(y_{k_{n_0}}) \neq \emptyset$, this is contradictory with the assumption. □

Appendix C. Proof of Proposition 2.2

Proof. Consider δ_0 defined in this proposition. For any two different points x, y with distance less than δ_0, at least one of the two is a full label point, therefore $tag(x) \cap tag(y) \neq \emptyset$. For any $\delta > \delta_0$, according to the definition of δ_0, there exist two points x_0, y_0 satisfying

$$\|x_0 - y_0\| < \delta, \quad tag(x_0) \cap tag(y_0) = \emptyset.$$

The two facts imply that δ_0 is the supremum of δ satisfying Proposition 2.1. □

Appendix D. Proof of Proposition 2.3

Proof. According to the definition,

$$\begin{aligned}\sqrt{d}\rho_{\mathcal{T}} &= \int_0^{\sqrt{d}} h^{\mu}_{\mathcal{T}}(t)dt \\ &= \int_0^{r} h^{\mu}_{\mathcal{T}}(t)dt + \int_r^{\sqrt{d}} h^{\mu}_{\mathcal{T}}(t)dt \\ &< r \cdot h^{\mu}_{\mathcal{T}}(r) + (\sqrt{d} - r) \cdot 1 \\ &= \sqrt{d} - r(1 - h^{\mu}_{\mathcal{T}}(r)),\end{aligned}$$

thus

$$h^{\mu}_{\mathcal{T}}(r) > 1 - \frac{\sqrt{d}}{r}(1 - \rho_{\mathcal{T}}).$$

□

Appendix E. Estimate of total cover

In this section, we estimate the TC by the number of samples in training set. The notations, such as D, d, P, μ, $\rho_{\mathcal{T}}$, as well as training set

$$\mathcal{T} = \{x_1, x_2, \dots, x_n\} \subseteq D$$

are the same as before. Note that samples in $\mathcal{T}$ are drawn from D by P. Before performing the analysis, we display the following preliminary issues (**Definitions Appendix E.1-Appendix E.4, Theorem Appendix E.1**) which are easily found in Mitzenmacher and Upfal (2017):

Definition Appendix E.1. *A range space is a pair $(X, \mathcal{R})$ where:*

1. *X is a (finite or infinite) set of points;*

2. *$\mathcal{R}$ is a family of subsets of X, called ranges.*

Definition Appendix E.2. *Let $(X, \mathcal{R})$ be a range space and let $Y \subseteq X$. The projection of $\mathcal{R}$ on Y is*

$$\mathcal{R}_Y = \{R \cap Y | R \in \mathcal{R}\}.$$

Definition Appendix E.3. *Let $(X, \mathcal{R})$ be a range space. A set $Y \subseteq X$ is shattered by $\mathcal{R}$ if $|\mathcal{R}_Y| = 2^{|Y|}$. The Vapnik-Chervonenkis (VC) dimension of a range space $(X, \mathcal{R})$ is the maximum cardinality of a set $Y \subseteq X$ that is shattered by $\mathcal{R}$. If there are arbitrarily large finite sets that are shattered by $\mathcal{R}$, then the VC dimension is infinite.*

Definition Appendix E.4. *Let $(X, \mathcal{R})$ be a range space, and let $\mathcal{F}$ be a probability distribution on X. A set $N \subseteq X$ is an $\epsilon - net$ for X with respect to $\mathcal{F}$ if for any set $R \in \mathcal{R}$ such that $Pr_{\mathcal{F}}(R) \geq \epsilon$, the set R contains at least one point from N, i.e.,*

$$\forall R \in \mathcal{R}, Pr_{\mathcal{F}}(R) \geq \epsilon \Rightarrow R \cap N \neq \emptyset.$$

Theorem Appendix E.1. *Let $(X, \mathcal{R})$ be a range space with VC dimension d_{vc} and let $\mathcal{F}$ be a probability distribution on X. For any $0 < \eta, \epsilon \leq 1/2$, there is an*

$$m = O\left(\frac{d_{vc}}{\epsilon} \ln \frac{d_{vc}}{\epsilon} + \frac{1}{\epsilon} \ln \frac{1}{\eta}\right)$$

such that a random sample from $\mathcal{F}$ of size greater than or equal to m is an $\epsilon - net$ for X with probability at least $1 - \eta$.

Now let

$$B_D = \{B(x, r) \cap D | x \in \mathbb{R}^d, r > 0\},$$

we first show (D, B_D) is a range space with VC dimension $d+1$.

Lemma Appendix E.1. *The VC dimension of range space (D, B_D) is $d + 1$.*

Proof. All vertices of a simplex of dimension d in D is shattered by B_D, hence the VC dimension of (D, B_D) is at least $d + 1$. Furthermore, Dudley (1979) proves that the VC dimension of (D, B_D) is at most $d + 1$. □

Set

$$B^{\epsilon}_D(r) = \{A \in B_D | A = B(x, s) \cap D, 0 < s \leq r, \mu(A) \geq \epsilon\},$$

and

$$\varrho(\epsilon) = \frac{1}{\sqrt{d}} \int_0^{\sqrt{d}} \mu\left(\bigcup_{A \in B^{\epsilon}_D(\frac{1}{2}r)} A\right) dr,$$

we have the following lemmas.

Lemma Appendix E.2. $\bigcup_{A \in B^{\epsilon}_D(\frac{1}{2}r)} A \subseteq \bigcup_{x_i \in \mathcal{T}} B(x_i, r) \bigcap D$ *when $\mathcal{T}$ is an $\epsilon - net$ for (D, B_D).*

Proof. For any $x \in \bigcup_{A \in B^{\epsilon}_D(\frac{1}{2}r)} A$, assume that $x \in A^* = B(x^*, s^*) \cap D$, $s^* \leq \frac{1}{2}r$, $\mu(A^*) \geq \epsilon$. Since $\mathcal{T}$ is an $\epsilon - net$ and $\mu(A^*) \geq \epsilon$, we know $\mathcal{T} \cap A^* \neq \emptyset$. Thus there exists $x_t \in \mathcal{T}$ such that $\|x_t - x^*\| < s^* \leq \frac{1}{2}r$. Therefore

$$\|x - x_t\| \leq \|x - x^*\| + \|x^* - x_t\| < \frac{1}{2}r + \frac{1}{2}r = r.$$

The above inequality shows that $x \in B(x_t, r) \cap D \subseteq \bigcup_{x_i \in \mathcal{T}} B(x_i, r) \cap D$. □

Lemma Appendix E.3. $\lim_{\epsilon \to 0} \varrho(\epsilon) = 1.$

Proof. Since

$$\lim_{\epsilon \to 0} \mu\left(\bigcup_{A \in B_D^\epsilon(\frac{1}{2}r)} A\right) = 1 \; and \; \mu\left(\bigcup_{A \in B_D^\epsilon(\frac{1}{2}r)} A\right) \leq 1,$$

by dominated convergence theorem, we have

$$\begin{aligned}
\lim_{\epsilon \to 0} \varrho(\epsilon) &= \lim_{\epsilon \to 0} \frac{1}{\sqrt{d}} \int_0^{\sqrt{d}} \mu\left(\bigcup_{A \in B_D^\epsilon(\frac{1}{2}r)} A\right) dr \\
&= \frac{1}{\sqrt{d}} \int_0^{\sqrt{d}} \lim_{\epsilon \to 0} \mu\left(\bigcup_{A \in B_D^\epsilon(\frac{1}{2}r)} A\right) dr \\
&= \frac{1}{\sqrt{d}} \int_0^{\sqrt{d}} 1 dr \\
&= 1.
\end{aligned}$$

□

According to the aforementioned lemmas, we deduce the following theorem.

Theorem Appendix E.2. *Let $\mathcal{T} = \{x_1, x_2, \ldots, x_n\}$ be the training set drawn from D by P, then for any $0 < \eta, \epsilon \leq \frac{1}{2}$, there exists an*

$$m = O\left(\frac{d+1}{\epsilon} \ln \frac{d+1}{\epsilon} + \frac{1}{\epsilon} \ln \frac{1}{\eta}\right)$$

such that

$$\rho_{\mathcal{T}} \geq \varrho(\epsilon)$$

holds with probability at least $1 - \eta$ when $n \geq m$. Note that $\varrho(\epsilon) \to 1$ when $\epsilon \to 0$.

Proof. Theorem Appendix E.1 shows that $\mathcal{T}$ is an $\epsilon - net$ for range space (D, B_D) with probability at least $1 - \eta$ when $n \geq m$. By lemma Appendix E.2, we have

$$\begin{aligned}
\rho_{\mathcal{T}} &= \frac{1}{\sqrt{d}} \int_0^{\sqrt{d}} \mu\left(\bigcup_{x_i \in \mathcal{T}} B(x_i, r) \cap D\right) dr \\
&\geq \frac{1}{\sqrt{d}} \int_0^{\sqrt{d}} \mu\left(\bigcup_{A \in B_D^\epsilon(\frac{1}{2}r)} A\right) dr \\
&= \varrho(\epsilon).
\end{aligned}$$

□

From this theorem, we know that a large number of samples lead to a sufficiently large $\rho_{\mathcal{T}}$ with a high probability. It is worth noting that the estimate of number of samples is independent of probability density P, consequently we only need to enlarge the training set and do not need to care about the real probability distribution on D.

In the previous sections, there is an assumption that our obtained training set is of single label, so we will naturally consider this special case in the sequel.

Denote

$$D_{sin} = \{x \in D | \; tag(x) \in \{\{1\}, \{2\}, \cdots, \{K\}\} \},$$

$$B_{D_{sin}} = \{B(x, r) \cap D_{sin} | x \in \mathbb{R}^d, r > 0\},$$

$$P_{sin} = \frac{P}{\mu(D_{sin})}, \quad \mu_{sin}(A) = \frac{\mu(A \cap D_{sin})}{\mu(D_{sin})},$$

and $(D_{sin}, B_{D_{sin}})$ is a range space with VC dimension at most $d + 1$. Let

$$\mathcal{T} = \{x_1, \cdots, x_n\} \subseteq D_{sin},$$

that is, the samples in $\mathcal{T}$ are drawn from D_{sin} by P_{sin}. As before, denote

$$B^\epsilon_{D_{sin}}(r) = \{A \in B_D | A = B(x, s) \cap D, 0 < s \leq r, \mu_{sin}(A) \geq \epsilon\},$$

$$\varrho_{sin}(\epsilon) = \frac{1}{\sqrt{d}} \int_0^{\sqrt{d}} \mu\left(\bigcup_{A \in B^\epsilon_{D_{sin}}(\frac{1}{2}r)} A\right) dr.$$

We have the following lemmas.

Lemma Appendix E.4. $\bigcup_{A \in B^\epsilon_{D_{sin}}(\frac{1}{2}r)} A \subseteq \bigcup_{x_i \in \mathcal{T}} B(x_i, r) \cap D$ *when $\mathcal{T}$ is an $\epsilon - net$ for $(D_{sin}, B_{D_{sin}})$.*

Lemma Appendix E.5. $\lim_{\epsilon \to 0} \varrho_{sin}(\epsilon) = c_{sin}$, *here*

$$c_{sin} = \frac{1}{\sqrt{d}} \int_0^{\sqrt{d}} \mu\left(\bigcup_{A \in C_{D_{sin}}(\frac{1}{2}r)} A\right) dr,$$

$$C_{D_{sin}}(r) = \{A \in B_D | A = B(x, s) \cap D, 0 < s \leq r, \mu_{sin}(A) > 0\}.$$

From these two lemmas we deduce the following theorem.

Theorem Appendix E.3. *Let $\mathcal{T} = \{x_1, x_2, \ldots, x_n\}$ be the training set drawn from D_{sin} by P_{sin}, then for any $0 < \eta, \epsilon \leq \frac{1}{2}$, there exists an*

$$m = O\left(\frac{d+1}{\epsilon} \ln \frac{d+1}{\epsilon} + \frac{1}{\epsilon} \ln \frac{1}{\eta}\right)$$

such that

$$\rho_{\mathcal{T}} \geq \varrho_{sin}(\epsilon)$$

holds with probability at least $1 - \eta$ when $n \geq m$. Note that $\varrho_{sin}(\epsilon) \to c_{sin}$ when $\epsilon \to 0$.

The proofs for Lemmas Appendix E.4-Appendix E.5 and Theorem Appendix E.3 are very similar to those for Lemmas Appendix E.2-Appendix E.3 and Theorem Appendix E.2, respectively. We omit them here. It is noteworthy that c_{sin} is intuitively very close to 1, even equal to 1. At worst, c_{sin} is at least greater than $\mu(D_{sin})$, which may be quite large in practice.

Appendix F. Proof of Proposition 2.4

Proof. Let $\mathcal{T}$ be the training set and λ be a positive constant greater than 1, $\widetilde{\mathcal{T}} = \mathcal{T}/\lambda$, then

$$\begin{aligned}
1 - \rho_{\widetilde{\mathcal{T}}} &= \frac{1}{\sqrt{d}} \int_0^{\sqrt{d}} (1 - h^{\widetilde{\mu}}_{\widetilde{\mathcal{T}}}(r))dr \\
&= \frac{1}{\sqrt{d}} \int_0^{\infty} (1 - h^{\widetilde{\mu}}_{\widetilde{\mathcal{T}}}(r))dr \\
&= \frac{1}{\sqrt{d}} \int_0^{\infty} (1 - h^{\mu}_{\mathcal{T}}(\lambda r))dr \\
&= \frac{1}{\lambda\sqrt{d}} \int_0^{\infty} (1 - h^{\mu}_{\mathcal{T}}(r))dr \\
&= (1 - \rho_{\mathcal{T}})/\lambda.
\end{aligned}$$

For the same reason,

$$\begin{aligned}
CD(\widetilde{\mathcal{T}}) &= \frac{1}{K(K-1)} \sum_{i \neq j} (1 - \rho(\widetilde{\mathcal{T}}_i, \widetilde{\mu}_j)) - \frac{1}{K} \sum_i (1 - \rho(\widetilde{\mathcal{T}}_i, \widetilde{\mu}_i)) \\
&= \frac{1}{\lambda K(K-1)} \sum_{i \neq j} (1 - \rho(\mathcal{T}_i, \mu_j)) - \frac{1}{\lambda K} \sum_i (1 - \rho(\mathcal{T}_i, \mu_i)) \\
&= CD(\mathcal{T})/\lambda,
\end{aligned}$$

therefore

$$CC(\widetilde{\mathcal{T}}) = CC(\mathcal{T}).$$

□

Appendix G. Proof of Proposition 3.1

Proof. Denote by $\widetilde{\mathcal{T}} = \{x_i \in \mathcal{T} \mid f \text{ is } c_1\text{−accurate at } x_i\}$, $\delta = \min(\delta_0, \delta_f(c_1 - c_2))$. For any $x_i \in \widetilde{\mathcal{T}}$, denote by $tag(x_i) = \{k_i\}$, from the definition of $\widetilde{\mathcal{T}}$ we know that $f_{k_i}(x_i) > c_1$.

For any $x \in B(x_i, \delta) \cap D$, from Proposition 2.1 we know that $tag(x) \cap tag(x_i) \neq \varnothing$, then $k_i \in tag(x)$. On the other hand, $\|f(x) - f(x_i)\|_\infty < c_1 - c_2$, so we have $|f_{k_i}(x) - f_{k_i}(x_i)| < c_1 - c_2$, therefore

$$f_{k_i}(x) > f_{k_i}(x_i) - (c_1 - c_2) > c_1 - (c_1 - c_2) = c_2,$$

which means that f is c_2−accurate at x, that is to say

$$H_f^{c_2} \supseteq \bigcup_{x_i \in \widetilde{\mathcal{T}}} B(x_i, \delta) \bigcap D.$$

Then

$$\begin{aligned}
p_{c_2} &= \mu(H_f^{c_2}) \\
&\geq \mu\left(\bigcup_{x_i \in \widetilde{\mathcal{T}}} B(x_i, \delta) \bigcap D\right) \\
&= h^{\widetilde{\mu}}_{\widetilde{\mathcal{T}}}(\delta) \\
&> 1 - \frac{\sqrt{d}}{\delta}(1 - \rho_{\widetilde{\mathcal{T}}}) \\
&= 1 - \frac{\sqrt{d}}{\delta}(1 - p^{\mathcal{T}}_{c_1}\rho_{\mathcal{T}}).
\end{aligned}$$

The second inequality can be derived from Proposition 2.3. □

Appendix H. Proof of Theorem 3.1

Proof. Denote by $\delta = \min(\delta_0, \delta_f(e^{-L_f^{max}} - c))$. For any $x_i \in \mathcal{T}$, $x \in B(x_i, \delta) \cap D$. As $\|x - x_i\| < \delta_0$, then $tag(x) \cap tag(x_i) \neq \varnothing$, therefore $k_i \in tag(x)$.

As

$$\|x - x_i\| < \delta_f(e^{-L_f^{max}} - c),$$

then

$$\begin{aligned}
\|f(x) - f(x_i)\|_\infty &< e^{-L_f^{max}} - c, \\
|f_{k_i}(x) - f_{k_i}(x_i)| &< e^{-L_f^{max}} - c,
\end{aligned}$$

furthermore

$$f_{k_i}(x) > f_{k_i}(x_i) - e^{-L_f^{max}} + c \geq e^{-L_f^{max}} - e^{-L_f^{max}} + c = c,$$

therefore f is c-accurate at x. Then we obtain

$$\begin{aligned}
p_c &= \mu(H_f^c) \\
&\geq \mu\left(\bigcup_{x_i \in \mathcal{T}} B(x_i, \delta) \bigcap D\right) \\
&= h_{\mathcal{T}}(\delta) \\
&> 1 - \frac{\sqrt{d}}{\delta}(1 - \rho_{\mathcal{T}}).
\end{aligned}$$

The second inequality can be derived from Proposition 2.3. □

Appendix I. Proof of Theorem 3.2

Proof. We will prove it by contradiction. Consider two points x_1 and x_2 with different labels, and $\|x_1 - x_2\| = \delta_{\mathcal{T}}$.

If $\delta_f(e^{-L_f^{max}} - c) > \delta_{\mathcal{T}}/2$, then by the definition of $\delta_f(e^{-L_f^{max}} - c)$, $\|f(\frac{x_1+x_2}{2}) - f(x_1)\|_\infty < e^{-L_f^{max}} - c$, since $\|\frac{x_1+x_2}{2} - x_1\| = \|\frac{x_2-x_1}{2}\| = \delta_{\mathcal{T}}/2$. Similarly, $\|f(\frac{x_1+x_2}{2}) - f(x_2)\|_\infty < e^{-L_f^{max}} - c$. Then we have

$$\|f(x_1) - f(x_2)\|_\infty < 2(e^{-L_f^{max}} - c) \leq 2e^{-L_f^{max}} - 1.$$

On the other hand, by the definition of L_f^{max}, $-\ln f_{k_1}(x_1) \leq L_f^{max}$ and $-\ln f_{k_2}(x_2) \leq L_f^{max}$, so $f_{k_1}(x_1) \geq e^{-L_f^{max}}$, $f_{k_2}(x_2) \geq e^{-L_f^{max}}$ and $f_{k_1}(x_2) \leq 1 - f_{k_2}(x_2) \leq 1 - e^{-L_f^{max}}$, therefore

$$\begin{aligned}
\|f(x_1) - f(x_2)\|_\infty &\geq f_{k_1}(x_1) - f_{k_1}(x_2) \\
&\geq e^{-L_f^{max}} - (1 - e^{-L_f^{max}}) \\
&= 2e^{-L_f^{max}} - 1.
\end{aligned}$$

For $c = 0.5$, the bound is achieved when f is linear from $(x_1, f(x_1))$ to $(x_2, f(x_2))$, and $f_{k_1}(x_1) = e^{-L_f^{max}}$, $f_{k_1}(x_2) = 1 - e^{-L_f^{max}}$. □

Appendix J. Proof of Theorem 3.3

Proof. From Theorem 3.1 and assumption we know that

$$\begin{aligned}
p \geq p_{0.5} &> 1 - \frac{\sqrt{d}}{\min(\delta_0, \delta_f(e^{-L_f^{max}} - 0.5))}(1 - \rho_{\mathcal{T}}) \\
&\geq 1 - \frac{\sqrt{d}}{\min(\delta_0, \kappa\delta_{\mathcal{T}})}(1 - \rho_{\mathcal{T}}) \\
&\geq 1 - \frac{\sqrt{d}}{\kappa\delta_0}(1 - \rho_{\mathcal{T}}),
\end{aligned}$$

which implies $\lim_{\rho_T \to 1} p = 1$. Note that κ is less than 0.5 and δ_0 is only determined by the classification problem itself. The above inequality is easy to convert into form (ii). □

Appendix K. Detailed information of data and parameters for training

First list the information of data selection.

1. MNIST: Last 55000 samples of training set for training and all the 10000 samples of test set for test.

2. CIFAR-10: First 49000 samples of training set for training and all the 10000 samples of test set for test.

3. CIFAR-100: First 49000 samples of training set for training and all the 10000 samples of test set for test.

4. COIL-20: 1200 samples whose end numbers are not multiples of 6 for training and 240 samples whose end numbers are multiples of 6 for test.

5. COIL-100: 6000 samples whose end numbers are not multiples of 30 for training and 1200 samples whose end numbers are multiples of 30 for test.

6. SVHN: First 50000 samples of training set for training and first 10000 samples of test set for test.

Parameters for networks are listed in Table K.3.

For generating convolution data, we choose the following structure

$$\begin{aligned}&conv[128] - relu - batchnorm - conv[256] - relu-\\&batchnorm - pool - conv[512] - relu - batchnorm-\\&conv[256] - relu - batchnorm - conv[64] - relu-\\&batchnorm - pool - (extract\ data) - dense[512]-\\&batchnorm - dropout - dense[128] - batchnorm-\\&dropout - dense[output]\end{aligned}$$

with kernel size 3×3(strides 1) and pool size 2×2(strides 2), then train this CNN with batch size 64, learning rate 0.001 and optimizer RMSProp for 5 epochs. After that, extract new data at location mentioned in above structure.

References

Allen-Zhu, Z., Li, Y., Liang, Y., 2018a. Learning and generalization in overparameterized neural networks, going beyond two layers. arXiv preprint arXiv:1811.04918 .

Allen-Zhu, Z., Li, Y., Song, Z., 2018b. A convergence theory for deep learning via over-parameterization. arXiv preprint arXiv:1811.03962 .

Arora, S., Du, S., Hu, W., Li, Z., Wang, R., 2019. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. arXiv preprint arXiv:1901.08584 .

Arora, S., Ge, R., Neyshabur, B., Zhang, Y., 2018. Stronger generalization bounds for deep nets via a compression approach. arXiv preprint arXiv:1802.05296 .

Bartlett, P., Foster, D., Telgarsky, M., 2017a. Spectrally-normalized margin bounds for neural networks, in: Advances in Neural Information Processing Systems, pp. 6240–6249.

Bartlett, P., Harvey, N., Liaw, C., Mehrabian, A., 2017b. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. arXiv preprint arXiv:1703.02930 .

Bartlett, P., Mendelson, S., 2002. Rademacher and gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research 3, 463–482.

Baykal, C., Liebenwein, L., Gilitschenski, I., Feldman, D., Rus, D., 2018. Data-dependent coresets for compressing neural networks with applications to generalization bounds. arXiv preprint arXiv:1804.05345 .

Belkin, M., Hsu, D., Ma, S., Mandal, S., 2018. Reconciling modern machine learning and the bias-variance trade-off. arXiv preprint arXiv:1812.11118 .

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent, in: Proceedings of COMPSTAT'2010. Springer, pp. 177–186.

Bottou, L., Bousquet, O., 2008. The tradeoffs of large scale learning, in: Advances in neural information processing systems, pp. 161–168.

Cao, Y., Gu, Q., 2019. A generalization theory of gradient descent for learning over-parameterized deep ReLU networks. arXiv preprint arXiv:1902.01384 .

Chen, Y., Jin, C., Yu, B., 2018. Stability and convergence trade-off of iterative optimization algorithms. arXiv preprint arXiv:1804.01619 .

Cheng, Y., Wang, D., Zhou, P., Zhang, T., 2018. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. IEEE Signal Processing Magazine 35, 126–136.

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

Dinh, L., Pascanu, R., Bengio, S., Bengio, Y., 2017. Sharp minima can generalize for deep nets, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org. pp. 1019–1028.

Du, S., Lee, J., Li, H., Wang, L., Zhai, X., 2018. Gradient descent finds global minima of deep neural networks. arXiv preprint arXiv:1811.03804 .

Dudley, R., 1979. Balls in rk do not cut all subsets of k + 2 points. Advances in Mathematics 31, 306 – 308.

Dziugaite, G., Roy, D., 2017. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. arXiv preprint arXiv:1703.11008 .

Friedman, J., Hastie, T., Tibshirani, R., 2001. The elements of statistical learning. volume 1. Springer series in statistics New York.

Gonen, A., Shalev-Shwartz, S., 2017. Fast rates for empirical risk minimization of strict saddle problems. arXiv preprint arXiv:1701.04271 .

Gunasekar, S., Lee, J., Soudry, D., Srebro, N., 2018. Implicit bias of gradient descent on linear convolutional networks, in: Advances in Neural Information Processing Systems, pp. 9461–9471.

Hardt, M., Recht, B., Singer, Y., 2015. Train faster, generalize better: Stability of stochastic gradient descent. arXiv preprint arXiv:1509.01240 .

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

Kawaguchi, K., Kaelbling, L., Bengio, Y., 2017. Generalization in deep learning. arXiv preprint arXiv:1710.05468 .

Keskar, N., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P., 2016. On large-batch training for deep learning: Generalization gap and sharp minima. arXiv preprint arXiv:1609.04836 .

Kingma, D., Ba, J., 2015. Adam: A method for stochastic optimization. International Conference on Learning Representations .

Krizhevsky, A., Hinton, G., 2009. Learning multiple layers of features from tiny images. Technical Report. Citeseer.

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. Imagenet classification with deep convolutional neural networks. Neural Information Processing Systems 25. doi:10.1145/3065386.

Kuzborskij, I., Lampert, C., 2017. Data-dependent stability of stochastic gradient descent. arXiv preprint arXiv:1703.01678 .

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324.

Lee, J., Simchowitz, M., Jordan, M., Recht, B., 2016. Gradient descent converges to minimizers. arXiv preprint arXiv:1602.04915 .

Li, Y., Liang, Y., 2018. Learning overparameterized neural networks via stochastic gradient descent on structured data, in: Advances in Neural Information Processing Systems, pp. 8157–8166.

Liang, T., Poggio, T., Rakhlin, A., Stokes, J., 2017. Fisher-rao metric, geometry, and complexity of neural networks. arXiv preprint arXiv:1711.01530 .

Data Set	Version	*i*	*ii*	*iii*	*iv*	*v*	*vi*	*vii*	*viii*	*ix*	*x*	*xi*	*xii*	Best Error
MNIST	Original	.02	.02	.05	.02	.02	.04	.02	.02	.04	.01	.02	.03	.01
CIFAR-10	Original	.47	.46	.52	.48	.46	.51	.47	.45	.50	.47	.45	.49	.45
CIFAR-10	Grey	.55	.55	.63	.55	.54	.62	.54	.53	.61	.54	.53	.59	.53
CIFAR-10	Conv	.18	.18	.19	.19	.18	.19	.18	.18	.18	.18	.18	.18	.18
SVHN	Original	.80	.59	.49	.80	.73	.60	.80	.69	.51	.80	.72	.64	.49
SVHN	Grey	.80	.64	.56	.80	.76	.66	.80	.64	.58	.80	.75	.66	.56
SVHN	Conv	.27	.23	.23	.31	.24	.23	.31	.24	.23	.69	.25	.24	.23
CIFAR-100	Original(coarse)	.64	.64	.69	.64	.63	.68	.63	.62	.67	.63	.62	.66	.62
CIFAR-100	Grey(coarse)	.74	.73	.79	.74	.73	.78	.74	.72	.78	.74	.72	.77	.72
CIFAR-100	Conv(coarse)	.40	.41	.45	.41	.41	.44	.40	.41	.44	.41	.40	.43	.40
COIL-20	Original	.08	.05	.05	.07	.06	.05	.08	.05	.05	.07	.05	.03	.03
CIFAR-100	Original(fine)	.75	.75	.82	.75	.74	.81	.74	.73	.80	.75	.73	.79	.73
CIFAR-100	Grey(fine)	.83	.83	.90	.83	.83	.90	.82	.81	.88	.83	.81	.87	.81
CIFAR-100	Conv(fine)	.53	.54	.64	.53	.54	.63	.52	.52	.61	.53	.52	.59	.52
COIL-100	Original	.02	.01	.01	.03	.02	.01	.03	.02	.01	.02	.02	.01	.01

Table K.3: Set activation function and optimizer to ReLU and Adam respectively, and choose several different network structures and learning rates. Train the networks for 10000 iterations with batch size of 300, then select the best test error as final result for each training procedure. Table K.4 is the detail setup for each case.

Structure \ Learning Rate	10^{-3}	10^{-4}	10^{-5}
[Input 256 256 Output]	*i*	*ii*	*iii*
[Input 256 256 256 Output]	*iv*	*v*	*vi*
[Input 512 512 Output]	*vii*	*viii*	*ix*
[Input 512 512 512 Output]	*x*	*xi*	*xii*

Table K.4: Detailed setup for each case.

Liao, Q., Poggio, T., 2017. Theory ii: Landscape of the empirical risk in deep learning. arXiv preprint arXiv:1703.09833 .

Lu, L., Shin, Y., Su, Y., Karniadakis, G., 2019. Dying ReLU and initialization: Theory and numerical examples. arXiv preprint arXiv:1903.06733 .

Maas, A., Hannun, A., Ng, A., 2013. Rectifier nonlinearities improve neural network acoustic models, in: Proc. icml, p. 3.

Mitzenmacher, M., Upfal, E., 2017. Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis. 2nd ed., Cambridge University Press, New York, NY, USA.

Nagarajan, V., Kolter, J., 2019a. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience, in: International Conference on Learning Representations.

Nagarajan, V., Kolter, J., 2019b. Generalization in deep networks: The role of distance from initialization. arXiv preprint arXiv:1901.01672 .

Nene, S., Nayar, S., Murase, H., 1996a. Columbia object image library (coil-100). Citeseer .

Nene, S., Nayar, S., Murase, H., et al., 1996b. Columbia object image library (coil-20). Technical report CUCS-005-96 .

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A., 2011. Reading digits in natural images with unsupervised feature learning, in: Advances in Neural Information Processing Systems.

Neyshabur, B., Bhojanapalli, S., McAllester, D., Srebro, N., 2017a. Exploring generalization in deep learning, in: Advances in Neural Information Processing Systems, pp. 5947–5956.

Neyshabur, B., Bhojanapalli, S., Srebro, N., 2017b. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. arXiv preprint arXiv:1707.09564 .

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., Srebro, N., 2019. The role of over-parametrization in generalization of neural networks, in: International Conference on Learning Representations.

Neyshabur, B., Salakhutdinov, R., Srebro, N., 2015. Path-sgd: Path-normalized optimization in deep neural networks, in: Advances in Neural Information Processing Systems, pp. 2422–2430.

Neyshabur, B., Tomioka, R., Srebro, N., 2014. In search of the real inductive bias: On the role of implicit regularization in deep learning. arXiv preprint arXiv:1412.6614 .

Poggio, T., Kawaguchi, K., Liao, Q., Miranda, B., Rosasco, L., Boix, X., Hidary, J., Mhaskar, H., 2017. Theory of deep learning iii: explaining the non-overfitting puzzle. arXiv preprint arXiv:1801.00173 .

Rahaman, N., Arpit, D., Baratin, A., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A., 2018. On the spectral bias of deep neural networks. arXiv preprint arXiv:1806.08734 .

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

Sokolic, J., Giryes, R., Sapiro, G., Rodrigues, M., 2016. Generalization error of invariant classifiers. arXiv preprint arXiv:1610.04574 .

Sokolić, J., Giryes, R., Sapiro, G., Rodrigues, M., 2017. Robust large margin deep neural networks. IEEE Transactions on Signal Processing 65, 4265–4280.

Soudry, D., Hoffer, E., Nacson, M., Gunasekar, S., Srebro, N., 2018. The implicit bias of gradient descent on separable data. The Journal of Machine Learning Research 19, 2822–2878.

Wei, C., Ma, T., 2019. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. arXiv preprint arXiv:1905.03684 .

Xu, Z., Zhang, Y., Luo, T., Xiao, Y., Ma, Z., 2019. Frequency principle: Fourier analysis sheds light on deep neural networks. arXiv preprint arXiv:1901.06523 .

Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O., 2016. Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530 .

Zhang, C., Liao, Q., Rakhlin, A., Miranda, B., Golowich, N., Poggio, T., 2018. Theory of deep learning iib: Optimization properties of sgd. arXiv preprint arXiv:1801.02254 .

Zhou, W., Veitch, V., Austern, M., Adams, R., Orbanz, P., 2018. Compressibility and generalization in large-scale deep learning. arXiv preprint arXiv:1804.05862 .