Understanding Deep Learning via Decision Boundary

Shiyi Lei*, Fengxiang He*, Member, IEEE, Yancheng Yuan, and Dacheng Tao*, Fellow, IEEE

Abstract—This article discovers that the neural network (NN) with lower decision boundary (DB) variability has better generalizability. Two new notions, algorithm DB variability and \((\epsilon, \eta)\)-data DB variability, are proposed to measure the DB variability from the algorithm and data perspectives. Extensive experiments show significant negative correlations between the DB variability and the generalizability. From the theoretical view, two lower bounds based on algorithm DB variability are proposed and do not explicitly depend on the sample size. We also prove an upper bound of order \(O((1/\sqrt{m}) + \epsilon + \eta \log(1/\eta))\) based on data DB variability. The bound is convenient to estimate without the requirement of labels and does not explicitly depend on the network size which is usually prohibitively large in deep learning.

Index Terms—Decision boundary (DB), deep learning theory, explainability of deep learning, neural network (NN).

I. INTRODUCTION

NEURAL networks (NNs) have achieved significant success in vast applications [1], [2], including computer vision [3], natural language processing [4], and data mining [5]. However, the advance of NNs is arduous to be characterized by conventional statistical learning theory based on hypothesis complexity [6], such as Vapnik–Chervonenkis (VC)-dimension [7] and Rademacher complexity [8]. According to the conventional theory, models of larger hypothesis complexity possess worse generalizability, while NNs are usually over-parameterized but have excellent generalizability.

In this article, we attempt to explain the excellent generalizability of deep learning from the perspective of decision boundary (DB) variability. Intuitively, the DB variability of NNs is largely determined by two means: 1) the randomness introduced by the optimization algorithm and 2) the fluctuations of the training data when they are sampled from the data generating distribution. Following this intuition, we design two terms, algorithm DB variability and \((\epsilon, \eta)\)-data DB variability, to measure the DB variability.

Algorithm DB variability measures the variability of DBs in different training repeats. We conduct extensive experiments on the CIFAR-10/100 datasets [9] to explore which factors would determine algorithm DB variability. We visualize the trend of the algorithm DB variability with respect to (w.r.t.) training strategies, training time, sample sizes, and label noise ratios. The empirical results demonstrate that algorithm DB variability has: 1) negative correlations with the training time and the sample size; 2) a positive correlation with the label noise; and 3) a negative correlation with the generalizability (or, test accuracy, in experiments). Benefiting from this significant correlation, algorithm DB variability can be employed to do model selection without the requirement of test labels, and shows superior performance than conventional marginal likelihood measurement. From the theoretical view, we prove two lower bounds based on the algorithm DB variability, which fully supports our experiments.

\((\epsilon, \eta)\)-data DB variability is proposed to characterize the DB from the view of the randomness in training data. Given a NN, if its DB can be “reconstructed” by training a network with the same architecture from scratch on a smaller \(\eta\)-subset (which contains \(\eta\%\) examples of the source training set), while the “error” of the reconstruction is not larger than \(\epsilon\), we call the model has \((\epsilon, \eta)\)-data DB variability. Specifically, we may define the reconstruction error as the approximation error of the reconstructed DB on the whole training set. Moreover, an \(\eta-\epsilon\) curve can be drawn numerically. The area under the \(\eta-\epsilon\) curve could be an informative indicator for characterizing the generalization of NNs. An \(O((1/\sqrt{m}) + \epsilon + \eta \log(1/\eta))\)
generalization bound based on the \((\epsilon, \eta)\)-data DB variability is proved, which demonstrates the relationship between the generalization of NNs and DB variability. In contrast to many existing generalization bounds based on hypothesis complexity that require access to the weight norm, our bounds only need the predictions. This brings significant advantages in empirically approximating the generalization bound in: 1) black-model settings, where model parameters are unavailable and 2) over-parameterized settings, where calculating the weight norm is of a prohibitively high computing burden.

To our best knowledge, this is the first work on explaining deep learning via the variability of DB. Our research also sheds light on understanding a variety of interesting phenomena, including the entropy and the complexity of DB. Through the lens of DB variability, we may also design novel algorithms by reducing the DB variability.

II. RELATED WORKS

A. Deep Learning Theory

In learning theory, generalizability refers to the capability of well-trained models predicting on unseen data. Conventional theory suggests that the generalizability has a negative correlation with the hypothesis complexity \([6]\), such as VCDimension \([7]\) and Rademacher complexity \([8]\); models with larger complexity fit the training data better. This is usually summarized as the “bias-variance trade-off.” This principle faces significant challenges in deep learning \([10]\), \([11]\). Zhang et al. \([13]\) summarized as the “bias-variance trade-off.” This principle faces significant challenges in deep learning \([10]\), \([11]\). Belkin et al. \([13]\) showed that NNs and then gradually learn fit more complex patterns \([38]\). Numerous researchers \([13]\), \([14]\), \([15]\), \([16]\). Belkin et al. \([13]\) showed that NNs and then gradually learn fit more complex patterns \([38]\). Numerous researchers \([13]\), \([14]\), \([15]\), \([16]\). Belkin et al. \([13]\) showed that NNs and then gradually learn fit more complex patterns \([38]\). Numerous researchers \([13]\), \([14]\), \([15]\), \([16]\). Belkin et al. \([13]\) showed that NNs and then gradually learn fit more complex patterns \([38]\). Numerous researchers \([13]\), \([14]\), \([15]\), \([16]\).

Many works attribute the success of NNs to the effectiveness of the stochastic gradient descent (SGD) algorithm \([18]\), \([19]\), \([20]\), \([21]\). For example, Jin et al. \([22]\) showed that SGD can escape from the local minima. The loss landscape of the networks has also been extensively analyzed and it has been proven that there are no spurious local minima for linear NNs \([23]\), \([24]\), \([25]\). Nevertheless, this elegant property does not hold for general networks where nonlinear activation functions are involved \([26]\), \([27]\). A recent study also attempts to explore the implicit bias of NNs in the over-parameterized regime. Soudry et al. \([28]\) showed that the over-parameterized networks converge to the max-margin solution when the training data is linear-separable. Some other research has also been conducted along this line \([29]\), \([30]\), \([31]\).

Empirical studies have also attempted to explain the decent performance of networks by uncovering their learning properties \([32]\), \([33]\), \([34]\), \([35]\). For instance, NNs are shown to tend to fit the low-frequency information first \([36]\), \([37]\) and then gradually learn fit more complex patterns \([38]\) during the training procedure. He and Su \([39]\) showed that NNs own the unique property of local elasticity that the predictions on the input data \(x'\) will not be significantly perturbed when the neural net is updated via SGD at the training example \(x\) if \(x'\) is “dissimilar” to \(x\). Similar phenomena are also observed by Fort et al. \([40]\). Besides, Papyan et al. \([41]\) uncover a novel phenomenon, neural collapse, which sheds light on interpreting the effectiveness of deep models \([42]\).

B. Decision Boundaries in Neural Networks

DB, which partitions the input space with different labels, is an important notion in machine learning. Recent studies attempt to understand NNs from the aspect of decision boundaries \([43]\), \([44]\), \([45]\). Alfarra et al. \([46]\) employ the tropical geometry to represent the DB of NNs. Guan and Loew \([47]\) empirically show a negative correlation between the complexity of DB and the generalization performance of NNs. Mickisch et al. \([48]\) reveal an insightful phenomenon that the distance from data to the DB continuously decreases during the training. More recently, researchers uncover that NNs only rely on the most discriminative or the simplest features to construct the DB \([49]\), \([50]\). Besides, Samangouei et al. \([51]\) also explain the predictions of NNs via constructing examples crossing the DB. To our best knowledge, this article is the first work on: 1) theoretically characterizing the complexity of DB via the new measure, DB variability and 2) explaining the negative correlation between the generalizability and DB variability.

III. PRELIMINARIES

We denote the training set by \(S = \{(x_i, y_i), i = 1, \ldots, m\}\), where \(x_i \in \mathbb{R}^n\), \(n\) is the dimension of input data, \(y_i \in [k] = \{1, \ldots, k\}\), \(k\) is the number of classes, and \(m = |S|\) is the training sample size. We assume that \((x_i, y_i)\) are independent and identically distributed (i.i.d.) random variables drawn from the data generating distribution \(D\). Denote the classifier as \(f_\theta(x) : \mathbb{R}^n \rightarrow \mathbb{R}^k\), which is a NN parameterized by \(\theta\). The output of \(f_\theta(x)\) is a \(k\)-dimensional vector and is assumed to be a discrete probability density function. Let \(f_\theta^{(i)}(x)\) be the \(i\)-th component of \(f_\theta(x)\), hence \(\sum_{i=1}^k f_\theta^{(i)}(x) = 1\). We define \(T(f_\theta, x) = \{i \in \{1, \ldots, k\} | f_\theta^{(i)}(x) = \max_j f_\theta^{(j)}(x)\}\) to denote the set of predicted labels by \(f_\theta\) on \(x\). Due to the randomness of the learning algorithm \(\mathcal{A}\), let \(Q(\theta) = \mathcal{A}(S)\) denote the posterior distribution returned by the learning algorithm \(\mathcal{A}\) leveraged on the training set \(S\). Hence, we focus on the Gibbs classifier (a.k.a. random classifier) \(f_\theta \sim Q\). 0–1 loss is employed in this article, and the expected risks in terms of \(\theta\) and \(Q\) are defined as follows:

\[
R_D(\theta) = E_{(x, y) \sim D}[\mathbb{I}(y \neq T(f_\theta, x))]
\]

(1)

and

\[
R_D(Q) = E_{(x, y) \sim D}E_{\theta \sim Q}[\mathbb{I}(y \neq T(f_\theta, x))]
\]

(2)

respectively. Here \(\mathbb{I}(\cdot)\) is the indicator function. Since the data generating distribution \(D\) is unknown, evaluating the expected risk \(R_D\) is not practical. Therefore, it is a practical way to estimate the expected risk by the empirical risk \(R_S\), which is
defined as follows:

\[
\mathcal{R}_\mathcal{S}(\theta) = \mathbb{E}_{(x, y) \sim \mathcal{S}}[\mathbb{I}(y \neq T(f_\theta(x)))] \\
= \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}(y_i \neq T(f_\theta(x_i))) \\
\mathcal{R}_\mathcal{Q}(q) = \mathbb{E}_{(x, y) \sim \mathcal{Q}}[\mathbb{I}(y \neq T(f_\theta(x)))] \\
= \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_\theta[q][\mathbb{I}(y_i \neq T(f_\theta(x_i)))]
\]

where \((x_i, y_i) \in \mathcal{S}\) and \(m = |\mathcal{S}|\).

A. Decision Boundary

Intuitively, if the output \(k\)-dimensional vector \(f_\theta(x)\) on the input example \(x\) has a tie, i.e., more than one entry of the vector has the maximum value, then \(x\) is considered to locate on the DB of \(f_\theta\). Formally, the DB can be formally defined as below.

**Definition 1 (Decision Boundary):** Let \(f_\theta : \mathbb{R}^n \rightarrow \mathbb{R}^k\) be a NN for classification parameterized by \(\theta\), where \(n\) and \(k\) are the dimensions of input and output, respectively. Then, the DB of \(f_\theta\) is defined by

\[
\{x \in \mathbb{R}^n | \exists i \neq j \in [k], f_\theta^{(i)}(x) = f_\theta^{(j)}(x) = \max_q f_\theta^{(q)}(x)\}.
\]

Consequently, we have the following remark.

**Remark 1:**

1) If an input example \((x, y)\) is not located on the DB of \(f_\theta\), \(T(f_\theta(x))\) is a singleton, and we have

\[
\mathbb{I}(y \neq T(f_\theta(x))) = \mathbb{I}(y \neq T(f_\theta(x))).
\]

2) If the input \(x\) is a boundary point, in practice, we randomly draw a label from the set \(T(f_\theta(x))\) as the prediction of \(f_\theta\) on \(x\).

B. Adversarial Training

Adversarial training (AT) can enhance the adversarial robustness of NNs against adversarial examples, which is generated through a popular approach projected gradient descent (PGD) [52] in our empirical studies as an example. More specifically, AT can be formulated as solving the minimax-loss problem as follows:

\[
\min_{\theta} \frac{1}{m} \sum_{i=1}^{m} \max_{\gamma} \ell(f_\theta(x_i), y_i)
\]

where \(\gamma\) is the radius to limit the distance between adversarial examples and original examples. Intuitively, AT enlarges the distances from training examples to decision boundaries to at least \(\gamma\), which is formerly very close to the DB.

C. Model Selection

For a NN \(f_\theta\), it is specified by a model \(\mathcal{M}\) which consists of a network architecture and hyperparameters. For a list of model candidates \(\mathcal{M}\) and given the training data \(\mathcal{S}\), log marginal likelihood (LML) is often employed to select the best model from these candidates [53], as shown as below

\[
\mathcal{M}^* = \arg \max_{\mathcal{M}} \log p(\mathcal{S}|\mathcal{M})
\]

and

\[
p(\mathcal{S}|\mathcal{M}) = \int p(\mathcal{S}|\theta, \mathcal{M}) p(\theta|\mathcal{M}) d\theta.
\]

Because the parameter \(\theta\) is hard to enumerate, Laplace’s method is often used to estimate the LML [54]

\[
\log p(\mathcal{S}|\mathcal{M}) \approx \log q(\mathcal{S}|\mathcal{M})
\]

\[
:= \log p(\mathcal{S}, \theta^*|\mathcal{M}) - \frac{1}{2} \log \left| \begin{bmatrix} 1 & 1 \end{bmatrix} \right|^2
\]

where \(q(\mathcal{S}|\mathcal{M})\) is a Gaussian approximation to \(p(\mathcal{S}|\mathcal{M})\), \(\theta^*\) is the optimum to local minimum, and the Hessian \(\nabla^2_{\theta^*} \log p(\mathcal{S}|\theta^*|\mathcal{M})\). In this article, we follow the implementation in Daxberger et al. [55] to calculate the LML of NNs.

IV. ALGORITHM DB VARIABILITY

Due to the randomness of learning algorithms, there is no doubt that the parameters have a substantial variation in training repeats. However, do the decision boundaries in the different training repeats have a large discrepancy? Quantitatively, we define the algorithm DB variability (AV)
to measure the variability of DBs caused by the randomness of algorithms in different training repeats.

**Definition 2 (Algorithm DB Variability):** Let $f_\theta(x): \mathbb{R}^n \rightarrow \mathbb{R}^k$ be a NN for classification parameterized by $\theta$. Suppose $Q(\theta)$ is the distribution over $\theta$. Then, the algorithm DB variability for $f_\theta$ on $D$ is defined as below

$$AV(f_\theta, D) = \mathbb{E}_{(x,y) \sim D} \mathbb{E}_{\theta \sim Q} [I(T(f_\theta, x) \neq T(f_\theta', x))]$$

(12)

where $T(f_\theta, x) = \{i \in [k] | f_\theta(i)(x) = \max_j f_\theta(j)(x)\}$.

According to the definition, algorithm DB variability reflects the similarity of decision boundaries during different training repeats. An illustration of algorithm DB variability is provided in Fig. 1(a).

### A. Algorithm DB Variability and Generalization

To explore the relationship between the algorithm DB variability and generalization in NNs, we conduct experiments with various popular network architectures, VGG-16 [56], ResNet-18 [3], and WideResNet-28 [57], on standard datasets, CIFAR-10 and CIFAR-100. In detail, VGG-16 [56], ResNet-18 [3], and WideResNet-28 [57] are optimized by standard, nondata-augmentation and AT, respectively, until the training procedure converges. Each training setting (determined by different datasets, architectures, and/or training strategies) is repeated for ten trials with different random seeds to estimate the parameter distribution $Q(\theta)$. In order to simulate the data generating distribution, we trained two conditional BigGANs [58] to produce 100,000 fake (or, synthetic) images for CIFAR-10 and CIFAR-100, respectively. Examples of fake images are shown in Fig. 2(a) and (b).

These generative fake images enable estimating the algorithm DB variability. In every training setting, we plot the average test accuracy versus the algorithm DB variability; as shown in Fig. 2(c). From the plots, we obtain the following four observations.

1) AT dramatically decreases the test accuracy and promotes the algorithm DB variability compared to the standard training.

2) Data augmentation decreases the algorithm DB variability. Intuitively, the images augmented by cropping or flipping are still located on the data generating distribution, so data augmentation can expand the training set. Hence, the expanded training set can characterize wider decision boundaries on the data generating distribution.

3) WideResNet has better test accuracy and lower algorithm DB variability than ResNet and VGG.

4) A negative correlation exists between the test accuracy and the algorithm DB variability. Based on these observations, we propose the following conjecture.

**Hypothesis 1:** NNs with smaller algorithm DB variability on the data generating distributions possess better generalization performance.

We then conduct experiments on the algorithm DB variability w.r.t. training time, sample size, and label noise to concrete this hypothesis.

### B. Algorithm DB Variability and Training Time

To investigate the relationship between algorithm DB variability and training time, we train 40 ResNet-18 with different initial learning rates of 0.1 and 0.01 on CIFAR-10 and CIFAR-100. Then, the algorithm DB variability and test error are calculated at each epoch; see Fig. 3(a). From the plots, two observations can be obtained: 1) algorithm DB variability and test error share a very similar curve w.r.t. the training time and 2) algorithm DB variability decreases during the training process. The decline of algorithm DB variability shows that the interpolation on examples reduces the variability of decision boundaries on data generating distribution. As shown in Fig. 3(b), we collect the points of (algorithm DB variability, test error) from different epochs, and the scatter plots present a significant positive correlation between test error and the algorithm DB variability, and thus supports Hypothesis 1.

### C. Algorithm DB Variability and Sample Size

We next investigate how sample size influences the algorithm DB variability. 100 ResNet-18 are trained on five training sample sets of different sizes randomly drawn from CIFAR-10 and CIFAR-100, while all irrelevant variables are strictly controlled. Then, the algorithm DB variability and test error are calculated in all cases; see Fig. 4(a). From the plots, we have the following two observations: 1) test error and algorithm DB variability share a very similar curve w.r.t. the training sample size; 2) larger sample size, which intuitively helps obtain a more smooth estimation of the DB, also contributes to smaller algorithm DB variability; and 3) there is a significant positive correlation between test error and algorithm DB variability, which fully supports our argument of Hypothesis 1.

### D. Algorithm DB Variability and Label Noise

Belkin et al. [13] and Nakkiran et al. [14] show a surprising epoch-wise double descent of test error, especially with the existence of label noise. We explore in this section the trend of algorithm DB variability when the label noise exists. 20 ResNet-10 are trained for 500 epochs with a constant learning rate of 0.0001 on CIFAR-10 and CIFAR-100 with 20% label noise. We clarify that the noise labels remain constant in different training repeats, which is necessary to estimate the algorithm DB variability. Then, the average test error and algorithm DB variability are calculated at every training epoch, as shown in Fig. 4(b). From the plots, two observations can be derived: 1) the algorithm DB variability also undergoes an epoch-wise double descent during the training process, especially in the left panel of Fig. 4(c) and 2) test error and algorithm DB variability still share a very similar curve w.r.t. the training time with the existence of label noise, which implies that factors influence the generalization of networks can also have an influence on the algorithm DB variability. Hence, algorithm DB variability is an excellent indicator for the generalization ability of networks.

Here, we propose an insightful explanation about the epoch-wise double descent phenomenon of the data DB variability w.r.t. the training time: the increase of algorithm...
DB variability shows that fitting the noisy examples has a considerable effect on the formation of decision boundaries on data generating distribution. However, the algorithm DB variability decreases when the training proceeds further. This indicates that the negative effects brought by fitting the noisy training examples is gradually weakened. In other words, as the training proceeds, NNs can automatically reduce the impact brought by interpolating hard-to-fit examples, which is insightful in explaining the decent generalizability of NNs.

E. DB Variability and Model Selection

In this section, we investigate the algorithm DB variability in model selection. We employ 25 CNN and 25 ResNet with different widths and depths to consist of the model candidate pool. Each network is trained on CIFAR-10/100 with the initial learning rate of 0.01 for 50 epochs and 0.02 for the next 50 epochs. We repeat the training process five times to compute the algorithm DB variability. Then, the algorithm DB variability and average test accuracy are calculated for each model; see Fig. 5, and we also plot the correlation between average LML and test accuracy for comparison. For each plot, we also calculate Spearman’s rank-order correlation coefficients (SCCs) and the corresponding p value of the collected data to investigate the statistical significance of the correlations; please see the bottom of the plots. For the plots, two observations can be obtained.

1) There is a positive correlation between test accuracy and algorithm DB variability (SCCs $> 0.9$) and the correlation is statistically significant ($p < 0.005$).\(^1\)

2) Compared to the correlation between LML and test accuracy, the correlation between algorithm DB variability and test accuracy is more significant. Therefore, algorithm DB variability is a better measurement for

\(^1\)The definition of “statistically significant” has various versions, such as $p < 0.05$ and $p < 0.1$. This article uses a more rigorous one ($p < 0.005$).
model selection than LML, while both of them do not require a test set for validation.

F. Theoretical Evidence

In this section, we explore and develop the theoretical foundations for the algorithm DB variability on data generating distributions. In most practical cases, the dimension of decision boundaries is smaller than the data space. For example, the DB in a 3-D data space is usually two. Thus, we may make the following mild assumption.

Assumption 1: The DB of the classifier network \( f_\theta \) on data generating distribution \( D \) is a set with measure zero.

We then have the following lemma.

Lemma 1: Let \( f_\theta(x) : \mathbb{R}^n \rightarrow \mathbb{R}^k \) be a classifier network parameterized by \( \theta \). If Assumption 1 holds for all \( \theta \sim Q \), then, for all \( i \in \{1, \ldots, k\} \), we have

\[
E_{(x,y) \sim D}[\mathbb{I}(i \in T(f_\theta, x))] = E_{(x,y) \sim D}[\mathbb{I}(T(f_\theta, x) = i)]
\]

and

\[
E_{(x,y) \sim D}[\mathbb{I}(i \notin T(f_\theta, x))] = E_{(x,y) \sim D}[\mathbb{I}(T(f_\theta, x) \neq i)]
\]

Then, we can prove the following theorem.

Theorem 1 (Lower Bound on Expected Risk): Let \( f_\theta(x) : \mathbb{R}^n \rightarrow \mathbb{R}^k \) be a NN for classification parameterized by \( \theta \). Suppose \( Q(\theta) \) is the distribution over \( \theta \). Then, if Assumption 1 holds for all \( \theta \sim Q \), we have

\[
\mathcal{R}_D(Q) \geq 1 - \sqrt{1 - AV(f_\theta, D)}
\]

where \( AV(f_\theta, D) \) is the algorithm DB variability for \( f_\theta \) on data generating distribution \( D \).

Theorem 1 provides a lower bound on the expected risk \( \mathcal{R}_D(Q) \) based on the algorithm DB variability \( AV(f_\theta, D) \). Moreover, when we consider the binary classification, i.e., \( k = 2 \), there is a tighter lower bound.

Theorem 2 (Lower Bound for Binary Case): Let \( f_\theta(x) : \mathbb{R}^n \rightarrow \mathbb{R}^2 \) be a binary classifier network parameterized by \( \theta \) and let \( Q(\theta) \) be the distribution over \( \theta \). Suppose the expected risk \( \mathcal{R}_D(Q) \leq (1/2) \) and Assumption 1 hold for all \( \theta \sim Q \), then we have

\[
\mathcal{R}_D(Q) \geq \frac{1 - \sqrt{1 - 2AV(f_\theta, D)}}{2}
\]

Remark 2: The combination of the empirical and theoretical results suggests a meaningful correlation between algorithm DB variability and expected risk in deep learning.

V. DATA DB VARIABILITY

In Section IV, we introduced the algorithm DB variability, which measures the DB variability caused by the randomness of learning algorithms. In this section, we define the data DB variability to characterize DB variability caused by the randomness in training data.

Definition 3 (Data DB Variability): Let \( f_\theta(x) : \mathbb{R}^n \rightarrow \mathbb{R}^k \) be an NN for classification parameterized by \( \theta \), where \( \theta \sim \mathcal{A}(S) \) is returned by leveraging the stochastic learning algorithm \( \mathcal{A} \) on the training set \( S \), which is sampled from the data generating distribution \( D \). We term \( S_\eta \subset S \) as a \( \eta \)-subset of \( S \) if \( |S_\eta|/|S| = \eta \). Then, if we fixed \( \eta \) and

\[
\inf_{S_\eta \subset S} E_D E_{\theta \sim \mathcal{A}(S)} [I(T(f_\theta, x) \neq T(f_\theta, x))] = \epsilon
\]

the DB of \( f_{\mathcal{A}(S)} \) is said to possess a \( (\epsilon, \eta) \)-data DB variability.

An illustration of data DB variability is presented in Fig. 1(b). The data DB variability contains two parameters of \( \epsilon \) and \( \eta \), respectively. That Gibbs classifier \( f_{\mathcal{A}(S)} \) has a \( (\epsilon, \eta) \)-data DB variability means that only the proportion of \( \eta \) of \( S \), i.e., \( S_\eta \) (which can be considered as “support vector set”) is enough to reconstruct a similar DB with the reconstruction error \( \epsilon \). The data DB variability can also be connected with the complexity of decision boundaries if we assume that simpler decision boundaries rely on a smaller number of “support vectors;” we provide a detailed discussion in Section VI-B.

A. \( \eta \)-\( \epsilon \) curves About Data DB Variability

According to Definition 3, the data DB variability degrades to the algorithm DB variability \( AV(f_\theta, D) \) when \( \eta = 1 \). In other words, the algorithm DB variability is a special case of the data DB variability with \( \eta = 1 \). Therefore, the data DB variability could present more detailed information on reflecting how the DB variability depends on the training set, especially when we observe the variation of the reconstruction error \( \epsilon \) w.r.t. different \( \eta \).

To explore the relationship between the reconstruction error \( \epsilon \) and the proportion of subset \( \eta \), we train 1000 networks of ResNet-18 on CIFAR-10 and CIFAR-100 of different sample sizes \( m \). Albeit finding the most suitable \( \eta \)-subset is intractable, we adopt a coreset selection approach named, selection via proxy [59], which can rank the importance of training examples, to estimate the \( \eta \)-subset for a given training set \( S \) and proportion \( \eta \). Then, through repeatedly training the network on \( S_\eta \), we can estimate the reconstruction error \( \epsilon \). The \( \eta \)-\( \epsilon \) curves of CIFAR-10 and CIFAR-100 are presented in Fig. 6(a) and (b), respectively. From the plots, we have an observation that there is a more rapid decline in \( \epsilon \) along with small \( \eta \) and also a smaller algorithm DB variability when the training sample size \( m \) is larger. Furthermore, we plot the schematic of \( \eta \)-\( \epsilon \) curves w.r.t. different sample size \( m \), as shown in Fig. 6(c). When \( \eta = 0 \), \( f_{\mathcal{A}(S)} \) cannot be better than random guess, and hence \( \epsilon = (k - 1)/k \), where \( k \) is the number of potential categories. It is worth noting that \( \epsilon \) has a sharper drop along with \( \eta \) when the sample size \( m \) is larger. Therefore, we rationally propose the following assumption, which is also shown by the right angle with \( m_\infty \) in Fig. 6(c).

Assumption 2: If \( m \rightarrow \infty \), we have \( \epsilon \rightarrow 0 \) when \( \eta \rightarrow 0 \).

These plots indicate that the area under the \( \eta \)-\( \epsilon \) curve could be a more meticulous predictor for the generalization ability of NNs compared to the algorithm DB variability, which is only a point on the \( \eta \)-\( \epsilon \) curve when \( \eta = 1 \). Hence, the area under the \( \eta \)-\( \epsilon \) curve can also be considered as an extension of the algorithm DB variability: if the Gibbs classifier \( f_{\mathcal{A}(S)} \) possesses a smaller area under the \( \eta \)-\( \epsilon \) curve, it produces more stable decision boundaries with varying training subsets.
subset $S$ fully supports our theory. Better data DB variability possesses better generalization, which complement the set $S$ from simpler distributions than $D$. Theoretical Evidence $\lim_{m \to \infty}$ respectively. (b) Fig. 6. (a) Fig. 5. Scatter plots of LML and algorithm DB variability to test accuracy with different architectures on (a) CIFAR-10 and (b) CIFAR-100.

Fig. 6. (a) $\eta$-$\epsilon$ curves on CIFAR-10 with different training sample sizes 2000 ($m_{2000}$), 2000 ($m_{2000}$), 10000 ($m_{10000}$), 20000 ($m_{20000}$), and 50000 ($m_{50000}$), respectively. (b) $\eta$-$\epsilon$ curves on CIFAR-100 with different training sample sizes. (c) Schematic of the $\eta$-$\epsilon$ curves w.r.t. small ($m_1$), medium ($m_m$), large ($m_l$), and infinite ($m_\infty$) sample sizes, respectively.

B. Theoretical Evidence

In this section, we develop the theoretical foundations for the data DB variability. Our theory suggests that NNs with better data DB variability possess better generalization, which fully supports our theory.

According to the definition of data DB variability, the $\eta$-subset $S_\eta$ plays a similar role of “support vector set,” and the complement set $S \setminus S_\eta = S - S_\eta$ is supposed to be sampled from simpler distributions than $D$. Therefore, we make the following mild assumption.

Assumption 3: Examples in $S \setminus S_\eta$ are assumed to be drawn from the distribution $D_1$, where for all $(x, y) \sim D_1$, $E_{\theta \sim A(S_\eta)}[I(y \in T(f_\theta, x))] = \max_{i \in I} E_{\theta \sim A(S_\eta)}[I(i \in T(f_\theta, x))]$ holds, and

\[
E_D[E_{\theta \sim A(S_\eta), \theta' \sim A(S_\eta)}[I(T(f_\theta, x) \neq T(f_{\theta'}, x))]] \\
\leq E_D[E_{\theta \sim A(S_\eta), \theta' \sim A(S_\eta)}[I(T(f_\theta, x) \neq T(f_{\theta'}, x))] = \epsilon \quad (18)
\]

Remark 3: Assumption 3 can also be stated as follows: the data $(D_1)$ correctly classified by $f(A(S_\eta))$ possesses a lower data DB variability than the average data DB variability on $D$.

Given Assumption 3, we can further derive a probably approximately correct bound w.r.t. $R_{S \setminus S_\eta}(A(S_\eta))$, as shown in the following lemma.

Lemma 2: If the decision boundaries of $f(A(S))$ possess a $(\epsilon, \eta)$-data DB variability and Assumption 3 holds, then, with the probability of at least $1 - \delta$ over a sample of size $m$, we have

\[
R_{S \setminus S_\eta}(A(S_\eta)) \leq \epsilon + \frac{1}{2(1 - \eta)m} \log \frac{1}{\delta}. \quad (19)
\]

We also conduct experiments to show the correlation between $R_{S \setminus S_\eta}(A(S_\eta))$ and $\epsilon$; see Fig. 7. From the plots we obtain an observation that $R_{S \setminus S_\eta}(A(S_\eta))$ is stable when $R_{S \setminus S_\eta}(A(S_\eta))$ is small (about less than 0.5).

Lemma 3: If the decision boundaries of $f(A(S))$ possess a $(\epsilon, \eta)$-data DB variability, then we have

\[
|R_D(A(S)) - R_D(A(S_\eta))| \leq \epsilon.
\]

Lemma 3 shows that the difference between the expected risk of $A(S)$ and $A(S_\eta)$ can be bounded by their difference in decision boundaries. Then, we continue to prove the generalization bound with data DB variability.

Theorem 3 (Data DB Variability-Based Upper Bound on Expected Risk): If the decision boundaries of $f(A(S))$ possess a $(\epsilon, \eta)$-data DB variability on the data generating distribution $D$, and assume $\eta \leq 0.5$ and Assumption 3 holds, then, with the
probability of at least $1 - \delta$ over a sample of size $m$, we have
\[ R_D(A(S)) \leq \Omega + \sqrt{4\Omega\Delta} + 8\Delta + \epsilon \] (20)
where
\[ \Omega = \epsilon + \sqrt{\frac{1}{2(1 - \eta)m} \log \frac{1}{\delta}} \] (21)
\[ \Delta = \eta \log \frac{e}{\eta} + \frac{1}{m} \log \frac{2}{\delta}. \] (22)
Moreover, for sufficiently large $m$, we have
\[ R_D(A(S)) \leq \mathcal{O}\left(\frac{1}{\sqrt{m}} + \epsilon + \eta \log \frac{1}{\eta}\right). \] (23)
According to Assumption 2, when $m \to \infty$, $\eta \to 0$ and $\epsilon \to 0$, while according to (23), $R_D(A(S)) \to 0$. Therefore, the generalization bound is asymptotically converged. Theorem 3 suggests that a smaller data DB variability corresponds to a tighter upper bound on the expected risk, which theoretically verifies the relationship between the data DB variability and the generalization ability of NNs.

VI. DISCUSSION

This section discusses how our findings would shed light on understanding other interesting phenomena.

A. Algorithm DB Variability and the Entropy of Decision Boundaries

If Assumption 1 holds for all $\theta \sim \mathcal{Q}$, $1 - AV(f_\mathcal{Q}, D)$ can be rewritten as follows:
\[ \mathbb{E}_{(x, y) \sim D} \sum_{i=1}^{k} \mathbb{E}_{\theta \sim \mathcal{Q}}[\mathbb{I}(T(f_\theta, x) = i)]. \] (24)
The term $\sum_{i=1}^{k} \mathbb{E}_{\theta \sim \mathcal{Q}}[\mathbb{I}(T(f_\theta, x) = i)]$ can be considered to measure the degree of prediction uncertainty for the given voxel $x$ in the input space $\mathbb{R}^n$. If we leverage $-\log(i)$ on the term $\sum_{i=1}^{k} \mathbb{E}_{\theta \sim \mathcal{Q}}[\mathbb{I}(T(f_\theta, x) = i)]$, we have that
\[ -\log \sum_{i=1}^{k} \mathbb{E}_{\theta \sim \mathcal{Q}}[\mathbb{I}(T(f_\theta, x) = i)] \] (25)
denotes the collision entropy of prediction made by the Gibbs classifier $f_\mathcal{Q}$ on $x$. We can also replace the collision entropy with canonical Shannon entropy
\[ -\sum_{i=1}^{k} \mathbb{E}_{\theta \sim \mathcal{Q}}[\mathbb{I}(T(f_\theta, x) = i)] \log \mathbb{E}_{\theta \sim \mathcal{Q}}[\mathbb{I}(T(f_\theta, x) = i)] \] (26)
in the future research. As such, the algorithm DB variability is closely related to the “entropy of DB,” and the uncanny generalization in NNs might be further uncovered by investigating this low entropy of DB.

B. Data DB Variability and the Complexity of DBs

According to the work by Guan and Loew [47], a complex DB has large curvatures and conjectured to indicate inferior generalization. Nevertheless, from the perspective of causality, we argue that the large curvatures or the nonlinearity of DBs is the result of hard classification tasks, not the cause, and the primary factor in shaping a complex DB during the training procedure should be the significant nonlinearity of the training data. If only the geometric properties of decision boundaries are analyzed without investigating the data, the results might be incomplete and even misleading. Another obstacle to describing the complexity of DBs by their geometric properties is the huge dimensional input space, which makes the geometric properties of DBs hard to quantify and estimate. Therefore, defining the complexity of DBs based on their curvature is not rational and impractical.

Here, we consider the complexity of DBs from the perspective of the training set. During the training procedure, a small part of training examples, considered as “support vectors,” play a more critical role in supervising the formation of decision boundaries and compelling the DB to be gradually more complicated. If the construction of decision boundaries relies on fewer “support vectors,” the DB should be simpler. In other words, if these “support vectors” are excluded from the training sample, the DB will be notably dissimilar when the network is retrained on the modified training set. Hence, the complexity of DBs can be also defined with the notion of the data DB variability: with $(\epsilon, \eta)$-data DB variability, the DB of $f_{A(S)}$ is said to possess $a (\epsilon, \eta)$-complexity.

By considering the data DB variability as the complexity of DB, many phenomenons w.r.t. generalization in deep learning can be easily understood: 1) difficult tasks generally have more complex decision boundaries, since their datum are more nonlinear and contain more “support vectors;” 2) in AT, each data point is converted to a “data ball” with the radius of the adversarial perturbation and has more impact on forming the DBs. Hence, AT contributes to a more complex DB by enlarging the “support vector set,” and thus causes the decline in generalization DB performance; and 3) for data augmentation, generated images are also considered to obey the data generation distribution $D$. Hence, data augmentation decreases the complexity of decision boundaries by greatly expanding the training set $S$, while $|S_i|$ has only a slight growth.

VII. EXPERIMENTAL IMPLEMENTATION DETAILS

This section provides all the additional implementation details for our experiments.

A. Model Training

We employ SGD to optimize all the models and the momentum factor is 0.9. The weight decay factor is set to 5e-4, and the learning rate is decayed by 0.2 every 50 epochs. Besides, basic data augmentation (crop and flip) [57] is adopted in both standard and AT, and only the basic data augmentation is considered in our experiments and analysis.
1) Additional Details in Section IV-A: We train VGG-16, ResNet-18, Wide-ResNet-28 on CIFAR-10 and CIFAR-100. In the training procedure, the model is trained for 200 epochs, in which the batch size is set to 128, and the learning rate is initialized as 0.1. There are three training strategies included in this experiment: standard training, nondata-augmentation training, and AT. In the AT, the radius of the adversarial perturbation is set as 10/255 and \( l_\infty \) distance is selected. The basic data augmentation (cropping and flipping) in the standard training and AT is achieved by the following Pytorch code.

```
1 transforms.RandomCrop(32, padding=4)
2 transforms.RandomHorizontalFlip()
```

The experiment is repeated for ten trials for each (dataset, architecture, training strategy) setting.

2) Additional Details in Section IV-B: We repeatedly train 10 ResNet-18 on CIFAR-10 and CIFAR-100, respectively, with different random seeds. In the training procedure, the model is trained for 200 epochs, in which the batch size is set to 128, and the learning rate is initialized as 0.1 and 0.01, respectively. Basic data augmentation is included during the training process.

3) Additional Details in Section IV-C: We randomly sample examples from the training set of CIFAR-10 and CIFAR-100 to form five datasets with different sizes of [2000, 5000, 10000, 20000, 50000], respectively. 10 ResNet-18 is trained for each dataset. In the training procedure, the model is trained for 200 epochs, in which the batch size is set to 128, and the learning rate is initialized as 0.1. Basic data augmentation is included during the training process.

4) Additional Details in Section IV-D: We randomly change the labels of 20% examples in the training set of CIFAR-10 and CIFAR-100. Then, 10 ResNet-18 are optimized by SGD for 500 epochs on the noise CIFAR-10 and CIFAR-100, respectively. The momentum factor is 0.9, and the learning rate is 0.001 and does not decay during the training process.

5) Additional Details in Section IV-E: We conduct the model selection experiments with convolutional (CNN) and residual (ResNet) networks for CIFAR-10/100 from Lotfi et al. [60]. We use the same architectures.

1) The CNNs consist of up to five blocks of 33 convolutions, followed by a rectified linear unit (ReLU) activation function, and MaxPooling, except in the first layer. BatchNorm is replaced by the fixup initialization [61] as in Immer et al. [62]. The width of the first channel varies from 2 to 32 for both datasets. The last layer is a fully-connected layer to the class logit.

2) ResNets of depths varying from 8 to 32 are used for CIFAR-10 and from 20 to 101 for CIFAR-100. The width varies from 16 to 48 for CIFAR-10 and from 32 to 64 for CIFAR-100.

All models are trained for 100 epochs with initialized learning rate 0.01, and other settings are the same as those in Section VII-A including SGD optimizer, learning rate decay, and data augmentation. We used the Kronecker Laplace approximation to compute the averaging LML over 55 repeated trainings.

6) Additional Details in V-A: We randomly sample examples from the training set of CIFAR-10 and CIFAR-100 to form five datasets with different sizes of [2000, 5000, 10000, 20000, 50000], respectively. For each dataset, we obtain 1010 \( \eta \)-subsets with different \( \eta \) of [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0] via a coreset selection approach named selection via proxy [59]. The related code can be downloaded from https://github.com/stanford-futuredata/selection-via-proxy. The ResNet-18 is repeatedly trained for ten trials to estimate the complexity of decision boundaries for each \( \eta \)-subset.

VIII. PROOFS

The section collects detailed proofs of the results that are omitted in Sections IV-F and V-B. To avoid technicalities, the measurability/integrability issues are ignored throughout this article. Moreover, Fubini’s theorem is assumed to be applicable for any integration w.r.t. multiple variables. In other words, the order of integrations is exchangeable.

A. Proof of Theorem 1

**Proof**: If Assumption 1 holds for all \( \theta \sim Q \), for any networks \( f_\theta \) and \( f_\theta' \) we have

\[
E_{(x,y) \sim D}[\mathbb{I}(y \in T(f_\theta, x)) \mathbb{I}(y \in T(f_{\theta'}, x)) \mathbb{I}(T(f_\theta, x) \neq T(f_{\theta'}, x))] = 0.
\]

Hence,

\[
AV(f_Q, D) = E_{(x,y) \sim D}E_{\theta, \theta' \sim Q}[\mathbb{I}(y \in T(f_\theta, x) \neq T(f_{\theta'}, x))]
\]

\[
= E_{(x,y) \sim D}E_{\theta \sim Q}[\mathbb{I}(y \in T(f_\theta, x)) \mathbb{I}(y \neq T(f_\theta, x))
+ \mathbb{I}(y \neq T(f_\theta, x))\mathbb{I}(T(f_\theta, x) \neq T(f_{\theta'}, x))]
\]

\[
\leq E_{(x,y) \sim D}E_{\theta \sim Q}[\mathbb{I}(y \neq T(f_\theta, x))] + R_D(Q)
\]

\[
= 2R_D(Q) - E_{(x,y) \sim D}E_{\theta \sim Q}[\mathbb{I}(y \neq T(f_\theta, x))]^2
\]

\[
\leq 2R_D(Q) - R_2^2(D).
\]

Solving the inequality yields the desired bound and finishes the proof.

B. Proof of Theorem 2

**Proof**: When the classification is binary, i.e., \( k = 2 \), with Assumption 1, we have

\[
E_{(x,y) \sim D}E_{\theta \sim Q}[\mathbb{I}(T(f_\theta, x) = T(f_{\theta'}, x))]
\]

\[
= E_{(x,y) \sim D}E_{\theta \sim Q}[\mathbb{I}(y \in T(f_\theta, x))
+ E_{(x,y) \sim D}E_{\theta \sim Q}[\mathbb{I}(y \neq T(f_\theta, x))]
\]

\[
= E_{(x,y) \sim D}E_{\theta \sim Q}[\mathbb{I}(y \in T(f_\theta, x))
+ E_{(x,y) \sim D}[1 - E_{\theta \sim Q}[\mathbb{I}(y \in T(f_\theta, x))]]^2
\]

\[
= 2E_{(x,y) \sim D}E_{\theta \sim Q}[\mathbb{I}(y \in T(f_\theta, x))] + 2R_D(Q) - 1.
\]
C. Proof of Lemma 2

The proof is completed.

D. Proof of Lemma 3

Proof: According to Assumption 3 that for all \((x, y) \sim D_1\), we have

\[
\mathbb{E}_{\theta \sim A(S)}[\mathbb{I}(y \in T(f_\theta(x)))] = \max_{i \in [k]} \mathbb{E}_{\theta \sim A(S)}[\mathbb{I}(i \in T(f_\theta(x)))]
\]

then

\[
\mathcal{R}_{D_1}(A(S)) = 1 - \mathbb{E}_{(x, y) \sim D_1}[\mathbb{I}(y = T(f_\theta(x)))]
\]

\[
= 1 - \sum_{i=1}^{k} \mathbb{E}_{A(S)}[\mathbb{I}(i = T(f_\theta(x)))\mathbb{E}_{A(S)}[\mathbb{I}(y = T(f_\theta(x)))]
\]

\[
= 1 - \sum_{i=1}^{k} \mathbb{E}_{A(S)}[\mathbb{I}(i = T(f_\theta(x)))\mathbb{E}_{A(S)}[\mathbb{I}(i = T(f_\theta(x)))]
\]

\[
= \mathbb{E}_{D_1, \theta \sim A(S)}[\mathbb{I}(T(f_\theta(x)) \neq T(f_\theta'(x))]
\]

\[
\leq \mathbb{E}_{D_2, \theta \sim A(S), \theta' \sim A(S)}[\mathbb{I}(T(f_\theta(x)) \neq T(f_\theta'(x))]
\]

\[
= \epsilon.
\]

Because the examples in \(S \setminus S_\eta\) are drawn from \(D_1\), by applying Hoeffding’s Inequality, we have

\[
\Pr[\mathcal{R}_{S_\eta \setminus S_\eta} - \mathcal{R}_{D_1} \geq \epsilon] \leq \exp(-2(1 - \eta)m\epsilon^2).
\]

Plug in \(\delta = \exp(-2(1 - \eta)m\epsilon^2)\) into (27), thus, with the probability of at least \(1 - \delta\), we have

\[
\mathcal{R}_{S_\eta \setminus S_\eta}(A(S)) \leq \epsilon + \sqrt{\frac{2(1 - \eta)m \log \frac{1}{\delta}}{\delta}}.
\]

E. Proof of Theorem 3

We first introduce Lemmas 4 and 5 as below.

**Lemma 4:** [63, Lemma 30.1 in] Assume \(T\) and \(V\) are two datasets independently sampled from the data generating distribution \(D\), then, with the probability of at least \(1 - \delta\), we have

\[
\mathcal{R}_D(A(T)) \leq \mathcal{R}_V(A(T)) + \sqrt{\frac{2\mathcal{R}_V(A(T)) \log(1/\delta)}{|V|}} + 4 \log(1/\delta).
\]

**Proof:** [Proof of Lemma 5]

\[
\Pr[\exists S_\eta \subseteq S \text{ s.t. } \mathcal{R}_D(A(S)) \leq \mathcal{R}_{S_\eta \setminus S_\eta}(A(S))]
\]

\[
\leq \sum_{S_\eta \subseteq S} \Pr[\mathcal{R}_D(A(S)) \leq \mathcal{R}_{S_\eta \setminus S_\eta}(A(S))]
\]

\[
= \left(\frac{m}{\eta m}\right) \delta \leq \left(\frac{\epsilon}{\eta}\right)^{\eta m} \delta.
\]

Plug in \(\delta' = (\epsilon/\eta)^{\eta m} \delta\), and use the assumption \(\eta \leq 1/2\), which implies \(|S_\eta \setminus S_\eta| \geq (m/2)\), then, with the probability of at least \(1 - \delta'\), we have that

\[
\mathcal{R}_D(A(S_\eta)) \leq \mathcal{R}_{S_\eta \setminus S_\eta}(A(S_\eta))
\]

\[
+ \sqrt{\frac{2\mathcal{R}_{S_\eta \setminus S_\eta}(A(S_\eta)) \log(1/\delta')}{|S_\eta \setminus S_\eta|}} + 4 \log(1/\delta')
\]

\[
+ 8 \left(\frac{\epsilon}{\eta} + \frac{1}{m} \log \frac{1}{\delta'}\right)
\]

which concludes the proof.

With the above lemmas, we can derive the generalization bound based on the complexity of DB.

**Proof:** [Proof of Theorem 3]

According to Lemma 5, with the probability of at least \(1 - \delta\), we have

\[
\mathcal{R}_{S_\eta \setminus S_\eta}(A(S)) \leq \epsilon + \sqrt{\frac{2(1 - \eta)m \log \frac{1}{\delta}}{\delta}} + 8 \left(\frac{\epsilon}{\eta} + \frac{1}{m} \log \frac{1}{\delta'}\right)
\]

Through combining this with Lemma 5, with the probability of at least \(1 - 2\delta\), we have

\[
\mathcal{R}_D(A(S_\eta)) \leq \Omega + \sqrt{4\Omega \Delta + 8\Delta}
\]

\[
\text{(28)}
\]
The proof of Theorem 3 is finished. □

IX. CONCLUSION

In this article, we empirically and theoretically explored the relationship between DB variability and generalization in NNs, through the notions of algorithm DB variability and data DB variability, respectively. A significant negative correlation between the DB variability and generalization performance is observed in our experiments. As for the theoretical results, two lower bounds based on algorithm DB variability and an upper bound based on data DB variability are proposed, respectively, for the sake of enhancing our findings.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), vol. 25. Stateline, NV, USA, Dec. 2012, pp. 1097–1105.

[2] A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008.

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). Jun. 2016, pp. 770–778.

[4] T. B. Brown, “Language models are few-shot learners,” in Proc. Adv. Neural Inf. Process. Syst., vol. 33, 2020, pp. 1877–1901.

[5] H. Wang, N. Wang, and D. Y. Yeung, “Collaborative deep learning for recommender systems,” in Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2015, pp. 1235–1244.

[6] M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning. Cambridge, MA, USA: MIT Press, 2018.

[7] V. Vapnik, E. Levin, and Y. L. Cun, “Measuring the VC-dimension of a learning machine,” Neural Comput., vol. 6, no. 5, pp. 851–876, Sep. 1994.

[8] P. Bartlett and S. Mendelson, “Rademacher and Gaussian complexities: Risk bounds and structural results,” J. Mach. Learn. Res., vol. 3, pp. 463–482, Nov. 2002.

[9] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.

[10] C. Ma, S. Wojnowytsch, and L. Wu, “Towards a mathematical understanding of neural network-based machine learning: What we know and what we don’t,” 2020, arXiv:2009.10713.

[11] F. He and D. Tao, “Recent advances in deep learning theory,” 2020, arXiv:2012.10931.

[12] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning (still) requires rethinking generalization,” Commun. ACM, vol. 64, no. 3, pp. 107–115, Mar. 2021.

[13] M. Belkin, D. Hsu, S. Ma, and S. Mandal, “Reconciling modern machine-learning practice and the classical bias–variance trade-off,” Proc. Nat. Acad. Sci. USA, vol. 116, no. 32, pp. 15849–15854, Aug. 2019.

[14] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, “Deep double descent: Where bigger models and more data hurt,” 2019, arXiv:1912.02292.

[15] Q. Hu, H. Zhang, F. Gao, C. Xing, and J. An, “Analysis on the number of linear regions of piecewise linear neural networks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 2, pp. 644–653, Feb. 2022.

[16] H. Chen, F. He, S. Lei, and D. Tao, “Spectral complexity–scaled generalisation bound of complex–valued neural networks,” Artif. Intell., vol. 322, Sep. 2023, Art. no. 103951.

[17] J. Li, W. Su, and D. Sejdinovic, “Benign overfitting and noisy features,” 2020, arXiv:2008.02901.

[18] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in Proc. COMPSTAT’2010. Cham, Switzerland: Springer, 2010, pp. 177–186.

[19] M. Hardt, B. Recht, and Y. Singer, “Train faster, generalize better: Stability of stochastic gradient descent,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 1225–1234.

[20] F. He, T. Liu, and D. Tao, “Control batch size and learning rate to generalize well: Theoretical and empirical evidence,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 1143–1152.

[21] H. H. Tan and K. H. Lim, “Two-phase switching optimization strategy in deep neural networks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 1, pp. 330–339, Jan. 2022.

[22] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, “How to escape saddle points efficiently,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 1724–1732.

[23] K. Kawaguchi, “Deep learning without poor local minima,” in Proc. Advances Neural Inf. Process. Syst., vol. 29, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds. Curran, 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/5d99065c71294f51a5a183a23b9f0e-Paper.pdf

[24] H. Lu and K. Kawaguchi, “Depth creates no bad local minima,” 2017, arXiv:1702.08580.

[25] Y. Zhou and Y. Liang, “Critical points of neural networks: Analytical forms and landscape properties,” in Proc. Int. Conf. Learn. Represent., 2018.

[26] F. He, B. Wang, and D. Tao, “Piecewise linear activations substantially shape the loss surfaces of neural networks,” in Proc. Int. Conf. Learn. Represent., 2020.

[27] M. Goldblum, J. Geiping, A. Schwartzschild, M. Moeller, and T. Goldstein, “Truth or backpropaganda? An empirical investigation of deep learning theory,” in Proc. Int. Conf. Learn. Represent., 2020.

[28] Y. Jiang, V. Nagarajan, C. Baek, and J. Zico Kolter, “Assessing generalization of SGD via disagreement,” 2021, arXiv:2010.08127.

[29] Z. Ji and M. Telgarsky, “Directional convergence and alignment in deep learning,” in Proc. Adv. Neural Inf. Process. Syst., H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Red Hook, NY, USA: Curran Associates, 2020, pp. 17176–17186. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/c76e4b2fa5f48056719a5c0d14e2b9-Paper.pdf

[30] K. Lyu and J. Li, “Gradient descent maximizes the margin of homogeneous neural networks,” in Proc. Int. Conf. Learn. Represent., [Online]. Available: https://openreview.net/forum?id=SJel8KBP5

[31] L. Chizat and F. Bach, “Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss,” in Proc. Learn. Theory, 2020, pp. 1305–1338.

[32] P. Nakkiran, B. Neysabur, and H. Sedghi, “The deep bootstrap framework: Good online learners are good offline generalizers,” 2020, arXiv:2010.08127.

[33] Y. Jiang, V. Nagarajan, C. Baek, and J. Zico Kolter, “Assessing generalization of SGD via disagreement,” 2021, arXiv:2106.13799.

[34] Y. Qian, P. Expert, T. Rieu, P. Panzarasa, and M. Barahona, “Quantifying the alignment of graph and features in deep learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 4, pp. 1663–1672, Apr. 2022.

[35] Z. Q. John Xu, Y. Zhang, T. Luo, Y. Xiao, and Z. Ma, “Frequency principle: Fourier analysis sheds light on deep neural networks,” 2019, arXiv:1901.06523.
[38] D. Kalimeris et al., “SGD on neural networks learns functions of increasing complexity,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 3496–3506.

[39] H. He and W. Su, “The local elasticity of neural networks,” in Proc. Int. Conf. Learn. Represent., 2020.

[40] S. Fort, P. K. Nowak, S. Jastrzębski, and S. Narayanan, “Stiffness: A new perspective on generalization in neural networks,” 2019, arXiv:1901.09491.

[41] V. Papyan, X. Y. Han, and D. L. Donoho, “Prevalence of neural collapse during the terminal phase of deep learning training,” Proc. Nat. Acad. Sci. USA, vol. 117, no. 40, pp. 24652–24663, Oct. 2020.

[42] C. Fang, H. He, Q. Long, and W. J. Su, “Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training,” Proc. Nat. Acad. Sci. USA, vol. 118, no. 43, Oct. 2021, Art. no. e2105091118.

[43] W. He, B. Li, and D. Song, “Decision boundary analysis of adversarial examples,” in Proc. Int. Conf. Learn. Represent., 2018.

[44] H. Karimi, T. Derr, and J. Tang, “Characterizing the decision boundary of deep neural networks,” 2019, arXiv:1912.11460.

[45] H. Karimi and J. Tang, “Decision boundary of deep neural networks: Challenges and opportunities,” in Proc. 13th Int. Conf. Web Search Data Mining. Jan. 2020, pp. 919–920.

[46] M. Alfarra, A. Bibi, H. Hamoud, M. Gaafar, and B. Ghanean. (2020), On the Decision Boundaries of Deep Neural Networks: A Tropical Geometry Perspective. [Online]. Available: https://openreview.net/forum?id=BylIddNfW5

[47] S. Guan and M. Loew, “Analysis of generalizability of deep neural networks based on the complexity of decision boundary,” in Proc. 19th Int. Conf. Mach. Learn. Appl. (ICMLA), Dec. 2020, pp. 101–106.

[48] D. Mickisch, F. Assion, F. Greßner, W. Günther, and M. Motta, “Understanding the decision boundary of deep neural networks: An empirical study,” 2020, arXiv:2002.01810.

[49] G. Ortiz-Jimenez, A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard, “Hold me tight! Influence of discriminative features on deep network boundaries,” 2020, arXiv:2002.06349.

[50] H. Shah, K. Tanmuly, A. Raghuathan, P. Jain, and P. Netrapalli, “The pitfalls of simplicity bias in neural networks,” 2020, arXiv:2006.07710.

[51] P. Samangouei, A. Sadecil, L. Nakagawa, and N. Silberman, “Explain-GAN: Model explanation via decision boundary crossing transformations,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 666–681.

[52] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in Proc. Int. Conf. Learn. Represent., 2018.

[53] D. J. MacKay and D. J. M. Kay, Information Theory, Inference and Learning Algorithms. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[54] D. J. C. MacKay, “A practical Bayesian framework for backpropagation networks,” Neural Comput., vol. 4, no. 3, pp. 448–472, May 1992.

[55] E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and V. Fortuin, “Scalable marginal likelihood estimation for model selection in deep learning,” in Proc. NeurIPS, 2021.

[56] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

[57] S. Zhang and N. Komodakis, “Wide residual networks,” in Proc. Brit. Mach. Vis. Conf. (BMVC), R. C. Wilson, E. R. Hancock, and W. A. P. Smith, Eds. BMVA Press, Sep. 2016, pp. 87.1–87.12, Art. no. 87, doi: 10.5244/C.30.87.

[58] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, “Differentiable augmentation for data-efficient GAN training,” 2020, arXiv:2006.10738.

[59] C. Coleman et al., “Selection via proxy: Efficient data selection for deep learning,” in Proc. Int. Conf. Learn. Represent., 2020. [Online]. Available: https://openreview.net/forum?id=H1g2b0VYDv

[60] S. Lotfi, P. Izmailov, G. Benton, M. Goldblum, and A. Gordon Wilson, “Bayesian model selection, the marginal likelihood, and generalization,” 2022, arXiv:2002.11678.

[61] H. Zhang, Y. N. Dauphin, and T. Ma, “Fixup initialization: Residual learning without normalization,” 2019, arXiv:1901.09321.

[62] A. Immer, M. Bauer, V. Fortuin, G. Rätsch, and K. M. Emtyiyaz, “Scalable marginal likelihood estimation for model selection in deep learning,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 4563–4573.

[63] S. Shalev-Shwartz and S. Ben-David, Understanding Machine Learning: From Theory to Algorithms. Cambridge, U.K.: Cambridge Univ. Press, 2014.