Recent advances in deep learning theory

Fengxiang He * Dacheng Tao *

Abstract

Deep learning is usually described as an experiment-driven field under continuous criticizes of lacking theoretical foundations. This problem has been partially fixed by a large volume of literature which has so far not been well organized. This paper reviews and organizes the recent advances in deep learning theory. The literature is categorized in six groups: (1) complexity and capacity-based approaches for analyzing the generalizability of deep learning; (2) stochastic differential equations and their dynamic systems for modelling stochastic gradient descent and its variants, which characterize the optimization and generalization of deep learning, partially inspired by Bayesian inference; (3) the geometrical structures of the loss landscape that drives the trajectories of the dynamic systems; (4) the roles of over-parameterization of deep neural networks from both positive and negative perspectives; (5) theoretical foundations of several special structures in network architectures; and (6) the increasingly intensive concerns in ethics and security and their relationships with generalizability.

Keywords: neural networks, deep learning, deep learning theory.

1 Introduction

Deep learning can be broadly defined as a family of algorithms employing artificial neural networks to discover knowledge from experience for predictions or decision-making [138]. Canonical forms of the experience can be human-annotated electronic records as a dataset or the interactions between the learner or the electronic environment depending on the scenarios [169]. A normal artificial neural network in deep learning usually connects a sequence of weight matrix followed by a nonlinear activation function as a network, which is typical of a considerably large parameter size.

The term of deep learning was introduced to machine learning by Dechter [62] and then to brain-inspired algorithms by Aizenberg et al. [5], while several major concepts wherein can be dated back to early 1940s. The research of deep learning saw two rises and then falls in 1940s-1960s [162, 111, 199] and 1980s1990s [201]. The third and current wave starts in 2006 [24, 114, 196] and lasts till now. The recent wave has substantially reshaped many real-world application domains, including computer vision [110], natural language processing [63, 184], speech processing [64], 3D point cloud processing [98], data mining [232], recommender system [247], autonomous vehicles [152, 215], medical diagnosis [135, 209], and drug discovery [43].

To date, however, the development of deep learning heavily relies on experiments but without solid theoretical foundations. Many facets of the mechanism of deep learning still remain unknown. We are continuously surprised by finding heuristic approaches can achieve excellent performance across extensive areas, though sometimes also considerably instable. Meanwhile, the suggested intuitions are usually left un-verified or

*The authors are with UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering, the University of Sydney, Darlington NSW 2008, Australia. Email: fengxiang.he@sydney.edu.au and dacheng.tao@sydney.edu.au.
even un-checked. Such practice is tolerated and has become common in deep learning research. This black-box nature introduces unknown risks to deep learning applications. This unawareness considerably undermines our capabilities in identifying, managing, and preventing the algorithm-led disasters, and further severely hurts the confidence in applying the recent progress to many industrial sectors, especially those in security-critical areas, such as autonomous vehicles, medical diagnosis, and drug discovery. This also backlashes on the future development of innovative deep learning algorithms designing.

A major part of potential theoretical foundations is the generalizability, which refers to the capability of models well trained on the training data by deep learning algorithms predicting on the unseen data [224, 169]. Since the training data can not cover every future circumstance, good generalizability secures the learned model to be able to handle unseen events. This is particularly important where long-tail events constantly appear and are possible to cause fatal disasters.

Statistical learning theory has established theories for the generalizability relying on the hypothesis complexity [224, 169]. Could these tools solve the problem in deep learning theory? The answer is no in general. Conventional tools usually develops generalization bounds based on hypothesis complexities, such as VC-dimension [28, 223], Rademacher complexity [130, 129, 21], and covering number [73, 104]. These complexities, in classic results, heavily rely on the model sizes. This introduces the Occam’s razor principle:

\textit{Plurality should not be posited without necessity;}

\textit{i.e., one needs to find a model sufficiently small to prevent overfitting as long as the model can fit the training sample. However, deep learning models usually have extremely large model sizes, which sometimes makes the generalization bound even larger than the potential maximal value of the loss function. Moreover, the Occam’s razor principle suggests a positive correlation between the generalizability and the model size, which is no longer observed in deep learning. In contrast, deeper and wider networks often have advantaged performance [38]. The irreconciliation between the excellent generalizability of deep learning and its extreme over-parameterization is like a “cloud” to conventional complexity-based learning theory.}

1.1 A brief overview of the status quo

Early works have attempted to establish the theoretical foundations of deep learning [172, 90, 22, 20, 23, 158, 11] but largely stagnated with the dip of the wider deep learning research.

The recent story started from the work by Zhang et al. [244] in 2017. The authors conducted systematic experiments to explore the generalization ability of deep neural networks. They show that neural networks can almost perfectly fit the training data even when the training labels are random. This paper attracts the community of learning theory to the important topic that how to theoretically interpret the success of deep neural networks. Kawaguchi et al. [122] discuss many open problems regarding the excellent generalization ability of deep neural networks despite the large capacity, complexity, possible algorithmic instability, non-robustness, and sharp minima. The authors also provide some insights to solve the problems. Since then, the importance of deep learning theory has been widely recognized. A large volume of literature has emerged to establish theoretical foundations of deep learning. In this paper, we review the related literature and organize them into following six categories.

**Complexity and capacity-based method for analyzing the generalizability of deep learning.** Conventional statistical learning theory has established the a series of upper bounds on the generalization error (generalization bounds) based on the complexity of hypothesis space, such as VC-dimension [28, 223], Rademacher complexity [130, 129, 21], and covering number [73, 104]. Usually, these generalization bounds explicitly rely on the model size. They suggest that controlling the model size can help models
generalize better. However, the colossal model sizes of deep learning models also make the generalization bounds vacuous. It is thus much desired if we can develop size-independent hypothesis complexity measure and generalization bounds. A promising idea is to characterize the complexity of the “effective” hypothesis space which can be learned in deep learning. The effective hypothesis space can be significantly smaller than the whole hypothesis space. We thereby can expect to obtain a much smaller generalization guarantee.

**Stochastic differential equations (SDE) for modelling stochastic gradient descent (SGD) and its variants, which dominate the optimization algorithms in deep learning.** The dynamic systems of these SDEs determine the trajectory of the weights in training neural networks and their steady distributions represent the learned networks. Through the SDEs and their dynamics, many works establish guarantees for the optimization and generalization of deep learning. The “effective” hypothesis space is exactly the hypothesis space “that can be found by SGD”. Thus, studying the generalizability of deep learning via SGD would be straightforward. Moreover, this family of methods are partially inspired by Bayesian inference. This echoes the previous story of variation inference which solves Bayesian inference in an optimization manner in order to address the scaling issue. This interplay between stochastic gradient methods and Bayesian inference would help advance both areas.

**The geometrical structures of the highly complex empirical risk landscape that drive the trajectories of the dynamic systems.** The geometry of the loss landscape plays major roles in driving the trajectories of the SDEs: (1) the derivatives of the loss are components of the SDEs; and (2) the loss would serve as boundaries conditions of the SDEs. Therefore, understanding the loss surface would be a key step in establishing theoretical foundations for deep learning.

In convention, the learnability and optimization ability are usually guaranteed in “regularized” problems. The “regularization” can be characterized by a variety of terms, including convexity, Lipschitz continuity, and differentiability. However, these factors are no longer secured in deep learning, at least not obviously. Neural networks usually consist of a large number of nonlinear activations. The nonlinearity in activations makes the loss surface extremely non-smooth and non-convex. The established guarantees for convex optimization become invalid. The prohibitive complexity of loss surface checks the community touching the geometry of the loss landscape and even deep learning theory for a long time. However, the complex geometry of the loss surface exactly characterizes the behavior of deep learning. It is the “high way” to understand deep learning via its loss surface.

**The roles of over-parameterization of deep neural networks.** Over-parameterization is usually treated as a major barrier in developing meaningful generalization bounds for deep learning via complexity-based approaches. However, recent studies suggest that over-parameterization would make major contributes in shaping the loss surface of deep learning - making the loss surface more smooth and even “analogous” convex. Moreover, many works have proven that neural networks in the extreme over-parameterization regime are equivalent to some simpler models, such as Gaussian kernels.

**Theoretical foundations for several special structures in network architectures.** In the previous review, we focus on the result generally stand for all neural networks. Meanwhile, deep neural networks designing involves many special techniques. These structures also contribute significantly to deep learning’s excellent performance. We review the theoretical results on convolutional neural networks, recurrent neural networks, and networks for permutation invariant/equivariant functions.

**The intensive concerns in ethics and security and their relationships with deep learning theory.** Deep learning has been deployed in an increasingly wide spectrum of application domains. Some of them

---

1It is worth noting that the “regularized” here or the “regularization” later refer to the approaches to make ill-posed problems solvable. It is not necessarily relating to adding “regularizers”. 
involve highly private personal data, such as images and videos on mobile phones, health data, and final
records. Some other scenarios may require deep learning to deliver highly sensitive decision-making, such
as mortgage approval, college admission, and credit assessment. Moreover, deep learning models have been
shown vulnerable to adversarial examples. How to secure a deep learning system from breaches related to
privacy preservation, fairness concerning, and adversarial attacks are of significant importance.

1.2 Overview of this paper

This paper reviews the recent advances in establishing the theoretical foundations of deep learning. We
acknowledge that a few papers have reviewed deep learning theory. Alom et al. [9] give a survey of the
[81] summarize results and challenges related to approximation and Rademacher complexity, loss surface,
and the convergence and implicit regularization of optimization in deep learning. Our survey is the most
comprehensive. We organize the literature in a unique view and provide new insights for future works.

The excellent generalizability of deep learning is like a “cloud” to conventional complexity-based learning
theory: the over-parameterization of deep learning makes almost all existing tools vacuous. Existing works
try to tackle this problem through three major paths: (1) developing size-independent complexity measures,
which can characterize the complexity of the effective hypothesis space that can be learned, instead of
the whole hypothesis space. The related works are discussed in Section 2; and (2) modelling the learned
hypothesis through stochastic gradient methods, the dominant optimizers in deep learning, based on the
stochastic differential functions and the geometry of the associated loss functions. The related works are
reviewed in Sections 3 and 4; and (3) over-parameterization surprisingly bring many good properties to the
loss functions and further secures the optimization and generalization performances. The related works are
given in Section 5. Parallel to these, Section 6 reviews the theoretical foundations on special structures of
network architecture.

Another important facet of machine learning is the rising concern on the ethical and security issues, including
privacy preservation, adversarial robustness, and fairness protection. Specifically, privacy preservation and
adversarial robustness have been discovered closely related to generalizability: a good generalizability usually
means a good privacy-preserving ability; and more robust algorithms might have. Efforts of understanding
the interplay between these issues in the scenario of deep learning are also discussed in this paper. The
related works are discussed in Section 7.

1.3 Notations

Suppose $S = \{(x_1, y_1), \ldots, (x_N, y_N)\mid X_i \in \mathcal{X} \subset \mathbb{R}^{d_X}, Y_i \in \mathcal{Y} \subset \mathbb{R}^{d_Y}, i = 1, \ldots, N\}$ is a training sample
set, where $d_X$ and $d_Y$ are the dimensions of the feature $X$ and the label $Y$, respectively. For the brevity, we
define $z_i = (x_i, y_i) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. All $z_i$ are independent and identically distributed (i.i.d.) observations
of the random variable $Z = (X, Y) \in \mathcal{Z}$. Machine learning algorithms are designed to learn hypothesis $h$
from the training data; all potential hypotheses constitute a hypothesis space $\mathcal{H}$.

A neural network is usually defined as the following form,

$$x \mapsto W_D \sigma_{D-1}(W_{D-1} \sigma_{D-2}(\ldots \sigma_1(W_1 x))),$$

where $D$ is the depth of the neural network, $W_j$ is the $j$-th weight matrix, and $\sigma_j$ is the $j$-th nonlinear activation
function. Usually, we assume the activation function $\sigma_j$ is a continuous function between Euclidean spaces
and $\sigma_j(0) = 0$. Popular activations in practice including softmax, sigmoid, and tanh functions.
Generalization bound measures the generalization ability of an algorithm. For any hypothesis $h$ learned by an algorithm $A$, the expected risk $R(h)$ and empirical risk $\hat{R}_S(h)$ with respect to the training sample set $S$ are respectively defined as follows,

$$R(h) = \mathbb{E}_Z l(h, Z), \quad \hat{R}_S(h) = \frac{1}{N} \sum_{i=1}^{N} l(h, z_i).$$

Additionally, when the output hypothesis $h$ of algorithm $A$ is stochastic, we usually calculate the expectations of the expected risk $R(h)$ and empirical risk $\hat{R}_S(h)$ with respect to the randomness introduced by the algorithm $A$ as follows,

$$R(A) = \mathbb{E}_{A(S)} R(A(S)), \quad \hat{R}_S(A) = \mathbb{E}_{A(S)} \hat{R}_S(A(S)),$$

where $A(S)$ is the hypothesis learned by the algorithm $A$ on the sample $S$. Then, the generalization error can be defined as the difference between the expected risk and empirical risk.

## 2 Capacity and complexity: Advances and predicament

Conventional statistical learning theory establishes generalization guarantees based on the complexity of the hypothesis space. Following this line, some works have proposed upper bounds of the generalization error of deep learning.

### 2.1 Worst-case bounds based on VC-dimension

A major measure for evaluating the hypothesis complexity in conventional statistical learning theory is the Vapnik-Chervonenkis dimension (VC-dimension, $\text{VCdim}([28, 223])$) which is defined as follows.

**Definition 1** (growth function, shattering, and VC-dimension). For any non-negative integer $m$, the growth function of hypothesis space $H$ is defined as follows,

$$\Pi_H(m) := \max_{x_1, \ldots, x_m \in \mathcal{X}} |\{(h(x_1), \ldots, h(x_m)) : h \in \mathcal{H}\}|.$$

If $\Pi_H(m) = 2^m$, we say $\mathcal{H}$ shatters the dataset $\{x_1, \ldots, x_m \in \mathcal{X}\}$. We define the VC-dimension $\text{VCdim}(\mathcal{H})$ as the largest size of shattered set.

One can obtain a uniform generalization bound via the VC-dimension as follows.

**Theorem 1.** Assume hypothesis space $\mathcal{H}$ has VC-dimension $\text{VCdim}(\mathcal{H})$. Then, for any $\delta > 0$, with probability $1 - \delta$, the following inequality holds for any $h \in \mathcal{H},$

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{2\text{VCdim}(\mathcal{H}) \log \frac{em}{\text{VCdim}(\mathcal{H})}}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}},$$

where $m$ is the training sample size.

Thus, results on the VC-dimension of neural networks can characterize its generalizability. Goldberg and Jerrum [90] gives an $O(W^2)$ upper bound for the VC dimension of neural networks with parameter size $W$ and depth $D$. It was improved to $O(W \log(WD))$ by Bartlett and Williamson [22] and then to $O(WL \log W + WL^2)$ by Bartlett et al. [20]. Other on VC-dimension were also given by Baum and Haussler [23] and Maass [158].
The tightest upper bound for the VC-dimension so far is proven by Harvey et al. [102] as follows,

**Theorem 2.** Consider a neural network with \( W \) parameters and \( U \) units with activation functions that are piecewise polynomials with at most \( p \) pieces and of degree at most \( d \). Let \( F \) be the set of (real-valued) functions computed by this network. Then,

\[
\text{VCdim}(\text{sgn}(F)) = O(WU \log((d + 1)p)).
\]

However, such upper bounds on the VC-dimensions all heavily rely on the model size \( W \), which is extremely large in deep learning. In some cases, the generalization bound can be significantly larger than the largest potential value of the loss function (such as 1 for the 0-1 loss).

Harvey et al. [102] also give a lower bound for the VC-dimension as the following theorem.

### 2.2 Margin bounds: From exponential depth dependence to depth independence

Margin bounds are another family of generalization bounds [223, 207, 17, 131, 219]. Compared with the worst-case bounds based on the VC-dimension, margin bounds deliver strong guarantees for the learned models that they can achieve small empirical margin loss for large confidence margin. Similar generalization guarantees can be obtained via the luckiness [207, 131].

**Margin bound.** For any distribution \( D \) and margin \( \gamma > 0 \), we define the expected margin loss for hypothesis \( h \) as follows.

\[
L_\gamma(h) = P(h(x) \sim D) \left( h(x)[y] \leq \gamma + \max_{j \neq y} h(x)[j] \right),
\]

where \( h(x)[j] \) is the \( j \)-th component of the vector \( h(x) \). The generalization bounds under the margin loss are called margin bounds.

One can prove a margin bound via the the covering number [73, 104] or Rademacher complexity [130, 129, 21] of the hypothesis space defined as follows.

**Definition 2 (covering number).** The covering number of covering number \( \mathcal{N}(\mathcal{H}, \epsilon, \| \cdot \|) \) of space \( \mathcal{H} \) is defined to be the least cardinality of any subset \( V \subset \mathcal{H} \) that covers \( \mathcal{H} \) at scale \( \epsilon \) under metric \( \| \cdot \| \); i.e.,

\[
\sup_{A \in \mathcal{H}} \min_{B \in V} \| A - B \| \leq \epsilon.
\]

**Definition 3 (empirical Rademacher complexity and Rademacher complexity).** Given a real-valued function class \( \mathcal{H} \) and a dataset \( S \), the empirical Rademacher complexity is defined as follows,

\[
\hat{\mathcal{R}}(\mathcal{H}) = E_{\epsilon} \left[ \sup_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \epsilon_i h(x_i) \right],
\]

where \( \epsilon = \{\epsilon_1, \ldots, \epsilon_m\} \) is a random vector distributed in the space \( \{-1, +1\}^m \). Further, we can define Rademacher complexity as follows,

\[
\mathcal{R}(\mathcal{H}) = E_{S} \hat{\mathcal{R}}(\mathcal{H}).
\]

Intuitively, covering number indicates how many balls are needed to cover the hypothesis space; and Rademacher complexity measures the maximal correlations between the output of a hypothesis and a noise vector, and thus characterizes the “goodness-of-fit” of the hypothesis space to a noise. Apparently, a larger covering number or Rademacher complexity corresponds to a larger hypothesis complexity.
One can also upper bound the Rademacher complexity via the covering number through the Dudley entropy integral \cite{169} as follows.

**Theorem 3** (Dudley entropy integral). Let \( \mathcal{H} \) be a real-valued function class taking values in \([0, 1]\), and assume that \( 0 \in \mathcal{H} \). Then,

\[
\hat{\mathcal{R}}(\mathcal{H}) \leq \inf_{\alpha > 0} \left( \frac{4\alpha}{\sqrt{m}} + \frac{12}{m} \int_0^{\sqrt{m}} \sqrt{\log \mathcal{N}(\mathcal{H}, \epsilon, \| \cdot \|_2)} d\epsilon \right).
\]

Besides, one can obtain margin bounds via PAC-Bayesian theory as below, which corporates PAC theory and Bayesian statistics \cite{160, 161}.

**Theorem 4** (PAC-Bayesian theory). Suppose the prior and the posterior of the model parameter \( \theta \) is \( \mathcal{P} \) and \( \mathcal{Q} \). Then,

\[
\mathbb{E}_\theta \mathcal{R}(h(\theta)) = \mathbb{E}_\theta \hat{\mathcal{R}}(h(\theta)) + \sqrt{\frac{K\ell(Q\|P) + \log \frac{m}{\delta}}{2(m - 1)}}. \tag{1}
\]

When the input data \( \| x \| \leq B \) and the Frobenius norms of the \( D \) weight matrices are upper bounded by \( M_F(1), \ldots, M_F(D) \), respectively, Neyshabur et al. \cite{177} prove via Rademacher complexity that with high probability, the generalization error scales as

\[
\mathcal{O}\left( \frac{B^2 D^2 \prod_{j=1}^D M_F(j)}{\sqrt{m}} \right). \tag{2}
\]

However, this bound is exponentially dependent with the depth.

Then, through covering number, Dudley entropy integral, and Rademacher complexity, Bartlett et al. \cite{19} obtain the following spectrally-normalized upper bound for the generalization error,

\[
\mathcal{O}\left( \frac{\| X \|_2 \log W}{\gamma m} R_A + \frac{1/\delta}{m} \right),
\]

where

\[
R_A = \left( \prod_{i=1}^D \rho_i \| A \|_\sigma \right) \left( \sum_{i=1}^D \frac{\| A_i^T - M_i^T \|_{2/3}}{\| A \|_{2/3}^{2/3}} \right)^{3/2}.
\]

This bound is strictly tighter than Neyshabur et al. \cite{177}. Moreover, this bound suggests that one can boost the generalizability via controlling the spectral norm of the network. This explains spectral normalization \cite{168}, an effective regularizer mainly used in generative adversarial networks, and singular value bounding \cite{118} for boosting neural network training performance.

Concurrently, Neyshabur et al. \cite{176} prove a similar generalization bound as below via the PAC-Bayesian theory,

\[
\mathcal{O}\left( \sqrt{\frac{B^2 D^2 W \log(DW) \prod_{i=1}^D \frac{\| W_i \|_2^2}{\| W_i \|_2^4} + \log \frac{Dm}{\delta}}}{\gamma^2 m} \right).
\]

The proofs are in two-step: (1) they prove a perturbation bound which controls the changes of the output when perturbation introduced to the weights. Thereby, we can bound the sharpness of the network; and (2)
the perturbation bound further leads to a generalization bound via the PAC-Bayesian theory. However, this bound has been shown strictly weaker than the one given by Bartlett et al. [19].

Through introducing some norm constraints to the weights, Golowich et al. [92] improve the exponential depth dependence of eq. (2) by Neyshabur et al. [177] to polynomial dependence as follows,

$$\mathcal{O} \left( B \sqrt{D} \prod_{j=1}^{D} M_F(j) \right),$$

or

$$\mathcal{O} \left( B \sqrt{D} + \log m \prod_{j=1}^{D} M_F(j) \right).$$

When assuming

$$\prod_{j=1}^{D} \|W_j\|^2 \leq M,$$

where $M > 0$ is a constant and $W_j$ is the $j$-th weight matrix, Golowich et al. [92] further improves the bounds to depth-independent as follows,

$$\tilde{\mathcal{O}} \left( B \left( \prod_{j=1}^{D} M_F(j) \right) R_G \right),$$

where

$$R_G = \min \left\{ \sqrt{\frac{\log \left( \frac{1}{p} \prod_{j=1}^{D} M_F(j) \right)}{\sqrt{m}}}, \sqrt{\frac{D}{m}} \right\},$$

and $\Gamma$ is the upper bound of the spectral norm of all the weight matrices.

Moreover, Golowich et al. [92] also prove a lower bound as follows,

$$\Omega \left( B \prod_{j=1}^{D} M_F(j) \max \{0, \frac{1}{2} - \frac{1}{p} \} \right).$$

It suggests that a tight bound would be $\mathcal{O} \left( \frac{1}{\sqrt{m}} \right)$.

### 3 Stochastic gradient descent: An implicit regularization

The over-parameterization makes a colossal hypothesis space. Despite that Golowich et al. [92] have provided generalization bounds explicitly independent with the network size, one might observe some implicit dependence. For example, the factor $M_F(j), j = 1, \ldots, D,$ are the upper bounds of weight norms which might be also influenced by the model size.

Stochastic gradient methods (SGM), including stochastic gradient descent (SGD) and its variants, are the dominant optimization methods in deep learning. They only explore a small part of the hypothesis space. This performs an implicit regularization to the deep learning models that controls the effective model capacity in an algorithm-dependent and data-dependent manner. Therefore, approaching algorithm-dependent and
data-dependent bounds via SGM may overcome the conventional statistical learning theory based on VC
dimension or Rademacher complexity.

3.1 Stochastic gradient methods

To optimize the expected risk, a natural tool is the gradient descent (GD). Specifically, the gradient of the
empirical risk in terms of the parameter \( \theta \) below,

\[
g(\theta(t)) \overset{\Delta}{=} \nabla_{\theta(t)} \mathcal{R}(\theta(t)) = \nabla_{\theta(t)} \mathbb{E}_{(X,Y)} l(F_{\theta(t)}(X), Y),
\]

and the corresponding update equation are defined as follows,

\[
\theta(t+1) = \theta(t) - \eta g(\theta(t)),
\]

where \( \theta(t) \) is the parameter at the iteration \( t \) and \( \eta > 0 \) is the learning rate.

Stochastic gradient descent (SGD) estimates the gradient from mini batches of the training sample set to
estimate the gradient \( g(\theta) \). Let \( S \) be the indices of a mini batch, in which all indices are independently
and identically (i.i.d.) drawn from \( \{1, 2, \ldots, N\} \), where \( N \) is the training sample size. Then similar to the
gradient, the iteration of SGD on the mini-batch \( S \) is defined as follows,

\[
\hat{g}_S(\theta(t)) = \nabla_{\theta(t)} \hat{\mathcal{R}}(\theta(t)) = \frac{1}{|S|} \sum_{n \in S} \nabla_{\theta(t)} l(F_{\theta(t)}(X_n), Y_n),
\]

and

\[
\theta(t+1) = \theta(t) - \eta \hat{g}(\theta(t)),
\]

where

\[
\hat{\mathcal{R}}(\theta) = \frac{1}{|S|} \sum_{n \in S} l(F_{\theta}(X_n), Y_n)
\]

is the empirical risk on the mini batch and \( |S| \) is the cardinality of the set \( S \). For brevity, we rewrite
\( l(F_{\theta}(X_n), Y_n) = l_n(\theta) \)
in the rest of this paper.

Also, suppose that in step \( i \), the distribution of parameter is \( Q_i \), the initial distribution is \( Q_0 \), and the
convergent distribution is \( Q \). Then SGD is used to find \( Q \) from \( Q_0 \) through a series of \( Q_i \).

3.2 Generalization bounds on convex loss surfaces

When the loss is continuous and convex, the generalizability of SGMs has been intensively studied.

Ying and Pontil [238] prove the generalization bound in the following rate, when the step size is fixed:

\[
\eta_t \asymp m^{-\frac{2\zeta}{2\zeta+1}},
\]

\[
\mathcal{O} \left( m^{-\frac{2\zeta}{2\zeta+1}} \log m \right),
\]

where \( \zeta \) is a constant.
Then, Dieuleveut et al. [65] improves the result to the following rate for the average, when $2\zeta + \gamma > 1$,

$$O\left(m^{-\min\left(\frac{1}{\zeta}, 1\right) + \gamma}\right).$$

Other related works include Lin et al. [149]; Lin and Rosasco [150]; Chen et al. [50]; Wei et al. [227]. However, the convex assumption for the loss surface is usually too strong in deep learning. However, the loss surface is usually highly non-convex in deep learning. This makes the aforementioned results invalid.

### 3.3 Generalization bounds on non-convex loss surface

Results on non-convex loss surface have also been seen in the literature. Related works include generalization bounds via algorithmic stability and PAC-Bayesian theory.

**Algorithmic stability.** Bousquet and Elisseeff [29] propose to employ algorithmic stability to measure the stability of the output hypothesis when the training sample set is disturbed. Algorithmic stability has many versions. Here, we introduce a popular one as follows.

**Definition 4** (uniform stability; cf. [29], Definition 6). A machine learning algorithm $A$ is uniformly stable, if for any neighboring sample pair $S$ and $S'$ which are different by only one example, we have the following inequality,

$$|\mathbb{E}_{A(S)} l(A(S), Z) - \mathbb{E}_{A(S')} l(A(S'), Z)| \leq \beta,$$

where $Z$ is an arbitrary example, $A(S)$ and $A(S')$ are the output hypotheses learned on the training sets $S$ and $S'$, respectively, and $\beta$ is a positive real constant. The constant $\beta$ is called the uniform stability of the algorithm $A$.

It is natural that an algorithm insensitive to the disturbance the training data have good generalization abilities. Following this intuition, some generalization bounds have been proved based on the algorithmic stability [29, 236]. For example, based on the uniform stability, Bousquet and Elisseeff [29] prove the following theorem for the following generalization in expectation.

**Theorem 5.** Let algorithm $A$ is $\beta$-uniformly stable. Then,

$$|\mathbb{E}_{S,A}[\hat{R}[A(S)] - \mathbb{R}[A(S)]]| \leq \beta.$$

Recent works usually model SGM via stochastic differential equations (SDEs). Mini-batch stochastic gradients introduce randomness into SGM (i.e., the stochastic gradient can be decomposed into sample-estimation of the gradient on the whole training set plus a noise). So the weight update in SGM can be formulated as a stochastic process. The output hypothesis is then drawn from the steady distribution of this stochastic process. We can eventually analyze the generalization and optimization of deep learning via the steady distribution.

SGM is usually formulated by eqs. (3) and (4). The stochastic gradient introduces gradient noise into the weight updates. When the gradient noise is modeled by a Gaussian distribution, SGM reduces to stochastic gradient Langevin dynamics (SGLD) as follows [159],

$$\Delta \theta(t) = \theta(t + 1) - \theta(t) = -\eta \hat{g}_S(\theta(t)) = -\eta g(\theta) + \sqrt{\frac{2\eta}{\beta}} \Delta W, \Delta W \sim N(0, I),$$

where $\eta$ is the learning rate and $\beta > 0$ is the inverse temperature parameter.
Hardt et al. [101] prove the following upper bound for uniform stability, which exponentially depends on aggregated step sizes and smoothness parameter.

**Theorem 6.** Suppose the loss function is $\epsilon$-smooth, $L$-Lipschitz, and not larger than 1 for every data. We run SGM with monotonically increasing step sizes $\alpha_t \leq c/t$ for $T$ steps. Then, SGM is uniform stable with

$$\beta \leq \frac{1 + 1/\beta c}{m - 1} \left(2cL^2\right)^{1/\beta c} T^{1/\beta c}.$$ 

This proof of this theorem mimics the proofs for convergence. Suppose the loss function $l$ is $L$-Lipschitz constant with respect to the weight for any example; i.e.,

$$|l(w; z) - l(w'; z)| \leq L\|w - w'\|.$$ 

In other words, the stabilities can be measured by the stability of the weight. Therefore, one can just analyze how the weight diverges as a function of time $t$. The authors then deliver generalization bounds relying on the step sizes and the iteration numbers. Further, we can easily obtain a generalization bound based on this theorem.

Under five more assumptions, Raginsky et al. [194] prove the following upper bound for the expected excess risk,

$$\tilde{O} \left( \frac{\beta(\beta + U)^2}{\lambda_*} d^{1/4} \log \left( \frac{1}{\epsilon + \epsilon} \right) \right) + \tilde{O} \left( \frac{(\beta + U)^2}{\lambda_* + m} + \frac{U \log(\beta + 1)}{\beta} \right),$$

provided,

$$k = \tilde{\Omega} \left( \frac{\beta(\beta + U)}{\lambda_* \epsilon^4} \log^5 \left( \frac{1}{\epsilon} \right) \right),$$

and

$$\eta \leq \left( \frac{\epsilon}{\log(1/\epsilon)} \right)^4,$$

where $U$ is the parameter size, $m$ is the training sample size, the inverse temperature parameter $\beta > 1 \lor 2/m$, $\lambda_*$ is a spectral gap parameter that controls the exponential rate of convergence of the dynamic of SGM to its stationary distribution, $d$ is the factor concerning the dissipativity of the hypothesis $h$; i.e., for any datum $z$ and any parameter $w$, for some $m > 0$ and $b > 0$,

$$< w, \nabla h(w, z) > \leq m\|w\|^2 - b,$$

and

$$\epsilon \in \left( 0, \frac{m}{4M^2} \land e^{-\tilde{\Omega} \left( \frac{\lambda_*}{\beta(\beta + U)} \right)} \right).$$

The proofs are completed in the following three steps: (1) when the step size is small enough, the authors prove that the dynamics of SGLD converge to continuous-time Langevin dynamic processes under 2-Wasserstein distance; (2) the authors then show that the Langevin dynamic then converge to a Gibbs distribution when the training time has been sufficiently long; and (3) the authors construct a Gibbs algorithm where the output hypothesis is almost an empirical risk minimizer. Also, the Gibbs algorithm is proven to be algorithmic stable under 2-Wasserstein distance. However, this result is based on the convergence of the weight distribution to the stationary distribution, which usually exponentially relies on the model dimension.
A canonical mutual information-based generalization bound is given in Xu and Raginsky [235],
\[ \sqrt{\frac{2\sigma^2}{m} I(S; \theta)}, \]  
(5)
where \( S \) is the training sample, \( \theta \) is the model parameter, and the loss is \( \sigma \)-sub-Gaussian under the data distribution. This result stems from the following result on the informational theoretical “stability” of a two-input function \( f(X, Y) \),
\[ |f(X, Y) - f(\bar{X}, \bar{Y})| \leq \sqrt{2\sigma^2 I(X; Y)}, \]
if the \( f(\bar{X}, \bar{Y}) \) is sub-Gaussian under distribution \( P_{X,Y} = P_X \otimes P_Y \) where \( P_X \) and \( P_Y \) are distributions of \( X \) and \( Y \), respectively. It is worth noting that the mutual information is calculated between the learned weights and all training data; it is an “on-average” estimate of how much information has been transformed into the learned model from the data.

Based on this result, Pensia et al. [190] gives a generalization error bound for SGLD,
\[ \sqrt{\frac{R^2}{m} \sum_{t=1}^{T} \frac{\eta_t^2 L^2}{\beta_t^2}}, \]  
(6)
where the hypothesis is assumed to \( R \)-sub-Gaussian, \( \eta_t \) is the learning rate in the \( t \)-th iteration, \( L \) is the upper bound of the weight update \( \Delta \theta_t \), and the gradient noise is assumed to be \( N(0, \sigma_t^2 I) \). This result is then extended by Bu et al. [33]:

- Assume the loss \( l(\theta, z) \) is \( R \)-sub-Gaussian under \( z \sim \mu \), where \( z_i \) is the \( i \)-th datum, \( \theta \) is the model parameter, and \( \mu \) is the data generating distribution. Then, the generalization error has the following upper bound,
\[ \frac{1}{m} \sum_{i=1}^{m} \sqrt{2R^2 I(\theta; z_i)}, \]  
(7)
where \( m \) is the sample size.

- Assume the loss \( l(\bar{\theta}, \bar{z}) \) is \( R \)-sub-Gaussian under distribution
\[ P_{\bar{\theta}, \bar{z}} = P_{\theta} \otimes P_z. \]

Then, the generalization error has the same upper bound (eq. 7).

In contrast with Xu and Raginsky [235], Bu et al. [33] proposes an “individual” version metric to measure the information transformed into the learned model from the data; see eq. (7) where the whole training sample \( S \) is replaced by individual datums \( I(\theta; z_i) \).

Mou et al. [171] models SGLD as a Langevin diffusion dynamic and then prove an \( \mathcal{O}(1/m) \) generalization bound and an \( \mathcal{O}(1/\sqrt{m}) \) generalization bound for SGLD via the stability and the PAC-Bayesian theory, respectively:

- **Algorithmic stability:** Assume that the loss function is upper bounded by \( C \), the hypothesis is \( L \)-Lipschitz continuous, and the step size \( \eta_t \leq \frac{\log^2 2}{\beta t^2} \) in any iteration \( t \), the generalization error has the
following upper bound in average,

\[ O \left( \frac{2LC}{m} \left( \beta \sum_{t=1}^{N} \eta_t \right)^2 \right). \]

- **PAC-Bayesian theory**: Assume the empirical risk is regularized by the \( l_2 \) norm of the weight; i.e., \( \frac{1}{2} \| \theta \|^2 \), and the loss function is sub-Gaussian. Then, we have the following high-probability generalization bound,

\[ O \left( \frac{\beta}{m} \sum_{k=1}^{N} \eta_k e^{-\frac{\beta}{2} (T_N - T_k) \mathbb{E} [\| g_k (\theta_k) \|^2]} \right), \]

where \( N \) is the iteration number, \( T_k = \sum_{i=1}^{k} \eta_i \), \( \beta \) is the temporal parameter, \( g_k (\theta_k) \) is the gradient in the \( k \) iteration.

Recently, Chen et al. [51] proves a trade-off between convergence and stability for all iterative algorithms respectively under convex smooth setting and strong convex smooth setting, which leads to an \( O (1/m) \) generalization bound. Other advances include the works by He et al. [106], Tzen [222], Zhou [254], Wen et al. [230], and Li et al. [145].

### 3.4 Generalization bounds relying on data-dependent priors

In Bayesian statistics, priors are given according to the “practice experience” or domain knowledge. In algorithm designing, one usually cannot assume they have understood the data. Thus, the prior needs to be independent with the training sample. Bayesian statistics usually adopts priors containing no “information” of the data, such as uniform distributions or Gaussian distributions. Moreover, the impact of prior usually goes to vanishing when training progress. From this aspect, setting data-independent priors would not comprise the performance in an asymptotic manner. Recent works slightly relax the restrictions in convention and advance the PAC-Bayesian theory.

**Distribution-dependent priors.** Some works design priors relying on the data generation distribution but still not directly relying on the training data. This would be reasonable since we can assume the data distribution has been fixed before the data was collected [142]. Such distribution-dependent priors have shown to be able to considerably tighten the generalization bounds.

Following this line, Lever et al. [142] tightens the PAC-Bayesian bound (see Theorem 4) to the following bound. For any positive constant \( C \), we have that at least probability \( 1 - \delta > 0 \) with respect to the drawn sample, the generalization error is upper bounded as follows,

\[ \mathcal{R} (Q(h)) - C^* \mathcal{R} (Q(h)) \leq \frac{C^*}{C m} \left( \frac{2}{m} \log \frac{2 \xi (m)}{\delta} + \frac{\gamma^2}{2m} + \log \frac{2}{\delta} \right) \],

where \( Q(h) \) is the posterior of the learned model, its density is as below,

\[ q(h) \propto e^{-\gamma \mathcal{R} (Q(h))}, \quad \xi (m) = \mathcal{O} (\sqrt{m}), \quad C^* = \frac{C}{1 - e^{-C}}, \]

and \( \gamma \) is a positive constant.

Li et al. [145] further design a Bayes-stability framework to deliver generalization bounds based on
distribution-dependent priors. They tighten the result of Mou et al. [171] (eq. 8) to the follows,

\[
O \left( \frac{C}{m} \left[ \mathbb{E}_S \left[ \sum_{t=1}^{T} \frac{\eta_t^2}{\sigma_t^2} g_e(t) \right] \right] \right),
\]

where the gradient noise is assumed to be \( \frac{\sigma_e}{\sqrt{2}} I \), \( C > 0 \) is the upper bound of the loss function, \( T \) is the number of iterations, \( \eta_t \) is the learning rate in the \( t \)-th iteration, and

\[
g_e(t) = \mathbb{E}_{W_{t-1}} \left[ \frac{1}{m} \sum_{i=1}^{m} \| \nabla F(W_{t-1}, z_i) \|^2 \right].
\]

**Data-dependent priors.** Negrea et al. [175] further push the frontier that constructs priors not independent with data. Suppose \( S_j \) is a subset of \( S \) with size of \( n < m \). One may design a prior exploiting \( S_j \) to deliver a data-dependent forecast of the posterior \( Q \). We denote the subset as \( S_J = (z_{j_1}, \ldots, z_{j_n}) \), where all indices constitute a set \( J \). The index set \( J \) is randomly drawn from \( \{1, \ldots, m\} \). Suppose that we have the following generalization bound,

\[
\mathbb{E}_F \left[ \mathcal{R}(\theta) - \mathcal{R}_{S_J}(\theta) \right] \leq B,
\]

where \( \mathcal{R}_{S_J} \) is the empirical risk on the set \( S_J \), \( B > 0 \) is the bound, \( F \) is a \( \sigma \)-field such that \( \sigma(S_J) \subset F \perp \sigma(S_J^c) \). Also, we denote by \( U \) the uncertainty introduced by the machine learning algorithm. Negrea et al. [175] prove the following theorem, which considerably improve the one given in Xu and Raginsky [235] (eq. 5),

**Theorem 7.** Let \( J \subset [m], |J| = n \) be uniformly distributed and independent from the training sample set \( S \) and the model parameter \( \theta \). Suppose the loss is \( \sigma \)-sub-Gaussian under the data distribution. When the posterior \( Q = \mathbb{P}^S(\theta) \) and the prior \( P \) be a \( \sigma(S_J) \)-measurable data-dependent on the data space,

\[
\mathbb{E} \left[ \mathcal{R}(\theta) - \mathcal{R}_{S_J}(\theta) \right] \leq \sqrt{\frac{2\sigma^2}{m-n} I(\theta; S_J^c)}
\]

\[
\leq \sqrt{\frac{2\sigma^2}{m-n} \mathbb{E}[\text{KL}(Q||P)]}.
\]

Moreover, let \( U \) be independent with \( S \) and \( \theta \). When the posterior \( Q = \mathbb{P}^{S,U}(\theta) \) and the prior \( P \) be a \( \sigma(S_J, U) \)-measurable data-dependent on the data space,

\[
\mathbb{E} \left[ \mathcal{R}(\theta) - \mathcal{R}_{S_J}(\theta) \right] \leq \sqrt{\frac{2\sigma^2}{m-n} I^{S,J,U}(\theta; S_J^c)}
\]

\[
\leq \sqrt{\frac{2\sigma^2}{m-n} \mathbb{E}^{S,J,U}[\text{KL}(Q||P)]}.
\]

Following this path, further works include Haghifam et al. [99] cooperating with Steinke and Zakynthinou [213], Dziugaite et al. [80], Hellström and Durisi [112], and their applications [53].
3.5 The role of learning rate and batch size in shaping generalizability

Keskar et al. [124] conducted experiments which show large-batch SGM usually finds sharp local minima, while its small-batch counterpart leads to flat minima. The authors also explain that flatter local minima can make SGM generalize better. However, the experiments wherein only involve two values of batch size, which considerably limits the generality of the results. Then, Dinh et al. [67] argue that most current notions for sharpness or flatness are ill-defined. The flatness/sharpness is still under intensive study.

Goyal et al. [95] further propose a linear scaling rule for SGM: “when the minibatch size is multiplied by $k$, multiply the learning rate by $k$”. A similar statement is also given by Smith and Le [210]. Jastrzebski et al. [117] present empirical results on the relationships between the ratio $|S|/\eta$ and the dynamics, geometry, and the generalization, but still, only two values of both the batch size and learning rate are investigated.

Similar statements on the relationship between the generalizability and the hyperparameters are also presented in an empirical work by Chaudhari and Soatto [41]. Fig. 1 gives plots of the test accuracies of a multi-layer perceptron (MLP) trained on the dataset MNIST [139] in term of the ratio of the batch size to the learning rate respectively when 25.0% and 50.0% labels are randomly settled (see the original figure in [117]). These plots express the memorization ability.2

Figure 1: Scatter plots of test accuracy to ratio of batch size to learning rate. See [117].

Figure 2: Curves of test accuracy to batch size and learning rate. The four rows are respectively for (1) ResNet-110 trained on CIFAR-10, (2) ResNet-110 trained on CIFAR-100, (3) VGG-19 trained on CIFAR-10, and (4) VGG-19 trained on CIFAR-10. Each point represents a model. Each curve is based on 20 networks. See [106].

\textsuperscript{2}The concept of memorization is given by [?]. It is defined as the ability of a model to fit a dataset with random labels. If the training error can be very small even when the training labels are random, we argue the model is “memorizing” data, but not “learn” the data. Therefore, it is an important index expressing the generalization ability of a model.
He et al. [106] proves a PAC-Bayesian generalization bound under some assumptions. The generalization bound also suggests a positive correlation between the generalizability and the ratio of the learning rate and the batch size. He et al. [106] then conduct experiments that show a clear correlation between the generalizability and the ratio of batch size to learning rate; see fig. 3.

For more details and advances, please see recent works by Arpit et al. [13], Krueger et al. [133], Carlini et al. [40], and Smith and Le [211].

3.6 Interplay of optimization and Bayesian inference

An interesting interplay is observed in the research of optimization and Bayesian inference [234]: (1) Bayesian inference (or sampling) can be treated as optimization in the probability space (e.g., in stochastic gradient Markov chain Monte Carlo) and can be scaled through optimization (e.g., in variational inference); and (2) stochastic gradient Markov chain Monte Carlo also performs Bayesian approximation.

**Bayesian inference.** For parametric machine learning models, Bayesian learning aims to obtain the posterior of model parameters and then approach the best parameter. However, the analytic expression of the posterior is unknown in most real-world cases. To solve this problem, classical methods employ Markov chain Monte Carlo (MCMC) methods to infer the posterior, such as the Metropolis-Hastings algorithms [103, 87] and hybrid Monte Carlo (HMC; [72]). Bayesian inference has been used in many areas, including topic model [136, 246], Bayesian neural network [155, 200], and generative models [127, 93, 128].

**Scaling Bayesian inference via optimization.** Bayesian inference would be prohibitively time-consuming on large-scale data. To address this issue,

1. **Stochastic gradient Markov chain Monte Carlo** (SGMC; [157]) introduces stochastic gradient estimates [198] into MCMC. The family of SGMCMC algorithms includes stochastic gradient Langevin dynamics (SGLD; [229]), stochastic gradient Riemannian Langevin dynamics (SGRLD; [189]), stochastic gradient Fisher scoring (SGFS; [4]), stochastic gradient Hamiltonian Monte Carlo (SGHMC; [49]), stochastic gradient Nosé-Hoover Thermostat (SGNHT; [66]), etc. This paper analyses SGLD as an example of the SGMCMC scheme.

2. **Variational inference** [115, 27, 245] employs a two-step process to inference the posterior:
   
   1. We define a family of distributions,
      \[ \mathcal{Q} = \{q_\lambda | \lambda \in \Lambda \} \],
      \[ \text{where } \lambda \text{ is the parameter.} \]
   2. We search for the element from the distribution family \( \mathcal{Q} \) that is the closest to the posterior \( p(\theta|S) \), under KL divergence, i.e.,
      \[ \min_{\lambda} \text{KL}(q_\lambda(\theta)||p(\theta|S)). \]

Minimizing the KL divergence is equivalent to maximizing the evidence lower bound (ELBO) defined as below,

\[ \text{ELBO}(\lambda, S) = \mathbb{E}_{q_\lambda} \log p(\theta, S) - \mathbb{E}_{q_\lambda} \log q_\lambda(\theta). \]

**Understanding stochastic gradient-based optimization via Bayesian inference.** E [228] proposed to understand the optimization in deep learning via stochastic differential equations. Mandt et al. [159] propose to understand SGM as approximate Bayesian inference. Following this line, a large volume of literature has been delivered on SGM as shown in the previous subsections.
4 **Taking the high way: Study on loss surface**

A major barrier recognized by the whole community is that deep neural networks’ loss surfaces are extremely non-convex and even non-smooth. Such non-convexity and non-smoothness make the analysis of the optimization and generalization properties prohibitively difficult. A natural idea is to bypass the geometrical properties and then approach a theoretical explanation. Some papers argue that such “intimidating” geometrical properties are exactly the major factors that shape the properties of deep neural networks, and also the key to explaining deep learning.

4.1 **Does spurious local minimum exist?**

If all local minima are globally optimal, then we would not need to worry about optimization for deep learning. Then, is it true?

4.1.1 **Linear networks have no spurious local minima**

Linear neural networks refer to the neural networks whose activations are all linear functions. For linear neural networks, the loss surfaces do not have any spurious local minima: all local minima are equally good; they are all global minima.

Kawaguchi [121] proves that linear neural networks do not have any spurious local minimum under several assumptions: (1) the loss functions are squared losses; (2) $XX^T$ and $XY^T$ are both full-rank, where $X$ is the data matrix and $Y$ is the label matrix; (3) the dimension of the input layer is larger than the dimension of the output layer; (4) all the eigenvalues of matrix $YX^T(XX^T)^{-1}XY^T$ are different from each other. Lu and Kawaguchi [156] replace the assumptions to one: the data matrix $X$ and the label matrix $Y$ are both full-rank, which is, however, more restrictive.

Later, Zhou and Liang [253] prove that all critical points are global minima when some conditions hold. The authors prove this based on a new result on the analytic formulation of the critical points.
4.1.2 Nonlinear activations bring infinite spurious local minima

However, this good property no longer stands when the activations are nonlinear. Zhou and Liang [253] show that one-hidden layer ReLU neural networks have spurious local minima. However, the proof relies on that the hidden layer has only two nodes. Swirszcz et al. [217] relax the assumption to that most of the neurons are not activated. Safran and Shamir [202] show that two-layer ReLU networks have spurious local minima in a computer-assisted. Yun et al. [240] prove that one-hidden-layer ReLU neural networks have infinitely many spurious local minima when the outputs are one-dimensional. Goldblum et al. [91] prove that the training performance of all the local minima equals to a linear model for neural networks of any depth, and verify it by experiments. He et al. [108] further extend the condition to arbitrary depth, arbitrary activations, and arbitrary output dimension.

4.2 Geometric structure of loss surface.

Then, how would the loss surface look like? Knowledge of the geometrical structures can considerably advance the current understanding of deep learning.

**Linear partition of the loss surface.** Soudry and Hoffer [212] first highlight a smooth and multilinear partition exists on the loss surfaces of neural networks. The nonlinearities in the piecewise linear activations divide the loss surface into multiple smooth and multilinear open regions. Specifically, every nonlinear point in the activation functions creates a group of the non-differentiable boundaries between the cells, while the linear parts of activations correspond to the smooth and multilinear interiors.

He et al. [108] prove several sophisticated properties of the partition: (1) if there are local minima within an open cell, they are all equally good in the sense of empirical risk, and therefore, all local minima are global minima within this cell; (2) the local minima in any open cell form an equivalence class. Moreover, these local minima are concentrated in a valley; and (3) when all activations are linear, the partition collapses to one single cell, covering linear neural networks as a special case.

The property (2) by He et al. [108] explains an important property of mode connectivity: the solutions found by stochastic gradient descent or its variants are connected with each other by a path in the weight space, on which all points have almost the same empirical risk. This property is found in two empirical works by Garipov et al. [85] and Draxler et al. [68]. A recent theoretical work by Kuditipudi et al. [134] proves that the mode connectivity can be guaranteed by dropout stability and noise stability.

**Correspondence between the landscapes of empirical risk and expected risk.** Two works by Zhou and Feng [251] and Mei et al. and [163] prove a linkage between the landscapes of the empirical risk surface and expected risk. This correspondence highlights the possibility of employing the results on the geometry of loss surfaces (empirical risk surfaces) to understand of generalizability (the gap between expected risks and empirical risks). Specifically, the authors prove that all of (1) the gradient of empirical risk, (2) the stationary points of the empirical risk, and (3) the empirical risk itself can be asymptotically approximated by their population counterparts. Moreover, they prove a generalization bound for non-convex case: at least by probability $1 - \delta$,

$$O\left(\tau \sqrt{\frac{9}{8}[1 + c_r(D - 1)]} \sqrt{\frac{s \log(mU/D) + \log(4/\delta)}{m}}\right),$$

where all datums $x$ are assumed to be $\tau$-sub-Gaussian, $c_r = \max\left(\frac{r^2}{16}, \left(\frac{r^2}{16}\right)^{l-1}\right)$, and $s$ is the number of non-zero components in the weights. Later, [163] prove a similar correspondence between the Hessian...
matrices of the expected risk and the empirical risk under an assumption that the sample size is larger than the parameter size.

**Eigenvalues of the Hessian.** Sagun et al. in 2016 [203] and 2018 [204] and Papan in 2018 [188] conduct experiments to study the the eigenvalues of the Hessian of the loss surface. Sagun et al. [203, 204] discover that: (1) a large bulk of eigenvalues are centered close to zero; and (2) several outliers are located away from the bulk. Papan [188] gives the full spectrums of the Hessian matrix. Fig. 4 compares among the spectrum of Hessian matrices when training and testing a VGG-11 on the datasets MNIST and CIFAR-10 (originally from a recent empirical work by [188, p. 2, figs. 1(a) and 1(c) and p. 3, figs. 2(a) and 2(c)]).

5 Reflecting the role of over-parameterization: Is it only harmful?

One might blame the over-parameterization of deep learning for its lack of theoretical foundations. However, recent works also discover that such over-parameterization would have also contributed to the success of deep learning.

**Neural tangent kernel.** Neal in 1995 [173] and 1996 [174], Williams [231], and Hazan and Jaakkola[105] prove that an infinite-width shallow neural network is equivalent to a Gaussian process. Damianou and Lawrence [61], Duvenaud et al. [74], Hensman and Lawrence [113], Lawrence and Moore [137], Bui et al. [34], and Lee et al. [140], gradually extend the equivalence to any infinite-width neural network of an arbitrary depth. Lee et al. [141] further prove that infinite-width neural networks of any depth behaves as linear models during training with gradient descent. Jacot et al. [116] prove that the learned model for the least-squares regression problem during training is characterized by a linear differential equation the setting under the infinite-width regime.

**Influence of over-parameterization on the loss surface.** The over-parameterization would significantly
reshape the loss surfaces of neural networks. Choromanska et al. [55] conducted an experiment that shows: (1) a large proportion of the local minima on the loss surface of an over-parameterized network are “equivalent” with each other (they have the same empirical risk); and (2) neural networks with small or normal sizes have spurious local minima but the proportion of the spurious local minima decreases rapidly when the network size increase. Li et al. [144] prove that over-parameterized fully-connected deep neural networks do not have any spurious local minimum under assumptions: (1) the activation functions are continuous; and (2) the loss functions are convex and differentiable. Nguyen et al. [180] extend the “no bad local minima” property to networks trained under the cross-entropy loss. Nguyen [179] further discover that the global minima are connected with each other and concentrated on a unique valley if the neural networks are sufficiently over-parameterized.

**Influence of over-parameterization on the optimization.** It has been shown that the over-parameterization contribute to securing the optimization performance of gradient-based optimizers. Du et al. [71] prove that gradient descent converges to global local minima under the quadratic loss at a linear convergence rate for two-layer neural networks as long as the network widths are sufficiently large. The authors also suggest that the over-parameterization can restrict the weights in all iterations close to the random initialization. Chizat and Bach [54] model the optimization of one-hidden-layer networks as minimizing a convex function of a measure discretized into a mixture of particles by continuous-time gradient descent. They prove that the gradient flow characterizing the optimization converges to the global minimizers.

Going beyond one hidden layer, Allen-Zhu et al. [8], Du et al. [69], and Zou et al. [255] concurrently prove that SGD converge to global optimal solution of over-parameterized deep neural networks in polynomial time with under slightly different assumptions on the network architectures and training data. Chen et al. [52] prove that the convergence of optimization is guaranteed if the network width is polylogarithmic with the sample size $n$ and $1/\epsilon$ where $\epsilon$ is the targeted error.

**Influence of over-parameterization on the generalization and learnability.** Andoni et al. [10] study two-layer neural networks under assumptions that the node in the first layer are all linear functions while the hidden nodes in the last layer are non-linear. The authors prove that sufficiently wide neural network can learn any low-degree polynomial functions when trained by generic gradient descent algorithm and all weights are randomly initialized. Brutzkus et al. [32] also give generalization guarantees for two-layer neural networks under some assumptions. Li and Yuan [147] prove that over-parameterized one-hidden-layer ReLU networks have guaranteed small generalization error if they are trained by SGD for multi-class classification problems from random initialization. Also, the data is assumed to be a mixture of well-separated distributions.

Arora et al. [12] and Allen-Zhu et al. [7] prove that the sample complexity for neural networks with two or three layers is almost independent with the parameter size. Chen et al. [52] also show that the generalization is guaranteed if the network width is polylogarithmic with the sample size $n$ and $\epsilon^{-1}$ where $\epsilon$ is the targeted error. Cao et al. [39] also prove generalization bounds for wide and deep neural networks. Wei et al. [226] prove that regularizers can significantly reshape the generalization and optimization.

**Influence of the network depth on the generalizability.** Canziani et al. [38] suggest that deeper neural networks could have better generalizabilities, summarizing the applications of neural networks in practice. This plays as a major part in the irreconciliation between the over-parameterization of neural networks and their excellent generalizabilities. Besides the previous results, another possible explanation is from information theory. Zhang et al. [249] prove a generalization bound that characterizes how the generalizability evolves when the network becomes deeper, based on some techniques developed by Xu and Raginsky [235].
6 Theoretical foundations for specific architectures

In the previous sections, we discuss the generalizabilities of neural networks in general. This section discusses the influence of some specific network structures.

Convolutional neural networks. Convolutional neural networks (CNNs) introduce convolutional layers into deep learning. It has been widely employed in computer vision [132], natural language process [239], and deep reinforcement learning [209].

Du et al. [70] study several statistical properties of CNNs. The authors prove that for an \( c \)-dimensional convolutional filter with linear activation needs \( \tilde{O}(c/\epsilon^2) \) training examples for achieving a population prediction error \( \epsilon \) on \( d_X \)-dimensional input, while a fully-connected counterpart needs \( \Omega(d/\epsilon^2) \) training examples. Moreover, the authors prove that a one-hidden-layer CNN needs \( \tilde{O}((c+r)/\epsilon^2) \) training examples when the output dimension is \( r \) and the ratio of stride size to filter size is fixed. Zhou and Feng [252] prove (1) an \( O(\log \prod_{i=0}^{D} b_i) \) generalization bound for CNN, where \( b_i \) is the magnitude of the parameter in the \( i \)-th layer; and (2) one-to-one correspondence between convergence guarantees between the stationary points on the empirical risk landscape and their population counterparts. Lin and Zhang [151] studies how the sparsity and permutation of convolutional layers would influence the spectral norm and then the generalizability.

Several works have also studied the generalizability of CNNs with residual connections [110], including He et al. [107] and Chen et al. [44].

Recurrent neural networks. Recurrent Neural Networks (RNNs) possess a recurrent structure and have demonstrated promising performance in processing sequential data analysis, and have been deployed in many real-world areas, including natural language processing [15, 216] and speech recognition [96, 97]. Chen et al. [48] develops a generalization bound for RNNs based on works by Neyshabur et al. [176] and Bartlett et al. [19]. Allen-Zhu and Li [6] also analyze the generalizability of RNNs.

Tu et al. [220] develop a gradient measure based on Fisher-Rao norm, which delivers a generalization bound as follows.

**Theorem 8.** Fix margin parameter \( \alpha \), then for any \( \delta > 0 \), with probability at least \( 1 - \delta \), the following holds for every RNN whose weight matrices \( \theta = (U, W, V) \) satisfy \( ||V^T||_1 \leq \beta V, ||W^T||_1 \leq \beta W, ||U^T||_1 \leq \beta U \) and \( ||\theta||_{fs} \leq r \):

\[
\mathbb{E}[\mathbf{1}_{\mathcal{M}_{UL}(x,y) \leq 0}] \leq \frac{4k}{\alpha} \left( \frac{r||X||_F}{2n} \sqrt{\frac{1}{\lambda_{\min}(\mathbb{E}(xx^T))}} + \frac{1}{n} \beta_V \beta_U ||X^T||_1 \Lambda \right) + \frac{1}{n} \sum_{i} \mathbf{1}_{\mathcal{M}_{UL}(x_i,y_i) \leq a} + 3 \sqrt{\frac{\log \frac{2}{\delta}}{2n}}. \tag{9}
\]

The gradient measure considerably tightens the bound and characterizes the relationship between generalizability and trainability. It suggests adding noise to the input data to boost the generalizability. It also justifies the technique gradient clipping [165, 86, 191].

 Networks for permutation invariant/equivariant functions. Zaheer et al. [242], Cohen and Welling [56], and Cohen et al. [57] adopt machine learning to tasks based on “sets”, which are usually permutation invariant or equivariant. These models are then applied to a variety of scenarios, including 3D point cloud processing.
Sannai et al. [206] prove a universal approximation theorem for this family of networks. Moreover, Sannai et al. [206] show that the free parameters in this network family are exponentially fewer than usual models, which further suggests a significantly better generalizability. Sannai and Imaizumi [205] prove that the deep model in Zaheer et al. [242] can improve generalization bound by $\sqrt{n!}$, where $n$ is the number of permuting coordinates of data. Brutzkus and Globerson [31] also demonstrate that larger model size can boost the generalizability from both theoretical and empirical aspects.

7 Looking beyond: Rising concerns of ethics and security

Rising concerns are seen on the ethical and security issues of deep learning, including privacy preservation, fairness protection, and adversarial robustness. Understanding the theoretical foundations of such issues are also of high importance. This section summarizes the recent advances in these areas.

7.1 Privacy preservation

Deep learning has been deployed to process massive personal data, including financial and medical records. It is of profound importance to discover the high-value population knowledge while protecting the highly sensitive privacy [77, 78, ?].

**Differential privacy.** A major measure in measuring the privacy-preserving ability of an algorithm is differential privacy [78]. We call a learning algorithm is $(\varepsilon, \delta)$-differential private if when the training data is exposed to some small disturbance, the change in the output hypothesis is restricted as follows,

$$\log \left[ \frac{P_{A(S)}(A(S) \in B) - \delta}{P_{A(S')} (A(S') \in B)} \right] \leq \varepsilon,$$

where $B$ is an arbitrary subset of the hypothesis space and $(S, S')$ is a neighboring sample set pair in which $S$ and $S'$ only differ by one example. The left-hand side of eq. (10) is also called privacy loss.

In eq. (10), we use division operation to measure the change of the output hypothesis. Through modifying the division operation or inserting some assumptions, many variants of notions for measuring privacy-preserving ability are proposed: (1) concentration differential privacy assumes that the privacy loss is sub-Gaussian [79, 35]; (2) mutual-information differential privacy and KL differential privacy replace the division operation in privacy loss by mutual information and KL divergence, respectively [58, 225, 148, 42]; (3) Rényi differential privacy further replaces the KL divergence by Rényi divergence [167, 88]; etc.

These techniques have been also employed for privacy preservation in deep learning [1].

**Generalization-privacy relationship.** Some works have suggested that the privacy-preserving ability measured by differential privacy is almost equivalent to the generalizability.

Dwork et al. [75] prove a high-probability generalization bound for $(\varepsilon, \delta)$-differentially private machine learning algorithms as follows,

$$\mathbb{P} \left[ \mathcal{R}(A(S)) - \hat{\mathcal{R}}_S(A(S)) < 4\varepsilon \right] > 1 - 8\delta^2.$$
Oneto et al. [183] improves the generalization bound as below

\[ P \left[ \text{Diff } \mathcal{R} < \sqrt{6 \hat{\mathcal{R}}_S(A(S))} \hat{\epsilon} + 6 \left( \frac{\hat{\epsilon}^2 + 1}{N} \right) \right] > 1 - 3e^{-N\hat{\epsilon}^2}, \]

and

\[ P \left[ \text{Diff } \mathcal{R} < \sqrt{4 \hat{\mathcal{V}}_S(A(S))} \hat{\epsilon} + \frac{5N}{N-1} \left( \frac{\hat{\epsilon}^2 + 1}{N} \right) \right] > 1 - 3e^{-N\hat{\epsilon}^2}, \]

where

\[ \text{Diff } \mathcal{R} = \mathcal{R}(A(S)) - \hat{\mathcal{R}}_S(A(S)), \]

\[ \hat{\epsilon} = \epsilon + \sqrt{\frac{1}{N}}, \]

and \( \hat{\mathcal{V}}_S(A(S)) \) is the empirical variance of \( l(A(S), \cdot) \):

\[ \hat{\mathcal{V}}_S(A(S)) = \frac{1}{2N(N-1)} \sum_{i \neq j} \left[ l(A(S), x_i) - l(A(S), x_j) \right]^2. \]

Nissim and Stemmer [181] further prove that

\[ P \left[ \mathcal{R}(A(S)) - \hat{\mathcal{R}}_S(A(S)) < 13\hat{\epsilon} \right] > 1 - \frac{2\delta}{\epsilon} \log \left( \frac{2}{\epsilon} \right). \]

The tightest generalization bound by far is given by He et al. [109] as follows,

\[ P \left[ \left| \hat{\mathcal{R}}_S(A(S)) - \mathcal{R}(A(S)) \right| < 9\epsilon \right] > 1 - \frac{e^{-\epsilon}\delta}{\epsilon} \ln \left( \frac{2}{\epsilon} \right). \] (11)

This generalization bound is delivered in three stages. The authors first prove an on-average generalization bound for any \((\epsilon, \delta)\)-differentially private multi-database learning algorithm

\[ \tilde{A} : \tilde{S} \rightarrow \mathcal{H} \times \{1, \ldots, k\}, \]

as follows,

\[ \left| \frac{\mathbb{E}_{\tilde{S} \sim D^N} \left[ \mathbb{E}_{A(\tilde{S})} \left[ \hat{\mathcal{R}}_{S_i \tilde{A}(\tilde{S})} \left( h_{\tilde{A}(\tilde{S})} \right) \right] - \mathbb{E}_{A(\tilde{S})} \left[ \mathcal{R} \left( h_{\tilde{A}(\tilde{S})} \right) \right] \right] }{\mathbb{E}_{A(\tilde{S})} \left[ \mathcal{R} \left( h_{\tilde{A}(\tilde{S})} \right) \right]} \right| \leq e^{-\epsilon}k\delta + 1 - e^{-\epsilon}, \] (12)

where the loss function \( \|l\|_\infty \leq 1. \)

They then obtain a high-probability generalization bound for multi-database algorithms as follows,

\[ P \left[ \hat{\mathcal{R}}_{S_{\tilde{A}(\tilde{S})}} \left( h_{\tilde{A}(\tilde{S})} \right) \leq \mathcal{R} \left( h_{\tilde{A}(\tilde{S})} \right) + ke^{-\epsilon}\delta + 3\epsilon \right] \geq \epsilon. \] (13)

They eventually prove eq. (11) by reduction to absurdity.
7.2 Algorithmic fairness

Deep learning been deployed in many critical decision-making areas, such as mortgage approval [195], credit card assessment [125, 45], college admission [2, 46], employee selection [83], and recidivism prediction [30].

However, long-standing systematical discrimination has been observed in these areas against people based on their backgrounds, including races, genders, nationalities, ages, and religious. Additionally, people with minority backgrounds are institutionally under-represented in historical data. Meanwhile, it has been revealed in an increasing number of reports that the discrimination exists in the data are inherited and even intensified in the modeled learned by machine learning-based algorithms. The fairness concerns are particularly severe due to the black-box nature. For example, facial recognition is reportedly to be “accurate, if you are a white guy” [153]; a police system for recidivism prediction, Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) [30], tends to label minority people with significantly higher possibility to be reoffenders. Mitigating the fairness risk is timely and important.

Many approaches have been presented to address the fairness issues. They can be roughly categorized into three parts: (1) pre-processing methods, which aims to “rectify” data” before processed by learning algorithms [36, 84, 119, 76, 37, 100, 243, 154]; (2) post-processing methods redress the outputs of learning systems to remove the potential discrimination [233]; and (3) in-processing methods adapt constraints or regularizers in the training procedures to enforce fairness [241, 3, 120, 164, 123].

7.3 Adversarial robustness

Szegedy et al. [218] discover that neural networks are vulnerable to adversarial examples which are slightly different from true datums but are likely to be misclassified by neural networks. Since then, many approaches have been proposed on attacking neural networks by adversarial examples or defending the adversarial attacks [26, 185, 186, 187, 94, 178, 186, 89].

Generalizability in the presence of adversarial examples. Schmidt et al. [208] first study the generalizability in the presence of adversarial examples. They define a new notion, adversarial robustness generalization, for measuring the generalizability when adversarial examples exist in data. In contrast, the generalizability when there is no adversarial examples is evaluated by the standard generalization. They prove that to secure the same generalization, adversarial robustness generalization needs a higher sample size, when the data is drawn from some specific distributions and the adversarial examples are under the $l_\infty$ norm.

Cullina et al. [59] extends the PAC-learning framework to learning with adversarial examples. The authors propose a corrupted hypothesis classes which is defined to be the hypothesis classes for the classification problem in the presence of adversaries. The VC dimension of a corrupted hypothesis class is then defined to be adversarial VC-dimension, while the standard VC-dimension is defined on clean datasets. They show that the adversarial VC-dimension can be either larger or smaller than the standard VC-dimension. Montasser et al. [170] further define an agnostic robust PAC learnability and realizable robust PAC learnability. When a learning algorithm whose rule meets the requirements of both agnostic robust PAC learnability and realizable robust PAC learnability, the learning is defined to be improper. They then show that under an improper learning rule, any hypothesis space $\mathcal{H}$ is robustly PAC learnable, if its VC dimension is finite.

Generalizability of adversarial training. Amongst the defense approaches, adversarial training [60, 143, 16, 250] can considerably improve the adversarial robustness of deep neural networks against adversarial examples. Specifically, adversarial training can be formulated as solving the following minimax-loss problem.
\[
\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \max_{\|x_i'-x_i\| \leq \rho} l(h_{\theta}(x'_i), y_i),
\]

where \( h_{\theta} \) is the hypothesis parameterized by \( \theta \), \( N \) is the training sample size, \( x_i \) is a feature, \( y_i \) is the corresponding label, and \( l \) is the loss function. Intuitively, adversarial training optimizes neural networks according to the performance on worst-case examples, which are most likely to be adversarial examples.

Based on the minimax loss framework, several generalization bounds have been proposed \([237, 221, 126]\). All these results suggest that adversarial training would comprise generalizability when improving the adversarial robustness.

Yin et al. \([237]\) studies how adversarial training would influence the generalizability based on the techniques developed by Bartlett et al. \([19]\) and Raghunathan et al. \([193]\). They prove a tight upper bound on the Rademacher complexity which leads to a generalization bound. However, the generalization bound is only for neural networks with one hidden layer and ReLU activation functions.

To overcome this shortcoming, Khim et al. \([126]\) next prove a generalization bound via function transformation. They prove an upper bound for the Rademacher complexity via techniques developed by Golowich et al. \([92]\).

Tu et al. \([221]\) develop another path for evaluating the generalizability: (1) they first show that one can remove the maximization operation in the objective function in the minimax framework, \( \max_{\|x'_i-x_i\| \leq \rho} l(h_{\theta}(x'_i), y_i) \) via a re-parameterization mapping, while the parameter distribution would not shift much under the Wasserstein distribution; (2) they prove a local worst-case risk bound for the re-parameterized optimization problem which has been a minimization problem; and (3) they prove a generalization bound based on the local worst-case risk bound. This generalization bound is more generalization in two aspects: (1) the adversarial examples encompass all bounded adversaries under the \( l_q \) norm where \( l_q \) is arbitrary; (2) the approach can be applied to multi-class classification and almost all practical loss functions, including hinge loss and ramp loss.

On the other side, Bhagoji et al. \([25]\) also prove a lower bound for the generalization error of classifiers in the presence of adversarial examples. Min et al. \([166]\) discover that the performance of adversarial training does not always have a clear correlation with the training sample size. The authors categorize all learning problems in three parts: (1) in the weak adversary regime, an increasing training sample size corresponds to an improving generalizability; (2) in the medium adversary regime, one may observe a double descent curve of the generalization error, when the training sample size increases; and (3) in the strong adversary regime, more training data causes the generalization error to increase. Here, the three regimes can be determined by the level of adversaries. Similar phenomena to the medium adversary regime have also been discovered by Chen et al. \([47]\).

Other theoretical analyses include an improved generalization bound via Rademacher complexity by Attias et al. \([14]\), the trade-off between accuracy and robustness \([248]\), a generalization bound via optimal transport and optimal couplings by Pydi and Jog \([192]\).

### 8 Summary and discussion

In this paper, we review the literature on the deep learning theory from six aspects: (1) complexity and capacity-based method for analyzing the generalizability of deep learning; (2) stochastic differential equations and their dynamic systems for modelling stochastic gradient methods (SGM), which characterize the implicit
regularization introduced by the limited exploration space; (3) the geometry of the loss landscape governing the trajectories of the dynamic systems; (4) the roles of over-parameterization of deep neural networks from both positive and negative perspectives; (5) parallel to the previous parts, the theoretical foundations for some special structures in the neural network architecture; and (6) the rising concerns on ethics and security and their relationships with deep learning theory.

Many works give upper bounds for the generalization error of deep learning. The results are summarized in the following table.

Table 1: Comparisons of major generalization bounds for deep learning.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Generalization bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goldberg and Jerrum [90]</td>
<td>$O\left(\sqrt{\frac{\log(mW^2)}{mW^2}}\right)$</td>
</tr>
<tr>
<td>Bartlett et al. [20]</td>
<td>$O\left(\sqrt{\frac{\log(mW^2L^2)}{mWL^2}}\right)$</td>
</tr>
<tr>
<td>Harvey et al. [102]</td>
<td>$O\left(\sqrt{\frac{\log(mW^2L^2)}{mWL^2}}\right)$</td>
</tr>
<tr>
<td>Neyshabur et al. [177]</td>
<td>$O\left(\frac{\beta^2M}{\gamma^2m}\prod_{i=1}^{D}M_F(j)\right)$</td>
</tr>
<tr>
<td>Neyshabur et al. [176]</td>
<td>$O\left(\sqrt{\frac{\beta^2D^2W\log(DW)}{\gamma^2m}\prod_{i=1}^{D}1+\log \frac{2m}{\delta}}\right)$</td>
</tr>
<tr>
<td>Golowich et al. [92]</td>
<td>$O\left(\frac{B\sqrt{T}}{\sqrt{m}}\prod_{i=1}^{D}M_F(j)\right)$ or $O\left(\frac{B\sqrt{D}T\log m}{\sqrt{m}}\prod_{i=1}^{D}M_F(j)\right)$</td>
</tr>
<tr>
<td>Dieuleveut et al. [65]</td>
<td>$O\left(\frac{\beta}{m^{\frac{2}{3}}}\right)$</td>
</tr>
<tr>
<td>Hardt et al. [101]</td>
<td>$O\left(\frac{1+1/8C}{m}T\right)$</td>
</tr>
<tr>
<td>Raginsky et al. [194]</td>
<td>$\tilde{O}\left(\frac{\beta(\beta+1)^2\delta^{1/4}}{X_\theta}\log(\frac{1}{\epsilon}) + \frac{\beta(\beta+1)^2\delta^{1/4}}{X_\theta} + \frac{U\log(\beta+1)}{\beta}\right)$</td>
</tr>
<tr>
<td>Pensia et al. [190]</td>
<td>$\tilde{O}\left(\sqrt{\frac{\beta^2}{m}\sum_{t=1}^{T}\frac{L^2}{\sigma^2_t}}\right)$</td>
</tr>
<tr>
<td>Mou et al. [171]</td>
<td>$\tilde{O}\left(\frac{2LC}{m}\left(\sum_{t=1}^{T}1\right)^2\right)$</td>
</tr>
<tr>
<td>Mou et al. [171]</td>
<td>$O\left(\sqrt{\frac{2}{m}\sum_{k=1}^{N}T_{k}e^{-\frac{1}{2}(T_k-T_k)}E\left[</td>
</tr>
<tr>
<td>Lever et al. [142]</td>
<td>$\tilde{O}\left(\frac{C^*}{cm}\left(\lambda\sqrt{\frac{2}{m}\log\frac{2(m)}{\delta}} + \frac{\gamma^2}{2m} + \log \frac{2}{\delta}\right)\right)$</td>
</tr>
<tr>
<td>Li. et al. [145]</td>
<td>$\tilde{O}\left(\frac{C}{m}\sqrt{E_{S_\theta}^{T}T_{\theta}}\left(\sum_{t=1}^{T}w_{t}^{2}p_{t}(t)\right)\right)$</td>
</tr>
<tr>
<td>Negrea et al. [175]</td>
<td>$\sqrt{\frac{2\epsilon^2}{m-n}I_{S_\theta}(\theta;S_{\theta})} \leq \sqrt{\frac{2\epsilon^2}{m-n}E_{S_\theta}(\theta;S_{\theta})[KL(Q</td>
</tr>
</tbody>
</table>

From our view, these six directions are promising in establishing the theoretical foundations of deep learning. However, the current "states of the art" are still far away from sufficient; all of them are facing major
difficulties. In this paper, we list three main directions for future works:

• The current measures for the complexity evaluate the whole hypothesis space, but can hardly capture the complexity of the “effective” hypothesis space that the training algorithm would output. Thus, a “local” version of complexity would be helpful; see an example [18].

• The current models for SGM usually ignore some of the popular training techniques, such as momentum, adaptive learning rate, gradient clipping, batch normalization, etc. Some works have been studying on thesis problems, but a full “landscape” is still missing. Also, the works are usually assuming the gradient noise is drawn from some specific distributions, such as Gaussian distributions or $\alpha$-stable distributions.

• The geometry of the loss surface remains premature. The extreme complexity of neural networks makes simple explanations unaccessible. Efforts on understanding some sophisticated geometrical properties are demanded, such as the sharpness/flatness of the local minima, the potential categorization of the local minima (valley) according to their performance, and the volumes of the local minima from different categories.

References


