Why ResNet Works? Residuals Generalize

Fengxiang He, *Student Member, IEEE*, Tongliang Liu, *Member, IEEE*, and Dacheng Tao, *Fellow, IEEE*

***Abstract*— Residual connections significantly boost the performance of deep neural networks. However, few theoretical results address the influence of residuals on the hypothesis complexity and the generalization ability of deep neural networks. This article studies the influence of residual connections on the hypothesis complexity of the neural network in terms of the covering number of its hypothesis space. We first present an upper bound of the covering number of networks with residual connections. This bound shares a similar structure with that of neural networks without residual connections. This result suggests that moving a weight matrix or nonlinear activation from the bone to a vine would not increase the hypothesis space. Afterward, an $\mathcal{O}(1/\sqrt{N})$ margin-based multiclass generalization bound is obtained for ResNet, as an exemplary case of any deep neural network with residual connections. Generalization guarantees for similar state-of-the-art neural network architectures, such as DenseNet and ResNeXt, are straightforward. According to the obtained generalization bound, we should introduce regularization terms to control the magnitude of the norms of weight matrices not to increase too much, in practice, to ensure a good generalization ability, which justifies the technique of weight decay.**

***Index Terms*— Deep learning, learning theory.**

I. Introduction

THE recent years witnessed dramatic progress of deep neural networks [1]–[6]. Since ResNet [7], residual connections have been widely used in many state-of-the-art neural network architectures [7]–[9] and lead a series of breakthroughs in computer vision [10]–[14], data mining [15], and so on. Numerous empirical results are showing that residual connections can significantly ease the difficulty of training deep neural networks to fit the training sample while maintaining excellent generalization ability on test examples. However, little theoretical analysis has been presented on the effect of residual connections on the generalization ability of deep neural networks.

Residuals connect layers that are not neighbored in chain-like neural networks. These new constructions break the convention that stacking layers one by one to build a chain-like neural network. They introduce loops into neural networks that are previously chain-like. Thus, intuitively, residual connections could significantly increase the complexity of the hypothesis space of the deep neural network and, therefore, lead to significantly worse generalization ability according to the principle of Occam's razor, which demonstrates a negative correlation between the generalization ability of an algorithm and its hypothesis complexity. Leaving this problem elusive could set restrictions on applying the recent progress of neural networks with residual connections to safety-critical domains, from autonomous vehicles [16] to medical diagnose [17], in which algorithmic mistakes could lead to fatal disasters.

In this article, we explore the influence on the hypothesis complexity induced by residual connections in terms of the covering number of the hypothesis space. An upper bound for the covering number is proposed. Our bound demonstrate that, when the total number of weight matrices involved in a neural network is fixed, the upper bound on the covering number remains the same, no matter whether the weight matrices are in the residual connections or the "stem"[1]. This result indicates that residual connections may not increase the complexity of the hypothesis space compared with a chain-like neural network if the total numbers of the weight matrices and the nonlinearities are fixed. Based on the upper bound on the covering number, we further prove an $\mathcal{O}(1/\sqrt{N})$ generalization bound for ResNet as an exemplary case for all neural networks with residual connections, where N is the training sample size. Based on our framework, generalization bounds for similar architectures constructed by adding residual connections to chain-like neural networks can be straightly obtained.

Our generalization bound closely depends on the product of the norms of all weight matrices. Specifically, there is a negative correlation between the generalization ability of a neural network with the product of the norms of all weight matrices. This feature leads to a practical implementation: to approach a good generalization ability, we need to use regularization terms to control the magnitude of the norms of weight matrices. This implementation justifies the techniques of weight decay and spectral normalization in training deep neural networks, which uses the L_2-norm of the weights as a regularization term [18]. This technique is also suggested by [19].

The rest of this article is structured as follows. Section II reviews the existing literature regarding the generalization ability of deep neural networks in both theoretical and empirical aspects. Section III provides necessary preliminaries. Section IV summarizes the notation for deep neural networks with residual connections as the stem-vine framework.

Manuscript received February 14, 2019; revised September 13, 2019 and December 20, 2019; accepted January 1, 2020. Date of publication February 5, 2020; date of current version December 1, 2020. This work was supported by the Australian Research Council under Project FL-170100117 and Project DP-180103424. *(Corresponding author: Dacheng Tao.)*

The authors are with the School of Computer Science, Faculty of Engineering, University of Sydney, Darlington, NSW 2008, Australia (e-mail: fengxiang.he@sydney.edu.au; tongliang.liu@sydney.edu.au; dacheng.tao@sydney.edu.au).

Digital Object Identifier 10.1109/TNNLS.2020.2966319

[1] The "stem" is defined to denote the chain-like part of the neural network besides all the residuals. For more details, please refer to Section IV.

Section V presents our main results: a covering bound for deep neural networks with residual connections, a covering bound for ResNet, a generalization bound for ResNet, and a practical implementation from the theoretical results. Section VI collects all the proofs. Section VII concludes this article.

II. Related Works

Understanding the generalization ability has vital importance to the development of deep neural networks. There already exist some results approaching this goal.

Zhang *et al.* [20] conduct 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 article attracts the community of learning theory to the important topic: how to theoretically interpret the success of deep neural networks.

Kawaguchi *et al.* [21] discuss many open problems regarding the excellent generalization ability of deep neural networks despite the large capacity, complexity, possible algorithmic instability, nonrobustness, and sharp minima. They also provide some insights to solve the problems.

Harvey *et al.* [22] prove upper and lower bounds on the VC-dimension of the hypothesis space of deep neural networks with the activation function of ReLU. Specifically, this article presents an $\mathcal{O}(WL\log(W))$ upper bound for the VC-dimension and an example of such networks with the VC-dimension $\Omega(WL\log(W/L))$, where W and L are, respectively, denoted to the width and depth of the neural network. This article also gives a tight bound $\Theta(WU)$ for the VC-dimension of any deep neural network, where U is the number of the hidden units in the neural network. The upper bounds of the VC-dimensions lead to an $\mathcal{O}(h/N)$ generalization bound, where h is the VC-dimension and N is the training sample size [23].

Golowich *et al.* study the sample complexity of deep neural networks and present upper bounds on the Rademacher complexity of the neural networks in terms of the norm of the weight matrix in each layer [24]. Compared with previous works, these complexity bounds have improved dependence on the network depth and, under some additional assumptions, are fully independent of the network size (both depth and width). The upper bounds on the Rademacher complexity further lead to $\mathcal{O}(1/\sqrt{N})$ upper bounds on the generalization error of neural networks.

Neyshabur *et al.* explore several methods that could explain the generalization ability of deep neural networks, including norm-based control, sharpness, and robustness [25]. They study the potentials of these methods and highlight the importance of scale normalization. In addition, they propose a definition of the sharpness and present a connection between the sharpness and the PAC-Bayes theory. They also demonstrate how well their theories can explain the observed experimental results.

Lang *et al.* explore the capacity measures for deep neural networks from a geometrical invariance viewpoint [26]. They propose to use the Fisher–Rao norm to measure the capacity of deep neural networks. Motivated by information geometry, they reveal the invariance property of the Fisher–Rao norm. The authors further establish some norm-comparison inequalities that demonstrate that the Fisher–Rao norm is an umbrella for many existing norm-based complexity measures. They also present experimental results to support their theoretical findings.

Novak *et al.* [27] conduct comparative experiments to study the generalization ability of deep neural networks. The empirical results demonstrate that the input–output Jacobian norm and linear region counting play vital roles in the generalization ability of networks. In addition, the generalization bound is also highly dependent on how close the output hypothesis is to the data manifold.

Two recent works, respectively, by Bartlett *et al.* [19] and Neyshabur *et al.* [28] provide upper bounds for the generalization error of chain-like deep neural networks. Specifically, [19] proposes an $\mathcal{O}(1/\sqrt{N})$ spectral-normalized margin-based generalization bound by upper-bounding the Rademacher complexity/covering number of the hypothesis space through the divide-and-conquer strategy. Meanwhile, [28] obtains a similar result under the PAC-bayesian framework. This work is partially motivated by the analysis in [19].

As regards the difference of methods, [21]–[26] adapt the VC-dimension, the Rademacher complexity, the PAC-Bayes, and the Fisher–Rao metric, respectively. Novak *et al.* [27] employ the cover number and the Rademacher complexity to obtain a spectrally normalized generalization bound. Neyshabur *et al.* [28] receive a similar result by employing the PAC-Bayes theory. However, all of these works study the generalization of a general neural network but not the influence of residual connections. The proof techniques in our work are inspired by [27], but we also analyze the influence of residual connections.

Some works have studied ResNet from the theoretical perspective, but the understanding of its generalization is still elusive. Li *et al.* [29] visualize the loss function curvature of ResNet. The empirical results demonstrate that residual connections can promote flat minimizers and also prevent the transition to chaotic behavior. Shamir *et al.* [30] prove that for a single residual block, all the local minima in the optimization are not above those of a linear predictor (equivalent to a one-layer network). Yun *et al.* [31] extend this result to deep ResNets. Kawaguchi and Bengio [32] prove that all local minima of deep ResNets are no worse than the global minima of corresponding classical machine learning models under two assumptions: 1) the output dimension satisfies $d_y \leq \min(d_x, d_z)$, where d_y is the output dimension, d_x is the input dimension, and d_z is the dimension of the outputs of the residual functions; and 2) the loss surface is convex and differentiable with respect to the hypothesis function. Other advances include [29] and [33]–[38].

III. Preliminary

In this section, we present the preliminaries necessary to develop our theory. It has two main parts: 1) important concepts to express the generalization capability of an algorithm and 2) a margin-based generalization bound for multiclass classification algorithms. The preliminaries provide general tools for us to theoretically analyze multiclass classification algorithms.

Generalization bound is the upper bound of the generalization error that is defined as the difference between the expected risk (or, equivalently, the expectation of test error) of the output hypothesis of an algorithm and the corresponding empirical risk (or, equivalently, the training error).[2] Thus, the generalization bound quantitatively expresses the generalization capability of an algorithm.

As indicated by the principle of Occam's razor, there is a negative correlation between the generalization capability of an algorithm and the complexity of the hypothesis space that the algorithm can compute. Two fundamental measures for the complexity are the VC-dimension and the Rademacher complexity (see [39] and [40]). Furthermore, they can be upper-bounded by another important complexity covering number (see [41] and [42]). Recent advances include the local Rademacher complexity and algorithmic stability (see [43]–[45]). These theoretical tools have been widely applied to analyze many algorithms (see [46]–[49]).

To formally formularize the problem, we first define the margin operator $\mathcal{M}$ for the k-class classification task as

$$\mathcal{M} : \mathbb{R}^k \times \{1, \ldots, k\} \to \mathbb{R}, \quad (v, y) \mapsto v_y - \max_{i \neq y} v_i. \tag{1}$$

Then, the ramp loss $l_\lambda : \mathbb{R} \to \mathbb{R}^+$ is defined as

$$l_\lambda(r) = \begin{cases} 0, & r < -\lambda \\ 1 + r/\lambda, & -\lambda \leq r \leq 0 \\ 1, & r > 0. \end{cases} \tag{2}$$

Furthermore, given a hypothesis function $F : \mathbb{R}^{n_0} \to \mathbb{R}^k$ for the k-class classification, the empirical ramp risk on a data set $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is defined as

$$\hat{\mathcal{R}}_\lambda(F) = \frac{1}{n}\sum_{i=1}^{n}(l_\lambda(-\mathcal{M}(F(x_i), y_i))). \tag{3}$$

Empirical ramp risk $\hat{\mathcal{R}}_\lambda(F)$ expresses the training error of the hypothesis function F on the data set D.

Meanwhile, the expected risk (and also, equivalently, the expected test error) of the hypothesis function F under 0-1 loss is

$$\Pr\{\arg\max_i F(x)_i \neq y\} \tag{4}$$

where x is an arbitrary feature, y is the corresponding correct label, and the probability is in term of the pair (x, y).

Suppose a hypothesis space $\mathcal{H}|_D$ is constituted by all hypothesis functions that can be computed by a neural network trained on a data set D. The empirical Rademacher complexity of the hypothesis space $\mathcal{H}|_D$ is defined as

$$\hat{\mathfrak{R}}(\mathcal{H}|_D) = \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{F\in\mathcal{H}|_D} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i F(x_i, y_i)\right] \tag{5}$$

where $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)$ and ϵ_i is a uniform variable on $\{-1, +1\}$. A margin-based bound for multiclass classifiers is given as the following lemma.

Lemma 1 [19, Lemma 3.1]: Given a function set $\mathcal{H}$ that $\mathcal{H} \ni F : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ and any margin $\lambda > 0$, define

$$\mathcal{H}_\lambda \triangleq \{(x, y) \mapsto l_\lambda(-\mathcal{M}(F(x), y)) : F \in \mathcal{H}\}. \tag{6}$$

Then, for any $\delta \in (0, 1)$, with probability of at least $1-\delta$ over a data set D of size n, every $F \in \mathcal{H}|_D$ satisfies

$$\Pr\{\arg\max_i F(x)_i \neq y\} - \hat{\mathcal{R}}_\lambda(F) \leq 2\hat{\mathfrak{R}}(\mathcal{H}_\lambda|_D) + 3\sqrt{\frac{\log(1/\delta)}{2n}}. \tag{7}$$

This generalization bound is developed by employing the Rademacher complexity that is upper-bounded by covering number (see [23], [42], [50], and [51]). A detailed proof can be found in [19]. Lemma 1 relates the generalization capability (expressed by $\Pr\{\arg\max_i F(x)_i \neq y\} - \hat{\mathcal{R}}_\lambda(F)$) to the hypothesis complexity (expressed by $\hat{\mathfrak{R}}(\mathcal{H}_\lambda|_D)$). It suggests that if one can find an upper bound for empirical Rademacher complexity, an upper bound of generalization error can be straightly obtained. Bartlett *et al.* give a lemma that bounds empirical Rademacher complexity via upper-bounding covering number [19] derived from the Dudley entropy integral bound [41], [52]. Specifically, if the ε-covering number $\mathcal{N}(\mathcal{H}_\lambda|_D, \varepsilon, \|\cdot\|)$ is defined as the minimum number of the balls with radius $\varepsilon > 0$ needed to cover the space $\mathcal{H}_\lambda|_D$ with a norm $\|\cdot\|$, the lemma is as follows.

Lemma 2 [19, Lemma A.5]: Suppose $\mathbf{0} \in \mathcal{H}_\lambda$, while all conditions in Lemma 1 hold. Then

$$\hat{\mathfrak{R}}(\mathcal{H}_\lambda|_D) \leq \inf_{\alpha>0}\left(\frac{4\alpha}{\sqrt{n}} + \frac{12}{n}\int_{\alpha}^{\sqrt{n}} \sqrt{\log\mathcal{N}(\mathcal{H}_\lambda|_D, \varepsilon, \|\cdot|_2)}\mathrm{d}\varepsilon\right). \tag{8}$$

Combining Lemmas 1 and 2, we relate the covering bound of an algorithm to the generalization bound of the algorithm. In the rest of this article, we develop generalization bounds for deep neural networks with residual connections via upper-bounding covering numbers.

To avoid technicalities, the measurability/integrability issues are ignored throughout this article. Moreover, Fubini's theorem is assumed to be applicable for any integration with respect to multiple variables that the order of integrations is exchangeable.

IV. Stem-Vine Framework

This section provides a notation system for deep neural networks with residual connections. Motivated by the topological structure, we call it the stem-vine framework.

In general, deep neural networks are constructed by connecting many weight matrices and nonlinear operators (nonlinearities), including ReLU, sigmoid, and max-pooling. In this

[2] Some works define generalization error as the expected error of an algorithm (see, e.g., [23]). As the training error is fixed when both training data and the algorithm are fixed, this difference in definitions can only lead to a tiny difference in results. In this article, we select one for the brevity and would not limit any generality.

article, we consider a neural network constructed by adding multiple residual connections to a "chain-like" neural network that stacks a series of weight matrices and nonlinearities forward one by one. Motivated by the topological structure, we call the chain-like part as the stem of the neural network and call the residual connections as the vines. Both stems and vines themselves are constructed by stacking multiple weight matrices and nonlinearities.

We denote the weight matrices and the nonlinearities in the stem S, respectively, as

$$A_i \in \mathbb{R}^{n_{i-1}\times n_i} \tag{9}$$

$$\sigma_j : \mathbb{R}^{n_j} \to \mathbb{R}^{n_j} \tag{10}$$

where $i = 1, \dots, L$, L is the number of weight matrices in the stem, $j = 1, \dots, L_N$, L_N is the number of nonlinearities in the stem, n_i is the dimension of the output of the ith weight matrix, n_0 is the dimension of the input data to the network, and n_L is the dimension of the output of the network. Thus, we can write the stem S as a vector to express the chain-like structure. Here, for the simplicity and without any loss of the generality, we give an example that the numbers of weight matrices and nonlinearities are equal[3], i.e., $L_N = L$, as the following equation:

$$S = (A_1, \sigma_1, A_2, \sigma_2, \dots, A_L, \sigma_L). \tag{11}$$

For the brevity, we give an index j to each vertex between a weight matrix and a nonlinearity and denote the jth vertex as $N(j)$. Specifically, we give the index 1 to the vertex that receives the input data and $L + L_N + 1$ to the vertex after the last weight matrix/nonlinearity. Taken (11) as an example, the vertex between the nonlinearity σ_{i-1} and the weight matrix A_i is denoted as $N(2i-1)$, and the vertex between the weight matrix A_i and the nonlinearity σ_i is denoted as $N(2i)$.

Vines are constructed to connect the stem at two different vertexes. There could be over one vine connecting the same pair of vertexes. Therefore, we use a triple vector (s, t, i) to index the ith vine connecting the vertexes $N(s)$ and $N(t)$ and denote the vine as $V(s, t, i)$. All triple vectors (s, t, i) constitute an index set I_V, i.e., $(s, t, i) \in I_V$. Similar to the stem, each vine $V(s, t, i)$ is also constructed by a series of weight matrices $A_1^{s,t,i}, \dots, A_{L^{s,t,i}}^{s,t,i}$ and nonlinearities $\sigma_1^{s,t,i}, \dots, \sigma_{L_N^{s,t,i}}^{s,t,i}$, where $L^{s,t,i}$ is the number of weight matrices in the vine, while $L_N^{u,v,i}$ is the number of the nonlinearities.

Multiplying by a weight matrix corresponds to an affine transformation on the data matrix. Also, nonlinearities induce nonlinear transformations. Through a series of affine transformations and nonlinear transformations, hierarchical features are extracted from the input data by neural networks. Usually, we use the spectrum norms of weight matrices and the Lipschitz constants of nonlinearities to express the intensities, respectively, of the affine transformations and the nonlinear transformations. We call a function $f(x)$ is the ρ-Lipschitz continuous if, for any x_1 and x_2 in the support domain of $f(x)$, it holds that

$$\|f(x_1) - f(x_2)\|_f \le \rho \|x_1 - x_2\|_x \tag{12}$$

where $\|\cdot\|_f$ and $\|\cdot\|_x$ are, respectively, the norms defined on the spaces of $f(x)$ and x. Fortunately, almost all nonlinearities normally used in neural networks are Lipschitz continuous, such as ReLU, max-pooling, and sigmoid (see [19]).

Many important tasks for deep neural networks can be categorized into multiclass classification. Suppose input examples $z_1, \dots, z_n$ are given, where $z_i = (x_i, y_i)$, $x_i \in \mathbb{R}^{n_0}$ is an instance, $y \in \{1, \dots, n_L\}$ is the corresponding label, and n_L is the number of the classes. Collect all instances $x_1, \dots, x_n$ as a matrix $X = (x_1, \dots, x_n)^T \in \mathbb{R}^{n\times n_0}$, where each row of X represents a data point. By employing optimization methods [usually, stochastic gradient decent (SGD)], neural networks are trained to fit the training data and then predict on test data. In mathematics, a trained deep neural network with all parameters fixed computes a hypothesis function $F : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$. A natural way to convert F to a multiclass classifier is to select the coordinate of $F(x)$ with the largest magnitude. In other words, for an instance x, the classifier is $x \mapsto \arg\max_i F(x)_i$. Correspondingly, the margin for an instance x labeled as y_i is defined as $F(x)_y - \max_{i\neq y} F(x)_i$. It quantitatively expresses the confidence of assigning a label to an instance.

To express F, we first define the functions, respectively, computed by the stem and vines. Specifically, we denote the function computed by a vine $V(s, t, i)$ as

$$F_V^{s,t,i}(X) = \sigma_{L^{u,v,i}}^{u,v,i}\left(A_{L^{u,v,i}}^{u,v,i}\sigma_{L^{u,v,i}-1}^{u,v,i}\left(\dots\sigma_1\left(A_1^{u,v,i}X\right)\dots\right)\right). \tag{13}$$

Similarly, the stem computes a function as the following equation:

$$F_S(X) = \sigma_L(A_L\sigma_{L-1}(\dots\sigma_1(A_1X)\dots)). \tag{14}$$

Furthermore, we denote the output of the stem at the vertex $N(j)$ as the following equation:

$$F_S^j(X) = \sigma_j(A_j\sigma_{j-1}(\dots\sigma_1(A_1X)\dots)). \tag{15}$$

$F_S^j(X)$ is also the input of the rest part of the stem. Eventually, with all residual connections, the output hypothesis function $F^j(X)$ at the vertex $N(j)$ is expressed by the following equation:

$$F^j(X) = F_S^j(X) + \sum_{(u,j,i)\in I_V} F_V^{u,j,i}(X). \tag{16}$$

Apparently

$$F_S(X) = F_S^L(X), \quad F(X) = F^L(X). \tag{17}$$

Naturally, we call this notation system as the stem-vine framework, and Fig. 1 shows an example.

[3] If two weight matrices, A_i and A_{i+1}, are connected directly without a nonlinearity between them, we define a new weight matrix $A = A_i \cdot A_{i+1}$. The situations that nonlinearities are directly connected are similar, as the composition of any two nonlinearities is still a nonlinearity. Meanwhile, the number of weight matrices does not necessarily equal the number of nonlinearities. Sometimes, if a vine connects the stem at a vertex between two weight matrices (or two nonlinearities), the number of the weight matrices (nonlinearities) would be larger than the number of nonlinearities (weight matrices). Taken the 34-layer ResNet as an example, a vine connects the stem between two nonlinearities σ_{33} and σ_{34}. In this situation, we cannot merge the two nonlinearities, so the number of nonlinearities is larger than the number of weight matrices.

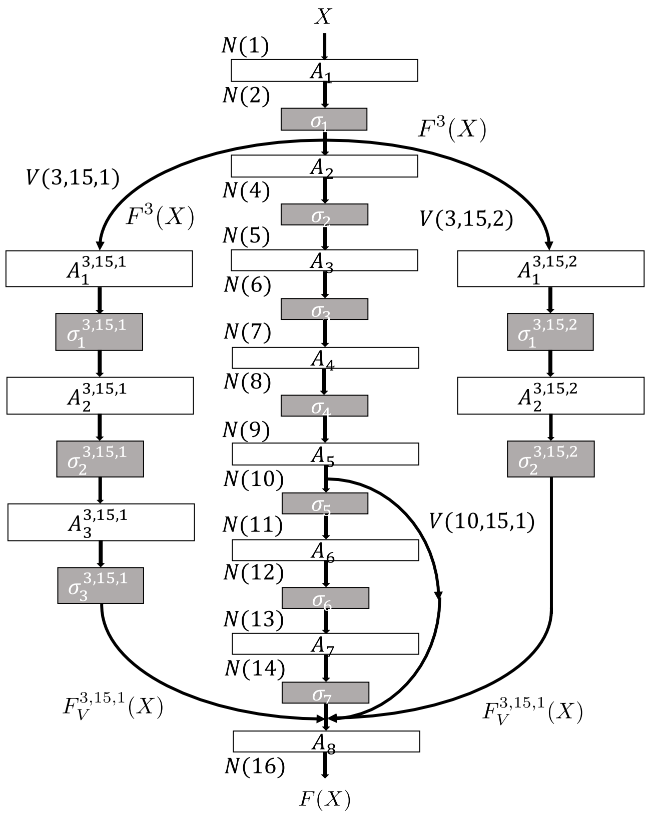

Fig. 1. Deep neural network with residual connections under the stem-vine framework.

V. Generalization Bound

In this section, we study the generalization capability of deep neural networks with residual connections and provide a generalization bound for ResNet as an exemplary case. This generalization bound is derived upon the margin-based multiclass bound given by Lemmas 1 and 2. Indicated by Lemmas 1 and 2, a natural way to approach the generalization bound is to explore the covering number of the corresponding hypothesis space. Motivated by this intuition, we first propose an upper bound of the covering number (or briefly, covering bound) generally for any deep neural networks under the stem-vine framework. Then, as an exemplary case, we obtain a covering bound for ResNet. Applying Lemmas 1 and 2, a generalization bound for ResNet is eventually presented. The proofs for covering bounds will be given in Section VI.

As a convention, when we introduce a new structure to boost the training performance (including training accuracy and training time), we should be very careful to prevent the algorithm from overfitting (which manifests itself as an unacceptably large generalization error). ResNet introduces "loops" into chain-like neural networks by residual connections and, therefore, becomes a more complex model. Empirical results indicate that the residual connections significantly reduce the training error and accelerate the training speed, while maintains generalization capability at the same time. However, there is, so far, no theoretical evidence to explain/support the empirical results.

Our result in covering bound indicates that when the total number of weight matrices is fixed, no matter where the weight matrices are (either in the stem or in the vines, and even when there is no vine at all), the complexities of the hypothesis spaces that computed by deep neural networks remain invariant. Combing various classic results in statistical learning theories (see Lemmas 1 and 2), our results further indicate that the generalization capability of deep neural networks with residual connections could be as equivalently good as the ones without any residual connection at least in the worst cases. Our theoretical result gives an insight into why the deep neural networks with residual connections have equivalently good generalization capability compared with the chain-like ones while having competitive training performance.

A. Covering Bound for Deep Neural Networks With Residuals

In this section, we give a covering bound generally for any deep neural network with residual connections.

Theorem 1 [Covering Bound for Deep Neural Network]: Suppose a deep neural network is constituted by a stem and a series of vines.

For the stem, let $(\varepsilon_1, \ldots, \varepsilon_L)$ be given, along with L_N fixed nonlinearities $(\sigma_1, \ldots, \sigma_{L_N})$. Suppose the L weight matrices $(A_1, \ldots, A_L)$ lies in $\mathcal{B}_1 \times \ldots \times \mathcal{B}_L$, where $\mathcal{B}_i$ is a ball centered at 0 with radius of s_i, i.e., $\|A_i\| \le s_i$. Suppose the vertex that directly follows the weight matrix A_i is $N(M(i))$ ($M(i)$ is the index of the vertex). All $M(i)$ constitute an index set I_M. When the output $F_{M(j-1)}(X)$ of the weight matrix A_{j-1} is fixed, suppose all output hypotheses $F_{M(j)}(X)$ of the weight matrix A_j constitute a hypothesis space $\mathcal{H}_{M(j)}$ with an $\varepsilon_{M(j)}$-cover $\mathcal{W}_{M(j)}$ with covering number $\mathcal{N}_{M(j)}$. Specifically, we define $M(0) = 0$ and $F_0(X) = X$.

Each vine $V(u, v, i)$, $(u, v, i) \in I_V$ is also a chain-like neural network that constructed by multiple weight matrices $A_j^{u,v,i}$, $j \in \{1, \ldots, L^{u,v,i}\}$ and nonlinearities $\sigma_j^{u,v,i}$, $j \in \{1, \ldots, L_N^{u,v,i}\}$. Suppose for any weight matrix $A_j^{u,v,i}$, there is a $s_j^{u,v,i} > 0$ such that $\|A_j^{u,v,i}\|_\sigma \le s_j^{u,v,i}$. Also, all nonlinearities $\sigma_j^{u,v,i}$ are the Lipschitz continuous. Similar to the stem, when the input of the vine $F_u(X)$ is fixed, suppose the vine $V(u, v, i)$ computes a hypothesis space $\mathcal{H}_V^{u,v,i}$, constituted by all hypotheses $F_V^{u,v,i}(X)$, and has an $\varepsilon_{u,v,i}$-cover $\mathcal{W}_V^{u,v,i}$ with covering number $\mathcal{N}_V^{u,v,i}$.

Eventually, we denote the hypothesis space computed by the neural network is $\mathcal{H}$. Then, there exists an ε in terms of ε_i, $i = \{1, \ldots, L\}$ and $\varepsilon_{u,v,i}$, $(u, v, i) \in I_V$ such that the following inequality holds:

$$\mathcal{N}(\mathcal{H}, \varepsilon, \|\cdot\|) \le \prod_{j=1}^{L} \sup_{F_{M(j)}} \mathcal{N}_{M(j+1)} \prod_{(u,v,i)\in I_V} \sup_{F_u} \mathcal{N}_V^{u,v,i}. \qquad (18)$$

A detailed proof will be given in Section VI-C.

As vines are chain-like neural networks, we can further obtain an upper bound for $\sup_{F_u} \mathcal{N}_V^{u,v,i}$ via a lemma slightly modified from [19]. The lemma is summarized as follows.

Lemma 3: (Covering Bound for Chain-Like Deep Neural Network, see [19, Lemma A.7]): Suppose there are L weight matrices in a chain-like neural network. Let $(\varepsilon_1, \ldots, \varepsilon_L)$ be given. Suppose the L weight matrices $(A_1, \ldots, A_L)$ lie in $\mathcal{B}_1 \times \cdots \times \mathcal{B}_L$, where $\mathcal{B}_i$ is a ball centered at 0 with the radius of s_i, i.e., $\mathcal{B}_i = \{A_i : \|A_i\| \leq s_i\}$. Furthermore, suppose the input data matrix X is restricted in a ball centered at 0 with the radius of B, i.e., $\|X\| \leq B$. Suppose F is a hypothesis function computed by the neural network. If we define

$$\mathcal{H} = \{F(X) : A_i \in \mathcal{B}_i\} \tag{19}$$

where $i = 1, \ldots, L$ and $t \in \{1, \ldots, L^{u,v,s}\}$. Let $\varepsilon = \sum_{j=1}^{L} \varepsilon_j \rho_j \prod_{l=j+1}^{L} \rho_l s_l$. Then, we have the following inequality:

$$\mathcal{N}(\mathcal{H}, \varepsilon, \|\cdot\|) \leq \prod_{i=1}^{L} \sup_{\mathbf{A}_{i-1} \in \boldsymbol{\mathcal{B}}_{i-1}} \mathcal{N}_i \tag{20}$$

where $\mathbf{A}_{i-1} = (A_1, \ldots, A_{i-1})$, $\boldsymbol{\mathcal{B}}_{i-1} = \mathcal{B}_1 \times \cdots \times \mathcal{B}_{i-1}$, and

$$\mathcal{N}_i = \mathcal{N}(\{A_i F_{\mathbf{A}_{i-1}}(X) : A_i \in \mathcal{B}_i\}\varepsilon_i, \|\cdot\|). \tag{21}$$

Remark 1: The mapping induced by a chain-like neural network can be formularized as the composition of a series of affine/nonlinear transformations. The proof of Lemma 3, thus, can decompose the covering bound for a chain-like neural network into the product of the covering bounds for all layers (see a detailed proof in [19]). However, residual connections introduce paralleling structures into neural networks. Therefore, the computed mapping cannot be directly expressed as a series of compositions of affine/nonlinear transformations. Instead, to approach a covering bound for the whole network, we are facing many additions of function spaces [see (16)], where the former results cannot be straightly applied. To address this issue, we provide a novel proof collected in Section VI-C.

Contrary to the different proofs, the result for deep neural networks with residual connections shares similarities with the one for the chain-like network [see (18) and (20)]. The similarities lead to the property summarized as follows.

> The influences on the hypothesis complexity of weight matrices are in the same way, no matter whether they are in the stem or the vines. Specifically, adding an identity vine could not affect the hypothesis complexity of the deep neural network.

As indicated by (20) in Lemma 3, the covering number of the hypothesis computed by a chain-like neural network (including the stem and all the vines) is upper-bounded by the product of the covering number of all single layers. Specifically, the contribution of the stem on the covering bound is the product of a series of covering numbers, i.e., $\prod_{j=1}^{L} \sup_{F_{M(j)}} \mathcal{N}_{M(j+1)}$. In the meantime, applying (20) in Lemma 3, the contribution $\sup_{F_u} \mathcal{N}_V^{u,v,i}$ of the vine $V(u,v,i)$ can also be decomposed as the product of a series of covering numbers. Apparently, the contributions, respectively, by the weight matrices in the stem and the ones in the vines have similar formulations. This result gives an insight that residuals would not undermine the generalization capability of deep neural networks. Also, if a vine $V(u,v,i)$ is an identity mapping, the term in (18) that relates to it is definitely 1, i.e., $\mathcal{N}_V^{u,v,i} = 1$. This is because there is no parameter to tune in an identity vine. This result gives an insight that adding an identity vine to a neural network would not affect the hypothesis complexity. However, it is worth noting that the vines could influence the part of the stem in the covering bound, i.e., $\mathcal{N}_{M(j+1)}$ in (18). The mechanism of the cross-influence between the stem and the vines is an open problem.

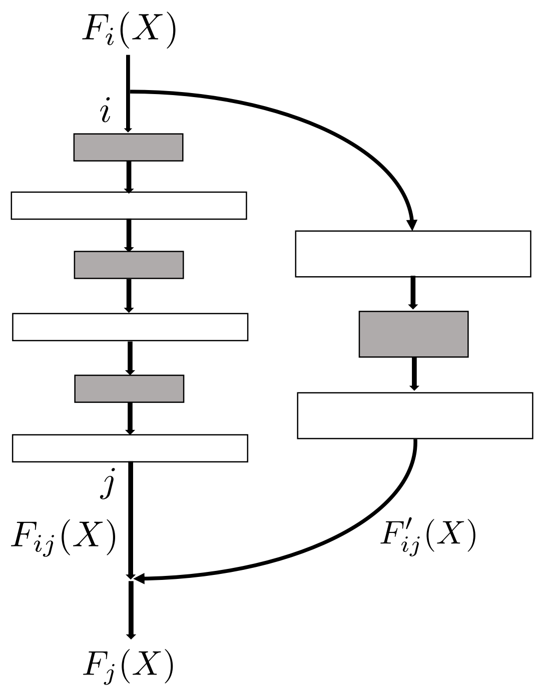

Fig. 2. Illustration of influence of a residual connection to the hypothesis space.

Intuitively, introducing residual connections to a neural network may not change the hypothesis space. Here, we discuss the following case as an example. Consider a residual connection that links nodes i and j. Suppose the hypothesis functions computed in the nodes i and j are F_i and F_j, respectively. Also, we denote the hypothesis functions computed by the bone and residual connection between nodes i and j as $F_{i,j}$ and $F'_{i,j}$, respectively. See an illumination in Fig. 2. Then, we have the following equation:

$$F_j = F_{i,j} \circ F_i + F'_{i,j} \circ F_i.$$

Usually, the residual connection is a simpler subnetwork of the bone part. Therefore, the hypothesis space constituted by all $F'_{i,j}$ is a subspace of the one of $F_{i,j}$. Thus, the hypothesis space constituted by all $F'_{i,j} \circ F_i$ is a subspace of the one of $F_{i,j} \circ F_i$. In other words, the hypothesis space computed by a residual connection is only a subspace computed by the bone part. Introducing a residual connection is merging the two spaces by addition operation, which obtains the larger space. This property guarantees that introducing residual connections may not change the hypothesis space.

B. Covering Bound for ResNet

As an example, we analyze the generalization capability of the 34-layer ResNet. Analysis of other deep neural networks under the stem-vine framework is similar. For convenience, we give a detailed illustration of the 34-layer ResNet under the stem-vine framework in Fig. 3.

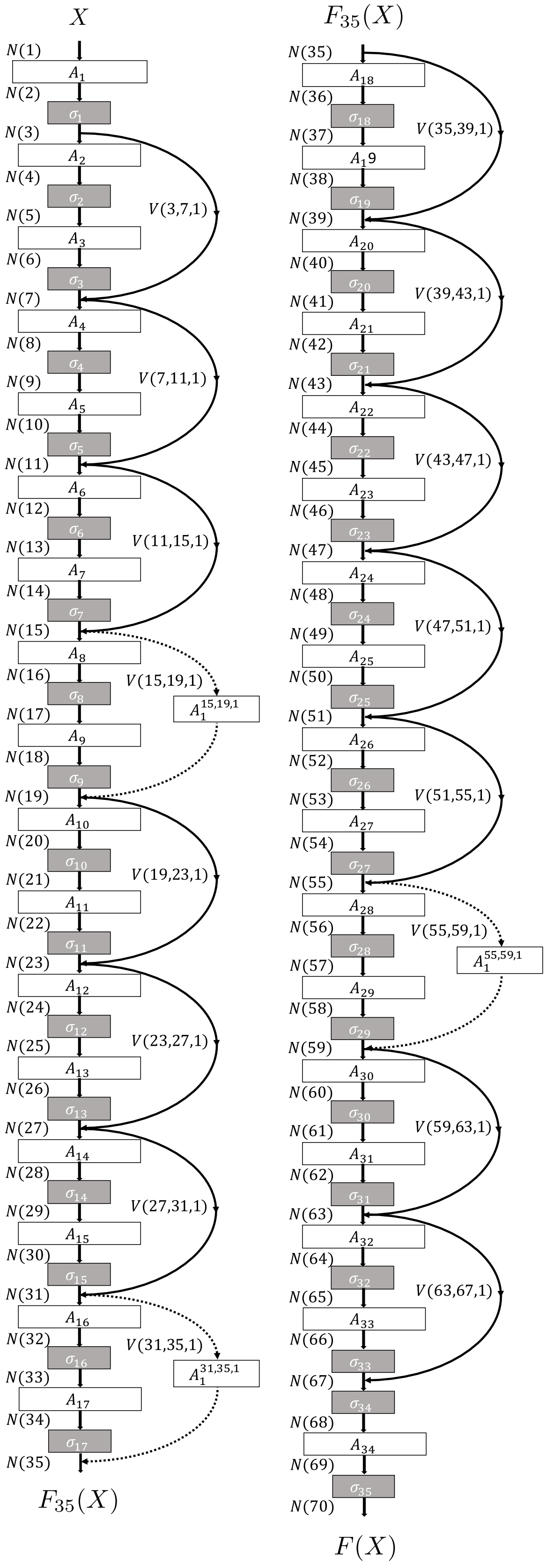

Fig. 3. 34-layer ResNet under the Stem-Vine framework.

There are one 34-layer stem and 16 vines in the 34-layer ResNet. Each layer in the stem contains one weight matrix and several Lipschitz-continuous nonlinearities. For most layers with over one nonlinearity, the multiple nonlinearities are connected one by one directly; we merge the nonlinearities as one single nonlinearity.[4] However, the vine links the stem at a vertex between two nonlinearities after the 33th weight matrix, and thus, we cannot merge the two nonlinearities. Hence, the stem of ResNet can be expressed as follows:

$$S_{\text{res}} = (A_1, \sigma_1, \ldots, A_{33}, \sigma_{33}, \sigma_{34}, A_{34}, \sigma_{35}). \tag{22}$$

From the vertex that receives the input data to the vertex that outputs classification functions, there are $34 + 35 + 1 = 70$ vertexes (34 is the number of weight matrices and 35 is the number of nonlinearities). We denote them as $N(1)$ to $N(70)$. In addition, we assume the norm of the weight matrix A_i has an upper bound s_i, i.e., $\|A_i\|_\sigma \le s_i$, while the Lipschitz constant of the nonlinearity σ_i is denoted as b_i.

Under the stem-vine framework, the 16 vines in ResNet are, respectively, denoted as $V(3,7,1), V(7,11,1), \ldots, V(63,67,1)$. Among these 16 vines, there are three vines—$V(15,19,1)$, $V(31,35,1)$, and $V(55,59,1)$—that, respectively, contain one weight matrix, while all others are identity mappings. Let us denote the weight matrices in the vines $V(15,19,1)$, $V(31,35,1)$, and $V(55,59,1)$, respectively, as $A_1^{15,19,1}$, $A_1^{31,35,1}$, and $A_1^{55,59,1}$. Suppose the norms of $A_1^{15,19,1}$, $A_1^{31,35,1}$, and $A_1^{55,59,1}$ are, respectively, upper-bounded by $s_1^{15,19,1}$, $s_1^{31,35,1}$, and $s_1^{55,59,1}$. Denote the reference matrices that correspond to weight matrices $(A_1, \ldots, A_{34})$ as $(M_1, \ldots, M_{34})$. Suppose the distance between each weight matrix A_i and the corresponding reference matrix M_i is upper-bounded by b_i, i.e., $\|A_i^T - M_i^T\| \le b_i$. Similarly, suppose there are reference matrices $M_1^{s,t,1}$, $(s,t) \in \{(15,19),(31,35),(55,59)\}$, respectively, for weight matrices $A_1^{s,t,1}$, and the distance between $A_1^{s,t}$ and $M_1^{s,t,1}$ is upper-bounded by $b_1^{s,t,1}$, i.e., $\|(A_i^{s,t,1})^T - (M_i^{s,t,1})^T\| \le b_1^{s,t,1}$. We then have the following lemma.

Lemma 4 [Covering Number Bound for ResNet]: For a ResNet R that satisfies all the above-mentioned conditions, suppose the hypothesis space is $\mathcal{H}_R$. Then, we have

$$\begin{aligned}\log \mathcal{N}&(\mathcal{H}_R, \varepsilon, \|\cdot\|)\\ &\le \sum_{u\in\{15,31,55\}} \frac{\left(b_1^{u,u+4,1}\right)^2 \|F_u(X^T)^T\|_2^2}{\varepsilon_{u,u+4,1}^2}\log(2W^2)\\ &\quad+\sum_{j=1}^{34}\frac{b_j^2\|F_{2j-1}(X^T)^T\|_2^2}{\varepsilon_{2j+1}^2}\log(2W^2)\\ &\quad+\frac{b_{34}^2\|F_{68}(X^T)^T\|_2^2}{\varepsilon_{70}^2}\log(2W^2)\end{aligned} \tag{23}$$

[4] Specifically, if there are two Lipschitz-continuous nonlinearities connected directly somewhere in the stem, such as one max-pooling and one ReLU, we compose the two nonlinearities as one single nonlinearity. The composition is well-defined, as the composition of two Lipschitz-continuous nonlinearities is still a Lipschitz-continuous nonlinearity. The Lipschitz constant of the composition function is the product of the Lipschitz constants, respectively, of the two nonlinearities. In addition, the composition would not limit any generality, as in our theory, different nonlinearities with the same Lipschitz constant have the same influence on the generalization bound (this argument is supported by (23) of Lemma 4, Theorem 2, and so on).

where $\mathcal{N}(\mathcal{H}_R, \varepsilon, \|\cdot\|)$ is the ε-covering number of $\mathcal{H}_R$. When $j = 1, \ldots, 16$

$$\begin{aligned}
&\|F_{4j+1}(X)\|_2^2 \\
&\quad \le \|X\|^2 \rho_1^2 s_1^2 \rho_{2j}^2 s_{2j}^2 \prod_{\substack{1 \le i \le j-1 \\ i \notin \{4,8,14\}}} \left(\rho_{2i}^2 s_{2i}^2 \rho_{2i+1}^2 s_{2i+1}^2 + 1\right) \\
&\qquad \times \prod_{\substack{1 \le i \le j-1 \\ i \in \{4,8,14\}}} \left[\rho_{2i}^2 s_{2i}^2 \rho_{2i+1}^2 s_{2i+1}^2 + \left(s_1^{4i-1,4i+3,1}\right)^2\right] \qquad (24)
\end{aligned}$$

and

$$\begin{aligned}
&\|F_{4j+3}(X)\|_2^2 \\
&\quad \le \|X\|^2 \rho_1^2 s_1^2 \prod_{\substack{1 \le i \le j \\ i \notin \{4,8,14\}}} \left(\rho_{2i}^2 s_{2i}^2 \rho_{2i+1}^2 s_{2i+1}^2 + 1\right) \\
&\qquad \times \prod_{\substack{1 \le i \le j \\ i \in \{4,8,14\}}} \left[\rho_{2i}^2 s_{2i}^2 \rho_{2i+1}^2 s_{2i+1}^2 + \left(s_1^{4i-1,4i+3,1}\right)^2\right] \qquad (25)
\end{aligned}$$

and specifically

$$\begin{aligned}
&\|F_{68}(X^T)^T\|_2^2 \\
&\quad \le \|X\|^2 \rho_1^2 s_1^2 \rho_{34}^2 \prod_{\substack{1 \le i \le 16 \\ i \notin \{4,8,14\}}} \left(\rho_{2i}^2 s_{2i}^2 \rho_{2i+1}^2 s_{2i+1}^2 + 1\right) \\
&\qquad \times \prod_{\substack{1 \le i \le 16 \\ i \in \{4,8,14\}}} \left[\rho_{2i}^2 s_{2i}^2 \rho_{2i+1}^2 s_{2i+1}^2 + \left(s_1^{4i-1,4i+3,1}\right)^2\right]. \qquad (26)
\end{aligned}$$

Also, when $j = 1, \ldots, 16$

$$\begin{aligned}
\varepsilon_{4j+1} = (1 + s_1)\rho_1(1 + s_{2j})\rho_{2j} &\prod_{\substack{1 \le i \le j-1 \\ i \notin \{4,8,14\}}} [(*) + 1] \\
&\times \prod_{\substack{1 \le i \le j-1 \\ i \in \{4,8,14\}}} \left[(*) + 1 + s_1^{4i-1,4i+3,1}\right] \qquad (27)
\end{aligned}$$

and

$$\begin{aligned}
\varepsilon_{4j+3} = (1 + s_1)\rho_1 &\prod_{\substack{1 \le i \le j \\ i \notin \{4,8,14\}}} [(*) + 1] \\
&\times \prod_{\substack{1 \le i \le j \\ i \in \{4,8,14\}}} \left[(*) + 1 + s_1^{4i-1,4i+3,1}\right] \qquad (28)
\end{aligned}$$

and for $u = 15, 31, 55$

$$\varepsilon_{u,u+4,1} = \varepsilon_u\left(1 + s_1^{u,u+4,1}\right). \qquad (29)$$

In above-mentioned equations/inequalities

$$\begin{aligned}
\bar{\alpha} = (s_1 + 1)\rho_1\rho_{34}(s_{34} + 1)\rho_{35} &\prod_{\substack{1 \le i \le 16 \\ i \notin \{4,8,14\}}} [(*) + 1] \\
&\times \prod_{i \in \{4,8,14\}} \left[(*) + s_1^{4i-1,4i+3,1} + 1\right] \qquad (30)
\end{aligned}$$

and

$$(*) = \rho_{2i}(s_{2i} + 1)\rho_{2i+1}(s_{2i+1} + 1). \qquad (31)$$

A detailed proof is omitted and will be given in Section VI-C.

C. Generalization Bound for ResNet

Lemmas 1 and 2 guarantee that when the covering number of a hypothesis space is upper-bounded, the corresponding generalization error is upper-bounded. Therefore, combining the covering bound for ResNet given by Lemma 4, a generalization bound for ResNet is straightforward. In this section, the generalization bound is summarized as Theorem 2.

For the brevity, we rewrite the radius ε_{2j+1} and $\varepsilon_{u,u+4,1}$ as follows:

$$\varepsilon_{2j+1} = \hat{\varepsilon}_{2j+1} \qquad (32)$$

$$\varepsilon_{u,u+4,1} = \hat{\varepsilon}_{u,u+4,1}\varepsilon. \qquad (33)$$

In addition, we rewrite (23) of Lemma 4 as the following inequality:

$$\log \mathcal{N}(\mathcal{H}, \varepsilon, \|\cdot\|) \le \frac{R}{\varepsilon^2} \qquad (34)$$

where

$$\begin{aligned}
R = &\sum_{u \in \{15,31,55\}} \frac{\left(b_1^{u,u+4,1}\right)^2 \|F_u(X^T)^T\|_2^2}{\hat{\varepsilon}_{u,u+4,1}^2} \log(2W^2) \\
&+ \sum_{j=1}^{33} \frac{b_j^2 \|F_{2j-1}(X^T)^T\|_2^2}{\hat{\varepsilon}_{2j+1}^2} \log(2W^2) \\
&+ \frac{b_{34}^2 \|F_{68}(X^T)^T\|_2^2}{\hat{\varepsilon}_{70}^2} \log(2W^2). \qquad (35)
\end{aligned}$$

Then, we can obtain the following theorem.

Theorem 2 [Generalization Bound for ResNet]: Suppose a ResNet satisfies all conditions in Lemma 4. Suppose a given series of examples $(x_1, y_1), \ldots, (x_n, y_n)$ are arbitrary independent identically distributed (i.i.d.) variables drawn from any distribution over $\mathcal{R}^{n_0} \times \{1, \ldots, n_L\}$. Suppose hypothesis function $F_{\mathcal{A}} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ is computed by a ResNet with weight matrices $\mathcal{A} = (A_1, \ldots, A_{34}, A_1^{15,19,1}, A_1^{31,35,1}, A_1^{55,59,1})$. Then, for any margin $\lambda > 0$ and any real $\delta \in (0, 1)$, with probability at least $1 - \delta$, we have the following inequality:

$$\begin{aligned}
&\Pr\{\arg\max_i F(x)_i \ne y\} \\
&\quad \le \hat{\mathcal{R}}_\lambda(F) + \frac{8}{n^{\frac{3}{2}}} + \frac{36}{n}\sqrt{R}\log n + 3\sqrt{\frac{\log(1/\delta)}{2n}} \qquad (36)
\end{aligned}$$

where R is defined as (35).

This generalization bound is established via the hypothesis complexity, which is measured by the covering number of the hypothesis space. This relationship is characterized by Lemma 1. Intuitively, Lemma 1 follows the principle of Occam's razor, which demonstrates a negative correlation between the generalization ability of an algorithm and its hypothesis complexity. Directly applying Lemma 1 with the covering number bound (Theorem 1), we can obtain Theorem 2. A proof is omitted here and will be given in Section VI-E.

Indicated by Theorem 2, the generalization bound of ResNet relies on its covering bound. Specifically, when the sample size n and the probability δ are fixed, the generalization error satisfies that

$$\Pr\{\arg\max_i F(x)_i \ne y\} - \hat{\mathcal{R}}_\lambda(F) = \mathcal{O}(\sqrt{R}) \qquad (37)$$

where R expresses the magnitude of the covering number (R/ε^2 is an ε-covering bound). Combining the property generally for any neural network under the stem-vine framework, (37) gives two insights about the effects of residual connections on the generalization capability of neural networks: 1) the influences of weight matrices on the generalization capability are invariant, no matter where they are (either in the stem or in the vines) and 2) adding an identity vine could not affect the generalization. These results give a theoretical explanation of why ResNet has equivalently good generalization capability as the chain-like neural networks.

As indicated by (36), the expected risk (or, equivalently, the expectation of the test error) of ResNet equals the sum of the empirical risk (or, equivalently, the training error) and the generalization error. In the meantime, residual connections significantly reduce the training error of the neural network in many tasks. Our results, therefore, theoretically explain why ResNet has a significantly lower test error in these tasks.

D. Practical Implementation

Besides the sample size N, our generalization bound (36) has a positive correlation with the norms of all the weight matrices. Specifically, weight matrices with higher norms lead to a higher generalization bound of the neural network and, therefore, lead to a worse generalization ability. This feature induces a practical implementation that justifies the standard of technique weight decay.

Weight decay can be dated back to a paper by Krogh and Hertz [18] and is widely used in training deep neural networks. It uses the L_2-norm of all the weights as a regularization term to control the magnitude of the norms of the weights not to increase too much.

Remark 2: The technique of weight decay can improve the generalization ability of deep neural networks. It refers to adding the L_2-norm of the weights $w = (w_1, \ldots, w_D)$ to the objective function as a regularization term

$$\mathcal{L}'(w) = \mathcal{L}(w) + \frac{1}{2}\lambda \sum_{i=1}^{D} w_i^2$$

where λ is a tunable parameter, $\mathcal{L}(w)$ is the original objective function, and $\mathcal{L}'(w)$ is the objective function with weight decay.

The term $1/2\lambda \sum_{i=1}^{D} w_i^2$ can be easily reexpressed by the L_2 norms of all the weight matrices. Therefore, using weight decay can control the magnitude of the norms of all the weights matrices not to increase too much. Also, our generalization bound (36) provides a positive correlation between the generalization bound and the norms of all the weight matrices. Thus, this work gives a justification for why weight decay leads to a better generalization ability.

A recent systematic experiment conducted by Li *et al.* [29] studies the influence of weight decay on the loss surface of the deep neural networks. It trains a nine-layer VGGNet [14] on the data set CIFAR-10 [53] by employing stochastic gradient descent with batch sizes of 128 (0.26% of the training set of CIFAR-10) and 8192 (16.28% of the training set of CIFAR-10). The results demonstrate that by employing weight decay, SGD can find flatter minima[5] of the loss surface with lower test errors (see [29, p. 6, Fig. 3]). Other technical advances and empirical analysis include [56]–[59].

VI. Proofs

This section collects various proofs omitted in Section V. We first give a proof of the covering bound for an affine transformation induced by a single weight matrix. It is the foundation of the other proofs. Then, we provide a proof of the covering bound for deep neural networks under the stem-vine framework (see Theorem 1). Furthermore, we present a proof of the covering bound for ResNet (see Lemma 4). Eventually, we provide a proof of the generalization bound for ResNet (see Theorem 2).

A. Proof of the Covering Bound for the Hypothesis Space of a Single Weight Matrix

In this section, we provide an upper bound for the covering number of the hypothesis space induced by a single weight matrix A. This covering bound relies on the Maurey sparsification lemma [60] and has been introduced in machine learning by previous works (see [19], [50]).

Suppose a data matrix X is the input of a weight matrix A. All possible values of the output XA constitute a space. We use the following lemma to express the complexity of all XA via the covering number.

Lemma 5 [19, Lemma 3.2]: Let conjugate exponents (p, q) and (r, s) be given with $p \leq 2$, as well as positive reals (a, b, ε) and positive integer m. Let matrix $X \in \mathbb{R}^{n\times d}$ be given with $\|X\|_p \leq b$. Let $\mathcal{H}_A$ denote the family of matrices obtained by evaluating X with all choices of matrix A

$$\mathcal{H}_A \triangleq \{XA | A \in \mathbb{R}^{d\times m}, \|A\|_{q,s} \leq a\}. \tag{38}$$

Then

$$\log \mathcal{N}(\mathcal{H}_A, \varepsilon, \|\cdot\|_2) \leq \left\lceil \frac{a^2 b^2 m^{2/r}}{\varepsilon^2} \right\rceil \log(2dm). \tag{39}$$

B. Covering Bound for the Hypothesis Space of Chain-Like Neural Network

This section considers the upper bound for the covering number of the hypothesis space induced by the stem of a deep neural network. Intuitively, following the stem from the first vertex $N(1)$ to the last one $N(L)$, every weight matrices and nonlinearities increase the complexity of the hypothesis space that could be computed by the stem. Following this intuition, we use an induction method to approach the upper bound. The result is summarized as Lemma 3. This lemma is originally given in the work by Bartlett *et al.* [19]. Here, to make this work complete, we recall the main part of the proof but omit the part for ε.

Proof of Lemma 3: We use an induction procedure to prove the lemma.

[5] The flatness (or equivalently sharpness) of the loss surface around the minima is considered as an important index expressing the generalization ability. However, the mechanism remains elusive. For more details, please refer to [54] and [55].

1) The covering number of the hypothesis space computed by the first weight matrix A_1 can be straightly upper-bounded by Lemma 5.

2) The vertex after the jth nonlinearity is $N(2j+1)$. Suppose $\mathcal{W}_{2j+1}$ is an ε-cover of the hypothesis space $\mathcal{H}_{2j+1}$ induced by the output hypotheses in the vertex $N(2j+1)$. Suppose there is a weight matrix A_{j+1} that directly follows the vertex $N(2j+1)$. We then analyze the contribution of the weight matrix A_{j+1}. Assume that there exists an upper bound s_{j+1} of the norm of A_{j+1}. For any $F_{2j+1}(X) \in \mathcal{H}_{2j+1}$, there exists a $W(X) \in \mathcal{W}_{2j+1}$ such that

$$\|F_{2j+1}(X) - W(X)\| \le \varepsilon_{2j+1}. \tag{40}$$

Lemma 5 guarantees that for any $W(X) \in \mathcal{W}_{2j+1}$, there exists an ε_{2j+1}-cover $\mathcal{W}_{2j+2}(W)$ for the function space $\{W(X)A_{j+1} : W(X) \in \mathcal{W}_{2j+1}, \|A_{j+1}\| \le s_{j+1}\}$, i.e., for any $W'(X) \in \hat{\mathcal{H}}_{2j+1}$, there exists a $V(X) \in \{W(X)A_{j+1} : W(X) \in \mathcal{W}_{2j+1}, \|A_{j+1}\| \le s_{j+1}\}$ such that

$$\|W'(X) - V(X)\| \le \varepsilon_{2j+1}. \tag{41}$$

As for any $F'_{2j+1}(X) \in \mathcal{H}_{2j+2} \triangleq \{F_{2j+1}(X)A_{j+1} : F_{2j+1}(X) \in \mathcal{H}_{2j+1}, \|A_{j+1}\| \le c\}$, there is a $F_{2j+1}(X) \in \mathcal{H}_{2j+1}$ such that

$$F'_{2j+1}(X) = F_{2j+1}(X)A_{j+1}. \tag{42}$$

Thus, applying (40)–(42), we get the following inequality:

$$\begin{aligned}
&\|F'_{2j+1}(X) - V(X)\| \\
&\quad = \|F_{2j+1}(X)A_{j+1} - V(X)\| \\
&\quad = \|F_{2j+1}(X)A_{j+1} - W(X)A_{j+1} \\
&\qquad + W(X)A_{j+1} - V(X)\| \\
&\quad \le \|F_{2j+1}(X)A_{j+1} - W(X)A_{j+1}\| \\
&\qquad + \|W(X)A_{j+1} - V(X)\| \\
&\quad \le \|F_{2j+1}(X) - W(X)\|\|A_{j+1}\| + \varepsilon_{2j+1} \\
&\quad \le s_{j+1}\varepsilon_{2j+1} + \varepsilon_{2j+1} \\
&\quad = (s_{j+1}+1)\varepsilon_{2j+1}.
\end{aligned} \tag{43}$$

Therefore, $\bigcup_{W\in\mathcal{W}_{2j+1}} \mathcal{W}_{2j+2}(W)$ is a $(s_{j+1}+1)\varepsilon_{2j+1}$-cover of $\mathcal{H}_{2j+2}$. Let us denote $(s_{j+1}+1)\varepsilon_{2j+1}$ as ε_{2j+2}. Apparently

$$\begin{aligned}
&\mathcal{N}(\mathcal{H}_{2j+2}, \varepsilon_{2j+2}, \|\cdot\|) \\
&\quad \le \left| \bigcup_{W\in\mathcal{W}_{2j+1}} \mathcal{W}_{2j+2}(W) \right| \\
&\quad \le \left|\mathcal{W}_{2j+1}\right| \cdot \sup_{W\in\mathcal{W}_{2j+1}} \left|\mathcal{W}_{2j+2}(W)\right| \\
&\quad \le \mathcal{N}(\mathcal{H}_{2j+1}, \varepsilon_{2j+1}, \|\cdot\|) \\
&\qquad \times \sup_{\substack{(A_1,\ldots,A_j) \\ \forall j\le j,\ A_i\in\mathcal{B}_i}} \mathcal{N}\left((**), \varepsilon_{2j+1}, \|\cdot\|_{2j+1}\right)
\end{aligned} \tag{44}$$

where

$$(**) = \left\{A_{j+1}F_{2j+1}(X) : A_{j+1} \in \mathcal{B}_{j+1}\right\}.$$

Thus, $\mathcal{N}(\mathcal{W}_{2j+1}, \varepsilon_{2j+1}, \|\cdot\|) \cdot \mathcal{N}(\mathcal{W}_{2j+2}, \varepsilon_{2j+2}, \|\cdot\|)$ is an upper bound for the ε_{2j+2}-covering number of the hypotheses space $\mathcal{H}_{i+1}$.

3) The vertex after the jth weight matrix is $N(2j-1)$. Suppose $\mathcal{W}_{2j-1}$ is an ε_{2j-1}-cover of the hypothesis space $\mathcal{H}_{2j-1}$ induced by the output hypotheses in the vertex $N(2j-1)$. Suppose there is a nonlinearity σ_j that directly follows the vertex $N(2j-1)$. We then analyze the contribution of the nonlinearity σ_j. Assume that the nonlinearity σ_j is the ρ_j-Lipschitz continuous. Apparently, $\sigma_j(\mathcal{W}_{2j-1})$ is a $\rho\varepsilon_{2j-1}$-cover of the hypothesis space $\sigma_j(\mathcal{H}_{2j-1})$. Specifically, for any $F' \in \sigma(\mathcal{H}_{2j-1})$, there exists a $F \in \mathcal{H}_{2j-1}$ that $F' = \sigma_j(F)$. Since $\mathcal{W}_{2j-1}$ is an ε_{2j-1}-cover of the hypothesis space $\mathcal{H}_{2j-1}$, there exists a $W \in \mathcal{W}_{2j-1}$ such that

$$\|F - W_{2j-1}\| \le \varepsilon_{2j-1}. \tag{45}$$

Therefore, we have the following equation:

$$\begin{aligned}
\|F' - \sigma_j(W_{2j-1})\| &= \|\sigma_j(F) - \sigma_j(W_{2j-1})\| \\
&\le \rho_j\|F - W_{2j-1}\| = \rho_j\varepsilon_{2j-1}.
\end{aligned} \tag{46}$$

We, thus, prove that $\mathcal{W}_{2j} \triangleq \sigma_j(\mathcal{W}_{2j-1})$ is a $\rho_j\varepsilon_{2j-1}$-cover of the hypothesis space $\sigma_j(\mathcal{H}_{2j-1})$. In addition, the covering number remains the same while applying a nonlinearity to the neural network.

By analyzing the influence of weight matrices and nonlinearities one by one, we can get (20). As for ε, the above part indeed gives a constructive method to obtain ε from all ε_i and $\varepsilon_{u,v,j}$. Here, we omit the explicit formulation of ε in terms of ε_i and $\varepsilon_{u,v,j}$ since it could not benefit our theory.

C. Covering Bound for the Hypothesis Space of Deep Neural Networks With Residual Connections

In Section V-A, we give a covering bound generally for all deep neural networks with residual connections. The result is summarized as Theorem 1. In this section, we give a detailed proof of Theorem 1.

Proof of Theorem 1: To approach the covering bound for the deep neural networks with residuals, we first analyze the influence of adding a vine to a deep neural network and then use an induction method to obtain a covering bound for the whole network.

All vines are connected with the stem at two points that are, respectively, after a nonlinearity and before a weight matrix. When the input $F_u(X)$ of the vine $V(u, v, i)$ is fixed, suppose all the hypothesis functions $F_V^{u,v,i}(X)$ computed by the vine $V(u, v, i)$ constitute a hypothesis space $\mathcal{H}_V^{u,v,i}$. As a vine is also a chain-like neural network constructed by stacking a series of weight matrices and nonlinearities, we can straightly apply Lemma 3 to approach an upper bound for the covering number of the hypothesis space $\mathcal{H}_V^{u,v,i}$. It is worth noting that vines could be identity mappings. This situation is normal in ResNet—there are 13 out of all the 16 vines are identities. For the circumstances that the vines are identities, the hypothesis space computed by the vine only contains one element—an identity mapping. The covering number of the hypothesis space for the identities are apparently 1.

Applying Lemmas 5 and 3, there exists an ε_v-cover $\mathcal{W}_v$ for the hypothesis space $\mathcal{H}_v$ with a covering number $\mathcal{N}(\mathcal{H}_v, \varepsilon_i, \|\cdot\|)$, as well as an $\varepsilon_V^{u,v,i}$-cover $\mathcal{W}_V^{u,v,i}$ for the hypothesis space $\mathcal{H}_V^{u,v,i}$ with a covering number $\mathcal{N}(\mathcal{H}_V^{u,v,i}, \varepsilon_i, \|\cdot\|)$.

The hypotheses computed by the vine $V(u,v,i)$ and the deep neural network without $V(u,v,i)$, i.e., respectively, $F_v(X)$ and $F_V^{u,v,i}$, are added elementwisely at the vertex $V(v)$. We denote the space constituted by all $F' \triangleq F_v(X) + F_V^{u,v,i}(X)$ as $\mathcal{H}_v'$.

Let us define a function space as $\mathcal{W}_v' \triangleq \{W_S + W_V : W_S \in \mathcal{W}_v, W_V \in \mathcal{W}_V^{u,v,i}\}$. For any hypothesis $F' \in \mathcal{H}_v'$, there must exist an $F_S \in \mathcal{H}_v$ and $F_V \in \mathcal{H}_V^{u,v,i}$ such that

$$F'(X) = F_S(X) + F_V(X) \tag{47}$$

because $\mathcal{W}_v$ is an ε_v-cover of the hypothesis space $\mathcal{H}_v$. For any hypothesis $F_S \in \mathcal{H}_v$, there exists an element $W_{F_S}(X) \in \mathcal{W}_v$ such that

$$\|F_S(X) - W_{F_S}(X)\| \le \varepsilon_v. \tag{48}$$

Similarly, as $\mathcal{W}_V^{u,v,i}$ is an $\varepsilon_V^{u,v,i}$-cover of $\mathcal{H}_V^{u,v,i}$, we can get a similar result. For any hypothesis $F_V(X) \in \mathcal{H}_V^{u,v,i}$, there exists an element $W_{F_V}(X) \in \mathcal{W}_V^{u,v,i}$ such that

$$\|F_V(X) - W_{F_V}(X)\| \le \varepsilon_V^{u,v,i}. \tag{49}$$

Therefore, for any hypothesis $F'(X) \in \mathcal{H}_v'$, there exists an element $W(X) \in \mathcal{W}'$ such that $W(X) = W_{F_S}(X) + W_{F_V}(X)$ satisfying (48) and (49), and furthermore

$$\begin{aligned} &\|F'(X) - W(X)\| \\ &\quad = \|F_V(X) + F_S(X) - W_{F_V}(X) - W_{F_S}(X)\| \\ &\quad = \|(F_V(X) - W_{F_V}(X)) + (F_S(X) - W_{F_S}(X))\| \\ &\quad \le \|F_V(X) - W_{F_V}(X)\| + \|F_S(X) - W_{F_S}(X)\| \\ &\quad \le \varepsilon_V^{u,v,i} + \varepsilon_v. \end{aligned} \tag{50}$$

Therefore, the function space $\mathcal{W}_v'$ is an $(\varepsilon_V^{u,v,i} + \varepsilon_v)$-cover of the hypothesis space $\mathcal{H}_v'$. An upper bound for the cardinality of the function space $\mathcal{W}_v'$ is given as below (it is also an $\varepsilon_V^{u,v,i} + \varepsilon_v$-covering number of the hypothesis space $\mathcal{H}_v'$)

$$\begin{aligned} &\mathcal{N}(\mathcal{H}_v', \varepsilon_V^{u,v,i} + \varepsilon_v, \|\cdot\|) \\ &\quad \le \left|\mathcal{W}_v'\right| \le |\mathcal{W}_v| \cdot \left|\mathcal{W}_V^{u,v,i}\right| \\ &\quad \le \sup_{F_{v-2}} \mathcal{N}(\mathcal{H}_v, \varepsilon_i, \|\cdot\|) \cdot \sup_{F_u} \mathcal{N}\big(\mathcal{H}_V^{u,v,i}, \varepsilon_V^{u,v,i}, \|\cdot\|\big) \\ &\quad \le \sup_{F_{v-2}} \mathcal{N}_v \cdot \sup_{F_u} \mathcal{N}_V^{u,v,i} \end{aligned} \tag{51}$$

where $\mathcal{N}_v$ and $\mathcal{N}_V^{u,v,i}$ can be obtained from (20) in Lemma 3, as the stem and all the vines are chain-like neural networks.

By adding vines to the stem one by one, we can construct the whole deep neural network. Combining Lemma 3 for the covering number of $F_{v-1}(X)$ and $F_u(X)$, we further get the following inequality:

$$\mathcal{N}(\mathcal{H}, \varepsilon, \|\cdot\|) \le \prod_{j=1}^{L} \sup_{F_{M(j)}} \mathcal{N}_{M(j+1)} \prod_{(u,v,i)\in I_V} \sup_{F_u} \mathcal{N}_V^{u,v,i}. \tag{52}$$

Thus, we prove (18) of Theorem 1.

As for ε, the above part indeed gives a constructive method to obtain ε from all ε_i and $\varepsilon_{u,v,j}$. Here, we omit the explicit formulation of ε in terms of ε_i and $\varepsilon_{u,v,j}$ since it could be extremely complex and does not benefit our theory.

D. Covering Bound for the Hypothesis Space of ResNet

In Section V-B, we give a covering bound for ResNet. The result is summarized as Lemma 4. In this section, we give a detailed proof of Lemma 4.

Proof of Lemma 4: There are 34 weight matrices and 35 nonlinearities in the stem of the 34-ResNet. Let us denote the weight matrices, respectively, as $A_1, \ldots, A_{34}$ and denote the nonlinearities, respectively, as $\sigma_1, \ldots, \sigma_{35}$. Apparently, there are $34 + 35 + 1 = 70$ vertexes in the network, where 34 is the number of weight matrices and 35 is the number of nonlinearities. We denote them, respectively, as $N(1), \ldots, N(70)$. In addition, there are 16 vines that are, respectively, denoted as $V(4i-1, 4i+3, 1)$, $i = \{1, \ldots, 16\}$, where $4i-1$ and $4i+3$ are the indexes of the vertexes that the vine connected. Among all the 16 vines, there are 3, $V(15, 19, 1)$, $V(31, 35, 1)$, and $V(55, 59, 1)$, respectively, contain one weight matrix, while all others are identities mappings. For the vine $V(4i-1, 4i+3, 1)$, $i = 4, 8, 14$, we denote the weight matrix in the vine as $A_1^{4i-1,4i+3,1}$.

Applying Theorem 1, we straightly get the following inequality:

$$\begin{aligned} \log \mathcal{N}(\mathcal{H}, \varepsilon, \|\cdot\|) \le &\sum_{j=1}^{34} \sup_{F_{2j-1}(X)} \log \mathcal{N}_{2j+1} \\ &+ \sum_{(u,v,i)\in I_V} \sup_{F_u(X)} \log \mathcal{N}_V^{u,v,1} \end{aligned} \tag{53}$$

where $\mathcal{N}_{2j+1}$ is the covering number of the hypothesis space constituted by all outputs $F_{2j+1}(X)$ at the vertex $N(2j+1)$ when the input $F_{2j-1}(X)$ of the vertex $N(2j-1)$ is fixed, $\mathcal{N}_V^{u,v,1}$ is the covering number of the hypothesis space constituted by all outputs $F_V^{u,v,i}(X)$ of the vine $V(u,v,1)$ when the input $F_v(X)$ is fixed, and I_V is the index set $\{(4i-1, 4i+3, 1), i = 1, \ldots, 16\}$.

Applying Lemma 5, we can further get an upper bound for the ε_{2j+1}-covering number $\mathcal{N}_{2j+1}$. The bound is expressed as the following inequality:

$$\log \mathcal{N}_{2j+1} \le \frac{b_{j+1}^2 \|F_{2j+1}(X^T)^T\|_2^2}{\varepsilon_{2j+1}^2} \log(2W^2) \tag{54}$$

where W is the maximum dimension among all features through the ResNet, i.e., $W = \max_i n_i$, $i = 0, 1, \ldots, L$. Also, we can decompose $\|F_{2j+1}(X^T)^T\|_2^2$ and utilize an induction method to obtain an upper bound for it.

1) If there is no vine connected with the stem at the vertex $N(2j-1)$, we have the following inequality:

$$\begin{aligned} &\|F_{2j+1}(X^T)^T\|_2 \\ &\quad = \|\sigma_j(A_j F_{2j-1}(X^T))^T\|_2 \\ &\quad = \|\sigma_j(A_j F_{2j-1}(X^T))^T - \sigma_j(0)\|_2 \\ &\quad \le \rho_j \|A_j F_{2j-1}(X^T)^T - 0\|_2 \end{aligned}$$

$$
\begin{aligned}
&= \rho_j \|A_j F_{2j-1}(X^T)^T\|_2 \\
&\leq \rho_j \|A_j\|_\sigma \cdot \|F_{2j-1}(X^T)^T\|_2. \qquad (55)
\end{aligned}
$$

2) If there is a vine $V(2j-3, 2j+1, 1)$ connected at the vertex $N(2j+1)$, then we get the following inequality:

$$
\begin{aligned}
&\|F_{2j+1}(X^T)^T\|_2 \\
&\quad = \left\|\sigma_j(A_j\sigma_j(A_jF_{2j-3}(X^T)))^T + A_1^{2j-3,2j+1,1}F_{2j-3}(X^T)^T\right\|_2 \\
&\quad \leq \|\sigma_j(A_j\sigma_j(A_jF_{2j-3}(X^T)))^T\|_2 + \left\|A_1^{2j-3,2j+1,1}F_{2j-3}(X^T)^T\right\|_2 \\
&\quad \leq \rho_j\|A_j\|_\sigma\rho_{j-1}\|A_{j-1}\|_\sigma \cdot \|F_{2j-3}(X^T)^T\|_2 + \left\|A_1^{2j-3,2j+1,1}\right\|_\sigma \cdot \left\|F_{2j-3}(X^T)^T\right\|_2 \\
&\quad = \left(\rho_j\rho_{j-1}\|A_j\|_\sigma \cdot \|A_{j-1}\|_\sigma + \left\|A_1^{2j-3,2j+1,1}\right\|_\sigma\right) \\
&\qquad \times \|F_{2j-3}(X^T)^T\|_2. \qquad (56)
\end{aligned}
$$

Therefore, based on (55) and (56), we can get the norm of output of ResNet as in the main text.

Similar to $\mathcal{N}_{2j+1}$, we can obtain an upper bound for the $\varepsilon_{u,v,1}$-covering number $\mathcal{N}_V^{u,v,1}$. Suppose the output computed at the vertex $N(u)$ is $F_u(X^T)$. Then, we can get the following inequality:

$$
\log \mathcal{N}_V^{u,v,1} \leq \frac{\left(b_1^{u,v,1}\right)^2 \|F_u(X^T)^T\|_2^2}{\varepsilon_{u,v,1}^2} \log(2W^2). \qquad (57)
$$

Applying (54) and (57) to (53), we, thus, prove (23).

As for the formulation of the radiuses of the covers, we also employ an induction method.

1) Suppose the radius of the cover for the hypothesis space computed by the weight matrix A_1 and the nonlinearity σ_1 is ε_3. Then, applying (43) and (46), after the weight matrix A_2 and the nonlinearity σ_2, we get the following equation:

$$
\varepsilon_3 = (s_2+1)\rho_2\varepsilon_1. \qquad (58)
$$

2) Suppose the radius of the cover for the hypothesis space computed by the weight matrix A_{j-1} and the nonlinearity σ_{j-1} is ε_{2j-1}. Assume there is no vine connected around. Then, similarly, after the weight matrix A_2 and the nonlinearity σ_j, we get the following equation:

$$
\varepsilon_{2j+1} = \rho_j(s_j+1)\varepsilon_{2j-1}. \qquad (59)
$$

3) Suppose the radius of the cover at the vertex $N(i)$ is ε_i. Assume there is a vine $V(u, u+4, 1)$ that links the stem at the vertex $N(u)$ and $N(u+4)$. Then, similarly, after the weight matrix A_2 and the nonlinearity σ_j, we get the following equation:

$$
\begin{aligned}
\varepsilon_{2j+1} &= \varepsilon_{u+2}(s_{\frac{u-1}{2}}+1)\rho_{\frac{u-1}{2}} + \varepsilon_u(s_{u,u+4,1}+1) \\
&= \varepsilon_u(s_{\frac{u-1}{2}}+1)\rho_{\frac{u-1}{2}}(s_{\frac{u-3}{2}}+1)\rho_{\frac{u-3}{2}} \\
&\quad + \varepsilon_u(s_{u,u+4,1}+1) \\
&= \varepsilon_u(s_{\frac{u-1}{2}}+1)(s_{\frac{u-3}{2}}+1)\rho_{\frac{u-1}{2}}\rho_{\frac{u-3}{2}} \\
&\quad + \varepsilon_u(s_{u,u+4,1}+1). \qquad (60)
\end{aligned}
$$

From (58)–(60), we can obtain the following equation

$$
\begin{aligned}
\varepsilon = \varepsilon_1\rho_1(s_1+1)\rho_{34}(s_{34}+1)\rho_{35} &\prod_{\substack{1\leq i\leq 16\\ i\notin\{4,8,14\}}} [(***)+1] \\
\times &\prod_{i\in\{4,8,14\}} \left[(***)+s_1^{4i-1,4i+3,1}+1\right] \qquad (61)
\end{aligned}
$$

where

$$
(***) = \rho_{2i}(s_{2i}+1)\rho_{2i+1}(s_{2i+1}+1). \qquad (62)
$$

Combining the definition of $\bar{\alpha}$

$$
\begin{aligned}
\bar{\alpha} = \rho_1(s_1+1)\rho_{34}(s_{34}+1)\rho_{35} &\prod_{\substack{\leq i\leq 16\\ i\notin\{4,8,14\}}} [(***)+1] \\
\times &\prod_{i\in\{4,8,14\}} \left[(***)+s_1^{4i-1,4i+3,1}+1\right] \qquad (63)
\end{aligned}
$$

we can obtain that

$$
\varepsilon_1 = \frac{\varepsilon}{\bar{\alpha}}. \qquad (64)
$$

Applying (58)–(60), we can get all ε_{2j+1} and $\varepsilon^{u,u+4,1}$.

The proof is completed. □

E. Generalization Bound for ResNet

Proof of Theorem 2: We prove this theorem in two steps: 1) we first apply Lemma 2 to Lemma 4 in order to get an upper bound on the Rademacher complexity of the hypothesis space computed by ResNet and 2) we then apply the result of 1 to Lemma 1 in order to get a generalization bound.

1) Upper Bound on the Rademacher Complexity: Applying (8) of Lemma 2 to (34) of Lemma 4, we can get the following inequality:

$$
\begin{aligned}
&\mathfrak{R}(\mathcal{H}_\lambda|_D) \\
&\quad \leq \inf_{\alpha>0}\left(\frac{4\alpha}{\sqrt{n}} + \frac{12}{n}\int_\alpha^{\sqrt{n}} \sqrt{\log\mathcal{N}(\mathcal{H}_\lambda|_D, \varepsilon, \|\cdot|_2)}\mathrm{d}\varepsilon\right) \\
&\quad \leq \inf_{\alpha>0}\left(\frac{4\alpha}{\sqrt{n}} + \frac{12}{n}\int_\alpha^{\sqrt{n}} \frac{\sqrt{R}}{\varepsilon}\mathrm{d}\varepsilon\right) \\
&\quad \leq \inf_{\alpha>0}\left(\frac{4\alpha}{\sqrt{n}} + \frac{12}{n}\sqrt{R}\log\frac{\sqrt{n}}{\alpha}\right). \qquad (65)
\end{aligned}
$$

Apparently, the infinimum is reached uniquely at $\alpha = 3\sqrt{R/n}$. Here, we use a simpler and also widely used choice $\alpha = 1/n$ and get the following inequality:

$$
\mathfrak{R}(\mathcal{H}_\lambda|_D) \leq \frac{4}{n^{\frac{3}{2}}} + \frac{18}{n}\sqrt{R}\log n. \qquad (66)
$$

2) Upper Bound on the Generalization Error: Combining with (7) of Lemma 1, we get the following inequality:

$$
\begin{aligned}
&\Pr\{\arg\max_i F(x)_i \neq y\} \\
&\quad \leq \hat{\mathcal{R}}_\lambda(F) + \frac{8}{n^{\frac{3}{2}}} + \frac{36}{n}\sqrt{R}\log n + 3\sqrt{\frac{\log(1/\delta)}{2n}}. \qquad (67)
\end{aligned}
$$

The proof is completed. □

VII. Conclusion

We provide an upper bound for the covering number of the hypothesis space induced by deep neural networks with residual connections. The covering bound for ResNet, as an exemplary case, is then proposed. Combining various classic results in statistical learning theory, we further obtain a generalization bound for ResNet. With the generalization bound, we theoretically guarantee the performance of ResNet on unseen data. Considering the generality of our results, the generalization bound for ResNet can be easily extended to many state-of-the-art algorithms, such as DenseNet and ResNeXt.

This article is based on the complexity of the whole hypothesis space. Some recent experimental results give an insight that SGD only explores a part of the hypothesis space and never visits other places. Thus, involving localization properties into the analysis could lead to a tighter upper bound of the generalization error. However, there still lacks concrete evidence to support the localization property, and the exact mechanism still remains an open problem. We plan to explore this problem in future work.

Acknowledgment

The authors would like to thank the constructive feedback and helpful suggestions from the anonymous reviewers and editors.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[3] B. Shi, X. Bai, W. Liu, and J. Wang, "Face alignment with deep regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 183–194, Jan. 2018.

[4] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[5] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017.

[6] D. Chang, M. Lin, and C. Zhang, "On the generalization ability of online gradient descent algorithm under the quadratic growth condition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 5008–5019, Oct. 2018.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[8] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, vol. 1, no. 2, pp. 4700–4708.

[9] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5987–5995.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[11] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. USENIX Symp. Oper. Syst. Design Implement.*, vol. 16, 2016, pp. 265–283.

[12] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, vol. 1, no. 2, p. 4.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988.

[14] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[15] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2016.

[16] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," 2017, *arXiv:1704.05519*. [Online]. Available: https://arxiv.org/abs/1704.05519

[17] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, Feb. 2017.

[18] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Proc. Adv. Neural Inf. Process. Syst.*, 1992, pp. 950–957.

[19] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6240–6249.

[20] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[21] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, "Generalization in deep learning," 2017, *arXiv:1710.05468*. [Online]. Available: https://arxiv.org/abs/1710.05468

[22] N. Harvey, C. Liaw, and A. Mehrabian, "Nearly-tight VC-dimension bounds for piecewise linear neural networks," in *Proc. Annu. Conf. Learn. Theory*, 2017, pp. 1064–1068.

[23] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2012.

[24] N. Golowich, A. Rakhlin, and O. Shamir, "Size-independent sample complexity of neural networks," in *Proc. Annu. Conf. Learn. Theory*, 2018, pp. 297–299.

[25] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5947–5956.

[26] T. Liang, T. Poggio, A. Rakhlin, and J. Stokes, "Fisher-Rao metric, geometry, and complexity of neural networks," 2017, *arXiv:1711.01530*. [Online]. Available: https://arxiv.org/abs/1711.01530

[27] R. Novak, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein, "Sensitivity and generalization in neural networks: An empirical study," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[28] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[29] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6391–6401.

[30] O. Shamir, "Are resnets provably better than linear predictors?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 507–516.

[31] C. Yun, S. Sra, and A. Jadbabaie, "Are deep resnets provably better than linear predictors?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15660–15669.

[32] K. Kawaguchi and Y. Bengio, "Depth with nonlinearity creates no bad local minima in ResNets," 2018, *arXiv:1810.09038*. [Online]. Available: https://arxiv.org/abs/1810.09038

[33] H. Mhaskar, Q. Liao, and T. A. Poggio, "When and why are deep networks better than shallow ones?" in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2343–2349.

[34] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[35] A. Achille and S. Soatto, "Emergence of invariance and disentanglement in deep representations," *J. Mach. Learn. Res.*, vol. 19, no. 1, pp. 1947–1980, 2018.

[36] 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. 1141–1150.

[37] Z. Tu, F. He, and D. Tao, "Understanding generalization in recurrent neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[38] M. Soltanolkotabi, A. Javanmard, and J. D. Lee, "Theoretical insights into the optimization landscape of over-parameterized shallow neural networks," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 742–769, Feb. 2019.

[39] V. N. Vapnik and A. J. Chervonenkis, *Theory of Pattern Recognition*. Moscow, Russia: Nauka, 1974.

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

[41] R. M. Dudley, "The sizes of compact subsets of Hilbert space and continuity of Gaussian processes," in *Selected Works of RM Dudley*. New York, NY, USA: Springer, 2010, pp. 125–165.

[42] D. Haussler, "Sphere packing numbers for subsets of the Boolean n-cube with bounded Vapnik-Chervonenkis dimension," *J. Combinat. Theory A*, vol. 69, no. 2, pp. 217–232, Feb. 1995.

[43] P. L. Bartlett, O. Bousquet, and S. Mendelson, "Local Rademacher complexities," *Ann. Statist.*, vol. 33, no. 4, pp. 1497–1537, Aug. 2005.

[44] O. Bousquet and A. Elisseeff, "Algorithmic stability and generalization performance," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 196–202.

[45] T. Liu, G. Lugosi, G. Neu, and D. Tao, "Algorithmic stability and hypothesis complexity," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2159–2167.

[46] Y. Li, X. Tian, T. Liu, and D. Tao, "On better exploring and exploiting task relationships in multitask learning: Joint model and feature learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1975–1985, May 2018.

[47] Y. Han, Y. Yang, X. Li, Q. Liu, and Y. Ma, "Matrix-regularized multiple kernel learning via (*r*, *p*) norms," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4997–5007, Oct. 2018.

[48] Q. Meng, Y. Wang, W. Chen, T. Wang, Z. Ma, and T.-Y. Liu, "Generalization error bounds for optimization algorithms via stability," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2336–2342.

[49] X. Tian, Y. Li, T. Liu, X. Wang, and D. Tao, "Eigenfunction-based multitask learning in a reproducing kernel Hilbert space," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1818–1830, Jun. 2019.

[50] T. Zhang, "Covering number bounds of certain regularized linear function classes," *J. Mach. Learn. Res.*, vol. 2, pp. 527–550, Mar. 2002.

[51] T. Zhang, "Statistical analysis of some multi-category large margin classification methods," *J. Mach. Learn. Res.*, vol. 5, pp. 1225–1251, Oct. 2004.

[52] R. M. Dudley, "Universal Donsker classes and metric entropy," in *Selected Works of RM Dudley*. New York, NY, USA: Springer, 2010, pp. 345–365.

[53] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009, 2009.

[54] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[55] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, "Sharp minima can generalize for deep nets," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1019–1028.

[56] A. Galloway, T. Tanay, and G. W. Taylor, "Adversarial training versus weight decay," 2018, *arXiv:1804.03308*. [Online]. Available: https://arxiv.org/abs/1804.03308

[57] G. Zhang, C. Wang, B. Xu, and R. Grosse, "Three mechanisms of weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[58] J. Chen and Q. Gu, "Closing the generalization gap of adaptive gradient methods in training deep neural networks," 2018, *arXiv:1806.06763*. [Online]. Available: https://arxiv.org/abs/1806.06763

[59] J.-G. Park and S. Jo, "Bayesian weight decay on bounded approximation for deep convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2866–2875, Sep. 2019.

[60] G. Pisier, "Remarques sur un résultat non publié de B. Maurey," *Séminaire Analyse Fonctionnelle*, no. 5, pp. 1–12, 1981.

Fengxiang He (Student Member, IEEE) received the B.Sc. degree in statistics from University of Science and Technology of China, Hefei, China, in 2017, and the M.Phil. degree from the University of Sydney, Darlington, NSW, Australia, in 2019, where he is currently pursuing the Ph.D. degree.

His research interests are machine learning theory and the applications of machine learning in computer vision. He has published six papers in four prominent conferences, including ICLR, NeurIPS, ICCV, and CVPR.

Tongliang Liu (Member, IEEE) is a Lecturer with the School of Computer Science at the University of Sydney. His research interests include machine learning and computer vision. He has authored and coauthored more than 60 research articles including the IEEE T-PAMI, T-NNLS, T-IP, ICML, NeurIPS, CVPR, ECCV, AAAI, IJCAI, KDD, and ICME, with best paper awards, e.g., the 2019 ICME Best Paper Award. He is a recipient of Discovery Early Career Researcher Award (DECRA) from Australian Research Council (ARC) and was shortlisted for the J. G. Russell Award by Australian Academy of Science (AAS) in 2019.

Dacheng Tao (Fellow, IEEE) is Professor of Computer Science and ARC Laureate Fellow in the School of Computer Science, Faculty of Engineering, University of Sydney. His research results in artificial intelligence have expounded in one monograph and more than 200 publications at prestigious journals and prominent conferences, such as the IEEE T-PAMI, IJCV, JMLR, AAAI, IJCAI, NIPS, ICML, CVPR, ICCV, ECCV, ICDM, and KDD, with several best paper awards. He received the 2018 IEEE ICDM Research Contributions Award and the 2015 Australian Scopus-Eureka prize. He is a Fellow of the IEEE, ACM, and Australian Academy of Science.