Depth Selection for Deep ReLU Nets in Feature Extraction and Generalization

Zhi Han, Siquan Yu, Shao-Bo Lin, and Ding-Xuan Zhou

Abstract—Deep learning is recognized to be capable of discovering deep features for representation learning and pattern recognition without requiring elegant feature engineering techniques by taking advantages of human ingenuity and prior knowledge. Thus it has triggered enormous research activities in machine learning and pattern recognition. One of the most important challenges of deep learning is to figure out relations between a feature and the depth of deep neural networks (deep nets for short) to reflect the necessity of depth. Our purpose is to quantify this feature-depth correspondence in feature extraction and generalization. We present the adaptivity of features to depths and vice-verse via showing a depth-parameter trade-off in extracting both single feature and composite features. Based on these results, we prove that implementing the classical empirical risk minimization on deep nets can achieve the optimal generalization performance for numerous learning tasks. Our theoretical results are verified by a series of numerical experiments including toy simulations and a real application of earthquake seismic intensity prediction.

Index Terms—Deep nets, feature extractions, generalization, learning theory

1 Introduction

A systemic machine learning process frequently comes down to two steps: feature extraction and target-driven learning. The former focuses on designing preprocessing pipelines and data transformations that result in a tractable representation of data, while the latter utilizes learning algorithms related to specific targets, such as regression, classification and clustering on the data representation to finish the learning task. Studies in the second step abound in machine learning [5] and numerous learning schemes such as kernel methods [15], neural networks [20] and boosting [21] have been proposed. However, feature extraction in the first step is usually labor intensive, which requires elegant feature engineering techniques by taking advantages of human ingenuity and prior knowledge.

To extend the applicability of machine learning, it is crucial to make learning algorithms be less dependent of human factors. Deep learning [23], [16], which has been successfully used in image classification, natural language processing and game theory, provides a promising technique in machine learning. The heart of deep learning is to adopt deep neural networks (deep nets for short) with certain structures to extract data features and design target-driven algorithms, simultaneously. As shown in Figure 1, deep learning embodies the utilities of feature extraction algorithms such as bag of feature (BOF), local binary pattern (LBP), histogram of oriented gradient (HOG) and classification algorithms like support vector machine (SVM), random forest, via tuning parameters in a unified deep nets model.

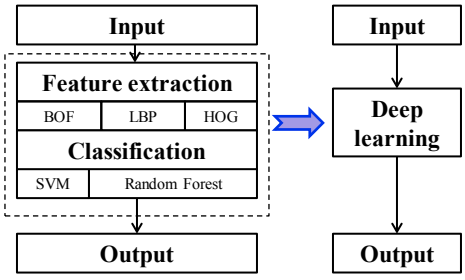


Fig. 1: Magic behind deep learning

The great success of deep learning in applications demonstrates the feasibility of deep nets in specific learning tasks. However, whether deep learning is generalizable to other learning tasks relies on rigorous theoretical verifications, which is unfortunately at its infancy. In particular, it is highly desired to clarify the following three important problems: 1) which data features[1] can be extracted by deep nets; 2) how to set the depth of deep nets in special learning tasks; 3) how about the generalization ability of deep learning algorithms. The first problem refers to the representation performance of deep nets, needing tools from information theory like coding theory [38] and entropy theory [18]. The second one concerns approximation abilities of deep nets with different depth, requiring approximation theory

- *Z. Han is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China and Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang, China.*
- *S. Yu is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China and Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang, China and also with the School of Information Science and Engineering, Northeastern University, Shenyang, China.*
- *S.B. Lin is with the Center of Intelligent Decision-Making and Machine Learning, School of Management, Xi'an Jiaotong University, Xi'an, China.*
- *D.X. Zhou is with the School of Data Science, Liu Bie Ju Centre for Mathematical Sciences and Department of Mathematics, City University of Hong Kong, Hong Kong, China.*
- *Corresponding author: S. B. Lin (sblin1983@gmail.com)*

1. Data feature in this paper means priors for presentation learning according to the terminology in the nice review paper [3]. It includes both the a-priori information of target functions and structures of the input space.

techniques such as local polynomial approximations [48], covering number estimates [25] and wavelets analysis [52] to quantify powers and limitations of deep nets. The last one focuses on the generalization capability of deep learning algorithms in machine learning, for which statistical learning theory as well as empirical processing [10] should be utilized.

Although lagging heavily behind applications, recent developments of deep learning theory provided some exciting theoretical results on these problems. For example, [44] proved that deep nets succeed in extracting some geometric structures of data, which has been adopted in [8] to design deep learning algorithm for regression problems with data generated on manifolds; [7] found that deep nets can extract local position information of data, which was recently employed in [31] to construct deep nets in handling sparsely located data; [38] proved that deep nets can extract piecewise features of data, which was utilized in [24] to develop learning algorithms to learn non-smooth functions efficiently. All these interesting studies presented theoretical verifications on the power of deep learning in the sense that deep nets succeed in extracting data features and deep learning significantly improves the generalization capabilities of learning schemes in-hand.

The problem is, however, that there are strongly exclusive correspondences between data features and network depth for these theoretical studies in the sense that each data feature requires a unique network depth and vice-verse. To be detailed, a hierarchal structure corresponds to a hierarchal deep net with the same depth [36]; smoothness a-priori information [48] is related to a deep net with accuracy-dependent depth; a translation-invariance property requires a convolutional neural network with accuracy-dependent layers [6]; and a rotation-invariance property is associated with a deep net with tree structures and four layers [9]. Such exclusive correspondences hinder heavily the use of deep nets in feature extraction, since data features such as the smoothness information, hierarchal structure, transformation-invariance are practically difficult to be specified before the learning process. Furthermore, it is questionable to determine the network depth for simultaneously extracting multiple data features like the translation-invariance and rotation-invariance, which is pretty common in practice. Our first purpose is to break through the feature-depth correspondences by means of proving that deep nets with certain depth can extract several data features and vice-verse.

We consider extracting both single data features such as smoothness, rotation-invariance, sparseness and composite data features combining smoothness, rotation-invariance and sparseness to demonstrate the adaptivity of network depth to features and vice-verse. Intuitively, it is difficult for deep ReLU nets to extract smoothness features due to the non-smooth property of the ReLU function $\sigma(t) = \max\{t, 0\}$. A natural remedy for this, as shown in [48], is to deepen the network with an accuracy-dependent depth to eliminate the negative effect of non-smoothness. Since under some specified capacity measurements such as the number of linear regions [37], Betti numbers [4], number of monomials [12] and covering numbers [18], the capacity of deep nets increases exponentially with respect to the depth, large depth usually means large capacity costs for feature extraction. Furthermore, from an optimization viewpoint, large depth requires to solve a highly nonconvex optimization problem [16, Sec. 8.2] involving the ill-conditioning of the Hessian, the existence of many local minima, saddle points, plateau and even some flat regions, making it difficult to design optimization algorithms for such deep nets with convergence guarantees. Based on these, we provide a theoretical guidance for depth selection to extract data features by showing that deep nets with various depths, larger than a specified value, are capable of extracting the smoothness and other data features. This shows an adaptivity of the depth to data features in the sense that any data features from a rich family can be extracted by deep nets with various depths. Conversely, we also provide theoretical guarantees on the adaptivity of the data feature to depths by showing that deep ReLU nets with some specific depth succeed in extracting the smoothness, spareness and composite features. All these remove the feature-depth correspondences in feature extraction for deep ReLU nets.

From feature extraction to machine learning, the tug of war between bias and variance [10] indicates that the prominent performance of deep nets in feature extraction is insufficient to demonstrate its success. The good generalization ability is frequently built upon the balance between the accuracy of feature extraction and capacity costs to achieve such an accuracy. This exhibits a bias-variance dilemma in selecting the capacity of deep nets. Different from shallow learning such as kernel methods and boosting, recent studies [22], [18] presented a depth-parameter dilemma in controlling the capacity of deep nets in the sense that different depth-parameter pairs may yield the same capacity. These two dilemmas as well as the optimization difficulty [16, Sec. 8.2] pose an urgent issue for deep learning theory on selecting the depth to guarantee the good generalization ability of deep learning algorithms. Our second purpose is not only to pursue the optimal generalization error for learning schemes based on deep nets, but also to demonstrate the depth selection strategy to realize this optimality.

We study the generalization ability of deep nets with different depths via empirical risk minimization (ERM). Based on the established adaptivity of the depth to data features in feature extraction, we establish almost optimal generalization error bounds for deep nets with numerous depth-parameter pairs. Our results show that the feature extraction step is necessary when the learning task is somewhat sophisticated and deep nets succeed in extracting deep data features of the data distribution, which illustrates the necessity of depth in deep learning. However, we also prove that the depth for realizing the optimal learning performance of deep nets is not unique. In fact, with depth larger than some specified value, all deep nets theoretically perform similarly and can achieve the optimal generalization error bounds. The only difference is that deeper nets involve less free parameters.

In a nutshell, our analysis implies three interesting findings in understanding the success of deep learning. The first is the flexibility on automatically selecting the accuracy in extracting data features via tuning the network parameters, which is different from the classical two-step

learning scheme presented in Figure 1 that usually involves extremely high capacity costs to fully extract data features. The second is the versatility of deep nets with fixed depth in the sense that they can extract various data features. The third one is that if the depth is larger than a specified value, we can always get a deep net estimator with almost optimal theoretical guarantee. The problem is, however, that it is difficult to design efficient optimization algorithms to solve ERM on deep nets with large depth (see [16, Sec. 8.2] and [1] for example)[2]. Under this circumstance, it is numerically difficult to get a deep net estimator with large depth from the optimization viewpoint. As a result, there is practically an optimal depth to realize the established optimal generalization error bounds, just as our experimental results exhibit.

The rest of the paper is organized as follows. In the next section, we introduce deep nets and show some recent developments of deep nets in feature extraction. Section 3 focuses on the depth selection for deep ReLU nets in extracting single data features, while Section 4 devotes to the depth selection in extracting composite data features. In Section 5, we are interested in the generalization error analysis for implementing ERM on deep nets. Section 6 exhibits some numerical results to verify our theoretical assertions. In the last section, we draw a simple conclusion and present some further discussions.

2 Necessity of Depth in Feature Extraction

Let $d \in \mathbb{N}$ be the dimension of input space. Denote $x = (x^{(1)}, \ldots, x^{(d)}) \in \mathbb{I}^d := [-1, 1]^d$. Let $L \in \mathbb{N}$ and $d_0, d_1, \ldots, d_L \in \mathbb{N}$ with $d_0 = d$. For $\vec{h} = (h^{(1)}, \ldots, h^{(d_k)})^T \in \mathbb{R}^{d_k}$, define $\vec{\sigma}(\vec{h}) = (\sigma(h^{(1)}), \ldots, \sigma(h^{(d_k)}))^T$. Deep ReLU nets with depth L and width d_j in the j-th hidden layer can be mathematically represented as

$$h_{\{d_0,\ldots,d_L,\sigma\}}(x) = \vec{a} \cdot \vec{h}_L(x), \tag{1}$$

where

$$\vec{h}_k(x) = \vec{\sigma}(W_k \cdot \vec{h}_{k-1}(x) + \vec{b}_k), \qquad k = 1, 2, \ldots, L, \tag{2}$$

$\vec{h}_0(x) = x$, $\vec{a} \in \mathbb{R}^{d_L}$, $\vec{b}_k \in \mathbb{R}^{d_k}$, and $W_k = (W_k^{i,j})_{i=1,j=1}^{d_k,d_{k-1}}$ is a $d_k \times d_{k-1}$ matrix. Denote by $\mathcal{H}_{\{d_0,\ldots,d_L,\sigma\}}$ the set of all these deep nets. When $L = 1$, the function defined by (1) is the classical shallow net.

The structure of deep nets is reflected by structures of weight matrices W_k and threshold vectors $\vec{b}_k$ and $\vec{a}$, $k = 1, 2, \ldots, L$. Besides the deep fully connected networks [48] that counts the number of free parameters in the k-th layer to be $d_k d_{k-1} + d_k$[3], we say that there are n_k free parameters in the k-th layer, if the weight matrix W_k and thresholds $\vec{b}_k$ are generated through the following three ways. The first way is that there are totally n_k tunable entries in W_k and $\vec{b}_k$, while the remainder $d_k d_{k-1} + d_k - n_k$ entries are fixed. An example is deep sparsely connected neural networks. The second way is that W_k and $\vec{b}_k$ are exactly generated by n_k free parameters including weight-sharing. The third way is that the weight matrix is generated jointly by both the above ways. Like the most widely used deep convolutional neural networks, we count the number of free parameters according to the third way by considering both sparse connections and weight-sharing [51], [52], [53]. It should be mentioned that such a way to count free parameter is different from [48] which considers deep fully connected neural networks. The different way to count free parameters is consistent with the structure of deep nets, which is the main reason why we can improve the approximation result of [48].

2.1 Capacity measurements of deep nets

It is meaningless to pursue the outperformance of deep nets over shallow nets without considering the capacity costs, since the universality of shallow nets [11], [26] demonstrates that shallow nets can extract an arbitrary data feature as long as the network is sufficiently wide. We adopt the concept of covering number [50] which is widely used in statistical learning and information theory to measure the capacity to cast the comparison into a unified framework.

Let $\mathbb{B}$ be a Banach space and V be a subset of $\mathbb{B}$. Denote by $\mathcal{N}(\varepsilon, V, \mathbb{B})$ the ε-covering number of V under the metric of $\mathbb{B}$, which is the minimal number of elements in an ε-net of V. Intuitively, the ε-covering number measures the capacity of V via counting the minimal number of balls in $\mathbb{B}$ with radius ε covering V. Figure 2 shows that the 0.1-covering number of A is 19 while that of B is 10, coinciding with the intuitive observation that A is larger than B.

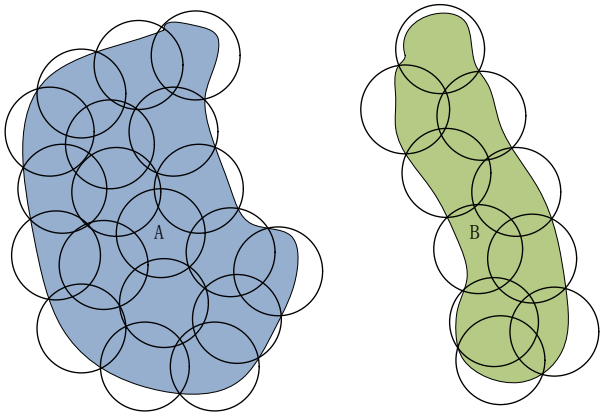


Fig. 2: Covering numbers of different sets

The quantity $H(\varepsilon, V, \mathbb{B}) = \log_2 \mathcal{N}(\varepsilon, V, \mathbb{B})$ is called the ε-entropy of V in $\mathbb{B}$ which is close to the coding length in information theory according to the encode-decode theory [13]. Thus, it is a powerful capacity measurement to show the expressivity of V in $\mathbb{B}$. Furthermore, the ε-covering number determines the limitation of approximation ability of V [18] and also the stability of learning algorithms defined on V [10]. All these demonstrate the rationality of adopting the covering number to measure the capacity of deep nets.

2. Here, the difficulty means that larger depth requires more free parameters under an over-parameterization setting to guarantee the convergence of SGD to a local minimization of ERM of high quality and larger depth results in more local minima, saddle points and flat regions.

3. If k=L, the number is $d_L d_{L-1} + 2d_L$ by taking the outer weights into accounts

Denote by $\mathcal{H}_{n,L}$ the set of all deep nets with L hidden layers, n free parameters and by

$$\mathcal{H}_{n,L,\mathcal{R}} := \{h_{n,L} \in \mathcal{H}_{n,L} : |w_k^{i,j}|, |b_k^i|, |a_i| \leq \mathcal{R}, \\ 1 \leq i \leq d_k, 1 \leq j \leq d_{k-1}, 1 \leq k \leq L\} \quad (3)$$

the set of deep nets whose weights and thresholds are uniformly bounded by $\mathcal{R}$, where $\mathcal{R}$ is some positive number that may depend on n, d_k, and L. The boundedness assumption is necessary since it can be found in [34], [18] that there exists some deep nets with two hidden layers and finitely many neurons possessing an infinite covering number.

The following lemma proved in [18] presents a tight estimate for the covering number of deep ReLU nets.

Lemma 1. *Let $\mathcal{H}_{n,L,\mathcal{R}}$ be defined by (3). Then*

$$\mathcal{N}\left(\varepsilon, \mathcal{H}_{n,L,\mathcal{R}}, L^\infty(\mathbb{I}^d)\right) \leq \left(C\mathcal{R}D_{\max}\right)^{3(L+1)^2 n} \varepsilon^{-n}, \quad (4)$$

where $D_{\max} := \max_{0\leq \ell \leq L} d_\ell$ and C is a constant depending only on d.

It was deduced in [19, Chap. 16] that

$$\log \mathcal{N}(\varepsilon, \mathcal{H}_{n,1,\mathcal{R}}, L^1(\mathbb{I}^d)) = \mathcal{O}\left(n \log \frac{\mathcal{R}}{\varepsilon}\right). \quad (5)$$

Comparing Lemma 1 with (5), we find that, up to a logarithmic factor, deep nets with controllable magnitudes of weights do not essentially enlarge the capacity of shallow nets, provided that they have the same number of free parameters and the depth of deep nets is at most $\log n$. Furthermore, Lemma 1 implies that the depth plays a similar role as the number of parameters in controlling the capacity of deep nets, when ε is not extremely small. This shows a novel depth-parameter dilemma in controlling the capacity.

2.2 Limitations of shallow nets in extracting features

The study of approximation capability of shallow nets is a classical topic in neural networks. We refer the readers to a fruitful review paper [40] for details on this topic. Compared with the classical linear approaches like polynomials, shallow nets with sigmoidal activation function possess better approximation ability [35] and are capable of conducting dimension-independent error estimates under certain restrictions on target functions [2]. More importantly, the universality [11], [26] showed that shallow nets can extract any data feature as long as the network is sufficiently wide. However, with fixed width, they have limitations in feature extraction, in terms of saturation [29], non-localization [7], [41], non-sparse approximation [27], [31] and bottleneck in extracting the smoothness feature [34], [30]. In particular, it was shown in [34] that shallow nets whose capacity satisfies (5) cannot extract the smoothness features within accuracy $\mathcal{O}(n^{-r/(d-1)})$ with high probability, where r denotes the degree of smoothness.

For shallow nets with ReLU (shallow ReLU nets), the limitation is even stricter. It was shown in [14] that there exist some analytic univariate functions which cannot be expressible for shallow ReLU nets. Recently, [48, Theorem 6] proved that any twice-differentiable nonlinear function defined on $\mathbb{I}^d$ cannot be ε-approximated by ReLU networks of fixed depth L with the number of free parameters less than $c\varepsilon^{-1/(2(L-2))}$, where c is a positive constant depending only on d. A direct consequence is that a ReLU network with depth $L = 3$ and n free parameters cannot extract the simple "square-feature", i.e., t^2, within accuracy $n^{-2-\tau}$ for an arbitrary $\tau > 0$. By noting t^2 is an infinitely differentiable function, it is well known that there exist linear tools to approximate t^2 within accuracy of order $n^{-\Gamma}$ [39] for an arbitrarily large $\Gamma < \infty$. All these results showed that shallow nets, especially shallow ReLU nets, are difficult to extract data features and thus have bottlenecks in complex learning tasks.

2.3 Necessity of the depth for ReLU nets

Advantages of deep nets over shallow nets were firstly revealed by [7] in the sense that deep nets can provide localized approximation but shallow nets fail. Since then, a great number of data features including those for sparseness, manifold structures, piecewise smoothness and rotation-invariance are proved to be unrealizable by shallow nets but can be easily extracted by deep nets. Under the capacity constraint (4) that is similar to (5) for shallow nets, the summary of advantages of deep ReLU nets in feature extraction are listed in the following Table 1.

TABLE 1: Deep nets in feature extraction (within accuracy ε, r-smooth function and d_m-dimensional manifold)

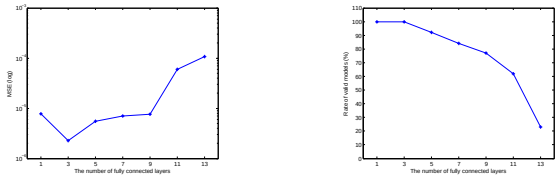
Ref.	Features	Parameters	Depth
[7]	Localized approximation	$2d+1$	2
[31]	k-spatially sparse	$k(2d+1)$	2
[44]	Smooth+Manifold	$\varepsilon^{-d_m/r}$	4
[38]	Piecewise smooth	$\varepsilon^{-d/r}$	Finite
[41]	ℓ_1 radial+smooth	$\varepsilon^{-1/r}$	$\log(\varepsilon^{-1})$
[43]	k-sparse (frequency)	$k\log(\varepsilon^{-1})$	$\log(\varepsilon^{-1})$

To extract the "square-feature", the following lemma has been shown in [48, Proposition 2] to verify that deep ReLU nets can overcome the bottleneck of shallow ReLU nets.

Lemma 2. *The function $f(t) = t^2$ on the segment $[0, 1]$ can be approximated with any error $\varepsilon > 0$ by a ReLU network having the depth and free parameters of order $\mathcal{O}(\log(1/\varepsilon))$.*

Due to the non-smoothness of ReLU, it is difficult for a shallow ReLU net with fixed width to extract smooth features within an arbitrary accuracy ε. However, by deepening the network, Lemma 2 shows that deep ReLU nets succeed in finishing such a task with only $\mathcal{O}(\log(1/\varepsilon))$ free parameters. With Lemma 2 and the relation $t_1 \cdot t_2 = [(t_1 + t_2)^2 - t_1^2 - t_2^2]/2$, deep ReLU nets can be used as a "product-gate" [48, Proposition 3] to extract the "product" relation between variables. Then, deep ReLU nets with $\mathcal{O}(\log(1/\varepsilon))$ hidden layers and free parameters can approximate arbitrary polynomials defined on $\mathbb{I}^d$ [43]. Therefore, even for some simple data features, deep ReLU nets theoretically beat shallow ReLU nets, showing the necessity of the depth in feature extraction. However, as shown in [16, Sec. 8.2] and [1], both the convergence issue of the stochastic gradient descent algorithm and the gradient vanishing phenomenon make it be difficult to practically derive a deep ReLU net estimator, which hinders the usefulness and efficiency of Lemma 2. Figure 3 presents the

difficulty for deep ReLU nets in extracting a 2-dimensional "square-feature" defined as $f(t) = t_1^2 + t_2^2$. For each depth, the network of the best performance is chosen and shown in the figure as a representation for the depth, by searching various (tens of) combinations of widths and step sizes. The statistics of each depth are made from 100 trials. The relation between accuracy and depth is recorded in Figure 3 (a) and that between the frequencies of valid models and depth is recorded in Figure 3 (b). As shown, the network performs less robust when it gets deeper.



(a) Accuracy and depth (b) Valid model and depth

Fig. 3: The role of depth for approximating t^2 using SGD

To end this section, we mention that the "square-gate" in Lemma 2 also holds for shallow nets with analytic activation functions and large weights [9]. With extremely large weights, it was proved in [34] that there is a deep net with two hidden layers and analytic activation functions that can approximate any continuous function to an arbitrary accuracy. Since large weights are difficult to be numerically realized, we focus on deep ReLU nets which require the magnitude of weights to be at most $\mathcal{O}(\varepsilon^{-\theta})$ for some $\theta > 0$.

3 Depth Selection for Extracting a Single Feature

In this section, we introduce several data features and study the role of depth in extracting these data features to break through the feature-depth correspondences. Since there are numerous symbols involved in different data features, we provide a table of notations as follows.

TABLE 2: Notations

L: number of layers	n: number of parameters
d_j: width in the j-th layer	d: input dimension
r: smoothness index	μ sparseness index
$\mathcal{R}$: bound of parameters	B: bound of coefficients
d^*: group structure dimension	$\jmath$: group structure degree
D_j: size of the j-th group	β: degree of polynomials

3.1 Data features

In a seminal review paper [3], Bengio et al. presented a fruitful review on intuitive and experimental explanations for the success of deep learning in feature extraction. From a numerical viewpoint, deep nets can extract numerous data features including those for smoothness, hierarchical organization, shared factors, manifold structures and sparsity, part of which were theoretically verified in the recent paper [18]. In particular, [18] rigorously proved that with the similar capacity costs measured by the covering number, deep nets beat shallow nets in extracting data features listed in Table 1 while perform not essentially better than shallow nets in extracting the smoothness feature.

Let $f^* : \mathbb{I}^d \to \mathbb{R}$ be a function to model the potential relation between input and output, i.e., $y \approx f^*(x)$ with $x \in \mathbb{I}^d$ and $y \in \mathbb{R}$ the input variable and output variable respectively. Both structures of x and properties of f^* are regarded as data features. In the following, we introduce the smoothness feature of f^*.

Definition 1. *Let $c_0 > 0$ and $r = s + v$ with $s \in \mathbb{N}_0 := \{0\} \cup \mathbb{N}$ and $0 < v \leq 1$. We say a function $f : \mathbb{I}^d \to \mathbb{R}$ is (r, c_0)-smooth if f is s-times differentiable and for every $\alpha_j \in \mathbb{N}_0$, $j = 1, \ldots, d$ with $\alpha_1 + \cdots + \alpha_d = s$, its s-th partial derivative satisfies the Lipschitz condition*

$$\left| \frac{\partial^s f}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}}(x) - \frac{\partial^s f}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}}(x') \right| \leq c_0 \|x - x'\|_2^v, \quad (6)$$

where $x, x' \in \mathbb{I}^d$ and $\|x\|_2$ denotes the Euclidean norm of x. Denote by $Lip^{(r,c_0)}$ the set of all (r, c_0)-smooth functions defined on $\mathbb{I}^d$.

The smoothness feature of f^* illustrates that $x \approx x'$ implies $f^*(x) \approx f^*(x')$. It is a standard feature to describe f^* and has been used in vast literature [7], [25], [17], [38], [48], [49], [32]. However, it remains open whether deep ReLU nets can achieve the optimal performance of algebraic polynomials for realizing the smoothness feature, though encouraging developments have been made in [48], [38]. Furthermore, as pointed out in [3], the smoothness feature of f^* is insufficient to get around the curse of dimensionality, which requires additional structure features of x. To this end, we introduce the following group structure feature for the input.

Definition 2. *Let $\jmath, d^* \in \mathbb{N}$ and $D_1, \ldots, D_{d^*} \in \mathbb{N}$ satisfy $d = D_1 + \cdots + D_{d^*}$. We say x possesses a $(D_1, \ldots, D_{d^*})$-group structure of order $\jmath$ with respect to f^*, if there exists some polynomials $P_{k,\jmath}$, $k = 1, \ldots, d^*$ defined on $\mathbb{I}^{D_k}$ and of degree at most $\jmath$ and a function $g : \mathbb{R}^{d^*} \to \mathbb{R}$ such that*

$$\begin{aligned} f^*(x) \quad = \quad & g\big[P_{1,\jmath}(x^{(1)}, \ldots, x^{(D_1)}), \ldots, \\ & P_{d^*,\jmath}(x^{(d-D_{d^*}+1)}, \ldots, x^{(d)})\big]. \end{aligned} \quad (7)$$

The group structure depicts the relation between different input variables. The case $d^* = d$ and $P_{k,\jmath}(t) = t$ for $k = 1, \ldots, d$ denotes that all variables in x are independent. The rotation-invariance assumption [9] is included in the case $d^* = 1$ and implies that variables in x possess strong dependence. The group structure assumption is more general than the manifold assumption [8], rotation-invariance assumption [9] and sparseness assumption [43] via imposing different restrictions on $P_{k,\jmath}$.

To show the outperformance of deep nets, we impose both the smoothness assumption on f^* and group structure assumption on the input. Such a smooth-structure assumption abounds in applications. For $d^* = 1$ and $P_{1,\jmath}(x) = (x^{(1)})^2 + \cdots + (x^{(d)})^2$, the smooth-structure assumption refers to a radial function that plays an important role in designing earthquake early warning systems [42]. For $d^* < d$ and $P_{1,\jmath}(x^{(1)}, \ldots, x^{(D_1)}) = (x^{(1)})^2 + \cdots + (x^{(D_1)})^2$, the feature assumption is related to a partially radial function

that is important in predicting the magnitude of earthquake [47]. For $d^* = d$ and

$$g(P_{1,j}(x^{(1)}), \ldots, P_{d,j}(x^{(d)}) = P_{1,j}(x^{(1)}) + \cdots + P_{d,j}(x^{(d)}),$$

the smooth-structure assumption corresponds to the well known additive model [25] with polynomial kernels in statistics. If there exists some $P_{k,j}(\cdot) = 0$, the assumption then implies sparseness which is standard in computer vision [33].

3.2 Depth selection for extracting the group structure

It was shown in [35] that for some fixed activation function, i.e., analytic and non-polynomials, shallow nets with $\binom{\beta+d}{d}$ neurons can approximate any polynomial defined on $\mathbb{I}^d$ of degree $\beta \in \mathbb{N}$ within an arbitrary accuracy. However, if the polynomial is sparse, then shallow nets fail to catch the sparseness information [27] in the sense that the same number of neurons is required to approximate sparse and non-sparse polynomials. However, [27], [43] found that deep nets essentially improve the performance of shallow nets by using the "product-gate" property of deep nets [48]. In particular, for deep ReLU nets, the following lemma was proved in [43, Proposition 3.3].

Lemma 3. *For any $0 < \varepsilon < 1$ and $\ell \in \mathbb{N}$, there exists a deep ReLU net $\tilde{\prod} : \mathbb{R}^\ell \to \mathbb{R}$ with $\mathcal{O}[(1 + \log \ell)\log(\ell/\varepsilon)]$ depth and $\mathcal{O}[(1 + \log \ell)\log(\ell/\varepsilon)]$ free parameters such that for any $u_1, \ldots, u_\ell$ satisfying $|u_k| \le 1, k = 1, \ldots, \ell$, there holds*

$$\left|\tilde{\prod}(u_1, \ldots, u_\ell) - \prod_{k=1}^{\ell} u_k\right| \le \varepsilon.$$

Noting that each monomial defined on $\mathbb{I}^d$ of degree at most β can be rewritten as β products of elements in $[0, 1]$, it requires a deep net with $\mathcal{O}[(1 + \log \beta)\log(\beta/\varepsilon)]$ depth and free parameters to extract the monomial feature according to Lemma 3. Based on the "product-gate-unit" (PGU), we can construct a deep net such that for any μ-sparse polynomials, there are only $\mu+\mathcal{O}[(1+\log \beta)\log(\beta/\varepsilon)]$ free parameters involved to extract this structure feature, which is much smaller than $\binom{\beta+d}{d}$ provided μ is small and ε is not extremely small.

Although the above interesting result illustrates the power of depth in extracting structure features, the depth of the constructed deep ReLU net depends on the approximation accuracy, making it be practically difficult to get a deep net estimator, just as Figure 3 purports to show. In this paper, we pursue a trade-off between depth and number of free parameters in extracting structure features by using an approach developed in a recent paper [38]. The following "product-gate" for deep ReLU nets is our main tool, whose proof can be found in Appendix A in Supplementary Materials.

Lemma 4. *Let $\theta > 0$ and $\tilde{L} \in \mathbb{N}$ with $\tilde{L} > (2\theta)^{-1}$. For any $\ell \in \{2, 3, \ldots, \}$ and $\varepsilon \in (0, 1)$, there exists a deep ReLU net $\tilde{\times}_\ell : \mathbb{R}^\ell \to \mathbb{R}$ with $2\ell\tilde{L} + 8\ell$ layers and at most $c\ell^\theta\varepsilon^{-\theta}$ free parameters bounded by $\ell^\gamma\varepsilon^{-\gamma}$ such that*

$$|u_1u_2\cdots u_\ell - \tilde{\times}_\ell(u_1, \ldots, u_\ell)| \le \varepsilon, \quad \forall u_1, \ldots, u_\ell \in [-1, 1],$$

where c and γ are constants depending only on θ and $\tilde{L}$.

Comparing Lemma 4 with Lemma 3, as a "product-gate", we reduce the depth of deep ReLU nets on the price of adding the number of free parameters. The positive number θ performs as a balance exponent in the sense that small θ implies large depth but few free parameters, while large θ means small depth but a great number of free parameters. For a fixed and ε-independent exponent θ, the depth of ReLU nets in Lemma 4 is independent of the accuracy ε, while the number of free parameters increases from $\mathcal{O}[(1 + \log \ell)\log(\ell/\varepsilon)]$ to $\ell^{\ell\theta}\varepsilon^{-\ell\theta}$. Therefore, Lemma 4 exhibits a trade-off between depth and parameters and removes the feature-depth correspondence.

Denote by $\mathcal{P}_\beta^d$ the set of algebraic polynomials defined on $\mathbb{I}^d$ with degree at most β. For $B \ge 1$, define further $\mathcal{P}_{\beta,B}^d := \left\{\sum_{|\alpha|\le\beta} c_\alpha x^\alpha : |c_\alpha| \le B\right\}$ the set of polynomials in $\mathcal{P}_\beta^d$ whose coefficients are uniformly bounded by B, where $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$, $|\alpha| = \alpha_1 + \cdots + \alpha_d$ and $x^\alpha = (x^{(1)})^{\alpha_1}\cdots(x^{(d)})^{\alpha_d}$. Define $\mathcal{P}_{\beta,B,\mu}^d$ the set of all μ-sparse polynomials in $\mathcal{P}_{\beta,B}^d$. So $P \in \mathcal{P}_{\beta,B,\mu}^d$ has at most μ nonzero coefficients. The following theorem shows the performance of deep ReLU nets in extracting the sparse polynomial feature, whose proof will be given in Appendix A in Supplementary Materials.

Theorem 1. *Let $\beta, \mu \in \mathbb{N}$, $B, \theta > 0$ and $\tilde{L} \in \mathbb{N}$ with $\tilde{L} > (2\theta)^{-1}$. For any $0 < \varepsilon < 1$, there is a deep ReLU net structure with $2\beta\tilde{L} + 8\beta + 1$ layers and at most $\mu + c(\mu\beta B)^\theta\varepsilon^{-\theta}$ nonzero parameters bounded by $\max\{B, (\mu\beta B)^\gamma\varepsilon^{-\gamma}\}$, such that for each $P \in \mathcal{P}_{\beta,B,\mu}^d$ there exists a h_P with the aforementioned structure satisfying*

$$|P(x) - h_P(x)| \le \varepsilon, \qquad \forall x \in \mathbb{I}^d,$$

where c and γ are the constants in Lemma 4.

A similar result has been established in [43] for deep ReLU nets with depth $\mathcal{O}(\log \beta \log(\mu B\beta/\varepsilon))$ and number of free parameters $\mu + \mathcal{O}(\log \beta \log(\mu B\beta/\varepsilon))$. Our result is different from [43] by introducing an exponent θ to balance the depth and number of free parameters. For a fixed θ, the depth of deep ReLU nets studied in Theorem 1 is independent of ε. Thus, Theorem 1 shows a novel relation between the depth and feature extraction for sparse polynomial features as well as the group structure features, by means of breaking through the exclusive feature-depth correspondence in [43]. Furthermore, if μ is not extremely small, we can select a θ in Theorem 1 such that the capacity of deep nets in Theorem 1 is smaller than that in [43] according to Lemma 1. That is, Theorem 1 provides a theoretical guidance on using smaller capacity costs than [43] to get a same accuracy in extracting the sparse feature.

3.3 Deep nets for extracting the smoothness feature

In [48], Yarotsky succeeded in establishing a tight error estimate of approximating smooth functions by deep ReLU nets by utilizing the "product-gate" property in Lemma 3. [48, Theorem 1] showed that for any $f \in Lip^{(r,c_0)}$ with $r \in \mathbb{N}$, there is a deep ReLU net $h_f^\diamond$ with fixed structure, n free parameters and $\mathcal{O}(\log n)$ layers such that

$$\|f - h_f^\diamond\|_{L^\infty(\mathbb{I}^d)} \le c' n^{-r/d}\log n, \tag{8}$$

where c' is a constant depending only on c_0, d, r and $p \in [1,\infty)$. Comparing with standard results for linear approximants such as algebraic polynomials [39], there is an additional logarithmic term in (8). This is due to the accuracy-dependent depth in Lemma 3.

This phenomenon was firstly noticed in [38]. After deriving the "product-gate" property for deep ReLU nets with accuracy-independent depth, [38, Theorem 3.1] proved that there exists a deep ReLU net h_f^* with fixed structure, n free parameters layered on $(2+\lceil \log r \rceil)(11+r/d)$ hidden layers such that

$$\|f - h_f^*\|_{L^p(\mathbb{I}^d)} \leq c^* n^{-r/d}, \tag{9}$$

where c^* is a constant depending only on c_0, d, r and $p \in [1,\infty)$. It is obvious that (9) improves (8) by removing the logarithmic term. However, the analysis in [38] relies on the localized approximation [7] of deep nets and thus, their result holds only under the $L^p(\mathbb{I}^d)$ norm with $1 \leq p < \infty$. Noting that for $f \in L^\infty(\mathbb{I}^d)$, $\|f\|_{L^p(\mathbb{I}^d)} \leq \|f\|_{L^\infty(\mathbb{I}^d)}$, (9) does not match the optimal rate of uniform approximation by linear approximants. In the following theorem, we combine the approaches in [38] and [48] to get a sharp error estimate of approximating smooth functions by deep ReLU nets under the $L^\infty(\mathbb{I}^d)$ metric.

Theorem 2. *Let $r = s + v$ with $s \in \mathbb{N}_0$ and $0 < v \leq 1$, $c_0, \theta > 0$ and $\tilde{L} \in \mathbb{N}$ with $\tilde{L} > (2\theta)^{-1}$. For any $\varepsilon \in (0,1)$, there exists a deep ReLU net structure with*

$$\mathcal{L}(d, r, \tilde{L}) := 2(d+s)\tilde{L} + 8(d+s) + 3 \tag{10}$$

layers and at most $c(d+s)^\theta \varepsilon^{-(r+d)\theta/r} + (8d+5)\binom{s+d}{s}\varepsilon^{-d/r}$ free parameters bounded by $\max\{\tilde{B}, 3\varepsilon^{-1/r}, (d+s)^\gamma \varepsilon^{-(r+d)\gamma/r}\}$, such that for any $f \in Lip^{(r,c_0)}$ there is a h_f with the aforementioned structure satisfying

$$\|f - h_f\|_{L^\infty(\mathbb{I}^d)} \leq c_1 \varepsilon, \tag{11}$$

where c_1 is a constant depending only on c_0, d and r and

$$\tilde{B} := \max_{k_1+\cdots+k_d \leq s} \max_{x \in \mathbb{I}^d} \left| \frac{1}{k_1! \ldots k_d!} \frac{\partial^{k_1+\ldots k_d} f(x)}{\partial^{k_1} x^{(1)} \ldots \partial^{k_d} x^{(d)}} \right|.$$

The proof of Theorem 2 will be presented in Appendix B in Supplementary Materials. Setting $\varepsilon = n^{-r/d}$, we get from Theorem 2 that there exists a deep net h_f with at most $\mathcal{O}(n^{\max\{1,(r+d)\theta/d\}})$ free parameters and $\mathcal{L}(d, r, \tilde{L})$ layers for $\tilde{L} > (2\theta)^{-1}$ such that

$$\|f - h_f\|_{L^\infty(\mathbb{I}^d)} \leq c_1 n^{-r/d}. \tag{12}$$

The depth plays a crucial role in extracting the smooth features in the sense that to derive a similar approximation accuracy as linear approximants, θ should be not larger than $d/(r+d)$, implying $\tilde{L} > (r+d)/(2d)$. However, when the depth $\mathcal{L}(d, r, \tilde{L})$ with $\tilde{L}$ reaching this critical value, deep nets with various depths are capable of extracting smooth features. This removes the feature-depth correspondence in extracting the smooth feature by making use of the structure of deep nets, since our constructed deep nets in the proof are sparse and share weights, which is different from the deep nets in the prominent work [48]. Recalling Lemma 1, for appropriately selected θ, the capacity of deep nets in our construction is smaller than that of [48] by removing the logarithmic term caused by the accuracy-dependent layers. Inequalities like (11) have been established for shallow nets with some sigmoid-type activation functions [35], [28]. However, different from Theorem 2, the magnitudes of weights in [35] are so large that the capacity restriction (5) does not hold and the result in [28] suffers from the well known saturation phenomenon in the sense that the approximation rate cannot be improved any further when the smoothness of the target function goes beyond a specific level. It can be found in Theorem 2 that deepening the networks succeeds in overcoming these problems. Although [18, Theorem 2] declares that to extract the smoothness feature, deep nets perform not essentially better than shallow nets or linear approximant, our result in Theorem 2 yields that deep ReLU nets are at least not worse than shallow nets.

4 Depth Selection in Exacting Composite Features

The previous section demonstrated the role of depth in extracting a single data feature. However, as shown in [3], it is much more important to simultaneously extract multiple features to feed the target-driven learning. Extracting composite features by deep nets, which is the purpose of this section, brings novel challenges in designing deep nets, including the junction of deep nets with different utilities, the balance of accuracy and depth, and the depth-parameter trade-off.

To build up a network to exact composite features, an intuitive approach is to stack deep nets by the a-priori information or human experiences in a tandem manner, just as Figure 1 implies. The problem is, however, that such a brutal stacking is practically inefficient, for both the unavailability of the a-priori information and lacking of the prescribed accuracy for extracting a specific feature. More importantly, the stacking scheme requires much more free parameters and depths of deep nets to extract composite features, adding additional capacity costs according to Lemma 1

In this section, we provide some theoretical guidance on selecting depth of deep nets to extract composite features by taking the depth-parameter trade-off into account. Without loss of generality, we are interested in extracting features exhibited in the following assumption.

Assumption 1. *Let $r = s + v$ with $s \in \mathbb{N}_0$ and $v \in (0,1]$, $D_1, \ldots, D_{d^*}, d^* \in \mathbb{N}$ with $d = D_1 + \cdots + D_{d^*}$, and $\jmath, \mu \in \mathbb{N}$. Assume that there is a function g defined on $\mathbb{I}^{d^*}$ satisfying $g \in Lip^{(r,c_0)}$ such that (7) holds with $P_{k,\jmath} \in \mathcal{P}^{D_k}_{\jmath,1/2,\mu}$ for $k = 1, \ldots, d^*$.*

There are totally three types of features in Assumption 1, the smoothness feature of g as well as f^*, the group structure feature of x, and the sparsity feature of the structure polynomials $P_{k,\jmath}$, $k = 1, \ldots, d^*$. An intuitive observation is that the depth and number of free parameters of deep nets to simultaneously extract these three features should be larger than those to extract each single feature. However, as shown in the following theorem, it is not necessarily the case, provided the deep nets for different utilities are appropriately combined.

Theorem 3. *Let $r = s+v$ with $s \in \mathbb{N}_0$ and $v \in (0,1]$, $d^*, \jmath, \mu \in \mathbb{N}$, $c_0, \theta > 0$ and $\tilde{L} \in \mathbb{N}$ with $\tilde{L} > (2\theta)^{-1}$. For any $0 < \varepsilon < 1/2$, there exists a deep ReLU net structure with at most*

$$\mathcal{L}^*(d^*, r, \tilde{L}, \jmath) = \overbrace{\mathcal{L}(d^*, r, \tilde{L})}^{\textit{smooth+group}} + \overbrace{2\jmath\tilde{L} + 8\jmath + 1}^{\textit{group}} \tag{13}$$

layers and at most

$$\mathcal{W}(d^*, \varepsilon, \mu, \jmath, \theta) := \overbrace{(8d^* + 5)\binom{s+d^*}{s}\varepsilon^{-d^*/r}}^{\textit{smooth+group}} \tag{14}$$
$$+ \overbrace{\mu d^*}^{\textit{sparse+group}} + \overbrace{c(d^*+s)^\theta \varepsilon^{-(r+d^*)\theta/r} + c(\mu\jmath)^\theta \varepsilon^{-\theta/\tau_r}}^{\textit{depth-parameter trade-off}}$$

free parameters bounded by

$$\max\{\tilde{B}_g, 3\varepsilon^{-1/r}, (d^*+s)^\gamma \varepsilon^{-(r+d^*)\gamma/r}, (\mu\jmath)^\gamma \varepsilon^{-\gamma/\tau_r}\} \tag{15}$$

such that, for any f^* satisfying Assumption 1, there is an h_{f^*} possessing the aforementioned structure satisfying

$$\|f^* - h_{f^*}\|_{L^\infty(\mathbb{I}^d)} \le c_2\varepsilon$$

where $\tau_r = \begin{cases} 1, & r \ge 1, \\ v, & r < 1, \end{cases}$ and $c_2, \tilde{B}_g$ are constants depending only on c_0, r, d^* and g.

The proof of Theorem 3 will be given in Appendix C in Supplementary Materials. Assumption 1 implies $f^* \in Lip^{(r,c_0)}$, which requires $\mathcal{L}(d, r, \tilde{L})$ layers to extract the smoothness feature according to Theorem 2. However, with the help of the group structure feature, (13) exhibits a reduction of layers from $\mathcal{L}(d^*, r, \tilde{L})$ to $\mathcal{L}(d, r, \tilde{L})$. To extract the group structure feature itself, additional $2\jmath\tilde{L} + 8\jmath + 1$ layers are required. This shows that the classical tandem stacking is not necessary. In particular, for some specific group structure features, taking $d^* = 1$ and $\jmath = 1$ for example, it is easy to select some $\theta > 0$ such that $\mathcal{L}^*(d^*, r, \tilde{L}, \jmath) \le \mathcal{L}(d, r, \tilde{L})$, implying a waste of source of the tandem stacking.

The number of free parameters, as exhibited in (14), reflects the price to pay for extracting three composite features. To yield an accuracy of order ε, the group structure and smoothness feature require at least $\varepsilon^{-d^*/r}$ free parameters. It should be mentioned that this number cannot be reduced further according to [18, Theorem 2] by noting in Assumption 1 that f^* corresponds a smooth function defined on $\mathbb{I}^{d^*}$. The second term in (14) reflects the difficulty in extracting the group structure feature. Without the sparseness assumption, it requires at least $\mathcal{O}\left(\sum_{k=1}^{d^*} \jmath^{D_k}\right)$ free parameters. If there is some k such that $\jmath^{D_k} > \varepsilon^{-d^*/r}$, extracting the group structure feature becomes the main difficulty in the learning process. This imposes a strict restriction on $\jmath$ to maintain the optimality. The sparsity assumption reduces this risk, allowing $\jmath$ to be very large. The rest two term in (14) illustrates a depth-parameter tradeoff in extracting composite features. In particular, to guarantee the optimal capability of feature extraction, θ must be smaller than the critical value $\theta_0 := \min\left\{\frac{d^*}{d^*+r}, \frac{d^*\tau_r}{r}\right\}$. This implies a smallest depth in (13) by noting $\tilde{L} > 1/(2\theta)$. In a word, less parameters requires smaller θ, which results in larger $\tilde{L}$ and consequently larger $\mathcal{L}^*(d^*, r, \tilde{L}, \jmath)$, while more parameters require larger θ, and consequently smaller $\tilde{L}$ and depth.

As Theorem 3 shows, the depth of network is not unique to extract composite features, provided it is larger than a certain level. Furthermore, our results imply two important advantages of deep nets in feature extraction. One is that, different from the classical tandem tackling, deep nets succeed in extracting composite features by embodying their interactions, and thus reduce the capacity costs. Such a reduction plays an important role in generalization, which will be analyzed in the next section. The other is the versatility of deep nets in extracting both single features and composite features in the sense that each feature corresponds to numerous depths, and vice versa. To end this section, we present two corollaries for deep nets to extract composite features. The first one is the smoothness and radial features. Let $d^* = 1$ and $P_{1,\jmath}(x) = \frac{1}{\sqrt{d}}[(x^{(1)})^2 + \cdots + (x^{(d)})^2]$, then f^* is a radial function [9]. Setting $\theta = \tau_r/(2+2r)$ and $\tilde{L} = 2(r+1)/\tau_r$, we have from Theorem 3 with $\jmath = 2$ and $\mu = 1$ the following corollary directly.

Corollary 1. *There exists a deep ReLU net structure with $4(d+s+2)(r+1)/\tau_r + 8(d+s) + 20$ layers and at most $c_3 n$ nonzero free parameters bounded by $c_4 n^{\max\{1,(r+1)\gamma,\gamma r/\tau_r\}}$ such that for any radial function $f^* \in Lip^{(r,c_0)}$ there is a deep net h_{f^*} with the aforementioned structure satisfying*

$$\|f^* - h_{f^*}\|_{L^\infty(\mathbb{I}^d)} \le c_5 n^{-r},$$

c_3, c_4, c_5 are constants depending only on c_0, r, d and f^*.

The derived approximation rate is almost optimal according to [9] in the sense that the best approximation error for all deep nets satisfying the capacity restriction (4) with n parameters is of order $(n/\log n)^{-r}$. Our second corollary considers using deep nets to simultaneously extract the partially radial and smooth features. Let $d' \le d$ and $f^*(x) = f(\overbrace{x^{(1)}, \ldots, x^{(d')}}^{\text{radial}}, x^{(d'+1)}, \ldots, x^{(d)}) = g(t_{d'}, x^{(d'+1)}, \ldots, x^{(d)})$ with $t_{d'} = (d')^{-1/2}((x^{(d_1)})^2) + \cdots + (x^{(d')})^2) \in [0,1]$. Let $\theta = \frac{(d-d'+1)\tau_r}{2(d-d'+1+r)}$ and $L = \frac{2(d-d'+1+r)}{(d-d'+1)\tau_r}$. The following corollary is a direct consequence of Theorem 3 with $d^* = d - d' + 1$, $\jmath = 2$ and $\mu = 1$.

Corollary 2. *There exists a deep ReLU net structure with*

$$\frac{4(d-d'+1+r)(d-d'+3+s)}{(d-d'+1)\tau_r} + 8(d-d'+1+s) + 20$$

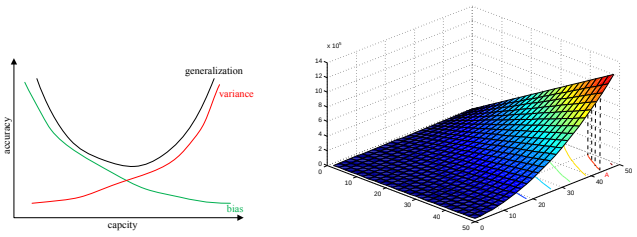
layers and at most $c_6 n$ free parameters bounded by $c_7 n^{\max\{1,(r+d-d'+1)\gamma,\gamma r/\tau_r\}/(d-d'+1)}$ such that for any partially radial function $f^* \in Lip^{(r,c_0)}$ there is a deep net h_{f^*} with the aforementioned structure satisfying

$$\|f^* - h_{f^*}\|_{L^\infty(\mathbb{I}^d)} \le c_8 n^{-r/(d-d'+1)},$$

where c_6, c_7, c_8 are constants depending only on c_0, r, d and f^*.

5 Generalization Capability of Deep Nets

This section aims at the generalization capability of deep ReLU nets. Our analysis is carried out in the standard learning theory framework [10], where a sample $D_m = \{(x_i, y_i)\}_{i=1}^m$ with $x_i \in \mathcal{X} = \mathbb{I}^d$ and $y_i \in \mathcal{Y} \subseteq [-M, M]$ for some



(a) Bias-variance trade-off (b) Depth-width trade-off

Fig. 4: Bias-variance trade-off for ERM on deep nets

$M > 0$ is assumed to be drawn independently according to an unknown Borel probability measure ρ on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. The generalization capability of an estimator f is measured by the generalization error, $\mathcal{E}(f) := \int_{\mathcal{Z}}(f(x) - y)^2 d\rho$, which quantifies the relation between the sample size m and prediction accuracy. The primary objective is to find an estimator f_D based on D_m of the regression function $f_\rho(x) = \int_{\mathcal{Y}} y d\rho(y|x)$ that minimizes the generalization error, where $\rho(y|x)$ denotes the conditional distribution at x induced by ρ. Therefore, the generalization capability of f_D is measured by $\mathcal{E}(f_D) - \mathcal{E}(f_\rho)$.

Let $\mathcal{H}_{n,L,\mathcal{R}}$ be defined by (3). We consider generalization error estimates for the following empirical risk minimization (ERM):

$$f_{D,n,L} := \arg \min_{f \in \mathcal{H}_{n,L,\mathcal{R}}} \frac{1}{m} \sum_{i=1}^{m} [f(x_i) - y_i]^2. \tag{16}$$

Since $|y_i| \leq M$, it is natural to project the final output $f_{D,n,L}$ to the interval $[-M, M]$ by the truncation operator $\pi_M f_{D,n,L}(x) := \text{sign}(f_{D,n,L}(x)) \min\{|f_{D,n,L}(x)|, M\}$.

From Theorems 1-3, the accuracy of feature extraction decreases as the capacity of deep nets increases, resulting in small bias for the ERM. However, too large capacity makes ERM be sensitive to noise and leads to large variance. This is the well known bias-variance dilemma [10, Chap.1]. The optimal generalization performance for ERM is obtained by balancing the bias and variance, just as Figure 4 (a) purports to show. For ERM on deep nets, the problem is that the capacity depends on both depth and number of free parameters. As shown in Figure 4 (b), all (L, n) pairs in the curve "A" share the same covering number bounds in Lemma 1 with $\varepsilon = 0.01$. In summary, there are two dilemmas to get a good generalization for ERM on deep nets: bias-variance dilemma in selecting the capacity and depth-parameter dilemma in controlling the bias. The purpose of our study is not only to pursue the optimal generalization error for ERM on deep nets, but also to derive feasible candidates of (L, n) pairs to realize the optimality. The main result is the following theorem.

Theorem 4. *Let $0 < \delta < 1$, $\mu, \jmath, d, d^* \in \mathbb{N}$, $r = s + v$ with $s \in \mathbb{N}_0$ and $0 < v \leq 1$. For any*

$$0 < \theta \leq \theta_0 := \min\left\{\frac{d^*}{d^* + r}, \frac{d^* \tau_r}{r}\right\}, \tag{17}$$

if $L = \mathcal{L}^*(d^*, r, \tilde{L}, \jmath)$ with $\tilde{L} > (2\theta)^{-1}$, $n = \left[C_1 m^{\frac{d^*}{2r+d^*}}\right]$, f_ρ satisfies Assumption 1,

$$\mu \jmath \leq n^{\frac{\tau_r d^* + \theta}{d^* \tau_r \theta}}, \qquad \text{and} \quad \mu \leq n, \tag{18}$$

then

$$\mathcal{E}(\pi_M f_{D,n,L}) - \mathcal{E}(f_\rho) \leq C_2 \tilde{L}^2 m^{-\frac{2r}{2r+d^*}} \log m \log \frac{3}{\delta} \tag{19}$$

holds with confidence at least $1 - \delta$, where C_1, C_2 are constants independent of δ, m, $\tilde{L}$ or n and $\mathcal{L}^*(d^*, r, \tilde{L}, \jmath)$ is defined in (13).

The proof of Theorem 4 will be given in Appendix D in Supplementary Materials. It should be noted that the established learning rate cannot be essentially improved in the sense that for some special $P_{k,\jmath}$, the learning rate is optimal [9, Theorem 3]. Condition (18) presents a restriction on the group structure and sparsity features in the sense that either μ or $\jmath$ should be relatively small with respect to the size of data. Since our result holds for any θ satisfying (17) while the depth L and number of free parameters depend on θ, there are numerous (L, n) pairs to achieve the optimal generalization error bound exhibited in (19), provided the constant factor is neglected. However, there is an additional $\tilde{L}^2$ on the right-hand sides of (19), implying that extremely small θ may affect the learning rate and deep ReLU nets with relatively small depth is preferable. As $\theta < \theta_0$ and $\tilde{L} > (2\theta)^{-1}$, Theorem 4 also implies that there is a critical depth, larger than which deep nets with suitable structures can achieve the established optimal generalization error bounds.

We then compare our established generalization error bounds with some related work. Without the group-structure assumption, optimal learning rates for learning (r, c_0)-smooth functions on $\mathbb{I}^d$ have been established for shallow nets in [19], [31] to achieve an order $\mathcal{O}(m^{-\frac{2r}{2r+d}})$. Since we impose additional assumption on the input space, our derived bounds in (19) are much sharper. This is highly nontrivial since most of our work is to demonstrate the power of depth in reflecting the group-structure of the input space. In particular, even restricting to learning the most commonly used group-structure, i.e., radial functions, shallow nets require at least $[m^{\frac{d-1}{2r+1}}]$ neurons to guarantee the generalization error of order $m^{-\frac{2r}{2r+1}}$ [9].

Theoretically, Theorem 4 shows that there are numerous depths to achieve a similar optimal generalization error bound, which contradicts our numerical results in the following section at the first glance, since numerical results show that there is always an optimal depth L to optimize the generalization capability of deep nets. We explain such an contradiction in the following remark.

Remark 1. *In our theoretical analysis, we focus on the power of depth from the model selection viewpoint, that is, we aim at deriving the generalization error of implementing ERM on deep ReLU nets without taking the availability of optimization algorithms into account. As solving ERM (16) involves highly non-convex optimization problems, it is difficult to design an optimization algorithm with perfect convergence guarantee. Therefore, there is a gap between our theoretical analysis and numerical results. To be detailed, in numerical experiments, we devote to the generalization capability of the estimator $f_{D,n,L,s}$ derived from the SGD algorithm, which can be written as*

$$\{\mathcal{E}(f_{D,n,L}) - \mathcal{E}(f_\rho)\} + \{\mathcal{E}(f_{D,n,L,s}) - \mathcal{E}(f_{D,n,L})\},$$

while our analysis is only carried out for the error $\mathcal{E}(f_{D,n,L}) - \mathcal{E}(f_\rho)$. To the best of our knowledge, it still remains open how to quantify the error term $\{\mathcal{E}(f_{D,n,L,s}) - \mathcal{E}(f_{D,n,L})\}$ under the setting of our paper. We will leave it as a future study.

6 Experimental Results

In this section, we present both toy simulations and real data experiments to show the role of depth for ReLU nets in feature selection and prediction. All the numerical experiments are carried out in the Python-3.5.4 environment running on a workstation with a Pascal Titan X 12-GB GPU and 24-GB memory. Our implementation is derived from the publicly available Tensorflow-1.4.0 framework by using AdamOptimizer. Our codes are available at http://vision.sia.cn/our%20team/Hanzhi-homepage/vision-ZhiHan%28English%29.html.

6.1 Experimental setting

The settings of simulations are described as follows.

Implementation and Evaluation: The are five purposes in our experimental study. The first one is to verify the adaptivity of depths to the feature. The second one is to declare the adaptivity of features to the depth. The third one aims at demonstrating the necessity of depth in feature extraction. The fourth one focuses on the necessity of depths in generalization. In our last experiment, we show the power of deep nets in some real applications. In each simulation, we randomly generate m sample points $\{x_i\}_{i=1}^m$ on $\mathcal{X} \in \mathbb{R}^d$ according to the uniform distribution. Each x_i corresponds to an output y_i with either $y_i = f(x_i)$ (Sections 6.2, 6.3, 6.4) or $y_i = f(x_i) + \varepsilon_i$ (Section 6.5) with ε_i some Gaussian noise. We repeat 10 times and record the average values of the following five quantities:

- Mean squared error (MSE): given an estimator f_D, MSE, defined by $\frac{1}{m}\sum_{i=1}^m (f_D(x_i) - y_i)^2$, measures the average squared difference between the estimated values and what is estimated. It is a standard measurement to quantify the prediction performance of an estimator.
- Mean absolute error (MAE): MAE, defined by $\frac{1}{m}\sum_{i=1}^m |f_D(x_i) - y_i|$, quantifies the fitting performance of f_D. It is another popular measurement, which is less sensitive to outliers than MSE, to quantify the prediction performance.
- Median absolute error (MdAE): MdAE, defined by $m_{0.5}(|f_D(x_i) - m_{0.5}(y_i)|)$, is a robust measure of the variability of an estimator, where $m_{0.5}$ means a median. Thus, MdAE, together with MAE, shows the robustness of the estimator.
- R squared score (R^2S): R^2S, defined by $R^2S(y, f) = 1 - \frac{\sum_{i=1}^n (y_i - f_D(x_i))^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$ with $\bar{y} = \frac{1}{m}\sum_{i=1}^m y_i$, is a statistical measurement that represents the proportion of the variance of an estimate by that of real outputs in a regression model. It measures the fitness of the model.
- Explained variance score (EVS): EVS, defined by $1 - \frac{\sum_{i=1}^m (y_i - f_D(x_i))^2}{\sum_{i=1}^m y_i^2}$, measures the proportion to which a mathematical model accounts for the variation (dispersion) of a given data set.

All these measurements quantify the prediction performance of an estimator in terms of the prediction accuracy, sensitivity to outliers, robustness and fitness.

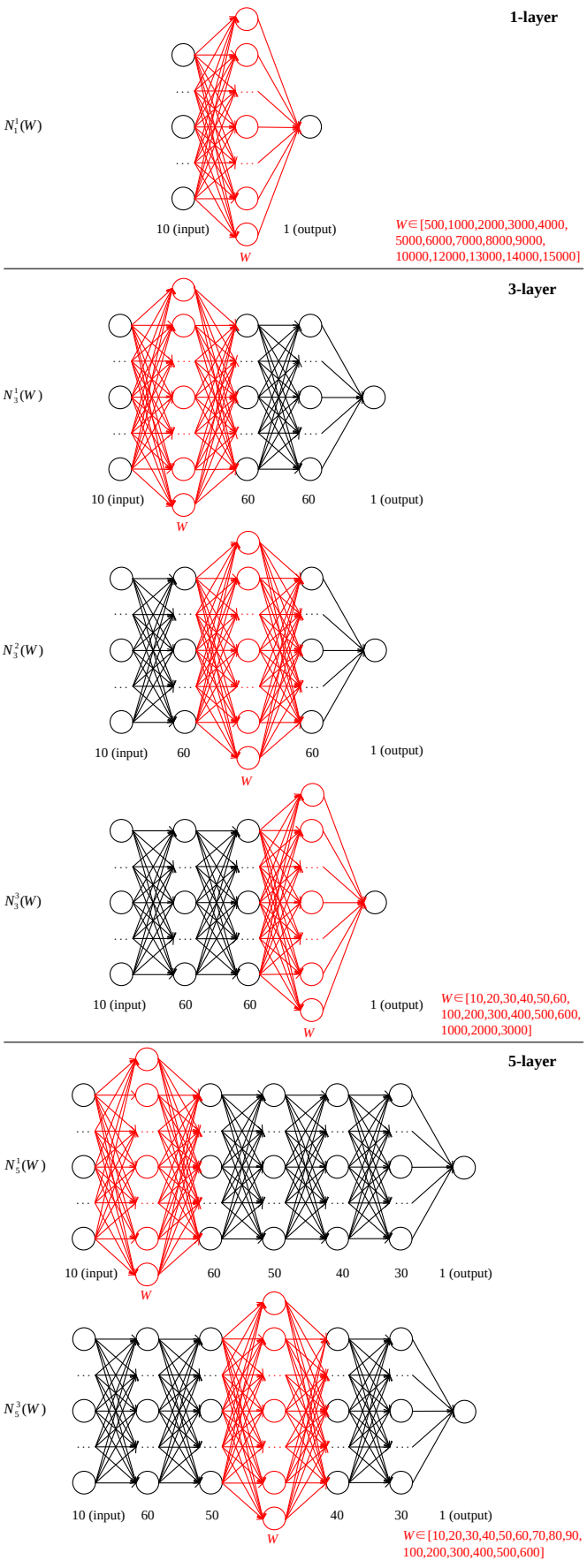


Fig. 5: Network architectures of various depths and widths.

Structures of deep nets: Generally speaking, there are four components in describing the structure of deep nets: depth, width in each layer, sparsity in conjunction, and sharing weights. In our experiments, network width is equivalent to the number of neurons. If the sparsity in conjunction and sharing weights are considered, there are too many structures even for a three-layer feed-forward network. Thus, we are only concerned with fully connected deep nets with different width and depth. In fact, we train over 200 networks of different depths and widths in our simulations. We use $N_L^\ell(W)$ to represent a network of L layers and width W in the ℓ-th layer (marked as red as in Figure 5). For example, $N_3^2(100)$ and $N_3^2(200)$ are both 3-layer networks. The widths in layer-1 and layer-3 are fixed. The only difference is the widths in layer-2 are 100 and 200, respectively. In Figure 5, we present some examples for the structures adopted in our simulations. The details of structures will be explained in each simulation, if it is needed.

6.2 Adaptivity of the Depth to features

In this subsection, we study the performance of deep nets in extracting the 10-dimensional "square-feature":

$$f(x) = \sum_{j=1}^{10} (x^{(j)})^2,$$

where $x = (x^{(1)}, \ldots, x^{(10)})$ is i.i.d. generated according to the uniform distribution on $[-100, 100]^{10}$. The sizes of training dataset and testing dataset are 3000 and 200, respectively. Our purpose is to show the adaptivity of structures to the square feature, i.e., there are various structures to extract the square feature.

6.2.1 *The necessity of depth*

For comparison, we train 135 networks of different depths and widths. The network architectures are illustrated in Figure 5. In particular, we choose 3 different depths $L = \{1, 3, 5\}$ and select 15 different widths, which are shown in different colors in Figure 6 and marked in different curves.

As shown in Figure 6, all the curves show similar patterns, i.e., along with the increasement of width, the MSE decreases at the beginning and increases later. The difference is, for the deeper networks, it generally needs smaller width to reach the best performance. The average widths in the varied layers corresponding to the best performance networks of 1-layer, 3-layer and 5-layer are 4000, 60 and 52, respectively.

The outperformance of 3-layer over shallow nets in Figure 6 verifies the necessity of depth and show that deep nets can extract the square feature better than shallow nets with much fewer neurons. The superiority of 3-layer over 5-layer deep nets demonstrates that there exists an optimal depth in extracting some specific feature. Here the optimality means not only the optimal accuracy, but also the solvability or convergence of the adopted AdamOptimizer algorithm, since its convergence issue is questionable when the depth increases [16]. Thus, although Theorem 1 proved that there are numerous depth-parameter pairs achieving the same accuracy, the convergence issue suggests to set the depth as small as possible.

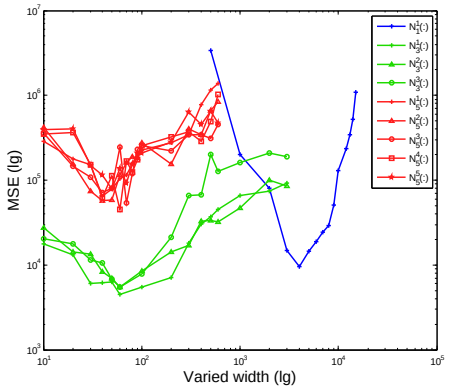


Fig. 6: MSE curves of networks with various structures

6.2.2 *Role of the width for 3-layer deep nets*

Theorem 1 presents a relation between the depth and the number of free parameters in extracting the square feature. However, it does not give any guidance on the distribution of the width in each hidden layer. In this experiment, we fix the total number of parameters of a 3-layer network at 8000 (slightly variation is allowed, and the range is $[8000, 8100]$). We manually change width of each layer, the numbers of parameters connecting Input and Layer-1 (C_1), connecting Layer-1 and Layer-2 (C_2), connecting Layer-2 and Layer-3 (C_3), and connecting Layer-3 and Output (C_4). Hence we generate a group of networks (20 in total) with different representative parameter distributions. The details of the networks are listed in Appendix E in Supplementary Materials.

We record the testing errors in Figure 7, where each spot represents one network and the coordinates $(p(C_1), p(C_2), p(C_3))$ are the percentage of the parameters occupied. As $p(C_1) + p(C_2) + p(C_3) + p(C_4) = 1$, the 3-layer networks of various distributions can be uniquely positioned by this 3-dimensional coordinate system. The size of the spot represents the MSE of the corresponding network, i.e., smaller spot indicates smaller MSE. To be noted, the biggest MSE that can be reflected by the size of the spot is 10000. The networks with MSE larger than 10000 are represented by yellow spots, while the red spots mean the corresponding networks do not converge at all.

Figure 7 exhibits two phenomena for deep nets in feature extraction. The first one is the huge impact of the width distribution. The pattern shown in this experiment is that more connections between Layer-1 and Layer-2 ($p(C_2)$) generally bring better results, while a large number of connections with Input or Output layers ($p(C_1)$ or $p(C_4)$) lead to bad performances. This phenomenon indicates why a network is usually designed in a spindle shape. The other one is the adaptivity of the structure to the "square-feature". It can be found in Figure 7 that all green points perform similarly, which means that if the depth is suitable selected, then there is a large range of the width distributions such that deep nets with such distributions succeed in extracting the "square-feature".

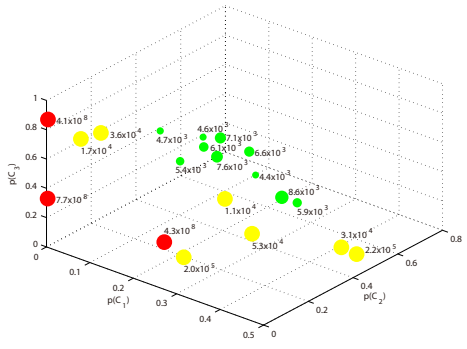


Fig. 7: Networks with various width distributions.

6.3 Adaptivity of features to the depth

In Theorems 1-3, it was proved that deep nets with fixed depth can extract different data features including the sparsity, group structure, and smoothness. In this simulation, we aim to verify this adaptivity of the feature to structures. We are interested in partially radial features defined by

$$f_k(x) = \sum_{j=1}^{k} (x^{(j)})^2 + \sum_{j=k+1}^{10} x^{(j)},$$

where x is generated by uniformly sampling from $[-100, 100]^{10}$.

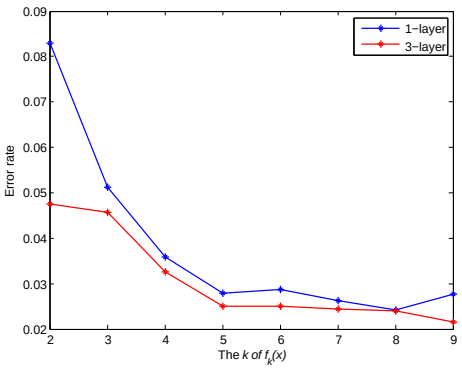


Fig. 8: Adaptivity of the feature to structures.

In the experiment, besides verifying the adaptivity of deep nets, we also compare the performances between deep nets and shallow nets to show the necessity of depth in extracting different data features. As k varies from 2 to 9, the structure of deep nets (3-layer net) is fixed as $50 - 60 - 60$, while the widths in shallow nets are selected according to the test data directly to optimize their performance.

Figure 8 shows the result curves (the detailed numerical results can be found in Appendix E in Supplementary Materials). It can be seen that a deep net with fixed structures performs robustly for dealing with different data features, and always outperforms shallow nets. This demonstrates adaptivity of features to structures. Additionally, we are also aware during the experiment that training shallow nets requires more iterations.

6.4 The role of depth in network

In this subsection, we study the role of depth in extracting the 10-dimensional "square-feature". The simulation setting

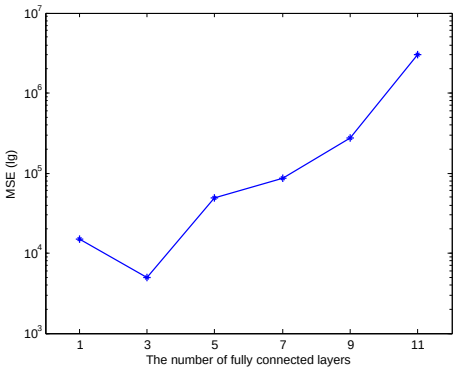


Fig. 9: Best results from networks of different depths.

is the same as that in Subsection 6.2. The only difference is that we select more deep nets with different structures to perceive the impact of depth. There are six candidates for the depth, 1, 3, 5, 7, 9 and 11 and the width is chosen according to the test data directly from much more candidates than those in Subsection 6.2. In particular, the number of neurons in the widest layers of are 4000, 60, 30, 30, 9 and 6, respectively. The MES curve of simulation results is shown in Figure 9 (the detailed numerical results can be found in Appendix E).

It is shown that the depth plays a crucial role in improving the performance of neural networks in feature extraction. We can see that a deep net performs better than a shallow net, but a larger depth does not necessarily lead to better performance. For this simple case (single feature), deep nets with 3 layers are enough. Besides the MSE curve in Figure 9, Appendix E in Supplementary Materials also present similar trends of MAE, MdAE, R^2S and EVS. All these exhibit similar patterns to that of MSE in Figure 9 and verify both the necessity of depth in feature extraction and limitations of deep nets with too many hidden layers.

6.5 Generalization capability verification

In this experiment, in order to test the generalization ability of networks, we train networks with noisy data in a more complex relationship. The underlying relationship between the input signal $x = (x_1, x_2)$ and output is:

$$y = \sin \|x\|_2^2 / \|x\|_2^2 + \varepsilon,$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is the Gaussian noise with the variance of σ^2. The training and test points are generated by i.i.d. sampling 2000 and 200 points on $[-1, 1]$ according to the uniform distribution, respectively, and the noise level is set $\sigma^2 = 0.1$.

TABLE 3: Network width candidates.

Depth	Range of width	Step length
1	$[16, 192]$	16
2	$[32, 64]$	8
3	$[32, 56]$	8
4	$[32, 56]$	8
5	$[16, 40]$	8
6	$[8, 20]$	4

We compare the optimal MSE of deep nets of different depths. The optimal results are obtained by tuning two

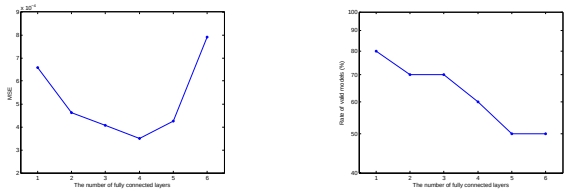
important parameters, the descent step (learning rate during network training) and the width of each layer. During the training process, the descent step changes dynamically as follow,

$$R_d = R_0 * D^{\lfloor (S_g/S_d) \rfloor},$$

where $\lfloor \cdot \rfloor$ is the floor function, R_0 is the initial descent step and R_d is the decayed descent step, D is the decay rate, S_g and S_d are global step and decay step, respectively. Global step represents the current iteration number. Decay step controls the change frequency of descent step. For example, in this experiment, the decay step is 1000, and D is 0.9. The descent step decays to 9% every 1000 iterations. For choosing adequate R_0, we tried values of $0.0001, 0.0005$ and 0.001 on various networks of different depths. Empirically, we notice that a deeper network needs a smaller descent step. Therefore, in the experiment, the descent step is 0.001 for 1, and 0.0005 for 2, 3, 4, 5 and 6-layer networks.

The optimal widths of networks are chosen from a group of candidates, which are set empirically. Table 3 shows the details. For example, for a 3-layer network, the width candidates of each layer are $\{32, 40, 48, 56\}$. As a result, there are 4^3 networks for testing. To alleviate the test burden, in the experiment, we first fix widths of non-middle layers at the medians of the corresponding ranges and test the middle layer width with all the candidates to elect the optimal one, then we tune other layers one by one by testing the candidates around the optimal width of the middle layer.

In Figure 10, we recorded the optimal MSE and the rate of successful convergence of deep nets with different depths. Noting that the function $\sin \|x\|_2^2/\|x\|_2^2$ is smooth and radial, which are difficult for shallow nets to extract them simultaneously, according to the theoretical results in [9]. In our simulation, we show that combining the feature extraction with target-driven learning in deep net is feasible. In fact, a deep net with four layers can significantly improve the performance of shallow nets. Table 4 presents the regression result in terms of MSE, MAE, MdAE, R^2S, and EVS, respectively and exhibits the same pattern as that of MSE in Figure 10.



(a) Accuracy and depth (b) Convergence and depth

Fig. 10: The generalization error result of deep nets.

6.6 Applications for the earthquake seismic intensity prediction

For verifying our theoretical assertions on real applications, in this subsection, we do experiments on earthquake seismic intensity estimations. Earthquake early warning (EEW) systems serve as the tools for coseismic risk reduction. One of the challenges in the development of EEW systems is the accuracy of seismic intensity estimation at the largest possible warning time. Seismic intensity is the intensity or severity of ground shaking at a given location. The level of seismic intensity depends heavily on the distance between the observation site and the epicenter. It can be realized that the level of seismic intensity is a radial function by taking the epicenter as the origin. In the experiments, we test on synthetic data and then deal with a real world dataset.

6.6.1 Synthetic data experiment

The Modified Mercalli intensity scale (MM or MMI), descended from Giuseppe Mercalli's Mercalli intensity scale of 1902, is the most used seismic intensity scale for measuring the intensity of shaking at a given location. It has been a common sense that seismic intensity is an expression of the amplitude, duration and frequency of ground motion. Thus, many attempts have been made to estimate MMI with the ground motion parameters [45]. Fourier amplitude spectrum (FAS) is one of the best features meeting the requirement based on which [46] gives an estimation of MMI as

$$\begin{aligned} MMI = &\exp\{1.2655 + 0.2089\mathcal{M} \\ &- 0.0011d - 0.2451\log(d + 2.1502\mathcal{M})\}, \end{aligned} \tag{20}$$

where d is the estimated Joyner-Boore distance (in kilometers), and $\mathcal{M}$ is the moment magnitude. Figure 11 shows a dense synthetic MMI map generated by (20).

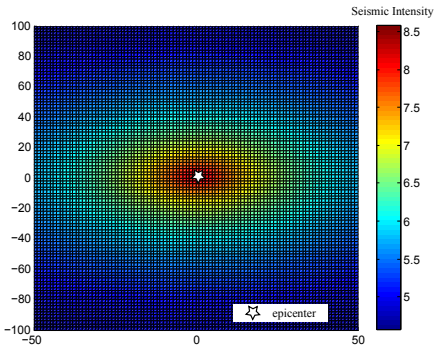


Fig. 11: Dense synthetic MMI map generated by (20).

For network testing, we generate 900 samples according to (20) for training networks of 6 different depths. The testing results are reported in Figure 12. Similar to the experiment in Subsection 6.5, networks perform well and the best result is also given by a 4-layer network.

6.6.2 Real data experiment

For the real data experiment, data are from the U.S. Earthquake Intensity Database[4], which collects damage and felt reports for over 23,000 U.S. earthquakes. The digital database contains information regarding epicentral coordinates, magnitudes, focal depths, names and coordinates of reporting cities (or localities), reported intensities, and the distance from the city (or locality) to the epicenter. Some samples of the data are shown in Figure 13. The input of networks in this experiment are the latitude and longitude coordinates of the site where the earthquake occurred (green box), and the output is seismic intensity (red box). As

4. https://www.ngdc.noaa.gov/seg/hazard /earthqk.shtml

TABLE 4: Noisy data training by networks of various depths.

Depth	1-layer	2-layer	3-layer	4-layer	5-layer	6-layer
MAE	0.0168	0.0191	0.0182	0.0141	0.0184	0.0198
MSE	6.578×10^{-4}	4.613×10^{-4}	4.063×10^{-4}	3.503×10^{-4}	4.483×10^{-4}	7.917×10^{-4}
MdAE	0.0083	0.0065	0.0063	0.0047	0.0087	0.0105
R^2S	0.9752	0.9805	0.9884	0.9991	0.9767	0.09687
EVS	0.9807	0.9824	0.9833	0.9967	0.9834	0.9742

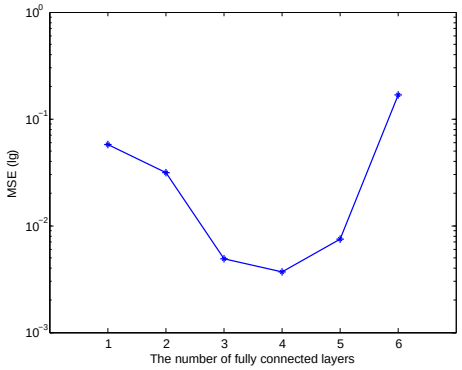


Fig. 12: MSE curve of network during training on synthesis synthetic intensity dataset.

the seismic intensity values in the database are integers in $\{2, \ldots, 6\}$, we consider this task as a classification problem rather than a regression problem.

In order to have enough data for network training, we collect the most data impacted by the same epicenter from the dataset as the experiment data. There are total 608 samples, in which, 500 are used for training and the rest 108 for testing. The parameter tuning strategy is similar to that of Section 6.5. Table 5 shows the comparison results between deep nets of different depths and traditional classification methods, i.e., support vector machine (SVM) and random forest (RF). It is shown that a 5-layer deep net gives the best performance, while networks with other depths cannot compete with SVM.

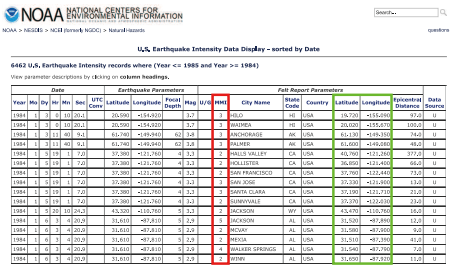


Fig. 13: U.S. Earthquake Intensity Data

7 Conclusion

In this paper, we studied theoretical advantages of deep nets via considering the role of depth in feature extraction and generalization. The main contributions are four folds. Firstly, under the same capacity costs (via covering numbers), we proved that deep nets are better than shallow nets in extracting the group structure features. Secondly, we proved that deep ReLU nets are one of the optimal tools in extracting the smoothness feature. Thirdly, we rigorously proved the adaptivity of features to depths and vice verse, which was adopted to derive the optimal learning rate for implementing empirical risk minimization on deep nets. Finally, we conducted extensive numerical experiments including toy simulations and real data verifications to show the outperformance of deep nets in feature extraction and generalization. All these results presented reasonable explanations for the success of deep learning and provided solid guidance on using deep nets. In this paper, we only consider the depth selection in regression problems. It would be interesting and important to develop similar conclusions for classification. We will consider this topic and report progress in our future study later.

TABLE 5: Comparisons with traditional methods.

Method		Recognition rate
SVM		62.96%
Random forest		58.33%
Deep networks	1-layer	57.41%
	3-layer	60.1%
	5-layer	**66.67%**
	7-layer	62.03%

Acknowledgments

The research of Z. Han and S. Yu was partially supported by the National Natural Science Foundation of China [Grant Nos. 61773367, 61821005], the Youth Innovation Promotion Association of the Chinese Academy of Sciences [Grant 2016183]. The research of S.B. Lin was supported by the National Natural Science Foundation of China [Grant No. 61876133,61977038], and the research of D.X. Zhou was partially supported by the Research Grant Council of Hong Kong [Project No. CityU 11306617] and Hong Kong Institute for Data Science.

References

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

[2] A. Barron. Universal approximation bounds for superpositions of a sigmoidal function. IEEE Trans. Inform. Theory, 1993, 39(3): 930-945.

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intel., 3, 1798-1828, 2013.

[4] M. Bianchini and F. Scarselli. On the complexity of neural network classifiers: a comparison between shallow and deep architectures. IEEE. Trans. Neural Netw. & Learn. Sys., 25: 1553-1565, 2014.

[5] C. M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

[6] J. Bruna and S. Mallat. Invariant scattering convolution networks. IEEE Trans. Pattern Anal. Mach. Intel., 35: 1872-1886, 2013.

[7] C. K. Chui, X. Li, and H. N. Mhaskar. Neural networks for lozalized approximation. Math. Comput., 63: 607-623, 1994.
[8] C. K. Chui, S. B. Lin, and D. X. Zhou. Construction of neural networks for realization of localized deep learning. Front. Appl. Math. Statis., 4: 14, 2018.
[9] C. K. Chui, S. B. Lin, and D. X. Zhou, Deep neural networks for rotation-invariance approximation and learning. Anal. Appl., 17: 737-772, 2019.
[10] F. Cucker and D. X. Zhou. Learning Theory: An Approximation Theory Viewpoint, Cambridge University Press, Cambridge, 2007.
[11] G. Cybenko. Approximation by superpositions of sigmoidal function. Math. Control Signals Syst., 2: 303-314, 1989.
[12] O. Delalleau and Y. Bengio. Shallow vs. deep sum-product networks. NIPs, 666-674, 2011.
[13] D. L. Donoho. Unconditional bases are optimal bases for data compression and for statistical estimation. Appl. Comput. Harmonic Anal., 1(1):100-115, 1993.
[14] R. Eldan and O. Shamir. The power of depth for feedforward neural networks. arXiv preprint arXiv:1512.03965, 2015.
[15] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. Adv. Comput. Math., 13: 1-50, 2000.
[16] I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. MIT Press, 2016.
[17] Z. C. Guo, D. H. Xiang, X. Guo, and D. X. Zhou. Thresholded spectral algorithms for sparse approximations. Anal. Appl. 15: 433–455, 2017.
[18] Z. C. Guo, L. Shi, and S. B. Lin. Realizing data features by deep nets, IEEE Trans. Neural Netw. Learn. Syst., 31: 4036-4048, 2020.
[19] L. Györfy, M. Kohler, A. Krzyzak and H. Walk. A Distribution-Free Theory of Nonparametric Regression. Springer, Berlin, 2002.
[20] M. Hagan, M. Beale, and H. Demuth. Neural Network Design. PWS Publishing Company, Boston, 1996.
[21] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data mining, Inference and Prediction. Springer, New York, 2001.
[22] N. Harvey, C. Liaw, A. Mehrabian. Nearly-tight VC-dimension bounds for piecewise linear neural networks. Conference on Learning Theory. 2017: 1064-1068.
[23] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief netws. Neural Comput., 18:, 1527-1554, 2006.
[24] M. Imaizumi and K. Fukumizu. Deep Neural Networks Learn Non-Smooth Functions Effectively. arXiv preprint arXiv:1802.04474, 2018.
[25] M. Kohler and A. Krzyzak. Nonparametric regression based on hierarchical interaction models. IEEE Trans. Inform. Theory, 63: 1620-1630, 2017.
[26] M. Leshno, V. Y. Lin, A. Pinks, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximat any function. Neural Networks, 6: 861-867, 1993.
[27] H. W. Lin, M. Tegmark and D. Rolnick. Why does deep and cheap learning works so well?. J. Stat. Phys., 168: 1223-1247, 2017.
[28] S. Lin, Y. Rong and Z. Xu. Multivariate Jackson-type inequality for a new type neural network approximation. Appl. Math. Model., 38: 6031-6037, 2014.
[29] S. Lin, J. Zeng, and X. Zhang. Constructive neural network learning. IEEE Trans. Cyber., 49: 221 - 232, 2019.
[30] S. B. Lin. Limitations of shallow nets approximation. Neural Networks, 94: 96-102, 2017.
[31] S. B. Lin. Generalization and expressivity for deep nets. IEEE Trans. Neural Netw. Learn. Syst., 30: 1392-1406, 2018.
[32] S. B. Lin and D. X. Zhou. Distributed kernel-based gradient descent algorithms. Constr. Approx., 47: 249-276, 2018.
[33] T. Lin and H. Zha. Riemannian manifold learning. IEEE Trans. Pattern Anal. Mach. Intel., 30: 796-809, 2008.
[34] V. Maiorov and A. Pinkus. Lower bounds for approximation by MLP neural networks. Neurocomputing, 25: 81-91, 1999.
[35] H. N. Mhaskar. Neural networks for optimal approximation of smooth and analytic functions. Neural Comput., 8: 164-177, 1996.
[36] H. N. Mhaskar and T. Poggio. Deep vs. shallow networks: An approximation theory perspective. Anal. Appl., 14: 829-848, 2016.
[37] G. Montúfar, R. pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep nerual networks. Nips, 2014: 2924-2932.
[38] P. Petersen and F. Voigtlaender. Optimal aproximation of piecewise smooth functions using deep ReLU neural networks. Neural Networks, 108: 296-330, 2018.
[39] A. Pinkus. n-Widths in Approximation Theory. Springer-Velag, Berlin Heidelberg, 1985.
[40] A. Pinkus. Approximation theory of the MLP model in neural networks. Acta Numerica, 8: 143-195, 1999.
[41] I. Safran and O. Shamir. Depth-width tradeoffs in approximating natural functions with neural networks. arXiv reprint arXiv:1610.09887v2, 2016.
[42] C. Satriano, Y. M. Wu, A. Zollo, and H. Kanamori. Earthquake early warning: Concepts, methods and physical grounds. Soil Dynamics Earth. Engineer., 31: 106-118, 2011.
[43] C. Schwab and J. Zech. Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ. Anal. Appl., 17: 19-55, 2019.
[44] U. Shaham, A. Cloninger, and R. R. Coifman. Provable approximation properties for deep neural networks. Appl. Comput. Harmonic Anal., 44: 537-557, 2018.
[45] V. Y. Sokolov and Y. K. Chernov. On the correlation of seismic intensity with Fourier amplitude spectra. Earthquake Spectra., 14: 679-694, 1998.
[46] V. Y. Sokolov. Seismic intensity and Fourier acceleration spectra: revised relationship. Earthquake Spectra., 18: 161-187, 2002.
[47] K. Vikraman. A deep neural network to identify foreshocks in real time. arXiv preprint arXiv:1611.08655, 2016.
[48] D. Yarotsky. Error bounds for aproximations with deep ReLU networks. Neural Networks, 94: 103-114, 2017.
[49] Y. Ying and D. X. Zhou. Unregularized online learning algorithms with general loss functions. Appl. Comput. Harmonic Anal., 42: 224-244, 2017.
[50] D. X. Zhou. Capacity of reproducing kernel spaces in learning theory. IEEE Trans. Inform. Theory, 49: 1743-1752, 2003.
[51] D. X. Zhou. Deep distributed convolutional neural networks: Universality. Anal. Appl., 16: 895-919, 2018.
[52] D. X. Zhou. Universality of Deep Convolutional Neural Networks. Appl. Comput. Harmonic. Anal., 48: 784-794, 2020
[53] D. X. Zhou. Theory of deep convolutional neural networks: Downsampling. Neural Networks, 124: 319-327, 2020.

Zhi Han received his B.Sc., M.Sc., and Ph.D. degrees in applied mathematics from Xi'an Jiaotong University (XJTU), China, in 2005, 2007 and 2012, respectively. From 2009 to 2011, he was a joint Ph.D. candidate of statistics at the University of California, Los Angeles (UCLA), USA. He is currently a professor at the State Key Laboratory of Robotics in Shenyang Institute of Automation, Chinese Academy of Sciences (SIA, CAS). His research interests include image/video modeling, low-rank matrix recovery and deep neural networks.

Siquan Yu received the B.Sc. and M.Sc. degrees in automatic control from Liaoning Shihua University, China, in 2011 and 2015, respectively. He is currently pursuing the joint Ph.D. degree with Northeastern University, China, and State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences. His research interests include image/video representation and deep learning.

Shao-Bo Lin received the Ph. D. degree in Applied Mathematics in 2014 from Xi'an Jiaotong University, China. He was with the Department of Mathematics, Wenzhou University from 2014 to 2019. He is currently a professor in School of Management, Xi'an Jiaotong University. His research interests include deep learning theory and distributed learning.

Ding-Xuan Zhou received the B.Sc. and Ph.D. degrees in mathematics from Zhejiang University, Hangzhou, China, in 1988 and 1991, respectively. He joined the faculty of City University of Hong Kong in 1996, where he is currently a Chair Professor in the School of Data Science and Department of Mathematics, and Director of Liu Bie Ju Centre for Mathematical Sciences. He has authored over 100 research papers. His current research interests include deep learning, learning theory, data science, wavelet analysis, and approximation theory.

Prof. Zhou is serving on the Editorial Board of over 10 international journals, and is Editor-in-Chief of the journal "Analysis and Application". He received a National Science Fund of China for Distinguished Young Scholars in 2005 and a Humboldt Research Fellowship in 1993, and was rated in 2014, 2015, 2016 and 2017 by Thomson Reuters/Clarivate Analytics as a Highly-cited Researcher. He has co-organized over 20 international conferences and conducted over 20 research grants.