THE RATE OF CONVERGENCE OF VARIATION-CONSTRAINED
DEEP NEURAL NETWORKS

BY GEN LI\textsuperscript{1,}\textasteriskcentered*, YUANTAO GU\textsuperscript{1,†} AND JIE DING\textsuperscript{2}

\textsuperscript{1}Department of Electronic Engineering, Tsinghua University \textasteriskcentered*g-li16@mails.tsinghua.edu.cn; \textasteriskcentered†gyt@tsinghua.edu.cn

\textsuperscript{2}School of Statistics, University of Minnesota, jding@umn.edu

Multi-layer feedforward networks have been used to approximate a wide
range of nonlinear functions. An important and fundamental problem is to un-
derstand the learnability of a network model through its statistical risk, or the
expected prediction error on future data. Existing work have shown that the
rate of convergence of neural networks is bounded by the order of \(n^{-1/4}\) for
a sample size of \(n\). In this paper, we show that a class of variation-constrained
neural networks can achieve near-parametric rate \(n^{-1/2+\delta}\) for an arbitrarily
small positive constant \(\delta\). It is equivalent to \(n^{-1+2\delta}\) under the mean squared
error. The somewhat surprising result indicates that the neural function space
needed for approximating smooth functions may not be as large as what is of-
ten perceived. Our result also provides insight to the mysterious phenomena
that deep neural networks do not easily suffer from overfitting when the num-
ber of neurons, layers, and learning parameters rapidly grow with \(n\) or even
surpass \(n\). We also discuss the rate of convergence regarding other network
parameters, including the input dimension, network width, and coefficient
norm.

1. Introduction. Suppose we have \(n\) labeled observations \(\{\(x_i, y_i\)\}_{i=1,...,n}\), where \(y_i\)'s
are continuously-valued responses or labels. We assume that the underlying data generating
model is

\[ y_i = f_\text{\textsuperscript{*}}(x_i) + \varepsilon_i \]

for some unknown function \(f_\text{\textsuperscript{*}}(\cdot)\), where \(x_i\)'s \(\in \mathcal{X} \subset \mathbb{R}^d\) are independent and identically
distributed (IID), and \(\varepsilon_i\)'s are IID. We will measures the predictive performance of a learned

\textsuperscript{*}MSC 2010 subject classifications: Primary 62J02; secondary 62G05

\textsuperscript{†}Keywords and phrases: Generalization error, Model complexity, Model selection, Neural Network, Regular-
ization, Statistical risk
neural networks model $f$ using the $\ell_2$ statistical risk

$$\mathcal{R}_2(f) \overset{\Delta}{=} \sqrt{\mathbb{E}(f(x) - f^*(x))^2},$$

(1)

where $x$ denotes an observation independent with the data used to train $f$. The smaller $\mathcal{R}_2(f)$, the better. We will also study $\ell_1$ risks, which will be introduced in Subsection 2.6. Our work addresses the following question.

*How does a multi-layer feedforward network’s predictive performance relates to the sample size used for training?*

1.1. **Background of deep neural networks (DNNs).** Neural networks have been successfully applied to modeling nonlinear regression functions or classification decision boundaries in various applications. Despite its massive success in practical applications, a systematic understanding of its theoretical limit remains an ongoing challenge and has motivated research from various perspectives. It was shown in (Cybenko, 1989) that any continuous function can be approximated arbitrarily well by a two-layer perceptron with sigmoid activation functions. An approximation error bound of using two-layer neural networks to fit arbitrary smooth functions was established in (Barron, 1993, 1994), where statistical risk bounds were also developed. A theoretical connection between deep networks and approximation theory based on spline functions was established in (Baraniuk and Balestriero, 2018). A dimension-free Rademacher complexity for deep ReLU neural networks was recently developed (Golowich, Rakhlin and Shamir, 2017; Barron and Klusowski, 2019). A dimension-free statistical risk for two-layer neural networks was studied in (Li, Gu and Ding, 2020) using a combined analysis of Rademacher complexity and the number of neurons. Based on a contraction lemma, a series of norm-based complexities and their corresponding generalization errors were developed (Neyshabur, Tomioka and Srebro, 2015, and the references therein). The neural network learning problem was also cast as a tensor decomposition problem through the score function of the known or estimated input distribution (Janzamin, Sedghi and Anandkumar, 2015; Ge, Lee and Ma, 2017; Mondelli and Montanari, 2018). Most recently, tight error bounds have been established for deep neural networks that do not suffer from the curse of dimensionality, based on the assumption that the data are generated by a neural network model of a parsimonious structure. In that direction, the work of (Schmidt-Hieber, 2017) proved that specific deep neural networks with few non-zero network parameters could achieve minimax
rates of convergence. In (Bauer and Kohler, 2019), an error bound that is free from the input dimension was developed for a class of generalized hierarchical interaction models.

1.2. Related results on nonparametric regression. The $\ell_1$ or $\ell_2$ risks of typical parametric models such as finite-dimensional linear regressions are at the order of $(d/n)^{1/2}$, where $d$ and $n$ denote the number of coefficients (or ‘input dimension’) and the sample size, respectively. It is known that the minimax rate of convergence for smoothness function classes (e.g., Sobolev and Besov) is often at the order of $O(n^{-\alpha/(d+2\alpha)})$, where $\alpha$ is the order of smoothness (defined by the largest order of derivatives) and $d$ is the input data dimension. Obtaining the risk bound for general nonparametric regression models such as neural networks is highly nontrivial since it involves an approximation error (or bias) term and a generalization error (or variance) term that are both nontrivial. The work of (Barron, 1993, 1994) proved that the model class of two-layer feedforward networks has an approximation error bound of $O(r^{-1/2})$ when approximating a particular class of smooth functions, where $r$ denotes the number of neurons. The same work further developed a statistical risk error bound of $O((d/n)^{1/4})$, which is the tightest statistical risk bound for two-layer neural networks up to the authors’ knowledge (for $d < n$). Analysis of the approximation error of more general neural networks beyond two-layer networks remains an open problem. To bound the generalization error term, it is tempting to treat all the neural weights as free parameters and use the Akaike information criterion-type statistical error bounds (Ding, Tarokh and Yang, 2018). Those type of bounds is often derived from second-order Taylor expansion in conjunction with some regular conditions (see, e.g., Ding et al., 2020). However, the typical regular conditions such as non-degenerate Fisher information matrix and separated model conditions do not apply to neural networks. This observation has motivated the use of alternative complexity measures, e.g., the Rademacher complexity (Golowich, Rakhlin and Shamir, 2017), beyond those based on the number of unknown parameters.

1.3. Main contributions. We show that variation-constrained deep neural networks can achieve near-parametric rate $n^{-1/2+\delta}$, for an arbitrarily small positive constant $\delta$. Here, the variation means the smallest norm of neural weights needed to construct the underlying data generating function. The result has the following implications. First, to learn smooth regression functions, where the smoothness is indexed by its variation, the class of deep neural
networks can enjoy a statistical efficiency at the same order of classical parametric models (namely around $n^{-1/2}$). Second, the neural function space needed for approximating smooth functions may not be as large as what is often perceived. In particular, we will show that the minimax rate of the statistical risk is at least $n^{-1/2}$, which implies that our derived bound is rather tight. Third, a neural network’s predictive performance does not necessarily depend on its ‘nominal complexity’ as described by its number of neurons and layers. Instead, it only depends on the variation of the underlying function and $n$. To some extent, this explains the mysterious phenomena that deep neural networks do not easily suffer from overfitting when the number of unknown parameters is huge compared with $n$.

From the technical perspective, we derived the risk analysis from a norm-based approach, which is different from earlier work that used the number of neurons $r$ to characterize bias-variance tradeoffs. In particular, the risk bound of $O((d/n)^{1/4})$ in (Barron, 1994) is based on an appropriate choice of $r$, but our tighter bound $O((d/n)^{1/2})$ is based on an analysis of the neural weights regardless of $r$. Our analysis was inspired by but different from the norm-based complexity analysis (Neyshabur, Tomioka and Srebro, 2015; Barron and Klusowski, 2019). In particular, the standard technical tool of contraction lemma used to derive Rademacher complexity cannot apply to the $\ell_2$ statistical risk studied in our context. Our proof is based on new technical tools developed in this paper.

Apart from the statistical risk based on the $\ell_2$ loss, we will analyze the $\ell_1$ loss, which also ranks among the most popular loss functions in practice. We will show that a similar convergence rate can be derived, and the dependence on the input dimension can be removed. To the best of the authors’ knowledge, this is the first result regarding the $\ell_1$ statistical risk for deep neural networks.

The outline of the paper is given below. In Section 2, we introduce the formulation and some notations. In Section 3, we introduce the main result and make several remarks to illustrate the implications better. In Section 4, we introduce counterpart results when the typical $\ell_2$ loss is replaced with $\ell_1$ loss and show how the bound as a function of the norm may be further tightened. We conclude the work in Section 5.

2. Problem Formulation.
2.1. Notation. Throughout the paper, we use $n$ and $d$ to denote the sample size and the number of variables (or input dimension), respectively. We write $a_n \gtrsim b_n$, $b_n \lesssim a_n$, or $b_n = O(a_n)$, if $|b_n/a_n| < c$ for some constant $c$ for all sufficiently large $n$. Let $N(\mu, V)$ denote Gaussian distribution with mean $\mu$ and covariance $V$. Let $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the common $\ell_1$ and $\ell_2$ vector norms, respectively. For any vector $z \in \mathbb{R}^d$ and set $X$, we define $\|z\|_X \triangleq \sup_{x \in X} |x^\top z|$, which may or may not be infinity. If $X = \{x : \|x\|_\infty \leq M\}$, Throughout the paper, $E$ denotes the expectation for the underlying data generating distribution. The notation for neural network parameters is summarized in Subsection 2.3.

2.2. Data generating process. Suppose we have $n$ observations $\{(x_i, y_i)\}_{i=1}^n$, where $y_i$’s are continuously-valued responses or labels. We assume that the data are independently generated from the underlying data generating process

$$y_i = f_\ast(x_i) + \varepsilon_i$$

where $f_\ast(\cdot)$ is an unknown function. We will use a neural network to approximate the functionality of $f_\ast(\cdot)$. We also assume that $x_i$’s $\in X$ are IID variables, where $X \subset \mathbb{R}^d$ is a bounded set that contains zero, and $\varepsilon_i$’s are IID noises independent with $x_i$. Moreover, we assume the following conditions. It is satisfied if $\varepsilon_i$ follows a sub-Gaussian or sub-exponential distribution.

**Assumption 1.** The noise terms satisfy $E(\varepsilon_i^2) \leq \tau^2$ and $E \max_{1 \leq i \leq n} |\varepsilon_i| \lesssim \tau \log n$.

2.3. Deep neural network model class. Recall that our goal is learn a regression function $\hat{f}_n : x \mapsto \hat{f}_n(x)$ for prediction. In practice, $\hat{f}_n$ is estimated from a pre-specified regression model class, such as linear regression and nonparametric regression based on series expansion with polynomials, splines, or wavelets bases. We consider the class of multi-layer feed-forward neural networks (also illustrated in Figure 1), denoted by $\mathcal{F}_L$.

The neural network consists of $L$-layers, indexed by $l = 1, \ldots, L$, each containing $r_l$ neurons/nodes. Specially, the $L$-th layer is the the output layer, which outputs the regression result $y$ (so $r_L = 1$). Also, we may treat the input layer as layer $l = 0$ (so $r_0 = d$). Let $y_l \in \mathbb{R}^{r_l}$ denote the output of $l$-th layer, which is also the input of $l + 1$-th layer, for $l = 0, \ldots, L - 1$. In the $\ell$th layer ($\ell = 1, \ldots, L - 1$), the $i$th neuron takes a linear combination of the previous layer’s output, with weights (or coefficients) $w_{i,l} \in \mathbb{R}^{r_{l-1}}$, and passes it through a nonlin-
ear activation function $\sigma(\cdot)$. In our analysis, $r_1, \ldots, r_{L-1}$ are allowed to be arbitrarily large, meaning that the network may have arbitrarily many hidden neurons.

Suppose that the input data dimension is $x \in \mathbb{R}^d$. By our notation, we have $y_0 = x$ (output of the 0th layer), $y_{\ell,i} = \sigma(w_{\ell,i}^T y_{\ell-1})$ (output of the $\ell$th layer, $1 \leq \ell \leq L - 1$, and $y = y_L = w_L^T y_{L-1}$ (the final output). Without loss of generality, we do not separately consider the bias term in each neural network layer, because it can be absorbed into the neural weight. Specifically, we note that $y_{\ell,i} = \sigma(w_{\ell,i}^T y_{\ell-1} + b_{\ell,i})$ can be rewritten as $y_{\ell,i} = \sigma(\tilde{w}_{\ell,i}^T \tilde{y}_{\ell-1})$, where $\tilde{w}_{\ell,i} = [w_{\ell,i}, b_{\ell,i}/\sigma(0)]$ and $\tilde{y}_{\ell-1} = [y_{\ell-1}, \sigma(0)]$. Thus, we can simply realize $\sigma(0)$ by adding all-zero weights to the corresponding neuron, and the same proof will apply. With a slight abuse of notation, we used $y_j$ for the output of the layer $j$, while $y_i$ for the $i$th regression response. We will use $w_{\ell,i,j}$ to denote the $j$th entry of the vector $w_{\ell,i}$.

2.4. Activation function. We consider the following class of nonlinear activation function $\sigma(\cdot)$ for the technical analysis. A specific case is the popular activation function $\sigma(x) = 1/(1 + \exp(-x))$, also known as the logistic function. We note that some other popular activation functions, e.g., the rectifier function, do not satisfy the assumption.

Assumption 2. The activation function $\sigma(\cdot)$ is a bounded function on $\mathbb{R}$ satisfying $\sigma(z) \to 1$ as $z \to \infty$ and $\sigma(z) \to 0$ as $z \to -\infty$. Also, its derivatives satisfy $\sup_{z \in \mathbb{R}} |\sigma^{(k)}(z)| \leq M$, for any positive integer $k$ and a constant $M$.

2.5. Variation-constrained networks. Recall that $w_{\ell,i}$ denotes the neural weights for the $i$th neuron at the $\ell$th layer. For any positive integer $L$ and real value $v$, let $F_L(v)$ denote

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig1}
\caption{A graph showing multi-layer feedforward neural networks.}
\end{figure}
the set of all the $L$-layer feedforward neural networks (defined in Subsection 2.3) satisfying $\|w_{L,i}\|_1 \leq v$ for all $i, \ell$. By the above notation, $\mathcal{F}_L$ may be written as $\mathcal{F}_L(\infty)$. The existing result that any continuous function can be approximated arbitrarily well by a two-layer perceptron with sigmoid activation functions (Cybenko, 1989, and the references therein) indicates that the closure of $\{f : f \in \mathcal{F}_2(v), v \geq 0\}$ contains all the continuous functions. It is thus a quite expressive model class. Though there does not seem to exist a theory on the size for $\mathcal{F}_L(v)$ with a general $L$, it is conceivable that its expressive power is not smaller than $\mathcal{F}_2(v')$ for some $v'$ that depends on $v$. As a result, any smooth function can be expressed by a neural network with sufficiently many layers and neurons.

From now on, we suppose that the number of layers $L$ is fixed. For any function $f_*$, we define its variation to be

$$V(f_*) := \inf \{v : f_* \in \mathcal{F}_L(v)\} + 1,$$

Intuitively, $V(f_*)$ characterize the difficulty of learning $f_*$. The value of $V(f_*)$ is small for a smooth $f_*$ (e.g., a sigmoid function), and large for a bumpy $f_*$. We note that $V(f_*)$ can be infinity even if $f_*$ is a continuous function, since $f_*$ may not be in the interior of the closure of $\{f : f \in \mathcal{F}_L(v), v \geq 0\}$. In the sequel, we mainly consider the case $V(f_*) < +\infty$, and we will discuss the case $V(f_*) = +\infty$.

2.6. Training and evaluation. We learn the neural network from $n$ observations by solving the empirical risk minimizing problem

$$\hat{f}_n = \arg \min_{f \in \mathcal{F}_L(V)} \frac{1}{n} \sum_{i=1}^{n} \text{loss}(f(x_i), y_i)$$

for some loss function $\text{loss} \cdot$, where $V$ is a chosen constant that constrains the variation of $\hat{f}_n$. In practice, the above optimization can be operated by the alternative problem

$$\hat{f}_n = \arg \min_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \text{loss}(f(x_i), y_i) + \Omega(f) \right\},$$

where $\mathcal{F}$ is the unconstrained class of neural networks and $\Omega(\cdot)$ is an appropriately chosen regularization function. We will revisit the choice of $\Omega$ in later sections.

In the problem (2), the loss function $\text{loss}(\cdot)$ is usually pre-determined to be the $\ell_2$ loss defined by $\ell_2(y', y) = (y' - y)^2$. Correspondingly, the predictive performance of $\hat{f}_n$ is often evaluated by

$$\mathbb{E} \left( \text{loss}(\hat{f}_n(x, \tilde{y})) \right) = \mathbb{E} \left( \text{loss}(\hat{f}_n(x, \tilde{y}))^2 \right).$$
where \((\tilde{x}, \tilde{y})\) denotes a future observation independent with the data used to train \(\hat{f}_n\). Removing the constant \(\sigma^2\) and taking a square root for (4), we obtain the \(\ell_2\) risk introduced in (1). Another loss, the \(\ell_1\) loss defined by loss\((y', y) = |y' - y|\), is also used in practice. The training objective and the evaluation metric can be similarly defined. We will focus on the \(\ell_2\) loss in Section 3, and discuss the counterpart results for the \(\ell_1\) loss in Section 4.

3. Main Result. Suppose that we use \(\ell_2\) loss to train, namely

\[
\hat{f}_n = \arg\min_{f \in F_n} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2.
\]

THEOREM 3.1. Under Assumptions 1 and 2, for an arbitrarily small constant \(\delta > 0\) and any constant \(V \geq \max\{V(f_\star), 1\}\), the \(\hat{f}_n\) in (5) satisfies

\[
\mathcal{R}_2(\hat{f}_n)^2 = \mathbb{E} |\hat{f}_n(x) - f_\star(x)|^2 \lesssim V^{2+\delta+(L-1)d} n^{-1+2\delta}.
\]

REMARK 1 (Rate as a function of \(n\)). The above result shows that the \(\ell_2\) risk \(\mathcal{R}_2(\hat{f}_n)^2\) for the variation-constrained deep neural network class can achieve nearly parametric convergence rates \(n^{-1/2}\). The result implies that for multi-layer neural networks with fixed variation and input dimension, the statistical risk decays with the sample size in a way similar to parametric models. The implication is appealing in practice when we have a massive amount of data for learning an underlying function that is not so bumpy.

Though our convergence rate is tight only for fixed \(d\) and \(L\), its dependence \(n\) is highly nontrivial to derive, even for a small input dimension such as \(d = 1\) or \(d = 2\). In terms of the dependence on \(n\), our rate of convergence is much tighter than the existing results of \(O(n^{-1/4})\) in various settings (see Section 1). In fact, we will show in Proposition 3.5 that the \(n^{-1/2}\) is the best rate one could expect for neural networks. We will relax the dependence on \(d\) in Section 4.

Another interesting implication of Theorem 3.1 is that the predictive performance of a class of neural networks does not necessarily depend on their ‘nominal complexity’ as described by the number of neurons and layers. Instead, it only depends on the variation of the underlying function and \(n\). The result indicates that deep neural networks may not suffer from overfitting when the network model is excessively complex, a striking observation made in many application studies.
Figure 2 shows an experiment result that initially motivated our study. We generated $n$ samples from $y = \beta_1 x_1 + \cdots + \beta_5 x_5 + \epsilon$, where $x_1, \ldots, x_5, \epsilon$ followed the IID standard Gaussian, and $n = 2^5, 2^6, \ldots, 2^{11}$. We used a $[5, 50, 10, 1]$-neural network, which consists of an input layer, two hidden layers with 50 and 10 neurons, and an output layer, and implemented the training using Pytorch (Ketkar, 2017). We recorded the squared $\ell_2$ risk (also named the mean squared error) from 50 independent replications, using an independent test data with a size of $10^4$. To maintain a fixed variation, we approximately control the total $\ell_1$ norm of the parameters (around 250) by applying $\ell_1$-regularizations to the neural weights during the training.

Remark 2 (No ‘free lunch’). The statistical risk of deep neural networks, as a non-parametric regression method, depends on many factors, including the sample size, the input dimension, and the smoothness of the underlying function. Though Theorem 3.1 gives a rate of convergence much faster than the typical nonparametric rates, a result not previously recognized even for a fixed $d$, it is derived by assuming a bounded variation $V$. Without variation constraint, the rate could still be much slower than the parametric rate. Thus, Theorem 3.1 does not conflict with the usual impression that neural networks may sacrifice the convergence rate for expressive power.
Specifically, recall that the set of continuous functions can be stratified by sets in the form of $F_L(V)$ ($V \geq 0$), and that $V \geq V(f_*)$ implies that $f_* \in F_L(V)$. Theorem 3.1 implies that only if $f_*$ can be expressed by a $V$-variation neural network, the predictive performance of the learned $\hat{f}_n$ will improve in a similar manner as a parametric model as $n$ becomes large. The result does not hold if $f_*$ has a large (or infinite) variation. This will be further discussed in the next remark.

REMARK 3 (A new perspective of bias-variance tradeoffs in nonparametric regression). The tight statistical risk rate (regarding $n$) is at the cost of restricting the value of variation. Recall that the variation of the underlying data-generating function means the smallest norm of neural weights needed to construct it. A general function $f_*$ may require a large or even infinite variation. Though a large variation supports more regression functions with desirable approximation errors, it will enlarge the estimation variance and thus degrade the overall predictive performance. This naturally motivates the bias-variance tradeoff from a variation perspective.

We define $T = V^L$, which represents the total variation of the network. We introduce

$$\beta(f_*) = \lim_{V \to \infty} \inf_{V \geq V(f_*)} \frac{-\log \min_{f \in F_L(V)} \mathbb{E} \| f - f_* \|^2}{\log T},$$

which is well defined even for $V(f_*) = +\infty$. An interpretation of $\beta(f_*)$ is that $f_*$ can be approximated by a neural network in $F_L(V)$ with mean squared error no larger than $T^{-2\beta}$, for any constant $\beta < \beta(f_*)$. Then, Theorem 3.1 implies that

$$\mathbb{E} |\hat{f}_n(x) - f_*(x)|^2 \lesssim V^{2+\delta}(T/V)^d n^{-1+2\delta} + T^{-2\beta}$$

for any positive constant $\delta$. It can be easily verified that if $\log n \lesssim V^{d-2}$, we have

$$\mathcal{R}_2(\hat{f}_n) = \sqrt{\mathbb{E} |\hat{f}_n(x) - f_*(x)|^2} = O(n^{-\frac{\delta}{d+2\delta}}),$$

for any positive constant $\beta < \beta(f_*)$. The risk bound in (7) is a consequence of bias-variance tradeoffs regarding the choice of $T$ (and thus $V$). We note that (7) resembles the classical result of $O(n^{-\alpha/(d+2\alpha)})$ for the $\ell_2$ risk of smoothness function classes (Subsection 1.2). A comprehensive analysis of the above tradeoffs for different neural structures is beyond the scope of this paper and is left as future work.

PROOF. The proof relies on an $\epsilon$-net argument. The $L_\infty(X)$ metric of $F(V)$ refers to:

$$\|f_1 - f_2\|_{L_\infty(X)} = \max_{x \in X} |f_1(x) - f_2(x)|.$$

(8)
We use $\mathcal{N}_\epsilon$ to denote an $\epsilon$-covering of $\mathcal{F}(V)$ under the above $L_\infty(X)$ metric. The proof will be based on the following two key observations.

• First, the $\epsilon$-covering is dense enough.

**Lemma 3.2 (Proved in the Appendix A.1).** If $\|f - \hat{f}_n\|_{L_\infty(X)} \leq \epsilon$, then

$$\mathbb{E} \left( (\hat{f}_n(x) - f(x))^2 - (f(x) - f_*(x))^2 \right) \leq 4\epsilon,$$

and

$$\frac{1}{n} \sum_{i=1}^{n} \left[ (f(x_i) - f_*(x_i) - \epsilon_i)^2 - (\hat{f}_n(x_i) - f_*(x_i) - \epsilon_i)^2 \right] \leq 4(V + \max_i |\epsilon_i|)\epsilon.$$

• Second, there exists a finite $\epsilon$-covering.

**Lemma 3.3 (Proved in the Appendix A.2).** The covering number of $\mathcal{F}(V)$ satisfies

$$\log N(\epsilon) \leq \nu V^{d(L-1)+\delta} e^{-\delta},$$

where $\nu$ is a constant that only depends on $d$ and $\delta$.

With the above two lemmas, we first make the following decomposition.

$$\mathbb{E} (\hat{f}_n(x) - f_*(x))^2 = \mathbb{E} (\hat{f}_n(x) - f_*(x))^2 - \mathbb{E} (f(x) - f_*(x))^2$$

$$+ \mathbb{E} (f(x) - f_*(x))^2 - \frac{1}{n} \sum_{i=1}^{n} ([f(x_i) - f_*(x_i) - \epsilon_i]^2 - \epsilon_i^2)$$

$$+ \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - f_*(x_i) - \epsilon_i)^2 - \frac{1}{n} \sum_{i=1}^{n} (\hat{f}_n(x_i) - f_*(x_i) - \epsilon_i)^2$$

$$+ \frac{1}{n} \sum_{i=1}^{n} ([\hat{f}_n(x_i) - f_*(x_i) - \epsilon_i]^2 - \epsilon_i^2),$$

where $f \in \mathcal{N}_\epsilon$ satisfies

$$\|f - \hat{f}_n\|_{L_2(\mu)} \leq \epsilon.$$

Next, we bound the four terms in (11) separately.

• According to Lemma 3.2, the first term can be bounded by $4V\epsilon$.

• The second term is bounded by applying Bernstein’s inequality and the union bound on $\mathcal{N}_\epsilon$, which is formalized by the following lemma.
Lemma 3.4 (Proved in the Appendix A.3). For any given $\rho > 0$

$$
\mathbb{E}(f(x) - f_*(x))^2 - \frac{1}{n} \sum_{i=1}^{n} [(f(x_i) - f_*(x_i) - \xi_i)^2 - \xi_i^2]
$$

(13)

$$
\lesssim \frac{V(V + \max_i |\xi_i|)}{n} \log \frac{N(\epsilon)}{\rho}.
$$

holds uniformly for all $f \in \mathcal{N}_\epsilon$ with probability at least $1 - \rho$.

- Lemma 3.2 indicates that the third term can be bounded by $4(V + \max_i |\xi_i|)\epsilon$.
- As for the fourth term, according to the definition of $\hat{f}_n$, we have

$$
\frac{1}{n} \sum_{i=1}^{n} [(\hat{f}_n(x_i) - f_*(x_i) - \xi_i)^2 - \xi_i^2] \leq \frac{1}{n} \sum_{i=1}^{n} \xi_i^2,
$$

which implies that

$$
\frac{1}{n} \sum_{i=1}^{n} [(\hat{f}_n(x_i) - f_*(x_i) - \xi_i)^2 - \xi_i^2] \leq 0.
$$

Therefore, we have

(14) \hspace{1cm} \mathbb{E}(\hat{f}_n(x) - f_*(x))^2 \lesssim V\epsilon + \frac{V(V + \max_i |\xi_i|)}{n} \log \frac{N(\epsilon)}{\rho} + (V + \max_i |\xi_i|)\epsilon
$$

uniformly for all $f \in \mathcal{N}_\epsilon$ with probability at least $1 - \rho$.

Let $E_n$ denote the event that (14) occurs for all $f \in \mathcal{N}_\epsilon$. Then $\mathbb{P}(E_n^c) \leq \rho$. Recall that $\mathbb{E} \max_i |\xi_i| \lesssim \tau \log n$ (Assumption 1). Taking $\rho = 1/n$ and invoking (14), we have and for all $f \in \mathcal{N}_\epsilon$,

$$
\mathbb{E}(\hat{f}_n(x) - f_*(x))^2 = \mathbb{E}\left\{ (\hat{f}_n(x) - f_*(x))^2 1_{E_n} + (\hat{f}_n(x) - f_*(x))^2 1_{E_n^c} \right\}
$$

$$
\lesssim \frac{V(V + \tau \log n)}{n} \log \frac{N(\epsilon)}{\rho} + (V + \tau \log n)\epsilon + V^2 \mathbb{P}(E_n^c),
$$

$$
\lesssim \frac{V(V + \tau \log n)}{n} \log (nN(\epsilon)) + (V + \tau \log n)\epsilon.
$$

By using the upper bounds for $N(\epsilon)$ in (10) and letting $\epsilon = 1/n$, we obtain

(15) \hspace{1cm} \mathcal{R}_2(\hat{f}_n)^2 = \mathbb{E}(\hat{f}_n(x) - f_*(x))^2 \lesssim V^{1+\delta + (L-1)d}(V + \tau \log n)n^{-1+\delta}.
$$

The right-hand side in (15) is further upper bounded by $V^{2+\delta + (L-1)d} n^{-1+2\delta}$, which concludes the proof.

At the end of this section, we show that the minimax risk lower bounds for the neural network class $\mathcal{F}_2(V)$ is at the order of $n^{-1/2}$. The result implies that the upper bound in Theorem 3.1 is rather tight. We need the following additional assumption.
ASSUMPTION 3. There exists a constant $c > 0$ and a bounded subset $S \subset \mathbb{R}$ such that
\[
P(Z \in S) > c \quad \text{and} \quad \inf_{z \in S} \sigma'(z) > c \quad \text{for} \quad Z \sim \mathcal{N}(0, 1).
\]

PROPOSITION 3.5 (Minimax risk lower bound). Suppose that Assumptions 2 and 3 hold, and $x_1, x_2, \ldots, x_n \overset{iid}{\sim} \mathcal{N}(0, I_d)$ (with the probability measure denoted by $\mu$). Then, for $q = 1, 2,$
\[
\inf_{\hat{f}_n} \sup_{f \in F_2(V)} \| \hat{f}_n - f \|_{L_q(\mu)} \gtrsim V \sqrt{\frac{d}{n}}.
\]

PROOF. The proof is postponed to Appendix C. \hfill \square

4. Result under the $\ell_1$ Risk and Improvement of the Dependence on $V$. Suppose that we use the $\ell_1$ loss for training, namely
\[
\hat{f}_n = \arg \min_{f \in F_1(V)} \frac{1}{n} \sum_{i=1}^n |f(x_i) - y_i|.
\]
Correspondingly, we evaluate the predictive performance of a learned regression function $f$ using the $\ell_1$-statistical risk
\[
\mathcal{R}_1(f) = \mathbb{E} |f(\mathbf{x}) - y| - \mathbb{E} |\varepsilon|.
\]
It can be verified that $\mathcal{R}_1(f)$ is nonnegative for symmetric random variables $\varepsilon$. We note that the $\ell_2$ risk is equivalent to the squared $\ell_2(\mu)$ metric plus some constant (see (4)). But minimizing the $\ell_1$ risk does not necessarily minimize the $\ell_1(\mu)$ metric. We are interested in the $\ell_1$-based training and evaluation mainly because they are practically used in many learning problems, e.g., those for heterogeneous noises (Welsch, 1977), ordinal data (Pedregosa, Bach and Gramfort, 2017), and imaging data (Zhao et al., 2016). From a practical point of view, using $\ell_1$ loss for training is as easy and fast as using $\ell_2$ in prevalent computational frameworks such as Tensorflow (Google, 2016), Pytorch (Ketkar, 2017), and Keras (Gulli and Pal, 2017).

From the theoretical perspective, interestingly, we found that the $\ell_1$ loss provides technical convenience for deriving tight risk bounds. In particular, we will show that the bound in Theorem 3.1 can be improved in terms of the dependence on the input dimension $d$.

THEOREM 4.1. Under Assumption 2, for any constant $V$ with $V \geq V(f_*)$, the $\hat{f}_n$ in (17) satisfies
\[
\mathcal{R}_1(\hat{f}_n) \lesssim \frac{V \sqrt{\log n + \tau}}{\sqrt{n}}.
\]
PROOF. The proof is postponed to Appendix D.

REMARK 4 (Explicit regularization). Theorem 4.1 shows that variation-constrained neural networks, when trained and evaluated under the $\ell_1$ loss, does not explicitly depend on the input dimension $d$. Compared with the result in Theorem 3.1, the rate dependence on $n$ is similar, but the reliance on the variation $V$ is much relaxed.

In practice, we can operate the following regularized optimization to reach the desirable statistical risk. According to the method of Lagrange multipliers, the constrained optimization problem in (17) can be formulated as

(19) $\hat{f}_n = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} |y_i - f(x_i)| + \lambda V(f)^L$

where $\mathcal{F}$ is the same neural network class without any constraint, and $\lambda$ is some appropriately chosen parameter. Theorem 4.1 implies that for any $f \in \mathcal{F}$,

$R_1(f) \leq R_{1,n}(f) + O\left(\sqrt{\frac{\log n}{n}}\right) V(f)^L + O\left(\frac{\tau}{\sqrt{n}}\right)$,

which further implies that $\lambda$ can be chosen at the order of $O\left(\frac{\sqrt{\log n}}{\sqrt{n}}\right)$ in (19).

REMARK 5 (Implication on neural network model selection). There are two general ways of selecting a neural network model in practice. One is to consider a set of candidate architectures and choose the one with the best cross-validation performance. Classical asymptotic analysis of the generalization error alludes that more neural weights (or neurons) tend to cause overfitting. For example, an information criterion-type derivation indicates that the generalization error typically grows linearly with the number of free parameters (Ding, Tarokh and Yang, 2018). Nevertheless, recent research has shown that an overly-large network does not necessarily cause overfitting (Zhang et al., 2016). A possible reason is the failure of regularity conditions traditionally required for M-estimators.

The other way, which is perhaps more prevalent in practice these days, is to train an extensive neural network with properly tuned regularization terms. A practical benefit of the second approach is its more straightforward hardware implementation and computation, as we do not need to implement and train multiple models separately. Theoretically, Theorem 4.1 provides insight on when overfitting will not occur.

5. Conclusion. In this work, we showed tight statistical risk bounds for variation-constrained deep neural networks under both $\ell_1$ and $\ell_2$ loss functions. Several related prob-
lems need further study. First, a similar analysis may be emulated to study the performance of deep neural network-based classification models. Second, it would be interesting to study the $\ell_2$ risk of a model that is trained from the $\ell_1$ empirical risk, especially under non-IID or heavy-tail noises. The third problem is to study the relationship between the variation-based regularization and implicit regularization techniques (e.g., the early stopping and the dropout) practically operated in training deep neural networks.

**Acknowledgement.** The authors thank Yuhong Yang for his comments on improving the paper. The authors thank Enmao Diao for his help with the experimental studies.

**APPENDIX A: PROOF OF TECHNICAL LEMMAS**

In this Appendix, we prove the technical lemmas used in the proof of Theorem 3.1.

**A.1. Proof of Lemma 3.2.** Since

\[(20) \| f - \hat{f}_n \|_{L_\infty(X)} \leq \epsilon, \]

we have

\[
\mathbb{E} (\hat{f}_n(x) - f_s(x))^2 - \mathbb{E} (f(x) - f_s(x))^2
\]

\[
= \mathbb{E} (\hat{f}_n(x) - f_s(x) + f(x) - f_s(x))(\hat{f}_n(x) - f(x))
\]

\[
\leq 4V\mathbb{E} |\hat{f}_n(x) - f(x)|
\]

\[
\leq 4V\epsilon.
\]

and similarly,

\[
\frac{1}{n} \sum_{i=1}^{n} \left\{ (f(x_i) - f_s(x_i) - \varepsilon_i)^2 - (\hat{f}_n(x_i) - f_s(x_i) - \varepsilon_i)^2 \right\}
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} \left\{ (f(x_i) - f_s(x_i) - \varepsilon_i + \hat{f}_n(x_i) - f_s(x_i) - \varepsilon_i)(f(x_i) - \hat{f}_n(x_i)) \right\}
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - f_s(x_i) + \hat{f}_n(x_i) - f_s(x_i))(f(x_i) - \hat{f}_n(x_i))
\]

\[
- \frac{1}{n} \sum_{i=1}^{n} \{2\varepsilon_i(f(x_i) - \hat{f}_n(x_i))\}
\]

\[
\leq (4V + 2\max_i \varepsilon_i) \frac{1}{n} \sum_{i=1}^{n} |f(x_i) - \hat{f}_n(x_i)|
\]
\[ \leq 4(V + \max_i \varepsilon_i)\epsilon. \]

### A.2. Proof of Lemma 3.3.

We first show by the induction method that for any \(1 \leq l < L\) and \(1 \leq i \leq r_l\),

\[ (21) \quad \| \nabla^k y_{l,i} \|_\infty \lesssim V^l \]

for \(k \leq \alpha d\), where we let \(\alpha = 1/\delta\) and \(\nabla\) is the differentiation operator on \(x\). First, according to Assumption 2, we have for \(l = 1\),

\[
\| \nabla^k y_{1,i} \|_\infty = \| \nabla^k \sigma(w^\top_{1,i} x) \|_\infty \\
= \| \sigma^{(k)}(w^\top_{1,i} x) w^{\otimes k}_{1,i} \|_\infty \\
\lesssim \| w^{\otimes k}_{1,i} \|_\infty \\
\leq \| w_{1,i} \|^k \lesssim V^k.
\]

Assume that (21) holds for \(1, \ldots, l\). Next, we will prove the claim also holds for \(y_{l+1,i} = \sigma(w^\top_{l+1,i} y_l)\) by showing that

\[ (22) \quad \| \nabla^k \sigma^{(s)}(w^\top_{l+1,i} y_l) \|_\infty \lesssim V^{(l+1)k} \]

for any integers \(1 \leq i \leq r_{l+1}\) and \(1 \leq s \leq \alpha d - k\), also by using the induction method (on \(k\)).

For \(k = 1\), we have,

\[
\| \nabla \sigma^{(s)}(w^\top_{l+1,i} y_l) \|_\infty = \| \sigma^{(s+1)}(w^\top_{l+1,i} y_l) \nabla(w^\top_{l+1,i} y_l) \|_\infty \\
\lesssim \| \nabla(w^\top_{l+1,i} y_l) \|_\infty \\
\leq V \max_{1 \leq j \leq r_l} \| \nabla y_{l,j} \|_\infty \\
\lesssim V^{l+1}.
\]

Assume that (22) holds for \(1, \ldots, k - 1\). Then, for \(s \leq \alpha d - k\)

\[
\| \nabla^k \sigma^{(s)}(w^\top_{l+1,i} y_l) \|_\infty = \| \nabla^{k-1} \nabla \sigma^{(s)}(w^\top_{l+1,i} y_l) \|_\infty \\
= \| \nabla^{k-1} [\sigma^{(s+1)}(w^\top_{l+1,i} y_l) \nabla(w^\top_{l+1,i} y_l)] \|_\infty \\
= \| \nabla^{k-1} \sigma^{(s+1)}(w^\top_{l+1,i} y_l) \otimes \nabla(w^\top_{l+1,i} y_l) \\
+ \sigma^{(s+1)}(w^\top_{l+1,i} y_l) \nabla^k(w^\top_{l+1,i} y_l) \|_\infty
\]
\[
\begin{align*}
\|\nabla^k f(x)\|_\infty &= \|w_L^\top \nabla^k y_{L-1}\|_\infty \\
&\leq V \max_{1 \leq i \leq r_{L-1}} \|\nabla^k y_{L-1,i}\|_\infty \\
&\lesssim V \cdot V^{k(L-1)}.
\end{align*}
\]

The above result implies \(\mathcal{F}(V)\) belongs to the smooth functions class with order \(\alpha d\).

Then according to Lemma B.2, the covering number of \(\mathcal{F}(V)\) satisfies
\[
\log N(\epsilon) \leq \nu (V^{\alpha d(L-1)+1})^{\frac{1}{\delta}} \epsilon^{-\frac{2}{\delta}} = \nu V^{d(L-1)+\frac{1}{\delta}} \epsilon^{-\frac{2}{\delta}},
\]
where \(\nu\) is a constant that only depends on \(d, \alpha\). We conclude the proof by invoking \(\alpha = 1/\delta\).

**A.3. Proof of Lemma 3.4.** For any fixed \(f \in \mathcal{N}_\epsilon\), we let \(\xi_i(f) = (f(x_i) - f_*(x_i)) - \epsilon_i^2 - \epsilon_i^2\), or \(\xi_i\) for notational simplicity. It can be verified that
\[
\mathbb{E}(\xi_i) = \mathbb{E}(f(x) - f_*(x))^2,
\]
and
\[
\text{var}(\xi_i) = \text{var}\{ (f(x_i) - f_*(x_i))^2 - 2\epsilon_i(f(x_i) - f_*(x_i)) \}
\]
\[
\leq \mathbb{E}\{ (f(x_i) - f_*(x_i))^2 - 2\epsilon_i(f(x_i) - f_*(x_i)) \}^2
\]
\[
= \mathbb{E}\{ (f(x_i) - f_*(x_i))^4 + 4\epsilon_i^2(f(x_i) - f_*(x_i))^2 \}
\]
\[
\leq 4(V^2 + \tau^2)\mathbb{E}(f(x) - f_*(x))^2.
\]

In the last inequality, we have used the inequality that
\[
f(x) = w_L^\top y_{L-1} = w_L^\top [\sigma(w_{L-1,i}^\top y_{L-2}), \ldots, \sigma(w_{L-1,i}^\top y_{L-2})]^\top
\]
\[
(23)
\]
\[
\leq \|w_L^\top\|_1 \leq V, \quad \forall x \in X, \forall f \in \mathcal{F}_L(V).
\]
According to the Bernstein’s Inequality (Lemma B.1 in the Appendix) and (23), with probability at least $1 - \frac{\rho}{N(\epsilon)}$ we have

$$
\mathbb{E} (f(x) - f_*(x))^2 - \frac{1}{n} \sum_{i=1}^{n} \xi_i 
\leq \frac{\max_i |\xi_i|}{n} \log \frac{N(\epsilon)}{\rho} + \sqrt{\frac{\text{var}(\xi) \log N(\epsilon)}{n \rho}}.
$$

(24)

It can be verified that the above (24) implies that

$$
\mathbb{E} (f(x) - f_*(x))^2 - \frac{1}{n} \sum_{i=1}^{n} \xi_i \lesssim \frac{V(V + \max_i |\xi_i|)}{n} \log \frac{N(\epsilon)}{\rho} + \sqrt{\frac{(V^2 + \tau^2) \mathbb{E} (f(x) - f_*(x))^2}{n \rho}} \log \frac{N(\epsilon)}{\rho}.
$$

(25)

By the union bound over $f \in N_\epsilon$, (25) holds uniformly for all $f \in N_\epsilon$ with probability at least $1 - \rho$.

**APPENDIX B: AUXILIARY LEMMAS**

**LEMMA B.1 (Bernstein’s Inequality).** Assume $X_1, X_2, \ldots, X_n$ are independent random variables, then with probability at least $1 - \rho$,

$$
\left| \sum_{i=1}^{n} (X_i - \mathbb{E} X_i) \right| \lesssim \max_{1 \leq i \leq n} |X_i - \mathbb{E} X_i| \log \frac{1}{\rho} + \sqrt{\sum_{i=1}^{n} \text{var}(X_i) \log \frac{1}{\rho}}.
$$

**LEMMA B.2 (Metric Entropy Bound).** Let $F$ be the class of all functions $f : [0,1]^d \to \mathbb{R}$, whose partial derivatives up to order $\alpha d$ (which is supposed to be a positive integer) exist and are uniformly bounded by a constant $M$. Define the $L_\infty$ metric as $\|f_1 - f_2\|_{L_\infty} = \max_{x \in [0,1]^d} |f_1(x) - f_2(x)|$, and $N(\epsilon)$ is the minimum covering number under such a metric. Then,

$$
\log N(\epsilon) \leq \nu M^{\frac{1}{\alpha}} \epsilon^{-\frac{1}{\alpha}},
$$

where $\nu$ depends on $\alpha, d, M$ only.

**PROOF.** For two real-valued functions $l$ and $u$, the bracket $[l, u]$ is the set of all functions $f$ satisfying $l \leq f \leq u$. An $\epsilon$-bracket in $L_2(\mu)$-space is a bracket $[l, u]$ with $\|u - l\|_{L_2(\mu)} \leq \epsilon$. The bracketing number $N([\epsilon])$ is the smallest number of $\epsilon$-brackets needed to cover the function class. Since the $\epsilon$-bracket $[l, u]$ is contained in the ball with a radius of $\frac{\epsilon}{2}$ centered
on $\frac{1+u}{2}$ in $L_2(\mu)$, an $\varepsilon$-bracket covering is an $\varepsilon$ covering, and thus $N_\|\cdot\|_2(\varepsilon) \geq N(\varepsilon)$. Then, this lemma directly follows from Theorem 2.1 in (van der Vaart, 1994).

**Lemma B.3** (Contraction Lemma (Neyshabur, Tomioka and Srebro, 2015)). Suppose that a function $g$ is $L$-Lipschitz and $g(0) = 0$. Suppose that $\xi_1, \ldots, \xi_n$ are IID symmetric Bernoulli random variables taking values from $\{1, -1\}$. Then, for any function class $\mathcal{F}$ mapping from $\mathcal{X}$ to $\mathbb{R}$, and any set $\{x_1, x_2, \ldots, x_n\}$, we have

$$\mathbb{E} \sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n \xi_i g(f(x_i)) \right| \leq 2L \mathbb{E} \sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n \xi_i f(x_i) \right|.$$ 

**Appendix C: Proof of Proposition 3.5**

We define a subclass of $\mathcal{F}_2(V)$ by

$$\mathcal{F}_0 = \left\{ f : \mathbb{R}^d \to \mathbb{R} \mid f(x) = V \sigma(w^\top x), \|w\|_2 = 1 \right\}.$$ 

It can be verified that

$$\mathbb{E} \left| \sigma(w_1^\top x) - \sigma(w_2^\top x) \right| \geq \mathbb{E} \left\{ \inf_u \sigma'(u) \cdot |w_1^\top x - w_2^\top x| \cdot \mathbb{I}(w_1^\top x, w_2^\top x \in S) \right\} \gtrsim \|w_1 - w_2\|_2.$$ 

Let $M_1(\varepsilon)$ denote the packing $\varepsilon$-entropy of $\mathcal{F}_0$ with the $\ell_1$ distance. Then, $M_1(\varepsilon)$ is greater than the packing $\varepsilon$-entropy with the $\ell_2$ distance, written as $M_1(\varepsilon) \geq M_2(\varepsilon) \gtrsim d$. Let $V_k(\varepsilon)$ denote the covering $\varepsilon$-entropy of $\mathcal{F}_0$ with the square root Kullback-Leibler divergence, then according to its relation with the $\ell_2$ distance shown in (Yang and Barron, 1999), we have

$$V_k(\varepsilon) \leq M_2(\sqrt{2}\varepsilon) \lesssim d \log \frac{1}{\varepsilon}.$$ 

Hence, according to (Yang and Barron, 1999, Theorem 1), for $q = 1, 2$,

$$\inf_{\hat{f}_n} \sup_{f \in \mathcal{F}_2(V)} \|\hat{f}_n - f\|_{L_q(\mu)} \geq \inf_{\hat{f}_n} \sup_{f \in \mathcal{F}_0} \|\hat{f}_n - f\|_{L_q(\mu)} \gtrsim V \sqrt{\frac{d}{n}}.$$ 

This concludes the proof.

**Appendix D: Proof of Theorem 4.1**

We define the empirical risk

$$\mathcal{R}_{1,n}(f) = \mathbb{E} \left( \frac{1}{n} \sum_{i=1}^n |f_n(x_i) + \varepsilon_i - f(x_i)| \right) - \mathbb{E} \left| \varepsilon \right|.$$
Since \( \hat{f}_n \) minimizes \( n^{-1} \sum_{i=1}^{n} |f_\ast(x_i) + \varepsilon_i - f(x_i)| \) in \( \mathcal{F}_L(V) \), we have

\[
R_1(\hat{f}_n) \leq R_1(\hat{f}_n) - \{ R_{1,n}(\hat{f}_n) - R_{1,n}(f_\ast) \} = R_1(\hat{f}_n) - R_{1,n}(\hat{f}_n).
\]

In the following, we will analyze the term \( R_1(\hat{f}_n) - R_{1,n}(\hat{f}_n) \) in (29).

Let \( z_i \)'s denote IID copies of \( x_i \)'s.

\[
R_1(\hat{f}_n) - R_{1,n}(\hat{f}_n) = \mathbb{E} \frac{1}{n} \sum_{i=1}^{n} \left\{ |\hat{f}_n(z_i) - f_\ast(z_i) - \varepsilon_i| - |\hat{f}_n(x_i) - f_\ast(x_i) - \varepsilon_i| \right\}
\]

\[
\leq \mathbb{E} \sup_{f \in \mathcal{F}_L(V)} \frac{1}{n} \sum_{i=1}^{n} \left\{ |f(z_i) - f_\ast(z_i) - \varepsilon_i| - |f(x_i) - f_\ast(x_i) - \varepsilon_i| \right\}
\]

\[
\leq 2 \mathbb{E} \sup_{f \in \mathcal{F}_L(V)} \frac{1}{n} \sum_{i=1}^{n} \xi_i |f(z_i) - f_\ast(z_i) - \varepsilon_i|,
\]

where \( \xi_1, \ldots, \xi_n \) are IID symmetric Bernoulli random variables that are independent with \( z_i \)'s and take values from \( \{1, -1\} \). According to Lemma B.3, since \( g(x) = |x| \) is 1-Lipschitz and \( g(0) = 0 \), we have

\[
\mathbb{E} \sup_{f \in \mathcal{F}_L(V)} \frac{1}{n} \sum_{i=1}^{n} \xi_i |f(z_i) - f_\ast(z_i) - \varepsilon_i| \leq 2 \mathbb{E} \sup_{f \in \mathcal{F}_L(V)} \left| \frac{1}{n} \sum_{i=1}^{n} \xi_i (f(z_i) - f_\ast(z_i) - \varepsilon_i) \right|
\]

\[
\leq 2 \mathbb{E} \sup_{f \in \mathcal{F}_L(V)} \left| \frac{1}{n} \sum_{i=1}^{n} \xi_i f(z_i) \right| + 2 \sqrt{\frac{\mathbb{E} y^2}{n}}
\]

\[
\lesssim \mathbb{E} \sup_{f \in \mathcal{F}_L(V)} \left| \frac{1}{n} \sum_{i=1}^{n} \xi_i f(z_i) \right| + \frac{V + \tau}{\sqrt{n}}.
\]

To conclude the proof, it remains to show that

\[
\mathbb{E} \sup_{f \in \mathcal{F}_L(V)} \left| \frac{1}{n} \sum_{i=1}^{n} \xi_i f(z_i) \right| \lesssim V^{L_2} \sqrt{\frac{\log n}{n}}.
\]

Let \( W(f) = \{ w_1, \ldots, w_L \} \) denote the neural weights of \( f \) at all the layers. It can be verified that

\[
\mathbb{E} \sup_{f \in \mathcal{F}_L(V)} \left| \frac{1}{n} \sum_{i=1}^{n} \xi_i f(z_i) \right| = \mathbb{E} \sup_{W(f): f \in \mathcal{F}_L(V)} \left| \langle w_L, \frac{1}{n} \sum_{i=1}^{n} \xi_i y_{L-1}(z_i) \rangle \right|
\]

\[
\leq V \mathbb{E} \sup_{W(f): f \in \mathcal{F}_L(V)} \left\| \frac{1}{n} \sum_{i=1}^{n} \xi_i y_{L-1}(z_i) \right\|_\infty
\]

\[
\leq V \mathbb{E} \sup_{W(f): f \in \mathcal{F}_L(V)} \left| \frac{1}{n} \sum_{i=1}^{n} \xi_i \sigma(w_{L-1,j}^T y_{L-2}(z_i)) \right|
\]

\[
\lesssim V \mathbb{E} \sup_{W(f): f \in \mathcal{F}_L(V)} \left| \frac{1}{n} \sum_{i=1}^{n} \xi_i w_{L-1,j}^T y_{L-2}(z_i) \right|.
\]
\begin{equation}
\leq \ldots \\
\leq V^{L-1} \mathbb{E} \sup_{w_1 \in \mathbb{R}^{d}} \left| \frac{1}{n} \sum_{i=1}^{n} \xi_i \sigma(w_1^\top z_i) \right| \\
\leq V^L \sqrt{\log \frac{n}{n}}.
\end{equation}

REFERENCES

BARANIUK, R. and BALESTRIERO, R. (2018). A spline theory of deep learning. In Proc. ICML 374–383.

BARRON, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. IEEE Trans. Inf. Theory 39 930–945.

BARRON, A. R. (1994). Approximation and estimation bounds for artificial neural networks. Mach. Learn. 14 115–133.

BARRON, A. R. and KLUSOWSKI, J. M. (2019). Complexity, statistical risk, and metric entropy of deep nets using total path variation. arXiv Prepr. arXiv1902.00800.

BAUER, B. and KOHLER, M. (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. Ann. Stat. 47 2261–2285.

CYBENKO, G. (1989). Approximations by superpositions of a sigmoidal function. Math. Control Signals Syst. 2 183–192.

DING, J., TAROKH, V. and YANG, Y. (2018). Model selection techniques: An overview. IEEE Signal Process. Mag. 35 16–34.

DING, J., DIAO, E., ZHOU, J. and TAROKH, V. (2020). On Statistical Efficiency in Learning. http://jding.org/jie-uploads/2020/05/LoL.pdf 65 3034–3067.

GE, R., LEE, J. D. and MA, T. (2017). Learning one-hidden-layer neural networks with landscape design. arXiv Prepr. arXiv1711.00501.

GOLOWICH, N., RAKHLIN, A. and SHAMIR, O. (2017). Size-independent sample complexity of neural networks. arXiv Prepr. arXiv1712.06541.

GOOGLE, R. T. (2016). Tensorflow: A system for large-scale machine learning. Proc. 12th Symp. Oper. Syst. Des. Implement. 265–283.

GULLI, A. and PAL, S. (2017). Deep Learning with Keras. Packt Publishing Ltd.

JANZAMIN, M., SEDGHI, H. and ANANDKUMAR, A. (2015). Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. arXiv Prepr. arXiv1506.08473.

KETKAR, N. (2017). Introduction to pytorch. Deep Learn. with python 195–208.

LI, G., GU, Y. and DING, J. (2020). The Efficacy of L1 Regularization in Neural Networks. arXiv Prepr. arXiv2010.01048.

MONDELLI, M. and MONTANARI, A. (2018). On the connection between learning two-layers neural networks and tensor decomposition. arXiv Prepr. arXiv1802.07301.
NEYSHABUR, B., TOMIOKA, R. and SREBRO, N. (2015). Norm-based capacity control in neural networks. Conf. Learn. Theory 1376–1401.

PEDREGOSA, F., BACH, F. and GRAMFORT, A. (2017). On the consistency of ordinal regression methods. J. Mach. Learn. Res. 18 1769–1803.

SCHMIDT-HIEBER, J. (2017). Nonparametric regression using deep neural networks with ReLU activation function. arXiv Prepr. arXiv1708.06633.

VAN DER VAART, A. (1994). Bracketing smooth functions. Stochastic Processes and their Applications 52 93–105.

WELSH, R. E. (1977). Robust regression using iteratively reweighted least-squares. Commun. Stat. - Theory Methods.

YANG, Y. and BARRON, A. (1999). Information-theoretic determination of minimax rates of convergence. Ann. Stat. 1564–1599.

ZHAO, H., GALLO, O., FROSIO, I. and KAUTZ, J. (2016). Loss functions for image restoration with neural networks. IEEE Trans. Comput. 3 47–57.