Fast generalization error bound of deep learning from a kernel perspective

Taiji Suzuki
taiji@mist.i.u-tokyo.ac.jp
Graduate School of Information Science and Technology, The University of Tokyo
PRESTO, Japan Science and Technology Agency, Japan
Center for Advanced Integrated Intelligence Research, RIKEN, Tokyo, Japan

Abstract

We develop a new theoretical framework to analyze the generalization error of deep learning, and derive a new fast learning rate for two representative algorithms: *empirical risk minimization* and *Bayesian deep learning*. The series of theoretical analyses of deep learning has revealed its high expressive power and universal approximation capability. Our point of view is to deal with the ordinary finite dimensional deep neural network as a finite approximation of the infinite dimensional one. Our formulation of the infinite dimensional model naturally defines a reproducing kernel Hilbert space corresponding to each layer. The approximation error is evaluated by the *degree of freedom* of the reproducing kernel Hilbert space in each layer. We derive the generalization error bound of both of empirical risk minimization and Bayesian deep learning and it is shown that there appears bias-variance trade-off in terms of the number of parameters of the finite dimensional approximation. We show that the optimal width of the internal layers can be determined through the degree of freedom and derive the optimal convergence rate that is faster than $O(1/\sqrt{n})$ rate which has been shown in the existing studies.

1 Introduction

Deep learning has been showing great success in several applications such as computer vision, natural language processing, and many other area related to pattern recognition. Several high-performance methods have been developed and it has been revealed that deep learning possesses great potential. Despite the development of practical methodologies, its theoretical understanding is not satisfactory. Wide rage of researchers including theoreticians and practitioners are expecting deeper understanding of deep learning.

Among theories of deep learning, a well developed topic is its expressive power. It has been theoretically shown that deep neural network has exponentially large expressive power against the number of layers using several mathematical notions Montufar et al. (2014); Bianchini and Scarselli (2014); Cohen et al. (2016); Cohen and Shashua (2016); Poole et al. (2016). Another important issue in neural network theories is its universal approximation capability. It is well known that 3-layer neural networks have the ability, and thus the deep neural network also does (Cybenko, 1989; Hornik, 1991; Sonoda and Murata, 2015). When we discuss the universal approximation capability, the target function that is approximated is arbitrary and the theory is highly nonparametric in its nature.

Once we knew the expressive power and universal approximation capability of deep neural network, the next theoretical question naturally arises in its generalization error. The generalization ability is typically analyzed by evaluating the *Rademacher complexity* (see Bartlett (1998); Koltchinskii and Panchenko (2002); Neyshabur et al. (2015); Sun et al. (2015)). As a whole, these studies have characterized the generalization ability based on the properties of the weights (parameters) but the properties of the input distributions are not well incorporated which is not satisfactory to obtain a sharper bound. Moreover, the generalization error bound has been mainly given in finite dimensional models. As we have observed, the deep neural network possesses exponential expressive power and

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

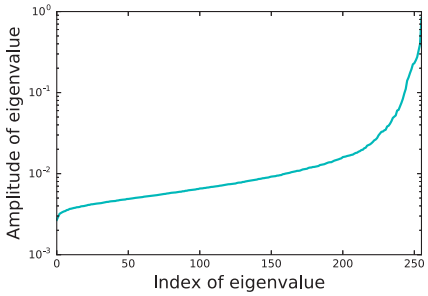

Figure 1: Distribution of eigen-values in the 9-th layer of VGG for CIFAR-10 dataset.

universal approximation capability which are highly nonparametric characterizations. This means that the theories are developed separately in the two regimes; finite dimensional parametric model and infinite dimensional nonparametric model. Therefore, theories that connect these two regimes are expected to comprehensively understand statistical performance of deep learning.

In this paper, we consider both of empirical risk minimization and Bayesian deep learning and analyze the generalization error using the terminology of kernel methods. The main purpose of the analysis is to find what kind of quantity affects the optimal structure of the deep learning. Especially, we see that the generalization error is well affected by the behavior of the eigenvalues of the (non-centered) covariance matrix for the output to each internal layer. Figure 1 shows the distribution of the eigenvalues of VGG-13 network where the eigenvalues are ordered in increasing order. We see that the distribution is distorted in a sense that a few eigenvalues are large and the smallest eigenvalue is much smaller than the largest one. This means that an effective information is included in a low-dimensional subspace. Our analysis aims to find the optimal dimension of the low dimensional subspace by borrowing the techniques developed in the analysis of *kernel methods*. Consequently, (i) we derive a faster learning rate than $O(1/\sqrt{n})$ and (ii) we connect the finite dimensional regime and the infinite dimensional regime based on the theories of kernel methods. To analyze a sharper generalization error bound, we utilize the so-called local Rademacher complexity technique for the empirical risk minimization method (Mendelson, 2002; Bartlett et al., 2005; Koltchinskii, 2006; Giné and Koltchinskii, 2006), and, as for the Bayesian method, we employ the theoretical techniques developed to analyze nonparametric Bayes methods (Ghosal et al., 2000; van der Vaart and van Zanten, 2008, 2011). These analyses are quite advantageous to the typical Rademacher complexity analysis because we can obtain convergence rate between $O(1/n)$ and $O(1/\sqrt{n})$ which is faster than that of the standard Rademacher complexity analysis $O(1/\sqrt{n})$ by making use of favorable properties of the loss function such as strong convexity. As for the second contribution, we first introduce an integral form of deep neural network as performed in the research of the universal approximation capability of 3-layer neural networks (Sonoda and Murata, 2015). Based on this integral form, a reproducing kernel Hilbert space (RKHS) naturally arises. Due to this formulation, we can borrow the terminology developed in the kernel method into the analysis of deep learning. In particular, we define the degree of freedom of the RKHS as a measure of complexity of the RKHS (Caponnetto and de Vito, 2007; Bach, 2017b), and based on that, we evaluate how large a finite dimensional model should be to approximate the original infinite dimensional model with a specified precision. These theoretical developments reveal that there appears *bias-variance trade-off*. An interesting observation here is that this bias-variance trade-off is completely characterized by the behaviors of the eigenvalues of the kernel functions corresponding to the RKHSs. In short, if the distribution of eigenvalues is peaky (that means, the decreasing rate of ordered eigenvalues is fast), then the optimal width (the number of units in each layer) can be small and consequently the variance can be reduced. This indicates that the notion of the degree of freedom gives a practical implication about determination of the width of the internal layers.

The obtained generalization error bound is summarized in Table 1[1].

2 Integral representation of deep neural network

Here we give our problem settings and the model that we consider in this paper. Suppose that n input-output observations $D_n = (x_i, y_i)_{i=1}^n \subset \mathbb{R}^{d_x} \times \mathbb{R}$ are independently identically generated from a regression model

$$y_i = f^{\mathrm{o}}(x_i) + \xi_i \quad (i = 1, \ldots, n)$$

where $(\xi_i)_{i=1}^n$ is an i.i.d. sequence of Gaussian noises $N(0, \sigma^2)$ with mean 0 and variance σ^2, and $(x_i)_{i=1}^n$ is generated independently identically from a distribution $P_X(X)$ with a compact support in $\mathbb{R}^{d_x}$. The purpose of the deep learning problem we consider in this paper is to estimate f^{o} from the n observations D_n. We may consider other situations such as classifications with margin assumptions, however, just for theoretical simplicity, we consider the simplest regression problem.

To analyze the generalization ability of deep learning, we specify a function class in which the true function

[1] $a \vee b$ indicates $\max\{a, b\}$.

Table 1: Summary of derived bounds for the generalization error $\|\widehat{f} - f^{\mathrm{o}}\|_{L_2(P_X)}^2$ where n is the sample size, R is the norm of the weight in the internal layers, $\hat{R}_\infty$ is an L_∞-norm bound of the functions in the model, σ is the observation noise, d_{x} is the dimension of the input, m_ℓ is the width of the ℓ-th internal layer and $N_\ell(\lambda_\ell)$ for $(\lambda_\ell > 0)$ is the degree of freedom (Eq. (5)).

	Error bound
General setting	$L\sum_{\ell=2}^{L} R^{L-\ell+1}\lambda_\ell + \frac{\sigma^2+\hat{R}_\infty^2}{n}\sum_{\ell=1}^{L} m_\ell m_{\ell+1}\log(n)$ under an assumption that $m_\ell \gtrsim N_\ell(\lambda_\ell)\log(N_\ell(\lambda_\ell))$.
Finite dimensional model	$\frac{\sigma^2+\hat{R}_\infty^2}{n}\sum_{\ell=1}^{L} m_\ell^* m_{\ell+1}^*\log(n)$ where m_ℓ^* is the true width of the ℓ-th internal layer.
Polynomial decay eigenvalue	$L\sum_{\ell=2}^{L}(R\vee 1)^{L-\ell+1} n^{-\frac{1}{1+2s_\ell}}\log(n) + \frac{d_{\mathrm{x}}^2}{n}\log(n)$ where s_ℓ is the decay rate of the eigenvalue of the kernel function on the ℓ-th layer.

f^{o} is included, and, by doing so, we characterize the "complexity" of the true function in a correct way.

In order to give a better intuition, we first start from the simplest model, the 3-layer neural network. Let η be a nonlinear activation function such as ReLU (Nair and Hinton, 2010; Glorot et al., 2011); $\eta(x) = (\max\{x_i, 0\})_{i=1}^d$ for a d-dimensional vector $x \in \mathbb{R}^d$. The 3-layer neural network model is represented by

$$f(x) = W^{(2)}\eta(W^{(1)}x + b^{(1)}) + b^{(2)}$$

where we denote by m_2 the number of nodes in the internal layer, and $W^{(2)} \in \mathbb{R}^{1\times m_2}$, $W^{(1)} \in \mathbb{R}^{m_2\times d_{\mathrm{x}}}$, $b^{(1)} \in \mathbb{R}^{m_2}$ and $b^{(2)} \in \mathbb{R}$. It is known that this model is *universal approximator* and it is important to consider its *integral form*

$$f(x) = \int h(w,b)\eta(w^\top x + b)\mathrm{d}w\mathrm{d}b + b^{(2)}. \qquad (1)$$

where $(w,b) \in \mathbb{R}^{d_{\mathrm{x}}} \times \mathbb{R}$ is a hidden parameter, $h : \mathbb{R}^{d_{\mathrm{x}}} \times \mathbb{R} \to \mathbb{R}$ is a function version of the weight matrix $W^{(2)}$, and $b^{(2)} \in \mathbb{R}$ is the bias term. This integral form appears in many places to analyze the capacity of the neural network. In particular, through the ridgelet analysis, it is shown that there exists the integral form corresponding to any $f \in L_1(\mathbb{R}^{d_{\mathrm{x}}})$ which has an integrable Fourier transform for an appropriately chosen activation function η such as ReLU (Sonoda and Murata, 2015).

Motivated by the integral form of the 3-layer neural network, we consider a more general representation for deeper neural network. To do so, we define a feature space on the ℓ-th layer. The feature space is a a probability space $(\mathcal{T}_\ell, \mathcal{B}_\ell, \mathcal{Q}_\ell)$ where $\mathcal{T}_\ell$ is a Polish space, $\mathcal{B}_\ell$ is its Borel algebra, and $\mathcal{Q}_\ell$ is a probability measure on $(\mathcal{T}_\ell, \mathcal{B}_\ell)$. This is introduced to represent a general (possibly) continuous set of features as well as a discrete set of features. For example, if the ℓ-th internal layer is endowed with a d_ℓ-dimensional finite feature space, then $\mathcal{T}_\ell = \{1, \dots, d_\ell\}$. On the other hand, the integral form (1) corresponds to a continuous feature space $\mathcal{T}_2 = \{(w,b) \in \mathbb{R}^{d_{\mathrm{x}}} \times \mathbb{R}\}$ in the second layer. Now the input x is a d_{x}-dimensional real vector, and thus we may set $\mathcal{T}_1 = \{1, \dots, d_{\mathrm{x}}\}$. Since the output is one dimensional, the output layer is just a singleton $\mathcal{T}_{L+1} = \{1\}$. Based on these feature spaces, our integral form of the deep neural network is constructed by stacking the map on the ℓ-th layer $f_\ell^{\mathrm{o}} : L_2(Q_\ell) \to L_2(Q_{\ell+1})$ given as

$$f_\ell^{\mathrm{o}}[g](\tau) = \int_{\mathcal{T}_\ell} h_\ell^{\mathrm{o}}(\tau, w)\eta(g(w))\mathrm{d}Q_\ell(w) + b_\ell^{\mathrm{o}}(\tau), \quad (2a)$$

where $h_\ell^{\mathrm{o}}(\tau, w)$ corresponds to the weight of the feature w for the output τ and $h_\ell^{\mathrm{o}} \in L_2(Q_{\ell+1} \times Q_\ell)$ and $h_\ell^{\mathrm{o}}(\tau, \cdot) \in L_2(Q_\ell)$ for all $\tau \in \mathcal{T}_{\ell+1}$ [2]. Specifically, the first and the last layers are represented as

$$f_1^{\mathrm{o}}[x](\tau) = \sum_{j=1}^{d_{\mathrm{x}}} h_1^{\mathrm{o}}(\tau, j)x_j Q_1(j) + b_1^{\mathrm{o}}(\tau), \qquad (2b)$$

$$f_L^{\mathrm{o}}[g](1) = \textstyle\int_{\mathcal{T}_L} h_L^{\mathrm{o}}(w)\eta(g(w))\mathrm{d}Q_L(w) + b_L^{\mathrm{o}}, \qquad (2c)$$

where we wrote $h_L^{\mathrm{o}}(w)$ to indicate $h_L^{\mathrm{o}}(1, w)$ for simplicity because $\mathcal{T}_{L+1} = \{1\}$. Then the true function f^{o} is given as

$$f^{\mathrm{o}}(x) = f_L^{\mathrm{o}} \circ f_{L-1}^{\mathrm{o}} \circ \cdots \circ f_1^{\mathrm{o}}(x), \qquad (3)$$

where $f_\ell^{\mathrm{o}} \circ F(x)$ indicates $f_\ell^{\mathrm{o}}[F(x)](\cdot) \in L_2(Q_{\ell+1})$ for $F(x)(\cdot) \in L_2(Q_\ell)$. Since, the shallow 3-layer neural network is a universal approximator, and so is our generalized deep neural network model (3). It is known that deep neural network gives more efficient representation of a function than shallow ones. Actually, Eldan and Shamir (2016) gave an example of a function that

[2] Note that, for $g \in L_2(Q_\ell)$, $f_\ell[g]$ is also square integrable with respect to $L_2(Q_{\ell+1})$ if η is Lipschitz continuous because $h \in L_2(Q_{\ell+1} \times Q_\ell)$.

the 3-layer neural network cannot approximate under a precision unless its with is exponential in the input dimension but the 4-layer neural network can approximate with polynomial order widths (see Safran and Shamir (2016) for other examples). Therefore, it is quite important to consider the integral representation of a deep neural network rather than a 3-layer network.

The integral representation is natural also from the practical point of view. Indeed, it is well known that the deep neural network learns a simple pattern in the early layers and it gradually extracts more complicated features as the layer is going up. The trained feature is usually continuous one. For example, in computer vision tasks, the second layer typically extracts gradients toward several degree angles (Krizhevsky et al., 2012). The angle is a continuous variable and thus the feature space should be continuous to cover all angles. On the other hand, the real network discretize the feature space because of limitation of computational resources. Our theory introduced in the next section offers a measure to evaluate this discretization error.

3 Kernels and corresponding RKHSs

When we estimate that, we need to discretize the integrals by finite sums due to limitation of computational resources as we do in practice. In other word, we consider the usual finite sum deep learning model as an approximation of the integral form. However, the discrete approximation induces approximation error. Here we give an upper bound of the approximation error. Naturally, there arises the notion of bias and variance trade-off, that is, as the complexity of the finite model increases the "bias" (approximation error) decreases but the "variance" for finding the best parameter in the model increases. Combining these two notions, it is possible to quantify the bias-variance trade-off and find the best strategy to minimize the entire generalization error.

The approximation error analysis of the deep neural network can be well executed by utilizing notions of the kernel method. Here we construct RKHS for each layer in a way analogous to Bach (2017b,a) who studied shallow learning and the kernel quadrature rule. Let the output of the ℓ-th layer be $F_\ell^{\mathrm{o}}(x,\tau) := (f_\ell^{\mathrm{o}} \circ \cdots \circ f_1^{\mathrm{o}}(x))(\tau)$. We define a *reproducing kernel Hilbert space* (RKHS) corresponding to the ℓ-th layer ($\ell \geq 2$) by introducing its associated kernel function $\mathsf{k}_\ell : \mathbb{R}^{d_{\mathrm{x}}} \times \mathbb{R}^{d_{\mathrm{x}}} \to \mathbb{R}$. We define the positive definite kernel k_ℓ as

$$\mathsf{k}_\ell(x,x') := \int_{\mathcal{T}_\ell} \eta(F_{\ell-1}^{\mathrm{o}}(x,\tau))\eta(F_{\ell-1}^{\mathrm{o}}(x',\tau))\mathrm{d}Q_\ell(\tau).$$

It is easy to check that k_ℓ is symmetric and positive definite. It is known that there exists a unique RKHS $\mathcal{H}_\ell$ corresponding the kernel k_ℓ (Aronszajn, 1950). Under this setting, all arguments at the ℓ-th layer can be carried out through the theories of kernel methods. Importantly, for $g \in \mathcal{H}_\ell$, there exists $h \in L_2(Q_\ell)$ such that

$$g(x) = \int_{T_\ell} h(\tau)\eta(F_{\ell-1}^{\mathrm{o}}(x,\tau))\mathrm{d}Q_\ell(\tau).$$

Moreover, the norms of g and h are connected as

$$\|g\|_{\mathcal{H}_\ell} = \|h\|_{L_2(Q_\ell)}, \tag{4}$$

(Bach, 2017b,a). Therefore, the function $x \mapsto \int_{\mathcal{T}_\ell} h_\ell^{\mathrm{o}}(\tau,w)\eta(F_{\ell-1}^{\mathrm{o}}(x,w))\mathrm{d}Q_\ell(w)$, representing the magnitude of a feature $\tau \in \mathcal{T}_{\ell+1}$ for the input x is included in the RKHS and its RKHS norm is equivalent to that of the internal layer weight $\|h^{\mathrm{o}}(\tau,\cdot)\|_{L_2(Q_\ell)}$ because of Eq. (4).

To derive the approximation error, we need to evaluate the "complexity" of the RKHS. Basically, the complexity of the ℓ-th layer RKHS $\mathcal{H}_\ell$ is controlled by the behavior of the eigenvalues of the kernel. To formally state this notion, we consider the Hilbert-Schmidt decomposition of the kernel given by

$$\mathsf{k}_\ell(x,x') = \sum_{j=1}^\infty \mu_j^{(\ell)}\phi_j^{(\ell)}(x)\phi_j^{(\ell)}(x'),$$

in $L_2(P_X \times P_X)$ where $(\mu_j^{(\ell)})_{j=1}^\infty$ is the sequence of the eigenvalues ordered in decreasing order, and $(\phi_j^{(\ell)})_{j=1}^\infty$ forms an orthonormal system in $L_2(P_X)$.

Based on the Hilbert-Schmidt decomposition we define the *degree of freedom* $N_\ell(\lambda)$ of the RKHS as

$$N_\ell(\lambda) = \sum_{j=1}^\infty \mu_j^{(\ell)}/(\mu_j^{(\ell)} + \lambda) \tag{5}$$

for arbitrary $\lambda > 0$. Roughly speaking, this is an effective dimension of the subspace that approximates the infinite dimensional RKHS with precision λ. Note that $N_\ell(\lambda)$ is a monotonically decreasing function with respect to λ.

Note that, by using the canonical expansion $\eta(F_{\ell-1}^{\mathrm{o}}(x,\tau)) = \sum_{j=1}^\infty \sqrt{\mu_j^{(\ell)}}\phi_j^{(\ell)}(x)\psi_j^{(\ell)}(\tau)$ in $L_2(P_X \times Q_\ell)$, the eigenvalues $(\mu_j^{(\ell)})_{j=1}^\infty$ are also those of the *covariance operator* $\Sigma_\ell : L_2(Q_\ell) \to L_2(Q_\ell)$ that is defined by $(\Sigma_\ell h)(\tau) = \int\{\int \eta(F_{\ell-1}^{\mathrm{o}}(x,\tau))\eta(F_{\ell-1}^{\mathrm{o}}(x,\tau'))\mathrm{d}P_X(x)\}h(\tau')\mathrm{d}Q_\ell(\tau')$ for $h \in L_2(Q_\ell)$ because Σ_ℓ can be represented by $\Sigma_\ell = \sum_{j=1}^\infty \mu_j^{(\ell)}\psi_j^{(\ell)}\langle\psi_j^{(\ell)},\cdot\rangle_{L_2(Q_\ell)}$.

4 Generalization error analysis: bias-variance trade-off

In this section, we give the generalization error analysis using the notion of kernel methods. To do so, first we give some assumptions.

We assume that the true function f° satisfies a norm condition as follows.

Assumption 1 *For each ℓ, h_ℓ° and b_ℓ° satisfy that*

$$\|h_\ell^{\circ}(\tau,\cdot)\|_{L_2(Q_\ell)} \le R,\ |b_\ell^{\circ}(\tau)| \le R_b\ (\forall \tau \in T_\ell).$$

By Eq. (4), the first assumption $\|h_\ell^{\circ}(\tau,\cdot)\|_{L_2(Q_\ell)} \le R$ is interpreted as $F_\ell^{\circ}(\tau,\cdot) \in \mathcal{H}_\ell$ and $\|F_\ell^{\circ}(\tau,\cdot)\|_{\mathcal{H}_\ell} \le R$. This means that the feature map $F_\ell^{\circ}(\tau,\cdot)$ in each internal layer is well regulated by the RKHS norm. Moreover, we also assume that the activation function is scale invariant.

Assumption 2 *We assume the following conditions on the activation function η.*

- *η is scale invariant: $\eta(ax) = a\eta(x)$ for all $a > 0$ and $x \in \mathbb{R}^d$ (for arbitrary d).*
- *η is 1-Lipschitz continuous: $|\eta(x) - \eta(x')| \le \|x - x'\|$ for all $x, x' \in \mathbb{R}^d$.*

The first assumption on the scale invariance is essential to derive tight error bounds. The second one ensures that deviation in each layer does not affect the output so much. The most important example of an activation function that satisfies these conditions is ReLU activation. Another one is the identity map $\eta(x) = x$.

Finally we assume that the input distribution has a compact support.

Assumption 3 *The support of P_X is compact and it is bounded as $\|x\|_\infty := \max_{1\le i\le d_x} |x_i| \le D_x$ $(\forall x \in \mathrm{supp}(P_X))$.*

From now on, we fix $\delta > 0$, and define $\hat{c}_\delta := \frac{4}{1-\delta}$. Then, we introduce the following constants corresponding to the norm bounds (Assumption 1): $\bar{R} := \sqrt{\hat{c}_\delta} R$, $\bar{R}_b := R_b/(1-\delta)$. Define a finite dimensional model with norm constrains as

$$\mathcal{F} := \{f(x) = (W^{(L)}\eta(\cdot) + b^{(L)}) \circ \cdots \circ (W^{(1)}x + b^{(1)}) \mid \|W^{(\ell)}\|_{\mathrm{F}} \le \bar{R},\ \|b^{(\ell)}\| \le \bar{R}_b\ (\ell = 1,\dots,L)\}.$$

This is used to approximate the infinite dimensional model characterized by the integral form. We can show that all functions in $\mathcal{F}$ has an infinite norm bound as $\|f\|_\infty \le \hat{R}_\infty$ where $\hat{R}_\infty$ is defined as

$$\hat{R}_\infty := \bar{R}^L D_x + \textstyle\sum_{\ell=1}^L \bar{R}^{L-\ell}\bar{R}_b.$$

The proof is given in Appendix A.2 in the supplementary material. Because of this, we can derive the generalization error bound with respect to the population L_2-norm instead of the empirical L_2-norm.

In the following, we derive the generalization error bounds for the two estimators: the empirical risk minimizer and the Bayes estimator. Then, we also give some examples in which the generalization error is analyzed in details. Before we state the generalization error bounds, we prepare some notations. Let $\hat{G} = L\bar{R}^{L-1}D_x + \sum_{\ell=1}^L \bar{R}^{L-\ell}$, and define $\hat{\delta}_{1,n}$, $\hat{\delta}_{2,n}$ as[3]

$$\hat{\delta}_{1,n} = \sum_{\ell=2}^{L} 2\sqrt{\hat{c}_\delta^{L-\ell}} R^{L-\ell+1}\sqrt{\lambda_\ell},$$

$$\hat{\delta}_{2,n}^2 = \frac{2}{n}\sum_{\ell=1}^{L} m_\ell m_{\ell+1} \log_+\left(1 + \frac{4\sqrt{2}\hat{G}\max\{\bar{R},\bar{R}_b\}\sqrt{n}}{\sigma\sqrt{\sum_{\ell=1}^L m_\ell m_{\ell+1}}}\right).$$

Roughly speaking, $\hat{\delta}_{1,n}$ corresponds to a finite dimensional approximation error (bias term) and $\hat{\delta}_{2,n}$ corresponds to the amount of deviation of the estimators in the finite dimensional model (variance term).

4.1 Empirical risk minimization

In this section, we define the empirical risk minimizer and investigate its generalization error. Let the empirical risk minimizer be $\widehat{f}$:

$$\widehat{f} := \operatorname*{argmin}_{f\in\mathcal{F}} \sum\nolimits_{i=1}^n (y_i - f(x_i))^2.$$

Note that there exists at least one minimizer because the parameter set corresponding to $\mathcal{F}$ is a compact set and η is a continuous function. $\widehat{f}$ needs not necessarily be the exact minimizer but it could be an approximated minimizer. We, however, assume $\widehat{f}$ is the exact minimizer for theoretical simplicity. In practice, the empirical risk minimizer is obtained by using the back-propagation technique. The regularization for the norm of the weight matrices and the bias terms are implemented by using the L_2-regularization and the drop-out techniques.

Our main result about generalization error analysis for the empirical risk minimizer is given in the following theorem.

Theorem 1 *For any $\delta > 0$ and $\lambda_\ell > 0$, suppose that*

$$m_\ell \ge 5N_\ell(\lambda_\ell)\log\left(32N_\ell(\lambda_\ell)/\delta\right) \quad (\ell = 2,\dots,L). \tag{6}$$

Then, there exists universal constants C_1 such that, for any $r > 0$ and $\tilde{r} > 1$,

$$\|\widehat{f} - f^{\circ}\|_{L_2(P_X)}^2 \le C_3\left\{\tilde{r}\hat{\delta}_{1,n}^2 + (\sigma^2 + \hat{R}_\infty^2)\hat{\delta}_{2,n}^2 + \frac{(\hat{R}_\infty^2 + \sigma^2)}{n}\left[\log_+\left(\frac{\sqrt{n}}{\min\{\sigma/\hat{R}_\infty, 1\}}\right) + r\right]\right\}$$

with probability $1 - \exp\left(-\frac{n\hat{\delta}_{1,n}^2(\tilde{r}-1)^2}{11\hat{R}_\infty^2}\right) - 2\exp(-r)$ for every $r > 0$ and $\tilde{r}' > 1$.

[3] We define $\log_+(x) = \max\{1, \log(x)\}$.

The proof is given in Appendix C in the supplementary material. It is easily checked that the third term of the right side ($\frac{(\hat{R}_\infty^2+\sigma^2)}{n}\big[\log_+\left(\frac{\sqrt{n}}{\min\{\sigma/\hat{R}_\infty,1\}}\right)+r\big]$) is smaller than the first two terms, therefore the generalization error bound can be simply evaluated as

$$\|\widehat{f}-f^{\mathrm{o}}\|_{L_2}^2 = O_p(\hat{\delta}_{1,n}^2+\hat{\delta}_{2,n}^2).$$

The first term $\hat{\delta}_{1,n}$ represents the bias that is induced by approximating f^{o} by the finite dimensional model $\mathcal{F}$. The inequality (6) represents a condition on the width m_ℓ under which the finite dimensional model $\mathcal{F}$ can approximate the true function f^{o} with accuracy $\hat{\delta}_{1,n}$. We can see that, as λ_ℓ decreases, the required width of the internal layer m_ℓ increases by the condition (6). Since $N_\ell(\lambda)$ is a decreasing function with respect to λ, a larger model is required to approximate the true function with smaller approximation error (small $\hat{\delta}_{1,n}$). On the other hand, $\hat{\delta}_{2,n}$ indicates the estimation error (or variance) to find the best element in the finite dimensional model $\mathcal{F}$. Since small λ_ℓ leads to large m_ℓ, the deviation $\hat{\delta}_{2,n}$ in the finite dimensional model should increase. Therefore, we observe bias-variance trade-off between $\hat{\delta}_{1,n}$ and $\hat{\delta}_{2,n}$. A key notion that characterizes the bias-variance trade-off is the degree of freedom $N_\ell(\lambda_\ell)$ which expresses the "complexity" of the RKHS $\mathcal{H}_\ell$ in each layer. The degree of freedom of a complicated RKHS grows up faster than a simpler one as λ goes to 0. In other words, if the eigenvalues of the kernels decreases rapidly, then the degree of freedom gets smaller, and we achieve a better generalization by using a simpler model.

This is also informative in practice because, to determine the width of each layer, the degree of freedom gives a good guidance. That is, if the degree of freedom is small compared with the sample size, then we may increase the width. An estimate of the degree of freedom can be computed from the trained network by computing the Gram matrix corresponding to the kernel induced from the trained network (where the kernel is defined by the finite sum instead of the integral form) and using the eigenvalue of the Gram matrix (see Figure 3).

To obtain the best generalization error bound, $(\lambda_\ell)_{\ell=1}^L$ should be tuned to balance the bias-variance terms (and accordingly $(m_\ell)_{\ell=2}^L$ should also be fine-tuned). The examples of the best achievable generalization error will be shown in Section 4.3.

4.2 Bayes estimator

In this section, we formulate a Bayes estimator and derive its generalization error. To define the Bayes estimator, we just need to specify the prior distribution. Let $\mathcal{B}_d(C)$ be the ball in the Euclidean space $\mathbb{R}^d$ with radius $C>0$ ($\mathcal{B}_d(C)=\{x\in\mathbb{R}^d \mid \|x\|\le C\}$), and $\mathrm{U}(\mathcal{B}_d(C))$ be the uniform distribution on the ball $\mathcal{B}_d(C)$. Here, we employ uniform distributions on the model $\mathcal{F}$:

$$W^{(\ell)} \sim \mathrm{U}(\mathcal{B}_{m_{\ell+1}\times m_\ell}(\bar{R})),\quad b^{(\ell)} \sim \mathrm{U}(\mathcal{B}_{m_{\ell+1}}(\bar{R}_b)).$$

In practice, the Gaussian distribution is also employed instead of the uniform distribution, but, for theoretical simplicity, we decided to analyze the uniform prior distribution.

The prior distribution on the parameters $(W^{(\ell)},b^{(\ell)})_{\ell=1}^L$ induces a distribution of the function f in the space of continuous functions endowed with the Borel algebra corresponding to the $L_\infty(\mathbb{R}^{d_{\mathrm{x}}})$-norm. We denote by Π the induced distribution. Using the prior, the posterior distribution is defined via the Bayes principle:

$$\Pi(\mathrm{d}f|D_n) = \frac{\exp\{-\sum_{i=1}^n \frac{(y_i-f(x_i))^2}{2\sigma^2}\}\Pi(\mathrm{d}f)}{\int \exp\{-\sum_{i=1}^n \frac{(y_i-f'(x_i))^2}{2\sigma^2}\}\Pi(\mathrm{d}f')}.$$

Although we do not pursue the computational issue of the Bayesian deep learning here, see, for example, Hernandez-Lobato and Adams (2015); Blundell et al. (2015) for practical algorithms. The following theorem gives how fast the Bayes posterior contracts around the true function.

Theorem 2 *Fix arbitrary $\delta>0$ and $\lambda_\ell>0$ ($\ell=1,\dots,L$), and suppose that the condition (6) on m_ℓ is satisfied. Then, for all $r\ge 1$, the posterior tail probability can be bounded as*

$$\begin{aligned}
&\mathrm{E}_{D_n}\Big[\Pi(f:\|f-f^{\mathrm{o}}\|_{L_2(P_X)} \\
&\qquad \ge (\hat{\delta}_{1,n}+\sigma\hat{\delta}_{2,n})r\sqrt{\max\{12,33\tfrac{\hat{R}_\infty^2}{\sigma^2}\}}|D_n)\Big] \\
&\le \exp\left[-n\hat{\delta}_{1,n}^2\tfrac{\min\{(r^2-1)^2,r^2-1\}}{11\hat{R}_\infty^2}\right] \\
&\qquad + 12\exp\left(-n(\hat{\delta}_{1,n}+\sigma\hat{\delta}_{2,n})^2\frac{r^2}{8\sigma^2}\right).
\end{aligned}$$

The proof is given in Appendix B in the supplementary material. Roughly speaking this theorem indicates that the posterior distribution concentrates in the distance $\hat{\delta}_{1,n}+\sigma\hat{\delta}_{2,n}$ from the true function f^{o}. The tail probability is sub-Gaussian and thus the posterior mass outside the distance $\hat{\delta}_{1,n}+\sigma\hat{\delta}_{2,n}$ from the true function rapidly decrease. Here we again observe that there appears bias-variance trade-off between $\hat{\delta}_{1,n}$ and $\hat{\delta}_{2,n}$. This can be understood essentially in the same way as the empirical risk minimization.

4.3 Examples

Here, we give some examples of the generalization error bound. We have seen that both of the empirical

risk minimizer and the Bayes estimators have a simplified generalization error bound as $\|\widehat{f} - f^{\mathrm{o}}\|_{L_2(P_X)}^2 = O_p\left(L\sum_{\ell=2}^{L}\bar{R}^{L-\ell+1}\lambda_\ell + \sum_{\ell=1}^{L}\frac{m_\ell m_{\ell+1}}{n}\log(n)\right)$, by supposing σ, $\hat{R}_\infty$ and $\sqrt{\hat{c}_\delta^L}R^L$ are in constant order. We evaluate the bound under the best choice of m_ℓ balancing the bias-variance trade-off.

Finite dimensional internal layer If all RKHSs are finite dimensional, say m_ℓ^*-dimensional. Then $N_\ell(\lambda) \leq m_\ell^*$ for all $\lambda \geq 0$. Therefore, by setting $\lambda_\ell = 0$ $(\forall \ell)$, the generalization error bound is obtained as

$$\|\widehat{f} - f^{\mathrm{o}}\|_{L_2(P_X)}^2 \lesssim \frac{\sigma^2 + \hat{R}_\infty^2}{n}\sum_{\ell=1}^{L} m_\ell^* m_{\ell+1}^* \log(n), \quad (7)$$

where we omitted the factors depending only on $\log(\bar{R}\bar{R}_b\hat{G})$. Note thatthis convergence rate is solely dependent on the number of parameters. This is much faster than the existing bounds that utilize the Rademacher complexity because their bounds are $O(1/\sqrt{n})$. This result matches more precise arguments for a finite dimensional 3-layer neural network based on asymptotic expansions (Fukumizu, 1999; Watanabe, 2001).

Polynomial decreasing rate of eigenvalues We assume that the eigenvalue $\mu_j^{(\ell)}$ decays in polynomial order as

$$\mu_j^{(\ell)} \leq a_\ell j^{-\frac{1}{s_\ell}}, \quad (8)$$

for a positive real $0 < s_\ell < 1$ and $a_\ell > 0$. This is a standard assumption in the analysis of kernel methods (Caponnetto and de Vito, 2007; Steinwart and Christmann, 2008), and it is known that this assumption is equivalent to the usual covering number assumption (Steinwart et al., 2009). For small s_ℓ, the decay rate is fast and it is easy to approximate the kernel by another one corresponding to a finite dimensional subspace. Therefore small s_ℓ corresponds to a simple model and large s_ℓ corresponds to a complicated model. In this setting, the degree of freedom is evaluated as

$$N_\ell(\lambda_\ell) \lesssim (\lambda_\ell / a_\ell)^{-s_\ell}. \quad (9)$$

Hence, we can show that $\lambda_\ell = a_\ell^{\frac{2s_\ell}{1+2s_\ell}} n^{-\frac{1}{1+2s_\ell}}$ gives the optimal rate, and we obtain the generalization error bound as

$$\|\widehat{f} - f^{\mathrm{o}}\|_{L_2(P_X)}^2 \lesssim \frac{d_{\mathrm{x}}^2}{n}\log(n) + L\sum_{\ell=2}^{L}(\bar{R}\vee 1)^{2(L-\ell+1)} a_\ell^{\frac{2s_\ell}{1+2s_\ell}} n^{-\frac{1}{1+2s_\ell}} \log(n), \quad (10)$$

where we omitted the factors depending on $s_\ell, \log(\bar{R}\bar{R}_b\hat{G})$, σ^2 and $\hat{R}_\infty$. This indicates that the complexity s_ℓ of the RKHS affects the convergence rate directly. As expected, if the RKHSs are simple (that is, $(s_\ell)_{\ell=2}^L$ are small), we obtain faster convergence.

One internal layer: kernel method Finally, we consider a simple but important situation in which there is only one internal layer ($L = 2$). In this setting, we only need to adjust m_2 because the dimensions of input and output are fixed as $m_1 = d_{\mathrm{x}}$ and $m_3 = 1$. We assume that the same condition (8) for $\ell = 2$. Then, it can be seen that the optimal width gives the following generalization error bound:

$$\|\widehat{f} - f^{\mathrm{o}}\|_{L_2(P_X)}^2 \lesssim ((\bar{R}\vee 1)^2 a_2^{\frac{s_2}{1+s_2}})(d_{\mathrm{x}}+1)^{\frac{1}{1+s_2}} n^{-\frac{1}{1+s_2}} \log(n).$$

This convergence rate is equivalent to the minimax optimal convergence rate of the kernel ridge regression (Caponnetto and de Vito, 2007; Steinwart et al., 2009) (up to constant and $\log(n)$ factors). It is known that the kernel method corresponds to the 3-layer neural network with an infinite dimensional internal layer. In that sense, our analysis includes that of kernel methods, and we can say that the deep neural network is a method that constructs an optimal kernel in a layer-wise manner.

5 Numerical experiments

Here, we give some numerical experiments to see connections between our theories and practical situations.

MNIST First, we investigate the relations between the width of each layer and the classification performance on MNIST dataset[4]. We constructed 5-layer convolutional neural network (CNN) which consists of 3 convolution layers and 2 fully connected layers. We changed the number of channels in the second and third convolution layers from 16 to 256 for the second layer and from 16 to 1024 for the third layer. The result is depicted in Figure 2. We see that for smallest widths $(16, 16)$, the classification accuracy is the worst. This means that the network is so simple that there appears large bias. On the other hand, for the large network $(254, 1024)$, the test accuracy is inferior to the best one. This is due to large variance.

CIFAR-10 We next conduct experiments on CIFAR-10 dataset (Krizhevsky, 2009) that contains 50,000 training color images of size 32×32 with 10-classes. We applied a VGG-type network (Simonyan and Zisserman, 2014) with 8 convolution layers, 3 max-pooling layers and 2 fully-connected layers. We constructed kernel functions for this

[4] http://yann.lecun.com/exdb/mnist/

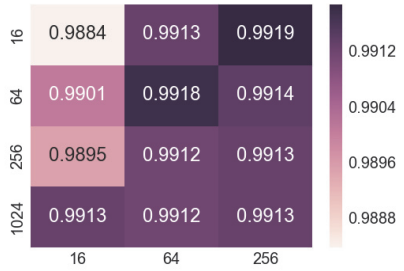

Figure 2: Test classification accuracy for different channel sizes in the second and third layers of CNN in MNIST dataset. Horizontal: second layer, vertical: third layer.

network where each feature in the feature space corresponds to each channel, and we computed the degree of freedom $N_\ell(\lambda)$ using the validation data. The degree of freedom for each convolution layer is depicted in Figure 3. We see that the degree of freedom increases moderately as $\lambda \to 0$. This means that the eigenvalues of the kernels decrease rapidly. Therefore, the eigenvalue decay rate assumption in Eq. (8) is justified, and because of this phenomenon the effective dimension of the network is less than the actual number of parameters.

6 Relations to existing work

The sample complexity of deep neural network has been extensively studied by analyzing its Rademacher complexity. For example, Bartlett (1998) characterized the generalization error of a 3-layer neural network by the norm of the weight vectors instead of the number of parameters. Koltchinskii and Panchenko (2002) studied more general deep neural network and derived its generalization error bound of deep neural network under a norm constraint. More recently, Neyshabur et al. (2015) analyzed the Rademacher complexity of the deep neural network based on generalized norms of the weight matrix ($(W^{(\ell)})_{\ell=1}^L$ in our paper). Sun et al. (2015) also derived the Rademacher complexity and showed the generalization error under

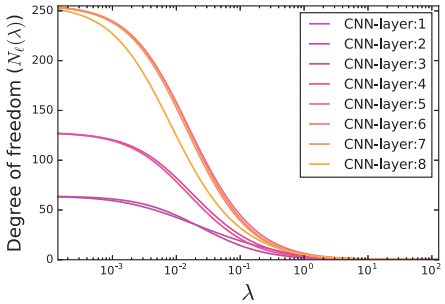

Figure 3: The degree of freedom against λ for each layer.

a large margin assumption. Basically, these studies have derived generalization error bounds depending on the properties of the parameters, and the properties of the data distributions are not much taken into account. On the other hand, our analysis involves such information as the degree of freedom, and thus can obtain a tighter bounds.

Analysis of the bias-variance trade-off in three layer neural network from the kernel point of view has been investigated by Bach (2017a). However, only shallow network is analyzed. Moreover, the loss function is not assumed to be strongly convex, and thus the obtained rate is not faster than $O(1/\sqrt{n})$.

Another important topic for the analysis of the generalization ability is VC-dimension analysis (Bartlett et al., 1998; Karpinski and Macintyre, 1997; Goldberg and Jerrum, 1995). However, VC-dimension is a notion independent of the input distributions. Here again, we note that the degree of freedom considered in our paper depends on the input distribution and is more data specific, and thus it gives a tighter bound and could be practically more useful.

7 Conclusion

In this paper, we proposed to use the integral form of deep neural network for generalization error analysis, and based on that, we derived the generalization error bound of the empirical risk minimizer and the Bayes estimator. The integral form enabled us to define an RKHS in each layer, and import the theoretical techniques developed in kernel methods into the analysis of deep learning. In particular, we defined the degree of freedom of each RKHS and showed that the approximation error between a finite dimensional model and the integral form can be characterized by the degree of freedom. In addition to the approximation error, we also derived the estimation error in the finite dimensional model. We have observed that there appears bias-variance trade-off depending on the size of the finite dimensional model, and the best choice of the size is characterized by the degree of freedom.

Our theoretical investigation would be particularly useful to determine the optimal widths of the internal layers. We believe this study opens up a new direction of a series of theoretical analyses of deep learning.

Acknowledgement

TS was partially supported by MEXT Kakenhi (25730013, 25120012, 26280009, 15H01678 and 15H05707), JST-PRESTO and JST-CREST.

References

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68: 337–404, 1950.

F. Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017a.

F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18(21):1–38, 2017b.

P. Bartlett, O. Bousquet, and S. Mendelson. Local Rademacher complexities. *The Annals of Statistics*, 33:1487–1537, 2005.

P. L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998.

P. L. Bartlett, V. Maiorov, and R. Meir. Almost linear VC-dimension bounds for piecewise polynomial networks. *Neural Computation*, 10(8):2159–2173, 1998.

M. Bianchini and F. Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE Transactions on Neural Networks and Learning Systems*, 25(8): 1553–1565, 2014.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, 2015.

A. Caponnetto and E. de Vito. Optimal rates for regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

N. Cohen and A. Shashua. Convolutional rectifier networks as generalized tensor decompositions. In *Proceedings of the 33th International Conference on Machine Learning*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 955–963, 2016.

N. Cohen, O. Sharir, and A. Shashua. On the expressive power of deep learning: A tensor analysis. In *Proceedings of the 29th Annual Conference on Learning Theory*, pages 698–728, 2016.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, 1989.

R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In *Proceedings of the 29th Annual Conference on Learning Theory*, pages 907–940, 2016.

K. Fukumizu. Generalization error of linear neural networks in unidentifiable cases. In *Proceedings of the 10th International Conference on Algorithmic Learning Theory*, pages 51–62. Springer, 1999.

S. Ghosal, J. K. Ghosh, and A. W. van der Vaart. Convergence rates of posterior distributions. *The Annals of Statistics*, 28(2):500–531, 2000.

E. Giné and V. Koltchinskii. Concentration inequalities and asymptotic results for ratio type empirical processes. *The Annals of Probability*, 34(3):1143–1216, 2006.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, 2011.

P. W. Goldberg and M. R. Jerrum. Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, 18(2-3):131–148, 1995.

J. M. Hernandez-Lobato and R. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1861–1869, 2015.

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

M. Karpinski and A. Macintyre. Polynomial bounds for VC dimension of sigmoidal and general pfaffian neural networks. *Journal of Computer and System Sciences*, 54(1):169–176, 1997.

V. Koltchinskii. Local Rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656, 2006.

V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30 (1):1–50, 2002.

A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105, 2012.

S. Mendelson. Improving the sample complexity using global data. *IEEE Transactions on Information Theory*, 48(7):1977–1991, 2002.

G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural

networks. In *Advances in Neural Information Processing Systems 27*, pages 2924–2932. 2014.

V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, pages 807–814, 2010.

B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. In *Proceedings of the 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 1376–1401, 2015.

B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in Neural Information Processing Systems 29*, pages 3360–3368. 2016.

I. Safran and O. Shamir. Depth separation in ReLU networks for approximating smooth non-linear functions. *arXiv preprint arXiv:1610.09887*, 2016.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

S. Sonoda and N. Murata. Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 2015.

I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 2008.

I. Steinwart, D. Hush, and C. Scovel. Optimal rates for regularized least squares regression. In *Proceedings of the 22nd Annual Conference on Learning Theory*, pages 79–93, 2009.

S. Sun, W. Chen, L. Wang, and T.-Y. Liu. Large margin deep neural networks: theory and algorithms. *arXiv preprint arXiv:1506.05232*, 2015.

A. W. van der Vaart and J. H. van Zanten. Rates of contraction of posterior distributions based on Gaussian process priors. *The Annals of Statistics*, 36(3):1435–1463, 2008.

A. W. van der Vaart and J. H. van Zanten. Information rates of nonparametric Gaussian process methods. *Journal of Machine Learning Research*, 12: 2095–2119, 2011.

S. Watanabe. Learning efficiency of redundant neural networks in bayesian estimation. *IEEE Transactions on Neural Networks*, 12(6):1475–1486, 2001.