Generalization error theory of deep learning and its application to model analysis

Taiji Suzuki (The University of Tokyo/RIKEN-AIP)*

Abstract

In this talk, we overview the recent developments of deep learning theory especially from the point of view of generalization and representation ability, and give some generalization error analysis from kernel methods and its applications to the model determination analysis. Along with rapid development of deep learning applications, its theoretical analysis has been developed extensively these days. In the first part, we overview the recent progress of deep learning theories. Second, we show what kind of quantity determines the generalization error. To do so, we define an intrinsic dimensionality from a kernel method perspective. We also show its application to model compression. Finally, we analyze representation ability of deep learning using wavelet analyses.

1. Introduction

This article gives an overview of our work about model compression and generalization. Along with the rapid development of the deep learning techniques, its network structure is getting extensively complicated. For example, SegNet [3] has skip connections, ResNet [12] and its variants [15, 5] also possess several skip connections. In addition to the model structure, the model size is getting larger, which prevents us to implement deep neural networks in edge-computing devices for such applications as smart phone services, autonomous vehicle driving and drone control.

To overcome this difficulty, model compression techniques have been studied extensively in the literature. One approach is pruning by an explicit regularization, such as ℓ_1 and ℓ_2 penalization during training [18, 29, 26, 13]. A similar effect can be realized by an implicit randomized regularization, such as DropConnect [25], which randomly removes connections during the training phase. The factorization method performs a matrix/tensor decomposition of the weight matrices to reduce the number of parameters [8, 9]. Information redundancy can be reduced by a quantization technique that expresses the network by smaller bit variable type or hash tables [4, 11]. More closely related ones are ThiNet [20] and Net-Trim [1] which prune the network weight so that the behaviors of the internal layers of the pruned network are as close as possible to those of the original network. [7] is quite close to ours, but it's theoretical support is not satisfactory. In particular, the suggested way of the best subset selection is just a random choice. [27] proposed parameter sharing technique to reduce redundant parameters based on similarity between the weights. A big issue in the literature is that

This is a joint work with †Hiroshi Abe, ‡Tomoya Murata, *Shingo Horiuchi, ‡Kotaro Ito, ‡Tokuma Wachi, *So Hirai, ‡Masatoshi Yukishima, and *Tomoaki Nishimura: †iPride Co., Ltd, ‡NTT DATA Mathematical Systems Inc., *NTT Data Corporation. This work was partially supported by MEXT KAKENHI (25730013, 25120012, 26280009, 15H05707 and 18H03201), JST-PRESTO (JPMJPR14E4), and JST-CREST (JPMJCR14D7, JPMJCR1304).
2010 Mathematics Subject Classification: MSC-62G08, MSC-68T01.
Keywords: machine learning, deep learning, generalization error analysis, kernel method.
* e-mail: taiji@mist.i.u-tokyo.ac.jp
web: http://ibis.t.u-tokyo.ac.jp/suzuki/

only few of them (e.g., Net-Trim [1]) are supported by statistical learning theory. In particular, it has been unclear what kind of quantity controls the compression ability. Another big issue is that the above mentioned methods can not be trivially applied to the recently developed networks with complicated structures such as skip connections like ResNet and SegNet.

In this article, we develop a new simple network compression method that is applicable to networks with complicated structures, and give theoretical support to explain what quantity controls the compression ability. The theoretical analysis is applicable not only to our method but also to the existing methods. Almost all of the existing methods try to find a smaller network structure that approximates only the "output" from each layer as well as possible. In contrast, our method also deals with the "input" to each layer. The information of the input is exploited as a covariance matrix, and redundant nodes are discarded on the basis of that information. It can be applied even if the "outputs" are split into several branches. Moreover, by combining the information of both input and output, it achieves better accuracy.

We also develop a theoretical analysis to characterize the compression error by utilizing the notion of *degree of freedom*. The degree of freedom represents a kind of intrinsic dimensionality of the model. This quantity is determined by the *eigenvalues of the covariance matrix* calculated in each layer. Usually, we observe that the eigenvalue drops rapidly, which leads to low degree of freedom. Because of this, we can compress the network effectively even though only the input information is used. Behind the theory, there is essentially a connection to the *kernel quadrature rule* [2]. In addition to the model compression ability analysis, we also develop a generalization error analysis. Finally, we conduct extensive numerical experiments to show the superiority of our method and give experimental verification of our theory.

2. Model compression problem and its algorithm

Suppose that the training data $D_{\mathrm{tr}} = \{(x_i, y_i)\}_{i=1}^{n_{\mathrm{tr}}}$ are observed, where $x_i \in \mathbb{R}^{d_x}$ and y_i could be a real number for regression ($y_i \in \mathbb{R}$) or a binary label for binary classification ($y_i \in \{\pm 1\}$). The distribution of X is denoted by $P_{\mathcal{X}}$. The training data are independently identically distributed. To train the appropriate relationship between x and y, we construct a deep neural network model as $f(x) = (W^{(L)}\eta(\cdot)+b^{(L)})\circ\cdots\circ(W^{(1)}x+b^{(1)})$, where $W^{(\ell)} \in \mathbb{R}^{m_{\ell+1}\times m_\ell}$, $b^{(\ell)} \in \mathbb{R}^{m_{\ell+1}}$ ($\ell = 1, \ldots, L$), and $\eta : \mathbb{R} \to \mathbb{R}$ is an activation function (here, the activation function is applied in an element-wise manner; for a vector $x \in \mathbb{R}^d$, $\eta(x) = (\eta(x_1), \ldots, \eta(x_d))^\top$). Furthermore, m_ℓ is the width of the ℓ-th layer such that $m_{L+1} = 1$ (output) and $m_1 = d_x$ (input). Let $\widehat{f}$ be a trained network obtained from a training data $D_{\mathrm{tr}} = \{(x_i, y_i)\}_{i=1}^{n_{\mathrm{tr}}}$. Accordingly, its parameters are denoted by $(\hat{W}^{(\ell)}, \hat{b}^{(\ell)})_{\ell=1}^{L}$, i.e., $\widehat{f}(x) = (\hat{W}^{(L)}\eta(\cdot)+\hat{b}^{(L)})\circ\cdots\circ(\hat{W}^{(1)}x+\hat{b}^{(1)})$, and the output of its internal layer (before activation) is denoted by $\hat{F}_\ell(x) = (\hat{W}^{(\ell)}\eta(\cdot)+\hat{b}^{(\ell)})\circ\cdots\circ(\hat{W}^{(1)}x+\hat{b}^{(1)})$. Here, we do not specify how to train the network $\widehat{f}$. Any learning method for training $\widehat{f}$ is valid for the following argument to be true. It might be the empirical risk minimizer, the Bayes estimator, or another estimator. We want to compress the trained network $\widehat{f}$ to another smaller network $f^\sharp$ having widths $(m_\ell^\sharp)_{\ell=1}^{L}$ which are as small as possible.

2.1. New model compression algorithm

To compress the trained network $\widehat{f}$, we propose a simple strategy called *Spectral-Pruning*. The method works in a layer-wise manner. The main idea of the method is to find the *most informative subset* of the nodes where the amount of information is

measured by how the selected nodes can explain the other nodes in the layer. If some nodes are heavily correlated to each other, then only one of them should be selected. The information redundancy can be computed by solving a simple regression problem, and requires only a covariance matrix. We do not need to solve some specific nonlinear optimization problem as in [18, 29, 26, 1]. Our method can be executed by only using the input to the layer. We call such an approach *input aware* one. On the other hand, it can also make use of the output from the layer as in the most existing methods. We call such approaches *output aware* ones. Another important characteristics of our method is to incorporate the distribution of the data while some existing pruning techniques try to approximate the parameter itself and is independent from the data distribution.

2.2. Algorithm description

Let $\phi(x) = \eta(\hat{F}_{\ell-1}(x)) \in \mathbb{R}^{m_\ell}$ be the input to the ℓ-th layer, and let $\phi_J(x) = (\phi_j(x))_{j\in J} \in \mathbb{R}^{|J|}$ be a subvector of $\phi(x)$ corresponding to an index set $J \in [m_\ell]^{|J|}$ where $[m] := \{1, \dots, m\}$. Basically, the strategy is to recover $\phi(x)$ from $\phi_J(x)$ as accurately as possible. To do so, we solve the following optimization problem:

$$\hat{A}_J = \operatorname*{argmin}_{A\in\mathbb{R}^{m_\ell\times|J|}} \widehat{\mathrm{E}}[\|\phi - A\phi_J\|^2] + \|A\|_w^2, \tag{1}$$

where $\widehat{\mathrm{E}}[\cdot]$ is the expectation with respect to the empirical distribution ($\widehat{\mathrm{E}}[f] = \frac{1}{n}\sum_{i=1}^{n_{\mathrm{tr}}} f(x_i)$) and $\|A\|_w^2 = \mathrm{Tr}[A\mathrm{I}_w A^\top]$ for a regularization parameter $w \in \mathbb{R}_+^{|J|}$ and $\mathrm{I}_w = \mathrm{diag}\,(w)$. The optimal solution $\hat{A}_J$ can be explicitly expressed by utilizing the *(non-centered) covariance matrix* in the ℓ-th layer of the trained network $\widehat{f}$ which is defined as $\widehat{\Sigma} := \widehat{\Sigma}_{(\ell)} = \frac{1}{n}\sum_{i=1}^{n} \eta(\hat{F}_{\ell-1}(x_i))\eta(\hat{F}_{\ell-1}(x_i))^\top$, defined on the empirical distribution (here, we omit the layer index ℓ for notational simplicity). Accordingly, let $\widehat{\Sigma}_{I,J} \in \mathbb{R}^{|I|\times|J|}$ be the submatrix of $\widehat{\Sigma}$ for index sets $I \subset [m_\ell]^{|I|}, J \subset [m_\ell]^{|J|}$ such that $\widehat{\Sigma}_{I,J} = (\widehat{\Sigma}_{i,j})_{i\in I, j\in J}$. Let $F = \{1, \dots, m_\ell\}$ be the full index set. By noting that $\widehat{\mathrm{E}}[\phi\phi^\top] = \widehat{\Sigma}$ due to its definition, we can easily check that $\hat{A}_J = \widehat{\Sigma}_{F,J}(\widehat{\Sigma}_{J,J} + \mathrm{I}_w)^{-1}$. Hence, we can decode the full vector $\phi(x)$ from $\phi_J(x)$ as $\phi(x) \approx \hat{A}_J\phi_J(X) = \widehat{\Sigma}_{F,J}(\widehat{\Sigma}_{J,J} + \mathrm{I}_w)^{-1}\phi_J(X)$. Another approach is to directly approximate a specific "output" $z^\top\phi$ for a specific $z \in \mathbb{R}^{m_\ell}$ instead of approximating the "input" ϕ as Eq. (1). This can be realized by solving the following regression problem which we call an "output-aware" approach:

$$\hat{a}_J = \operatorname*{argmin}_{a\in\mathbb{R}^{|J|}} \widehat{\mathrm{E}}[\|z^\top\phi - a^\top\phi_J\|^2] + \|a^\top\|_w^2.$$

It can be easily checked that the optimal solution $\hat{a}_J$ is given as $\hat{a}_J = \hat{A}_J^\top z$. Therefore, an output aware compression can be recovered from the input aware method (1). In particular, the output to the next layer $\hat{W}^{(\ell)}\phi(x) (= \hat{W}^{(\ell)}\eta(\hat{F}_{\ell-1}(x)))$ can be approximated by $\hat{W}^{(\ell)}\phi(x) \simeq \hat{W}^{(\ell)}\hat{A}_J\phi_J(x)$.

Selecting optimal subindices Next, we aim to optimize J. Since the output to the next layer is multi-variate and we need to bound the approximation error of multiple outputs *uniformly* to reduce the approximation error in the entire network, we minimize the following quantity with respect to J: $L_w^{(\mathrm{A})}(J) = \max_{z\in\mathbb{R}^{m_\ell}:\|z\|\le 1} \min_{a\in\mathbb{R}^{m_\ell}} \widehat{\mathrm{E}}[(z^\top\phi - a^\top\phi_J)^2] + \|a^\top\|_w^2$. By considering this, our method works no matter what branches there exist. The right hand side is equivalent to $\|\widehat{\mathrm{E}}[(\phi - \hat{A}_J\phi_J)(\phi - \hat{A}_J\phi_J)^\top] + \hat{A}_J\mathrm{I}_w\hat{A}_J^\top\|_{\mathrm{op}}$, where $\|\cdot\|_{\mathrm{op}}$ is the spectral norm (the maximum singular value of the matrix). By substituting the explicit formula of $\hat{A}_J$, this is further simplified as $L_w^{(\mathrm{A})}(J) = \|\widehat{\Sigma}_{F,F} -$

$\widehat{\Sigma}_{F,J}(\widehat{\Sigma}_{J,J}+\mathrm{I}_w)^{-1}\widehat{\Sigma}_{J,F}\|_{\mathrm{op}}$. To obtain the optimal J under a cardinality constraint $|J| \leq m_\ell^\sharp$ for a pre-specified width $m_\ell^\sharp$ of the compressed network, we propose to solve the following sparse subset selection problem:

$$\min_J \; L_w^{(\mathrm{A})}(J) \quad \text{s.t.} \quad J \in [m_\ell]^{m_\ell^\sharp}. \tag{2}$$

Let $\hat{J}$ be the optimal J that minimizes the objective. This optimization problem is NP-hard, but an approximate solution is obtained by the greedy algorithm since it is reduced to monotone submodular function maximization [17]. That is, we start from $J = \emptyset$, sequentially choose an element $j^* \in [m_\ell]$ that maximally reduces the objective $L_w^{(\mathrm{A})}$, and add this element j^* to J ($J \leftarrow J \cup \{j^*\}$) until $|J| = m_\ell^\sharp$ is satisfied.

An advantage of this approach is that it requires *only the covariance matrix*, and it is accomplished by purely linear algebraic procedures. Moreover, our method can be applied to a complicated network structure in which there are recurrent structures, several branches, or outputs from the internal layers that are widely distributed to several other units (e.g., skip connections).

Output aware method Suppose that there is an "important" subset of weight vectors, say $\mathcal{Z}_\ell \subset \mathbb{R}^{m_\ell}$, such that the output $z^\top\phi$ corresponding to $z \in \mathcal{Z}_\ell$ should be well approximated. Then it would be more effective to focus on approximating $z^\top\phi$ ($z \in \mathcal{Z}_\ell$) instead of all $z^\top\phi$ ($z \in \mathbb{R}^{m_\ell}$). Here, suppose that $\mathcal{Z}_\ell$ is a finite set, and let the weight matrix Z be the one each row which corresponds to each distinguish element in $\mathcal{Z}_\ell$: $Z_\ell = [z_1, \ldots, z_{|\mathcal{Z}_\ell|}]^\top$ where $z_j \in \mathcal{Z}_\ell$. If $\mathcal{Z}_\ell$ is not a finite set, we may set Z_ℓ as a projection matrix to the span of $\mathcal{Z}_\ell$. Then, we consider an objective $L_w^{(\mathrm{B})} := \max_{\|w\|\leq 1} \min_{a\in\mathbb{R}^{m_\ell}} \widehat{\mathrm{E}}[(w^\top Z_\ell\phi - a^\top\phi_J)^2] + \|a^\top\|_w^2$, which is equivalent to

$$L_w^{(\mathrm{B})}(J) = \|Z_\ell[\widehat{\Sigma}_{F,F} - \widehat{\Sigma}_{F,J}(\widehat{\Sigma}_{J,J}+\mathrm{I}_w)^{-1}\widehat{\Sigma}_{J,F}]Z_\ell^\top\|_{\mathrm{op}}.$$

A typical situation is to approximate the output $\hat{W}^{(\ell)}\phi$. In that situation, we may set $Z_\ell = \hat{W}^{(\ell)}$ which corresponds to $\mathcal{Z}_\ell = \{(\hat{W}^{(\ell)}_{j,:})^\top \mid j = 1, \ldots, m_{\ell+1}\}$.

Combination of input aware and output aware methods In our numerical experiments, we have found that only one of either input or output aware method does not give the best performance, but the combination of them achieved the best performance. Moreover, if the network has several branches, then it is not trivial which branches should be included in $\mathcal{Z}_\ell$ for the output aware method. In that situation, it is preferable to combine input aware and output aware methods instead of using only the output aware method. Therefore, we propose to take the *convex combination* of the both criteria given for a parameter $0 \leq \theta \leq 1$ as

$$\text{(Spectral-Pruning)} \qquad \min_J \; L_w^{(\theta)}(J) \quad \text{s.t.} \quad J \in [m_\ell]^{m_\ell^\sharp}. \tag{3}$$

2.3. Practical algorithm

Calculating the exact value of $L_w^{(\theta)}$ is computationally demanding for a large network because we need to compute the spectral norm. However, we do not need to obtain the exact solution for the problem (3) in practice, because, if we obtain a reasonable candidate that approximately achieves the optimal, then additional fine-tuning gives a much better network. Hence, instead of solving (3) directly, we upper bound $L_w^{(\mathrm{A})}$ and $L_w^{(\mathrm{B})}$ by replacing the operator norm in their definitions with trace, and minimize it as a practical variant of our method. By setting $w = \mathbf{0}$, the objective of the variational

method is reduced to $\mathrm{Tr}[(\theta \mathrm{I} + (1-\theta)R_{\mathcal{Z}})(\widehat{\Sigma}_{F,F} - \widehat{\Sigma}_{F,J}\widehat{\Sigma}_{J,J}^{-1}\widehat{\Sigma}_{J,F})]$. Then, the proposed optimization problem can be rearranged to the following problem:

$$\text{(Spectral-Pruning-2)} \quad \min_{J \subset \{1,\dots,m_\ell\}} |J| \ \text{ s.t. } \frac{\mathrm{Tr}[(\theta \mathrm{I} + (1-\theta) Z_\ell^\top Z_\ell)\widehat{\Sigma}_{F,J}\widehat{\Sigma}_{J,J}^{-1}\widehat{\Sigma}_{J,F}]}{\mathrm{Tr}[(\theta \mathrm{I} + (1-\theta) Z_\ell^\top Z_\ell)\widehat{\Sigma}_{F,F}]} \geq \alpha \qquad (4)$$

for a pre-specified $\alpha > 0$. Here, since the denominator in the constraint is the best achievable objective value of the numerator without cardinality constrain, α represents "information loss ratio." The index set J is restricted to a subset of $\{1, \dots, m_\ell\}$ that has no duplication. This problem is not only much simpler but also easier to implement than the original one (3). In our numerical experiments, we employed this simpler problem.

3. Compression accuracy and generalization error analysis

In this section, we give a theoretical guarantee of our model compression method. More specifically, we introduce a quantity called *degree of freedom* and show that it determines the approximation accuracy. Let $R > 0$ and $R_b > 0$ be upper bounds of the parameters, and define the norm constraint model as

$$\mathcal{F} := \{(W^{(L)}\eta(\cdot) + b^{(L)}) \circ \cdots \circ (W^{(1)}x + b^{(1)}) \mid \max_j \|W_{j,:}^{(\ell)}\| \leq R/\sqrt{m_\ell},\ \|b^{(\ell)}\|_\infty \leq R_b \ \},$$

where $W_{j,:}^{(\ell)}$ means the j-th column of the matrix $W^{(\ell)}$, $\|\cdot\|$ is the Euclidean norm and $\|\cdot\|_\infty$ is the ℓ_∞-norm. Here, we bound the approximation error induced by compressing the trained network $\widehat{f} \in \mathcal{F}$ into a smaller one $f^\sharp$. First, we make the following assumption.

Assumption 1. *We assume the following conditions on the activation function η which is satisfied by ReLU activation [22, 10].*

- *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: $|\eta(x) - \eta(x')| \leq \|x - x'\|$ for all $x, x' \in \mathbb{R}^d$ (for arbitrary d).*

3.1. Approximation error analysis

Recall that the empirical covariance matrix in the ℓ-th layer is denoted as $\widehat{\Sigma}_{(\ell)}$. Then, the degree of freedom is defined by

$$\hat{N}_\ell(\lambda) := \mathrm{Tr}[\widehat{\Sigma}_{(\ell)}(\widehat{\Sigma}_{(\ell)} + \lambda \mathrm{I})^{-1}] = \textstyle\sum_{j=1}^{m_\ell} \hat{\mu}_j^{(\ell)}/(\hat{\mu}_j^{(\ell)} + \lambda)$$

where $(\hat{\mu}_j^{(\ell)})_{j=1}^{m_\ell}$ are the eigenvalues of $\widehat{\Sigma}_{(\ell)}$. Let $(m_\ell^\sharp)_{\ell=1}^L$ denote the width of $f^\sharp$. The next theorem characterizes the approximation accuracy between $f^\sharp$ and $\widehat{f}$ on the basis of the degree of freedom with respect to the *empirical L_2-norm* $\|g\|_n^2 := \frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}} \|g(x_i)\|^2$ for a vector valued function g.

Theorem 1 (Compression rate via degree of freedom)**.** *Suppose that there exists $\tilde{J}_\ell \subset [m_{\ell+1}]$ such that $\mathcal{Z}_\ell = \{\hat{W}_{j,:}^{(\ell)} / \max_{1 \leq j' \leq m_{\ell+1}} \|\hat{W}_{j',:}^{(\ell)}\| \mid j \in \tilde{J}_\ell\}$. Let $\lambda_\ell > 0$ be*

$$\lambda_\ell = \inf\{\lambda \geq 0 \mid m_\ell^\sharp \geq 5\hat{N}_\ell(\lambda)\log(80\hat{N}_\ell(\lambda))\} \qquad (5)$$

*and the weight vector w for the regularization is defined by the "*leverage score*"; that is, $w_j = \frac{m_\ell^\sharp \lambda_\ell}{\hat{N}_\ell(\lambda_\ell)} \sum_{k=1}^{m_\ell} U_{\ell;j,k}^2 \hat{\mu}_k^{(\ell)}/(\hat{\mu}_k^{(\ell)} + \lambda_\ell)$ where $U_\ell = (U_{\ell;j,k})_{j,k}$ is the orthogonal matrix that diagonalizes $\widehat{\Sigma}_{(\ell)}$; $\widehat{\Sigma}_{(\ell)} = U_\ell \mathrm{diag}\left(\hat{\mu}_1^{(\ell)}, \dots, \hat{\mu}_{m_\ell}^{(\ell)}\right) U_\ell^\top$. Let $\alpha_{j,\ell} = \theta^{-1}$ $(j \notin$*

$\tilde{J}_\ell)$, 1 (*otherwise*) *and* $\zeta_{\ell,\theta} = \theta + (1-\theta)\|R_{\mathcal{Z}_\ell}\widehat{\Sigma}_\ell(\widehat{\Sigma}_\ell + \lambda_\ell \mathrm{I})^{-1}R_{\mathcal{Z}_\ell}\|_{\mathrm{op}}$. *Then, the solution* $\hat{A}_{\hat{J}}$ *obtained by Spectral-Pruning* (3) *satisfies*

$$\max_{1\le j\le m_{\ell+1}} \frac{\|\hat{W}^{(\ell)}_{j,:}\phi - \hat{W}^{(\ell)}_{j,:}\hat{A}_{\hat{J}}\phi_{\hat{J}}\|_n^2}{\alpha_{j,\ell}} \le \frac{4\zeta_{\ell,\theta}\lambda_\ell}{m_\ell}R^2. \tag{6}$$

Moreover, there exits a universal constant $\hat{c} > 0$ *such that the parameter of the compressed network satisfies the following norm bound:*

$$\|\hat{W}^{(\ell)}_{j,:}\hat{A}_{\hat{J}}\mathrm{diag}(w)^{1/2}\|^2 \le \hat{c}\frac{\lambda_\ell \alpha_{j,\ell}}{m_\ell}R^2 \tag{7}$$

Moreover, if we solve the optimization problem (3) *with an additional constraint* $\sum_{j\in J} w_j^{-1} \le \frac{5}{3}m_\ell\lambda_\ell^{-1}$ *for all* $1 \le \ell \le L$, *then the optimization problem is feasible and, the overall approximation error is bounded as*

$$\|\widehat{f} - f^\sharp\|_n \le \sum_{\ell=2}^L \bar{R}^{L-\ell+1}\sqrt{\alpha_{\max}\zeta_{\ell,\theta}\lambda_\ell}, \tag{8}$$

where $\alpha_{\max} = \max_{j,\ell}\{\alpha_{j,\ell}\}$ *and* $\bar{R} = \sqrt{\hat{c}\alpha_{\max}}R$.

It is basically proven using the techniques developed by [24]. This theorem indicates that the approximation error induced by the compression is directly controlled by the degree of freedom. Since the degree of freedom $\hat{N}_\ell(\lambda_\ell)$ is a monotonically decreasing function with respect to λ_ℓ, it becomes large as λ_ℓ is decreased to 0. The behavior of the eigenvalues determines how rapidly $\hat{N}_\ell(\lambda_\ell)$ increases as $\lambda_\ell \to 0$. We can see that if the eigenvalues $\hat{\mu}_1^{(\ell)} \ge \hat{\mu}_2^{(\ell)} \ge \ldots$ decrease rapidly, then λ_ℓ becomes small for a specific network size $m_\ell^\sharp$.

3.2. Generalization error analysis

So far, we have developed an approximation error bound with respect to the "empirical" L_2-distance. Here, we derive a generalization error bound for the compressed network, which is defined by the population L_2 distance. We see that there appears bias-variance trade-off induced by the network compression. For this purpose, we specify the data generation model. First, we consider a simple regression model:

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

where $f^{\mathrm{o}} : \mathbb{R}^{d_x} \to \mathbb{R}$ is the true function that we want to estimate, $(x_i)_{i=1}^{n_{\mathrm{tr}}}$ is independently identically distributed from $P_{\mathcal{X}}$, and $(\xi_i)_{i=1}^{n_{\mathrm{tr}}}$ is i.i.d. Gaussian noise with mean 0 and variance σ^2. A regression problem is considered for theoretical simplicity. Nearly the same discussion is applicable to classification problems with margin conditions such as Tsybakov's noise condition [21]. The *relative* generalization error of f is evaluated as $\mathrm{E}_{X,Y}[(Y - f(X))^2] - \mathrm{E}_{X,Y}[(Y - f^{\mathrm{o}}(X))^2] = \mathrm{E}[(f(X) - f^{\mathrm{o}}(X))^2] = \|f - f^{\mathrm{o}}\|_{L_2}^2$ where $\|\cdot\|_{L_2}$ is defined as $\|f\|_{L_2} = \sqrt{\mathrm{E}[f(X)^2]}$. Hence, we aim to bound $\|f^\sharp - f^{\mathrm{o}}\|_{L_2}^2$. The training error is denoted by $\hat{L}(f) := \frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}}(y_i - f(x_i))^2$. We assume (approximately) optimality of the trained network $\widehat{f}$ and the boundedness of the input as follows.

Assumption 2 (Optimality)**.** *There exists a constant* $\hat{\zeta} \ge 0$ *such that the following inequality holds almost surely:* $\hat{L}(\widehat{f}) \le \min_{f\in\mathcal{F}}\hat{L}(f) + \hat{\zeta}$.

Assumption 3. *The support of* $P_{\mathcal{X}}$ *is compact and its* ℓ_∞*-norm is bounded as* $\|x\|_\infty \le D_x$ $(\forall x \in \mathrm{supp}(P_{\mathcal{X}}))$.

Then, under the same setting as Theorem 1, we define the following constants corresponding to the norm bounds: $\hat{R}_\infty := \max\{\bar{R}^L D_x + \sum_{\ell=1}^L \bar{R}^{L-\ell}\bar{R}_b, \|f^{\mathrm{o}}\|_\infty\}$, $\hat{G} := L\bar{R}^{L-1}D_x + \sum_{\ell=1}^L \bar{R}^{L-\ell}$, where $\bar{R} = \sqrt{\hat{c}\alpha_{\max}}R$ and $\bar{R}_b = \sqrt{\hat{c}}R_b$ for the constants $\hat{c}$ and $\alpha_{\max}$ introduced in Theorem 1. To bound the generalization error, we introduce δ_1, δ_2 defined as[1], for $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_L) \in \mathbb{R}_+^L$ and $\boldsymbol{m'} = (m'_1, \dots, m'_L) \in [m_1] \times \cdots \times [m_L]$,

$$\delta_1 = \delta_1(\boldsymbol{\lambda}) = \sum_{\ell=2}^L \bar{R}^{L-\ell+1}\sqrt{\alpha_{\max}\zeta_{\ell,\theta}\lambda_\ell},\ \delta_2^2(\boldsymbol{m'}) = \frac{1}{n}\sum_{\ell=1}^L m'_\ell m'_{\ell+1}\log_+\Big(1 + \tfrac{4\sqrt{2}\hat{G}\max\{\bar{R},\bar{R}_b\}\sqrt{n}}{\sigma\wedge\hat{R}_\infty}\Big).$$

Under these notations, we obtain the following generalization error bound for the compressed network $f^\sharp$ with respect to the population L_2-norm $\|f^\sharp - f^{\mathrm{o}}\|_{L_2}^2$.

Theorem 2 (Generalization error bound of the compressed network)**.** *Suppose that Assumptions 1, 2 and 3 are satisfied. Consider a setting where $\theta = 1$ or $Z_\ell = \frac{\hat{W}^{(\ell)}}{\max_{1\le j'\le m_{\ell+1}}\|\hat{W}^{(\ell)}_{j',:}\|}$ for $\ell = 1, \dots, L$. Let $\lambda_\ell > 0$ $(\ell = 2, \dots, L)$ are the variables satisfying the condition* (5)*: $\lambda_\ell = \inf\{\lambda \ge 0 \mid m_\ell^\sharp \ge 5\hat{N}_\ell(\lambda)\log(80\hat{N}_\ell(\lambda))\}$, and assume that $f^\sharp$ satisfy the approximation error bound* (8) *with the norm bound* (7) *as given in Theorem 1. Let $R_{n,t} = \frac{(\hat{R}_\infty^2+\sigma^2)}{n}\left[\log_+\log_2\left[\frac{\sqrt{n}}{\frac{\sigma}{\hat{R}_\infty}\wedge 1}\right] + 1 + t + \sum_{\ell=2}^L \log(m_\ell)\right]$, and $\boldsymbol{m} = (m_1, \dots, m_L)$. Then, there exists a constant $C_1 > 0$ such that for all $t > 0$,*

$$\|f^\sharp - f^{\mathrm{o}}\|_{L_2}^2 \le C_1\Big\{\delta_1^2 + (\sigma^2 + \hat{R}_\infty^2)\delta_2^2(\boldsymbol{m}^\sharp) + \sigma\delta_1\delta_2(\boldsymbol{m}) + \min_{f\in\mathcal{F}}\|f - f^{\mathrm{o}}\|_{L_2}^2 + \hat{\zeta} + R_{n,t}\Big\}$$

uniformly over all choice of $\boldsymbol{m}^\sharp = (m_1^\sharp, \dots, m_L^\sharp)$ with probability $1 - 5e^{-t}$.

In a small noise situation $\sigma \simeq 0$, the main term becomes $\|f^\sharp - f^{\mathrm{o}}\|_{L_2}^2 \lesssim \delta_1^2 + \delta_2^2(\boldsymbol{m}^\sharp) \simeq (\sum_{\ell=2}^L\sqrt{\lambda_\ell})^2 + \frac{1}{n}\sum_{\ell=1}^L m_{\ell+1}^\sharp m_\ell^\sharp \log(n)$. The remaining terms are just residual terms. Actually, the term $R_{n,t}$ is basically $O(\sum_\ell \log(m_\ell)/n)$, which is much smaller than $\delta_1^2 + \delta_2^2$ and is thus negligible. By Theorem 1, δ_1 represents the approximation error between $\widehat{f}$ and $f^\sharp$; hence, it can be regarded as a *bias*. The second term $\delta_2(\boldsymbol{m}^\sharp)$ is the *variance* term that is induced by the sample deviation. Here, it should be noted that the variance term $\delta_2(\boldsymbol{m}^\sharp)$ depends only on the *size of the compressed network rather than the original network size*. On the other hand, a naive application of the theorem implies that $\|\widehat{f} - f^{\mathrm{o}}\|_{L_2}^2 \le \delta_2^2(\boldsymbol{m}) = \mathrm{O}\big(\frac{1}{n}\sum_{\ell=1}^L m_{\ell+1}m_\ell\log(n)\big)$ (here, the residual terms are omitted) which is much larger than $\delta_2^2(\boldsymbol{m}^\sharp)$ when $m_\ell^\sharp \ll m_\ell$. Therefore, the variance is reduced significantly by the model compression resulting in a much improved generalization error.

4. Numerical experiments

In this section, we conduct numerical experiments to show the effectiveness of the proposed method, and justify our theoretical analysis. As for our method, all experiments have been conducted by the practical variant (Spectral-Pruning-2 (Eq. (4))).

4.1. ImageNet

We apply our method to the ImageNet dataset [6]. We used the ILSVRC2012 dataset in ImageNet consisting of 1.3M training data and 50,000 validation data. Each image is annotated into one of 1,000 categories. We used a publicly available VGG-16

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

Table 1: Performance comparison on ImageNet dataset. Our proposed method is compared with APoZ-2 [14], SqueezeNet [16], and ThiNet [20]. Our method is indicated as "Spec-(type)."

Model	Top-1	Top-5	# Param.	FLOPs
Original VGG [23]	68.34%	88.44%	138.34M	30.94B
APoZ-2 [14]	70.15%	89.69%	51.24M	30.94B
SqueezeNet [16]	57.67%	80.39%	1.24M	1.72B
ThiNet-Conv [20]	69.80%	89.53%	131.44M	9.58B
ThiNet-GAP [20]	67.34%	87.92%	8.32M	9.34B
ThiNet-Tiny [20]	59.34%	81.97%	1.32M	2.01B
Spec-Conv ($\theta = 0.5$)	72.15%	91.06%	131.44M	22.13B
Spec-Conv-FC ($\theta = 1$)	68.66%	88.90%	45.77M	9.58B
Spec-GAP ($\theta = 0.5$)	67.55%	88.27%	8.31M	11.21B
Spec-Tiny ($\theta = 1$)	60.10%	82.89%	2.31M	2.07B
Spec-Conv2 ($\theta = 0.5$)	70.09%	89.82%	131.44M	9.58B
Spec-GAP2 ($\theta = 0.5$)	67.33%	87.99%	8.32M	9.34B
Spec-GAPe ($\theta = 0.5$)	67.78%	88.52%	8.25M	14.77B

network [23] as the original network. We applied our method to this network and compared it with existing state-of-the-art methods, namely APoZ [14], SqueezeNet [16], and ThiNet [20]. For fair comparison, we followed the same experimental settings as [20]. The results are summarized in Table 1. It summarizes the Top-1/Top-5 classification accuracies, the number of parameters (#Param), and the float point operations (FLOPs) to classify a single image. Our method is indicated by "Spec-(type)." In Spec-Conv, we applied our method only to the convolutional layers (it is not applied to the fully connected layers (FC)). The conv-layers are compressed gradually by solving Eq. (4) with $\alpha = 0.99$ a few times until the #Param becomes comparable to ThiNet-Conv which also applies the ThiNet method only to conv-layers. After each compression operation, we applied fine tuning. In our experiments, one or two iteration was sufficient to reach the comparable compression rate. Spec-Conv-FC compresses the FC layers as well as the conv-layers, whereas it is difficult to apply ThiNet to the FC layers because it is computationally too demanding. We employed $\theta = 1$ for Spec-Conv-FC. Spec-GAP is a method that replaces the FC layers of Spec-Conv with a global average pooling (GAP) layer [19, 28]. We chose α so that one time application of the compression achieves the same #Param as ThiNet-GAP which is also a method utilizing the GAP layer as the fully connected layer. We employed $\theta = 0.5$ for Spec-GAP. Spec-GAPe sets the parameter α in each layer as $\alpha_\ell = 0.9944^\ell$ for the ℓ-th layer. The other setting of Spec-GAPe is same as Spec-GAP. Spec-Tiny is a method where Eq. (4) with $\alpha = 0.97$ is performed to conv-layers several times until the #Params and FLOPs becomes comparable to that of ThiNet-Tiny. As for Spec-Conv2, we set the number of channels in each layer to be same as that of ThiNet-Conv. Similarly, in Spec-GAP2, we set the number of channels in each layer to be same as that of ThiNet-GAP.

APoZ shows favorable accuracy but this method can reduce the parameters in only non-convolutional layers. Thus, its applicability is limited; consequently, it does not reduce the FLOPs significantly. ThiNet is the most comparable method, but if the number of parameters is set to be equal, our method (especially Spec-Conv) yields better performance than it. We would like to remark that ThiNet (and existing methods) does not have a criterion to automatically determine the shape of compressed

network. On the other hand, our method may determine that through the degree of freedom or the formula (4).

5. Conclusion

We proposed a new model compression frame-work that utilizes both of "input" and "output", and showed that the *degree of freedom* characterizes the extent to which a trained network can be compressed. The algorithm is easily implemented and can be run in a layer-wise manner. There appeared bias and variance trade-off according to compression rate. The numerical experiments showed a favorable performance to the existing state-of-the-art methods despite its algorithmic simplicity.

References

[1] A. Aghasi, A. Abdi, N. Nguyen, and J. Romberg. Net-trim: Convex pruning of deep neural networks with performance guarantee. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3180–3189. Curran Associates, Inc., 2017.

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

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[4] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning*, pages 2285–2294, 2015.

[5] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng. Dual path networks. In *Advances in Neural Information Processing Systems*, pages 4470–4478, 2017.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[7] M. Denil, B. Shakibi, L. Dinh, N. De Freitas, et al. Predicting parameters in deep learning. In *Advances in neural information processing systems*, pages 2148–2156, 2013.

[8] M. Denil, B. Shakibi, L. Dinh, M. A. Ranzato, and N. de Freitas. Predicting parameters in deep learning. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2148–2156. Curran Associates, Inc., 2013.

[9] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1269–1277. Curran Associates, Inc., 2014.

[10] 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.

[11] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Y. He, X. Zhang, and J. Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*,

pages 1389–1397, 2017.

[14] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

[15] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[16] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[17] A. Krause and D. Golovin. Submodular function maximization, 2014.

[18] V. Lebedev and V. Lempitsky. Fast convnets using group-wise brain damage. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2554–2564, 2016.

[19] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[20] J.-H. Luo, J. Wu, and W. Lin. ThiNet: a filter level pruning method for deep neural network compression. In *International Conference on Computer Vision (ICCV)*, October 2017.

[21] E. Mammen and A. B. Tsybakov. Smooth discrimination analysis. *Ann. Statist.*, 27(6):1808–1829, 12 1999.

[22] 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.

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

[24] T. Suzuki. Fast generalization error bound of deep learning from a kernel perspective. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics (AISTATS2018)*, volume 84 of *Proceedings of Machine Learning Research*, pages 1397–1406, 2018.

[25] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus. Regularization of neural networks using dropconnect. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1058–1066, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[26] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li. Learning structured sparsity in deep neural networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2074–2082. Curran Associates, Inc., 2016.

[27] D. Zhang, H. Wang, M. Figueiredo, and L. Balzano. Learning to share: Simultaneous parameter tying and sparsification in deep learning. In *International Conference on Learning Representations*, 2018.

[28] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2921–2929. IEEE, 2016.

[29] H. Zhou, J. M. Alvarez, and F. Porikli. Less is more: Towards compact cnns. In *European Conference on Computer Vision*, pages 662–677. Springer, 2016.