

# Spectral Pruning: Compressing Deep Neural Networks via Spectral Analysis and its Generalization Error

Taiji Suzuki<sup>1,2,\*</sup>, Hiroshi Abe<sup>3,†</sup>, Tomoya Murata<sup>4</sup>, Shingo Horiuchi<sup>5</sup>, Kotaro Ito<sup>4</sup>, Tokuma Wachi<sup>4</sup>, So Hirai<sup>5</sup>, Masatoshi Yukishima<sup>4</sup> and Tomoaki Nishimura<sup>5,‡</sup>

<sup>1</sup>The University of Tokyo, Japan,

<sup>2</sup>Center for Advanced Intelligence Project, RIKEN, Japan,

<sup>3</sup>iPride Co., Ltd., Japan,

<sup>4</sup>NTT DATA Mathematical Systems Inc., Japan,

<sup>5</sup>NTT Data Corporation, Japan,

\*taiji@mist.i.u-tokyo.ac.jp, †abe@ipride.co.jp, ‡tomoaki.nishimura.jp@gmail.com

## Abstract

Compression techniques for deep neural network models are becoming very important for the efficient execution of high-performance deep learning systems on edge-computing devices. The concept of model compression is also important for analyzing the generalization error of deep learning, known as the compression-based error bound. However, there is still huge gap between a practically effective compression method and its rigorous background of statistical learning theory. To resolve this issue, we develop a new theoretical framework for model compression and propose a new pruning method called *spectral pruning* based on this framework. We define the “degrees of freedom” to quantify the intrinsic dimensionality of a model by using the eigenvalue distribution of the covariance matrix across the internal nodes and show that the compression ability is essentially controlled by this quantity. Moreover, we present a sharp generalization error bound of the compressed model and characterize the bias–variance tradeoff induced by the compression procedure. We apply our method to several datasets to justify our theoretical analyses and show the superiority of the proposed method.

## 1 Introduction

Currently, deep learning is the most promising approach adopted by various machine learning applications such as computer vision, natural language processing, and audio processing. Along with the rapid development of the deep learning techniques, its network structure is becoming considerably complicated. In addition to the model structure, the model size is also becoming larger, which prevents the implementation of deep neural network models in edge-computing devices for applications such as smartphone services, autonomous vehicle driving, and drone control. To overcome this problem, model compression techniques such as pruning, factorization [Denil *et al.*, 2013; Denton *et al.*, 2014], and

quantization [Han *et al.*, 2015] have been extensively studied in the literature.

Among these techniques, pruning is a typical approach that discards redundant nodes, e.g., by explicit regularization such as  $\ell_1$  and  $\ell_2$  penalization during training [Lebedev and Lempitsky, 2016; Wen *et al.*, 2016; He *et al.*, 2017]. It has been implemented as ThiNet [Luo *et al.*, 2017], Net-Trim [Aghasi *et al.*, 2017], NISP [Yu *et al.*, 2018], and so on [Denil *et al.*, 2013]. A similar effect can be realized by implicit randomized regularization such as DropConnect [Wan *et al.*, 2013], which randomly removes connections during the training phase. However, only few of these techniques (e.g., Net-Trim [Aghasi *et al.*, 2017]) are supported by statistical learning theory. In particular, it unclear which type of quantity controls the compression ability. On the theoretical side, compression-based generalization analysis is a promising approach for measuring the redundancy of a network [Arora *et al.*, 2018; Zhou *et al.*, 2019]. However, despite their theoretical novelty, the connection of these generalization error analyses to practically useful compression methods is not obvious.

In this paper, we develop a new compression based generalization error bound and propose a new simple pruning method that is compatible with the generalization error analysis. Our method aims to minimize the information loss induced by compression; in particular, it minimizes the redundancy among nodes instead of merely looking at the amount of information of each individual node. It can be executed by simply observing the covariance matrix in the internal layers and is easy to implement. The proposed method is supported by a comprehensive theoretical analysis. Notably, the approximation error induced by compression is characterized by the notion of the statistical *degrees of freedom* [Mallows, 1973; Caponnetto and de Vito, 2007]. It represents the intrinsic dimensionality of a model and is determined by the eigenvalues of the covariance matrix between each node in each layer. Usually, we observe that the eigenvalue rapidly decreases (Fig. 1a) for several reasons such as explicit regularization (Dropout [Wager *et al.*, 2013], weight decay [Krogh and Hertz, 1992]), and implicit regularization [Hardt *et al.*, 2016; Gunasekar *et al.*, 2018], which means that the amount of important information processed in each layer is not large. In

particular, the rapid decay in eigenvalues leads to a low number of degrees of freedom. Then, we can effectively compress a trained network into a smaller one that has fewer parameters than the original. Behind the theory, there is essentially a connection to the *random feature technique* for kernel methods [Bach, 2017]. Compression error analysis is directly connected to generalization error analysis. The derived bound is actually much tighter than the naive VC-theory bound on the uncompressed network [Bartlett *et al.*, 2019] and even tighter than recent compression-based bounds [Arora *et al.*, 2018]. Further, there is a tradeoff between the bias and the variance, where the bias is induced by the network compression and the variance is induced by the variation in the training data. In addition, we show the superiority of our method and experimentally verify our theory with extensive numerical experiments. Our contributions are summarized as follows:

- We give a theoretical compression bound which is compatible with a practically useful pruning method, and propose a new simple pruning method called *spectral pruning* for compressing deep neural networks.
- We characterize the model compression ability by utilizing the notion of the degrees of freedom, which represents the intrinsic dimensionality of the model. We also give a generalization error bound when a trained network is compressed by our method and show that the bias-variance tradeoff induced by model compression appears. The obtained bound is fairly tight compared with existing compression-based bounds and much tighter than the naive VC-dimension bound.

## 2 Model Compression Problem and its Algorithm

Suppose that the training data  $D_{\text{tr}} = \{(x_i, y_i)\}_{i=1}^n$  are observed, where  $x_i \in \mathbb{R}^{d_x}$  is an input and  $y_i$  is an output that could be a real number ( $y_i \in \mathbb{R}$ ), a binary label ( $y_i \in \{\pm 1\}$ ), and so on. The training data are independently and identically distributed. To train the appropriate relationship between  $x$  and  $y$ , we construct a deep neural network model as

$$f(x) = (W^{(L)}\eta(\cdot) + b^{(L)}) \circ \dots \circ (W^{(1)}x + b^{(1)}),$$

where  $W^{(\ell)} \in \mathbb{R}^{m_{\ell+1} \times m_\ell}$ ,  $b^{(\ell)} \in \mathbb{R}^{m_{\ell+1}}$  ( $\ell = 1, \dots, L$ ), and  $\eta : \mathbb{R} \rightarrow \mathbb{R}$  is an activation function (here, the activation function is applied in an element-wise manner; for a vector  $x \in \mathbb{R}^d$ ,  $\eta(x) = (\eta(x_1), \dots, \eta(x_d))^\top$ ). Here,  $m_\ell$  is the width of the  $\ell$ -th layer such that  $m_{L+1} = 1$  (output) and  $m_1 = d_x$  (input). Let  $\hat{f}$  be a trained network obtained from the training data  $D_{\text{tr}} = \{(x_i, y_i)\}_{i=1}^n$  where its parameters are denoted by  $(\hat{W}^{(\ell)}, \hat{b}^{(\ell)})_{\ell=1}^L$ , i.e.,  $\hat{f}(x) = (\hat{W}^{(L)}\eta(\cdot) + \hat{b}^{(L)}) \circ \dots \circ (\hat{W}^{(1)}x + \hat{b}^{(1)})$ . The input to the  $\ell$ -th layer (after activation) is denoted by  $\phi^{(\ell)}(x) = \eta \circ (\hat{W}^{(\ell-1)}\eta(\cdot) + \hat{b}^{(\ell-1)}) \circ \dots \circ (\hat{W}^{(1)}x + \hat{b}^{(1)})$ . We do not specify how to train the network  $\hat{f}$ , and the following argument can be applied to any learning method such as the empirical risk minimizer, the Bayes estimator, or another estimator. We want to compress the trained network  $\hat{f}$  to another

smaller one  $f^\#$  having widths  $(m_\ell^\#)_{\ell=1}^L$  with keeping the test accuracy as high as possible.

To compress the trained network  $\hat{f}$  to a smaller one  $f^\#$ , we propose a simple strategy called *spectral pruning*. The main idea of the method is to find the *most informative subset* of the nodes. The amount of information of the subset is measured by how well the selected nodes can explain the other nodes in the layer and recover the output to the next layer. For example, if some nodes are heavily correlated with each other, then only one of them will be selected by our method. The information redundancy can be computed by a covariance matrix between nodes and a simple regression problem. We do not need to solve a specific nonlinear optimization problem unlike the methods in [Lebedev and Lempitsky, 2016; Wen *et al.*, 2016; Aghasi *et al.*, 2017].

### 2.1 Algorithm Description

Our method basically simultaneously minimizes the *input information loss* and *output information loss*, which will be defined as follows.

**(i) Input information loss.** First, we explain the input information loss. Denote  $\phi(x) = \phi^{(\ell)}(x)$  for simplicity, and let  $\phi_J(x) = (\phi_j(x))_{j \in J} \in \mathbb{R}^{m_\ell^\#}$  be a subvector of  $\phi(x)$  corresponding to an index set  $J \in [m_\ell]^{m_\ell^\#}$ , where  $[m] := \{1, \dots, m\}$  (here, duplication of the index is allowed). The basic strategy is to solve the following optimization problem so that we can recover  $\phi(x)$  from  $\phi_J(x)$  as accurately as possible:

$$\hat{A}_J := (\hat{A}_J^{(\ell)}) = \underset{A \in \mathbb{R}^{m_\ell \times |J|}}{\text{argmin}} \hat{\mathbb{E}}[\|\phi - A\phi_J\|^2] + \|A\|_\tau^2, \quad (1)$$

where  $\hat{\mathbb{E}}[\cdot]$  is the expectation with respect to the empirical distribution ( $\hat{\mathbb{E}}[f] = \frac{1}{n} \sum_{i=1}^n f(x_i)$ ) and  $\|A\|_\tau^2 = \text{Tr}[AI_\tau A^\top]$  for a regularization parameter  $\tau \in \mathbb{R}_+^{|J|}$  and  $I_\tau := \text{diag}(\tau)$  (how to set the regularization parameter will be given in Theorem 1). The optimal solution  $\hat{A}_J$  can be explicitly expressed by utilizing the (*noncentered*) *covariance matrix* in the  $\ell$ -th layer of the trained network  $\hat{f}$ , which is defined as  $\hat{\Sigma} := \hat{\Sigma}^{(\ell)} = \frac{1}{n} \sum_{i=1}^n \phi(x_i)\phi(x_i)^\top$ , defined on the empirical distribution (here, we omit the layer index  $\ell$  for notational simplicity). Let  $\hat{\Sigma}_{I,I'} \in \mathbb{R}^{K \times H}$  for  $K, H \in \mathbb{N}$  be the submatrix of  $\hat{\Sigma}$  for the index sets  $I \in [m_\ell]^K$  and  $I' \in [m_\ell]^H$  such that  $\hat{\Sigma}_{I,I'} = (\hat{\Sigma}_{i,j})_{i \in I, j \in I'}$ . Let  $F = \{1, \dots, m_\ell\}$  be the full index set. Then, we can easily see that

$$\hat{A}_J = \hat{\Sigma}_{F,J}(\hat{\Sigma}_{J,J} + I_\tau)^{-1}.$$

Hence, the full vector  $\phi(x)$  can be decoded from  $\phi_J(x)$  as  $\phi(x) \approx \hat{A}_J\phi_J(x) = \hat{\Sigma}_{F,J}(\hat{\Sigma}_{J,J} + I_\tau)^{-1}\phi_J(x)$ . To measure the approximation error, we define  $L_\tau^{(A)}(J) = \min_{A \in \mathbb{R}^{m_\ell \times |J|}} \hat{\mathbb{E}}[\|\phi - A\phi_J\|^2] + \|A\|_\tau^2$ . By substituting the explicit formula  $\hat{A}_J$  into the objective, this is reformulated as

$$L_\tau^{(A)}(J) = \text{Tr}[\hat{\Sigma}_{F,F} - \hat{\Sigma}_{F,J}(\hat{\Sigma}_{J,J} + I_\tau)^{-1}\hat{\Sigma}_{J,F}].$$

**(ii) Output information loss.** Next, we explain the output information loss. Suppose that we aim to directly approximate

the outputs  $Z^{(\ell)}\phi$  for a weight matrix  $Z^{(\ell)} \in \mathbb{R}^{m \times m_\ell}$  with an output size  $m \in \mathbb{N}$ . A typical situation is that  $Z^{(\ell)} = \hat{W}^{(\ell)}$  so that we approximate the output  $\hat{W}^{(\ell)}\phi$  (the concrete setting of  $Z^{(\ell)}$  will be specified in Theorem 1). Then, we consider the objective

$$L_\tau^{(B)}(J) := \sum_{j=1}^m \min_{\alpha \in \mathbb{R}^{m_\ell}} \left\{ \widehat{\mathbb{E}}[(Z_{j,:}^{(\ell)}\phi - \alpha^\top \phi_J)^2] + \|\alpha^\top\|_\tau^2 \right\} \\ = \text{Tr}\{Z^{(\ell)}[\widehat{\Sigma}_{F,F} - \widehat{\Sigma}_{F,J}(\widehat{\Sigma}_{J,J} + \mathbf{I}_\tau)^{-1}\widehat{\Sigma}_{J,F}]Z^{(\ell)\top}\},$$

where  $Z_{j,:}^{(\ell)}$  means the  $j$ -th row of the matrix  $Z^{(\ell)}$ . It can be easily checked that the optimal solution  $\hat{\alpha}_J$  of the minimum in the definition of  $L_\tau^{(B)}$  is given as  $\hat{\alpha}_J = \hat{A}_J^\top Z_{j,:}^{(\ell)\top}$  for each  $j = 1, \dots, m$ .

**(iii) Combination of the input and output information losses.** Finally, we combine the input and output information losses and aim to minimize this combination. To do so, we propose to the use of the convex combination of both criteria for a parameter  $0 \leq \theta \leq 1$  and optimize it with respect to  $J$  under a cardinality constraint  $|J| = m_\ell^\#$  for a prespecified width  $m_\ell^\#$  of the compressed network:

$$\min_J L_\tau^{(\theta)}(J) = \theta L_\tau^{(A)}(J) + (1 - \theta)L_\tau^{(B)}(J) \\ \text{s.t. } J \in [m_\ell]^{m_\ell^\#}. \quad (2)$$

We call this method **spectral pruning**. There are the hyperparameter  $\theta$  and regularization parameter  $\tau$ . However, we see that it is robust against the choice of hyperparameter in experiments (Sec. 5). Let  $J_\ell^\#$  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 maximization of a monotonic submodular function [Krause and Golovin, 2014]. That is, we start from  $J = \emptyset$ , sequentially choose an element  $j^* \in [m_\ell]$  that maximally reduces the objective  $L_\tau^{(\theta)}$ , and add this element  $j^*$  to  $J$  ( $J \leftarrow J \cup \{j^*\}$ ) until  $|J| = m_\ell^\#$  is satisfied.

After we chose an index  $J_\ell^\#$  ( $\ell = 2, \dots, L$ ) for each layer, we construct the compressed network  $f^\#$  as  $f^\#(x) = (W^{(\#(L))}\eta(\cdot) + b^{(\#(L))}) \circ \dots \circ (W^{(\#(1))}x + b^{(\#(1))})$ , where  $W^{(\ell)} = W_{J_{\ell+1}, F}^{(\ell)} \hat{A}_{J_\ell^\#}^{(\ell)}$  and  $b^{(\ell)} = b_{J_\ell^\#}^{(\ell)}$ .

An application to a CNN is given in the appendix of the long version of this paper [Suzuki *et al.*, 2020a]. The method can be executed in a layer-wise manner, thus it can be applied to networks with complicated structures such as ResNet.

### 3 Compression accuracy Analysis and Generalization Error Bound

In this section, we give a theoretical guarantee of our method. First, we give the approximation error induced by our pruning procedure in Theorem 1. Next, we evaluate the generalization error of the compressed network in Theorem 2. More specifically, we introduce a quantity called the *degrees of freedom* [Mallows, 1973; Caponnetto and de Vito, 2007;

Suzuki, 2018; Suzuki *et al.*, 2020b] that represents the intrinsic dimensionality of the model and determines the approximation accuracy.

For the theoretical analysis, we define a neural network model with norm constraints on the parameters  $W^{(\ell)}$  and  $b^{(\ell)}$  ( $\ell = 1, \dots, L$ ). Let  $R > 0$  and  $R_b > 0$  be the upper bounds of the parameters, and define the norm-constrained model as

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

where  $W_{j,:}^{(\ell)}$  means the  $j$ -th row of the matrix  $W^{(\ell)}$ ,  $\|\cdot\|$  is the Euclidean norm, and  $\|\cdot\|_\infty$  is the  $\ell_\infty$ -norm<sup>1</sup>. We make the following assumption for the activation function, which is satisfied by ReLU and leaky ReLU [Maas *et al.*, 2013].

**Assumption 1.** We assume that the activation function  $\eta$  satisfies (1) *scale invariance*:  $\eta(ax) = a\eta(x)$  for all  $a > 0$  and  $x \in \mathbb{R}^d$  and (2) *1-Lipschitz continuity*:  $|\eta(x) - \eta(x')| \leq \|x - x'\|$  for all  $x, x' \in \mathbb{R}^d$ , where  $d$  is arbitrary.

#### 3.1 Approximation Error Analysis

Here, we evaluate the approximation error derived by our pruning procedure. Let  $(m_\ell^\#)_{\ell=1}^L$  denote the width of each layer of the compressed network  $f^\#$ . We characterize the approximation error between  $f^\#$  and  $\hat{f}$  on the basis of the degrees of freedom with respect to the *empirical  $L_2$ -norm*  $\|g\|_n^2 := \frac{1}{n} \sum_{i=1}^n \|g(x_i)\|^2$ , which is defined for a vector-valued function  $g$ . Recall that the empirical covariance matrix in the  $\ell$ -th layer is denoted by  $\widehat{\Sigma}^{(\ell)}$ . We define the degrees of freedom as

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

where  $(\hat{\mu}_j^{(\ell)})_{j=1}^{m_\ell}$  are the eigenvalues of  $\widehat{\Sigma}^{(\ell)}$  sorted in decreasing order. Roughly speaking, this quantity quantifies the number of eigenvalues above  $\lambda$ , and thus it is monotonically decreasing w.r.t.  $\lambda$ . The degrees of freedom play an essential role in investigating the predictive accuracy of ridge regression [Mallows, 1973; Caponnetto and de Vito, 2007; Bach, 2017]. To characterize the output information loss, we also define the *output aware degrees of freedom* with respect to a matrix  $Z^{(\ell)}$  as

$$\hat{N}_\ell'(\lambda; Z^{(\ell)}) := \text{Tr}[Z^{(\ell)}\widehat{\Sigma}^{(\ell)}(\widehat{\Sigma}^{(\ell)} + \lambda \mathbf{I})^{-1}Z^{(\ell)\top}].$$

This quantity measures the intrinsic dimensionality of the output from the  $\ell$ -th layer for a weight matrix  $Z^{(\ell)}$ . If the covariance  $\widehat{\Sigma}^{(\ell)}$  and the matrix  $Z^{(\ell)}$  are near low rank,  $\hat{N}_\ell'(\lambda; Z^{(\ell)})$  becomes much smaller than  $\hat{N}_\ell(\lambda)$ . Finally, we define  $N_\ell^\theta(\lambda) := \theta \hat{N}_\ell(\lambda) + (1 - \theta) \hat{N}_\ell'(\lambda; Z^{(\ell)})$ .

To evaluate the approximation error induced by compression, we define  $\lambda_\ell > 0$  as

$$\lambda_\ell = \inf\{\lambda \geq 0 \mid m_\ell^\# \geq 5 \hat{N}_\ell(\lambda) \log(80 \hat{N}_\ell(\lambda))\}. \quad (3)$$

<sup>1</sup>We are implicitly supposing  $R, R_b \simeq 1$  so that  $\|W^{(\ell)}\|_F, \|b^{(\ell)}\| = O(1)$ .

Conversely, we may determine  $m_\ell^\sharp$  from  $\lambda_\ell$  to obtain the theorems we will mention below. Along with the degrees of freedom, we define the *leverage score*  $\tilde{\tau}_j^{(\ell)} \in \mathbb{R}^{m_\ell}$  as  $\tilde{\tau}_j^{(\ell)} := \frac{1}{\hat{N}_\ell(\lambda_\ell)} [\hat{\Sigma}^{(\ell)} (\hat{\Sigma}^{(\ell)} + \lambda_\ell \mathbf{I})^{-1}]_{j,j}$  ( $j \in [m_\ell]$ ). Note that  $\sum_{j=1}^{m_\ell} \tilde{\tau}_j^{(\ell)} = 1$  originates from the definition of the degrees of freedom. The leverage score can be seen as the amount of contribution of node  $j \in [m_\ell]$  to the degrees of freedom. For simplicity, we assume that  $\tilde{\tau}_j^{(\ell)} > 0$  for all  $\ell, j$  (otherwise, we just need to neglect such a node with  $\tilde{\tau}_j^{(\ell)} = 0$ ).

For the approximation error bound, we consider two situations: (i) (Backward procedure) spectral pruning is applied from  $\ell = L$  to  $\ell = 2$  in order, and for pruning the  $\ell$ -th layer, we may utilize the selected index  $J_{\ell+1}^\sharp$  in the  $\ell + 1$ -th layer and (ii) (Simultaneous procedure) spectral pruning is simultaneously applied for all  $\ell = 2, \dots, L$ . We provide a statement for only the backward procedure. The simultaneous procedure also achieves a similar bound with some modifications. For the complete statement of the theorem, see the appendix of the long version [Suzuki *et al.*, 2020a].

As for  $Z^{(\ell)}$  for the output information loss, we set  $Z_{k,:}^{(\ell)} = \sqrt{m_\ell q_{jk}^{(\ell)}} (\max_{j'} \|\hat{W}_{j',:}^{(\ell)}\|)^{-1} \hat{W}_{j_k,:}^{(\ell)}$  ( $k = 1, \dots, m_{\ell+1}^\sharp$ ) where we let  $J_{\ell+1}^\sharp = \{j_1, \dots, j_{m_{\ell+1}^\sharp}\}$ , and  $q_j^{(\ell)} := \frac{(\tilde{\tau}_j^{(\ell+1)})^{-1}}{\sum_{j' \in J_{\ell+1}^\sharp} (\tilde{\tau}_{j'}^{(\ell+1)})^{-1}}$  ( $j \in J_{\ell+1}^\sharp$ ) and  $q_j^{(\ell)} = 0$  (otherwise). Finally, we set the regularization parameter  $\tau$  as  $\tau \leftarrow m_\ell^\sharp \lambda_\ell \tilde{\tau}^{(\ell)}$ .

**Theorem 1** (Compression rate via the degrees of freedom). *If we solve the optimization problem (2) with the additional constraint  $\sum_{j \in J} (\tilde{\tau}_j^{(\ell)})^{-1} \leq \frac{5}{3} m_\ell m_\ell^\sharp$  for the index set  $J$ , then the optimization problem is feasible, and the overall approximation error of  $f^\sharp$  is bounded by*

$$\|\hat{f} - f^\sharp\|_n \leq \sum_{\ell=2}^L \left( \bar{R}^{L-\ell+1} \sqrt{\prod_{\ell'=\ell}^L \zeta_{\ell',\theta}} \right) \sqrt{\lambda_\ell} \quad (4)$$

for  $\bar{R} = \sqrt{\hat{c}} R$ , where  $\hat{c}$  is a universal constant, and  $\zeta_{\ell,\theta} :=$

$$N_\ell^\theta(\lambda_\ell) \left( \theta \frac{\max_{j \in [m_{\ell+1}]} \|\hat{W}_{j,:}^{(\ell)}\|^2}{\|(\hat{W}^{(\ell)})^\top \mathbf{I}_{q^{(\ell)}} \hat{W}^{(\ell)}\|_{\text{op}}} + (1 - \theta) m_\ell \right)^{-1}.$$

The proof is given in the appendix of the long version [Suzuki *et al.*, 2020a]. To prove the theorem, we essentially need to use theories of random features in kernel methods [Bach, 2017; Suzuki, 2018]. The main message from the theorem is that the approximation error induced by compression is directly controlled by the degrees of freedom. Since the degrees of freedom  $\hat{N}_\ell(\lambda)$  are a monotonically decreasing function with respect to  $\lambda$ , they become large as  $\lambda$  decreases to 0. The behavior of the eigenvalues determines how rapidly  $\hat{N}_\ell(\lambda)$  increases as  $\lambda \rightarrow 0$ . We can see that if the eigenvalues  $\hat{\mu}_1^{(\ell)} \geq \hat{\mu}_2^{(\ell)} \geq \dots$  rapidly decrease, then the approximation error  $\lambda_\ell$  can be much smaller for a given model size  $m_\ell^\sharp$ . In other words,  $f^\sharp$  can be much closer to the original network  $\hat{f}$  if there are only a few large eigenvalues.

<sup>2</sup>  $\|\cdot\|_{\text{op}}$  represents the operator norm of a matrix (the largest absolute singular value).

The quantity  $\zeta_{\ell,\theta}$  characterizes how well the approximation error  $\lambda_{\ell'}$  of the lower layers  $\ell' \leq \ell$  propagates to the final output. We can see that a tradeoff between  $\zeta_{\ell,\theta}$  and  $\theta$  appears. By a simple evaluation,  $N_\ell^\theta$  in the numerator of  $\zeta_{\ell,\theta}$  is bounded by  $m_\ell$ ; thus,  $\theta = 1$  gives  $\zeta_{\ell,\theta} \leq 1$ . On the other hand, the

term  $\frac{\max_{j \in [m_{\ell+1}]} \|\hat{W}_{j,:}^{(\ell)}\|^2}{\|(\hat{W}^{(\ell)})^\top \mathbf{I}_{q^{(\ell)}} \hat{W}^{(\ell)}\|_{\text{op}}}$  takes a value between  $m_{\ell+1}$  and 1;

thus,  $\theta = 1$  is not necessarily the best choice to maximize the denominator. From this consideration, we can see that the value of  $\theta$  that best minimizes  $\zeta_{\ell,\theta}$  exists between 0 and 1, which supports our numerical result (Fig. 2b). In any situation, small degrees of freedom give a small  $\zeta_{\ell,\theta}$ , leading to a sharper bound.

### 3.2 Generalization Error Analysis

Here, we derive the generalization error bound of the compressed network with respect to the population risk. We will see that a bias-variance tradeoff induced by network compression appears. As usual, we train a network through the training error  $\hat{\Psi}(f) := \frac{1}{n} \sum_{i=1}^n \psi(y_i, f(x_i))$ , where  $\psi : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  is a loss function. Correspondingly, the expected error is denoted by  $\Psi(f) := \mathbb{E}[\psi(Y, f(X))]$ , where the expectation is taken with respect to  $(X, Y) \sim P$ . Our aim here is to bound the generalization error  $\Psi(f^\sharp)$  of the compressed network. Let the marginal distribution of  $X$  be  $P_X$  and that of  $y$  be  $P_Y$ . First, we assume the Lipschitz continuity for the loss function  $\psi$ .

**Assumption 2.** *The loss function  $\psi$  is  $\rho$ -Lipschitz continuous:  $|\psi(y, f) - \psi(y, f')| \leq \rho |f - f'|$  ( $\forall y \in \text{supp}(P_Y)$ ,  $\forall f, f' \in \mathbb{R}$ ). The support of  $P_X$  is bounded:  $\|x\| \leq D_x$  ( $\forall x \in \text{supp}(P_X)$ ).*

For a technical reason, we assume the following condition for the spectral pruning algorithm.

**Assumption 3.** *We assume that  $0 \leq \theta \leq 1$  is appropriately chosen so that  $\zeta_{\ell,\theta}$  in Theorem 1 satisfies  $\zeta_{\ell,\theta} \leq 1$  almost surely, and spectral pruning is solved under the condition  $\sum_{j \in J} (\tilde{\tau}_j^{(\ell)})^{-1} \leq \frac{5}{3} m_\ell m_\ell^\sharp$  on the index set  $J$ .*

As for the choice of  $\theta$ , this assumption is always satisfied at least by the backward procedure. The condition on the linear constraint on  $J$  is merely to ensure the leverage scores are balanced for the chosen index. Note that the bounds in Theorem 1 can be achieved even with this condition.

If  $L_\infty$ -norm of networks is loosely evaluated, the generalization error bound of deep learning can be unrealistically large because there appears  $L_\infty$ -norm in its evaluation. However, we may consider a truncated estimator  $[\hat{f}(x)] := \max\{-M, \min\{M, \hat{f}(x)\}\}$  for sufficiently large  $0 < M \leq \infty$  to moderate the  $L_\infty$ -norm (if  $M = \infty$ , this does not affect anything). Note that the truncation procedure does not affect the classification error for a classification task. To bound the generalization error, we define  $\delta_1$  and  $\delta_2$  for  $(m_2^\sharp, \dots, m_L^\sharp)$  and  $(\lambda_2, \dots, \lambda_L)$  satisfying relation (3) as <sup>3</sup>

$$\delta_1 = \sum_{\ell=2}^L \left( \bar{R}^{L-\ell+1} \sqrt{\prod_{\ell'=\ell}^L \zeta_{\ell',\theta}} \right) \sqrt{\lambda_\ell},$$

<sup>3</sup>  $\log_+(x) = \max\{1, \log(x)\}$ .

$$\delta_2^2 = \frac{1}{n} \sum_{\ell=1}^L m_\ell^\# m_{\ell+1}^\# \log_+ \left( 1 + \frac{4\hat{G} \max\{\bar{R}, \bar{R}_b\}}{\bar{R}_\infty} \right),$$

where  $\hat{R}_\infty := \min\{\bar{R}^L D_x + \sum_{\ell=1}^L \bar{R}^{L-\ell} \bar{R}_b, M\}$ ,  $\hat{G} := L\bar{R}^{L-1} D_x + \sum_{\ell=1}^L \bar{R}^{L-\ell}$  for  $\bar{R} = \sqrt{\hat{c}}R$  and  $\bar{R}_b = \sqrt{\hat{c}}R_b$  with the constants  $\hat{c}$  introduced in Theorem 1. Let  $R_{n,t} := \frac{1}{n} \left( t + \sum_{\ell=2}^L \log(m_\ell) \right)$  for  $t > 0$ . Then, we obtain the following generalization error bound for the compressed network  $f^\#$ .

**Theorem 2** (Generalization error bound of the compressed network). *Suppose that Assumptions 1, 2, and 3 are satisfied. Then, the spectral pruning method presented in Theorem 1 satisfies the following generalization error bound. There exists a universal constant  $C_1 > 0$  such that for any  $t > 0$ , it holds that*

$$\begin{aligned} \Psi(\|f^\#\|) &\leq \hat{\Psi}(\|\hat{f}\|) + \rho \left\{ \delta_1 + C_1 \hat{R}_\infty (\delta_2 + \delta_2^2 + \sqrt{R_{n,t}}) \right\} \\ &\lesssim \hat{\Psi}(\|\hat{f}\|) + \sum_{\ell=2}^L \sqrt{\lambda_\ell} + \sqrt{\frac{\sum_{\ell=1}^L m_{\ell+1}^\# m_\ell^\#}{n} \log_+(\hat{G})}, \end{aligned}$$

uniformly over all choices of  $\mathbf{m}^\# = (m_1^\#, \dots, m_L^\#)$  with probability  $1 - 2e^{-t}$ .

The proof is given in the appendix of the long version [Suzuki *et al.*, 2020a]. From this theorem, the generalization error of  $f^\#$  is upper-bounded by the training error of the *original* network  $\hat{f}$  (which is usually small) and an additional term. By Theorem 1,  $\delta_1$  represents the approximation error between  $\hat{f}$  and  $f^\#$ ; hence, it can be regarded as a *bias*. The second term  $\delta_2$  is the *variance* term induced by the sample deviation. It is noted that the variance term  $\delta_2$  only depends on the *size of the compressed network rather than the original network size*. On the other hand, a naive application of the theorem implies  $\Psi(\|f\|) - \hat{\Psi}(\|\hat{f}\|) \leq \tilde{O}\left(\sqrt{\frac{1}{n} \sum_{\ell=1}^L m_{\ell+1} m_\ell}\right)$  for the original network  $\hat{f}$ , which coincides with the VC-dimension based bound [Bartlett *et al.*, 2019] but is much larger than  $\delta_2$  when  $m_\ell^\# \ll m_\ell$ . Therefore, the variance is significantly reduced by model compression, resulting in a much improved generalization error. Note that the relation between  $\delta_1$  and  $\delta_2$  is a tradeoff due to the monotonicity of the degrees of freedom. When  $m_\ell^\#$  is large, the bias  $\delta_1$  becomes small owing to the monotonicity of the degrees of freedom, but the variance  $\delta_2(\mathbf{m}^\#)$  will be large. Hence, we need to tune the size  $(m_\ell^\#)_{\ell=1}^L$  to obtain the best generalization error by balancing the bias ( $\delta_1$ ) and variance ( $\delta_2$ ).

The generalization error bound is uniformly valid over the choice of  $\mathbf{m}^\#$  (to ensure this, the term  $R_{n,t}$  appears). Thus,  $\mathbf{m}^\#$  can be arbitrary and chosen in a data-dependent manner. This means that the bound is *a posteriori*, and the best choice of  $\mathbf{m}^\#$  can depend on the trained network.

## 4 Relation to Existing Work

A seminal work [Arora *et al.*, 2018] showed a generalization error bound based on how the network can be compressed. Although the theoretical insights provided by their

Layer	Original	[Arora <i>et al.</i> , 2018]	Spec Prun
1	1,728	1,645	1,013
4	147,456	644,654	84,499
6	589,824	3,457,882	270,216
9	1,179,648	36,920	50,768
12	2,359,296	22,735	4,583
15	2,359,296	26,584	3,886

Table 1: Comparison of the intrinsic dimensionality of our degrees of freedom and existing one. They are computed for a VGG-19 network trained on CIFAR-10.

analysis are quite instructive, the theory does not give a practical compression method. In fact, a random projection is proposed in the analysis, but it is not intended for practical use. The most difference is that their analysis exploits the near low rankness of the weight matrix  $W^{(\ell)}$ , while ours exploits the near low rankness of the covariance matrix  $\hat{\Sigma}^{(\ell)}$ . They are not directly comparable; thus, we numerically compare the intrinsic dimensionality of both with a VGG-19 network trained on CIFAR-10. Table 1 summarizes a comparison of the intrinsic dimensionalities. For our analysis, we used  $\hat{N}_\ell(\lambda_\ell) \hat{N}_{\ell+1}(\lambda_{\ell+1}) k^2$  for the intrinsic dimensionality of the  $\ell$ -th layer, where  $k$  is the kernel size<sup>4</sup>. This is the number of parameters in the  $\ell$ -th layer for the width  $m_\ell^\# \simeq \hat{N}_\ell(\lambda_\ell)$  where  $\lambda_\ell$  was set as  $\lambda_\ell = 10^{-3} \times \text{Tr}[\hat{\Sigma}^{(\ell)}]$ , which is sufficiently small. We can see that the quantity based on our degrees of freedom give significantly small values in almost all layers.

The PAC-Bayes bound [Dziugaite and Roy, 2017; Zhou *et al.*, 2019] is also a promising approach for obtaining the non-vacuous generalization error bound of a compressed network. However, these studies “assume” the existence of effective compression methods and do not provide any specific algorithm. [Suzuki, 2018; Suzuki *et al.*, 2020b] also pointed out the importance of the degrees of freedom for analyzing the generalization error of deep learning but did not give a practical algorithm.

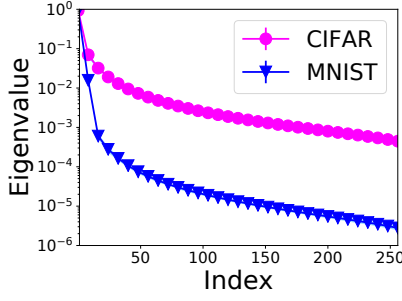
## 5 Numerical Experiments

In this section, we conduct numerical experiments to show the validity of our theory and the effectiveness of the proposed method.

### 5.1 Eigenvalue Distribution and Compression Ability

We show how the rate of decrease in the eigenvalues affects the compression accuracy to justify our theoretical analysis. We constructed a network (namely, NN3) consisting of three hidden fully connected layers with widths (300, 1000, 300) following the settings in [Aghasi *et al.*, 2017] and trained it with 60,000 images in MNIST and 50,000 images in CIFAR-10. Figure 1a shows the magnitudes of the eigenvalues of the 3rd hidden layers of the networks trained for each dataset

<sup>4</sup>We omitted quantities related to the depth  $L$  and log term, but the intrinsic dimensionality of [Arora *et al.*, 2018] also omits these factors.



(a) Eigenvalue distributions in each layer for MNIST and CIFAR-10.

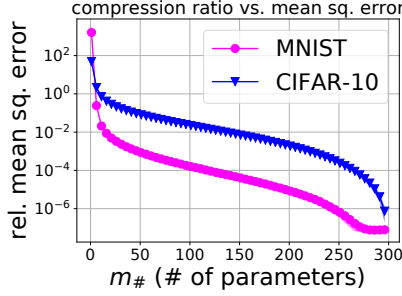

 (b) Approximation error with its s.d. versus the width  $m^\#$ 

Figure 1: Eigenvalue distribution and compression ability of a fully connected network in MNIST and CIFAR-10.

(plotted on a semilog scale). The eigenvalues are sorted in decreasing order, and they are normalized by division by the maximum eigenvalue. We see that eigenvalues for MNIST decrease much more rapidly than those for CIFAR-10. This indicates that MNIST is “easier” than CIFAR-10 because the degrees of freedom (an intrinsic dimensionality) of the network trained on MNIST are relatively smaller than those trained on CIFAR-10. Figure 1b presents the (relative) compression error  $\|\hat{f} - f^\#\|_n / \|\hat{f}\|_n$  versus the width  $m_3^\#$  of the compressed network where we compressed only the 3rd layer and  $\lambda_3$  was fixed to a constant  $10^{-6} \times \text{Tr}[\hat{\Sigma}_{(\ell)}]$  and  $\theta = 0.5$ . It shows a rapid decrease in the compression error for MNIST than CIFAR-10 (about 100 times smaller). This is because MNIST has faster eigenvalue decay than CIFAR-10.

Figure 2a shows the relation between the test classification accuracy and  $\lambda_\ell$ . It is plotted for a VGG-13 network trained on CIFAR-10. We chose the width  $m_\ell^\#$  that gave the best accuracy for each  $\lambda_\ell$  under the constraint of the compression rate (relative number of parameters). We see that as the compression rate increases, the best  $\lambda_\ell$  goes down. Our theorem tells that  $\lambda_\ell$  is related to the compression error through (3), that is, as the width goes up,  $\lambda_\ell$  must go down. This experiment supports the theoretical evaluation. Figure 2b shows the relation between the test classification accuracy and the hyperparameter  $\theta$ . We can see that the best accuracy is achieved around  $\theta = 0.3$  for all compression rates, which indicates the superiority of the “combination” of input- and output-

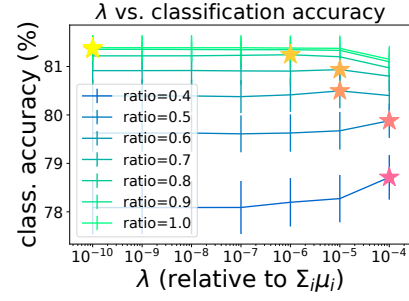
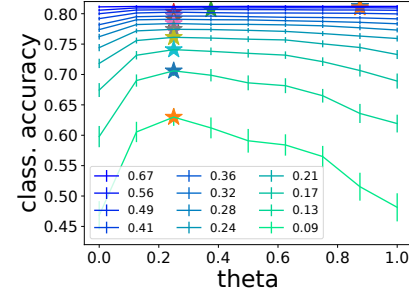

 (a) Accuracy versus  $\lambda_\ell$  in CIFAR-10 for each compression rate.

 (b) Accuracy versus  $\theta$  in CIFAR-10 for each compression rate.

 Figure 2: Relation between accuracy and the hyper parameters  $\lambda_\ell$  and  $\theta$ . The best  $\lambda$  and  $\theta$  are indicated by the star symbol.

information loss and supports our theoretical bound. For low compression rate, the choice of  $\lambda_\ell$  and  $\theta$  does not affect the result so much, which indicates the robustness of the hyper-parameter choice.

## 5.2 Compression on ImageNet Dataset

We applied our method to the ImageNet (ILSVRC2012) dataset [Deng *et al.*, 2009]. We compared our method using the ResNet-50 network [He *et al.*, 2016] (experiments for VGG-16 network [Simonyan and Zisserman, 2014] are also shown in the long version [Suzuki *et al.*, 2020a]). Our method was compared with the following pruning methods: ThiNet [Luo *et al.*, 2017], NISP [Yu *et al.*, 2018], and sparse regularization [He *et al.*, 2017] (which we call Sparse-reg). As the initial ResNet network, we used two types of networks: ResNet-50-1 and ResNet-50-2. For training ResNet-50-1, we followed the experimental settings in [Luo *et al.*, 2017] and [Yu *et al.*, 2018]. During training, images were resized as in [Luo *et al.*, 2017]. to  $256 \times 256$ ; then, a  $224 \times 224$  random crop was fed into the network. In the inference stage, we center-cropped the resized images to  $224 \times 224$ . For training ResNet-50-2, we followed the same settings as in [He *et al.*, 2017]. In particular, images were resized such that the shorter side was 256, and a center crop of  $224 \times 224$  pixels was used for testing. The augmentation for fine tuning was a  $224 \times 224$  random crop and its mirror.

We compared ThiNet and NISP for ResNet-50-1 (we call our model for this situation “Spec-ResA”) and Sparse-reg

Model	Top-1	Top-5	# Param.	FLOPs
ResNet-50-1	72.89%	91.07%	25.56M	7.75G
ThiNet-70	72.04 %	90.67%	16.94M	4.88G
ThiNet-50	71.01 %	90.02%	12.38M	3.41G
NISP-50-A	72.68%	—	18.63M	5.63G
NISP-50-B	71.99%	—	14.57M	4.32G
<b>Spec-ResA</b>	<b>72.99%</b>	<b>91.56%</b>	12.38M	3.45G
ResNet-50-2	75.21%	92.21%	25.56M	7.75G
Sparse-reg wo/ ft	—	84.2%	19.78M	5.25G
Sparse-reg w/ ft	—	90.8%	19.78M	5.25G
<b>Spec-ResB</b> wo/ ft	66.12%	<b>86.67%</b>	20.69M	5.25G
<b>Spec-ResB</b> w/ ft	74.04%	<b>91.77%</b>	20.69M	5.25G

Table 2: Performance comparison of our method and existing ones for ResNet-50 on ImageNet. “ft” indicates fine tuning after compression.

for ResNet-50-2 (we call our model for this situation “Spec-ResB”) for fair comparison. The size of compressed network  $f^\#$  was determined to be as close to the compared network as possible (except, for ResNet-50-2, we did not adopt the “channel sampler” proposed by [He *et al.*, 2017] in the first layer of the residual block; hence, our model became slightly larger). The accuracies are borrowed from the scores presented in each paper, and thus we used different models because the original papers of each model reported for each different model. We employed the simultaneous procedure for compression. After pruning, we carried out fine tuning over 10 epochs, where the learning rate was  $10^{-3}$  for the first four epochs,  $10^{-4}$  for the next four epochs, and  $10^{-5}$  for the last two epochs. We employed  $\lambda_\ell = 10^{-6} \times \text{Tr}[\hat{\Sigma}_{(\ell)}]$  and  $\theta = 0.5$ .

Table 2 summarizes the performance comparison for ResNet-50. We can see that for both settings, our method outperforms the others for about 1% accuracy. This is an interesting result because ResNet-50 is already compact [Luo *et al.*, 2017] and thus there is less room to produce better performance. Moreover, we remark that all layers were simultaneously trained in our method, while other methods were trained one layer after another. Since our method did not adopt the channel sampler proposed by [He *et al.*, 2017], our model was a bit larger. However, we could obtain better performance by combining it with our method.

## 6 Conclusion

In this paper, we proposed a simple pruning algorithm for compressing a network and gave its approximation and generalization error bounds using the *degrees of freedom*. Unlike the existing compression based generalization error analysis, our analysis is compatible with a practically useful method and further gives a tighter intrinsic dimensionality bound. The proposed algorithm is easily implemented and only requires linear algebraic operations. The numerical experiments showed that the compression ability is related to the eigenvalue distribution, and our algorithm has favorable performance compared to existing methods.

## Acknowledgements

TS was partially supported by MEXT Kakenhi (18K19793, 18H03201 and 20H00576) and JST-CREST, Japan.

## References

- [Aghasi *et al.*, 2017] Alireza Aghasi, Afshin Abdi, Nam Nguyen, and Justin Romberg. Net-trim: Convex pruning of deep neural networks with performance guarantee. In *Advances in Neural Information Processing Systems 30*, pages 3180–3189. 2017.
- [Arora *et al.*, 2018] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 254–263. PMLR, 2018.
- [Bach, 2017] Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18(21):1–38, 2017.
- [Bartlett *et al.*, 2019] Peter Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.
- [Caponnetto and de Vito, 2007] Andrea Caponnetto and Ernesto de Vito. Optimal rates for regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.
- [Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009*, pages 248–255, 2009.
- [Denil *et al.*, 2013] Misha Denil, Babak Shakibi, Laurent Dinh, and Nando de Freitas. Predicting parameters in deep learning. In *Advances in neural information processing systems 26*, pages 2148–2156, 2013.
- [Denton *et al.*, 2014] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in Neural Information Processing Systems 27*, pages 1269–1277. 2014.
- [Dziugaite and Roy, 2017] Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, 2017.
- [Gunasekar *et al.*, 2018] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems 31*, pages 9482–9491, 2018.
- [Han *et al.*, 2015] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

- [Hardt *et al.*, 2016] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, pages 1225–1234. PMLR, 2016.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [He *et al.*, 2017] Yihui He, Xiangyu Zhang, and Jian 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.
- [Krause and Golovin, 2014] Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability*, 2014.
- [Krogh and Hertz, 1992] Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems 5*, pages 950–957, 1992.
- [Lebedev and Lempitsky, 2016] Vadim Lebedev and Victor Lempitsky. Fast convnets using group-wise brain damage. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2554–2564, 2016.
- [Luo *et al.*, 2017] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. ThiNet: a filter level pruning method for deep neural network compression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5068–5076, 2017.
- [Maas *et al.*, 2013] Andrew Maas, Awni Hannun, and Andrew Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.
- [Mallows, 1973] Colin L Mallows. Some comments on Cp. *Technometrics*, 15(4):661–675, 1973.
- [Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [Suzuki *et al.*, 2020a] Taiji Suzuki, Hiroshi Abe, Tomoya Murata, Shingo Horiuchi, Kotaro Ito, Tokuma Wachi, So Hirai, Masatoshi Yukishima, and Tomoaki Nishimura. Spectral pruning: Compressing deep neural networks via spectral analysis and its generalization error. *arXiv preprint arXiv:1808.08558v2*, 2020.
- [Suzuki *et al.*, 2020b] Taiji Suzuki, Hiroshi Abe, and Tomoaki Nishimura. Compression based bound for non-compressed network: Unified generalization error analysis of large compressible deep neural network. In *International Conference on Learning Representations*, 2020.
- [Suzuki, 2018] Taiji Suzuki. Fast generalization error bound of deep learning from a kernel perspective. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, volume 84, pages 1397–1406. PMLR, 2018.
- [Wager *et al.*, 2013] Stefan Wager, Sida Wang, and Percy S Liang. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems 26*, pages 351–359. 2013.
- [Wan *et al.*, 2013] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 1058–1066. PMLR, 2013.
- [Wen *et al.*, 2016] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems 29*, pages 2074–2082. 2016.
- [Yu *et al.*, 2018] Ruichi Yu, Ang Li, Chun-Fu Chen, Jui-Hsin Lai, Vlad I Morariu, Xintong Han, Mingfei Gao, Ching-Yung Lin, and Larry S Davis. NISP: Pruning networks using neuron importance score propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9194–9203, 2018.
- [Zhou *et al.*, 2019] Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P. Adams, and Peter Orbanz. Non-vacuous generalization bounds at the imagenet scale: a PAC-bayesian compression approach. In *International Conference on Learning Representations*, 2019.