A Probabilistic Approach to Neural Network Pruning

Xin Qian [1] **Diego Klabjan** [1]

Abstract

Neural network pruning techniques reduce the number of parameters without compromising predicting ability of a network. Many algorithms have been developed for pruning both over-parameterized fully-connected networks (FCNs) and convolutional neural networks (CNNs), but analytical studies of capabilities and compression ratios of such pruned sub-networks are lacking. We theoretically study the performance of two pruning techniques (random and magnitude-based) on FCNs and CNNs. Given a target network whose weights are independently sampled from appropriate distributions, we provide a universal approach to bound the gap between a pruned and the target network in a probabilistic sense. The results establish that there exist pruned networks with expressive power within any specified bound from the target network.

1. Introduction

The common neural network architectures that achieve the state-of-the-art results usually have tens of billions of trainable parameters (Goodfellow et al., 2016; Radford et al., 2019; Brown et al., 2020), leading to a problem that training and inference of these models are computationally expensive and memory intensive. To address this problem, researchers have developed many practical algorithms to compress the network structure while keeping the original network's expressive power (Li et al., 2016; Han et al., 2015a;b; Cheng et al., 2017).

Recently, Frankle & Carbin (2018) conjecture that, every successfully trained neural network contains much smaller subnetworks (winning tickets) that—when trained in isolation from the original initialization—reach test accuracy comparable to the original network. This conjecture is called the Lottery Ticket Hypothesis (LTH). Ramanujan et al. (2020) further conjecture that, a sufficiently over-parameterized neural network with random initialization contains subnetworks that can achieve competitive accuracy without any training, when comparing to a large trained network. This conjecture can be viewed as a stronger version of the LTH in the sense that we do not need to train this over-parameterized random network. However, to determine the lottery tickets from this over-parameterized network is NP-Hard in the worst case (Malach et al., 2020; Pensia et al., 2020). In addition, since over-parameterization is compared to a trained neural network, which is usually already over-parameterized, the random initialized network is over-over-parameterized and thus too large to consider.

Although the development of such network pruning algorithms dates back to late 80s, there have been only limited studies of the theoretical guarantees of network pruning. The existence and the representation power of good subnetworks are lacking (see the related sections of the survey papers (Sun, 2019; Fan et al., 2019)). Recently, Malach et al. (2020) prove the strong LTH for fully-connected networks with ReLU activations. They show that, given a target FCN of depth l and width d, any random initialized network with depth $2l$ and width $O\left(d^5 l^2/\epsilon^2\right)$ contains subnetworks that can approximate the target network with ϵ error with high probability. In the following works, Pensia et al. (2020) and Orseau et al. (2020) concurrently and independently prove that the width of the random initialized network can be reduced to $O\left(d\log(dl/\epsilon)\right)$. Pensia et al. (2020) further show that this logarithmic over-parameterization is essentially optimal for networks with constant depth. Note that a random initialized network is introduced and pruning is applied on this new network, instead of the target network. Thus, these results cannot provide much insights for developing model pruning algorithms that are applied on the target network directly. Besides, the proof ideas heavily rely on the fact the the random initialized network is well over-parameterized so that a subnetwork with a specific structure that can replicate a single neuron of the target network exists. Although the researchers have improved the polynomial dependency of width to logarithm, the size of the random initialized network is still very large.

In this work, we focus on the theoretical results of pruning an over-parameterized target network directly. There are two

[1]Department of Industrial Engineering and Management Science, Northwestern University. Correspondence to: Xin Qian <xinqian2017@u.northwestern.edu>, Diego Klabjan <d-klabjan@northwestern.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

types of subnetworks: subnetworks where specific weights are pruned (weight-subnetwork) and subnetworks where entire neurons are pruned (neuron-subnetworks). We focus on weight-subnetworks and show that, for both magnitude-based pruning (prune the smallest entries in the weight matrices based on magnitude) and random pruning (randomly select some entries in the weight matrices to prune), we can prune some weights of the target network while maintaining comparable expressive power with positive probability. We show the random pruning and magnitude-based pruning results for FCNs and the random pruning result for CNNs, where the latter one requires a sophisticated formulation to translate the convolutional layers into fully-connected layers with a specific ciuculant structure. The proof framework, which bounds the gap between the output of the pruned and trained networks layer by layer, are universal for both FCNs and CNNs. We rely on the results from probability theory and theoretical computer sciences to give precise bounds of the norms of weight matrices and other random variables.

Our results, as one of the rare studies about the existence of good subnetworks, provide relationships between the width of the target network, the number of pruned weights in each layer, the universal expressive error over a closed region, and the probability that such good subnetworks exist. These results also give guidance for practical researchers by providing the probability that a good subnetwork exists and an estimation of how many entries can be pruned at one time by magnitude-based pruning and random pruning.

The rest of the manuscript is structured as follows. In Section 2 we review the literature while in Section 3 we show the preliminaries and notations that are used throughout the paper. Sections 4 and 5 discuss the theoretical results of pruning of FCNs and CNNs, respectively. We conclude the results and discuss the potential future works in Section 6. We present technical lemmas in Appendix A and the complete proofs of the theorems in Appendix B. In Appendix C we discuss how to extend the theorems to more general settings and in Appendix D we show some numerical results that support our theorems and assumptions.

2. Literature Review

Empirical Neural Network Pruning There has been a long history of neural network pruning. Early studies of pruning reduce the number of connections based on the information of second-order derivatives of the loss function (LeCun et al., 1989). Following works focus on magnitude-based pruning. Han et al. (2015a) propose to reduce the total number of parameters and operations in the entire network. Other works explore pruning neurons and design various methods to determine the redundancy of neurons (Hu et al., 2016; Srinivas & Babu, 2015). Similar approaches are also applied to CNNs to prune filters (Luo et al., 2017) or entire convolutional channels (Li et al., 2016). Recently, Frankle & Carbin (2018) conjecture the lottery ticket hypothesis that, a trained network contains a subnetwork that—when trained in isolation from the original initialization—can match the performance of the original network. Zhou et al. (2019a) claim that the good subnetworks in the LTH have better-than-random performance without any training. Based on the above two works, Ramanujan et al. (2020) conjecture the so-called strong LTH that, within a sufficiently over-parameterized neural network (comparing to the target network) with random weights at initialization, there exists a subnetwork that achieves competitive accuracy with the target network.

Theoretical Study of Neural Network Pruning The study of the theoretical properties of neural network pruning only started recently. Malach et al. (2020) prove the strong LTH for FCNs with ReLU activations. In particular, they show that one can approximate any target FCN of width d and depth l by pruning a sufficiently over-parameterized network of width $O(d^5l^2/\epsilon^2)$ and depth $2l$ such that the gap between the pruned and target networks is bounded by ϵ. Pensia et al. (2020) and Orseau et al. (2020) concurrently and independently improved the width of the random network to $O(\text{poly}(d)\log(dl/\epsilon))$. These results are based on the idea that, for a single-neuron ReLU connection, we can use a two-hidden-layer neural network with constant width to approximate it. In comparison, our results study pruning of the target FCNs and CNNs directly. Another line of research by Ye et al. (2020a;b) propose a greedy optimization based neural network pruning method. They also provide theoretical guarantees of the decreasing discrepancy between the pruned and target networks. Elesedy et al. (2020) stick with the iterative magnitude-based pruning procedure described in Frankle & Carbin (2018) and prove the LTH for linear models trained by gradient flow methods. Arora et al. (2018) and Zhou et al. (2019b) theoretically study a close connection between compressibility and generalization of neural networks. Another line of work (Baykal et al., 2019a; Liebenwein et al., 2020; Baykal et al., 2019b) propose sampling-based neural network pruning algorithms according to certain sensitivity scores and provide theoretical guarantees for both FCNs and CNNs.

Theoretical Study of CNNs Although CNNs are successful in many computer vision tasks (Goodfellow et al., 2016), there is less work discussing theoretical properties of CNNs. Jain (1989) shows that a linear transformation of a 2D convolutional filter can be represented by a doubly block circulant matrix. The circulant structure provides an efficient way to calculate the singular values of the linear transformation corresponding to a convoultional layer (Sedghi et al., 2018).

3. Preliminaries and Notations

We introduce some notations that are used in the sequel. For vector v, we use $\|v\|_0$ and $\|v\|_2$ to denote the L_0 and L_2 norm of v, respectively. For matrix $M \in \mathbb{R}^{m\times n}$, we use $M_{i,j}$ or $(M)_{i,j}$ to denote the element in the i-th row and j-th column of M; we use $M_{i,:}$ and $M_{:,j}$ to denote the i-th row and j-th column of M, respectively; the vectorization of M is defined as $\text{vec}(M) := [M_{1,1}, \ldots, M_{m,1}, \ldots, M_{1,n}, \ldots, M_{m,n}]^T$. We also use analogous notations for higher-order tensors. The operator norm and element-wise maximum norm of M is denoted by $\|M\|_2$ and $\|M\|_{\max} := \max_{i\in[m],j\in[n]} |A_{i,j}|$, respectively. The Hadamard (element-wise) product of two matrices $A, B \in \mathbb{R}^{m\times n}$ is denoted by $M := A \circ B$, where $M_{i,j} = A_{i,j}B_{i,j}$. We denote $\mathbf{0}_{m\times n}$ and $\mathbf{1}_{m\times n}$ as the zero matrix and all 1 matrix of dimension $m \times n$.

For $n \in \mathbb{N}^+$, we define $[n] := \{1, \ldots, n\}$. Given a sequence of real numbers a_n, we denote $\prod_{k=i}^{j} a_k = a_i \times \cdots \times a_j$ if $j \geqslant i$ and $\prod_{k=i}^{j} a_k = 1$ otherwise. For integers $n > 0$ and k, we use $k\%n = k \bmod n$ if $n \nmid k$ and $k\%n = n$ otherwise[1]. We use log to denote the natural logarithm and set $\log^{(2)}(x) := \log(\log(x))$.

We use $\mathcal{U}[a, b]$ to denote the uniform distribution on interval $[a, b]$, and $\mathcal{N}(\mu, \Sigma)$ to denote the multivariate normal distribution with mean $\mu \in \mathbb{R}^p$ and covariance matrix $\Sigma \in \mathbb{R}^{p\times p}$.

We are given a target neural network F of depth $l \geqslant 3$ of the form

$$F(x) = W_l^* \sigma_l \left(W_{l-1}^* \sigma_{l-1} \left(\cdots W_2^* \sigma_1 \left(W_1^* x \right) \right) \right) \quad (1)$$

where σ_k is the activation function and weight matrix $W_k^* \in \mathbb{R}^{d_k \times d_{k-1}}, k \in [l]$[2]. Typically, there are two types of subnetworks, namely weight-subnetworks and neuron-subnetworks, depending on whether we remove (or set to zero) the entire neuron or just the entries of a weight matrix. In this paper, we focus the theoretical results on weight-subnetworks. Mathematically, a pruned weight-subnetwork f of F is a network of the same architecture as F such that the weight matrix in the k-th layer of f is represented by $W_k = M_k \circ W_k^*$ for some mask $M_k \in \{0, 1\}^{d_k \times d_{k-1}}$. Throughout the paper, we fix M_1 and M_l as the all 1 matrix (i.e. we do not prune any weight on the first and last weight matrix of the target network). We aim at reducing the number of active weights while keeping the expressive power of the original network F.

The compression ratio of the k-th layer is defined as $\gamma_k := \|\text{vec}(W_k)\|_0 / D_k$, where $D_k := d_k d_{k-1}$ is the number of weights in the k-th layer. Obviously, we aim at reducing the compression ratios while keeping the expressive power of the original network F.

The error metric used throughout the paper is the universal approximation over the unit ball $\mathcal{B}_{d_0} := \left\{x \in \mathbb{R}^{d_0} : \|x\|_2 \leqslant 1\right\}$, or in the CNN results we use the unit cube $\mathcal{C}_{d_0} := \left\{x \in \mathbb{R}^{d_0} : x_i \in [0, 1], i \in [d_0]\right\}$ instead; i.e. f is ϵ-close to F if

$$\sup_{x\in\mathcal{B}_{d_0}} \|f(x) - F(x)\|_2 \leqslant \epsilon.$$

This definition of discrepancy is common in the theoretical model pruning literature (Malach et al., 2020; Pensia et al., 2020; Orseau et al., 2020). Note that the results of this paper can be easily generalized from the unit ball to any ball with radius r in $\mathbb{R}^{d_0}$. We use the unit ball (or unit cube) only for ease of notation. The discrepancy between the losses of the pruned and target network on a given set of samples can be derived similarly.

4. Pruning Fully-connected Neural Networks

In this section, we show that a fully-connected neural network can be approximated by pruning some of its entries while keeping comparable expressive power under mild assumptions.

We start with two different pruning approaches – random pruning and magnitude-based pruning. Given a target network F as defined in (1) and compression ratios $\gamma_k, k \in [l]$, random pruning refers to applying a set of masks $\{M_1, \ldots, M_l\}$ on F such that M_k is constructed by starting with $M_k = \mathbf{1}_{d_k d_{k-1}}$ and repeating $\lfloor \gamma_k D_k \rfloor$ times the following steps: (1) select $i \in [d_k]$ uniformly at random; (2) select $j \in [d_{k-1}]$ uniformly at random; (3) set $(M_k)_{i,j} = 0$[3]. The magnitude-based pruning refers to applying a set of masks $\{M_1, \ldots, M_l\}$ on F such that $(M_k)_{i,j} = 0$ if $(i, j) \in \mathcal{I}_k$ and $(M_k)_{i,j} = 1$ otherwise, where we order the entries of W_k^* such that $|W_k^*|_{i_1,j_1} \leqslant \cdots \leqslant |W_k^*|_{i_{D_k},j_{D_k}}$ and set $\mathcal{I}_k := \{(i_u, j_u) : 1 \leqslant u \leqslant \lfloor \gamma_k D_k \rfloor\}$. Recall that we assume $\gamma_1 = \gamma_l = 1$ and thus M_1 and M_l are all 1 matrices[4].

Our main theorems in this section show that, for both pruning approaches and under mild conditions, the target network F contains a weight-subnetwork that is ϵ-close to F with high probability. We present the results for magnitude-based pruning and random pruning in Sections 4.1 and 4.2, respectively. We outline the proof in Section 4.3 and defer

[1] Note that this definition is slightly different from the common definition of modulo.

[2] Throughout the paper, we skip the bias terms in the expression of the neural network.

[3] Note that this scheme corresponds to "with-replacement" sampling, i.e., an index pair (i, j) might be selected twice. There is another "without-replacement" strategy. For more details regarding these two strategies, please refer to Appendix E.2.

[4] There is another global version of magnitude-based pruning where the weights of the entire network are sorted and the weights with the smallest magnitudes are pruned. For comparison between these two approaches, please refer to Appendix E.3.

the complete proof to Appendix B.

4.1. Magnitude-based Pruning of FCN

We first present the result for magnitude-based pruning.

Theorem 1. *We are given a target network F as defined in (1). Let us assume that*

(i) σ_k *is* L_k*-Lipschitz and satisfies* $\sigma_k(0) = 0, k \in [l]$*;*
(ii) $d := \min\{d_1, \ldots, d_{l-1}\} \geqslant \max\{d_0, d_l\}$*;*
(iii) entries in W_k^* *are independent and identically distributed following* $\mathcal{U}\left[-\frac{K}{\sqrt{\max\{d_k, d_{k-1}\}}}, \frac{K}{\sqrt{\max\{d_k, d_{k-1}\}}}\right]$ *for a fixed positive constant* K*.*

Let $\epsilon > 0, \delta > 0$*, and* $\alpha \in (0, 1)$ *be such that*

$$d \geqslant \max\left\{C_1^{\frac{1}{\alpha}}, \left(\frac{C_2}{\epsilon}\right)^{\frac{1}{\alpha}}, \left(\frac{C_3}{\delta}\right)^{\frac{1}{\alpha}}, C_4 + C_5 \log\left(\frac{1}{\delta}\right)\right\}$$

for some positive constants C_1, C_2, C_3, C_4 *and* C_5 *(depending on* l *and* L_k*'s) as specified in the proof. Then with probability at least* $1 - \delta$*, the subnetwork* f *of* F *with mask* $M = \left\{M_1, \ldots, M_l, M_k \in \{0,1\}^{d_k \times d_{k-1}}\right\}$ *that prunes the smallest* $\lfloor D_k^{1-\alpha} \rfloor$ *entries of* $W_k^*, 1 < k < l$ *based on magnitude is* ϵ*-close to* F*, i.e.*

$$\sup_{x \in \mathcal{B}_{d_0}} \|f(x) - F(x)\|_2 \leqslant \epsilon. \tag{2}$$

Note that many activation functions, like ReLU and tanh, hold for assumption (i) with $L_k = 1$. In assumption (ii), we assume that the width of the target neural network is larger than the input and output dimensions. This is common in most of the theoretical and practical deep learning results. For assumption (iii), we take the upper/lower bound of the uniform distribution to be $\pm\frac{K}{\sqrt{\max\{d_k, d_{k-1}\}}}$ for a fixed positive constant K so that the variance of this distribution is of the same order as in the Xavier initialization (Glorot & Bengio, 2010). We are aware of the fact that for many trained networks, the weights in each layer do not fit a uniform distribution well. We use the uniform distribution since the closed-form of the order statistics is only available for this distribution. We utilize these closed-form results to give a precise relationship between the width d, error ϵ, probability $1 - \delta$, and the compression ratio that depends on α. Asymptotic results exist for order statistics of general distributions and can be used to estimate such relationships. We discuss more details on how to apply the results of intermediate order statistics to generalize Theorem 1 to other distributions in Appendix C.1. The weights are assumed to be independent for simplicity. For the near-independent and non-independent settings, please refer to Appendix E.1. Same discussions about independency apply for Theorems 2 and 3.

4.2. Random Pruning of FCN

In this section, we present the result for random pruning of FCNs. The key difference between random pruning and magnitude-based pruning is that, given the target network F, the mask corresponding to magnitude-based pruning is fixed while the mask of random pruning is random.

Given the compression ratio γ_k (or the number of weights to prune) in the k-th layer, a random pruning mask M_k can be viewed as random selecting $\lfloor \gamma_k D_k \rfloor$ entries of $\{0,1\}^{d_k \times d_{k-1}}$ with replacement and setting them to zero. These random selected masks are combined to form the mask $\{M_1, \ldots, M_l\}$ that represents a random pruned weight-subnetwork of F. This random property further complicates the proof, as we need to consider the randomness from the entries of the target network and the randomness from the mask at the same time.

Besides the difference of the two pruning approaches, we only assume that each entry of the weight matrix independently follows a distribution with bounded second-order and fourth-order moments, while in Theorem 1 we assume that all the entries in the weight matrix are independently and identically following a specific distribution.

Theorem 2. *We are given a target network F as defined in (1). Let us assume that*

(i) σ_k *is* L_k*-Lipschitz and satisfies* $\sigma_k(0) = 0, k \in [l]$*;*
(ii) $d := \min\{d_1, \ldots, d_{l-1}\} \geqslant \max\{d_0, d_l\}$*;*
(iii) $(W_k^*)_{i,j}$ *independently follows a distribution* $\mathcal{X}_{i,j}^k$*; further, there exist two positive constants* K_1 *and* K_2 *such that* $\mathbb{E}\mathcal{X}_{i,j}^k = 0$*,* $\mathbb{E}\left|\mathcal{X}_{i,j}^k\right|^2 \leqslant \frac{K_1}{\max\{d_k, d_{k-1}\}}$ *and* $\mathbb{E}\left|\mathcal{X}_{i,j}^k\right|^4 \leqslant \frac{K_2}{\max\{d_k, d_{k-1}\}^2}$*;*
(iv) for all $k \in [l]$*, there exists a positive constant* N_k *such that* $\|W_k^*\|_2 \leqslant N_k$ *with probability at least* $1 - \delta_k$*.*

Let $\epsilon > 0, \delta > 0$*, and* $\alpha \in (0, 1)$ *be such that*

$$\alpha \leqslant 1 - \frac{\log(d_{k+1} + 1) - \log^{(2)}(d_{k+1})}{\log(d_{k+1}) + \log(d_k)}, 1 < k < l, \tag{3}$$

$$\alpha \leqslant 1 - \frac{\log(d_k + 1) - \log^{(2)}(d_k)}{\log(d_{k+1}) + \log(d_k)}, 1 < k < l, \tag{4}$$

$$\delta_0 := \delta - \left[\delta_l + \sum_{i=1}^{l-1}(l-i)\delta_i\right] \geqslant 0, \tag{5}$$

and

$$d \geqslant \max\left\{C_1^{\frac{4}{\alpha}}, \left(\frac{C_2}{\epsilon}\right)^{\frac{4}{\alpha}}, \left(\frac{C_3}{\delta_0}\right)^3, \left(\frac{C_4}{\delta_0}\right)^{\frac{4}{\alpha}}\right\},$$

for some positive constants C_1, C_2, C_3 *and* C_4 *(depending on* l*,* L_k*'s, and* N_k*'s) specified in the proof. Then with probability at least* $1 - \delta \geqslant \left(1 - d^{-\frac{1}{3}}\right)^{2(l-2)}(1 - \delta_l)\left[1 -\right.$

$(l-2)c_2 d^{-\frac{\alpha}{4}} - \sum_{i=1}^{l-1}(l-i)\delta_i\Big]$ *over the randomness of masks and weights for some positive constant* c_2 *defined in the proof, the subnetwork* f *of* F *with mask* $M = \left\{M_1, \ldots, M_l, M_k \in \{0,1\}^{d_k \times d_{k-1}}\right\}$ *that randomly prunes* $\lfloor D_k^{1-\alpha} \rfloor$ *entries of* $W_k^*, 1 < k < l$ *is* ϵ*-close to* F*, i.e.,*

$$\sup_{x \in \mathcal{B}_{d_0}} \|f(x) - F(x)\|_2 \leqslant \epsilon. \tag{6}$$

We next discuss the feasibility of these assumptions. Assumptions (i) and (ii) have already been used in Theorem 1. These two assumptions are common in both practice and theory. Since the target network F is usually a trained one, a universal distribution for all entries in a layer might not be realistic. Thus we have assumption (iii) to allow non-homogeneous distributions of the entries in the weight matrices. The two bounds $\mathbb{E}\left|\mathcal{X}_{i,j}^k\right|^2 \leqslant \frac{K_1}{\max\{d_{k+1}, d_k\}}$ and $\mathbb{E}\left|\mathcal{X}_{i,j}^k\right|^4 \leqslant \frac{K_2}{\max\{d_{k+1}, d_k\}^2}$ hold for a variety of distributions, like the uniform distribution, normal distribution, and sub-Gaussian distribution, as long as the variance of the distribution is set to $O\left(\frac{1}{d}\right)$. This holds because, if we initialize the target network F following the Xavier initialization and train the network properly, the variance of the weights should remain of the same order, approximately. We further verify that this assumption holds by checking the distribution of some trained FCNs and CNNs. We train a 5-hidden-layer FCN with 1024 neurons in each hidden layer on the `Covertype` dataset (Blackard & Dean, 1998) by randomly selecting initial weights. Figure 1 shows the histogram of weights in different layers of the trained FCN. They exhibit a sub-Gaussian distribution and the second-order and fourth-order moments are well bounded by $O\left(\frac{1}{d}\right)$ and $O\left(\frac{1}{d^2}\right)$, respectively. See Appendix D.1 for more details. Assumption (iv) bounds the operator norm of the weight matrices, which is an important term in the proof. Without loss of generality, we assume that $N_k \geqslant 1$. We can also have $\delta \geqslant \delta_l + \sum_{i=1}^{l-1}(l-i)\delta_i$. This can be achieved by increasing the value of N_k and thus reducing the value of δ_k. However, we should carefully choose the values of N_k's and δ_k's, as larger N_k's also increase the lower-bound of the minimum number of neurons in the target network. In fact, assumption (iv) with certain N_k's and δ_k's can be derived from assumption (iii) with Lemma 4 and the Markov's inequality. We use assumption (iv) as it allows possible tighter values.

4.3. Further Discussions and Proof Outlines

In conclusion, under two different schemes, Theorems 1 and 2 show that, under certain conditions, we can prune $\lfloor D_k^{1-\alpha} \rfloor, \alpha > 0$ entries in the k-th layer of F while keeping the pruned network f to be ϵ-close to F with positive probability $1-\delta$. It is obvious that we cannot set $\alpha = 0$ as it makes the weight-subnetwork f to be the zero function.

By fixing α, our theorems show that a lower-bound of the minimum width of the target network can be represented as a polynomial in $\frac{1}{\epsilon}$, $\frac{1}{\delta}$ and $\log\left(\frac{1}{\delta}\right)$. Note that the constants in the theorems can be significantly improved by a finer analysis, but this is not the focus of this work. For example, by carefully discussing the independency of the events given in (30) and (41) in Appendix B, we can improve the constants related to δ greatly. We can also give a finer upper-bound of the norm of the output of each layer by studying the corresponding distribution as a whole; in this case, we can even set the constant $C_2 = 1$ in Theorems 1 and 2. The same argument holds for Theorem 3 (presented later) as well.

Next, we give a sketch on the universal framework for proving the theorems in this paper. For simplicity, we remove the statement about probabilities and use C and C' to denote universal positive constants which may vary by occurrence in this section. The details of the probabilities and constants are given in the full proof in Appendix B.

We use $y_k(x)$ and $y_k^*(x)$ to denote the output of the k-th layer of f and F, respectively. The basic building block of the proof is showing how to iteratively bound the error between $y_k(x)$ and $y_k^*(x)$. This is achieved by inducting on the upper-bounds of $\|y_k(x) - y_k^*(x)\|_2$ and $\|y_k^*(x)\|_2$ at the same time. Intuitively, we expect the error $\|y_k(x) - y_k^*(x)\|_2$ to be small and that the norm of the output $\|y_k^*(x)\|_2$ is not exploding.

By the Lipschitz continuity of the activation functions and several matrix norm inequalities, we show that the above norms heavily depend on bounding two random variables $\|W_k^*\|_2$ and $\|W_k - W_k^*\|_2$.

Recall that we assume different distributions for the weights in the two theorems. For example, in Theorem 1 we assume the entries in W_k^* are uniformly distributed. By Lemma 3 and the Markov's inequality, we derive the probability that $\|W_k^*\|_2 \leqslant C$. A similar approach, depending on the specific distribution we assume, is applied in the other theorems to upper-bound the probability.

We want to make sure that $\|W_k - W_k^*\|_2$ is small so that the gap between outputs can be small as well. In this sense, we cannot bound the two matrices W_k and W_k^* separately. Instead, we use the fact that $W_k - W_k^*$ is a zero matrix except for those pruned entries. In Theorem 1 we apply the closed-form order statistics of the uniform distribution to give a precise upper-bound of $\|W_k - W_k^*\|_2$, which is $O(d^{-\alpha})$. In the other proofs, we rely on the results of the "balls-into-bins" problem (Lemma 6) and the Latala's inequality (Lemma 4) to give similar upper-bounds.

The remaining part of the proofs are to estimate the probabilities that each event happens, and to determine the condi-

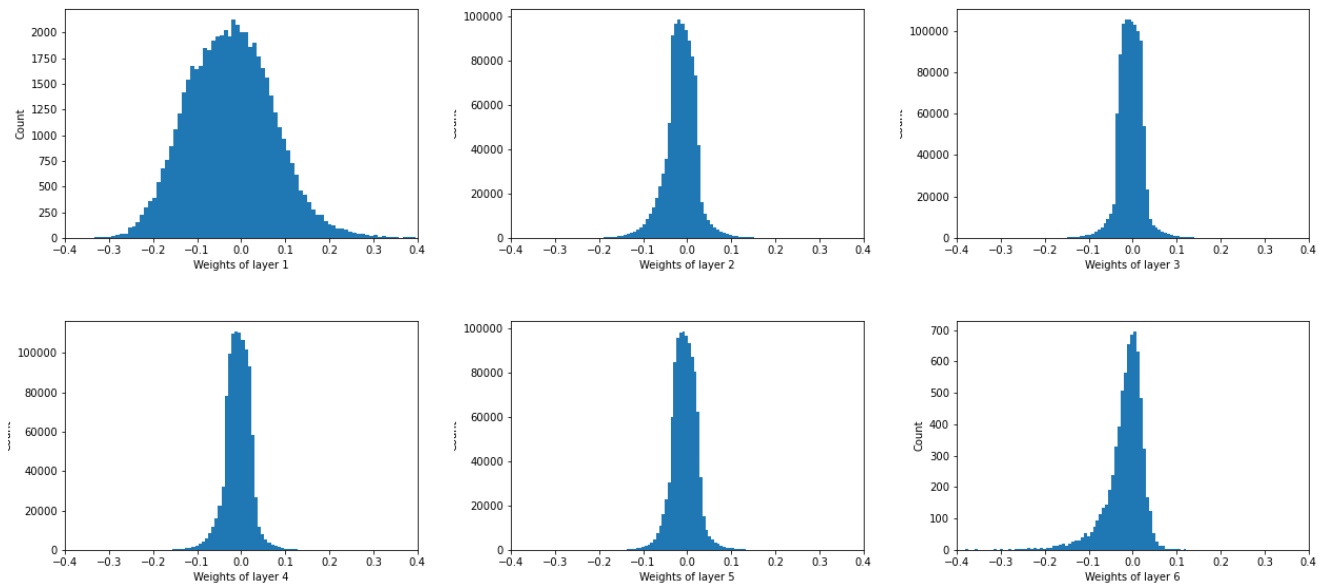

Figure 1: The histogram of entries of all the weight matrices from a trained FCN.

tions between variables d, ϵ, δ and α. For more details about the proofs, please refer to Appendix B.

5. Pruning Convolutional Neural Networks

In this section, we study model pruning of CNNs. We start with the mathematical definition of a single convolutional layer of CNN following the notations of Sedghi et al. (2018). We are given an input feature map $X \in \mathbb{R}^{d\times p\times p}$ where d denotes the number of input channels of the convolutional layer and p is the height/width of the input feature map[5]. The entry $(X)_{t,i,j}$ is the value of the input unit within channel t at row i and column j. The convolutional layer transforms X into an output feature map $Y \in \mathbb{R}^{d'\times p'\times p'}$, which becomes the input to the next convolutional layer. This is achieved by applying d' 3D filters $\mathcal{F}_s \in \mathbb{R}^{d\times q\times q}$ on the d input channels of X, where each $\mathcal{F}_s$ generates the s-th channel of $Y, s \in [d']$, and $p > q$. Each filter $\mathcal{F}_s$ is composed by d 2D convolutional kernels (we use kernels for abbreviation in the sequel) $\mathcal{F}_{s,t} \in \mathbb{R}^{q\times q}, t \in [d]$. All the filters are combined to form the convolutional tensor $\mathcal{F} \in \mathbb{R}^{d'\times d\times q\times q}$. Mathematically, we have $\mathcal{F}_{s,t} = \mathcal{F}_{s,t,:,:}$ and $\mathcal{F}_s = \mathcal{F}_{s,:,:,:}, s \in [d'], t \in [d]$.

Filter $\mathcal{F}_s$ is moved along the second and third axes of X to get the output feature maps. We assume that the stride is 1, i.e., we move the filter $\mathcal{F}_i$ by 1 pixel/element around every time. Note that there are two types of padding: (i) zero padding where we wrap the input feature maps with zeros around the edges; (ii) wrap-around padding where we pad the input feature maps in such a way that, if a pixel/element that is off the right end of the image is called by the filter, we use the pixel/element from the left end of the image instead; we do this similarly for all the edges and axes; mathematically, we set $X_{t,i,j} = X_{t,i\%p,j\%p}$. Throughout the paper, we use the second approach for padding, as it leads to a circulant representation of the filters[6].

With wrap-around padding and stride 1, the width and height of the output feature map are the same as the input feature map, i.e. we have $p = p'$. Let K be the $d' \times d \times p \times p$ tensor such that

$$K_{s,t,:,:} = \begin{bmatrix} \mathcal{F}_{s,t,:,:} & \mathbf{0}_{q\times(p-q)} \\ \mathbf{0}_{(p-q)\times q} & \mathbf{0}_{(p-q)\times(p-q)} \end{bmatrix}, s \in [d'], t \in [d]. \quad (7)$$

Then for $s \in [d'], a, b \in [p]$, we have

$$Y_{s,a,b} = \sum_{t\in[d]} \sum_{i\in[p]} \sum_{j\in[p]} X_{t,(a+i-1)\%p,(b+j-1)\%p} K_{s,t,i,j}.$$

For vector $a = (a_1, \ldots, a_n)^T$, we define

$$\mathrm{circ}(a) := \begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ a_n & a_1 & \cdots & a_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_2 & a_3 & \cdots & a_1 \end{bmatrix}.$$

[5] We assume that the input feature map has the same width and height for simplicity. All the statement in this section can be generalized to fit different width and height.

[6] The first approach leads to the Toeplitz representation and there exist numerous discussions regarding the error and (non-) asymptotic relationship between these two approaches in the CNN literature (Sedghi et al., 2018) and the matrix analysis literature (Gray, 2006; Zhu & Wakin, 2017). The error gap can be bounded by $O\left(\frac{1}{n}\right)$, where n is the dimension of the matrix.

Sedghi et al. (2018) show that a linear transformation $W \in \mathbb{R}^{p^2d' \times p^2 d}$ which satisfies $\text{vec}(Y) = W\text{vec}(X)$ can be represented by

$$W = \begin{bmatrix} B_{1,1} & \cdots & B_{1,d} \\ \vdots & \ddots & \vdots \\ B_{d',1} & \cdots & B_{d',d} \end{bmatrix}, \tag{8}$$

where each $B_{s,t}$ is a doubly block circulant matrix such that

$$B_{s,t} = \begin{bmatrix} \text{circ}\left(K_{s,t,1,:}\right) & \text{circ}\left(K_{s,t,2,:}\right) & \cdots & \text{circ}\left(K_{s,t,p,:}\right) \\ \text{circ}\left(K_{s,t,p,:}\right) & \text{circ}\left(K_{s,t,1,:}\right) & \cdots & \text{circ}\left(K_{s,t,p-1,:}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \text{circ}\left(K_{s,t,2,:}\right) & \text{circ}\left(K_{s,t,3,:}\right) & \cdots & \text{circ}\left(K_{s,t,1,:}\right) \end{bmatrix}. \tag{9}$$

Now we discuss the formulation of a convolutional neural network. Formally, consider a CNN F of depth $l \geqslant 3$. For $1 \leqslant k < l$, the k-th convolutional layer of F takes the input feature map of dimension $d_{k-1} \times p_{k-1} \times p_{k-1}$, and transforms it to an output feature map of dimension $d_k \times p_k \times p_k$ by applying the convolutional tensor $\mathcal{F}^{(k)} \in \mathbb{R}^{d_k \times d_{k-1} \times q_{k-1} \times q_{k-1}}$. Then we pass the output feature map through an activation function σ and feed it into the next layer. The last layer is a fully-connected layer that maps the output tensor of the previous layer with dimension $d_{l-1} \times p_{l-1} \times p_{l-1}$ into a vector of dimension d_l by matrix $W_l^* \in \mathbb{R}^{d_l \times d_{l-1} p_{l-1}^2}$. Mathematically, by reshaping the convolutional tensor $\mathcal{F}^{(k)}$ into the corresponding linear mapping $W_k^* = \left[B_{s,t}^{(k)}\right]_{s\in[d_k],t\in[d_{k-1}]}$, where $B_{s,t}^{(k)}$ is the doubly block circulant matrix induced by $K_{s,t,:,:}^{(k)}$ as defined in (7) – (9), we write the convolutional neural network as

$$F(x) = W_l^* \sigma\left(W_{l-1}^* \sigma_l\left(\cdots W_2^* \sigma\left(W_1^* x\right)\right)\right). \tag{10}$$

Similar to the definition of weight- and neuron-subnetworks of FCN, there are two analogous definitions for CNN. We define the channel-subnetwork of F as achieved by removing several 3D channels from the 4D tensor $\mathcal{F}^{(k)}, 1 \leqslant k < l$[7], and the filter-subnetwork of F by removing several 2D filters from the 4D tensor $\mathcal{F}^{(k)}$. The channel-subnetwork of F is equivalent to setting rows of W_k^* in terms of equation (10) (we are actually setting several rows of block matrices $B_{s,t}^{(k)}$) to be zero while the filter-subnetwork refers to setting some block sub-matrices $B_{s,t}^{(k)}$ of W_k^* to be zero. In the following, we focus on filter-subnetworks and present the result of random pruning on CNNs. We discuss the magnitude-based pruning of CNNs in Appendix C.2.

[7] In practice we usually remove the whole channel and hence reduce the size of $\mathcal{F}^{(k)}$ to $\mathbb{R}^{d_k' \times d_{k-1} \times q_{k-1} \times q_{k-1}}$ with $d_k' < d_k$. The size of the input of the next layer is also reduced to $d_k' \times p_k \times p_k$. In the presentation of this paper, we set the pruned channels to zero instead of removing them. It helps us to keep the dimension of pruned and original tensors to be the same while not changing any theoretical property of the CNNs.

For ease of presentation, in the theorem below, we assume that the number of channels and the width/height of each channel in all convolutional layers are equal, i.e., we define $d := d_1 = \cdots = d_{l-1}$ and $p := p_1 = \cdots p_{l-1}$. A similar result can be derived by the same approach for the general non-homogeneous setting.

Theorem 3. *We are given a target network F as defined in (10) and we denote by $\mathcal{F}^{(k)} \in \mathbb{R}^{d_k \times d_{k-1} \times p_k \times p_k}$ the convolutional tensor corresponding to $W_k^*, 1 \leqslant k < l$. Let us assume that*

(i) σ is L-Lipschitz and $\sigma(0) = 0$;

(ii) $d \geqslant \max\{d_0, d_l\}$;

(iii) for $s \in [d_k], t \in [d_{k-1}], i, j \in [p], k \in [l-1]$, $\mathcal{F}_{s,t,i,j}^{(k)}$ independently follows a distribution $\mathcal{X}_{s,t,i,j}^k$; further, there exist two positive constants C_1 and C_2 such that $\mathbb{E}\mathcal{X}_{s,t,i,j}^k = 0$, $\mathbb{E}\left|\mathcal{X}_{s,t,i,j}^k\right|^2 \leqslant \frac{C_1}{p^2 d}$ and $\mathbb{E}\left|\mathcal{X}_{s,t,i,j}^k\right|^4 \leqslant \frac{C_2}{p^4 d^2}$; the weights in W_l^* follow distributions with the same second-order and fourth-order moment upper-bounds.

Let us consider the subnetwork f of F with mask $M = \{M_1, \ldots, M_l\}$ that randomly prunes $\left\lfloor d^{2-\alpha} \right\rfloor, 0 < \alpha \leqslant 2 - \frac{\log(d+1)+\log^{(2)}(d)}{\log(d)}$ filters in the k-th layer of F, $1 < k < l$. For any positive constants $\beta_1 \in (0,1)$ and $\beta_2 \in \left(0, \frac{1}{4}\alpha\right)$, with probability at least $\left(1 - d^{-\frac{1}{3}}\right)^{2(l-2)} \overline{p}$, where $\overline{p} := 1 - (l-2)C_4 \frac{q^2}{p} d^{-\frac{1}{4}\alpha+\beta_2} - \frac{l^2-l-2}{2} C_3 \frac{q^2}{p^{1-\beta_1}} - \frac{C_5}{p^{1-\beta_1}}$ over the randomness of masks and weights, we have

$$\begin{aligned} &\sup_{x \in \mathcal{C}_{p_0^2 d_0}} \|f(x) - F(x)\|_2 \\ &\leqslant p^{-\beta_1} L^{l-1} p_0 \sqrt{d} \left[p^{-\beta_1}\left(p^{-\beta_1} + d^{-\beta_2}\right)^{l-2} - p^{-(l-1)\beta_1}\right] \end{aligned} \tag{11}$$

for some positive constants C_3, C_4 and C_5 specified in the proof.

The first two assumptions are common in all the theorems we present. We next discuss the feasibility of assumption (iii). Since we translate the target CNN into a FCN form and there are $p^2 d$ neurons (instead of d neurons) in the k-th layer of $F, 1 < k < l$, we change the denominators in the upper-bounds of moments accordingly. The mathematical definition of the masks is also revised to fit the CNN structure. Here we set the mask M_k to be the 0-1 matrices such that its sub-matrices are blocks of the zero matrices and all one matrices based on (8). Condition $\alpha \leqslant 2 - \frac{\log(d+1)+\log^{(2)}(d)}{\log(d)}$ is used to guarantee that Lemma 6 holds. As d goes to infinity, the left-hand side goes to 1 and then α becomes less and less constrained. For example, for $d = 128$ or $1,024$, the bound reads $0.6729, 0.7205$, respectively.

We next argue that the probability $\left(1-d^{-\frac{1}{3}}\right)^{2(l-2)}\overline{p}$ is positive in many of the use cases. Note that $\frac{q^2}{d}$ and $\frac{q^2}{p}$ are close to zero as we usually take $q=1,3,5$ as the dimension of the kernel, $p=28$ for images of MNIST (LeCun et al., 1998) and $p=32$ for images of CIFAR-10 (Krizhevsky & Hinton, 2009). For images with 4K resolution, we have $p=3,840$ or $2,160$. The number of channels d varies from 64 to 512 in famous CNN architectures, like VGG16 (Simonyan & Zisserman, 2014) and ResNet (He et al., 2016). The closed-forms of constants C_3, C_4, and C_5 are presented in Appendix B.3. Similar to the discussions in Section 4.3, these constants can be significantly improved by a finer analysis, but this is not the focus of this work. It is easy to see that the right-hand side of (11) is positive and it converges to 0 as d goes to infinity. Thus, by taking β_1 and β_2 appropriately small and d to be large, we can make sure that the probability $\left(1-d^{-\frac{1}{3}}\right)^{2(l-2)}\overline{p}$ is positive while the upper-bound of the gap between the pruned and target networks is small. We also point out that $\left(1-d^{-\frac{1}{3}}\right)^{2(l-2)}$ is the probability with respect to masks and $\overline{p}$ with respect to weights. As a result, the statement holds "for almost all masks."

6. Discussion and Future Works

In this paper, we establish theoretical results of model pruning for FCNs and CNNs under different schemes with mild assumptions. For magnitude-based pruning, we show the sub-network f of F, which prunes $\lfloor D_k^{1-\alpha}\rfloor$ out of D_k smallest entries of the k-th layer of F, can approximate the expressive power of F on the unit ball or the unit cube with positive probability. For random pruning, we show that most random masks, which prune $\lfloor D_k^{1-\alpha}\rfloor$ out of D_k entries of the k-th layer of F, approximate the expressive power of F on the unit ball or the unit cube with positive probability. Our results are enabled by many results from the random matrix theory. The essential building block of our analysis is to iteratively show that the gap between the pruned and target weight matrices and the gap between the outputs of the k-th layer of the pruned and target networks are small.

This is one of the rare theoretical works that discusses pruning of FCNs and CNNs. We not only cover model pruning of general FCNs, but also establish the results regarding pruning CNNs. The results can be applied to a variety of other network structures given the fact that almost all networks can be represented by a stack of fully-connected layers. Our theorems can provide precious insights to the iterative magnitude-based pruning as suggested by Frankle & Carbin (2018). For example, our results are able to determine how many weights we can prune in each iteration and the corresponding probability that the gap between the pruned and target networks is smaller than a given error.

As discussed in Appendix C, a direct extension of this work is to consider magnitude-based pruning for general distributions. Appendix E.1 discusses the assumption about the independency of weights in the target network and provide many approaches to alleviate it, but a detailed and strict theoretical study is definitely attractive. Besides, we usually use pooling layers and residual connections in practical CNN models. It is interesting to consider the impact of such non-parametric functions and skip connections on the theoretical neural network pruning results. Another interesting problem is trying to leverage additional information (e.g., gradients) of the target network to improve our results. Besides, it would be exciting if our results can provide guidance to improve the existing magnitude-based and random pruning algorithms.

Acknowledgement

The authors would like to thank Yiqiao Zhong, Dawei Li, and Feiyi Xiao for critically reading the manuscript and helpful discussions.

References

Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. Stronger generalization bounds for deep nets via a compression approach. In Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pp. 254–263, 2018.

Bai, Y. and Lee, J. D. Beyond linearization: On quadratic and higher-order approximation of wide neural networks. In International Conference on Learning Representations, 2020.

Baykal, C., Liebenwein, L., Gilitschenski, I., Feldman, D., and Rus, D. Data-dependent coresets for compressing neural networks with applications to generalization bounds. In International Conference on Learning Representations, 2019a.

Baykal, C., Liebenwein, L., Gilitschenski, I., Feldman, D., and Rus, D. Sipping neural networks: Sensitivity-informed provable pruning of neural networks. arXiv preprint arXiv:1910.05422, 2019b.

Blackard, J. A. and Dean, D. J. Comparative accuracies of neural networks and discriminant analysis in predicting forest cover types from cartographic variables. In Second Southern Forestry GIS Conference, pp. 189–199, 1998.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G.,

Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Advances in Neural Information Processing Systems, 2020.

Cheng, Y., Wang, D., Zhou, P., and Zhang, T. A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282, 2017.

Chibisov, D. M. On limit distributions for order statistics. Theory of Probability & Its Applications, 9(1):142–148, 1964.

David, H. A. and Nagaraja, H. N. Order statistics. Encyclopedia of statistical sciences, 2004.

Elesedy, B., Kanade, V., and Teh, Y. W. Lottery tickets in linear models: An analysis of iterative magnitude pruning. arXiv preprint arXiv:2007.08243, 2020.

Fan, J., Ma, C., and Zhong, Y. A selective overview of deep learning. arXiv preprint arXiv:1904.05526, 2019.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In International Conference on Learning Representations, 2018.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the 39th International Conference on Artificial Intelligence and Statistics, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. Deep learning, volume 1. MIT press Cambridge, 2016.

Gray, R. M. Toeplitz and circulant matrices: A review. Foundations and Trends in Communications and Information Theory, 2(3):155–239, 2006.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. International Conference on Learning Representations, 2015a.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. In Advances in Neural Information Processing Systems, pp. 1135–1143, 2015b.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.

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

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In Advances in Neural Information Processing Systems, 2018.

Jain, A. K. Fundamentals of Digital Image Processing. Prentice-Hall, Englewood Cliffs, 1989.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical Report, 2009.

Latała, R. Some estimates of norms of random matrices. Proceedings of the American Mathematical Society, 133 (5):1273–1282, 2005.

LeCun, Y., Denker, J. S., Solla, S. A., Howard, R. E., and Jackel, L. D. Optimal brain damage. In Advances in Neural Information Processing Systems, volume 2, pp. 598–605, 1989.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. In International Conference on Learning Representations, 2016.

Liebenwein, L., Baykal, C., Lang, H., Feldman, D., and Rus, D. Provable filter pruning for efficient neural networks. In International Conference on Learning Representations, 2020.

Luo, J.-H., Wu, J., and Lin, W. Thinet: A filter level pruning method for deep neural network compression. In Proceedings of the IEEE International Conference on Computer Vision, pp. 5058–5066, 2017.

Malach, E., Yehudai, G., Shalev-Schwartz, S., and Shamir, O. Proving the lottery ticket hypothesis: Pruning is all you need. In International Conference on Machine Learning, pp. 6682–6691, 2020.

Orseau, L., Hutter, M., and Rivasplata, O. Logarithmic pruning is all you need. Advances in Neural Information Processing Systems, 33, 2020.

Pastur, L. A. and Shcherbina, M. Eigenvalue distribution of large random matrices. Number 171. American Mathematical Soc., 2011.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison,

M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems, pp. 8024–8035. 2019.

Pensia, A., Rajput, S., Nagle, A., Vishwakarma, H., and Papailiopoulos, D. Optimal lottery tickets via subset sum: Logarithmic over-parameterization is sufficient. Advances in Neural Information Processing Systems, 33, 2020.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

Ramanujan, V., Wortsman, M., Kembhavi, A., Farhadi, A., and Rastegari, M. What's hidden in a randomly weighted neural network? In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11893–11902, 2020.

Richa, A. W., Mitzenmacher, M., and Sitaraman, R. The power of two random choices: A survey of techniques and results. Combinatorial Optimization, 9:255–304, 2001.

Rudelson, M. and Vershynin, R. Non-asymptotic theory of random matrices: extreme singular values. In Proceedings of the International Congress of Mathematicians, pp. 1576–1602, 2010.

Sedghi, H., Gupta, V., and Long, P. M. The singular values of convolutional layers. In International Conference on Learning Representations, 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations, 2014.

Srinivas, S. and Babu, R. V. Data-free parameter pruning for deep neural networks. In Proceedings of the British Machine Vision Conference, 2015.

Sun, R. Optimization for deep learning: theory and algorithms. arXiv preprint arXiv:1912.08957, 2019.

Vershynin, R. Introduction to the non-asymptotic analysis of random matrices, pp. 210–268. Cambridge University Press, 2012. doi: 10.1017/CBO9780511794308.006.

Ye, M., Gong, C., Nie, L., Zhou, D., Klivans, A., and Liu, Q. Good subnetworks provably exist: Pruning via greedy forward selection. In International Conference on Machine Learning, pp. 10820–10830, 2020a.

Ye, M., Wu, L., and Liu, Q. Greedy optimization provably wins the lottery: Logarithmic number of winning tickets is enough. arXiv preprint arXiv:2010.15969, 2020b.

Zhou, H., Lan, J., Liu, R., and Yosinski, J. Deconstructing lottery tickets: Zeros, signs, and the supermask. In Advances in Neural Information Processing Systems, volume 32, pp. 3597–3607, 2019a.

Zhou, W., Veitch, V., Austern, M., Adams, R. P., and Orbanz, P. Non-vacuous generalization bounds at the imagenet scale: A pac-bayesian compression approach, 2019b.

Zhu, Z. and Wakin, M. B. On the asymptotic equivalence of circulant and toeplitz matrices. IEEE Transactions on Information Theory, 63(5):2975–2992, 2017.