Statistical Mechanical Analysis of Neural Network Pruning

Rupam Acharyya[1] **Ankani Chattoraj**[*2] **Boyu Zhang**[*1] **Shouman Das**[3] **Daniel Štefankovič**[1]

[1]Computer Science Dept., University of Rochester, Rochester, New York, USA
[2]Brain and Cognitive Science Dept., University of Rochester, Rochester, New York, USA
[3]Mathematics Dept., University of Rochester, Rochester, New York, USA

Abstract

Deep learning architectures with a huge number of parameters are often compressed using *pruning* techniques to ensure computational efficiency of inference during deployment. Despite multitude of empirical advances, there is a lack of theoretical understanding of the effectiveness of different pruning methods. We inspect different pruning techniques under the statistical mechanics formulation of a teacher-student framework and derive their generalization error (GE) bounds. It has been shown that *Determinantal Point Process* (DPP) based *node* pruning method is notably superior to competing approaches when tested on real datasets. Using GE bounds in the aforementioned setup we provide theoretical guarantees for their empirical observations. Another consistent finding in literature is that sparse neural networks (*edge pruned*) generalize better than dense neural networks (*node pruned*) for a fixed number of parameters. We use our theoretical setup to prove this finding and show that even the baseline *random edge pruning* method performs better than the *DPP node pruning* method. We also validate this empirically on real datasets.

1 INTRODUCTION

Deep neural networks have achieved impressive results in a wide variety of applications such as classification [23, 31], image processing [30, 4], natural language processing [8, 7, 42], etc. Most of these networks use millions and sometimes even billions of parameters which makes inference computationally expensive and memory intensive [8]. To address this, researchers explore pruning techniques with the primary goal of comparing performance on real datasets. The broad scientific paradigm explored by most pruning techniques is to empirically and heuristically determine either how to prune a network or what to prune in a network (sometimes both). In this work, we take a step towards theoretical understanding of these two prime aspects of pruning methods.

We compare the quality of different pruning methods for feedforward neural networks under the *teacher-student* framework [37, 38, 39, 13] in the thermodynamic limit (input dimension goes to infinity) using *generalization error bounds* (GE), a theoretical measure of performance of machine learning models on unseen test data [46].

A fairly recent work by [34] empirically investigates a node pruning technique where a diverse subset of nodes are preserved in a given layer using Determinantal Point Process (DPP) [32, 24]. We provide theoretical guarantees for their empirical observations thereby showing that DPP based node pruning outperforms two standard paradigms of pruning (magnitude based node pruning and random node pruning). Thus, in the first part of this paper, we take a step towards theoretical understanding of the question: how to prune?

For the second part of this work we focus our attention to the study by [6]. This study reviewed multiple papers across decade on various pruning methods and closely analyzed their empirical results to conclude that sparse models obtained after edge/connection (used interchangeably) pruning outperforms dense ones obtained after node pruning for a fixed number of parameters. We extend our theoretical setup and compare node and edge pruning techniques which are within the scope of our investigation, to provide a theoretical justification of their empirical observation driven claim, thereby addressing the question: what to prune?

Our work has multiple contributions with regard to theoretical advancements in the domain of pruning:

- We use GE bounds on the teacher-student framework to compare different pruning methods within a class,

[*]equal contribution

Accepted for the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021).

which to the best of our knowledge, is the first theoretical advance in comparing pruning methods.

- We prove that DPP node pruning outperforms random and importance node pruning methods, previously shown by [34] empirically.
- We also theoretically show and validate on real datasets (`MNIST` and `CIFAR10`), that baseline random edge pruning performs better than DPP node pruning (superior in the node pruning regime explored in this paper) which is consistent with empirical observations from pruning literature that sparse models outperform dense models [6].

2 RELATED WORK

Pruning Methods: Studies under node pruning regime remove entire neurons/nodes (used interchangeably henceforth) keeping the networks dense [17, 29, 18]. Our work is closely related to [34], where a DPP sampling technique is used to select a set of diverse neurons/nodes to be preserved during pruning. The authors also introduce a *reweighting* procedure to compensate contributions of the pruned neurons in the network. Finally, they compare DIVNET (DPP node pruning with reweighting as in [34]) with random and importance node pruning [17] on real datasets. Seminal studies on edge pruning [26, 16] remove unimportant network weights based on the Hessian of the network's error function. Among others, alternative approaches include low rank matrix factorization of the final weight layers [40] or pruning the unimportant connections below a threshold [15]. Though dense networks can benefit from modern hardware, sparse models outperform dense ones for a fixed number of parameters across domains [27, 21, 14]. In a recent review this is highlighted based on observations from investigating 81 studies on pruning techniques [6].

The various existing methods can be broadly subsumed into a couple of categories [6]. These categories are mainly governed by the principles of pruning heuristics. First category is the magnitude-based approaches which have been extensively studied both globally and layerwise [15, 11]. As per [6], magnitude-based approaches are not only good and common baselines in the literature but they also give comparable performance to other methods such as the gradient-based methods [27, 47]. Another category is the random pruning which serves as an useful baseline for showing superior performance of any other pruning technique. We hence show all our theoretical results w.r.t these two categories, random pruning and importance pruning (same in concept as magnitude based pruning). We do not focus on any specific algorithm within these categories but explore the general concept for theoretical results. There are recent advances in pruning techniques which are complementary to these approaches, such as, being data independent [5, 43], single shot [28, 45] etc. However, these are beyond the scope of our investigation.

Theoretical Advances Towards Understanding Neural Networks: Despite promising performance in empirical data, providing theoretical guarantees for neural networks remains a known challenge. Researchers have explained the training dynamics of neural network from the information theoretic perspective [44, 41]. In another direction of work the learning dynamics of neural networks with infinitely wide hidden layers are explored [19, 9, 2, 48]. Pioneering work by [37, 38, 39] analyzes the generalization dynamics form the statistical mechanics perspective on *teacher-student* framework [12] to understand the performance of neural networks on unseen test data. All our theoretical analyses throughout this work closely follow [1, 13], who analyzed results for the case where the student networks are over parameterized, i.e., it has more number of hidden nodes than the teacher network.

3 PRELIMINARIES

Determinantal Point Process (DPP): DPP [32] is a probability distribution over power set of a ground set $\mathcal{G}$, here finite. DPP is a special case of negatively associated distributions [20] which assigns higher probability mass on diverse subsets. Formally, a DPP with a marginal kernel L ($\in \mathbb{R}^{|\mathcal{G}|\times|\mathcal{G}|}$) is: $\mathbb{P}[\mathbf{Y} = Y] = \frac{\det(L_Y)}{\det(L+I)}$, where $Y \subseteq \mathcal{G}$ and L_Y is the principal submatrix defined by the indices of Y. We use k-DPP to denote the probability distribution over subsets of fixed size k.

DPP Node Pruning: [34] uses DPP to propose a novel node pruning method for feedforward neural network. They define information at node i of layer l as $\boldsymbol{a}_i^l (= (a_{i1}^l, \ldots, a_{in}^l))$, where a_{ij}^l is the activity of node i of layer l on j^{th} input. Here $\boldsymbol{a}_i^l = g\left(\boldsymbol{b}_i^l\right)$, where $\boldsymbol{b}_i^l = \sum_{j=1}^{n_{l-1}} w_{ji}^{l-1} \boldsymbol{a}_j^{l-1}$ is the information at node i of layer l before activation. A layer is pruned by choosing a subset of hidden nodes using a DPP kernel: $\boldsymbol{L}\ (= \boldsymbol{L'} + \epsilon \boldsymbol{I})$, where, $\boldsymbol{L'}_{st} = \exp(-\beta \left\|\boldsymbol{a}_s^l - \boldsymbol{a}_t^l\right\|^2)$ and β is a bandwidth parameter. The matrix $\boldsymbol{L}$ is of dimension $n_l \times n_l$, as total number of nodes in layer l is n_l. By the property of DPP, this procedure will keep a diverse subset of nodes for each layer w.r.t. information obtained from the training data. A *reweighting* technique (see Section 2.2 of [34]) is then applied to outgoing edges of retained nodes to compensate for information lost in that layer due to node removal.

Remark: DIVNET denotes DPP node pruning with reweighting as in [34].

Online Learning in Teacher-Student Setup [13]: We use a two-layer perceptron which has N input units, M hidden units and 1 output unit as the *teacher network* to generate labels for i.i.d Gaussian input, $\boldsymbol{x}^t = (x_1^t, \ldots, x_N^t)$ where $x_i^t \sim \mathcal{N}(0, 1)\ \forall i \in \{1, \ldots, N\}$. Let $\theta^* = \{\boldsymbol{w}^*(\in$

Table 1: Notations used in Theorems

Notations	Explanations	Notations	Explanations	Notations	Explanations
n	number of inputs	N	dimension of the input	n_l	number of nodes in layer l
v_i^l	i^{th} node in layer l $(1 \leq i \leq n_l)$	a_{ij}^l	activation of v_i^l on j^{th} input	M	number of teacher hidden nodes
e_{ij}^l	edge from v_i^l to v_j^{l+1} $(1 \leq i \leq n_l$ and $1 \leq j \leq n_{l+1})$	w_{ij}^l	weight of e_{ij}^l $(1 \leq i \leq n_l$ and $1 \leq j \leq n_{l+1})$	K	number of student hidden nodes
k_n	number of student hidden nodes kept after node pruning	k_e	number of incoming edges of a hidden node kept after edge pruning	v^*	second layer weight of teacher network

$\mathbb{R}^{M\times N}), v^* \in \mathbb{R}^M\}$ denote the fixed parameters of the teacher network. The label y^t of the input $\boldsymbol{x}^t$ $(t = 1, 2, \ldots)$ is given as,

$$y^t = \sum_{m=1}^{M} v_m^* g\left(\frac{w_m^* \boldsymbol{x}^t}{\sqrt{N}}\right) + \sigma\zeta^t, \tag{1}$$

where $\zeta^t \sim \mathcal{N}(0, 1)$ is the output noise, and g is the sigmoid activation function. The input and teacher generated labels $(\{(\boldsymbol{x}^1, y^1), \ldots\})$ are used to train a two-layer *student network* with N input units, K hidden units $(K \geq M)$ and 1 output unit using online SGD learning method. We consider the quadratic training loss, i.e.,

$$L(f) = \frac{1}{2}\left[\sum_{k=1}^{K} v_k g\left(\frac{w_k \boldsymbol{x}^t}{\sqrt{N}}\right) - y^t\right]^2, \tag{2}$$

where $f = \{\boldsymbol{w}, \boldsymbol{v}\}$ denotes the parameter of the student network. One of the key quantities for evaluating performance of neural network is *generalization error* (GE). For the teacher student setup the GE of teacher network f^* and student network f is denoted as $\epsilon(f, f^*)$. It is defined as,

$$\epsilon(f, f^*) = \frac{1}{2}\langle[\phi(x, f) - \phi(x, f^*)]^2\rangle,$$

where $\phi(x, f) = \sum_{k=1}^{K} v_k g\left(\frac{w_k \boldsymbol{x}^t}{\sqrt{N}}\right)$ and $\langle\cdot\rangle$ denotes average over input data distribution. In the teacher student setup the weight of the teacher network (f^*) is fixed beforehand. Hence, from now onward we will denote the GE as a function of the student network, i.e., as $\epsilon(f)$. [13] showed that GE $\epsilon(f)$ (expected error on the unseen data, for details see S31 of [13]) for the student network is a function of the following *order parameters*,

$$Q_{ik} = \frac{w_i^T w_k}{N}, \quad R_{in} = \frac{w_i^T w_n^*}{N}, \quad R_{mn} = \frac{w_m^{*T} w_n^*}{N}. \tag{3}$$

Intuitively, these order parameters measure the similarities between and within the hidden nodes of teacher and student networks. Our theoretical results assume [13]:

(A1) If $\boldsymbol{x} = (x_1, \ldots, x_N)$ is an input then $x_i \in \mathcal{N}(0, 1)$. Also, $N \to \infty$.

(A2) Both the teacher and the student networks have only one hidden layer.

(A3) $K \geq M$ and $K = Z \cdot M$ where $Z \in \mathbb{Z}^+$.

(A4) The activation in the hidden layer is sigmoidal for both teacher and student network.

(A5) The output $\in \mathbb{R}$ (i.e., regression problem).

(A6) The order parameters (see section 3) satisfy the ansatz as in (S58) - (S60) of [13]. This ansatz intuitively states that the every student hidden node specializes in learning a specific teacher hidden node and for each teacher hidden node there is a student hidden node which learns that teacher node.

(A7) No noise is added to the labels generated by the teacher network, i.e., $\sigma = 0$ in (1).

4 GE OF PRUNED NETWORK IN TEACHER-STUDENT SETUP

We compare the performance of student networks pruned using different techniques as in Table 2 by analyzing their GE (see Figure 1). For node and edge pruning comparison, we choose the parameters k_n and k_e (see Table 1) such that the total number of parameters of the networks remain same, i.e., they satisfy,

$$\frac{k_n}{K} = \lim_{N\to\infty} \frac{k_e}{N} = c, \tag{4}$$

where $c \in [0, 1]$ is a constant. It is important to note that since we assume that the number of student nodes is more than the number of teacher nodes, which means multiple student nodes learn the same teacher node (see Figure 3 of [13]; also in Figure 1: two student hidden nodes learn one teacher hidden node, shown in same color). From [13], we know that, in noiseless case ($\sigma = 0$ in (1)), the student network learns the teacher network completely when trained till convergence, i.e., the GE becomes 0. When we prune the student network, this GE increases, which we then analyze for different types of pruning under certain assumptions (see Section 3 (A1)-(A7)).

Table 2: Different pruning methods and notations for their GE. Here f denotes the pruned student network. u.a.r. and w.p. stand for *uniformly at random* and *with probability* resepectively.

Pruning Method	**Procedure**	**Retained Parameters**	**GE without reweighting**	**GE with reweighting**
Random Node	Keep k_n nodes u.a.r.	k_n hidden nodes	$\epsilon_{k_n}^{Rand\ Node}(f)$	$\hat{\epsilon}_{k_n}^{Rand\ Node}(f)$
Importance Node	[17]	k_n hidden nodes	$\epsilon_{k_n}^{Imp\ Node}(f)$	$\hat{\epsilon}_{k_n}^{Imp\ Node}(f)$
DPP Node	see Section 3	k_n hidden nodes	$\epsilon_{k_n}^{DPP\ Node}(f)$	$\hat{\epsilon}_{k_n}^{DPP\ Node}(f)$
Random Edge	Keep an edge w.p. c for each hidden node	k_e incoming edges per hidden node	$\epsilon_{k_e}^{Rand\ Edge}(f)$	$\hat{\epsilon}_{k_e}^{Rand\ Edge}(f)$

4.1 COMPARING NODE PRUNING METHODS

We theoretically show that the increment in GE due to DIVNET is less than that for random and importance node pruning methods, justifying the empirical findings of [34]. The proof proceeds with the following steps: (1) Theorem 1 provides a closed form expression of the GE after DPP node pruning. (2) Theorem 2 shows that: (a) GE of random node pruning is greater than GE of DPP node pruning (b) GE of random node pruning with reweighting is greater than GE of DIVNET (c) GE of importance node pruning is greater than GE of DIVNET.

Theorem 1. *Assume $(A1) - (A7)$. Let $k_n \leq M$ nodes are selected by the DPP Node pruning method,*

$$\epsilon_{k_n}^{DPPNode}(f) = (v^*)^2 \left[\frac{k_n}{6} \left(1 - \frac{1}{Z}\right)^2 + \frac{M - k_n}{6} \right] \tag{5}$$

and

$$\hat{\epsilon}_{k_n}^{DPPNode}(f) = (M - k_n) \times \frac{(v^*)^2}{6}. \tag{6}$$

Proof Idea of Theorem 1: Proof of the above theorem (details in the appendix C) is based on two factors: (1) Results from [13] assure that analyzing the *order parameters* is enough to obtain closed form of GE. (2) We exploit the observation that the expected kernel of the DPP node pruning is same as the order parameter Q (see appendix B for proof and Figure 2 E) which, following [13], is a block diagonal matrix with M blocks. By property of DPP, the pruning method will retain a subset of student hidden nodes with at most 1 hidden node from each block when $k_n \leq M$ (see Figure 2 G).

Remark 1. *As the expected DPP kernel is block-diagonal matrix, the stochasticity in subset selection via DPP does not impact GE when subset size is fixed and it only depends on size of pruned subsets.*

Remark 2. *Our theorem uses $k_n \leq M$, however, in practice the kernel may have non-zero off-diagonal entries when the assumption (A1) about input data is violated. As a result the probability of sampling a subset of size $k_n > M$ may be nonzero.*

Connection to Lottery Ticket Hypothesis: An interesting direction of research is to find small sub-networks from an overparameterized network with comparable performance. The existence of such networks is hypothesized in Lottery Ticket hypothesis [10]. Interestingly, recent work shows that pruning helps find such networks even without retraining [36, 33] and in our work we explore a sub-network in the teacher student setup.

Note that from Eq (6), when M student nodes are kept after pruning, i.e., $k_n = M$, then the GE of the DPP node pruned network is 0 which is GE of the original student network. Hence, from the fact that $K > M$ we can conclude that DPP node pruning can find out the winning ticket, i.e., a small sub-network with much less number of parameters than the original unpruned network but with same performance guarantee.

Theorem 2. *Assume $(A1) - (A7)$. Then for $k_n \leq M$ we have,*

$$\mathbb{E}_f \left[\epsilon_{k_n}^{Rand\ Node}(f)\right] > \epsilon_{k_n}^{DPP\ Node}(f') \tag{7}$$

and

$$\mathbb{E}_f \left[\hat{\epsilon}_{k_n}^{Rand\ Node}(f)\right] > \hat{\epsilon}_{k_n}^{DPP\ Node}(f') \tag{8}$$

and,

$$\epsilon_{k_n}^{Imp\ Node}(f') > \hat{\epsilon}_{k_n}^{DPP\ Node}(f'), \tag{9}$$

i.e., DPP node pruning outperforms random node pruning in the above setup. Here the expectation is taken over the the subsets of hidden nodes of size k_n chosen u.a.r.

Remark 3. *Reweighting for DPP/random node pruning follow procedure in Section 2.2 of [34].*

Proof Idea of Theorem 2: In random and importance node pruning, two student nodes which learn the same teacher node may both survive after pruning with non-zero probability, unlike DPP node pruning (Figure 1 (B)). Hence, more teacher nodes may remain unexplained by the student network after random or importance node pruning, resulting in increased GE (details in appendix C).

Together, Theorem 1 and 2 gives theoretical guarantees for all empirical results of [34]. Theorem 1 further allows us to

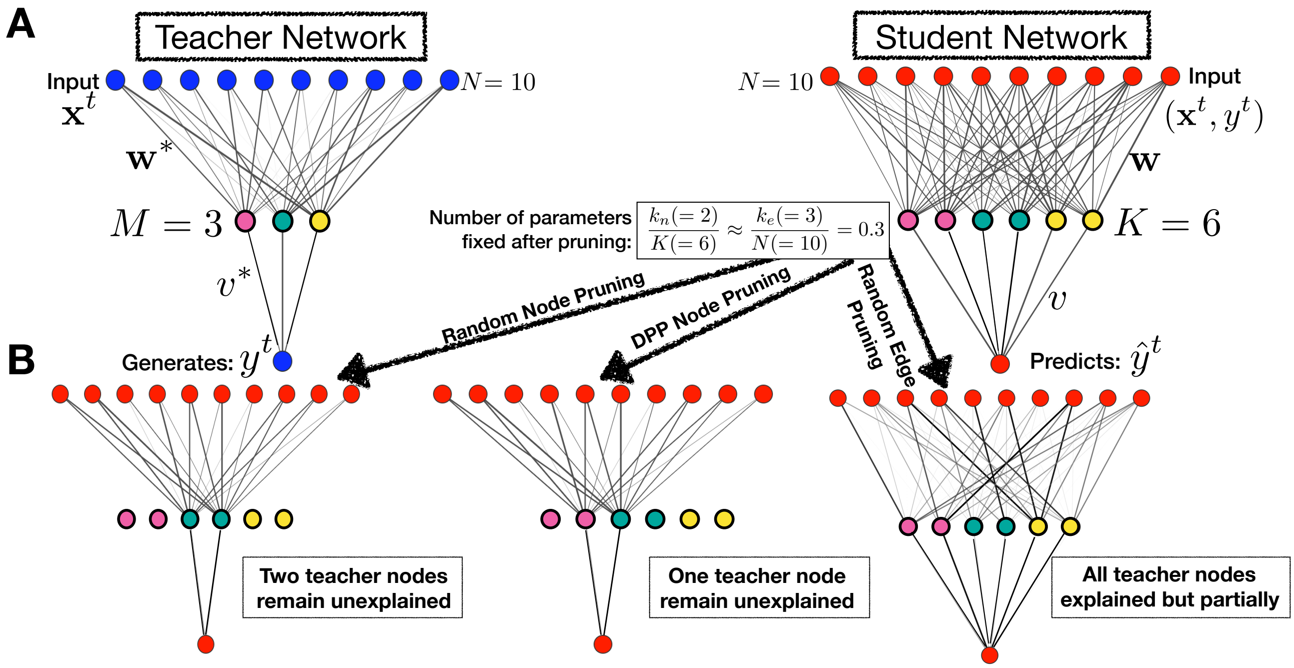

Figure 1: **(A)** Two layer teacher-student framework: A teacher neural network with 3 hidden nodes (left) and a student network with 6 hidden nodes (right). Input data (i.i.d) along with its label generated by teacher network are fed to student network to predict. **(B)** Intuitive example for 3 types of pruning on student network. For $k_n = 2$, random node pruning might only be able to explain 1 teacher hidden node, whereas DPP node pruning will always retain (partial) information about 2 teacher hidden nodes, hence preforms better. Random edge pruning retains sparse information about all 3 teacher nodes which is enough to outperform DPP node pruning. All notations follow Table 1.

show that DIVNET indeed satisfies the stronger version of Lottery Ticket Hypothesis as recently explored in [36, 35]. Importance node pruning with reweighting may be better than DIVNET and was not explored in [34].

4.2 COMPARING NODE AND EDGE PRUNING METHODS

In random edge pruning method, for each student hidden node, an incoming edge is kept with probability $c = \lim_{N\to\infty} \frac{k_e}{N}$. Majority of empirical studies throughout literature use random edge or node pruning as a baseline for empirical comparison (see papers in [6]) making it an obvious candidate for our theoretical comparisons as well. It has been shown empirically by [34] and theoretically by us that DPP node pruning is an above baseline node pruning method. In this section we show that baseline random edge pruning outperforms DPP node pruning which is consistent with the empirical observations that sparse models outperform dense models (section 3.2 of [6]). Specifically, here we show that GE after random edge pruning is less than GE after DPP node pruning. Our proof proceeds as follows: (1) Theorem 3 gives a closed form expression for the GE after random edge pruning (2) Theorem 4 then shows that GE of random edge pruning is less than GE of DPP node pruning.

Theorem 3. *Assume $(A1) - (A7)$. Consider the random edge pruning method with parameter $\lim_{N\to\infty} \frac{k_e}{N} = c$ (here c is a constant between 0 and 1). Then the GE $\epsilon_c^{Rand\ Edge}(\mathbb{E}[f])$ is,*

$$\frac{M(v^*)^2}{\pi}\left[\frac{1}{Z}\arcsin\frac{c}{1+c} + \left(1-\frac{1}{Z}\right)\arcsin\frac{c^2}{1+c} + \frac{\pi}{6} - 2\arcsin\frac{c}{\sqrt{2(1+c)}}\right]. \tag{10}$$

Remark 4. *Theorem 3 gives closed form for "GE of the expected network" after pruning instead of the "expected GE of the network" after pruning. However, in the thermodynamic limit ($N \to \infty$), the order parameters as in Section 3 are highly concentrated near their expected values and the two quantities hence become equal.*

Theorem 4. *Assume $(A1) - (A7)$. Let k_n and c satisfy* (4), *and $0 \le c \le \frac{1}{Z}$ and $Z \ge 4$. Then*

$$\epsilon_{k_n}^{DPP\ Node}(f) \ge \epsilon_c^{Rand\ Edge}(\mathbb{E}[f]), \tag{11}$$

i.e., Random edge pruning outperforms DPP node pruning in the above setup.

Proof Idea of Theorem 4: When $k_n \le M$, node pruned student network leaves at least $(M - k_n)$ teacher nodes unexplained, whereas after random edge pruning, student network can retain at least partial information about every teacher node (see Figure 1 (B)). After a pruning routine, the sum of partial information about all teacher nodes in an edge pruned student network dominates the sum of information for the explained subset of teacher nodes in a node pruned student network.

Observations: From Theorem 2 and 4, we conclude that random edge pruning outperforms random node pruning. Further, using Theorem 2 and the intuition that importance edge pruning is better than random edge pruning, we expect that importance edge pruning will outperform importance node pruning. Figure 2 confirms this empirically in the teacher student setup. These observations leads to the conjecture that for a fixed pruning method, edge pruning outperforms node pruning.

Conjecture 1. *Assume* $(A1) - (A7)$. *Let* k_n *and* c *satisfy* (4) *and* $Prune$ *denotes a fixed pruning method (e.g. Rand, Imp) which can be applied to both node and edge. Then,* $\exists c_\epsilon \in (0, 1]$ *such that for* $0 \le c \le c_\epsilon$,

$$\epsilon_{k_n}^{Prune\ Node}(f) \ge \epsilon_c^{Prune\ Edge}(f). \tag{12}$$

Together, Theorem 3, 4 and Conjecture 1 are consistent with empirical observations of [6]: sparse networks after edge pruning perform better on the unseen test data than dense networks after node pruning with fixed number of parameters. To the best of our knowledge, [6] based their claims from empirical observations of pruning studies in which the pruned networks were not reweighted. This motivated our choice of comparing GE for DPP node pruning and random edge pruning without any reweighting. However, with reweighting from [34], GE of DPP node pruning will be less than GE of random edge pruning, highlighting the impact of reweighting proposed by [34] (proof and details in appendix C).

We find that GE analysis on teacher-student setup is flexible for various pruning methods and this framework can be extended to theoretically understand other pruning methods which are outside the scope of this work.

5 EXPERIMENTS

5.1 SIMULATIONS

We run the DPP node, random edge/node, and importance edge/node pruning simulations under the teacher-student setup. For all the simulations, we sampled the 800000 i.i.d input samples from $\mathcal{N}(0, 1)$ as training data and 80000 as testing data. Following notations from Table 1, we set $M = 2$, $K = 6$, $N = 500$, and $v^* = 4$. The first layer teacher network weights $\boldsymbol{w}^*$ and all the student network parameters $\theta = \{\boldsymbol{w}, \boldsymbol{v}\}$ were drawn independently from $\mathcal{N}(0, 1)$ as initialization. We choose learning rate $\eta = 0.50$, and it is scaled to $\frac{\eta}{\sqrt{N}}$ for $\boldsymbol{w}$ and $\frac{\eta}{N}$ for $\boldsymbol{v}$. We run the simulations for both noiseless ($\sigma = 0$ in (1)) and noisy ($\sigma = 0.25$) output labels. For comparisons between node and edge pruning, we use the node-to-edge ratio $[1 : 83, 2 : 166, 3 : 250, 4 : 333, 5 : 417, 6 : 500]$ to keep the number of parameters the same, given $N = 500$, $K = 6$, and $M = 2$. In addition, we run the same simulation with $K = 5$ and $M = 20$, see Figure 2D. For other simulation details and results, see appendix F. Note that no pruning method undergoes reweighting for reported simulation results which we therefore use to verify and validate our theoretical results without reweighting.

Key Observations:

- The expected kernel of the DPP node pruning and the Q matrix are the same which we exploit for Theorem 1 (Figure 2E,F).
- For $k_n = 2$ and $M = 2$, DPP node pruning chooses exactly one node from each of the diagonal block of the kernel (see Figure 2G) which validates Theorem 1.
- DPP node pruning outperforms random and importance node in both noisy and noiseless case (see Figure 2C) which confirms Theorem 2.
- Random edge pruning is better than DPP node pruning for $c \le \frac{1}{Z}$ with $Z = 4$ and $M = 5$ in both noisy and noiseless cases (see Figure 2D), validating Theorem 4.
- We see Conjecture 1 holds for random, importance edge and node pruning (see Figure 2A,B)

5.2 REAL DATA

In this section, we compare DIVNET by [34] with random edge pruning with reweighting, and importance edge pruning with reweighting on the `MNIST` [25] and `CIFAR10` [22] datasets. We used the exact same network architectures as in Table 1 of [34] for `MNIST` and `CIFAR10`, respectively. Note that, for the real data we consider network structures with multiple layers. Following [34], we performed all pruning methods on the first layer. We compare the number of parameters as $k_e = \frac{k_n(d_{\text{input}} + h_2)}{h_1} - h_2$ where k_e is the number of edges kept for each node in edge pruning, and k_n is the number of nodes kept in the hidden layer for node pruning; d_{input}, h_1, and h_2 represent the dimension of the input, size of the first hidden layer, and size of the second hidden layer, respectively. As in [34], $h_1 = h_2$. We trained our model until the training error reaches predefined thresholds (Table 1 in [34]) and then perform the pruning. For hyperparameters and other details, see appendix F.

Remark: Note that we have not presented the results comparing different node pruning methods among themselves as they were already discussed in [34].

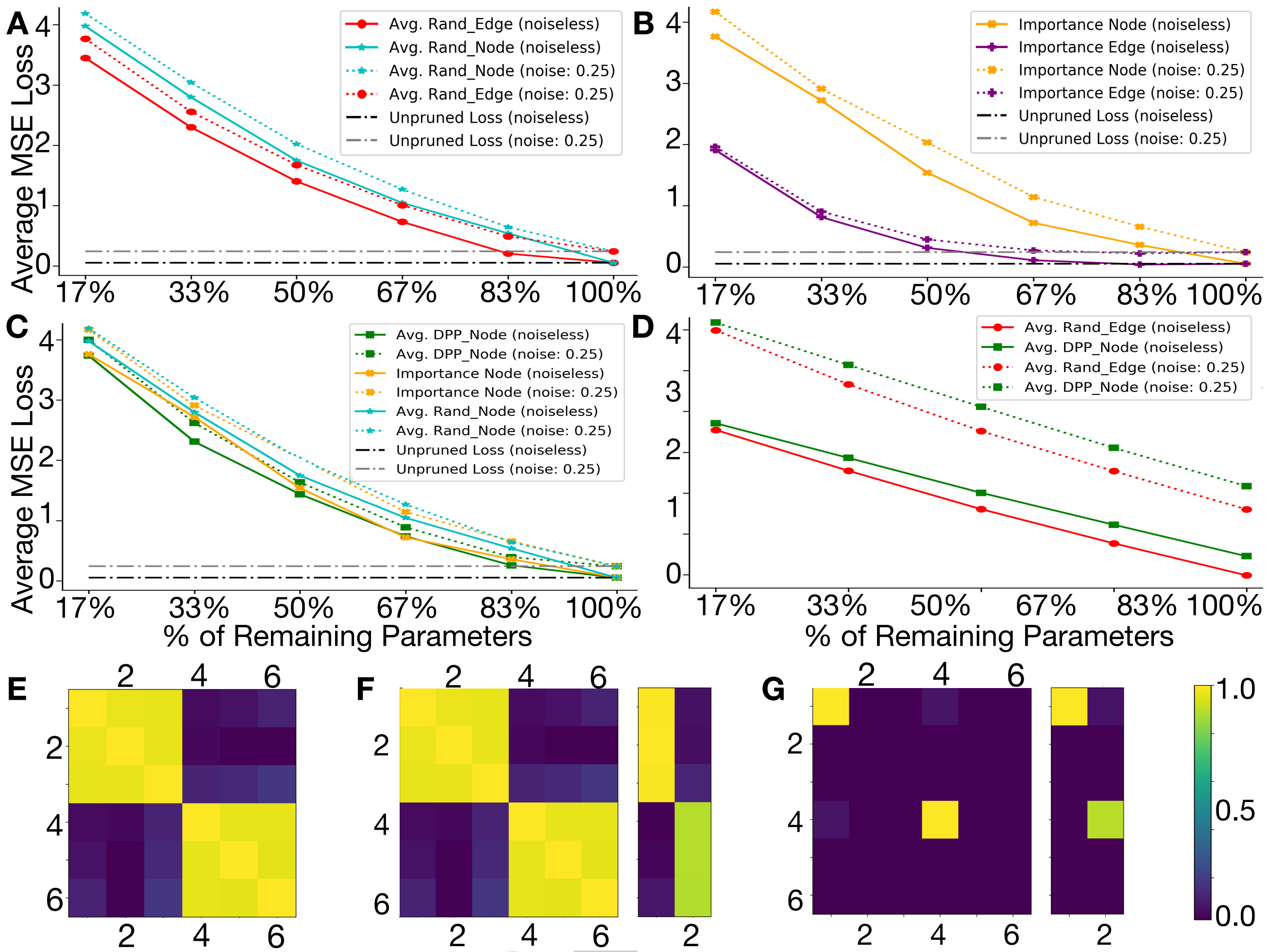

Figure 2: Simulation results in teacher student setup, $M = 2$ and $K = 6$ for (A-G). **(A-B)** Edge pruned networks perform better than node pruned networks in all 3 types of pruning methods (random (A), importance (B)), validating Conjecture 1. **(C)** DPP Node pruning performs better than importance and random node pruning (Theorem 2) **(D)** Baseline random edge pruning beats DPP node pruning (Theorem 4). For (D), $M = 5$ and $K = 20$. **(E)** The kernel of DPP node pruning is same as Q **(F)** Order parameters, Q (same as (E)) and R of the unpruned student network. **(G)** When only keeping 2 nodes, DPP node pruned student network keeps one from each block shown in (G).

Key observations:

- Baseline random edge pruning method outperforms DIVNET across all percentages of parameters retained in the network for `CIFAR10` dataset shown in Fig 3 B. However, for `MNIST` dataset, DIVNET performs better than random edge intitally but if $> 40\%$ of parameters are retained in the network random edge outperforms DIVNET (see Fig 3 A).

- Importance edge pruning performs better than both DIVNET and the baseline random edge pruning method on both the real data sets which highlighting the potential of magnitude based pruning method (see Fig 3 A and B).

6 DISCUSSION AND FUTURE WORK

Our work takes the first step to develop theoretical comparison for empirical observations of pruning methods in feed forward neural networks. We identify the usefulness of teacher-student setup for providing theoretical guarantees of pruning methods. We then use this setup to theoretically show that DIVNET should indeed outperform random and importance node pruning techniques. We further show that random edge pruning outperforms DPP node pruning providing a theoretical proof for the popular empirical observation: sparse (edge pruned) networks perform better than dense (node pruned) pruned networks for fixed number of parameters. Finally, we also are able to show that DIVNET satisfies a stronger version of the Lottery Ticket Hypothesis. Our work consolidates the understanding of a particular class of

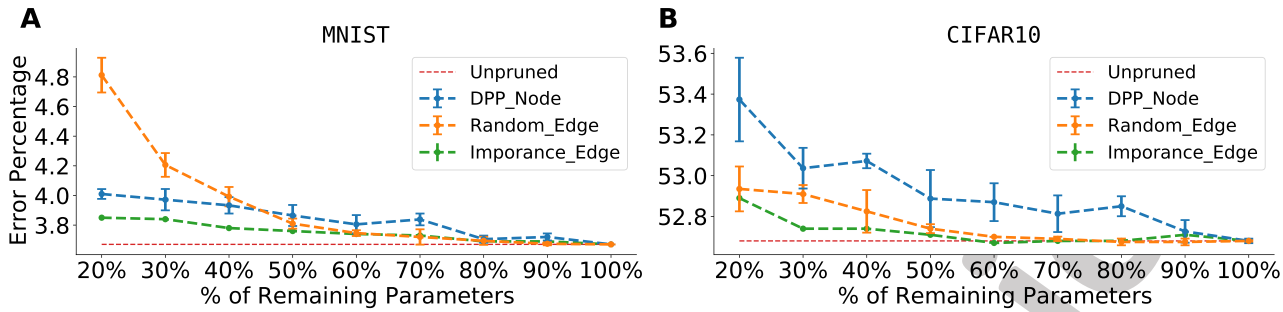

Figure 3: Comparing different edge pruning methods with DPP Node pruning method on the `MNIST` (A) and `CIFAR10` (B) dataset. Horizontal axis represents the percentage of remaining parameters in 1^{st} layer after pruning. The vertical axis shows corresponding test error. Both magnitude based edge pruning method (importance pruning) and baseline random edge pruning method outperforms DPP Node pruning which confirms Theorem 4 and the conjecture proposed in [6].

node and edge pruning theoretically.

When comparing two neural networks, using the number of parameters may not always be the optimal choice, instead, measuring the *capacity* and *expressiveness* of neural networks [3] can provide new insights. All our theoretical results have been proved on single hidden layer neural networks which gives future scope of extending them to multiple hidden layer networks. However, our empirical results hold for neural networks with multiple hidden layers suggesting the possibility of generalization of our results.

Throughout this work, we focus only on pruning methods in which a feedforward pre-trained neural network is pruned once without retraining. We choose this class for two primary reasons: (1) it is feasible to make theoretical comparisons with closed form solutions of GE, and, (2) with some assumptions, it has been shown by recent studies [36, 33, 35] that every sufficiently over-parameterized network contains a sub network that, even without training, achieves comparable accuracy to the trained large network. This proven conjecture is even stronger than the Lottery Ticket Hypothesis [10]. Hence, comparing performance of pruning methods within the aforementioned class in the teacher-student setup allowed us explore the existence of such a sub network.

We compare our theoretical results with random pruning and importance pruning which subsumes ideas underlying vast majority of pruning techniques and do not focus on any specific algorithm. A more specific algorithm based justification can also be an extension (may not always be trivial however) of this paradigm.

We introduce the teacher-student setup for proving results related to pruning methods which can further be extended to prove other empirical results in the pruning domain. Such theoretical insights can also be used as a means to guide development of theory-motivated new and better pruning algorithms on other neural network architectures like CNNs and RNNs in future work.

REFERENCES

[1] Madhu S Advani and Andrew M Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

[2] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*, 2019.

[3] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 224–232. JMLR. org, 2017.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[5] Mussay Ben, Margarita Osadchy, Vladimir Braverman, Samson Zhou, and Dan Feldman. Data-independent neural pruning via coresets. In *International Conference on Learning Representations (ICLR)*, 2020.

[6] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*, 2020.

[7] Li Deng and Yang Liu. *Deep learning in natural language processing*. Springer, 2018.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2018.

[10] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018.

[11] Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019.

[12] Elizabeth Gardner and Bernard Derrida. Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A: Mathematical and General*, 22(12):1983, 1989.

[13] Sebastian Goldt, Madhu Advani, Andrew M Saxe, Florent Krzakala, and Lenka Zdeborová. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. In *Advances in Neural Information Processing Systems*, pages 6979–6989, 2019.

[14] Scott Gray, Alec Radford, and Diederik P Kingma. Gpu kernels for block-sparse weights. *arXiv preprint arXiv:1711.09224*, 3, 2017.

[15] 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.

[16] Babak Hassibi and David G Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in neural information processing systems*, pages 164–171, 1993.

[17] Tianxing He, Yuchen Fan, Yanmin Qian, Tian Tan, and Kai Yu. Reshaping deep neural network for fast decoding by node-pruning. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 245–249. IEEE, 2014.

[18] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1389–1397, 2017.

[19] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: convergence and generalization in neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8580–8589, 2018.

[20] Kumar Joag-Dev, Frank Proschan, et al. Negative association of random variables with applications. *The Annals of Statistics*, 11(1):286–295, 1983.

[21] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. *arXiv preprint arXiv:1802.08435*, 2018.

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[24] Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

[25] Yann LeCun, Corinna Cortes, and Chris Burges. Mnist handwritten digit database. 2010. *URL http://yann. lecun. com/exdb/mnist*, 3(1), 2010.

[26] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990.

[27] Namhoon Lee, Thalaiyasingam Ajanthan, Stephen Gould, and Philip HS Torr. A signal propagation perspective for pruning neural networks at initialization. *arXiv preprint arXiv:1906.06307*, 2019.

[28] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2018.

[29] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[30] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[31] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.

[32] Odile Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1):83–122, 1975.

[33] Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning*, pages 6682–6691. PMLR, 2020.

[34] Zelda Mariet and Suvrit Sra. Diversity networks: Neural network compression using determinantal point processes. In *International Conference on Learning Representations*, 2016.

[35] Laurent Orseau, Marcus Hutter, and Omar Rivasplata. Logarithmic pruning is all you need. In *Proceedings of the 34nd International Conference on Neural Information Processing Systems*, 2020.

[36] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11893–11902, 2020.

[37] David Saad and Sara A Solla. Exact solution for on-line learning in multilayer neural networks. *Physical Review Letters*, 74(21):4337, 1995.

[38] David Saad and Sara A Solla. On-line learning in soft committee machines. *Physical Review E*, 52(4):4225, 1995.

[39] David Saad and Sara A Solla. Learning with noise and regularizers in multilayer neural networks. In *Advances in Neural Information Processing Systems*, pages 260–266, 1997.

[40] Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6655–6659. IEEE, 2013.

[41] Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

[42] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[43] Hidenori Tanaka, Daniel Kunin, Daniel LK Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Proceedings of the 34nd International Conference on Neural Information Processing Systems*, 2020.

[44] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

[45] Joost van Amersfoort, Milad Alizadeh, Sebastian Farquhar, Nicholas Lane, and Yarin Gal. Single shot structured pruning before training. *arXiv preprint arXiv:2007.00389*, 2020.

[46] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[47] 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.

[48] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 109(3):467–492, 2020.