ZEROTH-ORDER TOPOLOGICAL INSIGHTS INTO ITERATIVE MAGNITUDE PRUNING

Aishwarya Balwani & Jakob Krzyston
School of Electrical & Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
{abalwani6,jakobk}@gatech.edu

ABSTRACT

Modern-day neural networks are famously large, yet also highly redundant and compressible; there exist numerous pruning strategies in the deep learning literature that yield over 90% sparser sub-networks of fully-trained, dense architectures while still maintaining their original accuracies. Amongst these many methods though – thanks to its conceptual simplicity, ease of implementation, and efficacy – Iterative Magnitude Pruning (IMP) dominates in practice and is the de facto baseline to beat in the pruning community. However, theoretical explanations as to why a simplistic method such as IMP works at all are few and limited. In this work, we leverage persistent homology to show that IMP inherently encourages retention of those weights which preserve topological information in a trained network. Subsequently, we also provide bounds on how much different networks can be pruned while perfectly preserving their zeroth order topological features, and present a modified version of the IMP algorithm to do the same.

1 INTRODUCTION

The successes of deep neural networks (DNNs) across domains such as computer vision (Simonyan & Zisserman, 2014; He et al., 2016), speech recognition Graves et al. (2013), natural language processing (Vaswani et al., 2017; Brown et al., 2020), biomedicine (Ronneberger et al., 2015; Rajpurkar et al., 2017), and bioinformatics (Jumper et al., 2021) have made them ubiquitous in both academic and industrial settings. However, while DNNs boast of being able to achieve state of the art results on a plethora of complex problems, they also have the dubious honour of being untenably large and unsuitable for applications with power, memory, and latency constraints. One way of tackling these issues is to reduce the parameter-counts of DNNs, thereby decreasing their size and energy consumption while improving inference speeds. As a result, the field of *neural network pruning* which studies techniques for eliminating unnecessary weights in both pre-trained and randomly initialized DNNs without loss of accuracies (Mozer & Smolensky, 1988; Hanson & Pratt, 1988; LeCun et al., 1989; Hassibi & Stork, 1992) has seen renewed interest in recent years (Han et al., 2015; Li et al., 2016; Cheng et al., 2017; Frankle & Carbin, 2018; Lee et al., 2018; Blalock et al., 2020).

1.1 RELATED WORK

Increased activity on the methodological front has subsequently spurred principled analytical efforts to help explain when or how various pruning methods ostensibly work.

For instance, by way of deriving generalization bounds for DNNs via compression (Arora et al., 2018), previous work has provided some theoretical justification for pruned sub-networks. Iterative Magnitude Pruning (IMP) has been explored via the observation that sparse sub-networks that maintain accuracies of the original network are stable to stochastic gradient descent noise and optimize to linearly connected minima in the loss landscape (Frankle et al., 2020). A related empirical work has also looked into how fundamental phenomena such as weight evolution and emergence of distinctive connectivity patterns are affected by changes in the iterative pruning procedure (Paganini & Forde, 2020). A gradient-flow based framework (Lubana & Dick, 2020) has been used to show why certain importance measures work for pruning early on in the training cycle.

More recently, papers have also begun studying pruning at initialization, providing insights into gradient-based methods via conservation laws (Wang et al., 2020; Tanaka et al., 2020), presenting theoretical analyses of schemes that fall under the purview of sensitivity-based pruning (Hayou et al., 2020), developing a path-centric framework for studying pruning approaches (Gebhart et al., 2021), and looking at magnitude pruning in linear models trained using gradient flow (Elesedy et al., 2020).

Unfortunately, despite the flurry of contemporary work and results in the area at large, a mathematically rigorous yet intuitive explanation for *why* IMP works well remains missing.

1.2 Contributions

Given its integral place in the present DNN pruning research landscape, there is a strong impetus to establish a precise but flexible framework which uses the same language to speak of not only IMP, but also related problems of theoretical and empirical interest such as the Lottery Ticket Hypothesis (LTH) (Frankle & Carbin, 2018), DNN initialization and generalization (Morcos et al., 2019), weight rewinding, and fine-tuning (Renda et al., 2020). Towards this end we utilize the formalism of algebraic topology, which has found increasing application in the characterization of DNN properties such as learning capacity (Guss & Salakhutdinov, 2018), latent and activation space structure (Khrulkov & Oseledets, 2018; Gebhart et al., 2019; Carlsson & Gabrielsson, 2020), decision boundaries (Ramamurthy et al., 2019), and prediction confidence (Lacombe et al., 2021). Specifically, we use *neural persistence* (Rieck et al., 2018) – a measure based on persistent homology for assessing the topological complexity of neural networks – to ascertain which set of weights in the DNN capture its zeroth-dimensional topological features, and show that IMP with high probability preserves them. Following this result, the main contributions of our work are:

- A formal yet intuitive framework rooted in persistent homology that can reason about magnitude-based pruning and helps explain IMP's empirical success via theoretical lower bounds regarding its ability to preserve topological information in a trained DNN.
- Precise upper bounds on the maximum achievable compression ratios for fully-connected, convolutional, and recurrent layers such that they maintain their zeroth-dimensional topological features, and realizations of the same for some established architecture-dataset pairings in the pruning literature.
- A topologically-driven algorithm for iterative pruning, which would perfectly preserve their zeroth-order topological features throughout the pruning process.

2 Background & Notation

In this section we cover some necessary background on neural network pruning, persistent homology, and neural persistence, while also establishing the relevant notation.

2.1 Iterative Magnitude Pruning

A neural network *architecture* is a function family $f(x; \cdot)$, which consists of the configuration of the network's parameters and the sets of operations it uses to produce outputs from inputs, such as the arrangement of parameters into convolutions, activation functions, pooling, batch norm, etc. A *model* is a particular instantiation of an architecture, i.e., $f(x; \mathcal{W})$ with specific parameters $\mathcal{W}$.

Neural network *pruning* entails taking as input a model $f(x; \mathcal{W})$ and producing a new model $f(x; M \odot \mathcal{W}^*)$ where $M \in \{0, 1\}^{|\mathcal{W}^*|}$ is a binary mask that fixes certain parameters to 0, $\odot$ is the elementwise product operator, and $\mathcal{W}^*$ is a set of parameters that may differ from $\mathcal{W}$. A number of different heuristics called *scoring functions* may be used to construct the mask M, which decide which weights are to be pruned or kept. Popular scoring functions include the magnitude of the weights, or some form of the gradients of a specified loss with respect to the weights. If the mask M is constructed by scoring the parameters of a model per layer, the pruning scheme is said to be *local*, whereas if it is constructed by scoring all parameters in the set $\mathcal{W}$ collectively, the pruning scheme is said to be *global*.

The *sparsity* of the pruned model is $\frac{|f(x;M\odot\mathcal{W}^*)|_{\text{nnz}}}{|\mathcal{W}|}$ where $|\cdot|_{\text{nnz}}$ is a function that counts the number of non-zeros of the pruned model and $|\mathcal{W}|$ is the total number of parameters in the original model. The *compression ratio* (η) is given as $\frac{|\mathcal{W}|}{|f(x;M\odot\mathcal{W}^*)|_{\text{nnz}}}$ which is simply the inverse of the sparsity.

Given an initial untrained model $f(x, \mathcal{W}_0)$, *iterative magnitude pruning* (IMP) takes the following steps to obtain a sparsified model $f(x; \mathcal{W}_N)$ with $p\%$ target sparsity:

1. Train $f(x; \mathcal{W}_0)$ for t iterations, thereby obtaining the intermediate parameters $\mathcal{W}_{0,t}$
2. Mask $\frac{p}{N}\%$ non-zero parameters of lowest magnitude in $\mathcal{W}_{0,t}$ to arrive at parameters $\mathcal{W}_1$.
3. Repeat the aforementioned steps N times.

2.2 Persistent Homology

Persistent homology (Edelsbrunner et al., 2008) is a tool used in topological data analysis (TDA) to understand high-dimensional manifolds, and has been successfully employed in a range of applications such as analysing natural images (Carlsson et al., 2008), characterizing graphs (Sizemore et al., 2017; Rieck et al., 2017), and finding relevant features in unstructured data (Lum et al., 2013).

First, however, the space of interest must be represented as a *simplicial complex* which is effectively the extension of the idea of a graph to arbitrarily high dimensions. The sequence of *homology groups* (Edelsbrunner & Harer, 2022) of the simplicial complex then formalizes the notion of the topological features which correspond to the arbitrary dimensional "holes" in the space. For example, holes of dimension 0, 1, and 2 refer to connected components, tunnels, and voids respectively in the space of interest. Information from the d^{th} homology group is summarized by the d^{th} *Betti number* (β_d) which merely counts the number of d-dimensional holes; thus a circle has Betti numbers (1, 1), i.e., one connected component and one tunnel, while a disc has Betti numbers (1, 0), i.e., one connected component but no tunnel.

Betti numbers themselves unfortunately are of limited use in practical applications due to their instability and extremely coarse nature, which has prompted the development of persistent homology. Given a simplicial complex K with an additional set of scales $a_0 \leq a_1 \leq ... \leq a_{m-1} \leq a_m$, one can put K through a filtration, i.e., a nested sequence of simplicial complices $\emptyset = K_0 \subseteq K_1 \subseteq ...K_{m-1} \subseteq K_m = K$. The filtration essentially represents the growth of K as the scale is changed, and during this process topological features can be created (new vertices may be added, for example, which creates a new connected component) or destroyed (two connected components may merge into one).

Persistent homology tracks these changes, and represents the creation and destruction of a feature as a point $(a_i, a_j) \in \mathbb{R}^2$ for indices $i \leq j$ with respect to the filtration. The collection of all points corresponding to d-dimensional topological features is called the d^{th} *persistence diagram* ($\mathcal{D}_d$), and can be thought of as a collection of Betti numbers at multiple scales. Given a point $(x, y) \in \mathcal{D}_d$, the quantity $\text{pers}(x, y) := |y - x|$ is referred to as its *persistence*, where $|\cdot|$ is an appropriate metric. Typically, high persistence is considered to correspond to features, while low persistence is considered to indicate noise (Edelsbrunner et al., 2000).

2.3 Neural Persistence

Neural persistence is a recently proposed measure of structural complexity (Rieck et al., 2018) that exploits both network architecture and weight information through persistent homology, to capture how well trained a DNN is. For example, one can empirically verify that the "complexity" of a simple, fully connected network as measured by neural persistence increases with learning (Fig. 1).

Construction of the measure itself relies on the idea that one can view a model $f(x; \mathcal{W})$ as a stratified graph G with vertices V, edges E, with a mapping function $\varphi : E \rightarrow \mathcal{W}$ that allows for the calculation of the persistent homology of *every layer* G_k in the model using a filtration induced by sorting the weights. More precisely, given its set of weights $\mathcal{W}_k$ at any training step, let $w_{\max} := \max_{w\in\mathcal{W}_k} |w|$, and $\mathcal{W}'_k := \{|w|/w_{\max} \mid w \in \mathcal{W}\}$ be the set of transformed weights indexed in non-ascending order, such that $1 = w'_0 \geq w'_1 \geq ... \geq 0$. This permits one to define a filtration for the k^{th} layer G_k as $G_k^{(0)} \subseteq G_k^{(1)} \subseteq \ldots$, where $G_k^{(i)} :=$

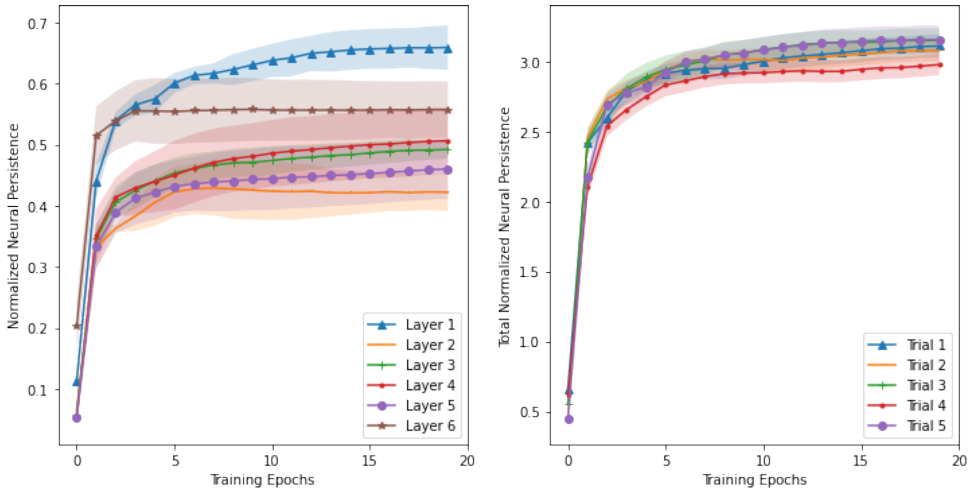

Figure 1: *Neural persistence for a fully-connected, 6-layer network trained on MNIST.* (Left) Layer-wise normalized neural persistences and (Right) Trial-wise total neural persistences for the entire network. Means are presented as solid lines, standard deviations are shaded.

$(V_k \sqcup V_{k+1}, \{(u,v) \mid (u,v) \in E_k \wedge \varphi'(u,v) \geq w'_i\})$, and $\varphi'_k(u,v) \in \mathcal{W}'_k$ denotes the transformed weight of an edge. The relative strength of a connection is thus preserved by the filtration, and weaker weights with $|w| \approx 0$ remain close to 0. Additionally, since $w' \in [0,1]$ for the transformed weights, the filtration makes the network invariant to scaling of $\mathcal{W}$, simplifying the comparison of different networks. Using this filtration one can calculate the persistent homology for every layer G_k. As the filtration contains at most 1-simplices (edges), the topological information captured is zero-dimensional, i.e. reflects how connected components are created and merged during the filtration, and can be shown graphically with a 0th persistence diagram (Fig. 2).

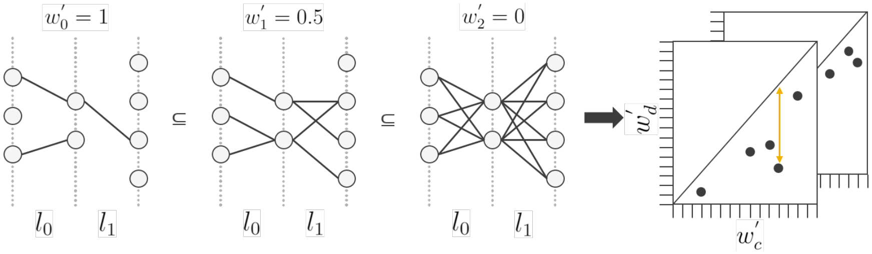

Figure 2: Persistence diagrams (PDs) show how long a particular model "persists" as the network undergoes the defined filtration (i.e, pruning). From left to right, as the weight threshold, w', decreases, the number of connections kept in the two layers (l_0 and l_1) increases. The PD shows the duration of each structure as a coordinate corresponding to the weight threshold at which the structure was created (w'_c), and the weight threshold at which it was destroyed (w'_d). The most prominent structure has the greatest *persistence*, which is measured by its distance from the diagonal. In this example, the point indicated by the yellow arrow would indicate the most persistent feature.

Neural persistence of the k^{th} layer G_k of a DNN is then defined as the p-norm of the persistence diagram $\mathcal{D}_k$ resulting from the previously discussed filtration, i.e.,

$$\text{NP}(G_k) := ||\mathcal{D}_k||_p := \left(\sum_{(c,d)\in\mathcal{D}_k} \text{pers}(c,d)^p \right)^{\frac{1}{p}}$$

Typically $p = 2$, which captures the Euclidean distance of the points in $\mathcal{D}_k$ to the diagonal.

We note that the above definition strictly corresponds to fully-connected layers, but the notion can easily be extended to convolutional and recurrent layers by representing the former using appropriately sized Toeplitz matrices (Goodfellow et al., 2016) and the latter as multiple "unrolled" fully-connected layers, with the same, shared set of weights.

Neural persistence can also be normalized to values in $[0, 1]$ in a scale-free manner, thus providing a simple method to compare the structural complexities of differently sized layers across various architectures. The total neural persistence of a model with L layers is given by the sum of all the individual layerwise neural persistences, i.e.,

$$\mathrm{NP}(G) := \sum_{k=1}^{L} \mathrm{NP}(G_k)$$

3 A Topological Perspective on Magnitude-Based Pruning

This section details our interpretation of magnitude pruning (MP) via the lens of neural persistence, followed by some insights we glean regarding IMP from this novel perspective. Consequently, we provide a lower bound on the relative topological information that is retained by IMP at every iteration, define a quantity that gives an upper bound on how much different types of neural network architectures may be pruned while still conserving its zeroth-dimensional topological features, and present a topologically-motivated version of IMP that guarantees the same.

3.1 Neural Persistence & Magnitude Pruning

Here we explicitly mention the key aspects of neural persistence which can be deduced from the background covered in Section 2 to arrive at a topological understanding of MP:

- Neural persistence relies on a super-level set filtration (Cohen-Steiner et al., 2009; Bubenik et al., 2015) and sorts only the edges of the bipartite graph G_k, the DNN layer it acts on.
- The weights $\mathcal{W}_k'$ of a layer G_k are normalized to values in $[0, 1]$, disregarding the signs of the weights $\mathcal{W}_k$ while still respecting their relative magnitudes.
- All the vertices of G_k are already present at the beginning of the filtration and result in $m_k + n_k$ connected components at the start of the filtration, where m_k, n_k are the cardinalities of the two vertex sets of G_k.
- Entries in the corresponding zeroth-dimensional persistence diagram $\mathcal{D}_k$ are of the form $(1, x), x \in \mathcal{W}_k'$, and are situated below the diagonal.
- As the filtration progresses, the weights greater than the threshold a_i are introduced in the zeroth-order persistence diagram, so long as it connects two vertices in G_k without creating any cycles (Lacombe et al., 2021).
- Subsequently, the filtration ends up with the maximum spanning tree[1] (MST) of G_k.

From the **viewpoint of persistent homology** this implies that all the zeroth-order topological information of G_k as captured by its neural persistence is encapsulated in the MST of $\mathcal{W}_k'$.

From the **standpoint of magnitude-based pruning**[2] we have a novel topologically-motivated scoring function, whose goal is to maintain the zeroth-order topological information in a set of weights, and the resulting mask M prunes any weights that are not part of the MST of a particular layer.

We therefore arrive at the following insight regarding IMP and its practical efficacy:

> *At every iteration, the weights retained by IMP in a layer are likely to overlap significantly with those present in its MST with relatively high probability. This ensures the pruning step itself does not severely degrade the zeroth-order topological information learnt by the layer in the previous training cycle, and in turn provides the network with a sufficiently informative initialization, allowing it train to high levels of accuracy once again.*

[1] For a proof, see Lemmas 1 & 2 in Doraiswamy et al.

[2] Since neural persistence by its current definition is applied layer-wise, the insights we gain from using it through the rest of this paper correspond to local pruning.

In the following subsections we formalize this intuition by presenting a lower bound on the expected overlap between the weights in a layer's MST and those retained by IMP, and upper bounds on how much a layer can be pruned whilst maintaining its zeroth-order topological information.

3.2 Topologically Critical Compression Ratio

Building off the observation that we only need as many weights as that of the MST of a layer G_k to maintain its zeroth-dimensional topological features, we define the following quantity that allows us to achieve maximal topologically conservative compression.

Definition 3.1. The topologically critical compression ratio (η_τ) for any graph G with edges E, vertices V, and mapping function $\varphi : E \to \mathcal{W}$ is defined as the quantity

$$\eta_\tau := \frac{|\mathcal{W}|}{|\operatorname{MST}(G)|}$$

where the function $\operatorname{MST}(\cdot)$ denotes the MST of the graph and $|\cdot|$ is the cardinality of a set.

η_τ is the maximal achievable compression for the graph G which would perfectly preserve its zeroth-order topological complexity, assuming the right set of weights (i.e., those in its MST) are retained.

Using Def. 3.1, in the context of DNN pruning we subsequently arrive at the following result

Theorem 3.2. *Given a layer G_k with weights $\mathcal{W}'_k$ joining m_k input nodes to n_k output nodes, for any compression ratio η that perfectly maintains the zeroth-order topological information of G_k, it holds that $\eta_\tau \geq \eta$.*

Furthermore,

- *When G_k is a fully connected layer*

$$\eta_\tau = \frac{m_k \cdot n_k}{m_k + n_k - 1}$$

- *When G_k is a recurrent layer with ℓ_k hidden units*

$$\eta_\tau = \frac{\ell_k^2}{2\ell_k - 1}$$

- *When G_k is a convolutional layer*

$$\eta_\tau = \frac{n_k \cdot f_1 \cdot f_2}{m_k + n_k - 1}$$

with (f_1, f_2) being size of the convolutional kernel. m_k, n_k are the input and output sizes respectively of the spatial activations.

3.3 Bounds on the MST – MP Fraction of Overlap

We now state a lower bound on the expected overlap in the MST of a layer G_k with its top-α weights, where α is the number of weights in its MST, to get a sense of how much of the zeroth-order topological information in a layer might be retained if we prune it down to its topologically critical compression ratio simply using the magnitude, thereby formally quantifying IMP's efficacy.

Theorem 3.3. *For a fully connected layer G_k with normalized weights $\mathcal{W}'_k$ joining m_k nodes at the input to n_k nodes at the output, for a compression ratio of η_τ, the fraction of overlap expected in its top-α weights by magnitude and those in its MST can be lower bounded as*

$$\mathbb{E}[X] \geq \frac{1}{m_k + n_k - 1} \cdot \sum_{i=0}^{j} \left(\frac{(m_k - i)(n_k - i)}{m_k \cdot n_k - i} \right)$$

where $j = \min(m_k, n_k) \geq 2$ and X is the fraction of overlap between the two quantities of interest.
If $j = 1, \mathbb{E}[X] = 1$

Corollary 3.4. *If a fully connected graph G_k is p-sparse, i.e., has a fraction of p non-zero weights $\geq \alpha$,*

$$\mathbb{E}[X] \geq \min\left\{1,\ \frac{1}{m_k + n_k - 1} \cdot \sum_{i=0}^{j} \left(\frac{(m_k - i)(n_k - i)}{p \cdot m_k \cdot n_k - i}\right)\right\}$$

where $j = \min(m_k, n_k) \geq 2$ and X is the fraction of overlap between the two quantities of interest. If $j = 1, \mathbb{E}[X] = 1$

We note here that these bounds (proofs for which can be found in the arXiv version of this paper at `https://arxiv.org/abs/2206.06563`) only give us a sense for how much of the topological complexity could be maintained; The exact values for the same would rely on the caluculation of the neural persistence and therefore depend on the exact distribution of weights $\mathcal{W}_k'$.

3.4 Topological Iterative Magnitude Pruning

The aforementioned insights and results thus naturally suggest a simple modification to the IMP algorithm that would ensure preservation of zeroth-order topological information in every layer. Following a similar structure as the IMP algorithm in Section 2.1, we now have Topological-IMP (T-IMP) that takes the following steps:

1. Find the weights which form the MST and retain them, accounting for α weights out of $\frac{p}{N}\%$ that one wishes to keep.
2. From the remaining $\frac{p}{N}\%$ - α weights, pick those with the highest magnitudes.
3. Retrain the network.
4. Repreat the process N times until the target sparsity-accuracy is reached.

4 Empirical Simulations & Results

4.1 Topologically Critical Compression

To see the practical significance of the topologically critical compression ratio and quantify the extent of pruning it can achieve, we experimented with combinations of popular datasets and architectures (Table 1). Our overall insights from these investigations are:

- Fully connected layers are a lot more redundant and ergo compressible than convolutional layers, and this seems to hold true across dataset-architecture pairings. The compressibility of these dense layers is what often seems to present incredibly high numbers for how compressible a particular model is.
- Amongst convolutional architectures, inherently more efficient architectures (e.g., ResNet) are less compressible than more redundant ones (e.g., VGG) even by topological metrics.

4.2 Bounds on the MST – MP Fraction of Overlap

The bound presented in Thm. 3.3 was also checked empirically with multiple simulations on different layers of the MNIST fully connected model (Fig. 3). While not the tightest, the bound (and simulations) still provided substantial support that MP does encourage preservation of zeroth-order topological features in the DNN weight space. However, preservation of zeroth-order topology only forms part of the story concerning which weights ought to be kept when pruning DNNs. Extensions to higher order homologies might perhaps help explain these discrepancies better.

5 Discussion & Future Work

In this work we presented a novel perspective on IMP leveraging a zeroth-order topological measure, viz., neural persistence. The resulting insights now provide us with the opportunity to pursue some

Table 1: Topologically Critical Compression: VGG11 & ResNet

	VGG11 (η_τ)	ResNet (η_τ)
CIFAR10*	Conv: 4.2295 Dense: 9.8273 Final: 4.3914	Conv: 4.1508 Dense: 8.7671 Final: 4.1679
CIFAR100*	Conv: 4.2295 Dense: 83.7971 Final: 6.9239	Conv: 4.1508 Dense: 39.2638 Final: 4.4389
Tiny-ImageNet†	Conv: 4.2672 Dense: 528.3911 Final: 183.3624	Conv: 4.3142 Dense: 144.0225 Final: 6.13182

* ResNet-20
† ResNet-18

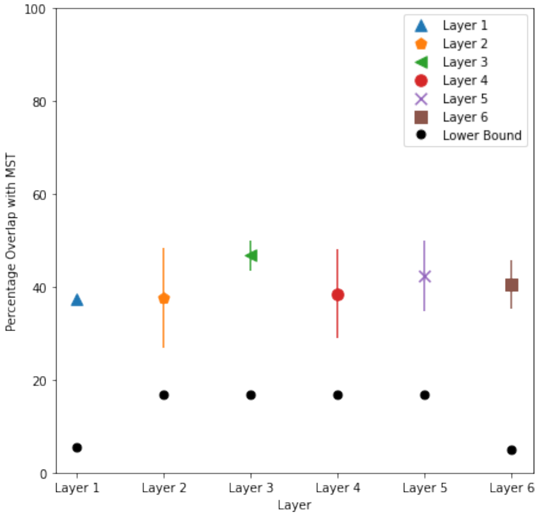

Figure 3: Top row: Mean percentage overlap (with Std Dev) the top-α weights have with the MST for different layers of a fully-connected network trained on MNIST. Black dots at the bottom represent the derived theoretical lower bound for the overlap for each of the respective layers.

exciting avenues on both, theoretical and empirical fronts. These include extensions of the stated bounds and subsequent theory to stratified graphs to explain global pruning, as well as the extension of NP to beyond zeroth-order homology which could help uncover a fuller picture of the topological complexities of DNNs.

Additionally, topological perspectives on LTH, weight rewinding, single-shot pruning, and other interesting IMP-adjacent phenomena could lead to not only insights into the interplay between DNN training and inference dynamics, but also topologically-motivated algorithms to achieve data- and compute-efficient deep learning pipelines.

Acknowledgments

The authors thank Georgia Tech's Graduate Student Association and College of Engineering for providing financial support to present this work and Nischita Kaza for helpful comments.

REFERENCES

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *International Conference on Machine Learning*, pp. 254–263. PMLR, 2018.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Peter Bubenik et al. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16(1):77–102, 2015.

Gunnar Carlsson and Rickard Brüel Gabrielsson. Topological approaches to deep learning. In *Topological data analysis*, pp. 119–146. Springer, 2020.

Gunnar Carlsson, Tigran Ishkhanov, Vin De Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *International journal of computer vision*, 76(1):1–12, 2008.

Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*, 2017.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Extending persistence using poincaré and lefschetz duality. *Foundations of Computational Mathematics*, 9(1):79–103, 2009.

Harish Doraiswamy, Julien Tierny, Paulo JS Silva, Luis Gustavo Nonato, and Claudio Silva. Topomap: A 0-dimensional homology preserving projection of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):561–571, 2020.

Herbert Edelsbrunner and John L Harer. *Computational topology: an introduction*. American Mathematical Society, 2022.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pp. 454–463. IEEE, 2000.

Herbert Edelsbrunner, John Harer, et al. Persistent homology-a survey. *Contemporary mathematics*, 453:257–282, 2008.

Bryn Elesedy, Varun Kanade, and Yee Whye Teh. Lottery tickets in linear models: An analysis of iterative magnitude pruning. *arXiv preprint arXiv:2007.08243*, 2020.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

Thomas Gebhart, Paul Schrater, and Alan Hylton. Characterizing the shape of activation space in deep neural networks. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1537–1542. IEEE, 2019.

Thomas Gebhart, Udit Saxena, and Paul Schrater. A unified paths perspective for pruning at initialization. *arXiv preprint arXiv:2101.10552*, 2021.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649. Ieee, 2013.

William H Guss and Ruslan Salakhutdinov. On characterizing the capacity of neural networks using algebraic topology. *arXiv preprint arXiv:1802.04443*, 2018.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Stephen Hanson and Lorien Pratt. Comparing biases for minimal network construction with back-propagation. *Advances in neural information processing systems*, 1, 1988.

Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in neural information processing systems*, 5, 1992.

Soufiane Hayou, Jean-Francois Ton, Arnaud Doucet, and Yee Whye Teh. Robust pruning at initialization. *arXiv preprint arXiv:2002.08797*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Valentin Khrulkov and Ivan Oseledets. Geometry score: A method for comparing generative adversarial networks. In *International Conference on Machine Learning*, pp. 2621–2629. PMLR, 2018.

Théo Lacombe, Yuichi Ike, Mathieu Carriere, Frédéric Chazal, Marc Glisse, and Yuhei Umeda. Topological uncertainty: Monitoring trained neural networks through persistence of activation graphs. *arXiv preprint arXiv:2105.04404*, 2021.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018.

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

Ekdeep Singh Lubana and Robert P Dick. A gradient flow framework for analyzing network pruning. *arXiv preprint arXiv:2009.11839*, 2020.

Pek Y Lum, Gurjeet Singh, Alan Lehman, Tigran Ishkanov, Mikael Vejdemo-Johansson, Muthu Alagappan, John Carlsson, and Gunnar Carlsson. Extracting insights from the shape of complex data using topology. *Scientific reports*, 3(1):1–8, 2013.

Ari Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. *Advances in neural information processing systems*, 32, 2019.

Michael C Mozer and Paul Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assessment. *Advances in neural information processing systems*, 1, 1988.

Michela Paganini and Jessica Forde. On iterative neural network pruning, reinitialization, and the similarity of masks. *arXiv preprint arXiv:2001.05050*, 2020.

Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

Karthikeyan Natesan Ramamurthy, Kush Varshney, and Krishnan Mody. Topological data analysis of decision boundaries with application to model selection. In *International Conference on Machine Learning*, pp. 5351–5360. PMLR, 2019.

Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. *arXiv preprint arXiv:2003.02389*, 2020.

Bastian Rieck, Ulderico Fugacci, Jonas Lukasczyk, and Heike Leitte. Clique community persistence: A topological visual analysis approach for complex networks. *IEEE transactions on visualization and computer graphics*, 24(1):822–831, 2017.

Bastian Rieck, Matteo Togninalli, Christian Bock, Michael Moor, Max Horn, Thomas Gumbsch, and Karsten Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. *arXiv preprint arXiv:1812.09764*, 2018.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

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

Ann Sizemore, Chad Giusti, and Danielle S Bassett. Classification of weighted networks through mesoscale homological features. *Journal of Complex Networks*, 5(2):245–273, 2017.

Hidenori Tanaka, Daniel Kunin, Daniel LK Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *arXiv preprint arXiv:2006.05467*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376*, 2020.