Is Complexity Required for Neural Network Pruning? A Case Study on Global Magnitude Pruning

Manas Gupta, Efe Camci, Vishandi Rudy Keneta, Abhishek Vaidyanathan, Ritwik Kanodia, Ashish James, Chuan-Sheng Foo, Min Wu, Jie Lin

Abstract—**Pruning neural networks has become popular in the last decade when it was shown that a large number of weights can be safely removed from modern neural networks without compromising accuracy. Numerous pruning methods have been proposed since, each claiming to be better than prior art, however, at the cost of increasingly complex pruning methodologies. These methodologies include utilizing importance scores, getting feedback through back-propagation or having heuristics-based pruning rules amongst others. In this work, we question whether this pattern of introducing complexity is really necessary to achieve better pruning results. We benchmark these SOTA techniques against a simple pruning baseline, namely, Global Magnitude Pruning (Global MP), that ranks weights in order of their magnitudes and prunes the smallest ones. Surprisingly, we find that vanilla Global MP performs very well against the SOTA techniques. When considering sparsity-accuracy trade-off, Global MP performs better than all SOTA techniques at all sparsity ratios. When considering FLOPs-accuracy trade-off, some SOTA techniques outperform Global MP at lower sparsity ratios, however, Global MP starts performing well at high sparsity ratios and performs very well at extremely high sparsity ratios. Moreover, we find that a common issue that many pruning algorithms run into at high sparsity rates, namely, layer-collapse, can be easily fixed in Global MP. We explore why layer collapse occurs in networks and how it can be mitigated in Global MP by utilizing a technique called Minimum Threshold. We showcase the above findings on various models (WRN-28-8, ResNet-32, ResNet-50, MobileNet-V1 and FastGRNN) and multiple datasets (CIFAR-10, ImageNet and HAR-2).**

Index Terms—**Neural Network Compression, Neural Network Pruning, Global Magnitude Pruning, Sparsity, Minimum Threshold**

I. Introduction

Scaling-up the size of neural networks is becoming a popular way to increase model performance [1]–[3]. However, this poses a significant cost to the environment [4] and makes deployment on edge devices difficult [5]. Neural network pruning has thus emerged as an essential tool to reduce the size of modern-day neural networks. New methods have utilized a myriad of pruning techniques consisting of gradient-based methods, sensitivity to or feedback from an objective function, distance or similarity measures, regularization-based techniques, amongst others. The state-of-the-art (SOTA) pruning techniques use complex rules like iterative pruning and re-growth of weight parameters using heuristics rules every few hundred iterations, as in DSR [6]. SM [7] uses sparse momentum that benefits from exponentially smoothed gradients (momentum) to find layers and weights that reduce error and then redistribute the pruned weights across layers using the mean momentum magnitude of each layer. For each layer, sparse momentum grows the weights using the momentum magnitude of zero-valued weights. Another popular SOTA technique, RigL [8], also iteratively prunes and re-grows weights every few iterations. They use uniform or Erdos-Renyi-Kernel (ERK) for pruning connections and re-grow connections based on the highest magnitude gradients. Among recent techniques, DPF [9] uses dynamic allocation of the sparsity pattern and incorporates a feedback signal to re-activate prematurely pruned weights, while STR [10] utilises soft threshold reparameterization and uses back-propagation to find sparsity ratios for each layer.

Despite the high number of new pruning algorithms proposed, the tangible benefits of many of them are still questionable. For instance, recently it has been shown that many pruning at initialization (PAI) schemes do not perform as well as expected [11]. In that paper, it is shown through a number of experiments that these PAI schemes are actually no better than random pruning, which is one of the most naive pruning baselines with no complexity involved. This finding indeed raises another question in our minds: if a well designed PAI does not even match the performance of random pruning, can simple pruning approaches like global pruning or their variants outperform other existing algorithms? We question the trend of proposing increasingly complex pruning algorithms and evaluate whether such complexity is really required to achieve superior results. We benchmark popular state-of-the-art (SOTA) pruning techniques against a naive pruning baseline, namely, Global Magnitude Pruning (Global MP). Global MP ranks all the weights in a neural network by their magnitudes and prunes off the smallest ones (Fig. 1).

Despite its simplicity, Global MP has not been comprehensively analyzed in the literature. Although, some prior works have used Global MP as a baseline [12]–[17], they missed out on conducting rigorous experiments with it, e.g., comparing it with SOTA, running it in both gradual and one-shot pruning

Manas Gupta, Efe Camci, Ashish James, Chuan-Sheng Foo, Min Wu and Jie Lin[1] are with the Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore. Chuan-Sheng Foo is also with the Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR), Singapore. Email: {manas_gupta, efe_camci, ashish_james, foo_chuan_sheng, wumin}@i2r.a-star.edu.sg. [1]Work done while at I^2R {jie.dellinger@gmail.com}. Vishandi Rudy Keneta is with the School of Computing (SoC), National University of Singapore (NUS), Singapore. Email: e0407662@u.nus.edu. Abhishek Vaidyanathan and Ritwik Kanodia[2] are with the School of Computer Science and Engineering (SCSE), Nanyang Technological University (NTU), Singapore. Email: abhishek033@e.ntu.edu.sg. [2]Work done while at NTU {ritwikkanodiain@gmail.com}.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00144

settings. Similarly, many SOTA papers do not use Global MP for benchmarking and miss out on capturing its remarkable performance [2], [8], [10], [18], [19]. We bridge this gap in evaluating the efficacy of Global MP and demonstrate its superior performance under multiple experimental conditions.

We show that with regards to the trade-off between sparsity and accuracy, Global MP consistently outperforms all state-of-the-art (SOTA) techniques across various sparsity ratios. In terms of the trade-off between FLOPs and accuracy, certain SOTA techniques exhibit superior performance than Global MP at lower sparsity ratios. However, Global MP demonstrates significant efficacy at higher sparsity ratios and excels particularly well at extremely high sparsity levels. While achieving such performance, Global MP does not require any additional algorithm-specific hyper-parameters to be tuned. We also shed light into a potential problem with pruning, known as layer-collapse, whereby an entire layer is pruned away, leading to a drastic loss in accuracy. The fix for it in Global MP is simple through introducing a Minimum Threshold (MT) to retain a minimum number of weights in every layer. We conduct experiments on WRN-28-8, ResNet-32, ResNet-50, MobileNet-V1, and FastGRNN models, and on CIFAR-10, ImageNet, and HAR-2 datasets. We test Global MP for both unstructured and structured as well as one-shot and gradual settings, and share our findings.

II. Related Work

Compression of neural networks has become an important research area due to the rapid increase in size of neural networks [20], the need for fast inference [21], application to real-world tasks [22]–[24] and concerns about the carbon footprint of training large neural networks [4]. Over the years, several compression techniques have emerged in the literature [25], such as quantisation, factorisation, attention, knowledge distillation, architecture search and pruning [26]–[29]. As compared to other categories, pruning is more general in nature and has shown strong performance [2].

Many pruning techniques have been developed over the years, which use first or second order derivatives [1], [30], [31], gradient-based methods [32]–[34], sensitivity to or feedback from some objective function [9], [35]–[38], distance or similarity measures [39], Bayesian optimisation [40], regularization-based techniques [10], [41]–[44], and magnitude-based criterion [8], [17], [18], [45], [46]. A key trick has been discovered in [47] to iteratively prune and retrain a network, thereby preserving high accuracy. Runtime Neural Pruning [48] attempts to use reinforcement learning (RL) for compression by training an RL agent to select smaller sub-networks during inference. [49] design the first approach using RL for pruning. However, RL training approaches typically require additional RL training budgets and careful RL action and state space design [50], [51].

Global MP on the other hand ranks all the parameters in a network by their absolute magnitudes and prunes the smallest ones. It is therefore, quite intuitive, logical and straightforward. It is also not to be confused with methods that utilize global pruning but do not conduct magnitude pruning, for example, SNIP [32]. These methods use complex criteria, first to determine the saliency of the weights globally and then apply pruning. We present here an in-depth comparison of the SOTA techniques vs. Global MP. Gradual Magnitude Pruning (GMP) [18] uses a uniform pruning schedule and prunes each layer by the same amount, thereby, not taking into account the relative importance of layers. Global MP on the other hand prunes every layer differently. Dynamic Sparse Reparameterization (DSR) [6] prunes and regrows weights every few hundred iterations. It also uses Global MP to prune the weights. However, it imposes some additional heuristic-based constraints on the pruning process, such as, not pruning some selected layers in the network. In case the weights to regrow outnumber the capacity of a layer, it then uses additional heuristics to redistribute the additional weights. These kinds of heuristics add both complexity and limit the potential pruning actions that can be taken. Discovering Neural Wirings (DNW) [19] focuses on learning connectivity of channels in a network. It is not primarily a pruning technique and is more akin to Neural Architecture Search (NAS).

Sparse Momentum (SM) [7] uses a heuristic-based approach to prune and regrow weights. They use average momentum to assess importance of every layer and assign parameters accordingly. Similar to DSR, a certain number of layers are never pruned and in cases of regrowth exceeding capacity of a layer, additional heuristics are used to redistribute weights. This method is also computationally demanding as gradients need to be stored in memory and additional FLOPs are required to calculate mean momentum per parameter. Hence, it is not as efficient as Global MP, which is computationally less expensive and does not rely on heuristics. Rigging the Lottery (RigL) [8] allocates sparsity based on number of parameters in a layer. Hence, importance of a layer is based on its size, which is not necessarily an accurate measure for all cases. Dynamic Pruning with Feedback (DPF) [9] also uses Global MP for pruning. However, it imposes an additional constraint of keeping the last layer fully dense. It is thus, not as flexible as pure Global MP that allows all layers to be pruned.

Soft Threshold Reparameterization (STR) [10] is a regularization-based technique that subtracts a certain value from the weights in each epoch. The exact sparsity target cannot be controlled in STR, and it requires heavy tuning of hyper-parameters to reach a required sparsity target. Hence, it is not as flexible as Global MP. Matrix-Free Approximations of Second-Order Information (M-FAC) [52] and its pre-cursor (WoodFisher) [53] use an approximation of the second order Hessian to prune weights. Hessian-based pruning is principled but computationally intractable. Therefore, approximations need to be made for the Hessian. These approximations result in not-so-accurate importance scores for the weights, while such approximations are not needed in Global MP. Thus, Global MP is generally more principled, heuristic-free, and computationally inexpensive compared to SOTA methods which might be the underlying reason for its superior performance.

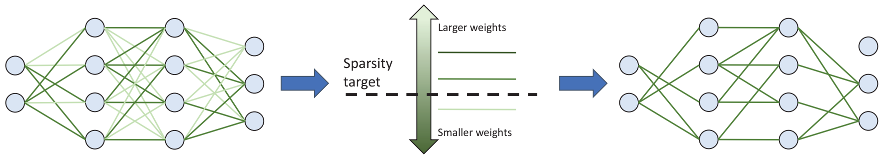

Fig. 1: Illustration of how Global MP works. Global MP ranks all the weights in a network by their magnitudes and prunes off the smallest weights until the target sparsity is met. Light green weights refer to the smaller-magnitude weights which are pruned off. A pruned network consisting of larger-magnitude weights (dark green weights) is obtained after the process.

III. Method

In this section, we explain how Global MP works by describing its key components. We also introduce a simple thresholding mechanism, called *Minimum Threshold (MT)*, to avoid the issue of layer-collapse at high sparsity levels.

A. Global Magnitude Pruning (Global MP)

Global MP is a magnitude-based pruning approach, whereby weights larger than a certain threshold are kept, and weights smaller than the threshold are pruned across a neural network. The threshold is calculated based on the target sparsity rate and is not a hyper-parameter that needs to be tuned or learnt. Given a target sparsity rate κ_{target}, the threshold t is simply calculated as the weight magnitude that serves as a separation point between the smallest κ_{target} percent of weights and the rest, once all weights are sorted into an array based on their magnitude. Formally, for a calculated threshold t and each individual weight w in any layer, the new weight w_{new} is defined as follows:

$$w_{new} = \begin{cases} 0 & |w| < t, \\ w & otherwise. \end{cases} \tag{1}$$

In Global MP, a single threshold is set for the entire network based on the target sparsity for the network. This is in contrast to layer-wise pruning, in which different threshold values have to be searched for each layer individually. In the case of uniform pruning on the other hand, a threshold for each layer needs to be calculated based on the sparsity target assigned to the layers uniformly across the network. In this aspect, Global MP is more efficient than layer-wise or uniform pruning because the threshold does not need to be searched or calculated for every layer individually.

B. Minimum Threshold (MT)

The Minimum Threshold (MT) refers to the fixed number of weights that are preserved in every layer of the neural network post pruning. The MT is a scalar value that is fixed before the start of the pruning cycle. The weights in a layer are sorted by their magnitude and the largest MT number of weights are preserved. For instance, an MT of 500 implies that 500 of the largest weights in every layer need to be preserved post pruning. If a layer originally has a smaller number of weights than the MT number, then all the weights of that layer will be preserved. This corresponds to:

$$\|W_l\|_0 \geq \begin{cases} \sigma & \text{if } m \geq \sigma_l, \\ m & \text{otherwise.} \end{cases} \tag{2}$$

The term $W_l \in \mathbb{R}^m$ denotes the weight vector for layer l, σ is the MT value in terms of the number of weights and $\|W_l\|_0$ indicates the number of non-zero elements in W_l.

C. The Pruning Workflow

The pruning pipeline for Global MP consists of pruning a model until the desired sparsity target is met and training or fine-tuning it for a specified number of epochs. It supports both one-shot and gradual pruning settings as well as with or without MT. The users may choose any pruning setting as per their use case. The procedure starts by selecting a pre-trained model in one-shot pruning, or an untrained model in gradual pruning. Next, the sparsity of the model is checked and if the sparsity is lower than the target sparsity, then the model is pruned using either vanilla Global MP or Global MP with MT, as per the choice of the user. Once, the model is pruned, it is trained for the case of gradual pruning or fine-tuned for the case of one-shot pruning. The Global MP framework allows the flexibility for previously pruned weights to regrow, if they become more active in the later epochs, for the case of gradual pruning. No hard pruning is done whereby weights are permanently zeroed out. The pruning mask is calculated afresh in each epoch thereby allowing previously pruned weights to regrow. The above procedure repeats until the final epoch is reached. For the case of one-shot pruning, the later epochs are just used for doing fine-tuning as the pruning happens in one-go in the first epoch itself. This finishes the procedure and the final result is a pruned and trained (or fine-tuned) model. See Appendix for pseudocode.

IV. Experiments

Below we describe experiments related to Global MP compared to state-of-the-art (SOTA) pruning algorithms.

A. Comparison with SOTA

We compare Global MP with various popular SOTA algorithms that are well known for pruning, such as SNIP [32], SM [7], DSR [6], DPF [9], GMP [18], DNW [19], RigL [8], and STR [10]. These include a broad spectrum of methods involving iteratively pruning and re-growing weights every few iterations, pruning at initialization, using gradients and feedback signals for pruning, and pruning using regularization.

Method	Sparsity	WRN-28-8 Acc.	ResNet-32 Acc.
Baseline	0.0%	96.06%	93.83 ± 0.12 %
SNIP [32]	90%	95.49 ± 0.21%	90.40 ± 0.26%
SM [7]	90%	95.67 ± 0.14%	91.54 ± 0.18%
DSR [6]	90%	95.81 ± 0.10%	91.41 ± 0.23%
DPF [9]	90%	96.08 ± 0.15%	92.42 ± 0.18%
Global MP	90%	**96.30 ± 0.03%**	**92.67 ± 0.03%**
SNIP [32]	95%	94.93 ± 0.13%	87.23 ± 0.29%
SM [7]	95%	95.64 ± 0.07%	88.68 ± 0.22%
DSR [6]	95%	95.55 ± 0.12%	84.12 ± 0.32%
DPF [9]	95%	95.98 ± 0.10%	**90.94 ± 0.35%**
Global MP	95%	**96.16 ± 0.02%**	90.65 ± 0.13%

TABLE I: Results of SOTA pruning algorithms on WideResNet-28-8 and ResNet-32 on CIFAR-10. The bold font denotes best performance. Global MP outperforms or yields comparable performance to other algorithms.

We report results from these algorithms whenever they report results for the specific dataset that is being experimented upon. See Appendix for hyper-parameters.

1) CIFAR-10: We conduct experiments to compare Global MP to SOTA pruning algorithms on the CIFAR-10 dataset. We compare One-shot Global MP with four algorithms in this case: SNIP [32], SM [7], DSR [6], and DPF [9]. Results for comparison algorithms are taken from [9] which conduct standardized testing on algorithms. We report results on two popular and widely pruned network architectures, WideResNet-28-8 (WRN-28-8) and ResNet-32 [54]. For both architectures, we start off with the original model having the same initial accuracy as the other algorithms to have a fair comparison. For WRN-28-8 (Table I), Global MP performs better than the rest of the competitors at both 90% and 95% sparsity levels. Global MP outperforms DSR and SM in all cases, because of their additional heuristics-based constraints which limit the selection of layers to be pruned. As for ResNet-32 (Table I), Global MP outperforms at 95% sparsity and is the second best at 90% sparsity. Global MP outperforms DSR and SM for all sparsity levels in this case as well. This is an indication of the capabilities of Global MP as compared to the other algorithms, while featuring no added complexity.

2) ImageNet: Following the favorable performance on CIFAR-10 dataset, we benchmark Global MP on ImageNet dataset. This is a highly challenging dataset as compared to CIFAR-10, featuring around 1.3 million RGB images with 1,000 classes. We compare Global MP with SOTA algorithms like GMP [18], DSR [6], DNW [19], SM [7], RigL [8], WoodFisher [53], MFAC [52], DPF [9], and STR [10]. Results for comparison algorithms are taken from [10] which conduct standardized testing on algorithms. The two network architectures we use for this comparison are ResNet-50 and MobileNet-V1 [55], the two most popular architectures for benchmarking pruning algorithms on ImageNet [14]. For ResNet-50, we include an additional experimental setting of gradual Global MP to provide more thorough comparison with SOTA methods. We again start from the same initial accuracy for the non-pruned models for all algorithms, either by matching the results in their original papers or reproducing their results whenever their code is available. We sample four sparsity levels ranging from low sparsity (80%) to extreme sparsity (98%) to provide a comprehensive snapshot across different sparsity levels.

The remarkable performance of Global MP becomes clear in ResNet-50 over ImageNet experiments. For the sparsity-accuracy trade-off, Global MP outperforms all the other competitors or achieves comparable accuracy in every sparsity level from 80% to 98% (see results in bold in Table II). MFAC performs closely for 95% sparsity, however, it is not a like-for-like comparison as their sparsity is slightly lower (95%) compared to Global MP (95.3%). We take the upper bound sparsity target for each sparsity level for Global MP to match the method with the highest reported sparsity in that sparsity level. For the case of extreme sparsity (98%), Global MP surpasses the second best algorithm (STR) by a large margin of 5.11%. For the FLOPs-accuracy trade-off, a like-for-like comparison is difficult to make because the methods report different FLOPs targets. However, an experienced practitioner can roughly gauge the efficacy of the methods based on the ratio between the additional FLOPs pruned and the decrease in accuracy of the methods. Based on this we find that certain SOTA techniques exhibit superior FLOPs-accuracy performance than Global MP at lower sparsity ratios of 80% and 90%. However, Global MP becomes competitive at 95% sparsity and performs very well at the extreme sparsity rate of 98%, gaining 5.11% accuracy vs. a drop of 2% FLOPs vis-a-vis the second best method (STR).

We also find that gradual Global MP performs better than one-shot Global MP at high and extremely high sparsity ratios. This is because the pruning mask is allowed to change multiple times in gradual Global MP compared to only once in one-shot Global MP, and hence, converges to a more optimized value. Most SOTA methods also follow the same approach whereby they allow the pruning mask to change each epoch or sometimes multiple times in an epoch. Overall, Global MP outperforms all SOTA algorithms on sparsity-accuracy trade-off and comes in second, after STR, for FLOPs-accuracy trade-off. It is an important finding that such a simple algorithm like Global MP can outperform other SOTA competitors that incorporate very complex design choices or computationally demanding procedures.

We also test another architecture on ImageNet, MobileNet-V1, which is a much smaller and more efficient architecture than ResNet-50. In this case, strong competitors are limited in the literature; only two of the aforementioned algorithms are able to present competitive results due to the fact that this architecture has less redundancy. We benchmark Global MP with two other competitors at two target sparsity levels: 75% and 90%. As can be seen in Table III, Global MP outperforms SOTA algorithms on the sparsity-accuracy trade-off by a margin of more than 2% at 75% sparsity, which is a significant result given how compact MobileNet-V1 is. At 90% sparsity on the other hand, the same compactness causes Global MP

Method	Top-1 Acc	Params	Sparsity	FLOPs pruned
ResNet-50	77.0%	25.6M	0.00%	0.0%
GMP [18]	75.60%	5.12M	80.00%	80.0%
DSR*# [6]	71.60%	5.12M	80.00%	69.9%
DNW [19]	76.00%	5.12M	80.00%	80.0%
SM [7]	74.90%	5.12M	80.00%	-
SM + ERK [7]	75.20%	5.12M	80.00%	58.9%
RigL* [8]	74.60%	5.12M	80.00%	77.5%
RigL + ERK [8]	75.10%	5.12M	80.00%	58.9%
DPF [9]	75.13%	5.12M	80.00%	80.0%
STR [10]	76.19%	5.22M	79.55%	81.3%
Global MP (One-shot)	**76.84%**	5.12M	**80.00%**	72.4%
Global MP (Gradual)	76.12%	5.12M	80.00%	76.7%
GMP [18]	73.91%	2.56M	90.00%	90.0%
DNW [19]	74.00%	2.56M	90.00%	90.0%
SM [7]	72.90%	2.56M	90.00%	60.1%
SM + ERK [7]	72.90%	2.56M	90.00%	76.5%
RigL* [8]	72.00%	2.56M	90.00%	87.4%
RigL + ERK [8]	73.00%	2.56M	90.00%	76.5%
DPF# [9]	74.55%	4.45M	82.60%	90.0%
STR [10]	74.73%	3.14M	87.70%	90.2%
Global MP (One-shot)	**75.28%**	2.56M	**90.00%**	82.8%
Global MP (Gradual)	74.83%	2.56M	90.00%	87.8%
GMP [18]	70.59%	1.28M	95.00%	95.0%
DNW [19]	68.30%	1.28M	95.00%	95.0%
RigL* [8]	67.50%	1.28M	95.00%	92.2%
RigL + ERK [8]	70.00%	1.28M	95.00%	85.3%
WoodFisher [53]	72.12%	1.28M	95.00%	-
MFAC MFAC [52]	**72.32%**	1.28M	**95.00%**	-
STR [10]	70.40%	1.27M	95.03%	96.1%
Global MP (One-shot)	71.56%	1.20M	95.30%	89.3%
Global MP (Gradual)	**72.14%**	1.20M	**95.30%**	93.1%
GMP [18]	57.90%	0.51M	98.00%	98.0%
DNW [19]	58.20%	0.51M	98.00%	98.0%
STR [10]	61.46%	0.50M	98.05%	98.2%
Global MP (One-shot)	61.80%	0.50M	98.05%	93.7%
Global MP (Gradual)	**66.57**%	0.50M	**98.05%**	96.2%

TABLE II: Results on ResNet-50 on ImageNet. Global MP outperforms SOTA pruning algorithms at all sparsity levels for the sparsity-accuracy trade-off and at high sparsity levels for the FLOPs-accuracy trade-off. Bold font denotes the best performance for the sparsity-accuracy trade-off while underlined font denotes the best performance for FLOPs-accuracy trade-off. * and # imply the first and the last layer are dense, respectively.

Method	Top-1 Acc	Params.	Sparsity	FLOPs pruned
MobileNet-V1	71.95%	4.21M	0.00%	0.0%
GMP [18]	67.70%	1.09M	74.11%	71.4%
STR [10]	68.35%	1.04M	75.28%	82.2%
Global MP	**70.74%**	1.04M	**75.28%**	68.9%
GMP [18]	61.80%	0.46M	89.03%	85.6%
STR [10]	61.51%	0.44M	89.62%	93.0%
Global MP	59.49%	0.42M	90.00%	83.7%
Global MP with MT	**63.94%**	0.42M	**90.00%**	72.9%

TABLE III: Results of pruning algorithms on MobileNet-V1 on ImageNet. The bold font denotes the algorithm with the best sparsity-accuracy performance while underlining denotes the best FLOPs-accuracy performance. Global MP with MT surpasses SOTA algorithms on sparsity-accuracy performance.

to over-prune certain layers in the network, which result in a significant accuracy drop. This is the above-mentioned problem of layer-collapse, and it is easily rectified when MT is introduced to Global MP. We use an MT value of 0.2% which is determined using the same grid-search procedure as any other hyper-parameter. Usually values between 0.01% to 0.3% work well for MT regardless of models and datasets. See Appendix for an ablation study on MT. We find that MT behaves like a typical hyper-parameter. On increasing the MT value initially, the accuracy increases, until it reaches a maximum value. Thereafter, increasing the MT value leads to a decrease in accuracy. Thus, a suitable value can be found by doing a search over MT values. The accuracy of Global MP at 90% sparsity goes beyond SOTA again with such a simple fix, and the accuracy margin to the next competitor gets higher than 2%. For the FLOPs-accuracy trade-off, a like-for-like comparison is again hard to make, but STR seems to perform better, owing to the smaller sparsities that MobileNet is pruned at, as compared to ResNet-50. MT also comes at the cost of a less FLOPs reduction, but it is useful especially for accuracy-critical applications where decreasing the size of the network is still important. All these findings clearly indicate that Global MP is a simple yet competitive pruning algorithm. It delivers top performance on sparsity-accuracy trade-off, and ranks second for FLOPs-accuracy trade-off, despite not having any complex design choices or additional hyper-parameters.

B. Structured pruning and generalizing to other domains and RNN architectures

We experiment with Global MP on other domains and non-convolutional networks as well to measure the generalizability of the algorithm on different domains and network types. We experiment on a FastGRNN model [56] on the HAR-2 Human Activity Recognition dataset [57]. HAR-2 dataset is a binarized version of the 6-class Human Activity Recognition dataset. From the full-rank model with $r_W = 9$ and $r_U = 80$ as suggested on the STR paper [10], we apply Global MP on the matrices W_1 and W_2. To do this, we find the weight mask by ranking the columns of W_1 and W_2 based on their absolute sum, then we prune the $9 - r_W^{new}$ lowest columns and $80 - r_U^{new}$ lowest columns from W_1 and W_2 respectively. In the end, we fine-tune this pruned model by retraining it with FastGRNN's trainer and applying the weight mask at every epoch. We test Global MP under different r_w-r_v configurations. We find that Global MP surpasses the other baselines on all the configurations (Table IV) and successfully prunes the model on a very different architecture and domain.

C. *Mitigating layer-collapse*

Layer-collapse is an issue that many pruning algorithms run into [15], [58], [59] and occurs when an entire layer is pruned, rendering the network untrainable. We investigate this phenomena and find that performance of a pruning algorithm gets substantially affected by the neural network architecture being pruned, especially in the high sparsity domain. We conduct experiments on MobileNet-V2 and WRN-22-8 models over the CIFAR-10 dataset. We report results averaged over multiple runs where each run uses a different pre-trained model to provide more robustness. We first prune a WRN-22-8 model to 99.9% sparsity. We find that at 99.9% sparsity, the WRN is still able to get decent accuracy (Table V). We then prune a MobileNet-V2 model to 98% sparsity. For MobileNet, however, accuracy drops to 10% using only Global MP, and the model is not able to learn (Table VI).

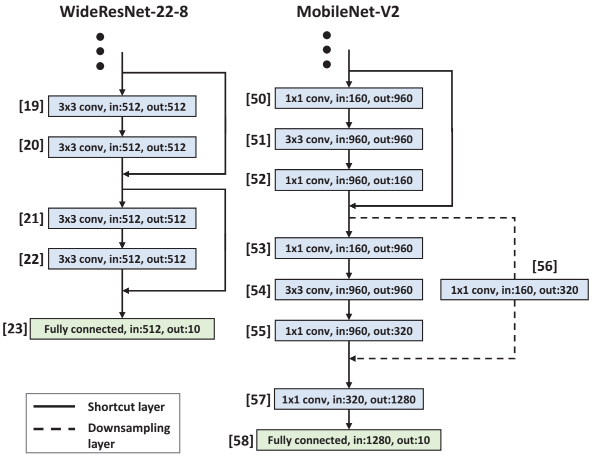
WideResNet-22-8
MobileNet-V2
[19] 3x3 conv, in:512, out:512
[20] 3x3 conv, in:512, out:512
[21] 3x3 conv, in:512, out:512
[22] 3x3 conv, in:512, out:512
[23] Fully connected, in:512, out:10
[50] 1x1 conv, in:160, out:960
[51] 3x3 conv, in:960, out:960
[52] 1x1 conv, in:960, out:160
[53] 1x1 conv, in:160, out:960
[54] 3x3 conv, in:960, out:960
[55] 1x1 conv, in:960, out:320
[56] 1x1 conv, in:160, out:320
[57] 1x1 conv, in:320, out:1280
[58] Fully connected, in:1280, out:10
Shortcut layer
Downsampling layer

Fig. 2: Difference in architectures between WRN and MobileNet. WRN does not have prunable residual connections in the last layers (dotted lines) while MobileNet does. This leads to different pruning behaviors on the two architectures.

The reason for this wide discrepancy in learning behavior lies in the shortcut connections [54]. Both WRN-22-8 and MobileNet-V2 use shortcut connections, however, their placement is different. Referring to Fig. 2, WRN uses identity shortcut connections from Layer 20 to Layer 23. This type of shortcut connections are simple identity mappings and do not require any extra parameters, and hence, they do not count towards the weights. However, MobileNet-V2 uses a convolutional shortcut mapping from Layer 52 to Layer 57. The weights in this mapping are counted towards the model's weights, and thus, they are prunable. Global MP completely prunes the two preceding layers before the last layer. However, because WRN uses identity mappings, it is still able to relay information to the last layer, and the model is still able to learn, whereas MobileNet-V2 faces catastrophic accuracy drop due to layer-collapse. Pruning algorithms can be susceptible to such catastrophic layer-collapse issues especially in the high sparsity domain. The MT rule can help overcome this issue. Retaining a small MT of 0.02% is sufficient for MobileNet-V2 to avoid layer-collapse and learn successfully. Hence, retaining a small amount of weights can help in the learning dynamics of models in high sparsity settings.

Method	Top-1 Acc	r_W	r_U
FastGRNN	96.10%	9	80
Vanilla Training	94.06%	9	8
STR [10]	95.76%	9	8
Global MP	**95.89**%	9	8
Vanilla Training	93.15%	9	7
STR [10]	95.62%	9	7
Global MP	**95.72**%	9	7
Vanilla Training	94.88%	8	7
STR [10]	95.59%	8	7
Global MP	**95.62**%	8	7

TABLE IV: Results on FastGRNN on HAR-2 dataset. The bold font denotes the algorithm with the best performance. Global MP outperforms other pruning algorithms.

Method	WRN-22-8 on CIFAR-10		
	Sparsity	Starting Acc.	Pruned Acc.
Global MP	99.9%	94.07% ± 0.05%	**67.68% ± 0.78%**

TABLE V: Performance of Global MP on WideResNet-22-8 in the high sparsity regime at 99.9% sparsity.

Method	MobileNet-V2 on CIFAR-10		
	Sparsity	Starting Acc.	Pruned Acc.
Global MP	98.0%	94.15% ± 0.23%	10% *(Unable to learn)*
Global MP with MT	98.0%	94.15% ± 0.23%	**82.97% ± 0.57%**

TABLE VI: Adding MT enables MobileNet-V2 to learn in the high sparsity regime.

V. Discussion, Limitations and Future Work

Our observations indicate that Global MP works very well and achieves superior performance on all datasets and architectures tested. It can work as a one-shot or as a gradual pruning algorithm. We test it on challenging datasets like ImageNet that require a high number of parameters to achieve good results and test its pruning efficacy on them. It also surpasses SOTA algorithms on ResNet-50 over ImageNet on the sparsity-accuracy trade-off and sets new SOTA results across many sparsity levels. For FLOPs-accuracy trade-off, it comes in second after STR, surpassing many SOTA techniques. At the same time, Global MP has very low algorithmic complexity and arguably is one of the simplest pruning algorithms. It is simpler than many other pruning algorithms like custom loss based regularization, RL-based procedures, heuristics-based layerwise pruning ratios, etc. It just ranks weights on their magnitude and removes the smallest ones. This raises a key question on whether complexity is really required for pruning and according to our results it seems that complexity in itself does not guarantee good performance. Practitioners developing new pruning algorithms should thus look carefully whether complexity is adding value to their algorithm or not. They should also benchmark their algorithms against simple baselines like Global MP.

A limitation of Global MP is that the theoretical foundations for it have not been well-established yet. Following our empirical work, we plan to conduct a theoretical analysis for better comprehending the dynamics of Global MP in the future. It would include finding analytical links between the magnitude of weights and their importance in a network, or even analytical relations of them to the resultant accuracy of the model. Another area for future work is jointly optimizing both weights and FLOPs during the pruning process. Currently, Global MP is used to reach a certain parameter sparsity, and FLOPs reduction comes as a by-product. In the future, FLOPs can also be added to an optimization function to jointly sparsify both parameters and FLOPs.

VI. Conclusions

In this work, we raised the question of whether utilizing complex and computationally demanding algorithms are really required to achieve superior DNN pruning results. This stemmed from the hike in the number of new pruning algorithms proposed in the recent years, each with a marginal performance increment, but increasingly complicated pruning procedures. This makes it hard for a practitioner to select the correct algorithm and the best set of algorithm-specific hyper-parameters for their application. We benchmarked these algorithms against a naive baseline, namely, Global MP, which does not incorporate any complex procedure or any hard-to-tune hyper-parameter. Despite its simplicity, we found that Global MP outperforms many SOTA pruning algorithms over multiple datasets, such as CIFAR-10, ImageNet, and HAR-2; with different network architectures, such as ResNet-50 and MobileNet-V1; and at various sparsity levels from 50% up to 99.9%. We also presented a few variants of Global MP, i.e., one-shot and gradual, together with a new, complementary technique, MT. While our results serves as an empirical proof that a naive pruning algorithm like Global MP can achieve SOTA results, it remains as a promising future research direction to shed light into theoretical aspects of how such performance is possible with Global MP. Another future direction includes extending the capabilities of Global MP, such as jointly optimizing both FLOPs and the number of weights.

References

[1] E. Kurtic, D. Campos, T. Nguyen, E. Frantar, M. Kurtz, B. Fineran, M. Goin, and D. Alistarh, "The optimal BERT surgeon: Scalable and accurate second-order pruning for large language models," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 4163–4181. [Online]. Available: https://aclanthology.org/2022.emnlp-main.279

[2] T. Gale, E. Elsen, and S. Hooker, "The state of sparsity in deep neural networks," *arXiv preprint arXiv:1902.09574*, 2019.

[3] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *CoRR*, vol. abs/2001.08361, 2020. [Online]. Available: https://arxiv.org/abs/2001.08361

[4] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[5] R. Bonatti, W. Wang, C. Ho, A. Ahuja, M. Gschwindt, E. Camci, E. Kayacan, S. Choudhury, and S. Scherer, "Autonomous aerial cinematography in unstructured environments with learned artistic decision-making," *Journal of Field Robotics*, vol. 37, no. 4, pp. 606–641, 2020.

[6] H. Mostafa and X. Wang, "Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019, pp. 4646–4655.

[7] T. Dettmers and L. Zettlemoyer, "Sparse networks from scratch: Faster training without losing performance," *CoRR*, vol. abs/1907.04840, 2019. [Online]. Available: http://arxiv.org/abs/1907.04840

[8] U. Evci, T. Gale, J. Menick, P. S. Castro, and E. Elsen, "Rigging the lottery: Making all tickets winners," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 2943–2952. [Online]. Available: https://proceedings.mlr.press/v119/evci20a.html

[9] T. Lin, S. U. Stich, L. Barba, D. Dmitriev, and M. Jaggi, "Dynamic model pruning with feedback," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SJem8lSFwB

[10] A. Kusupati, V. Ramanujan, R. Somani, M. Wortsman, P. Jain, S. Kakade, and A. Farhadi, "Soft threshold weight reparameterization for learnable sparsity," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 5544–5555. [Online]. Available: http://proceedings.mlr.press/v119/kusupati20a.html

[11] J. Frankle, G. K. Dziugaite, D. Roy, and M. Carbin, "Pruning neural networks at initialization: Why are we missing the mark?" in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=Ig-VyQc-MLK

[12] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," *arXiv preprint arXiv:1803.03635*, 2018.

[13] A. Morcos, H. Yu, M. Paganini, and Y. Tian, "One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/a4613e8d72a61b3b69b32d040f89ad81-Paper.pdf

[14] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" *arXiv preprint arXiv:2003.03033*, 2020.

[15] H. Tanaka, D. Kunin, D. L. Yamins, and S. Ganguli, "Pruning neural networks without any data by iteratively conserving synaptic flow," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6377–6389. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/46a4378f835dc8040c8057beb6a2da52-Paper.pdf

[16] A. Renda, J. Frankle, and M. Carbin, "Comparing rewinding and fine-tuning in neural network pruning," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=S1gSj0NKvB

[17] J. Lee, S. Park, S. Mo, S. Ahn, and J. Shin, "Layer-adaptive sparsity for the magnitude-based pruning," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=H6ATjJ0TKdf

[18] M. Zhu and S. Gupta, "To prune, or not to prune: exploring the efficacy of pruning for model compression," *ICLR Workshop*, vol. abs/1710.01878, 2018.

[19] M. Wortsman, A. Farhadi, and M. Rastegari, "Discovering neural wirings," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates,

Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/d010396ca8abf6ead8cacc2c2f2f26c7-Paper.pdf

[20] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, 2020.

[21] E. Camci, D. Campolo, and E. Kayacan, "Deep reinforcement learning for motion planning of quadrotors using raw depth images," in *2020 international joint conference on neural networks (IJCNN)*. IEEE, 2020, pp. 1–7.

[22] C. Liu, P. Liu, W. Zhao, and X. Tang, "Visual tracking by structurally optimizing pre-trained cnn," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 3153–3166, 2020.

[23] Y. Peng and J. Qi, "Quintuple-media joint correlation learning with deep compression and regularization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2709–2722, 2020.

[24] K. Liu, W. Liu, H. Ma, M. Tan, and C. Gan, "A real-time action representation with temporal encoding and deep compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, pp. 647–660, 2021.

[25] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," 2017.

[26] A. Almahairi, N. Ballas, T. Cooijmans, Y. Zheng, H. Larochelle, and A. Courville, "Dynamic capacity networks," in *International Conference on Machine Learning*, 2016, pp. 2549–2558.

[27] A. Ashok, N. Rhinehart, F. Beainy, and K. M. Kitani, "N2n learning: Network to network compression via policy gradient reinforcement learning," 2017.

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

[29] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," 2018.

[30] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. Morgan-Kaufmann, 1990, pp. 598–605. [Online]. Available: http://papers.nips.cc/paper/250-optimal-brain-damage.pdf

[31] B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," in *Advances in Neural Information Processing Systems 5*, S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds. Morgan-Kaufmann, 1993, pp. 164–171.

[32] N. Lee, T. Ajanthan, and P. H. Torr, "Snip: Single-shot network pruning based on connection sensitivity," *arXiv preprint arXiv:1810.02340*, 2018.

[33] C. Wang, G. Zhang, and R. Grosse, "Picking winning tickets before training by preserving gradient flow," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SkgsACVKPH

[34] A. Peste, E. Iofinova, A. Vladu, and D. Alistarh, "AC/DC: Alternating compressed/decompressed training of deep neural networks," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=T3_AJr9-R5g

[35] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," *International Conference on Learning Representations*, 2017.

[36] J. Liu, Z. XU, R. SHI, R. C. C. Cheung, and H. K. So, "Dynamic sparse training: Find efficient sparse network from scratch with trainable masked layers," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SJlbGJrtDB

[37] P. de Jorge, A. Sanyal, H. Behl, P. Torr, G. Rogez, and P. K. Dokania, "Progressive skeletonization: Trimming more fat from a network at initialization," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=9GsFOUyUPi

[38] J. Guo, W. Zhang, W. Ouyang, and D. Xu, "Model compression using progressive channel pruning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 1114–1124, 2021.

[39] S. Srinivas and R. V. Babu, "Data-free parameter pruning for deep neural networks," *Procedings of the British Machine Vision Conference 2015*, 2015. [Online]. Available: http://dx.doi.org/10.5244/C.29.31

[40] T. Kim, H. Choi, and Y. Choe, "Automated filter pruning based on high-dimensional bayesian optimization," *IEEE Access*, vol. 10, pp. 22 547–22 555, 2022.

[41] X. Ding, G. Ding, J. Han, and S. Tang, "Auto-balanced filter pruning for efficient convolutional neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/12262

[42] P. Savarese, H. Silva, and M. Maire, "Winning the lottery with continuous sparsification," *Advances in Neural Information Processing Systems*, 2020.

[43] H. Wang, C. Qin, Y. Zhang, and Y. Fu, "Neural pruning via growing regularization," in *International Conference on Learning Representations*, 2021.

[44] M. Zhao, J. Peng, S. Yu, L. Liu, and N. Wu, "Exploring structural sparsity in cnn via selective penalty," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1658–1666, 2022.

[45] N. Ström, "Sparse connection and pruning in large dynamic artificial neural networks," 1997.

[46] S. Park, J. Lee, S. Mo, and J. Shin, "Lookahead: a far-sighted alternative of magnitude-based pruning," *International Conference on Learning Representations*, 2020.

[47] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in neural information processing systems*, 2015, pp. 1135–1143.

[48] J. Lin, Y. Rao, J. Lu, and J. Zhou, "Runtime neural pruning," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 2181–2191. [Online]. Available: http://papers.nips.cc/paper/6813-runtime-neural-pruning.pdf

[49] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "Amc: Automl for model compression and acceleration on mobile devices," in *European Conference on Computer Vision (ECCV)*, 2018.

[50] M. Gupta, S. Aravindan, A. Kalisz, V. Chandrasekhar, and L. Jie, "Learning to prune deep neural networks via reinforcement learning," *International Conference on Machine Learning (ICML) AutoML Workshop*, 2020.

[51] E. Camci, M. Gupta, M. Wu, and J. Lin, "Qlp: Deep q-learning for pruning deep neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2022.

[52] E. Frantar, E. Kurtic, and D. Alistarh, "M-FAC: Efficient matrix-free approximations of second-order information," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=EEq6YUrDyfO

[53] S. P. Singh and D. Alistarh, "Woodfisher: Efficient second-order approximation for neural network compression," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 18 098–18 109. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/d1ff1ec86b62cd5f3903ff19c3a326b2-Paper.pdf

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[55] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[56] A. Kusupati, M. Singh, K. Bhatia, A. J. S. Kumar, P. Jain, and M. Varma, "Fastgrnn: A fast, accurate, stable and tiny kilobyte sized gated recurrent neural network," in *NeurIPS*, 2018.

[57] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium*, 2013.

[58] N. Lee, T. Ajanthan, S. Gould, and P. H. S. Torr, "A signal propagation perspective for pruning neural networks at initialization," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=HJeTo2VFwH

[59] S. Hayou, J.-F. Ton, A. Doucet, and Y. W. Teh, "Robust pruning at initialization," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=vXj_ucZQ4hA