HOLISTIC ADVERSARIALLY ROBUST PRUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks can be drastically shrinked in size by removing redundant parameters. While crucial for the deployment on resource-constraint hardware, oftentimes, compression comes with a severe drop in accuracy and lack of adversarial robustness. Despite recent advances, counteracting both aspects has only succeeded for moderate compression rates so far. We propose a novel method, HARP, that copes with aggressive pruning significantly better than prior work. For this, we consider the network holistically, learning a global compression strategy that, however, can be specific to each layer. Our method optimizes both how many parameters (compression rate) and which parameters (scoring connections) to prune. It fine-tunes an existing model with dynamic regularization, that follows a step-wise incremental function balancing the different objectives. It starts by favoring robustness before shifting focus on reaching the target compression rate and only then handles the objectives equally. The learned compression strategies allow us to maintain the pre-trained model's natural accuracy as well as its adversarially robustness for a reduction by $99\,\%$ of the network's original size. Moreover, we observe a crucial influence of non-uniform compression across layers.

1 INTRODUCTION

Deep neural networks (DNNs) yield remarkable performances in classification tasks in various domains (He et al., 2016; Cakir & Dogdu, 2018; Schroff et al., 2015; Li et al., 2022) but are vulnerable to input manipulation attacks such as adversarial examples (Szegedy et al., 2014). Small perturbations to benign inputs can cause worst-case errors in prediction.To date, the most promising defensive approach against this sort of attacks is adversarial training as introduced by Madry et al. (2018) and further refined ever since (Shafahi et al., 2019; Zhang et al., 2019; Wang et al., 2020). It introduces adversarial examples into the training process, diminishing the generalization gap between natural performance and adversarial robustness. However, there is evidence indicating that higher robustness requires over-parameterized networks that have wider layers and higher structural complexity (Madry et al., 2018; Zhang et al., 2019; Wu et al., 2021), rendering the task of combining both objectives—compression and robustness—inherently difficult.

Neural network pruning (Han et al., 2016; Yang et al., 2017; He et al., 2017), for instance, has been proven to be an extraordinary value tool for compressing neural networks. The model can be reduced to a fraction of its size by removing redundancy at different structural granularity (Li et al., 2017; Mao et al., 2017; Kollek et al., 2021). However, pruning inflicts a certain recession in model accuracy (Han et al., 2016) as well as adversarial robustness (Timpl et al., 2021) that is unavoidable the stronger the model in compressed. The aim of adversarial robust pruning hence is to maintain the accuracy and robustness of an already adversarially pre-trained model as good as possible. Despite great efforts in the recent past (Ye et al., 2019; Sehwag et al., 2020; Madaan et al., 2020; Özdenizci & Legenstein, 2021; Lee et al., 2022), this dual objective has only succeeded for moderate compression rates so far.

In this paper, we start from the hypothesis that effective adversarially robust pruning requires a non-uniform compression strategy with learnable pruning masks, and propose our method HARP. We follow the three-stage pruning pipeline proposed by Han et al. (2015) to improve upon pre-trained models, where we jointly optimize score-based pruning masks and layer-wise compression rates during fine-tuning. As high robustness challenges the compactness objective (Timpl et al., 2021), we employ a step-wise increasing weighted-control of the number of weights to be pruned, such that we can learn masks and rates simultaneously. Our approach explores a global pruning strategy that allows for on-par natural accuracy and less robustness degradation.

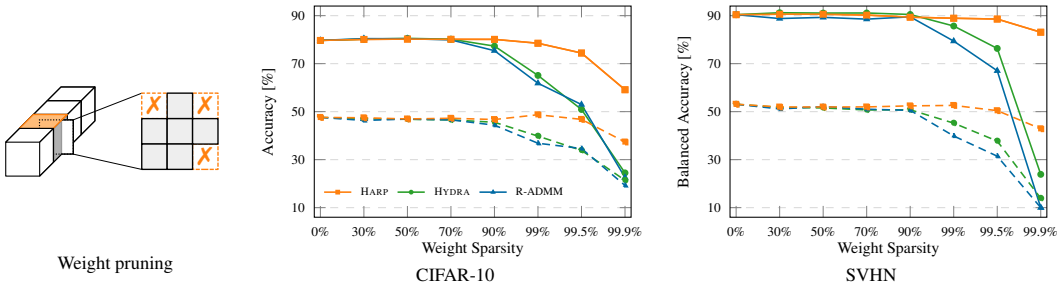


Figure 1: Overview of pruning weights of a VGG16 model for CIFAR-10 (left) and SVHN (right). Solid lines show the natural accuracy of HARP (orange), HYDRA (green), and Robust-ADMM (blue). Dashed lines represent the methods' adversarial robustness.

Contributions and impact. We conduct extensive experiments with small and large-scale datasets (CIFAR-10, SVHN, and ImageNet) to demonstrate our method's capabilities in compressing neural networks. HARP significantly outperforms related work in pruning weights from an adversarially pre-trained model across training methods, such as PGD-AT (Madry et al., 2018), TRADES-AT (Zhang et al., 2019), and MART (Wang et al., 2020), as well as popular attacks including PGD (Madry et al., 2018), C&W$_\infty$ (Carlini & Wagner, 2017), AUTOPGD, and AUTOATTACK (Croce & Hein, 2020a). An overview of our experiments with a PGD-AT trained VGG16 on CIFAR-10 and SVHN is presented in Fig. 1. Our method yields non-uniform pruning strategies, while popular prior work remain uniform. We thus compare HARP to novel extensions of HYDRA (Sehwag et al., 2020) and Robust-ADMM (Ye et al., 2019) by combining them with ERK (Evci et al., 2020) and LAMP (Lee et al., 2021) that determine non-uniform compression strategies that can be used for pruning. Moreover, we analyze the parameter distribution of the resulting pruned models, conduct ablation studies, and demonstrated an extension to channel pruning.

2 NEURAL NETWORK PRUNING

Removing structural redundant parameters enables compressing a network's overall memory footprint and reduces necessary computations, allowing for an demand-oriented adaptation of neural networks to resource-constrained environments (Han et al., 2015; 2016; Wen et al., 2016; Huang et al., 2018; He et al., 2018; 2017; Li et al., 2017; Mao et al., 2017; Molchanov et al., 2017). Neural network pruning attempts to find a binary mask $\boldsymbol{M}^{(l)}$ for each layer l of a network with L layers represented by its parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^{(l)}$, respectively. The overall pruning-mask thus is denoted as $\boldsymbol{M} = (\boldsymbol{M}^{(1)}, \ldots, \boldsymbol{M}^{(l)}, \ldots, \boldsymbol{M}^{(L)})$. These masks specify which parameters of the layer $\boldsymbol{\theta}^{(l)}$ to remove (zero out) and which to keep, yielding a reduced parameter set $\hat{\boldsymbol{\theta}}^{(l)} = \boldsymbol{\theta}^{(l)} \odot \boldsymbol{M}^{(l)}$, where $\odot$ is the Hadamard product. Based on this, we define the overall compression rate a as the ratio parameters preserved after pruning, $\Theta_{\neq 0}$, to the number of parameters in the model, Θ. Compression rates for individual layers $a^{(l)}$ are defined analogously. Note, that a network's *sparsity* is defined inversely, meaning, a $99.9\,\%$ sparsity refers to a compression of 0.001. In further follow, we consider $\boldsymbol{\theta}^{(l)} \in \mathbb{R}^{c_i^{(l)} \times c_o^{(l)} \times k^{(l)} \times k^{(l)}}$ where $c_i^{(l)}$ and $c_o^{(l)}$ represent input and output channels, respectively, and $k^{(l)}$ is the kernel size.

Han et al. (2015) propose a three stage pipeline for network pruning, starting with (1) training an over-parameterized network, followed by (2) removing redundancy per layer according to some pruning criterion, before (3) recovering network performance via fine-tuning. The actual pruning strategy, that decides on the pruning mask $\boldsymbol{M}$, is obtained in the second step. While also integrated approaches (pruning during model training) exist (e.g., Vemparala et al., 2021; Özdenizci & Legenstein, 2021), the staged process remains most common (Liu et al., 2019; Sehwag et al., 2020; Lee et al., 2022) as it allows to benefit from recent advantages in adversarial training (Shafahi et al., 2019; Zhang et al., 2019; Wang et al., 2020) out-of-the-box.

3 HOLISTIC ADVERSARIALLY ROBUST PRUNING

Two aspects are crucial when pruning neural networks: *How many and which parameters to prune*. Current state-of-the-art pruning methods primarily focus on the latter, allocate a compression budget per layer upfront and then either increase model sparsity during training (Ye et al., 2019) or searching for better structural connections (Sehwag et al., 2020) given this budget. We argue that also learning the optimal (layer-wise) pruning amount—the compression rate—is equally crucial. HARP thus integrates measures for compression rates and scoring connections in an empirical min-max optimization problem based on adversarial training to learn a suitable pruning mask $\boldsymbol{M}$ for a pre-trained model $\boldsymbol{\theta}$:

$$\min_{\boldsymbol{r},\boldsymbol{S}} \quad \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\delta} \left\{ \mathcal{L}_{robust}(\boldsymbol{\theta} \odot \boldsymbol{M}, \boldsymbol{x}+\delta, y) \right\} \right] + \gamma \cdot \mathcal{L}_{hw}(\boldsymbol{\theta} \odot \boldsymbol{M}, a_t), \tag{1}$$

The inner maximization generates adversarial examples from a benign input $\boldsymbol{x}$ with label y from a distribution $\mathcal{D} = \{\boldsymbol{x}, y\}$ by adding noise δ, and incorporates these when minimizing the training loss $\mathcal{L}_{ce}$ (Madry et al., 2018). The exactly formulation of $\mathcal{L}_{robust}$ depends on the pre-trained model, for which we evaluate PGD-AT (Madry et al., 2018), TRADES-AT (Zhang et al., 2019), and MART-AT (Wang et al., 2020). Simultaneously, the model is subject to a pruning mask that involve two parameters: First, the compression quota $\boldsymbol{r}$, which is a learnable representation of compression rates, and second, scores $\boldsymbol{S}$ for determining the importance of the network's connections. In the following, we discuss HARP's global compression control and describe how pruning strategies are learned, before we elaborate on how to dynamically regularize the training process using γ by following a step-wise incremental function balancing the different objectives.

Global compression control. The compression control is encoded as an additional loss term $\mathcal{L}_{hw}$, that considers compression holistically for the entire network to reach a specific target compression a_t. However, it explicitly allows for layer-specific rates, yielding an overall non-uniform compression strategy. In particular, we measure the number of currently preserved (i.e., non-zero) weights over the total number of parameters $\boldsymbol{\theta}^{(l)}$, summed up across all L layers of the network and put in relation to the target compression rate a_t:

$$\mathcal{L}_{hw} = \max \left\{ \frac{1}{a_t} \cdot \sum_{l=1}^{L} \frac{\left\| \left(\mathbb{1}_{w \neq 0} \right)_{w \in \hat{\boldsymbol{\theta}}^{(l)}} \right\|_1}{\Theta^{(l)}} - 1 \ , \ 0 \right\} \tag{2}$$

By constraining the loss to positive values and, thus, the rates that would exceed the targeted compression-rate, we ensure to not encourage even lower compression rates that would potentially harm accuracy and robustness unnecessarily.

Moreover, for learning the layer-specific compression rates, it is crucial that we constraint the reduction by a minimal compression a_{min} as we need to prevent a layer from been removed completely. We thus limit the layers' compression $a^{(l)}$ to $[a_{min}, 1]$. In order to better control learning of the optimal pruning rate in this range, we introduce a trainable parameter $r^{(l)} \in \mathbb{R}$, which we refer to as *compression quota*, and constrain it by an activation function $g : \mathbb{R} \to [a_{min}, 1]$. For the sake of the continuous derivability on r, we are using the sigmoid function squeezed to the desired output range, yielding a layer-wise compression rate $a^{(l)} = g(r^{(l)})$, where $g : r \mapsto (1 - a_{min}) \cdot \mathrm{sig}(r) + a_{min}$

Connection importance. For learning the actual pruning mask, in turn, we introduce another learnable parameter $\boldsymbol{S}$, the connection importance-score matrix, to rate each pruning connection. In weight pruning, $\boldsymbol{S}^{(l)}$ originates $\mathbb{R}^{c_i^{(l)} \times c_o^{(l)} \times k^{(l)} \times k^{(l)}}$, and is assigned values from activation function g. Additionally, we use $P(\alpha^{(l)}, \boldsymbol{S}^{(l)})$ to represent the function that determines the α-th percentile on layer l as the cutoff threshold for pruning parameters in accordance with compression rate $a^{(l)}$, that is: $\alpha^{(l)} = 1 - a^{(l)}$. Due to the relation with $\boldsymbol{S}^{(l)}$ and the indirect relation with $\boldsymbol{r}^{(l)}$ via compression rate $a^{(l)}$, the binary pruning mask $\boldsymbol{M}^{(l)}$ is thus trainable and defined as:

$$\boldsymbol{M}^{(l)} = \left(\mathbb{1}_{s > P(\alpha^{(l)}, \boldsymbol{S}^{(l)})} \right) \tag{3}$$

Pruning the l-th layer can hence be expressed by the Hadamard product of the mask and the model's parameters $\boldsymbol{\theta}$ yielding the pruned model parameters $\hat{\boldsymbol{\theta}}^{(l)} = \boldsymbol{\theta}^{(l)} \odot \boldsymbol{M}^{(l)}$.

Learning Pruning Strategies. Binarizing importance scores to obtain a pruning mask as described above is a non-differentiable operation and prevents us from performing gradient descent over the masks. We, thus, follow the "straight forward estimation" (STE) strategy (Hubara et al., 2016) and assign the updated gradients to the importance scores $\boldsymbol{S}^{(l)}$ directly and proceed similarly for compression quotas $r^{(l)}$:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{S}^{(l)}} = \frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{\theta}}^{(l)}} \cdot \frac{\partial \hat{\boldsymbol{\theta}}^{(l)}}{\partial \boldsymbol{M}^{(l)}} \cdot \frac{\partial \boldsymbol{M}^{(l)}}{\partial \boldsymbol{S}^{(l)}} \overset{\text{STE!}}{=} \frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{\theta}}^{(l)}} \cdot \frac{\partial \hat{\boldsymbol{\theta}}^{(l)}}{\partial \boldsymbol{M}^{(l)}} \tag{4}$$

$$\frac{\partial \mathcal{L}}{\partial r^{(l)}} = \frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{\theta}}^{(l)}} \cdot \frac{\partial \hat{\boldsymbol{\theta}}^{(l)}}{\partial \boldsymbol{M}^{(l)}} \cdot \frac{\partial \boldsymbol{M}}{\partial g(r^{(l)})} \cdot g'(r^{(l)}) \overset{\text{STE!}}{=} \langle \frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{\theta}}^{(l)}} \cdot \frac{\partial \hat{\boldsymbol{\theta}}^{(l)}}{\partial \boldsymbol{M}^{(l)}} \rangle \cdot g'(r^{(l)}) \tag{5}$$

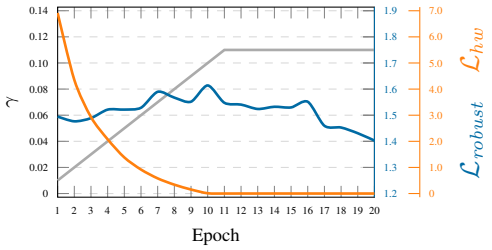
Finally, also the initialization of the connection importance-scores is crucial. We follow the lead of Sehwag et al. (2020) and initialize the scores proportional to the weights of the pre-trained model and use a scaling factor $\eta = \sqrt{k/\textit{fan-in}^{(l)}}$ with $k = 6$ and *fan-in* being the product of the receptive field size and the number of input channels. Our initialization for pruning weights, hence, follows the order of weight magnitudes (Ye et al., 2019):

$$\boldsymbol{S}_W^{(l)} = \left(\eta \cdot \frac{\boldsymbol{\theta}^{(l)}}{\max(|\boldsymbol{\theta}^{(l)}|)} \right)_{c_i^{(l)} \times c_o^{(l)} \times k^{(l)} \times k^{(l)}} \tag{6}$$

HARP does not use a fixed ratio for pruning parameters, but learns the optimal compression rate per layer. According to the analysis of strategy initialization in Appendix A.2, for initializing the compression quota $\boldsymbol{r}$, we first align with a uniform compression strategy that sets the same rate a_{init} on each layer and let the optimizer improve it. For the activation function $g(\cdot)$ defined above, we thus yield:

$$r^{(l)} = \log \left(\frac{a_{init} - a_{min}}{1 - a_{init}} \right) \tag{7}$$

Balancing pruning objectives. HARP strives to strike a balance between natural accuracy, adversarial robustness and compactness. For this, we employ a step-wise incremental function to adapt the regularization parameter γ throughout the learning process. We start by favoring robustness before shifting focus on reaching the target compression rate and only then handle the objectives equally. The figure on the right shows the relation of γ and the two different losses, $\mathcal{L}_{robust}$ against PGD-10 and $\mathcal{L}_{hw}$, on the y-axis over the epochs on the x-axis, exemplarily for weight pruning VGG16 (learned on CIFAR-10) with a target compression $a_t = 0.01$, that is, removing 99 % of the parameters. The model starts off from the uniform strategy with $a_{init} = 0.1$, learning to be more robust while being only slightly penalized by $\mathcal{L}_{hw}$. As the model has been adversarially pre-trained, $\mathcal{L}_{robust}$ is low already. Over time, we put more focus on $\mathcal{L}_{hw}$ up until we have reached our target compression ($\gamma = 0.11$ in this example), causing it to decrease while $\mathcal{L}_{robust}$ increases. Afterwards, the model aims for higher robustness under the found $\mathcal{L}_{hw}$ penalty, balancing both objectives. We elaborate on the influence on the model's natural accuracy and adversarial robustness in Appendix A.2 (cf. Fig. 6).



4 EVALUATION

We resume to demonstrate HARP's effectivity in pruning neural networks by comparing to existing state-of-the-art methods in Section 4.1. In Section 4.2, we then perform an ablation study regarding the learnable parameters used by our method. In particular, we show the influence of learning compression quotas $\boldsymbol{r}$ and importance scores $\boldsymbol{S}$, separately, to demonstrate that both are indeed needed for successful, holistic pruning. In Section 4.3 and Appendix A.2, we then discuss the non-uniform nature of HARP's pruning strategy and compare to novel extensions of HYDRA and Robust-ADMM to non-uniform compression. In the appendix, we additionally elaborate on the pruned

model's parameter distribution (Appendix A.3), extend HARP to channel pruning (Appendix A.4), and show HARP's performance on naturally trained models (Appendix A.5).

Experimental Setup. We evaluate HARP on two small-scale datasets, CIFAR-10 (Krizhevsky et al.) and SVHN (Netzer et al., 2011). While the first is balanced, the second is not. Consequently, we use accuracy (ACC) as performance measure for the former, and the balanced accuracy (BACC) for the latter (Brodersen et al., 2010). Each dataset is learned with a VGG16 (Simonyan & Zisserman, 2015) and ResNet18 (He et al., 2016) model. Additionally, we show the performance on the large-scale, balanced ImageNet dataset (Deng et al., 2009) trained with a ResNet50 network. We apply $\gamma = 0.01$ for small-scale datasets, and increase γ to 0.1 for ImageNet to guarantee the arrival at target compression rate a_t. For training the models, we employ adversarial training with different losses: PGD (Madry et al., 2018), TRADES (Zhang et al., 2019), and MART (Wang et al., 2020) for the small-scale datasets and FREE-AT (Shafahi et al., 2019) with 4 replays for the large-scale dataset. For validating the pruned networks' adversarial robustness, we use PGD (Madry et al., 2018), C&W$_\infty$ (Carlini & Wagner, 2017), AUTOPGD and AUTOATTACK (Croce & Hein, 2020a). Appendix A.1 provides further details on the experimental setup.

4.1 ROBUST PRUNING WITH HARP

In this section, we report results of our comparative evaluation with Robust-ADMM (Ye et al., 2019), HYDRA (Sehwag et al., 2020), BCS-P (Özdenizci & Legenstein, 2021), and MAD (Lee et al., 2022) in pruning a neural network's weights. In particular, for aggressive compression yielding high sparsity, HARP excels and outperforms prior work. For the comparison to Robust-ADMM and HYDRA, we perform a systematic evaluation with CIFAR-10, SVHN, and ImageNet. For BCS-P and MAD, we revert to the respective settings reported in the original papers.

Comparison to Robust-ADMM. The method by Ye et al. (2019) is designed to robustly prune neural networks on fine-grained (weights) and coarse-grained granularity (channels/filters). We focus on pruning weights at this point and present further results on pruning channels in Appendix A.4. Tables 1 to 3 summarize the results for pruning models on CIFAR-10, SVHN, and ImageNet, respectively. Robust-ADMM performs well for pruning a network's weights to a sparsity of $99\,\%$ for

Table 1: Comparing HARP with Robust-ADMM and HYDRA on models learned on CIFAR-10.

	Adversarial Training	Attack	99 % Sparsity			99.9 % Sparsity		
			R-ADMM	HYDRA	HARP	R-ADMM	HYDRA	HARP
ResNet18	PGD	–	71.42±0.37	75.53±1.95	**80.13**± 0.09	26.39±0.76	34.55±1.94	**63.80**± 0.47
		PGD	42.31±0.31	45.84±1.20	**50.41**± 0.16	20.62±0.41	26.33±1.15	**39.26**± 0.19
		C&W$_\infty$	39.91±0.23	43.05±1.20	**47.75**± 0.17	20.00±0.62	26.08±1.39	**36.19**± 0.25
		APGD	41.41±0.37	44.03±1.10	**48.53**± 0.24	20.26±0.39	25.57±1.23	**38.05**± 0.19
		AA	37.64±0.33	40.89±0.60	**45.58**± 0.27	18.80±0.38	23.97±1.52	**34.32**± 0.30
	TRADES	–	70.73±0.56	75.07±0.45	**77.90**± 0.34	37.39±0.25	34.98±2.12	**64.99**± 0.47
		PGD	41.34±0.33	46.60±0.26	**50.25**± 0.13	21.41±0.57	26.30±0.97	**37.42**± 0.30
		C&W$_\infty$	37.37±0.38	42.80±0.62	**46.54**± 0.07	20.53±0.50	27.19±1.28	**33.11**± 0.53
		APGD	40.72±0.34	45.41±0.24	**49.20**± 0.13	21.19±0.85	25.63±1.38	**36.61**± 0.35
		AA	36.69±0.37	41.99±0.63	**45.53**± 0.16	18.30±0.62	24.18±1.67	**32.32**± 0.56
	MART	–	74.14±0.31	72.88±1.36	**75.51**± 0.38	27.06±1.69	35.30±1.23	**60.21**± 0.19
		PGD	42.12±0.42	49.39±1.31	**51.21**± 0.19	18.22±1.50	27.72±0.91	**41.87**± 0.24
		C&W$_\infty$	37.57±0.45	43.99±0.46	**46.65**± 0.16	19.89±1.61	26.45±1.76	**36.36**± 0.17
		APGD	41.34±0.53	47.84±1.07	**50.26**± 0.18	18.04±1.47	26.91±1.04	**40.82**± 0.20
		AA	36.68±0.48	42.08±0.58	**44.89**± 0.20	17.05±1.22	23.78±1.88	**34.95**± 0.26
VGG16	PGD	–	62.28±0.42	67.33±1.30	**78.21**± 0.22	21.28±1.49	23.41±1.38	**59.44**± 0.28
		PGD	37.54±0.51	41.47±0.96	**48.59**± 0.16	17.45±2.22	20.58±0.68	**37.62**± 0.31
		C&W$_\infty$	34.35±0.59	38.09±0.88	**45.15**± 0.33	17.46±2.32	20.99±0.92	**33.93**± 0.39
		APGD	36.38±0.74	39.84±0.96	**46.61**± 0.19	16.84±2.26	20.32±0.49	**36.11**± 0.35
		AA	32.21±0.82	36.15±0.53	**42.48**± 0.37	16.29±2.48	19.74±0.40	**32.16**± 0.39
	TRADES	–	64.59±0.32	68.34±1.33	**76.91**± 0.31	22.76±1.04	28.23±0.43	**62.46**± 0.69
		PGD	35.69±0.58	40.61±0.68	**47.79**± 0.18	15.60±0.42	20.52±0.68	**34.32**± 0.49
		C&W$_\infty$	31.90±0.46	36.66±1.06	**43.85**± 0.18	15.91±0.56	20.71±0.92	**29.85**± 0.60
		APGD	34.91±0.59	39.39±0.43	**46.32**± 0.28	14.63±0.57	20.04±0.62	**33.26**± 0.50
		AA	30.99±0.51	35.67±0.88	**42.65**± 0.16	13.17±0.66	18.46±0.71	**29.21**± 0.50
	MART	–	66.24±0.53	64.69±0.74	**73.49**± 0.42	25.01±1.56	19.33±1.71	**56.85**± 0.94
		PGD	35.20±0.90	44.18±1.53	**51.03**± 0.13	17.95±2.34	17.95±0.89	**40.21**± 0.31
		C&W$_\infty$	30.75±0.47	38.20±0.34	**44.34**± 0.14	18.49±2.20	17.60±0.78	**33.34**± 0.45
		APGD	33.83±0.96	42.47±1.21	**48.94**± 0.23	16.93±2.77	17.23±0.80	**38.56**± 0.32
		AA	29.48±0.74	36.45±0.59	**42.20**± 0.15	15.54±2.81	16.14±0.92	**31.74**± 0.41

CIFAR-10 as well as SVHN. However, its performance drops drastically when increasing sparsity to 99.9 %. For SVHN, the models are even completely damaged, yielding a balanced accuracy of 10 %. Also for ImageNet, Robust-ADMM significantly harms the model robustness and natural performance. HARP, in turn, exceeds the performance of Robust-ADMM distinctively. Models pruned by our method preserve natural accuracy and robustness significantly better for high compression rates: At a sparsity of 99 %, the natural accuracy and the adversarial robustness is at a comparable level to the original model on CIFAR-10 and SVHN. Even for aggressive pruning to a sparsity of 99.9 %, the models show substantial resistance against adversarial inputs across training methods and considered attackers. On ImageNet, HARP outperforms Robust-ADMM in moderate and aggressive pruning with 20 percentage points for natural accuracy and 10 percentage points for robustness against various attacks.

Comparison to HYDRA. Sehwag et al. (2020) demonstrate remarkable performance on robustly pruning a model's weights, which we are able to confirm in our experiments. The results are presented in Tables 1 to 3 for CIFAR-10, SVHN, and ImageNet. HYDRA consistently improves upon Robust-ADMM for the sparsity levels considered in our experiments. However, while HYDRA maintains its natural accuracy and robustness for a sparsity of 99 % well, for 99.9 % its performance degrades heavily (similarly to Robust-ADMM). Differently, HARP is able to maintain both accuracy metrics significantly better for aggressive pruning. In most cases, our method doubles the accuracy of HYDRA at a sparsity of 99 %, yielding an unprecedentedly competitive performance for such highly compressed networks. Also, the additionally gained robustness stemming from using MART-AT rather than TRADES-AT nicely transfers to models pruned by HARP, while HYDRA cannot benefit from this advantage at the same level. For ImageNet, HARP is also able to outperform

Table 2: Comparing HARP with Robust-ADMM and HYDRA on models learned on SVHN.

	Adversarial Training	Attack	99 % Sparsity			99.9 % Sparsity		
			R-ADMM	HYDRA	HARP	R-ADMM	HYDRA	HARP
ResNet18	PGD	–	84.34±1.44	88.71 ±0.49	**90.70**± 0.79	10.00±0.00	36.00±1.34	**81.55**± 0.45
		PGD	43.55±1.73	50.56 ±0.51	**57.21**± 1.96	10.00±0.00	19.69±0.32	**42.38**± 0.24
		C&W$_\infty$	38.38±0.96	45.53 ±0.64	**51.37**± 1.44	10.00±0.00	20.17±1.36	**37.72**± 0.14
		APGD	40.15±0.87	46.92 ±0.28	**47.40**± 1.27	10.00±0.00	18.48±0.32	**38.80**± 0.16
		AA	35.47±0.77	**42.40**±0.28	42.07 ± 2.10	10.00±0.00	16.49±0.60	**34.31**± 0.32
	TRADES	–	79.35±1.45	86.72 ±2.99	**88.61**± 1.38	11.68±3.77	29.81±3.70	**73.33**± 0.99
		PGD	40.92±1.55	46.22 ±6.17	**57.80**± 1.56	10.55±1.24	15.17±3.20	**37.43**± 0.20
		C&W$_\infty$	35.31±1.67	42.05 ±4.01	**50.86**± 1.47	10.42±0.95	14.07±1.99	**32.74**± 0.13
		APGD	37.81±1.88	42.40 ±7.12	**43.64**± 1.64	10.49±1.10	14.37±3.74	**34.85**± 0.24
		AA	32.95±1.49	**38.43**±5.34	36.42 ± 2.42	10.35±0.79	12.52±2.81	**30.71**± 0.29
	MART	–	85.39±0.82	86.16 ±1.65	**88.29**± 0.95	25.18±8.65	35.32±4.04	**74.99**± 1.82
		PGD	44.13±1.21	50.87 ±0.82	**57.25**± 2.52	14.40±2.80	19.93±1.92	**41.16**± 0.76
		C&W$_\infty$	36.50±1.28	43.45 ±0.78	**48.77**± 1.50	13.30±2.28	20.95±1.61	**34.46**± 0.61
		APGD	40.13±1.16	47.88 ±0.64	**50.55**± 1.69	13.51±2.21	18.66±1.79	**38.27**± 0.69
		AA	34.29±1.34	40.08 ±0.42	**41.90**± 2.88	11.50±1.13	15.79±1.37	**31.92**± 0.30
VGG16	PGD	–	72.33±9.49	85.62 ±0.46	**88.97**± 0.71	10.00±0.00	21.79±3.10	**80.29**± 0.68
		PGD	34.48±6.21	45.31 ±0.55	**50.45**± 0.45	10.00±0.00	13.75±0.35	**40.85**± 0.60
		C&W$_\infty$	28.80±4.37	39.42 ±0.58	**45.54**± 0.48	10.00±0.00	13.95±0.51	**35.66**± 0.55
		APGD	31.22±4.05	41.19 ±0.49	**46.59**± 0.72	10.00±0.00	12.57±0.83	**37.17**± 0.44
		AA	25.79±4.13	36.24 ±0.48	**42.36**± 0.88	10.00±0.00	11.99±0.59	**32.47**± 0.57
	TRADES	–	74.10±2.12	81.87 ±2.61	**84.69**± 0.94	10.00±0.00	14.67±5.58	**75.78**± 0.88
		PGD	35.37±0.79	43.92 ±0.94	**50.11**± 0.47	10.00±0.00	10.94±1.33	**37.97**± 0.40
		C&W$_\infty$	29.63±0.79	37.94 ±0.71	**43.85**± 0.45	10.00±0.00	11.37±1.86	**32.56**± 0.18
		APGD	31.45±0.61	40.32 ±0.89	**46.33**± 0.55	10.00±0.00	10.23±0.58	**34.88**± 0.40
		AA	27.00±0.46	35.46 ±1.01	**41.32**± 0.62	10.00±0.00	10.07±0.44	**30.59**± 0.40
	MART	–	80.25±1.37	81.51 ±2.67	**84.09**± 0.51	10.00±0.00	17.02±3.95	**70.53**± 1.07
		PGD	39.07±2.11	46.00 ±0.87	**49.93**± 0.84	10.00±0.00	13.28±2.22	**37.56**± 0.79
		C&W$_\infty$	31.07±2.21	37.53 ±0.87	**42.70**± 0.74	10.00±0.00	13.10±2.18	**31.02**± 0.75
		APGD	33.24±1.64	38.88 ±0.85	**43.73**± 0.68	10.00±0.00	11.29±1.82	**33.42**± 1.28
		AA	26.69±1.09	31.31 ±0.95	**37.39**± 1.07	10.00±0.00	11.13±1.64	**27.18**± 1.05

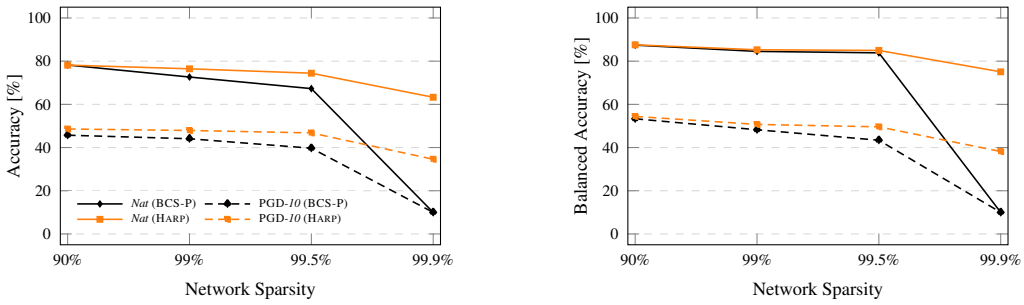
Table 3: Comparing HARP with Robust-ADMM and HYDRA on models learned on ImageNet.

	Attack	FREE-AT	90 % Sparsity			99 % Sparsity		
			R-ADMM	HYDRA	HARP	R-ADMM	HYDRA	HARP
ResNet18	–	60.25	35.83	49.08	**55.33**	11.38	27.88	**34.75**
	PGD	32.82	14.89	23.25	**27.32**	4.99	12.35	**14.81**
	C&W$_\infty$	30.67	12.60	21.21	**24.89**	3.87	11.32	**12.49**
	APGD	31.54	13.76	22.97	**25.69**	4.65	12.52	**13.47**
	AA	28.79	11.04	19.76	**22.98**	3.44	10.01	**11.08**

HYDRA noticeably and has the highest effectiveness in suppressing the model performance recession. However, the drop in performance from 90 % to 99 % sparsity is comparable to HYDRA and remains significant. Limiting the decay even further for large-scale datasets remains a challenge.

Comparison to BCS-P. Similar to our approach, BCS-P (Özdenizci & Legenstein, 2021) strives for a non-uniform compression strategy. To do so, they introduce "Bayesian Connectivity Sampling" for selecting important network elements from a global view. In contrast to HARP, it however does not follow the staged pruning-pipeline of Han et al. (2015) but achieves robust pruning via an end-to-end sparse training-procedure. While it does not allow for channel pruning, BCS-P outperforms both HYDRA and Robust-ADMM in pruning weights. In Fig. 2, we compare HARP to BCS-P on VGG16 trained with TRADES. We observe that HARP surpasses BCS-P significantly with increasing model sparsity and yields a functional model at 99.9 % sparsity, while BCS-P is not. The exact results of our experiments are presented in Table 9 in the appendix.

Figure 2: Comparing HARP with BCS-P for VGG16 trained on CIFAR-10 and SVHN.

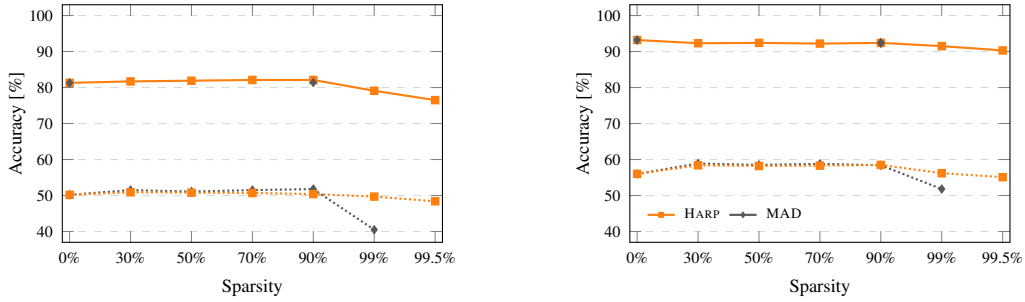


(a) CIFAR-10 (b) SVHN

Comparison to MAD. Lee et al. (2022) propose to compute an adversarial saliency map of a robustly pre-trained model via "Kronecker-Factored Approximate Curvature" (Martens & Grosse, 2015) and mask out weights that are least relevant for robustness. Unfortunately, we have not been able to fully reproduce the results for MAD. In Fig. 3, we thus present the numbers reported by the authors of MAD along-side results for HARP in the experimental setup described in the original publication and adjust pre-training accordingly for CIFAR-10 and SVHN. Lee et al. (2022) use a PGD step-size of 0.0069 for adversarial pre-training over 60 epochs and a 2×10^{-4} weight decay. At moderate compression (< 90 % sparsity), MAD and HARP are on a par. Neither one harms the performance of the neural network for CIFAR-10 and SVHN. As indicated in Fig. 3, showing the natural accuracy and adversarial robustness for both methods side-by-side, this changes for 99 % sparsity. While our method is able to still maintain robustness, MAD drops in performance. The natural accuracy remains at a comparable level. Please note, that Lee et al. (2022) only report isolated measurement points for the natural accuracy. Particularly at sparsity 99.5 %, we HARP continues to achieve comparable or even higher network robustness than MAD's results at sparsity 99 %.

Figure 3: Comparing HYDRA with MAD. All pre-trained robust models are originally from MAD. Solid lines and dotted lines represent the natural accuracy and PGD-10 adversarial robustness respectively on VGG16.

(a) CIFAR-10 (b) SVHN



4.2 Ablation Study

HARP features two important components: Learning compression quotas and learning what connections to prune. While we do argue that both aspects are crucial for effective adversarial robust pruning, in this section, we set out to analyze whether this indeed is the case. We perform an ablation study regarding the compression quota to be learned $\boldsymbol{r}$ and determining importance scores $\boldsymbol{S}$. For the experiments reported in Table 4, we thus either learn compression quotas or importance scores, while keeping the other one constant—consequently, we also do not balance pruning objectives here. Note that for the optimization on compression quotas $\boldsymbol{r}$, importance scores $\boldsymbol{S}$ remain as initialized by Eq. (9), while during investigating the influence of scores $\boldsymbol{S}$, we use uniform compression. We denote our method with compression-rate optimization only as HARP-$\boldsymbol{r}$, and with importance-score optimization only as HARP-$\boldsymbol{S}$.

As already observed in Fig. 1, moderate compression manages to maintain natural accuracy and adversarial robustness very well, such that in this study, we focus on 99 % and 99.9 % sparsity. Interestingly, HARP-$\boldsymbol{r}$ appears to be particular beneficial for moderate to high compression, while HARP-$\boldsymbol{S}$ is increasingly important for aggressive pruning. In particular for VGG16 with its cascading network architecture, this tendency is apparent for different pre-training strategies. However, at either sparsity level the combination of both, that is, optimizing compression rates $\boldsymbol{r}$ *and* importance scores $\boldsymbol{S}$, allows HARP to excel.

Table 4: Influence of optimizing either compression rates (HARP-$\boldsymbol{r}$) or importance scores (HARP-$\boldsymbol{S}$). The natural accuracy and PGD-10 adversarial robustness are presented left and right of the / character.

Model	Adv. Train	99 % Sparsity			99.9 % Sparsity		
		HARP-$\boldsymbol{r}$	HARP-$\boldsymbol{S}$	HARP	HARP-$\boldsymbol{r}$	HARP-$\boldsymbol{S}$	HARP
ResNet18	PGD	76.39 / 46.64	72.05 / 43.69	**80.25 / 50.36**	41.66 / 27.54	57.66 / 35.92	**63.99 / 39.39**
	TRADES	73.31 / 45.14	75.50 / 46.37	**77.78 / 50.16**	73.31 / 45.14	75.50 / 46.37	**77.78 / 50.16**
	MART	70.08 / 48.38	75.27 / 47.11	**75.88 / 50.79**	70.08 / 48.38	75.27 / 47.11	**75.88 / 50.79**
VGG16	PGD	76.17 / 46.74	65.09 / 39.80	**78.50 / 48.71**	36.76 / 28.02	50.33 / 34.03	**59.13 / 37.36**
	TRADES	72.91 / 44.52	66.75 / 41.79	**76.46 / 48.01**	41.63 / 26.95	56.08 / 31.51	**63.43 / 34.64**
	MART	71.63 / 48.64	64.37 / 41.46	**73.04 / 51.09**	37.19 / 30.68	49.51 / 36.29	**55.02 / 39.39**

4.3 Strategy Analysis

Also in conventional network pruning, non-uniform compression strategies have been proven effective, for instance, ERK by Evci et al. (2020) and LAMP by Lee et al. (2021). While we are the first to show an equivalent benefit for adversarial robust pruning, in this section, we investigate (a) if these approaches can also be applied to adversarial robust pruning, and (b) if they find similar compression strategies in comparison to HARP. Both ERK and LAMP focus on model parameters and support weight pruning only. We adjust HYDRA to accept different compression rates per layer and assign the strategies determined by ERK and LAMP. The results for pruning a PGD-AT pretrained VGG16 model are presented in Table 5. Surprisingly, LAMP in particular produces very promising compression strategies that, however, are surpassed by HARP.

In Fig. 4, we present the non-uniform compression strategies determined by the different approaches as (a) compression rates and (b) overall preserved parameters per layer. Additionally, we show a uniform compression yielding 99.9 % sparsity as reference. Note, that due to the layer's difference

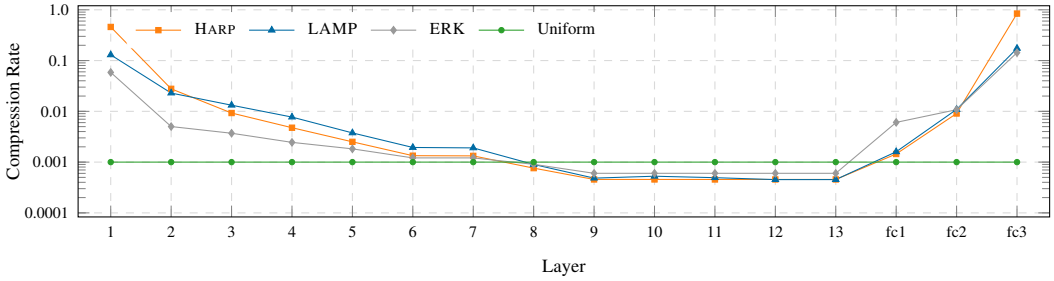
Table 5: Comparing performance improvement of Robust-ADMM and HYDRA by using ERK and LAMP and by HARP on CIFAR-10. The natural accuracy and PGD-10 robustness are presented left and right of the / character.

Model	Sparsity	R-ADMM			HYDRA			HARP
		Original	w/ ERK	w/ LAMP	Original	w/ ERK	w/ LAMP	
ResNet18	99 %	71.42 / 42.31	80.36 / 48.38	**80.64** / 48.28	75.53 / 45.84	79.09 / 49.17	80.16 / 50.07	**80.25 / 50.36**
	99.9 %	26.39 / 20.62	54.51 / 33.06	57.16 / 34.05	34.55 / 26.08	55.73 / 35.09	57.07 / 35.91	**63.99 / 39.39**
VGG16	99 %	62.28 / 37.54	70.33 / 43.30	74.38 / 46.39	67.33 / 41.47	72.19 / 45.05	76.75 / 47.96	**78.58 / 48.71**
	99.9 %	21.28 / 17.46	43.35 / 29.11	48.96 / 32.39	23.41 / 20.99	50.38 / 34.32	57.93 / 36.01	**59.13 / 37.36**

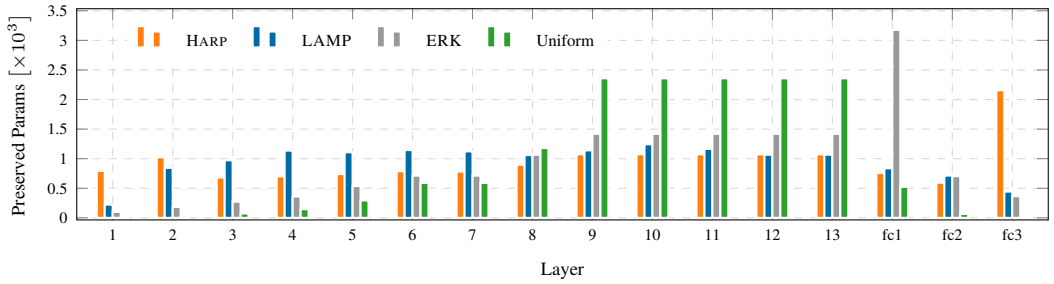
in size the absolute number of preserved parameters varies for the uniform strategy in Fig. 4b also. It is interesting to see that LAMP puts more focus on layers in the middle sacrificing input/output information, while HARP tends to focus on front and back layer stronger. ERK, in turn, even maintains large portions of `fc1`.

Figure 4: Strategies of HARP, LAMP, and ERK for VGG16 with a 99.9 % sparsity on CIFAR-10.

(a) Layer-wise compression rates



(b) Layer-wise preserved parameters



5 CONCLUSIONS

For pruning neural networks it is crucial to decide *how many and which parameters to prune*. This does not only affect natural accuracy, but also the robustness against adversarial input manipulations. Maintaining adversarial robustness is crucial for safety-critical applications, such as autonomous driving and edge AI. Our method, HARP, incorporates a global view of the network's compression and scores for gauging the importance of network connections into a dynamically regularized loss formulation that allows to significantly outperform related work. We are the first to reach competitive performance for highly aggressive pruning, aiming at up to 99.9 % network sparsity. Therewith, we show that learning a global, but layer-specific and, thus, non-uniform compression strategy is at least as important as deciding on what connections to prune.

Limitations. Similarly to the majority of related work in the field, our method is of empirical nature and, thus, can benefit from future work on theoretic analyses of how and why our pruning approach arrives at a particular compression strategy. In our evaluation, we show that HARP's compression is close to the theoretically founded strategies of LAMP, raising the hope that a rigorous theoretic justification is possible. Ideally this extends to a theoretically determined, one-shot calculation that spares the computational effort of the optimization problem considered for HARP.

CO_2 Emission. We have conducted all our experiments on Nvidia RTX-3090 GPU cards and have consumed about 8,580 GPU hours in total. This amounts to an estimated total CO_2 emissions of 1,831.83 kgCO2eq when using Google Cloud Platform in region `europe-west3`. However, our university consumes 100 % renewable-energy, such that our specific CO_2 emissions for the project amounts to 1.50 kgCO2eq. Estimates are conducted using the "Machine Learning Impact Calculator" (Lacoste et al., 2019).

REPRODUCIBILITY STATEMENT

For the sake of reproducibility and to foster future research, we make the implementations of HARP for holistic adversarially robust pruning publicly available at:

`https://—anonymized—`

Additionally, we provide the source code in the supplementary package in subfolder `harp-prune/`.

REFERENCES

Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In *Proc. of the International Conference on Pattern Recognition*, pp. 3121–3124, 2010. doi: 10.1109/ICPR.2010.764.

Bugra Cakir and Erdogan Dogdu. Malware classification using deep learning methods. In *Proc. of the ACM Southeast Conference (ACMSE)*, 2018.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Proc. of the IEEE Symposium on Security and Privacy*, pp. 39–57, 2017.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proc. of the International Conference on Machine Learning (ICML)*, 2020a.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *Proc. of the International Conference on Machine Learning (ICML)*, 2020b.

Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of relu networks via maximization of linear regions. In *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

J. Deng, W. Dong, R. Socher, et al. ImageNet: A large-scale hierarchical image database. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *Proc. of the International Conference on Machine Learning (ICML)*, pp. 2943–2952, 2020.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *Proc. of the International Conference on Learning Representations (ICLR)*, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. AMC: AutoML for model compression and acceleration on mobile devices. In *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 815–832, 2018.

Qiangui Huang, Shaohua Kevin Zhou, Suya You, and Ulrich Neumann. Learning to prune filters in convolutional neural networks. In *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 709–718, 2018.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020.

Kevin Kollek, Miguel Angel Aguilar, Marco Braun, and Anton Kummert. Improving neural network architecture compression by multi-grain pruning. In *Proc. of the IEEE International Conference on Progress in Informatics and Computing (PIC)*, pp. 6–13, 2021.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR (canadian institute for advanced research). URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *CoRR*, (arXiv:1910.09700), 2019.

Byung-Kwan Lee, Junho Kim, and Yong Man Ro. Masking adversarial damage: Finding adversarial saliency for robust and sparse network. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Jaeho Lee, Sejun Park, Sangwoo Mo, Sungsoo Ahn, and Jinwoo Shin. Layer-adaptive sparsity for the magnitude-based pruning. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2021.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient ConvNets. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2017.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology*, 13(2), apr 2022.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2019.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with restarts. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2016.

Divyam Madaan, Jinwoo Shin, and Sung Ju Hwang. Adversarial neural pruning with latent vulnerability suppression. In *Proc. of the International Conference on Machine Learning (ICML)*, pp. 6575–6585, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2018.

Huizi Mao, Song Han, Jeff Pool, Wenshuo Li, Xingyu Liu, Yu Wang, and William Dally. Exploring the granularity of sparsity in convolutional neural networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1927–1934, 07 2017.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *Proc. of the International Conference on Machine Learning (ICML)*, pp. 2408–2417, 2015.

Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *Proc. of the International Conference on Machine Learning (ICML)*, pp. 2498–2507, 2017.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Proc. of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Ozan Özdenizci and Robert Legenstein. Training adversarially robust sparse networks via bayesian connectivity sampling. In *Proc. of the International Conference on Machine Learning (ICML)*, pp. 8314–8324, 2021.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2021.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015.

Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. HYDRA: Pruning adversarially robust neural networks. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2015.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2014.

Lukas Timpl, Rahim Entezari, Hanie Sedghi, Behnam Neyshabur, and Olga Saukh. Understanding the effect of sparsity on neural networks robustness. In *Proc. of the ICML Workshop on Overparameterization: Pitfalls & Opportunities*, July 2021.

Manoj Rohit Vemparala, Nael Fasfous, Alexander Frickenstein, Sreetama Sarkar, Qi Zhao, Sabine Kuhn, Lukas Frickenstein, Anmol Singh, Christian Unger, Naveen Shankar Nagaraja, Christian Wressnegger, and Walter Stechele. Adversarial robust model compression using in-train pruning. In *Proc. of the CVPR Workshop on Safe Artificial Intelligence for Automated Driving (SAIAD)*, June 2021.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2020.

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 2082–2090, 2016.

Boxi Wu, Jinghui Chen, Deng Cai, Xiaofei He, and Quanquan Gu. Do wider neural networks really help adversarial robustness? In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Tien-Ju Yang, Yu-Hsin Chen, and Vivienne Sze. Designing energy-efficient convolutional neural networks using energy-aware pruning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6071–6079, July 2017.

Shaokai Ye, Xue Lin, Kaidi Xu, Sijia Liu, Hao Cheng, Jan-Henrik Lambrechts, Huan Zhang, Aojun Zhou, Kaisheng Ma, and Yanzhi Wang. Adversarial robustness vs. model compression, or both? In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 111–120, 2019.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proc. of the International Conference on Machine Learning (ICML)*, 2019.