# HYDRA: Pruning Adversarially Robust Neural Networks

Vikash Sehwag*, Shiqi Wang#, Prateek Mittal*, Suman Jana#

* Princeton University # Columbia University

PRINCETON UNIVERSITY

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

NEURAL INFORMATION PROCESSING SYSTEMS

**Project webpage**: https://vsehwag.github.io/hydra

## Motivation and Research Question

Deep neural networks face two key challenges: 1) Lack of robustness against adversarial examples and 2) Large size of neural networks. We often utilize robust training to solve the former and use network pruning techniques to solve the latter. However, both robust training and pruning methods are largely studied ***independently*** in earlier works.

**Motivation**: We observe that naively integrating pruning methods, such as pruning connections with the smallest magnitude weight, with robust training achieves poor performance.

**Research Question**: How to reduce the size of neural networks while preserving both accuracy and robustness?

## Our Contributions

1. We propose HYDRA, a novel pruning approach for robustly trained networks, including provably robust networks.
2. We achieve **state-of-the-art** accuracy and robustness, simultaneously for compressed networks over CIFAR-10, SVHN, and ImageNet dataset.
3. We demonstrate that provably robust subnetworks are ***hidden*** inside even non-robust networks.

## Our Approach

**Guiding Pruning with Robust Training Loss**: We integrate robust training objective in the pruning process. We solve the resulted empirical risk minimization problem efficiently using SGD.

$$\hat{m} = \underset{m \in \{0,\,1\}^N}{argmin} \underset{(x,y)\sim\mathcal{D}}{E}[L_{pruning}(\theta_{pretrain} \odot m, x, y)] \qquad s.t.\ \|m\|_0 \le k$$

**Importance Scores**: Since finding Boolean pruning mask is a discrete optimization, we convert it to a continuous optimization problem by assigning real-valued importance score to each connection.

**Scaled Initialization**: Crucial to the success of HYDRA is our proposed scaled initialization of importance scores based on pretrained network weights.

$$s_i^{(0)} \propto \frac{1}{max(|\theta_{pretrain,i}|)} \times \theta_{pretrain,i}$$

**Algorithm 1** End-to-end compression pipeline.

**Inputs**: Neural network parameters ($\theta$), Loss objective: $L_{pretrain}, L_{prune}, L_{finetune}$, Importance scores ($s$), pruning ratio ($p$)

**Output**: Compressed network, i.e., $\theta_{finetune}$

Step 1: Pre-train the network.

$$\theta_{pretrain} = \underset{\theta}{argmin} \underset{(x,y)\sim\mathcal{D}}{E}[L_{pretrain}(\theta, x, y)]$$

Step 2: Initialize scores ($s$) for each layer.

$$s_i^{(0)} = \sqrt{\frac{6}{fan\text{-}in_i}} \times \frac{1}{max(|\theta_{pretrain,i}|)} \times \theta_{pretrain,i}$$

Step 3: Minimize pruning loss.

$$\hat{s} = \underset{s}{argmin} \underset{(x,y)\sim\mathcal{D}}{E}[L_{prune}(\theta_{pretrain}, s, x, y)]$$

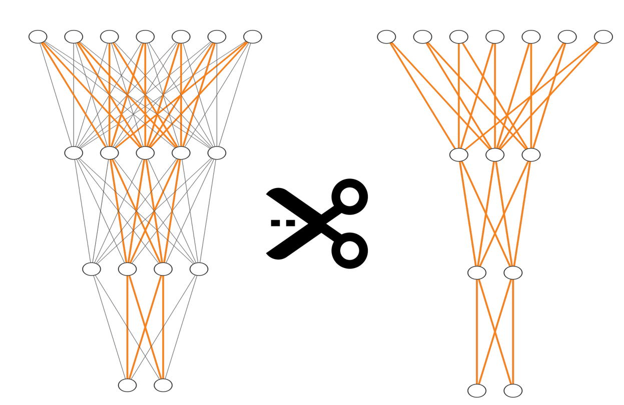
Step 4: Create binary pruning mask $\hat{m} = \mathbb{1}(|\hat{s}| > |\hat{s}|_k)$, $|\hat{s}|_k$: $k$th percentile of $|\hat{s}|$, $k = 100 - p$

Step 5: Finetune the non-pruned connections, starting from $\theta_{pretrain}$.

$$\theta_{finetune} = \underset{\theta}{argmin} \underset{(x,y)\sim\mathcal{D}}{E}[L_{finetune}(\theta \odot \hat{m}, x, y)]$$

## Hidden Robust Sub-Networks

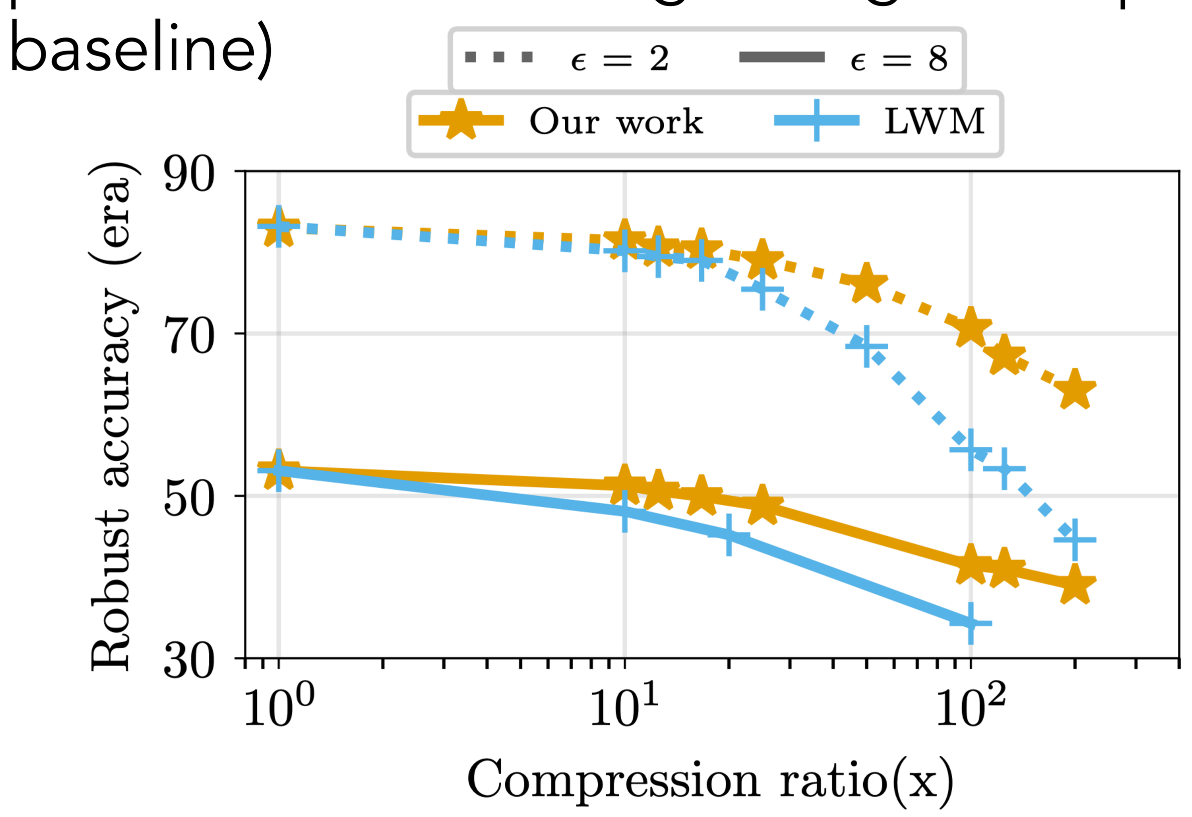
Given a non-robust pre-trained network, we optimize to maximize the robust accuracy only by pruning connections, i.e., no update in the weight of any connections.



Only by pruning certain connections, we find even provably robust sub-networks in non-robust networks.

## Experimental Results

Comparison with least weight magnitude pruning [1] (*first* baseline)




Comparison with Adv-ADMM [2] (*second* baseline)

Using (Benign test accuracy) / (Empirical robust accuracy) format

| Network | Adv-ADMM | Ours | Δ |
|---|---|---|---|
| ResNet-18 | 58.7/36.1 | 69.0/41.6 | +10.3/+5.5 |
| ResNet-34 | 68.8/41.5 | 71.8/44.4 | +3.0/+2.9 |
| ResNet-50 | 69.1/42.2 | 73.9/45.3 | +4.8/+3.1 |
| WRN-28-2 | 48.3/30.9 | 54.2/34.1 | +5.9/+3.2 |
| GoogleNet | 53.4/33.8 | 66.7/40.1 | +13.3/+6.3 |
| MobileNet-v2 | 10.0/10.0 | 39.7/26.4 | +29.7/+16.4 |

| Pre-training objective | | Targeted metric for each sub-network | | |
|---|---|---|---|---|
| | | benign accuracy | era | vra-s |
| Benign training | (benign accuracy = 95.0) | 95.0 | 43.5 | 53.0 |
| Adversarial training | (era = 51.9) | 94.1 | 51.4 | 63.6 |
| Randomized smoothing | (vra = 61.1) | 93.7 | 48.8 | 60.7 |

1. Han, Song, et al. "Learning both weights and connections for efficient neural network." *NeurIPS* 2015.
2. Ye, Shaokai, et al. "Adversarial Robustness vs. Model Compression, or Both?." ICCV 2019.