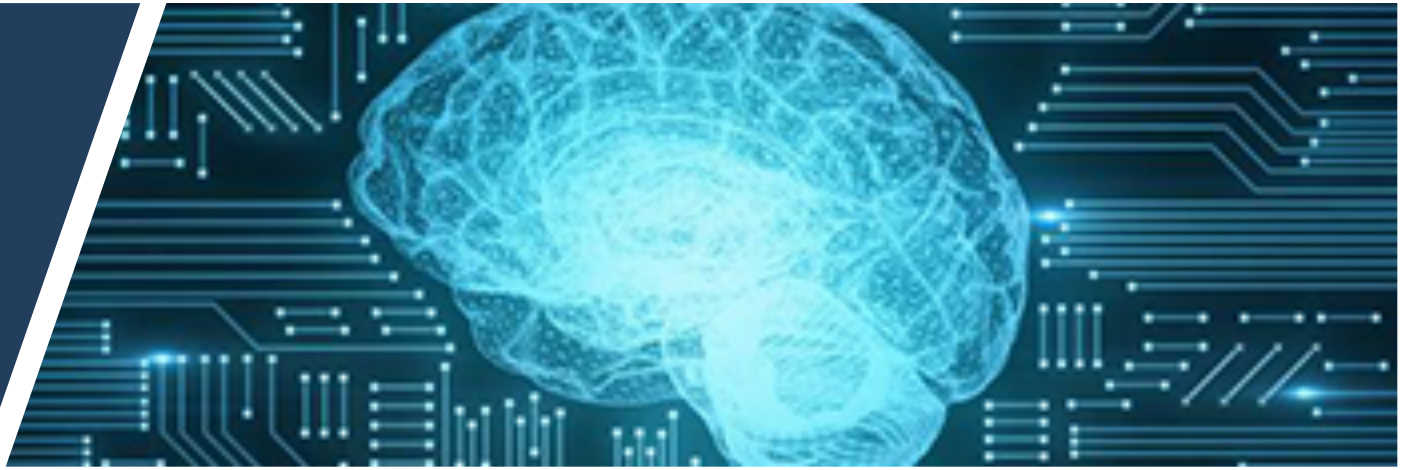


Activation Density Driven Efficient Pruning in Training

Tim Foldy-Porto, Yeshwanth Venkatesha*, and Priya Panda

Electrical engineering

Yale University

Yale University

LUX ET VERITAS

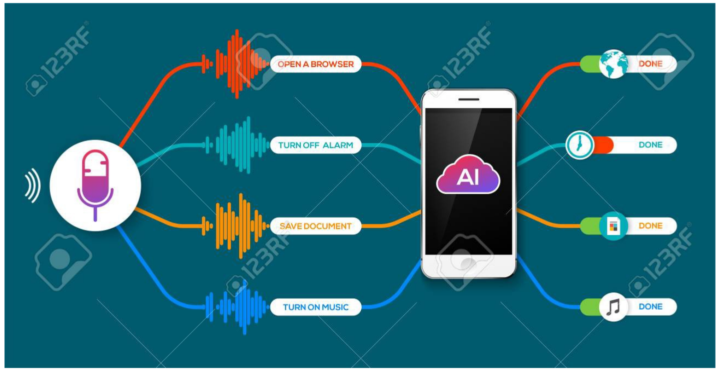
Intelligent Computing Lab

https://intelligentcomputinglab.yale.edu/

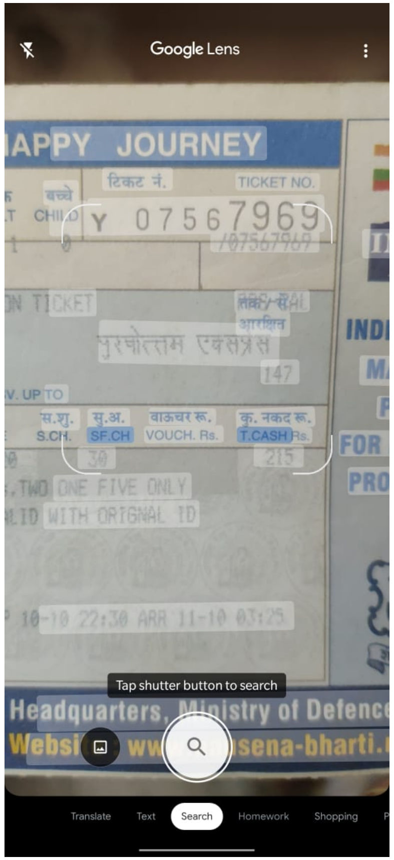
Motivation

- AI Applications on embedded devices
- Power and memory constraint

Autonomous driving



Speech Recognition



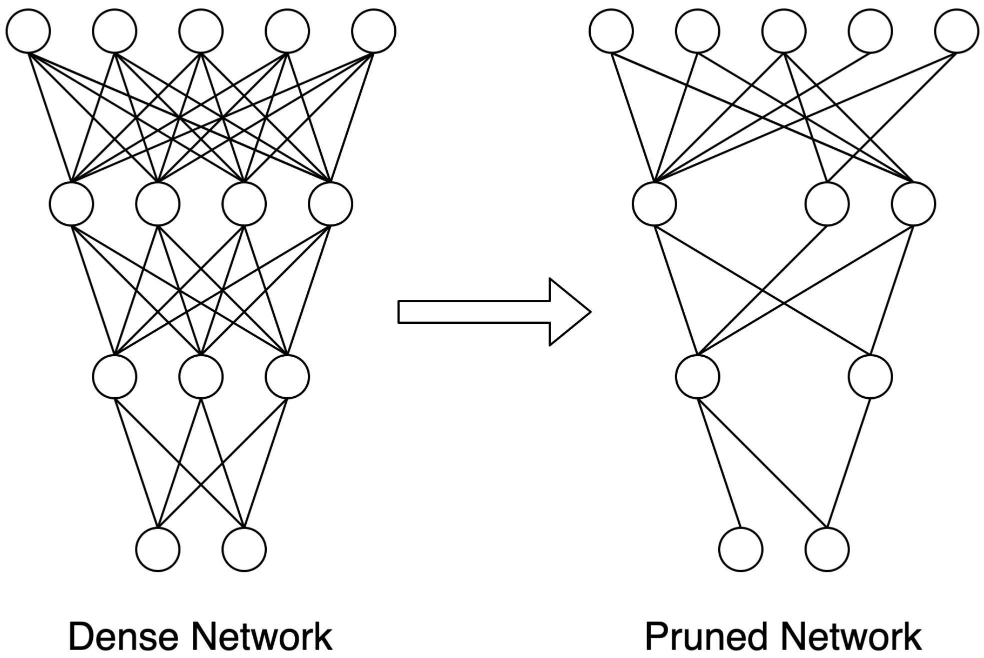
Personal Assistants

Yale University

https://www.pbs.org/newshour/science/in-a-crash-should-self-driving-cars-save-passengers-or-pedestrians-2-million-people-weigh-in
https://medium.com/analytics-vidhya/automatic-speech-recognition-systems-in-deep-learning-a6f91bbe7500

Pruning

- Remove redundant neurons and synapses while maintaining accuracy



Previous Work

Train a Dense Network

- Fully trained dense network

Prune

- Sparse network

Retrain

- To regain the accuracy drop

Yale University

[1] J. Frankle and M. Carbin, “The lottery ticket hypothesis: Finding sparse, trainable neural networks,” arXiv:1803.03635, 2018.
[2] L. Liu, L. Deng, X. Hu, M. Zhu, G. Li, Y. Ding, and Y. Xie, “Dynamic sparse graph for efficient deep learning,” arXiv:1810.00859, 2018.

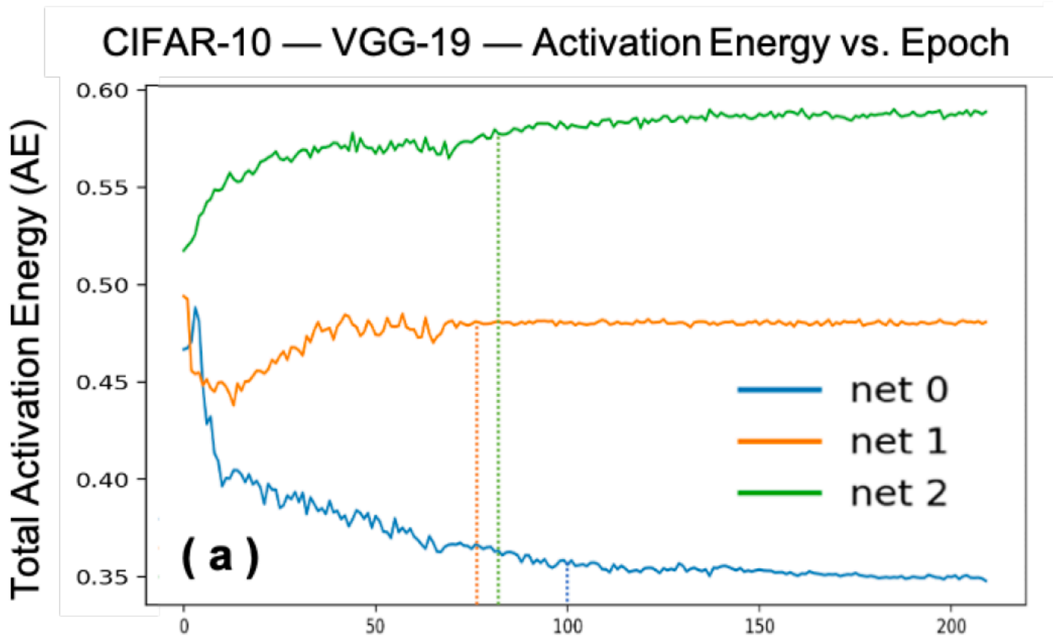
Contributions

- In-training pruning method
- Novel metric - Activation Density/Energy (AD/AE)
- OPS reduction on benchmark datasets
- Compute cost of networks during both training and inference.

Activation Energy

- Key Observation: Number of non-zero activations decreases as training progresses
- Activation Energy: The density of non-zero activations

$$AE = \frac{\#nonzero\ \ activations}{\#total\ \ activations}$$



Yale University

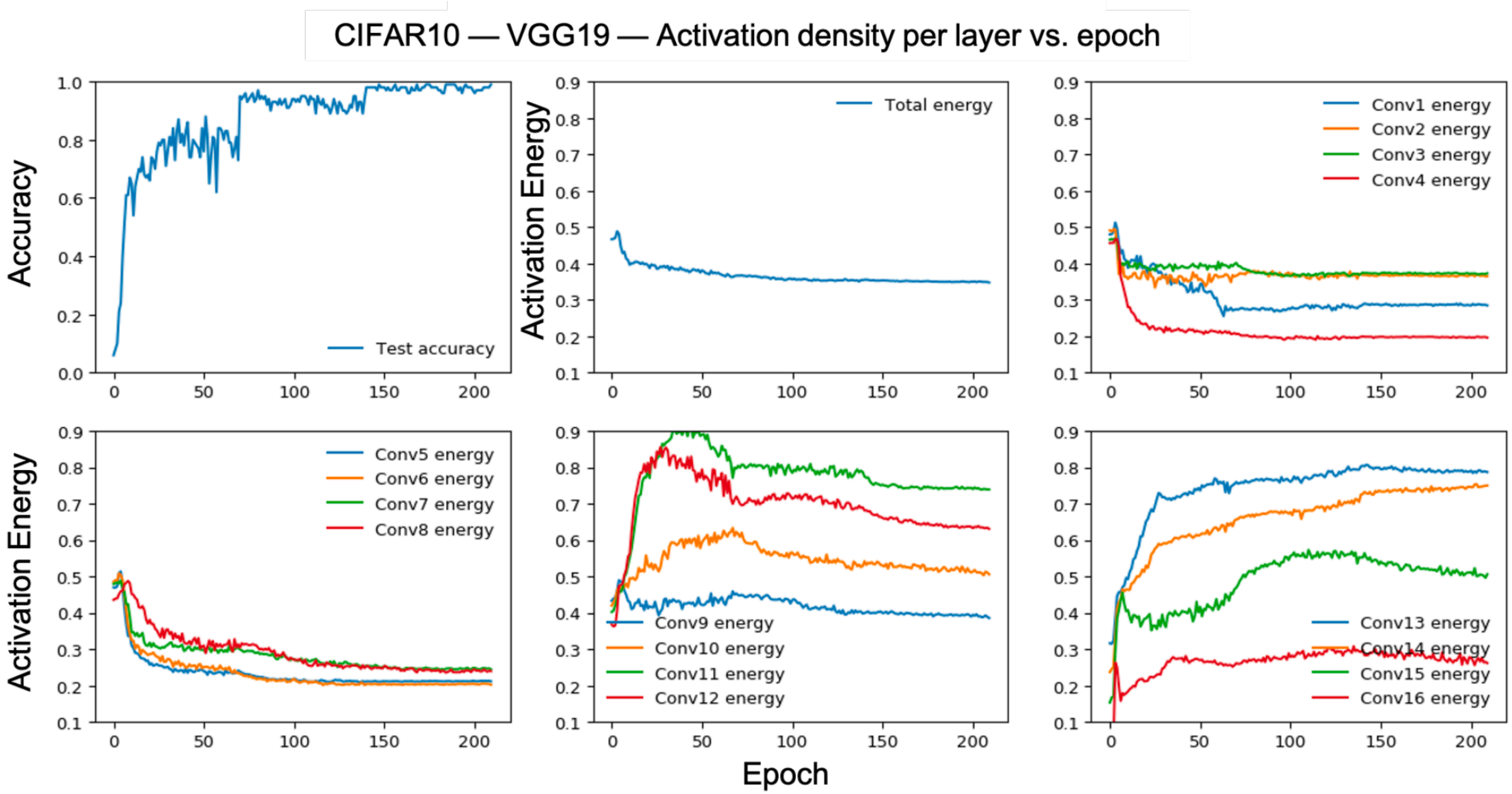
Algorithm

- Periodically monitor the AEs of the layers during the training process and prune the layers based on the density at regular training intervals
- Set pruning criteria to be equivalent to the saturation point.
- Stopping criteria based on the overall shape of AE vs. epoch curve for each network

Algorithm 1: Activation Density driven Pruning in Training

Input: Training dataset and randomly initialized network $net_{initial}$
Output: Trained and pruned network net_{final}
$net[0] = net_{initial}$
//Note, net[0] can be a large network like {VGG-19, ResNet18};
$epoch = 0$;
$index = 0$;
while ***not*** *stopping* (δ) *criteria* **do**
    net = Randomly Initialized ($net[index]$);
    while ***not*** *pruning* (ρ) *criteria* **do**
        train(net, $epoch$);
        for *L in net.Layers* **do**
            $\#nonzero[L] =$ count_nonzero_activations(L);
            AE[L] = $\frac{\#nonzero[L]}{\#total[L]}$;
        end
        $epoch++$;
        //Note, we train the network $net[index]$ while monitoring the layer-wise AE till ρ is satisfied.
    end
    $index++$;
    for *L in net.Layers* **do**
        $net[index].LayerSize[L] =$ AE[L] $\times net[index-1].LayerSize[L]$;
    end
    //Note, we prune the network $net[index-1]$ to get the compressed network $net[index]$ based on AE per layer. The pruning continues till δ is satisfied.
end
$net_{final} = net[index]$;

AE Trend



Yale University

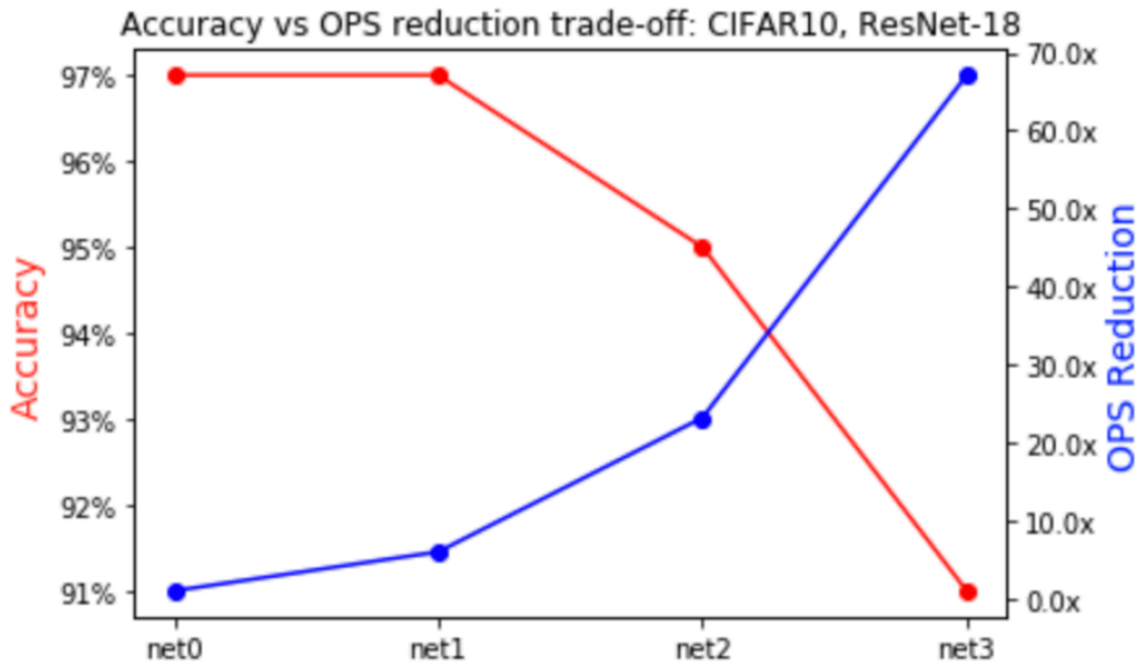
Results

- On an average 38% of channels are pruned in the first 8 layers layer 1-8 and 25% channels in the latter 8 layers layer 9-16 for VGG19 on CIFAR10.
- Shows effectiveness of the AE driven pruning for structured layer-wise network compression focused on overall OPS reduction.

Network	Configuration	Accuracy	Parameters reduction	OPS reduction	Training Epochs ρ
	CIFAR-10, ResNet18				
net 0	[64, 64, 64, 64, 64, 128, 128, 128, 128, 256, 256, 256, 256, 512, 512, 512, 512]	97 %	1×	1×	100 epochs
net 1	[34, 29, 41, 25, 33, 58, 78, 27, 65, 71, 83, 46, 69, 120, 191, 219, 288]	97 %	7.3×	6.0×	70 epochs
net 2	**[21, 16, 30, 10, 22, 24, 47, 9, 39, 26, 48, 12, 39, 41, 85, 63, 188]**	**95 %**	**41.2×**	**23.2×**	**70 epochs**
net 3	[14, 9, 21, 5, 15, 13, 32, 5, 26, 13, 34, 5, 25, 21, 45, 12, 142]	91 %	199.3×	67.1×	N/A
	CIFAR-10, VGG-19				
net 0	[64, 64, 128, 128, 256, 256, 256, 256, 512, 512, 512, 512, 512, 512, 512, 512]	97 %	1×	1×	100 epochs
net 1	**[18, 23, 47, 25, 54, 51, 62, 61, 197, 258, 378, 322, 402, 383, 259, 134]**	**94 %**	**3.1×**	**5.6×**	**70 epochs**
net 2	[10, 9, 30, 11, 21, 31, 22, 21, 62, 70, 113, 141, 256, 299, 194, 71]	93 %	10.3×	27.4×	N/A
	CIFAR-100, ResNet18				
net 0	[64, 64, 64, 64, 64, 128, 128, 128, 128, 256, 256, 256, 256, 512, 512, 512, 512]	81.0 %	1×	1×	25 epochs
net 1	**[39, 31, 49, 24, 44, 54, 90, 36, 84, 88, 155, 65, 136, 130, 231, 105, 300]**	**79.0 %**	**7.6×**	**5.1×**	**N/A**
	CIFAR-100, VGG-19				
net 0	[64, 64, 128, 128, 256, 256, 256, 256, 512, 512, 512, 512, 512, 512, 512, 512]	76.0 %	1×	1×	25 epochs
net 1	**[34, 23, 51, 30, 63, 63, 73, 82, 210, 285, 333, 357, 317, 259, 181, 106]**	**73.0 %**	**3.9×**	**5.3×**	**N/A**
	TinyImageNet, ResNet18				
net 0	[64, 64, 64, 64, 64, 128, 128, 128, 128, 256, 256, 256, 256, 512, 512, 512, 512]	51.54 %	1×	1×	25 epochs
net 1	**[31, 21, 47, 27, 48, 62, 99, 58, 94, 85, 161, 69, 133, 93, 152, 56, 247]**	**50.51 %**	**10.6×**	**4.7×**	**N/A**

Accuracy vs OPS Reduction Trade-off

- A decreasing AE implies we still have some redundancies in the network that can facilitate pruning without significant loss in accuracy
- Find a trade-off point when the AE curve starts increasing



Yale University

Training Complexity

- Captures the amount of time and training energy required to achieve a given model accuracy, compression and efficiency.

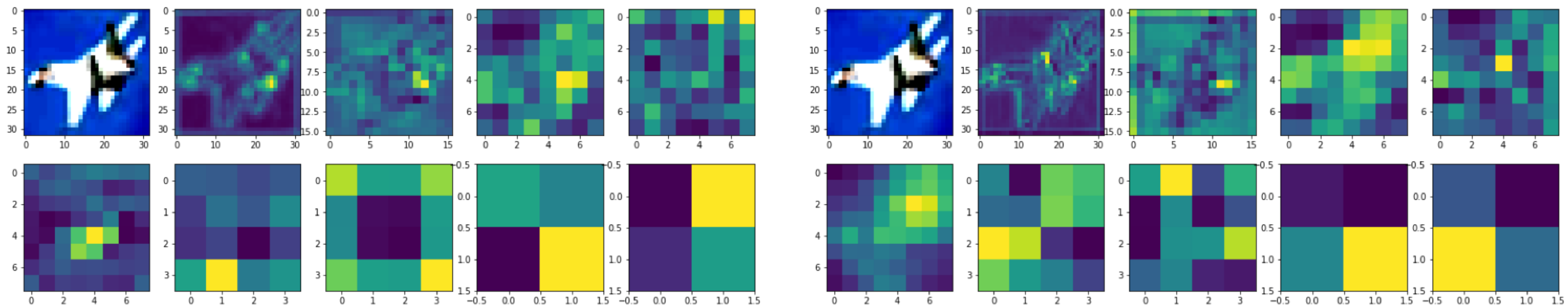
$$\sum_{net_i} (\text{OPS reduction}_{net_i})^{-1} \times (\#\ \text{training epochs}_{net_i})$$

Network	**ResNet18**			**VGG-19**	
	CIFAR-10	CIFAR-100	Tiny ImNet	CIFAR-10	CIFAR-100
net 0	210.0 (1×)	210.0 (1×)	60.0 (1×)	210.0 (1×)	210.0 (1×)
net 1	135.0 (0.64×)	66.2 (0.32×)	37.7 (0.62×)	120.2 (0.57×)	64.6 (0.31×)
net 2	120.8 (0.58×)	-	-	-	-

Activation Visualization

- Visualization of increasing activation density using a colormap (more color implying more neuronal activation)
- Although certain layers break the pattern, overall trend of higher AE in the layers of net2 than in the layers of net0

CIFAR-10 — VGG-19 — Activation visualization



net 0 *net 2*

Yale University

Comparison with Previous Work

- VGG19 on CIFAR100

Authors	Training complexity	Accuracy	Parameters reduction	OPS reduction
Garg et al. [5]	206.6	71 %	9.1×	3.9×
Liu et al. [13]	260.0	73 %	8.7×	1.6×
Ours	64.6	73 %	3.9×	5.3×

- Comparison with Lottery Ticket Hypothesis

Model	Authors	Training memory complexity	Accuracy	Parameters reduction	OPS reduction
ResNet18	LTH [8]	206.45	93 %	5.6×	N/A
	Ours	120.8	95 %	41.2×	23.2×
VGG-19	LTH [8]	105.1	93 %	35.7×	N/A
	Ours	129.4	93 %	10.3×	27.4×

Conclusion

- We propose an ‘Activation Density’ metric, a heuristic that provides a structured and interpretable way of optimizing the network architecture.
- We present a novel pruning in training method that yields significant compression benefits on state-of-the-art deep learning architectures.
- The progressive downsizing of a network during the training process yields training complexity benefits.
- We get considerable benefits in training complexity and compute-OPS-reduction over the baseline unpruned model, as well as over previously proposed pruning methods.

Yale University