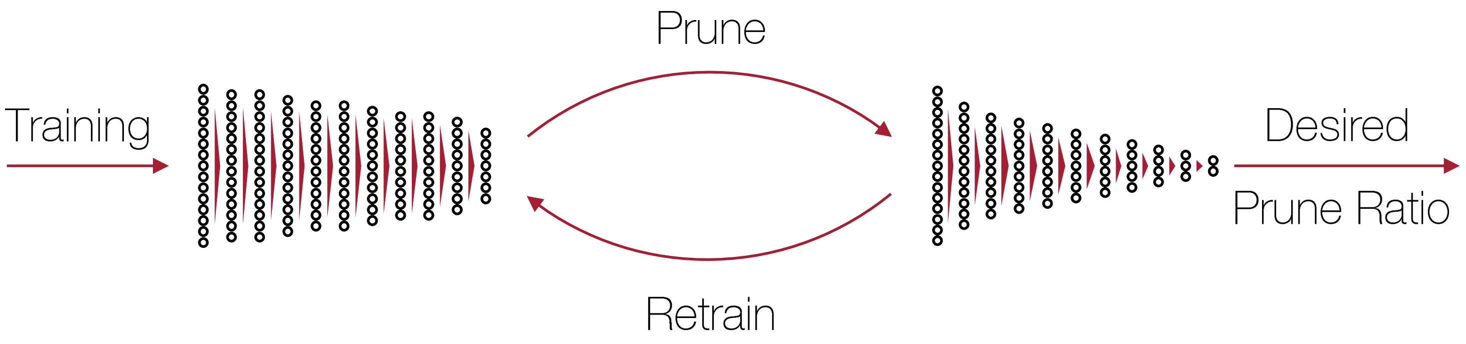
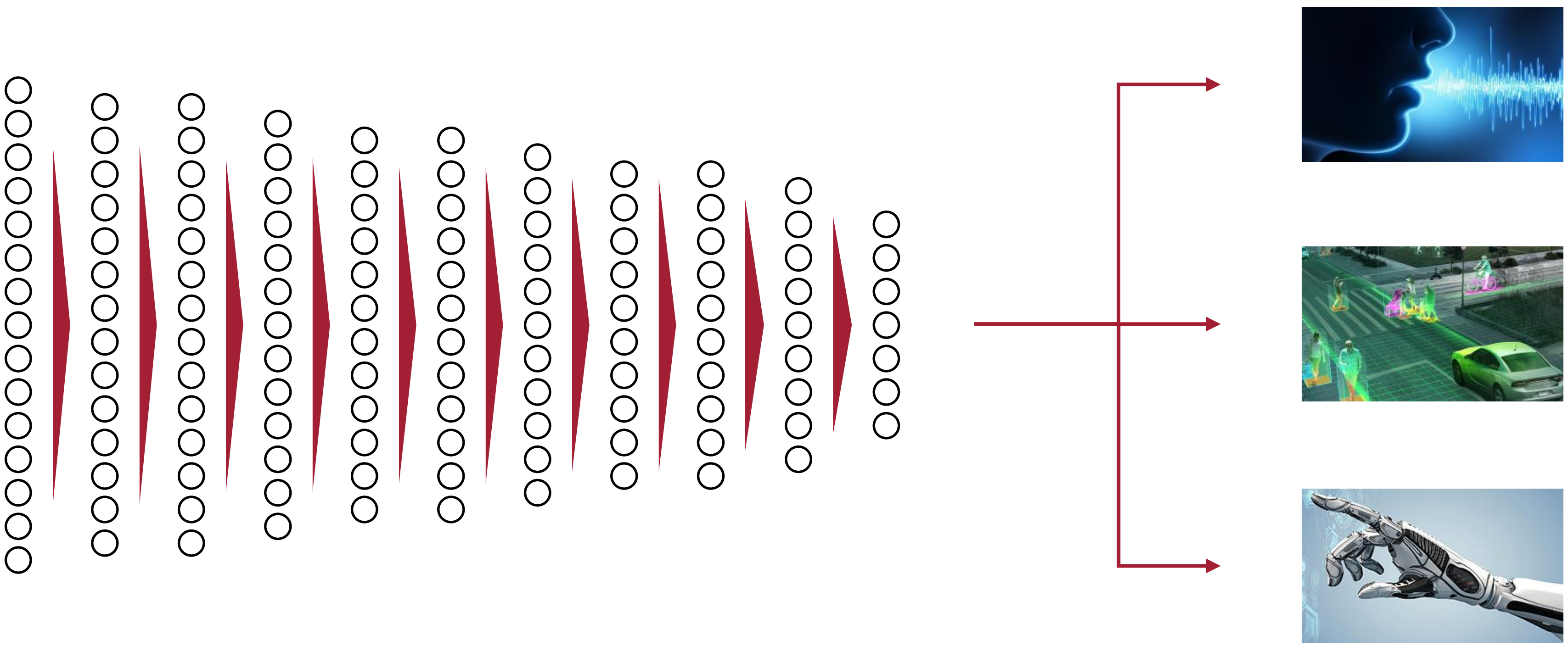


Lost in Pruning:
The Effects of Pruning Neural Networks beyond Test Accuracy

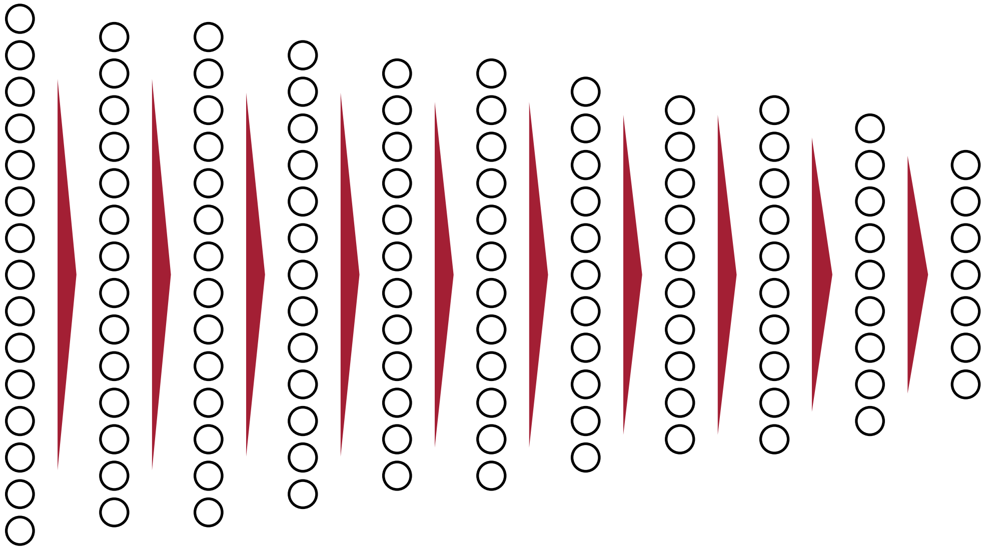
Lucas Liebenwein, lucasl@mit.edu, http://www.mit.edu/~lucasl/

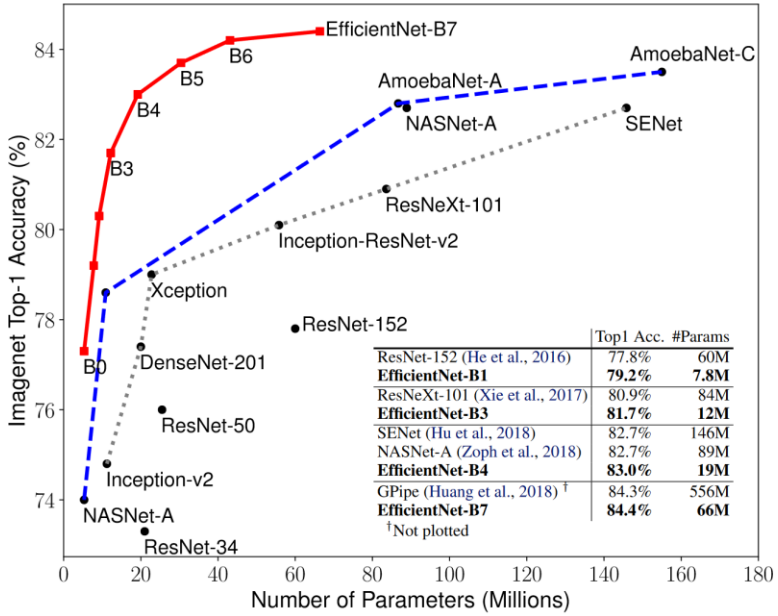
Distributed Robotics Lab, CSAIL, MIT

Neural networks are SOTA



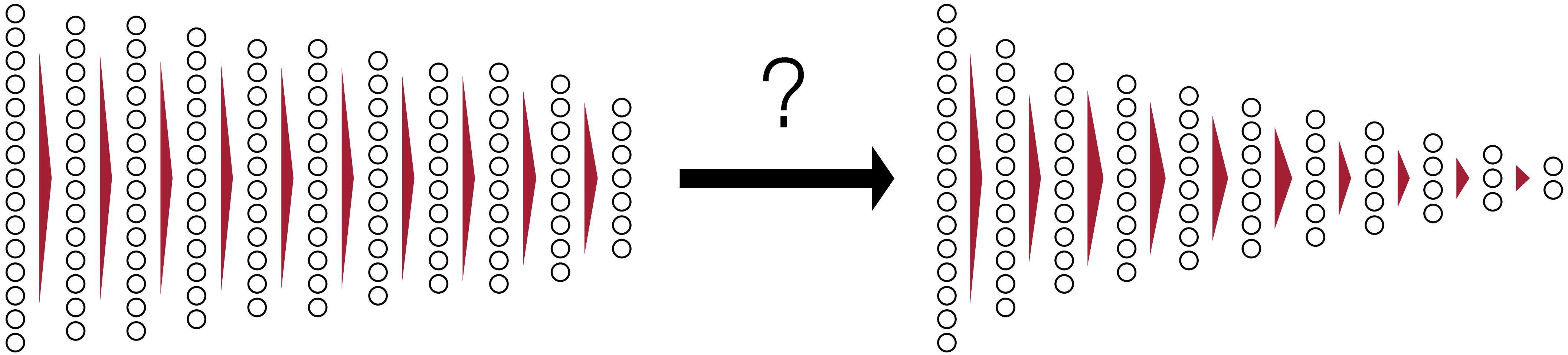
Larger size, better performance



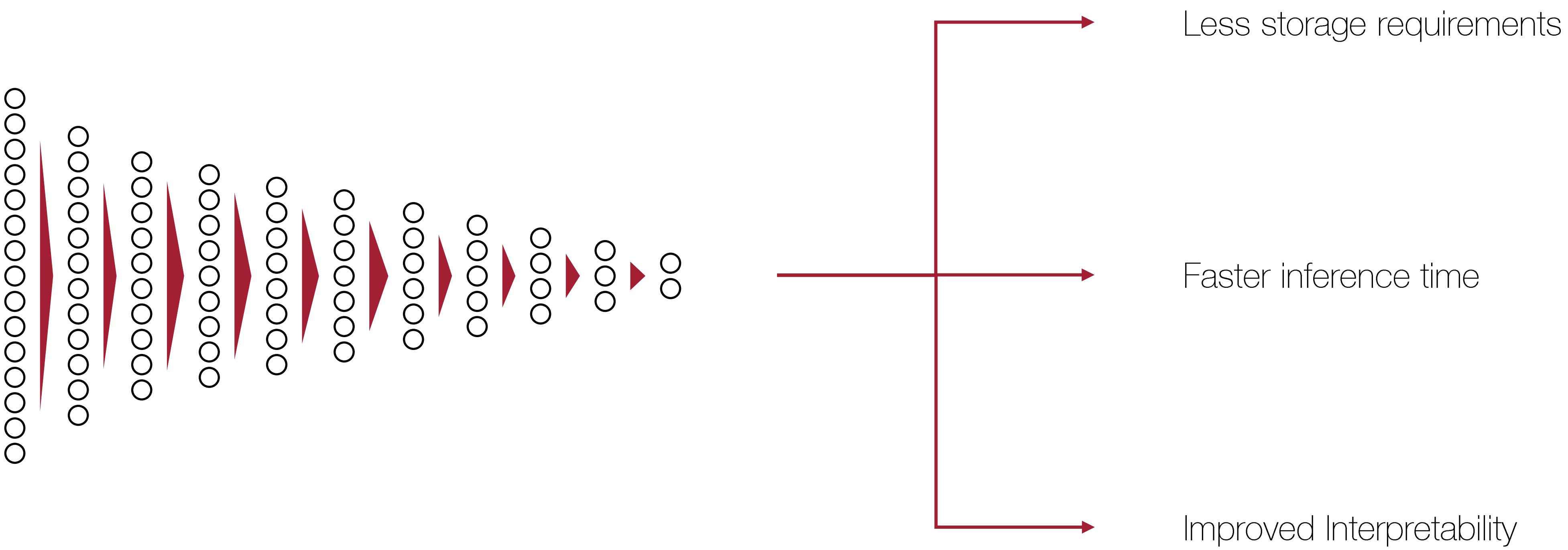


Tan, Mingxing, and Quoc Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." ICML. 2019.

Smaller size, same performance?



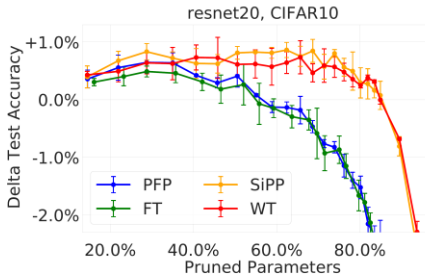
Smaller size, same performance?



Objective: what are the effects of pruning?

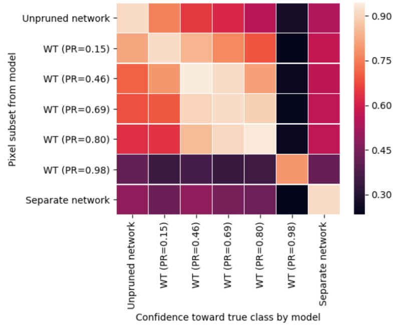
1

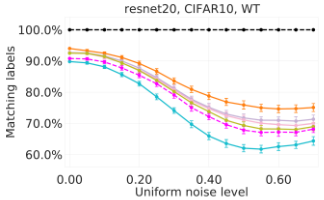
How we prune?



2

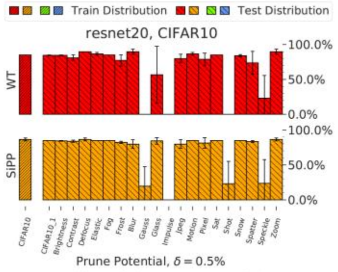
What is preserved?



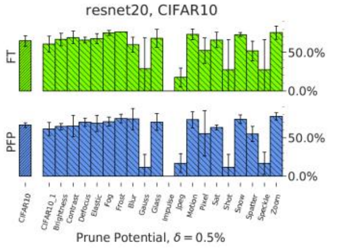


3

What is *lost*?

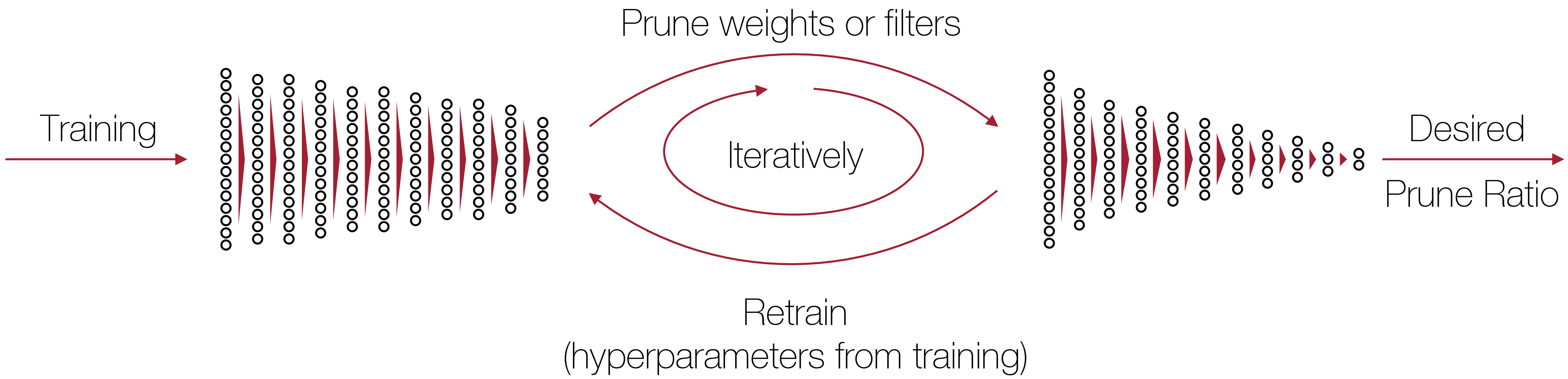


(b) Weight prune potential

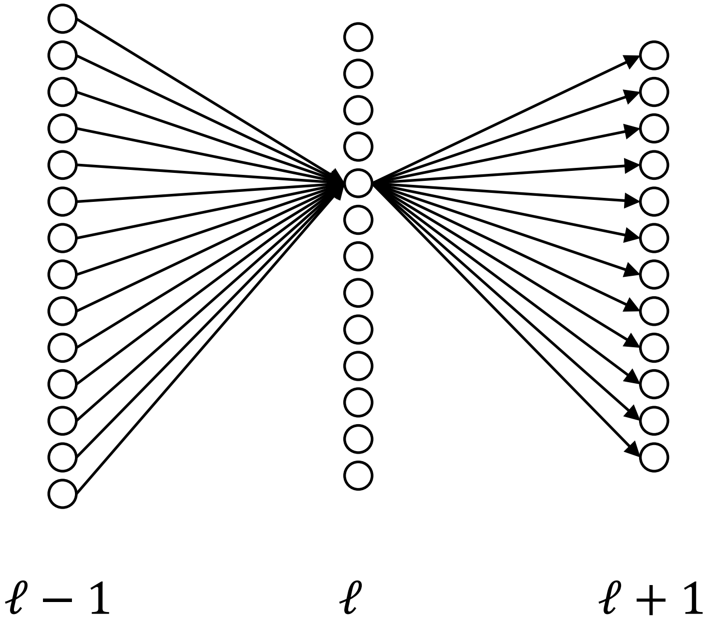


Lucas Liebenwein, lucasl@mit.edu

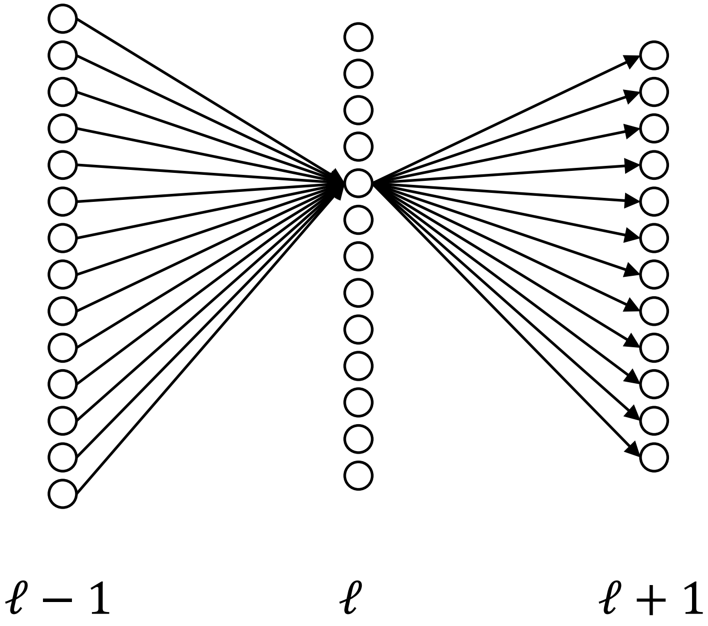
Prune pipeline



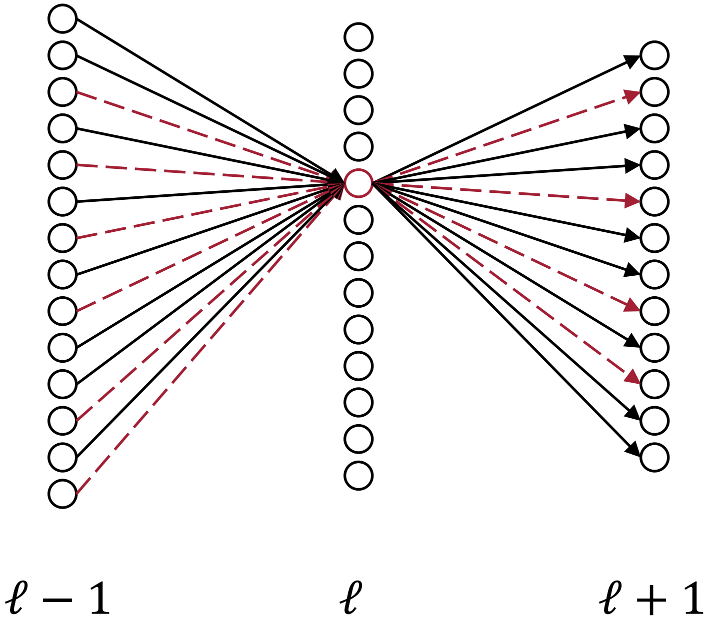
Weight pruning



Filter pruning

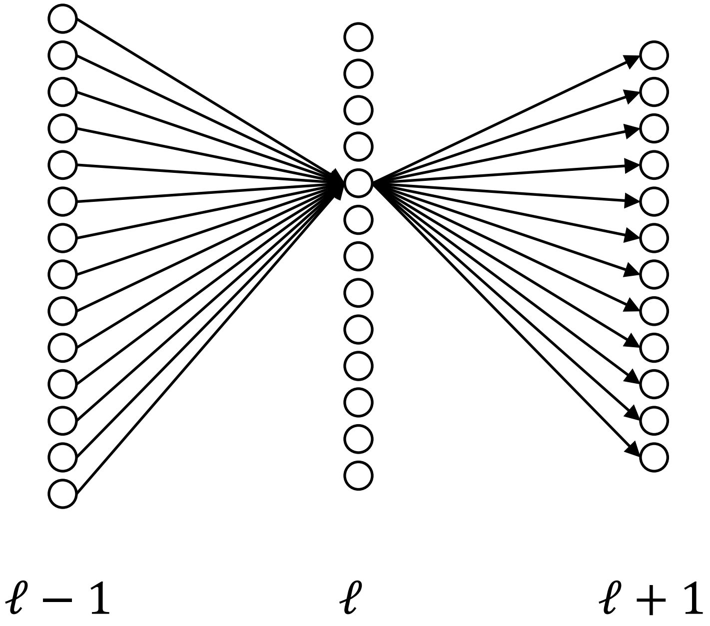


Weight pruning



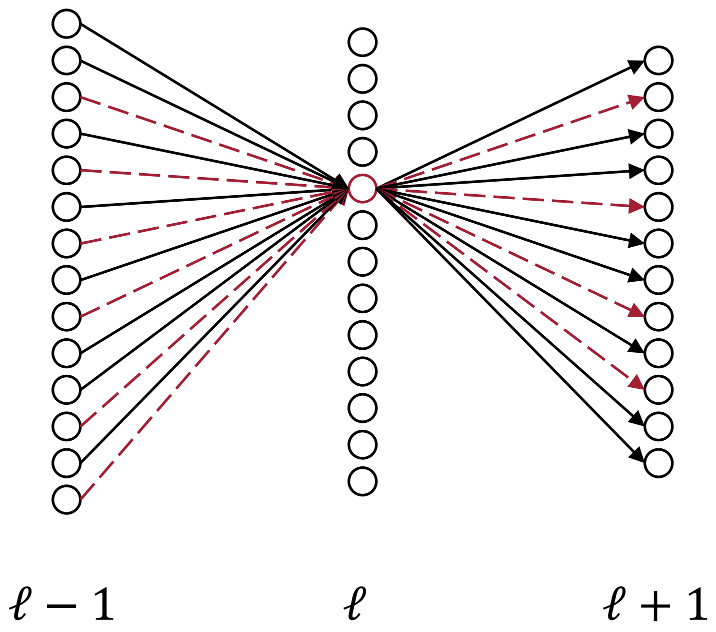
Remove individual edges

Filter pruning



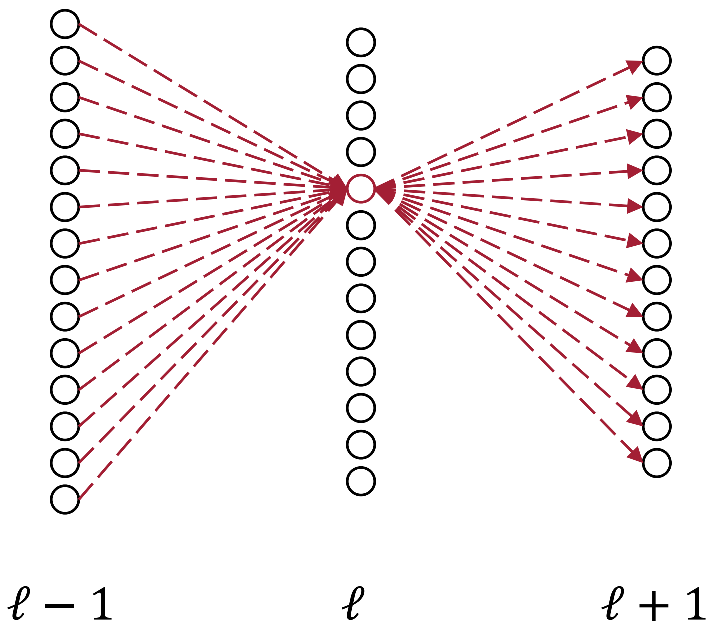
Lucas Liebenwein, lucasl@mit.edu

Weight pruning



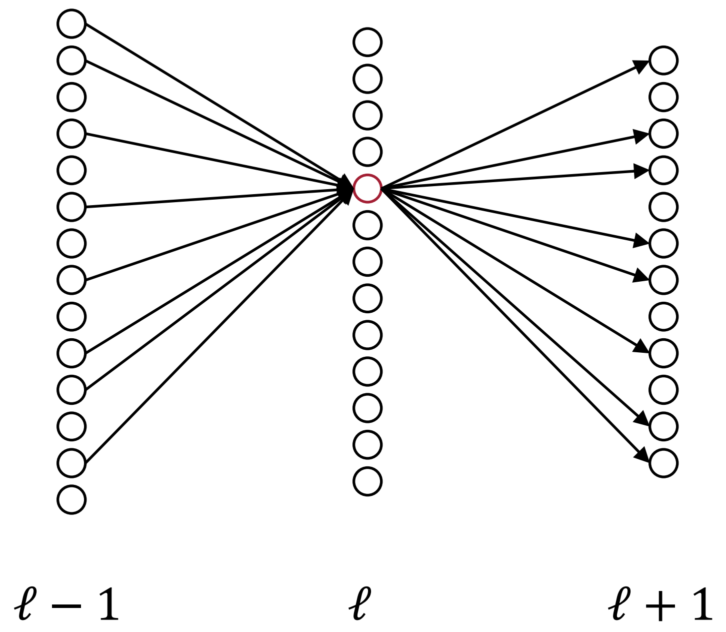
Remove individual edges

Filter pruning



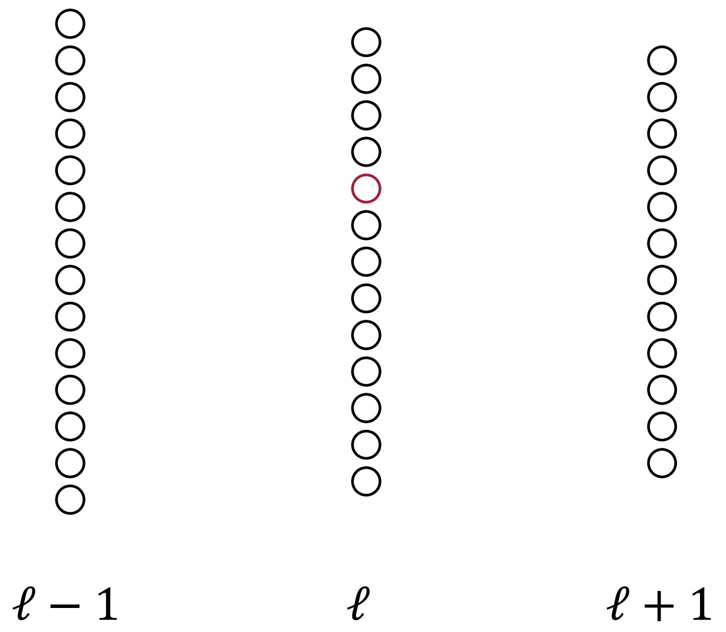
Remove neurons
(all incoming and outgoing edges)

Weight pruning



Remove individual edges

Filter pruning

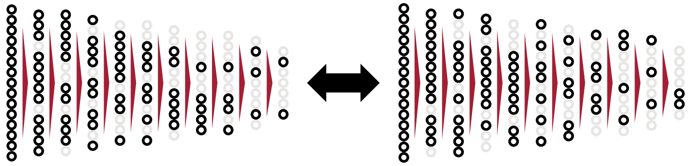
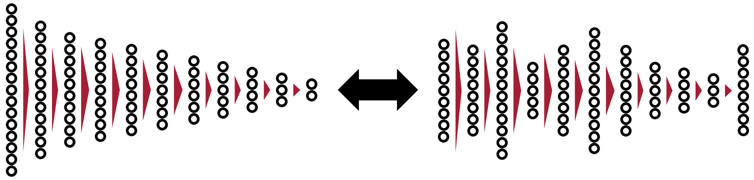


Remove neurons
(all incoming and outgoing edges)

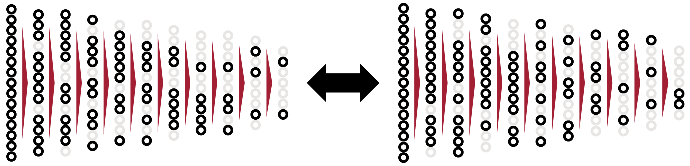
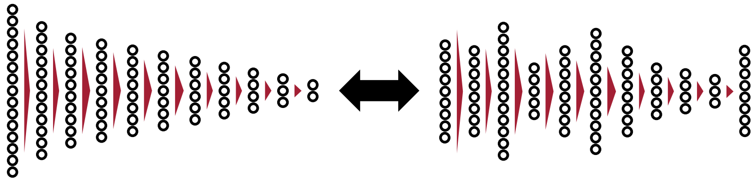
Prune methods

Unstructured (Weights)	
Structured (Filters/Neurons)	

Prune methods

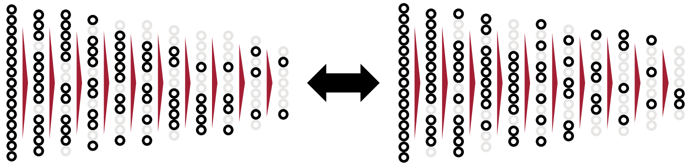
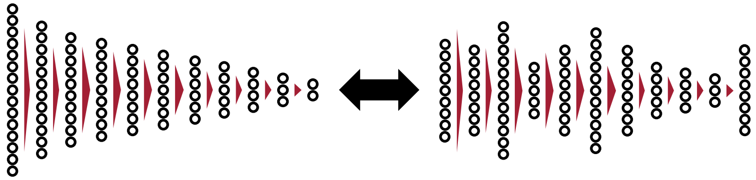
		 Relative importance within layer	 Budget allocation across layers
Unstructured (Weights)			
Structured (Filters/Neurons)			

Prune methods

		 Relative importance within layer	 Budget allocation across layers
Unstructured (Weights)	WT: Weight Thresholding (Renda et al. 2020)	$\propto \lvert W_{ij} \rvert$	Global pruning
	SiPP: Sensitivity Pruning (Baykal et al. 2019)	$\propto \lvert W_{ij} a_j(x) \rvert$	Global pruning
Structured (Filters/Neurons)			

i…filter, j…channel, $W^{\ell}_{i:}$…filter weights, $W^{\ell}_{:j}$…channel weights, x…input, $a(x)$…activation (layer input)

Prune methods

		 Relative importance within layer	 Budget allocation across layers
Unstructured (Weights)	WT: Weight Thresholding (Renda et al. 2020)	$\propto \lvert W_{ij} \rvert$	Global pruning
	SiPP: Sensitivity Pruning (Baykal et al. 2019)	$\propto \lvert W_{ij} a_j(x) \rvert$	Global pruning
Structured (Filters/Neurons)	FT: Filter Thresholding (Renda et al. 2020)	$\propto \lVert W_{i:} \rVert_1$	Manual (same percentage per layer)
	PFP: Provable Filter Pruning (Liebenwein et al. 2020)	$\propto \lVert W_{:j} a(x) \rVert_\infty$	Automatic via error analysis

i...filter, j...channel, $W^{\ell}_{i:}$...filter weights, $W^{\ell}_{:j}$...channel weights, x...input, $a(x)$...activation (layer input)

Prune results – how much can be pruned?

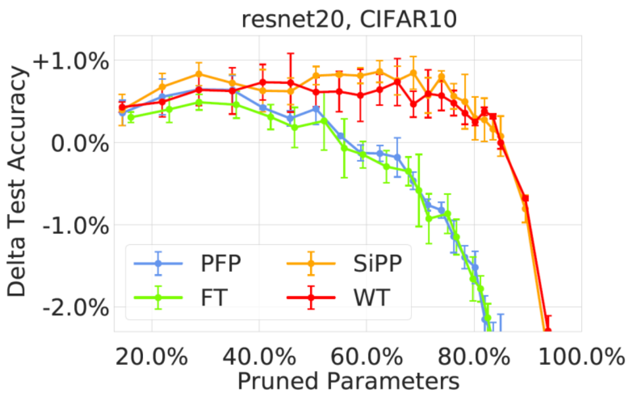
WT (Renda et al. 2020)

SiPP (Baykal et al. 2019)

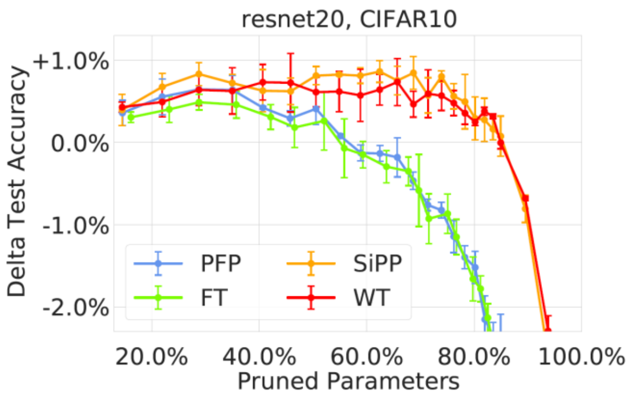
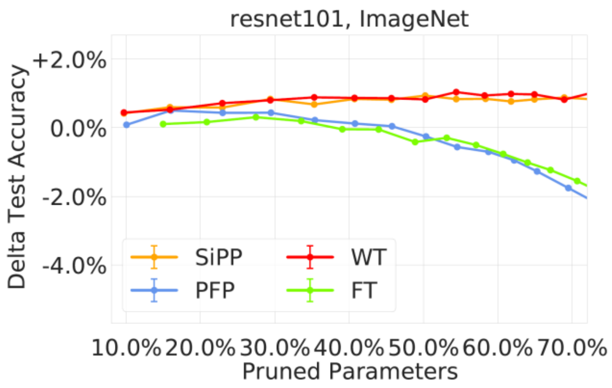
FT (Renda et al. 2020)

PFP (Liebenwein et al. 2020)

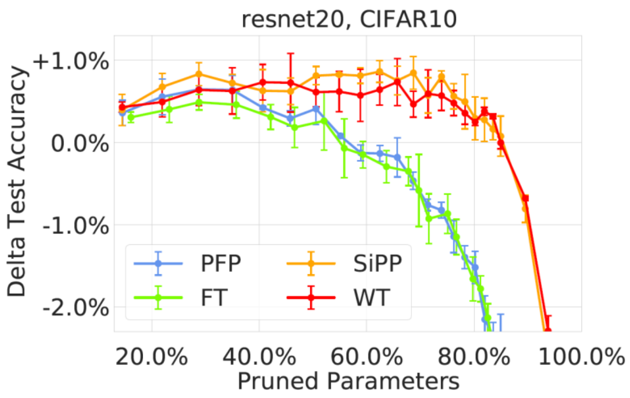
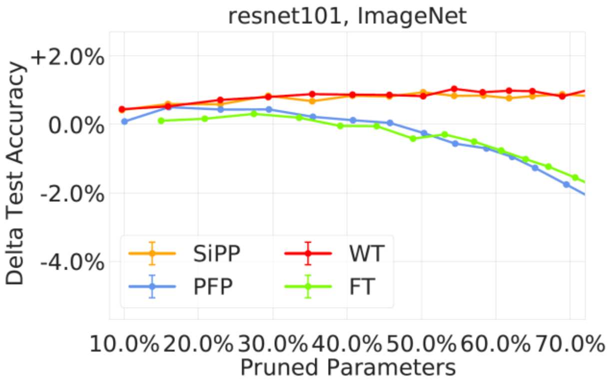
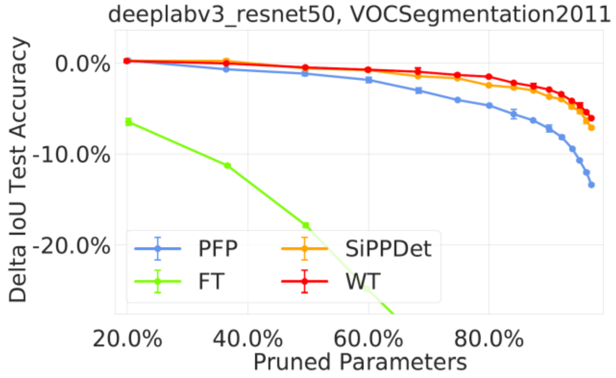
Prune results – how much can be pruned?

	ResNet20, CIFAR10
	
WT (Renda et al. 2020)	84.9%
SiPP (Baykal et al. 2019)	84.9%
FT (Renda et al. 2020)	52.1%
PFP (Liebenwein et al. 2020)	44.9%

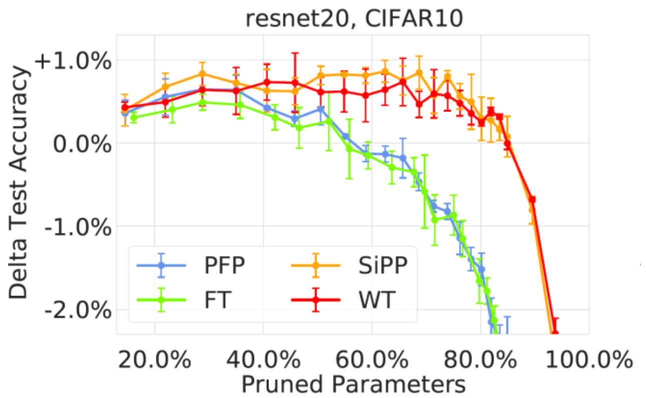
Prune results – how much can be pruned?

	ResNet20, CIFAR10	ResNet101, ImageNet
		
WT (Renda et al. 2020)	84.9%	81.6%
SiPP (Baykal et al. 2019)	84.9%	81.6%
FT (Renda et al. 2020)	52.1%	53.1%
PFP (Liebenwein et al. 2020)	44.9%	50.3%

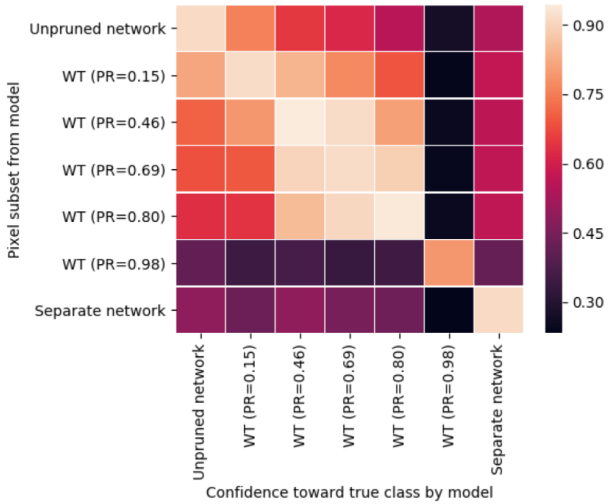
Prune results – how much can be pruned?

	ResNet20, CIFAR10	ResNet101, ImageNet	DeeplabV3, Pascal VOC2011
			
WT (Renda et al. 2020)	84.9%	81.6%	58.9%
SiPP (Baykal et al. 2019)	84.9%	81.6%	42.3%
FT (Renda et al. 2020)	52.1%	53.1%	0.0%
PFP (Liebenwein et al. 2020)	44.9%	50.3%	20.2%

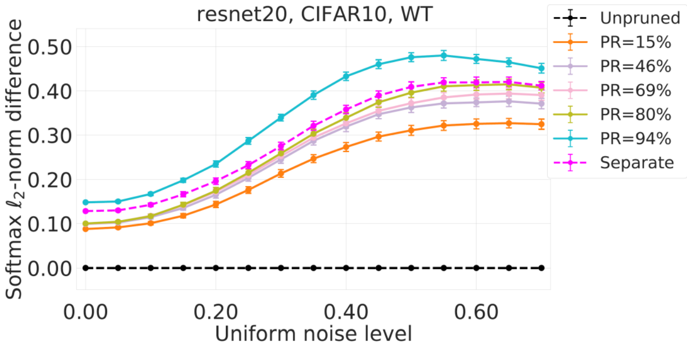
What is preserved during pruning?



Test accuracy



Informative features



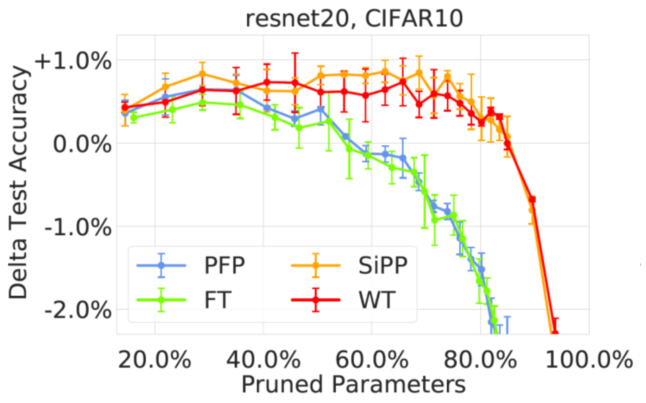
Functional similarities

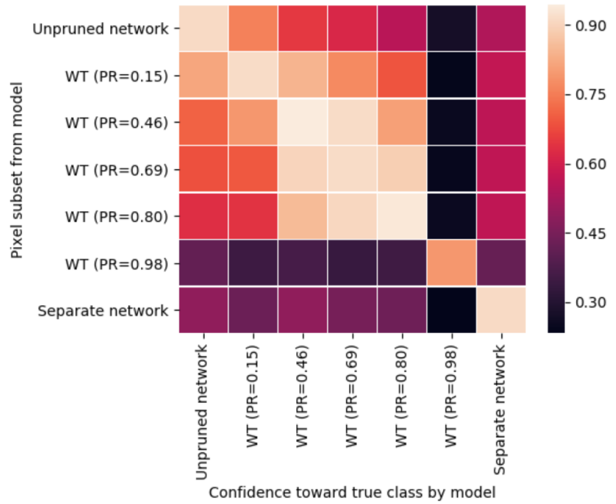
?

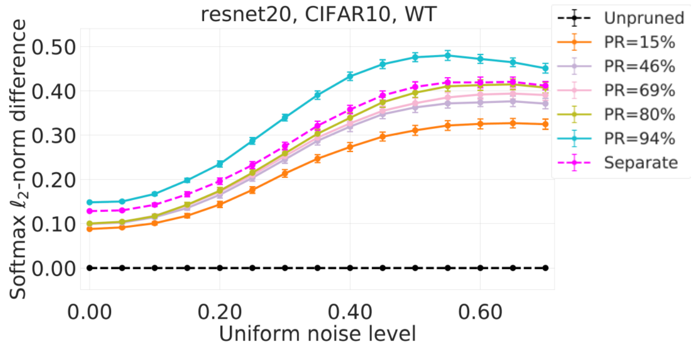
?

?

What is preserved during pruning?







Test accuracy

Informative features

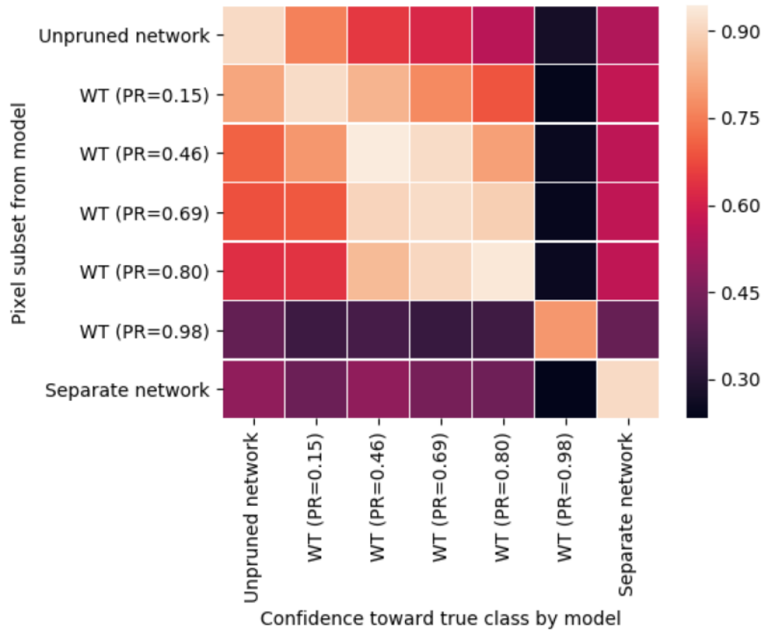
Functional similarities

Informative features

For a given network f_θ and data distribution $\mathbf{x} \sim \mathcal{D}$ the 10% of *most informative features* are defined as

$$m(\theta, \mathcal{D}) := \text{"}the\ mask\ of\ input\ features\ that\ change\ the\ output\ of\ the\ network\ the\ least\text{"}$$

- Captures the parts of the input that drive the network's decision
- Can be compared between pruned and unpruned networks

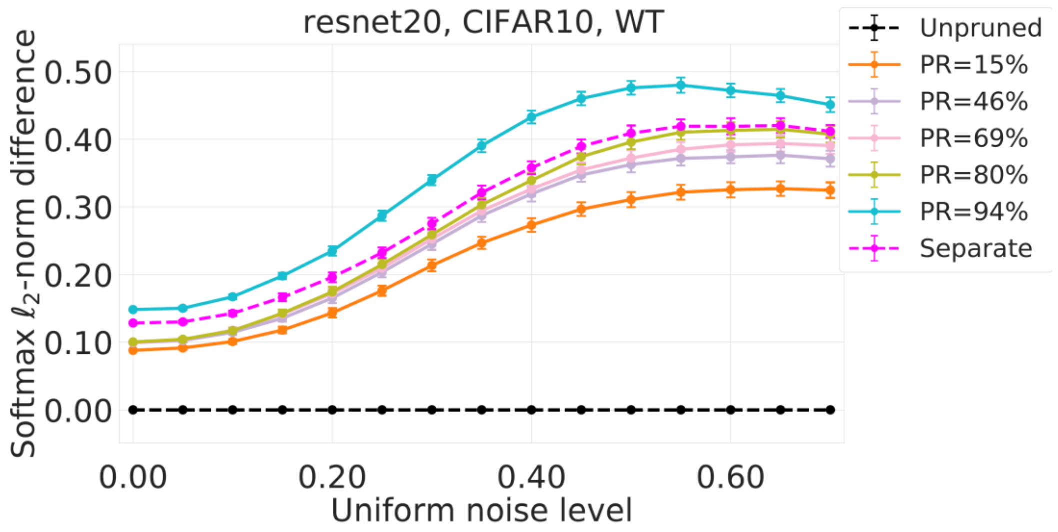


Functional (noise) similarity

For a given network f_θ and data distribution $\mathbf{x} \sim \mathcal{D}$ noise similarity measures

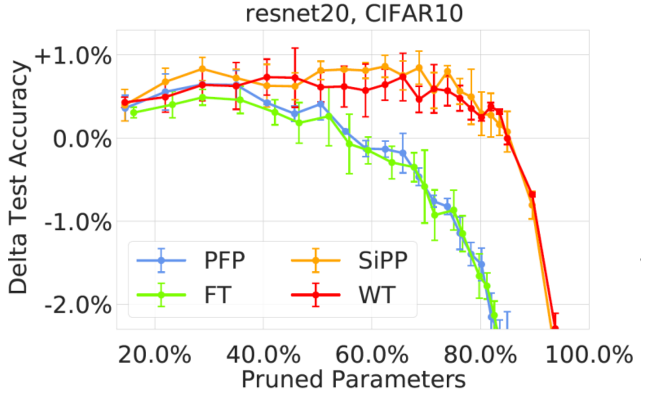
$$"how\ much\ the\ output\ of\ the\ pruned\ and\ unpruned\ network\ differ\ under\ noisy\ input"$$

- Captures Local Lipschitz behavior of networks
- Direct comparison between pruned and unpruned networks

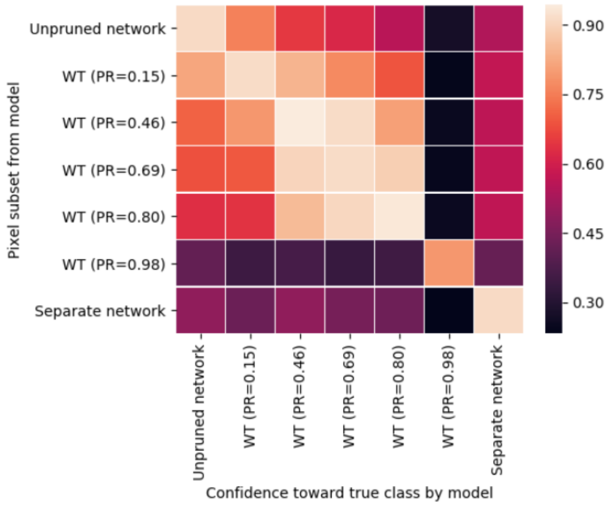


Lucas Liebenwein, lucasl@mit.edu

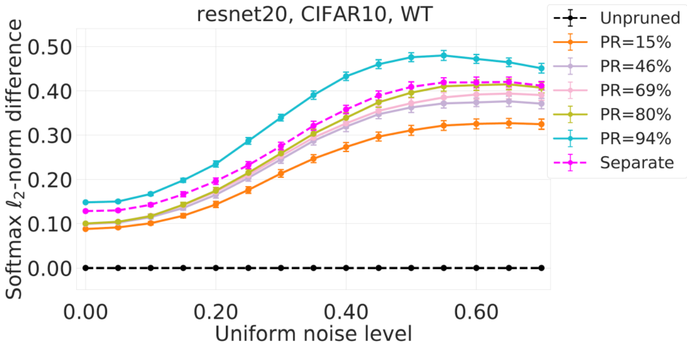
What is preserved during pruning?



Test accuracy

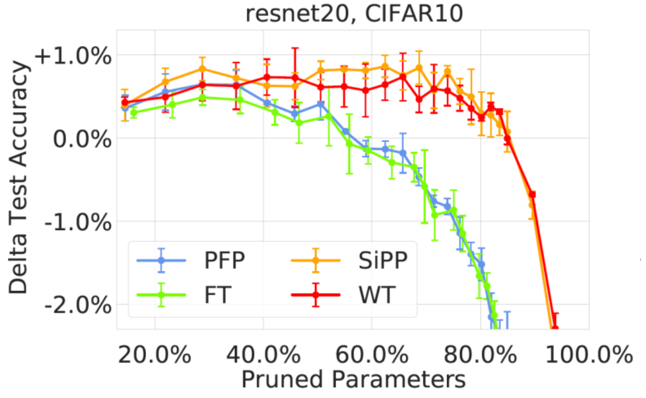


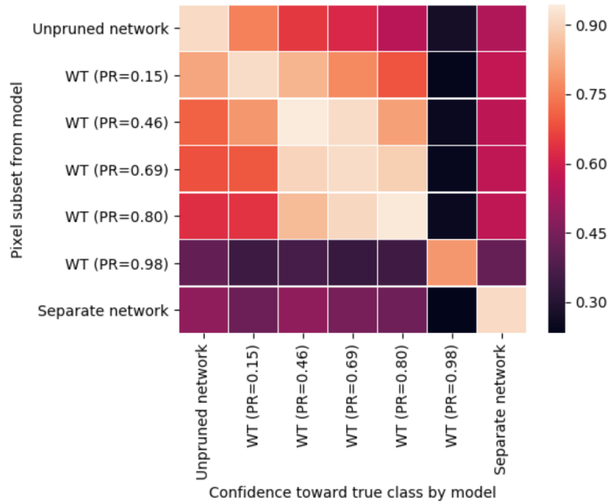
Informative features

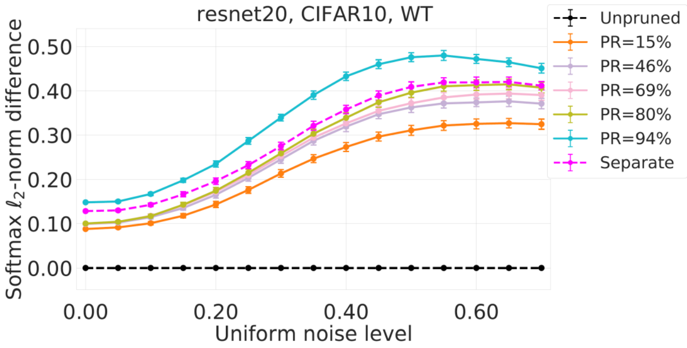


Functional similarities

What is preserved during pruning?





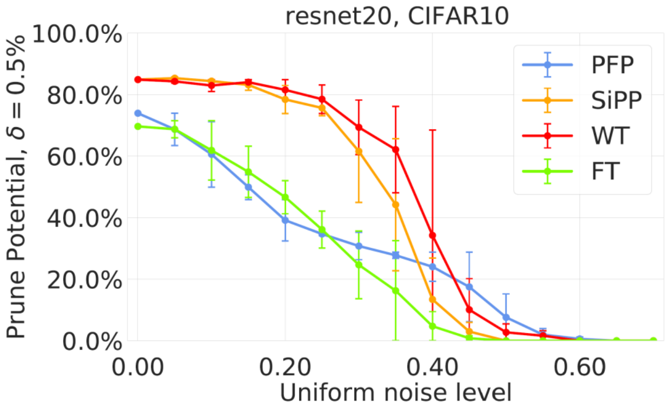


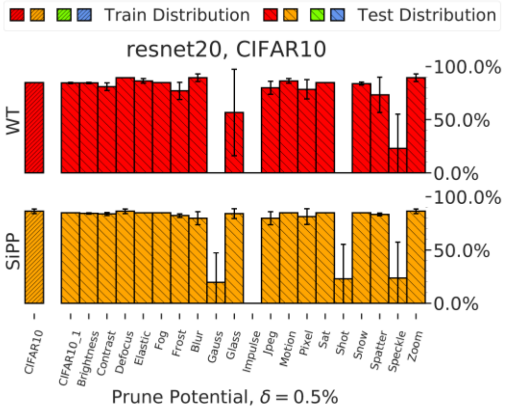
Test accuracy

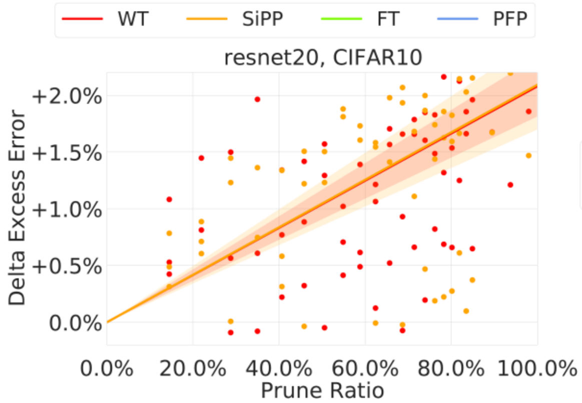
Informative features

Functional similarities

What is *lost* during pruning?







Prune potential for noise

?

Prune potential for corruptions

?

Adequate excess error

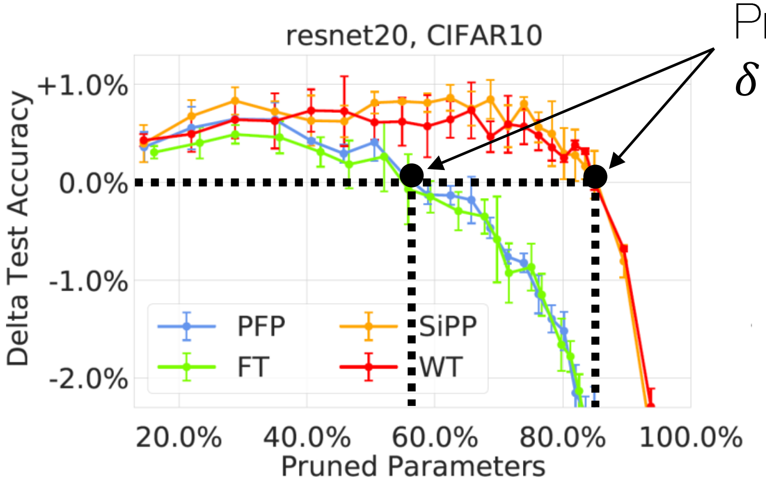
?

Prune potential

For $\delta \in [0,1)$, given network f_θ, and data distribution $\mathcal{D}$ the prune potential $P(\theta, \mathcal{D})$ is defined as

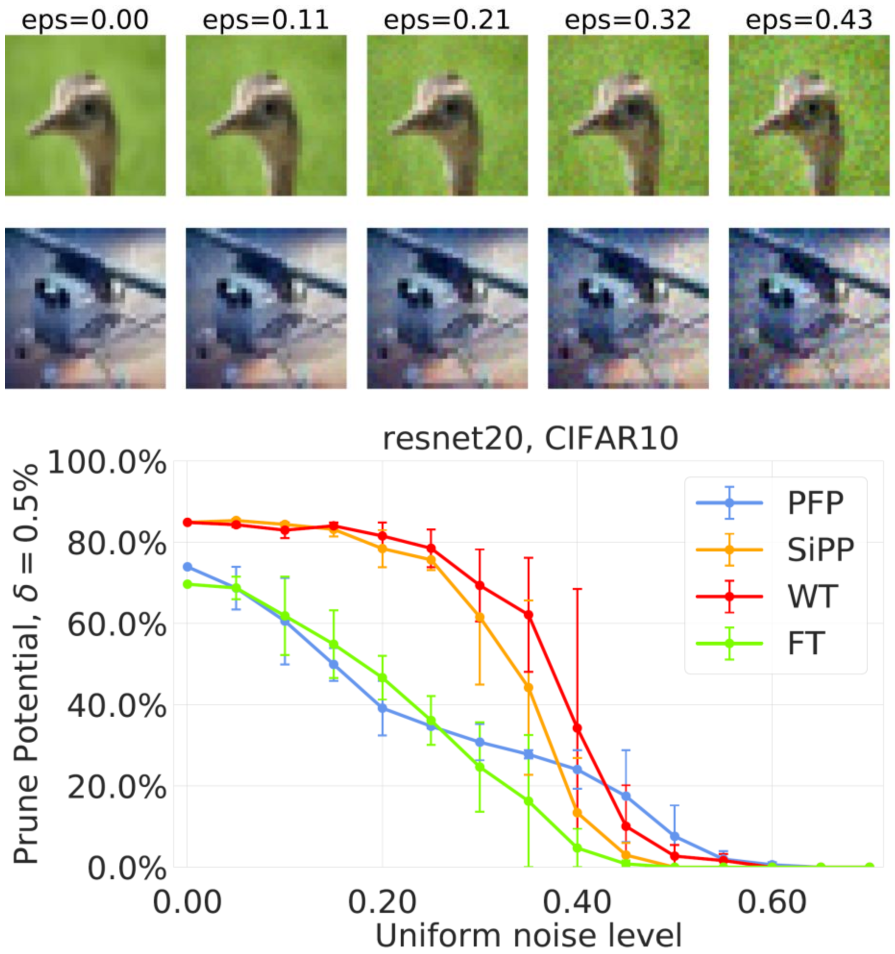
$$P(\theta, \mathcal{D}) \coloneqq \text{"}\mathit{maximum\ prune\ ratio\ with\ accuracy\ loss\ at\ most\ } \delta\text{"}$$

- Quantifies "overparameterization" of network
- Approximated using many prune-retrain cycles



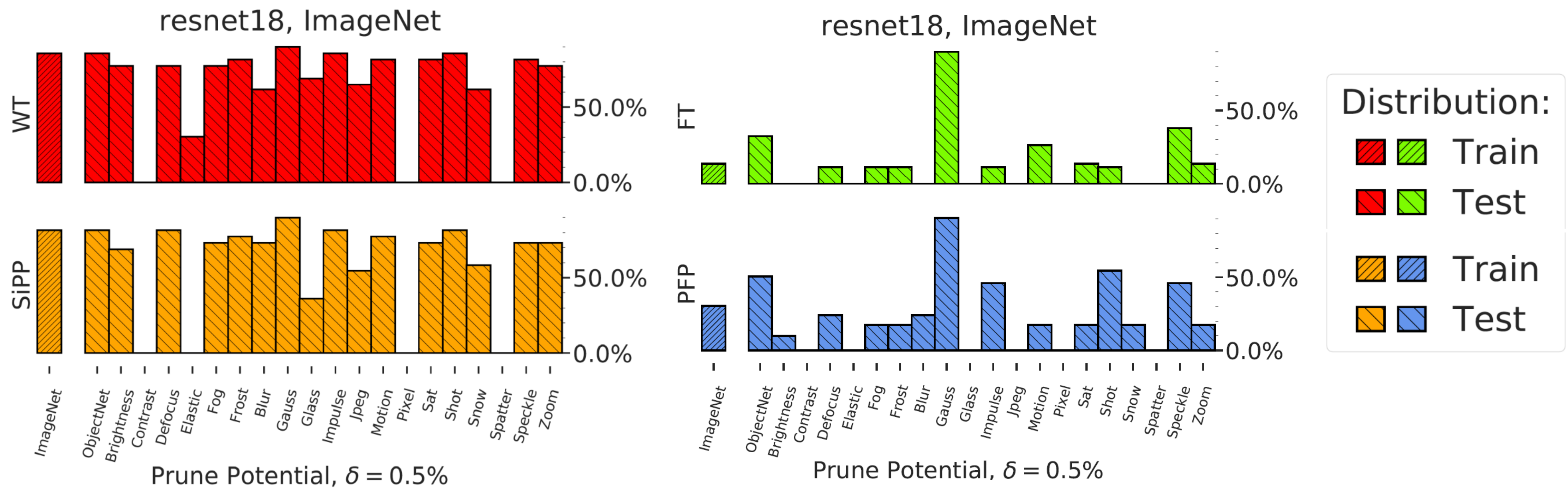
Prune potential for $\delta = 0$

Prune potential for noise



- Prune potential after small noise injections in input
- Pruned networks are more affected by out-of-distribution
- “Robust overparameterization” vs. “nominal overparameterization”

Prune potential for corruptions



Significantly reduced prune potential under distribution changes (!)

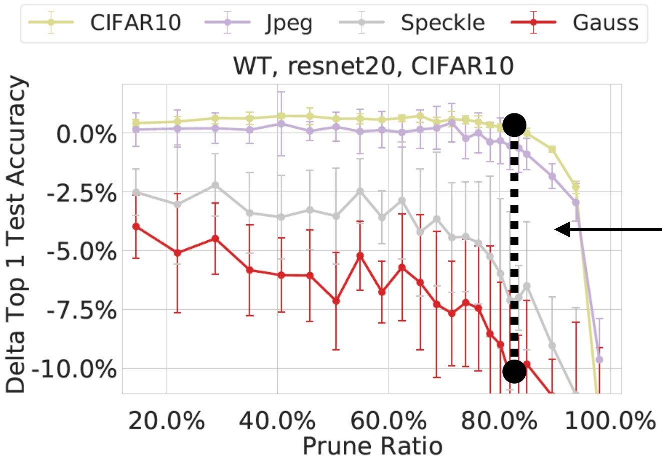
Lucas Liebenwein, lucasl@mit.edu

Excess error

For a network f_θ, train distribution $\mathcal{D}$, and *test distribution* $\mathcal{D}'$ the excess error is defined as

$$e(\theta, \mathcal{D}') \coloneqq \text{"}\textit{additional error incurred on test distribution}\text{"}$$

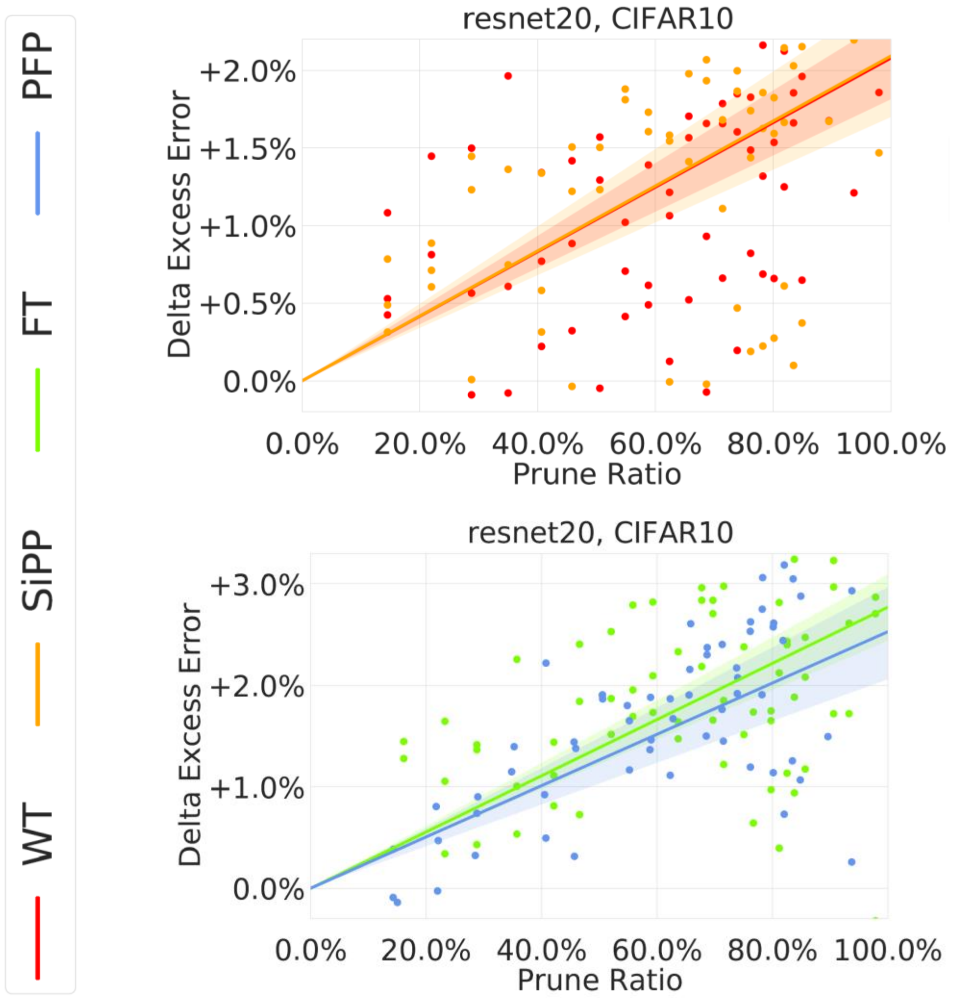
- Quantifies performance drop for distribution changes
- Difference in excess error for pruned and unpruned network indicates performance drop



Excess error for "Gauss" test distribution and 80% prune ratio

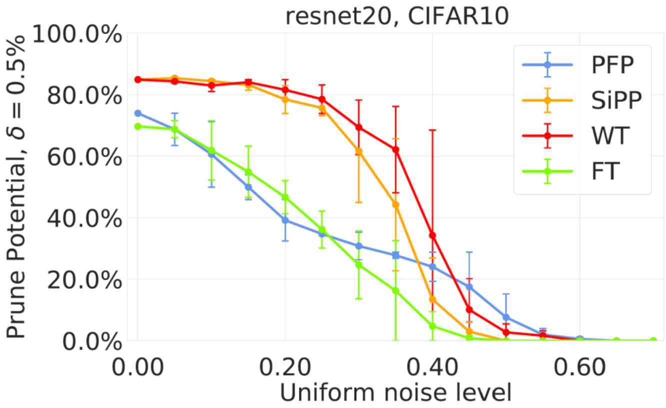
Lucas Liebenwein, lucasl@mit.edu

Excess error

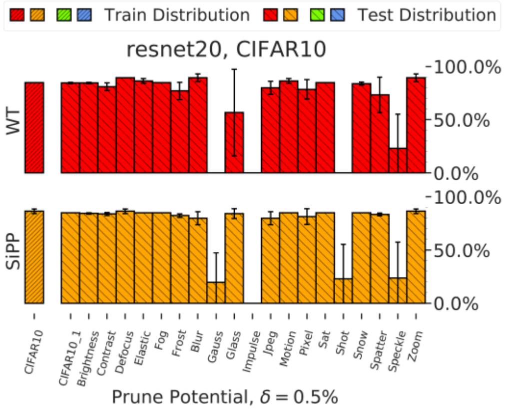


- Excess error averaged over multiple corruptions
- Pruned networks exhibit higher excess error
- Nominal prune curve is not indicative of out-of-distribution performance!

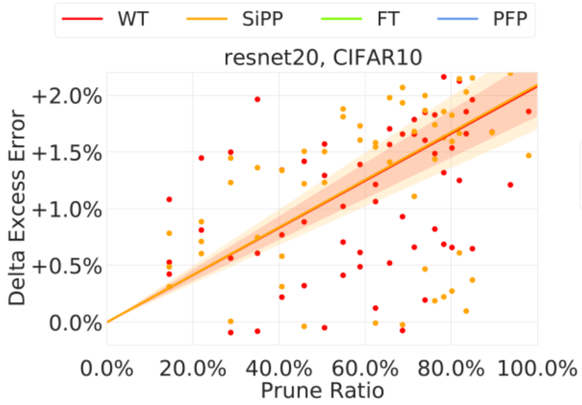
What is *lost* during pruning?



Prune potential for noise



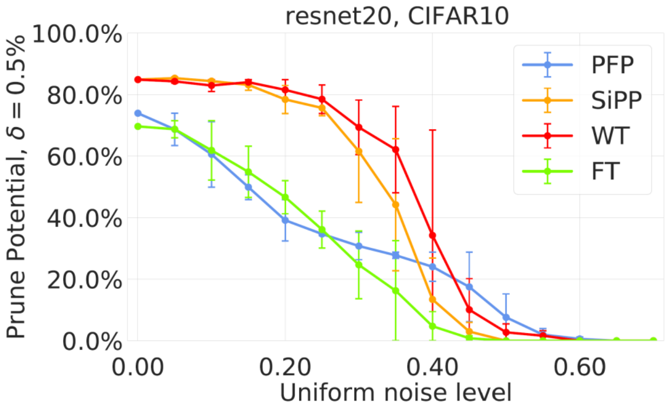
Prune potential for corruptions



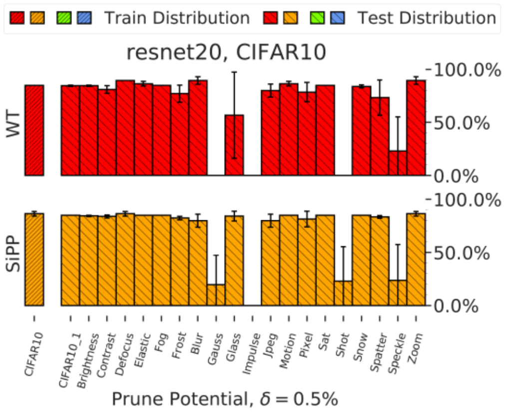
Adequate excess error

Lucas Liebenwein, lucasl@mit.edu

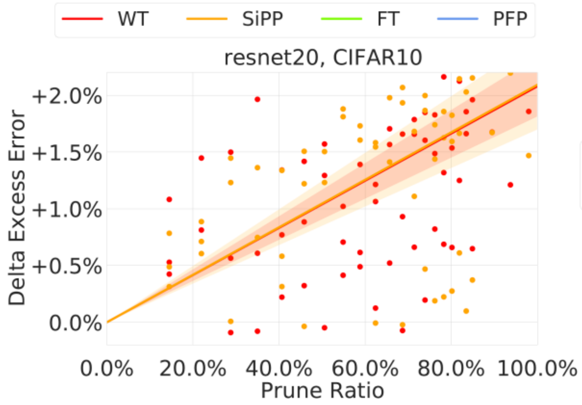
What is *lost* during pruning?



Prune potential
for noise

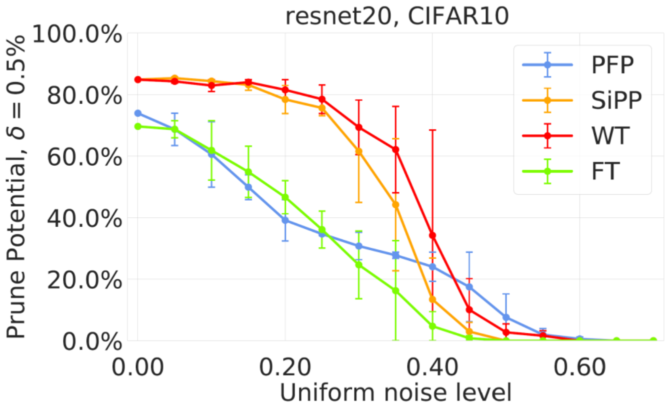


Prune potential
for corruptions



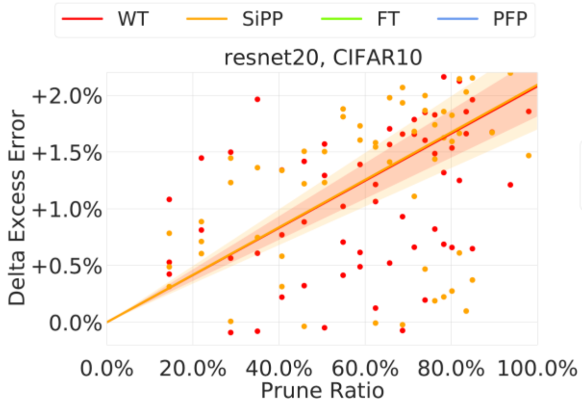
Adequate excess error

What is *lost* during pruning?



Prune potential for noise

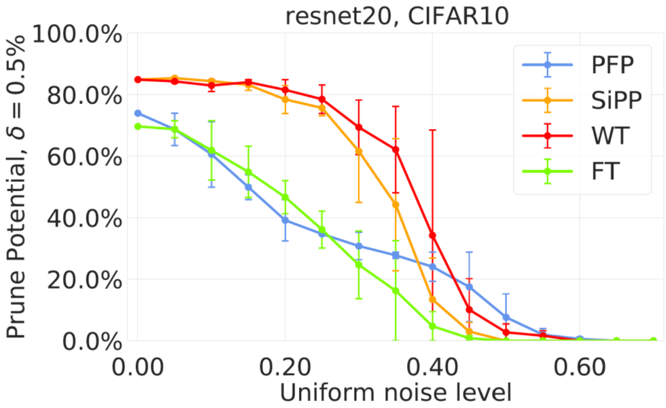
Prune potential for corruptions



Adequate excess error

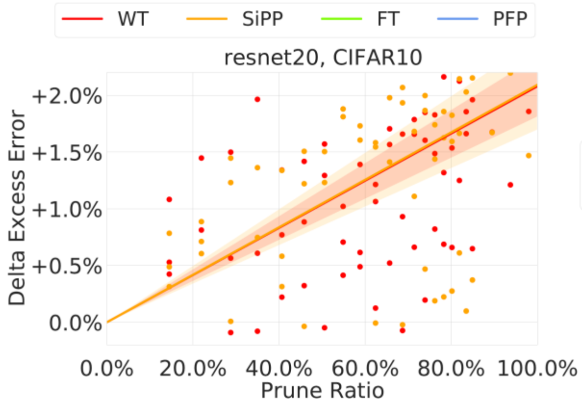
?

What is *lost* during pruning?



Prune potential for noise

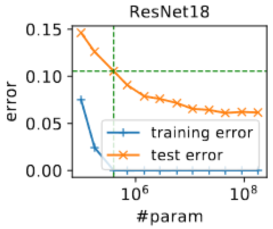
Prune potential for corruptions



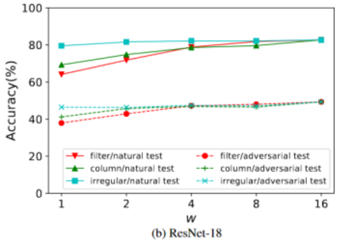
Adequate excess error

Related work and discussion

Nominal vs. robust overparameterization

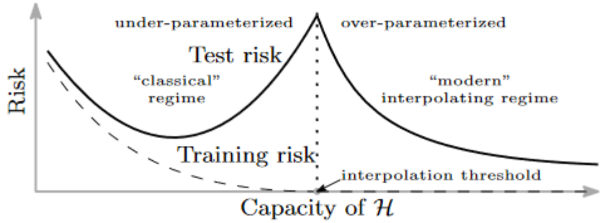


Neyshabur et al. 2019

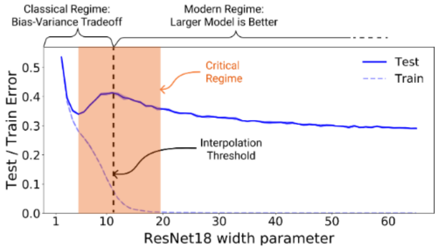


Ye et al. 2019

Implicit regularization via overparameterization

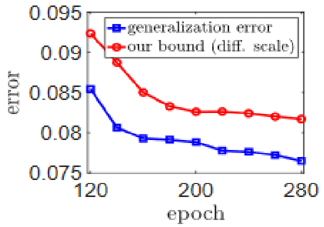


Belkin et al. 2019

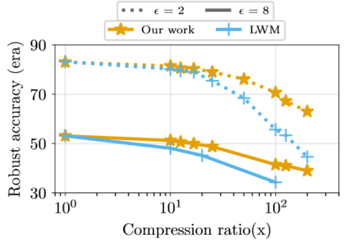


Nakkiran et al. 2020

Generalization-aware pruning



Arora et al. 2018



Sehwag et al. 2020

Guidelines for pruning in practice

1. Don’t prune if unexpected shifts in data may occur
2. Prune moderately if you can account for some data shifts during training
3. Prune fully if you can account for all data shifts during training
4. Maximize prune potential by considering data shifts during training

Lucas Liebenwein, lucasl@mit.edu

Lost in Pruning: The Effects of Pruning Neural Networks beyond Test Accuracy

Lucas Liebenwein, lucasl@mit.edu, http://www.mit.edu/~lucasl/

Distributed Robotics Lab, CSAIL, MIT

- Paper: https://arxiv.org/abs/2103.03014
- Code: https://github.com/lucaslie/torchprune
- Contact: lucasl@mit.edu

Collaborators (Thank you!)

Cenk Baykal

Brandon Carter

David Gifford

Daniela Rus