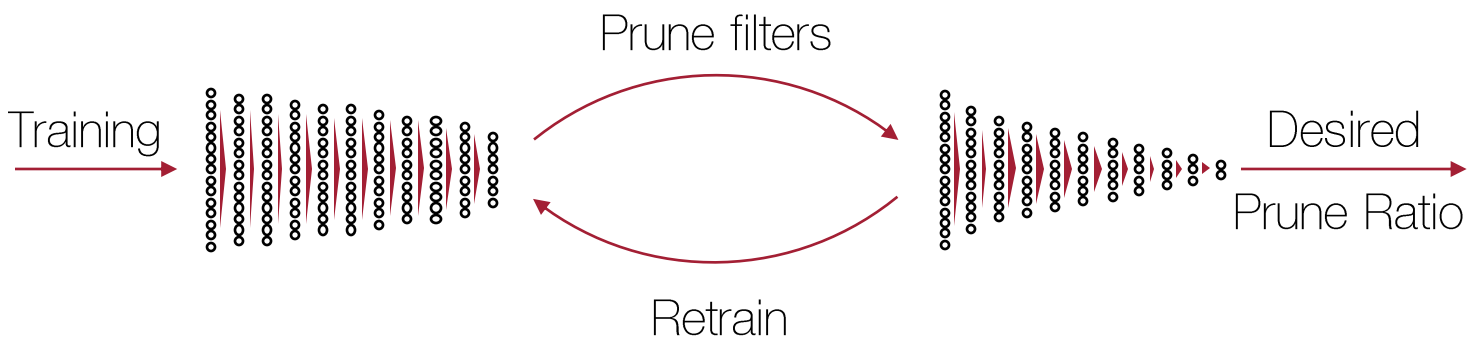
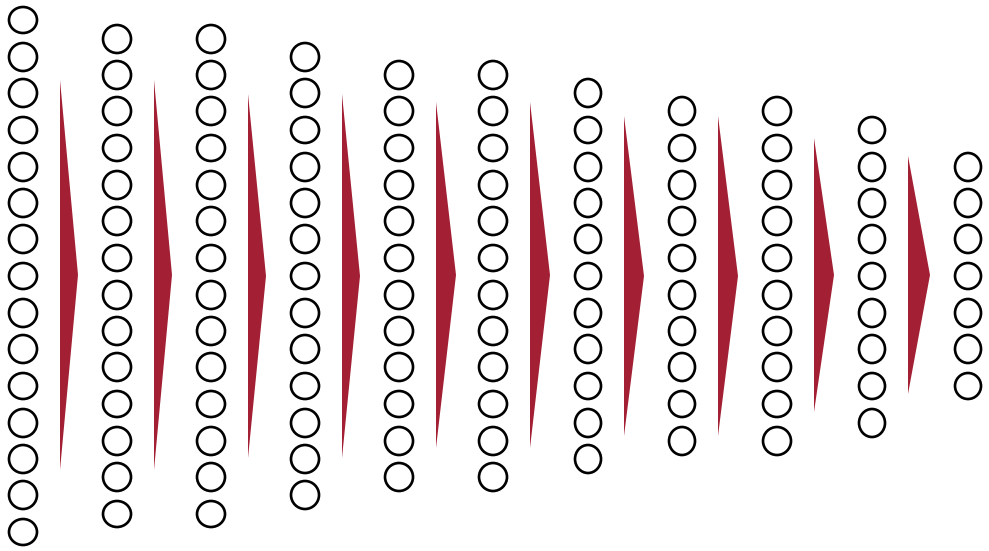


Provable Filter Pruning for Efficient Neural Networks

Lucas Liebenwein*, Cenk Baykal*, Harry Lang, Dan Feldman, Daniela Rus
Distributed Robotics Lab, CSAIL, MIT

Neural networks are SOTA



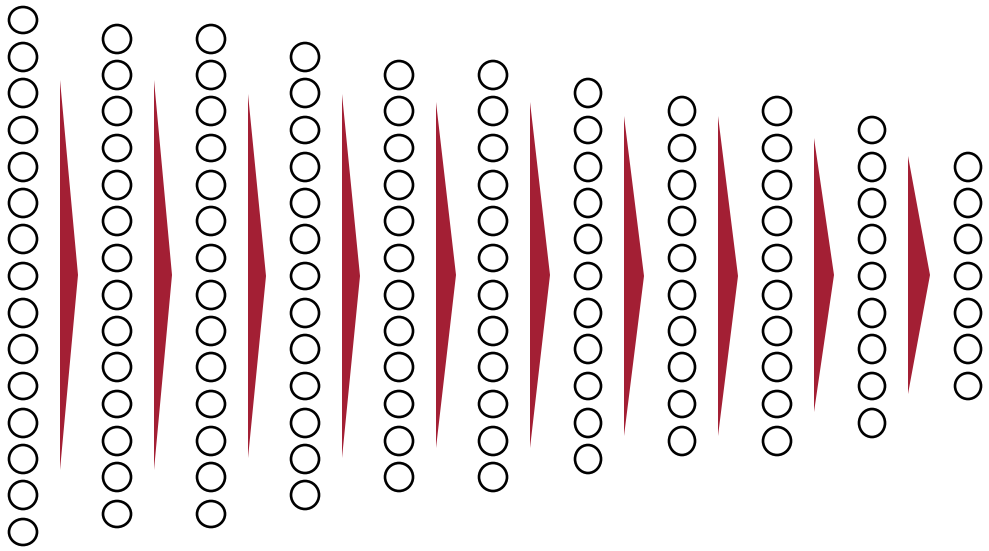
Natural Language Processing

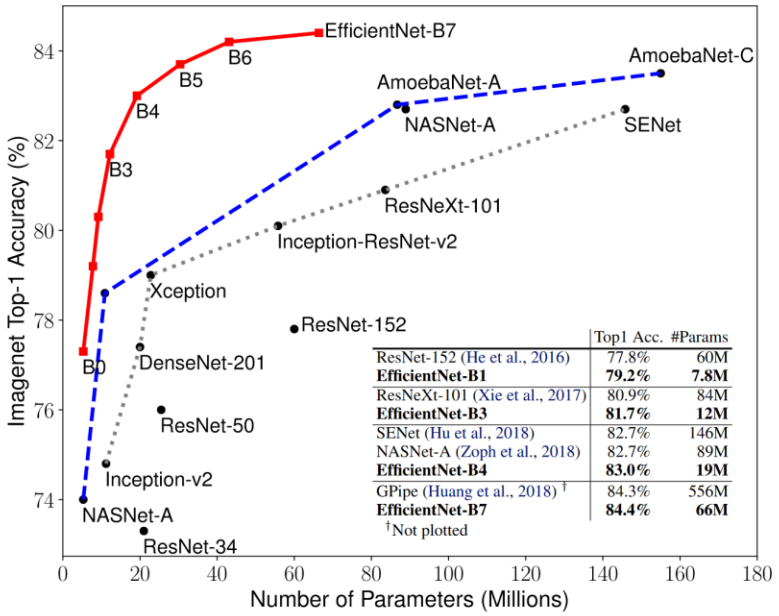
Computer Vision

Robotics

lucasl@mit.edu, baykal@mit.edu

Larger size, better performance



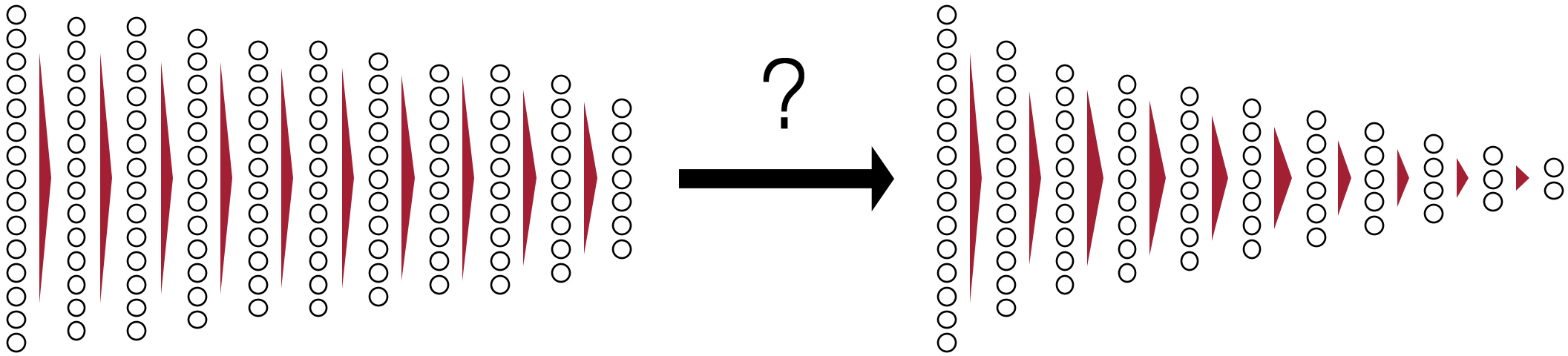


	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.2%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.7%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**83.0%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

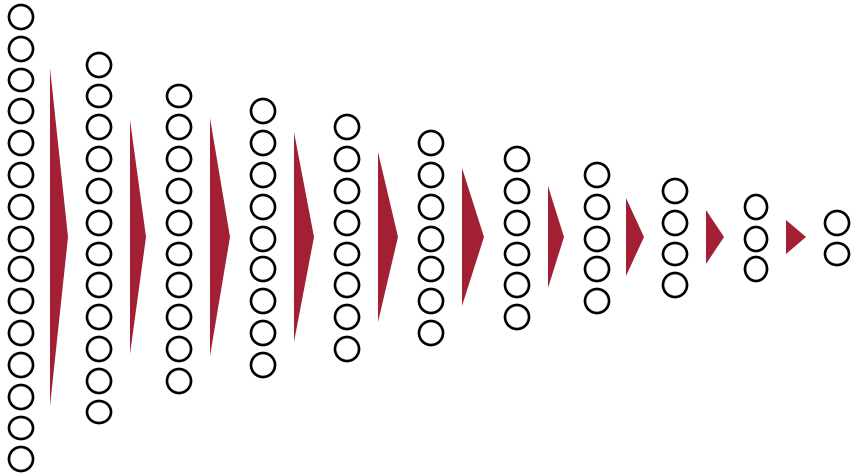
Tan, Mingxing, and Quoc Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." ICML. 2019.

Smaller size, same performance?



lucasl@mit.edu, baykal@mit.edu

Smaller size, same performance?



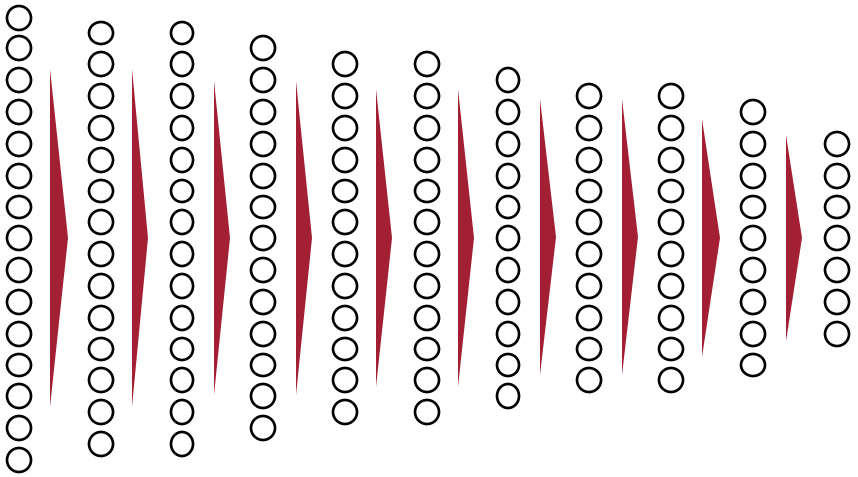
- Less storage requirements
- Faster inference time
- Improved Interpretability

lucasl@mit.edu, baykal@mit.edu

Objective: Filter Pruning

For a given network $f_\theta(x)$ with parameters θ and $\epsilon, \delta \in (0,1)$, generate a compressed, *dense* reparameterization $\hat{\theta}$ (i.e. with less filters) such that

$$\mathbb{P}_{x\sim\mathcal{D}}\left(f_{\hat{\theta}}(x) \in (1 \pm \epsilon) f_\theta(x)\right) \geq (1-\delta) \text{ and } size(\hat{\theta}) \ll size(\theta).$$

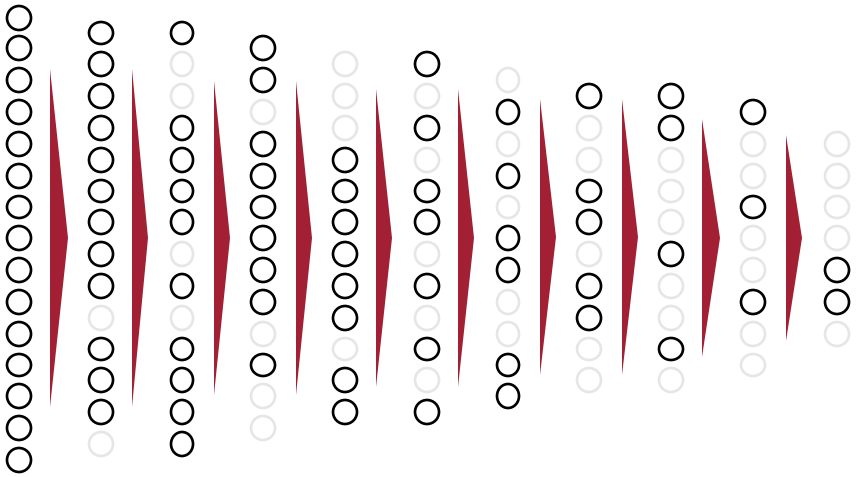


lucasl@mit.edu, baykal@mit.edu

Objective: Filter Pruning

For a given network $f_\theta(x)$ with parameters θ and $\epsilon, \delta \in (0,1)$, generate a compressed, *dense* reparameterization $\hat{\theta}$ (i.e. with less filters) such that

$$\mathbb{P}_{x\sim\mathcal{D}}\left(f_{\hat{\theta}}(x) \in (1\pm\epsilon) f_\theta(x)\right) \geq (1-\delta) \text{ and } size(\hat{\theta}) \ll size(\theta).$$



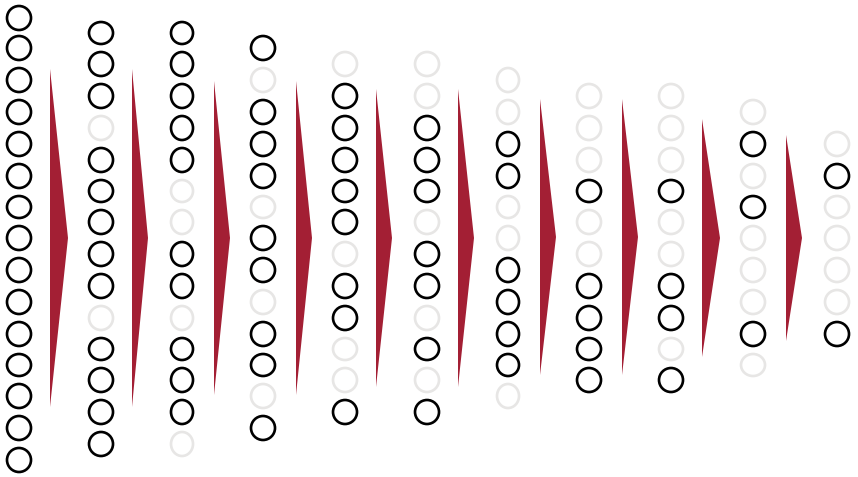
Relative importance s_j^ℓ of filter j in layer ℓ

lucasl@mit.edu, baykal@mit.edu

Objective: Filter Pruning

For a given network $f_\theta(x)$ with parameters θ and $\epsilon, \delta \in (0,1)$, generate a compressed, *dense* reparameterization $\hat{\theta}$ (i.e. with less filters) such that

$$\mathbb{P}_{x\sim\mathcal{D}}\left(f_{\hat{\theta}}(x) \in (1\pm\epsilon) f_\theta(x)\right) \geq (1-\delta) \text{ and } size(\hat{\theta}) \ll size(\theta).$$



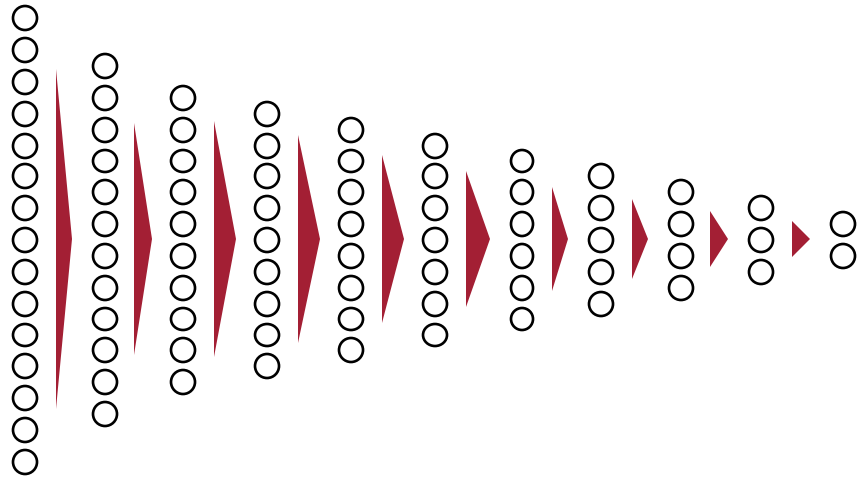
Relative importance s_j^ℓ of filter j in layer ℓ

lucasl@mit.edu, baykal@mit.edu

Objective: Filter Pruning

For a given network $f_\theta(x)$ with parameters θ and $\epsilon, \delta \in (0,1)$, generate a compressed, *dense* reparameterization $\hat{\theta}$ (i.e. with less filters) such that

$$\mathbb{P}_{x\sim\mathcal{D}}\left(f_{\hat{\theta}}(x) \in (1\pm\epsilon) f_\theta(x)\right) \geq (1-\delta) \text{ and } size(\hat{\theta}) \ll size(\theta).$$



Relative importance s_j^ℓ of filter j in layer ℓ

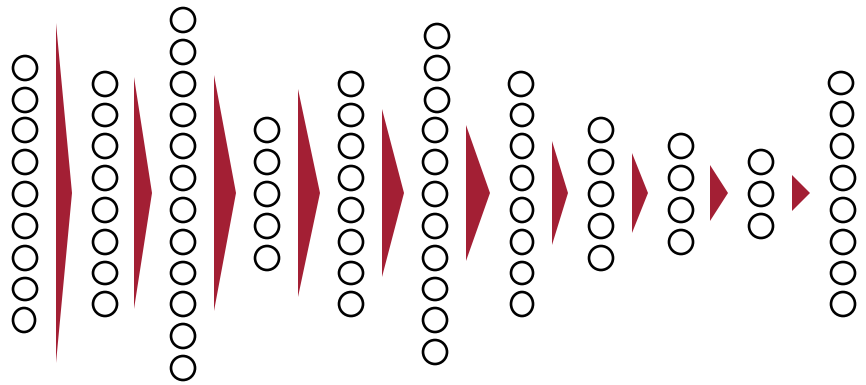
Budget allocation to layers $1, \ldots, \ell, \ldots, L$

lucasl@mit.edu, baykal@mit.edu

Objective: Filter Pruning

For a given network $f_\theta(x)$ with parameters θ and $\epsilon, \delta \in (0,1)$, generate a compressed, *dense* reparameterization $\hat{\theta}$ (i.e. with less filters) such that

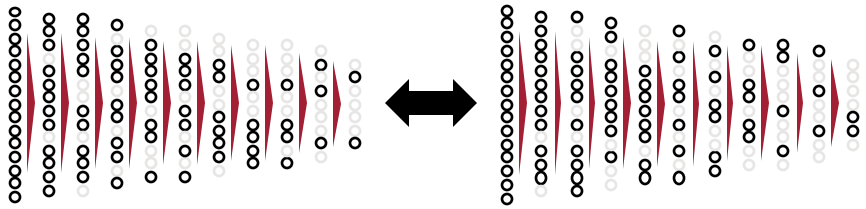
$$\mathbb{P}_{x\sim\mathcal{D}}\left(f_{\hat{\theta}}(x) \in (1\pm\epsilon) f_\theta(x)\right) \geq (1-\delta) \text{ and } size(\hat{\theta}) \ll size(\theta).$$



Relative importance s_j^ℓ of filter j in layer ℓ

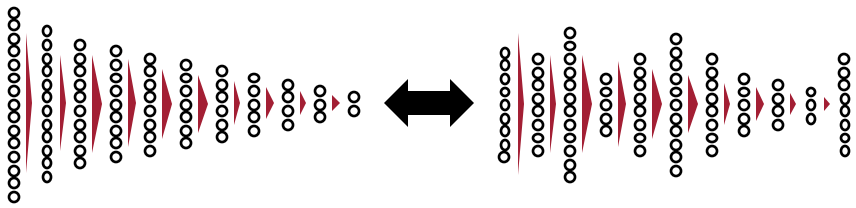
Budget allocation to layers $1, \ldots, \ell, \ldots, L$

Related Work



Relative importance s_j^ℓ of filter j in layer ℓ

- $s_j^\ell \sim \left\|W_j^\ell\right\|_2$, Li et al. 2016
- $s_j^\ell \sim \left\|W_j^\ell\right\|_1$, He et al., 2018
- $s_j^\ell \sim 1 \,/ \max_x \left| z^{\ell+1}(x) - z_{[j]}^{\ell+1}(x) \right|$, Luo et al., 2017

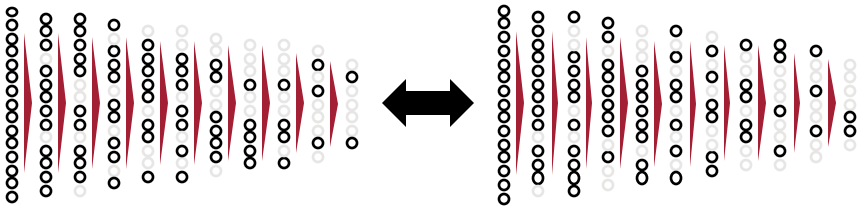


Budget allocation to layers $1, \ldots, \ell, \ldots, L$

- Uniform budget allocation, He et al., 2018
- Manual ablation study, Li et al. 2016
- Sequential pruning process, Luo et al. 2017

j…filter, ℓ…layer, s_j^ℓ…importance, W_j^ℓ…filter weights, x…input, $z^\ell(x)$…pre-activation, $z_{[j]}^\ell$…pre-activation with feature j only

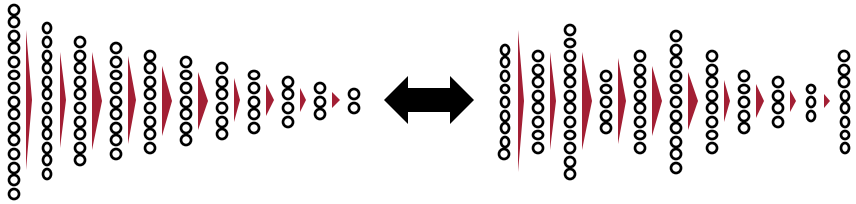
Related Work



Relative importance s_j^ℓ of filter j in layer ℓ

✘ no performance guarantees

✘ data-oblivious heuristics



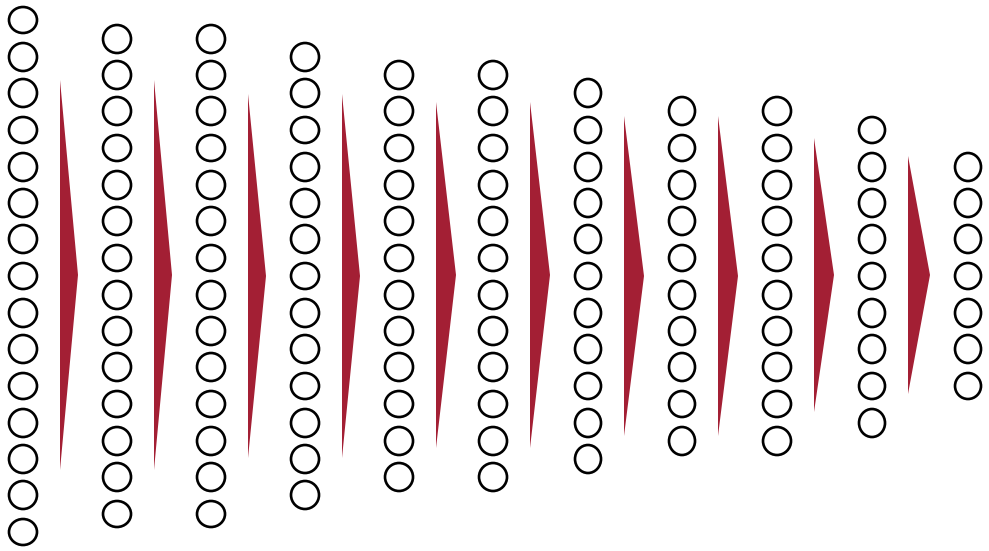
Budget allocation to layers $1, \ldots, \ell, \ldots, L$

✘ manual procedure

✘ architecture-specific

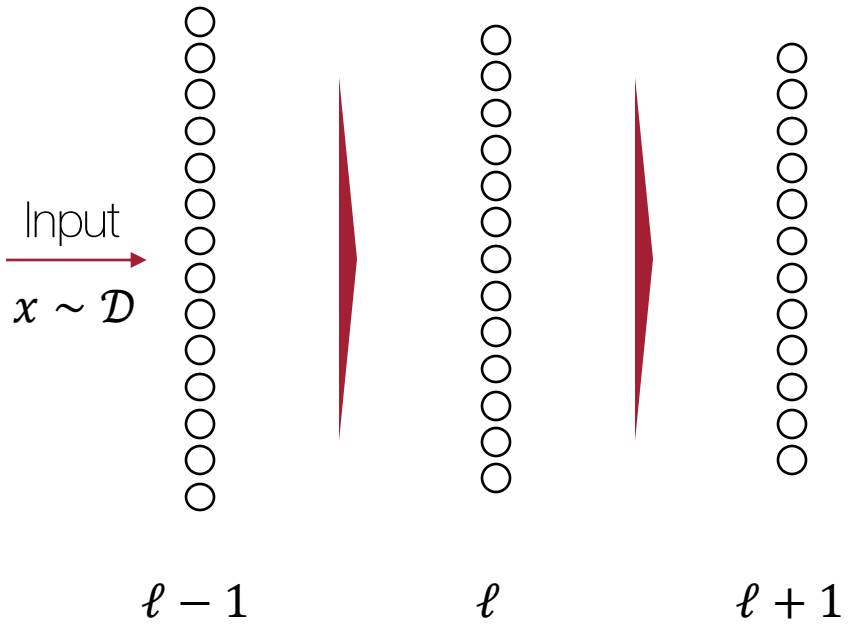
lucasl@mit.edu, baykal@mit.edu

Our method: filter importance



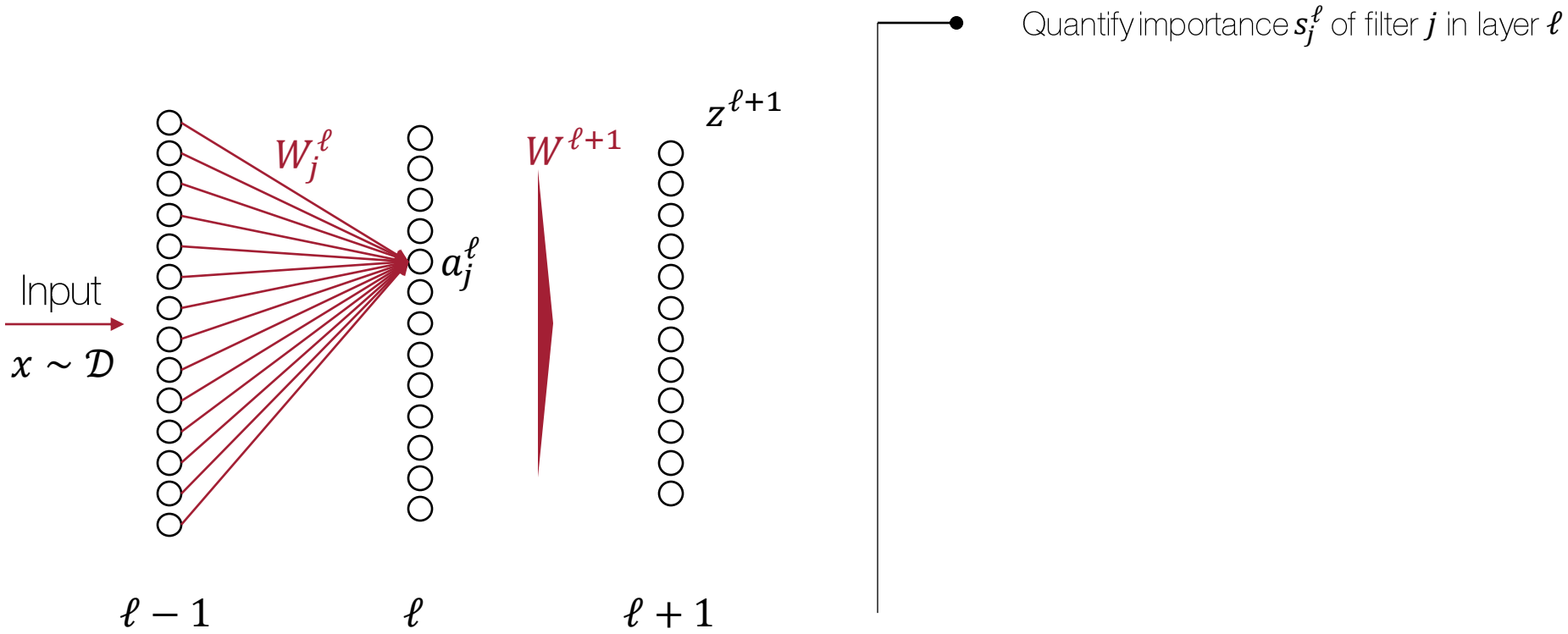
lucasl@mit.edu, baykal@mit.edu

Our method: filter importance



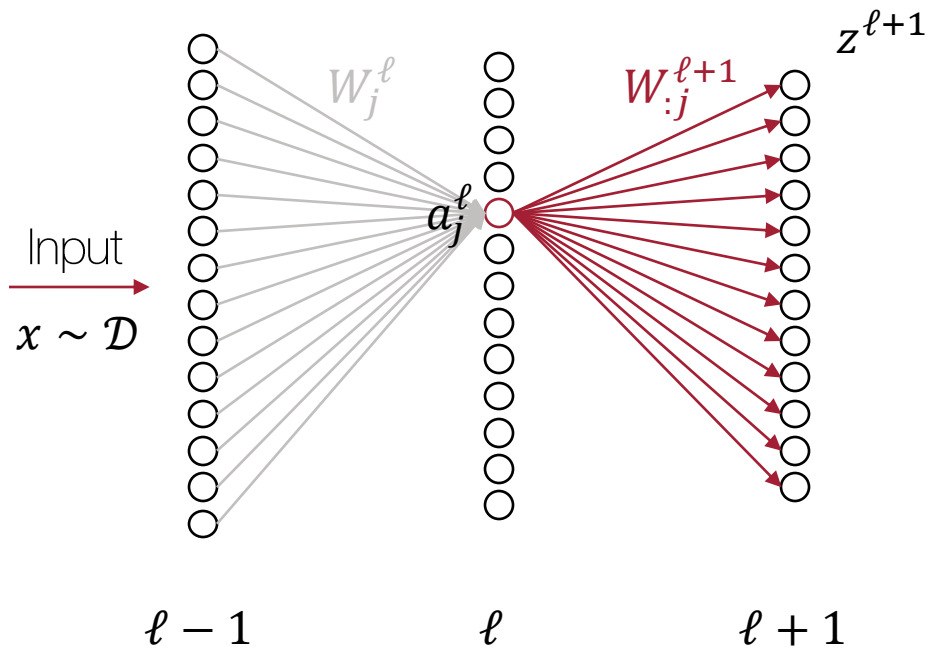
lucasl@mit.edu, baykal@mit.edu

Our method: filter importance



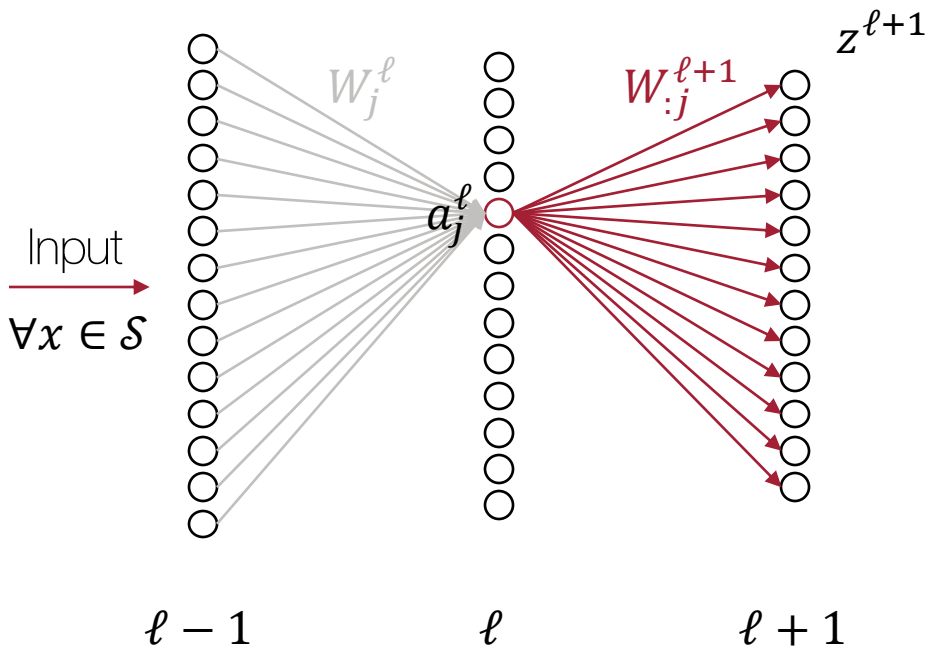
lucasl@mit.edu, baykal@mit.edu

Our method: filter importance



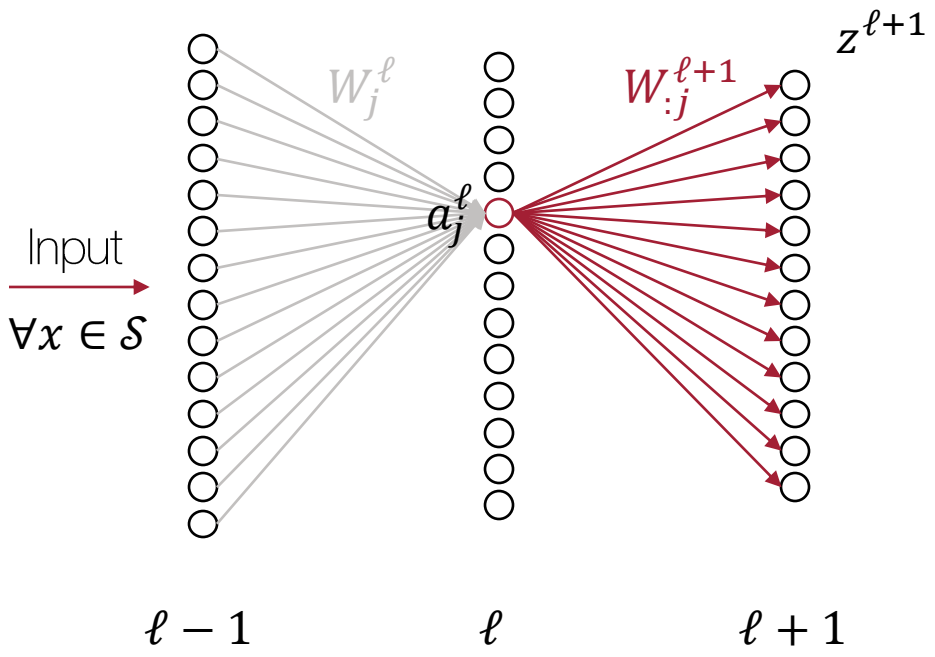
- Quantify importance s_j^ℓ of filter j in layer ℓ
- Consider maximum contribution of activation $a_j^\ell(x)$ in pre-activation $z^{\ell+1}(x)$
- $s_j^\ell \sim \max_i \frac{w_{ij}^\ell a_j^\ell(x)}{z_i^{\ell+1}(x)}$ … filter importance for fixed input $x \sim \mathcal{D}$

Our method: filter importance



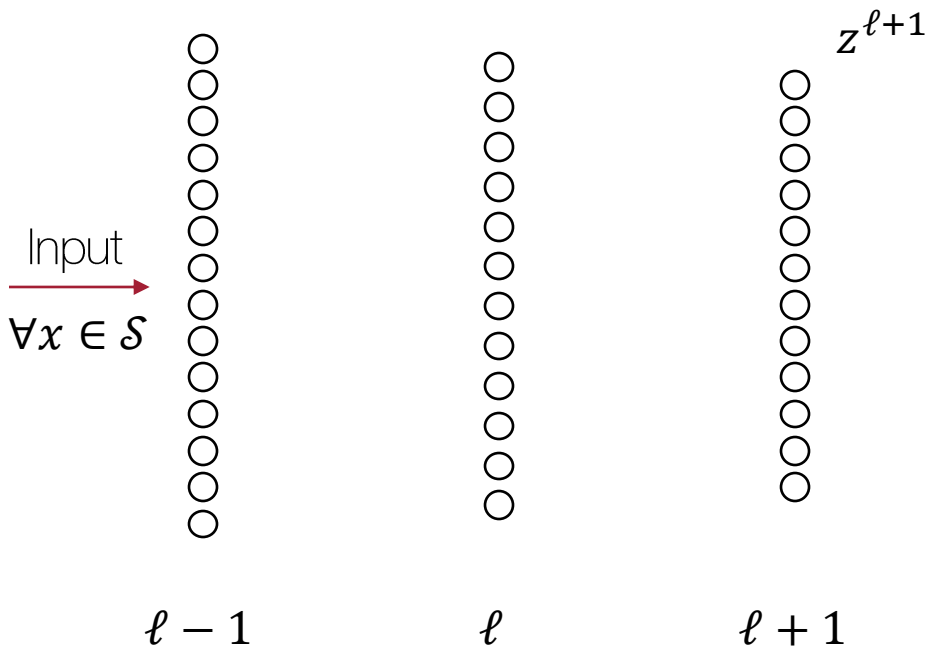
- Quantify importance s_j^ℓ of filter j in layer ℓ
- Consider maximum contribution of activation $a_j^\ell(x)$ in pre-activation $z^{\ell+1}(x)$
- $s_j^\ell \sim \max_{x \in S} \max_i \frac{w_{ij}^\ell a_j^\ell(x)}{z_i^{\ell+1}(x)}$ … filter importance for any input $x \sim \mathcal{D}$ with high probability

Our method: filter importance



- Quantify importance s_j^ℓ of filter j in layer ℓ
- Consider maximum contribution of activation $a_j^\ell(x)$ in pre-activation $z^{\ell+1}(x)$
- $s_j^\ell \sim \max_{x \in S} \max_i \frac{w_{ij}^\ell a_j^\ell(x)}{z_i^{\ell+1}(x)}$ … filter importance for any input $x \sim \mathcal{D}$ with high probability
- Provable Prune step:
  keep filter j in layer ℓ with probability $p_j^\ell \sim s_j^\ell$

Our method: filter importance



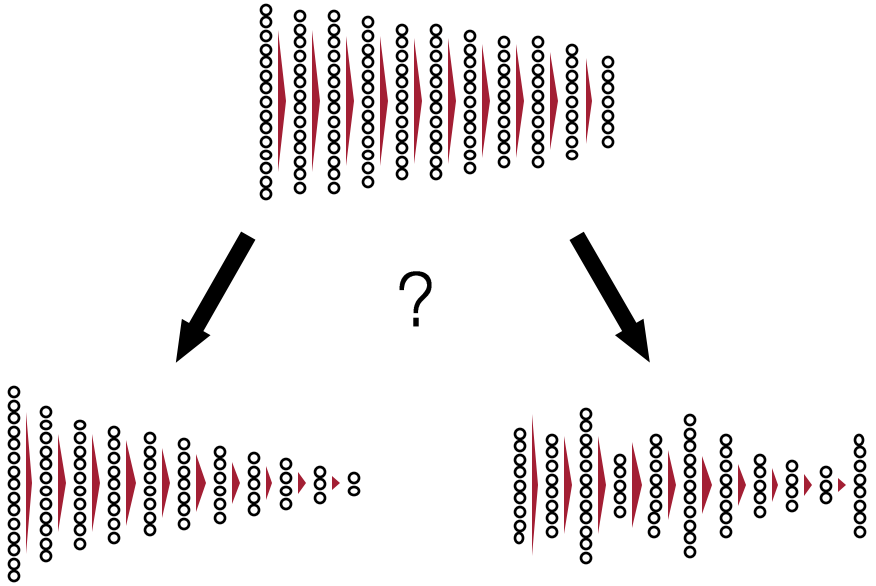
Quantify importance s_j^ℓ of filter j in layer ℓ

Consider maximum contribution of activation $a_j^\ell(x)$ in pre-activation $z^{\ell+1}(x)$

$s_j^\ell \sim \max_{x \in S} \max_i \frac{w_{ij}^\ell a_j^\ell(x)}{z_i^{\ell+1}(x)}$ … filter importance for any input $x \sim \mathcal{D}$ with high probability

Provable Prune step:
keep filter j in layer ℓ with probability $p_j^\ell \sim s_j^\ell$

Our method: budget allocation



Theorem: Relative error $\epsilon^{\ell} = f(m^{\ell})$ in layer ℓ depends on number of samples m^{ℓ}

Pruned size $size(\hat{\theta}) = g(m^1, \ldots, m^L)$

Allocate budget $\mathcal{B}$:

$$m^1, \ldots, m^L = \operatorname{argmin} \max_{\ell} \epsilon^{\ell}(m^{\ell})$$
$$s.t.\ size(\hat{\theta}) = g(m^1, \ldots, m^L) \leq \mathcal{B}$$

Solve efficiently via binary search

lucasl@mit.edu, baykal@mit.edu

Filter compression bounds

For a given network $f_\theta(x)$ with parameters θ and $\epsilon, \delta \in (0,1)$, PFP generates a compressed, *dense* reparameterization $\hat{\theta}$ (i.e. with less filters) such that $\mathbb{P}_{x\sim\mathcal{D}}\left(f_{\hat{\theta}}(x) \in (1 \pm \epsilon) f_\theta(x)\right) \geq (1-\delta)$ and the number of filters is bounded by $\mathcal{O}\left(\sum_{\ell=1}^{L} \frac{L^2 \left(\Delta^\ell\right)^2 S^\ell \log^\eta/_\delta}{\epsilon^2}\right)$

S^ℓ := "sum of sensitivities"

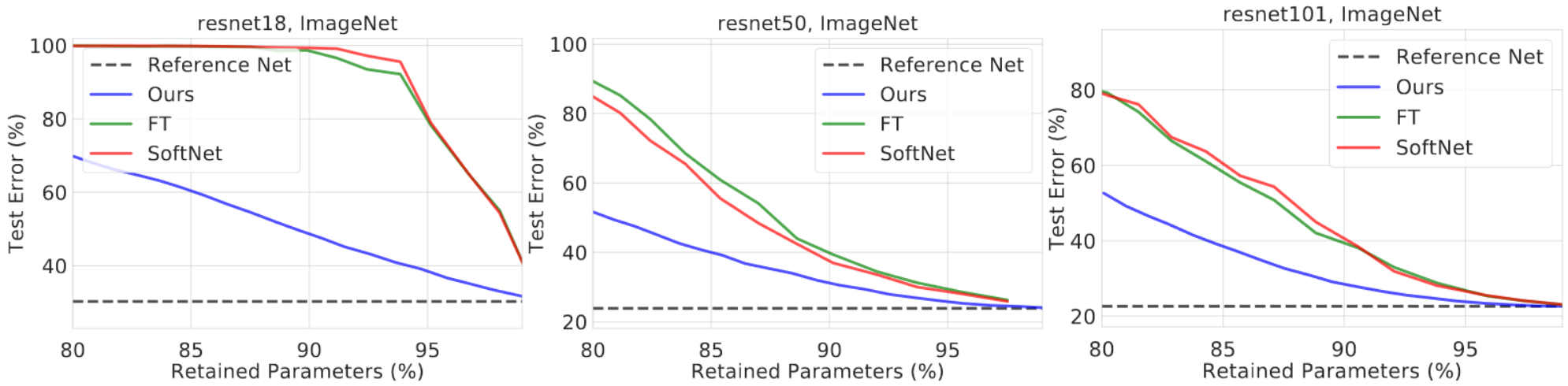
- $S^l = \sum_j s_j^\ell$
- Quantifies "spread" of importance within a layer

Δ^ℓ := "Propagation Complexity"

- Ensures desired relative error within layer
- Considers propagation of error across layers

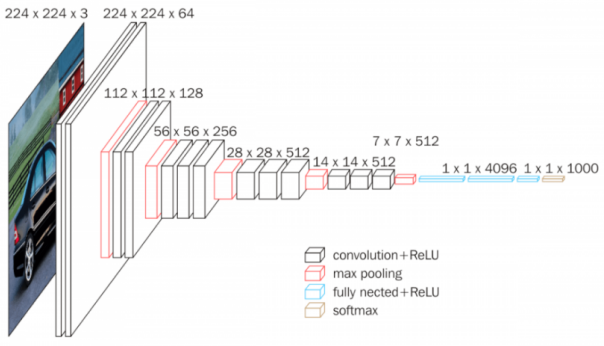
lucasl@mit.edu, baykal@mit.edu

Results: prune-only

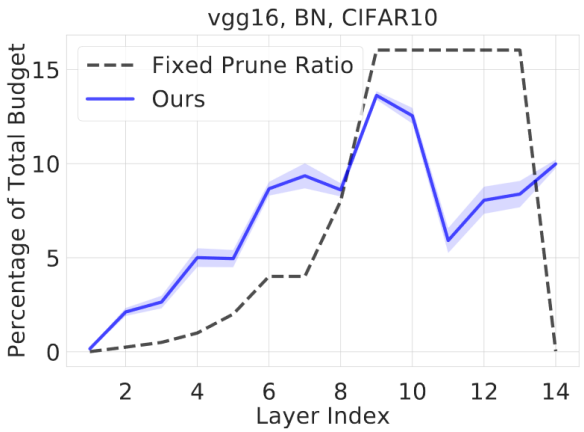


(a) ResNet18 (b) ResNet50 (c) ResNet101

Results: budget allocation



(a) VGG16 architecture



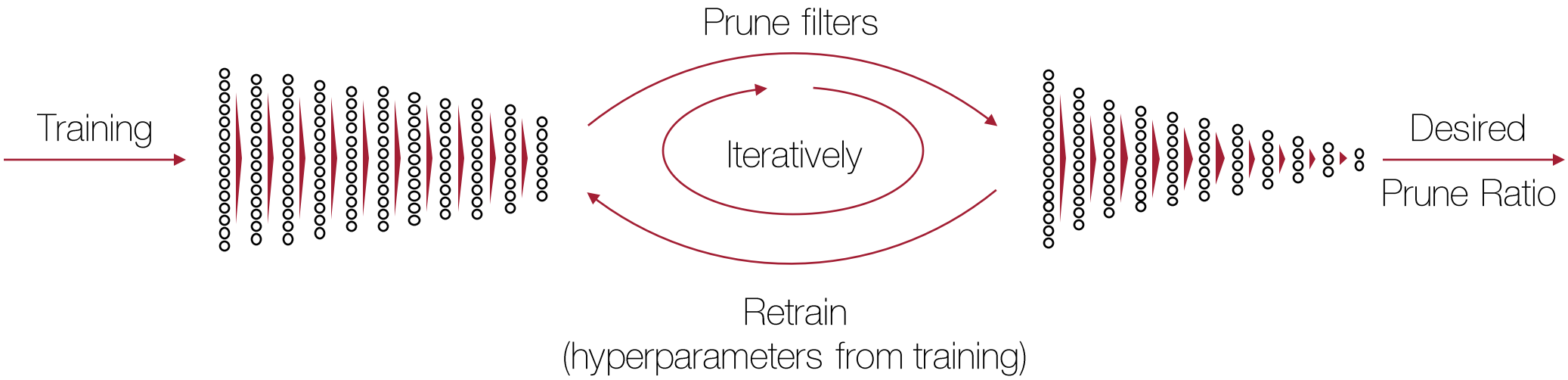
(b) Budget Allocation for VGG16

Early layers are over-sampled
→ small filters, large images

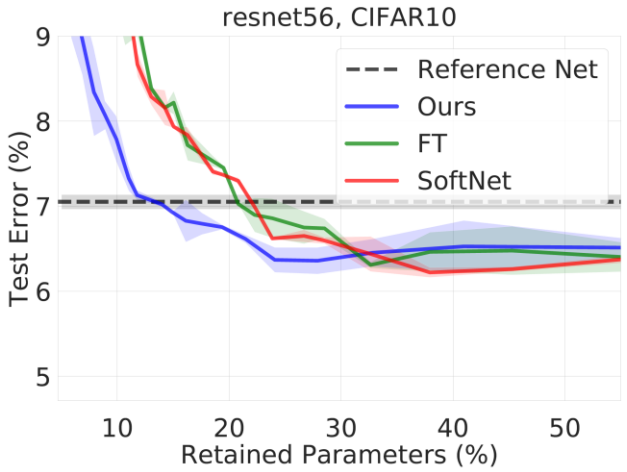
Middle layers are under-sampled
→ large filters, small images

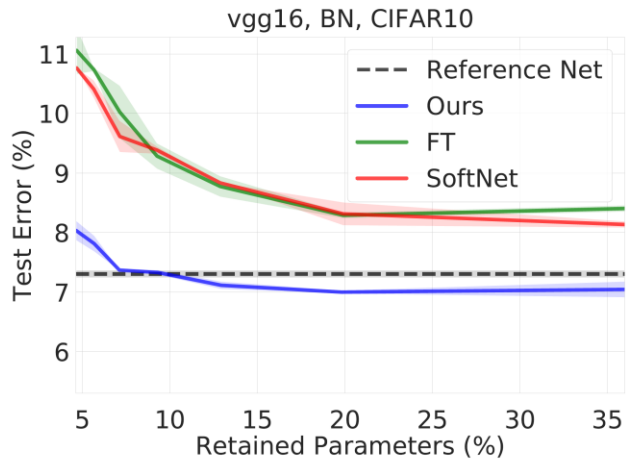
Last layer is over-sampled
→ classification layer!

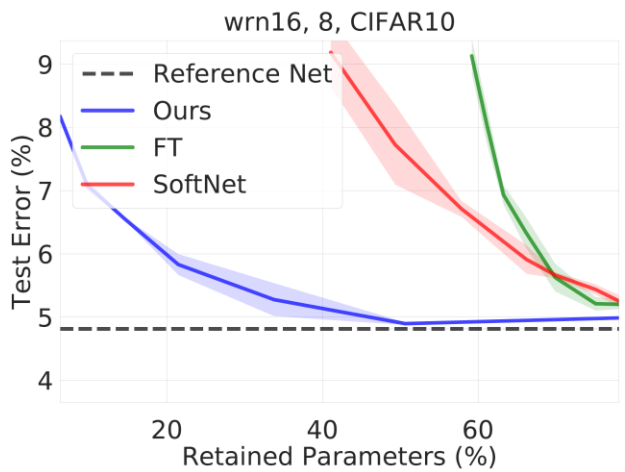
Results: iterative pruning



Results: iterative pruning







More results in the paper
including **ImageNet**

Provable Filter Pruning for Efficient Neural Networks

Lucas Liebenwein*, Cenk Baykal*, Harry Lang, Dan Feldman, Daniela Rus
Distributed Robotics Lab, CSAIL, MIT

- Paper: https://openreview.net/forum?id=BJxkOlSYDH
- Code: https://github.com/lucaslie/provable_pruning
- Contact: lucasl@mit.edu, baykal@mit.edu