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
Larger size, better performance
Smaller size, same performance?
Smaller size, same performance?

- Less storage requirements
- Faster inference time
- Improved Interpretability
Objective: Filter Pruning

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

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

For a given network $f_\theta(x)$ with parameters $\theta$ 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 $s_j^\ell$ of filter $j$ in layer $\ell$
Objective: Filter Pruning

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

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

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

$$\mathbb{P}_{x \sim \mathcal{D}}(f_\hat{\theta}(x) \in (1 \pm \epsilon)f_\theta(x)) \geq (1 - \delta)$$

and $\text{size}(\hat{\theta}) \ll \text{size}(\theta)$. 

Relative importance $s_j^\ell$ of filter $j$ in layer $\ell$

Budget allocation to layers $1, \ldots, \ell, \ldots, L$
Objective: Filter Pruning

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

$$\mathbb{P}_{x \sim \mathcal{D}}(f_{\hat{\theta}}(x) \in (1 \pm \epsilon)f_\theta(x)) \geq (1 - \delta)$$

and $size(\hat{\theta}) \ll size(\theta)$.

---

Relative importance $s_j^\ell$ of filter $j$ in layer $\ell$

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

Relative importance $s_{j}^{\ell}$ of filter $j$ in layer $\ell$

- $s_{j}^{\ell} \sim \|W_{j}^{\ell}\|_{2}$, Li et al. 2016
- $s_{j}^{\ell} \sim \|W_{j}^{\ell}\|_{1}$, He et al., 2018
- $s_{j}^{\ell} \sim 1 / \max_{x} | z^{\ell+1}(x) - z_{[j]}^{\ell+1}(x) |$, Luo et al., 2017

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

- Uniform budget allocation, He et al., 2018
- Sequential pruning process, Luo et al. 2017

$j$...filter, $\ell$...layer, $s_{j}^{\ell}$...importance, $W_{j}^{\ell}$...filter weights, $x$...input, $z^{\ell}(x)$...pre-activation, $z_{[j]}^{\ell}$...pre-activation with feature $j$ only
Related Work

- Relative importance $s_j^\ell$ of filter $j$ in layer $\ell$
- Budget allocation to layers $1, \ldots, \ell, \ldots, L$

- No performance guarantees
- Data-oblivious heuristics
- Manual procedure
- Architecture-specific
Our method: filter importance
Our method: filter importance
Our method: filter importance

\[ x \sim D \]

\[ a^\ell_j \]

\[ W^\ell_j \]

\[ z^{\ell+1} \]

\[ W^\ell_{+1} \]

Quantify importance \( s^\ell_j \) of filter \( j \) in layer \( \ell \)
Our method: filter importance

Quantify importance $s_j^\ell$ of filter $j$ in layer $\ell$

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

$s_j^\ell \sim \max_{\ell} \frac{w_j^\ell a_j^\ell(x)}{z_{\ell+1}(x)}$ \ldots filter importance for fixed input $x \sim \mathcal{D}$
Our method: filter importance

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

\[ \forall x \in S \]

\[ \mathcal{W}_j \]

\[ \ell - 1 \quad \ell \quad \ell + 1 \]
Our method: filter importance

Quantify importance $s_j^\ell$ of filter $j$ in layer $\ell$

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

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

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

Quantify importance $s^\ell_j$ of filter $j$ in layer $\ell$

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

$s^\ell_j \sim \max_{x \in S} \max_{\ell} \frac{w^\ell_{ij} a^\ell_j(x)}{z^{\ell+1}_i(x)}$ ... filter importance for any input $x \sim \mathcal{D}$ with high probability

Provable Prune step:
keep filter $j$ in layer $\ell$ with probability $p^\ell_j \sim s^\ell_j$
Our method: budget allocation

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

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

Allocate budget $B$: $m^1, ..., m^L = \arg\min \max_{\ell} \epsilon_\ell(m^\ell)$

s.t. $\text{size}(\theta) = g(m^1, ..., m^L) \leq B$

Solve efficiently via binary search
Filter compression bounds

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

$S_\ell := \text{“sum of sensitivities”}$

$S_\ell = \sum_j s^j_\ell$

Quantifies “spread” of importance within a layer

$\Delta_\ell := \text{“Propagation Complexity”}$

Ensures desired relative error within layer

Considers propagation of error across layers
Results: prune-only

(a) ResNet18

(b) ResNet50

(c) ResNet101
Results: budget allocation

- 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=RJxkQijSYDH

Code: https://github.com/lucaslie/provable_pruning

Contact: lucasl@mit.edu, baykal@mit.edu