Lookahead: A Far-sighted Alternative of Magnitude-based Pruning

Speaker: Sejun Park

Joint work with Jaeho Lee, Sangwoo Mo, and Jinwoo Shin

Korea Advanced Institute of Science and Technology (KAIST)

ICLR 2020

Motivation: Over-parametrization in Modern Deep Learning

Modern neural networks are severely over-parametrized

- For N training data, $O(N)$ parameters network can achieve zero training error [Yun et al.'19]
  - e.g., 16M parameters are enough for fitting ImageNet dataset perfectly

Number of parameters and ImageNet classification accuracy [Zoph et al.'18]

Model	image size	# parameters	Mult-Adds	Top 1 Acc. (%)	Top 5 Acc. (%)
Inception V2 [29]	224×224	11.2 M	1.94 B	74.8	92.2
NASNet-A (5 @ 1538)	**299×299**	**10.9 M**	**2.35 B**	**78.6**	**94.2**
Inception V3 [60]	299×299	23.8 M	5.72 B	78.8	94.4
Xception [9]	299×299	22.8 M	8.38 B	79.0	94.5
Inception ResNet V2 [58]	299×299	55.8 M	13.2 B	80.1	95.1
NASNet-A (7 @ 1920)	**299×299**	**22.6 M**	**4.93 B**	**80.8**	**95.3**
ResNeXt-101 (64 x 4d) [68]	320×320	83.6 M	31.5 B	80.9	95.6
PolyNet [69]	331×331	92 M	34.7 B	81.3	95.8
DPN-131 [8]	320×320	79.5 M	32.0 B	81.5	95.8
SENet [25]	**320×320**	**145.8 M**	**42.3 B**	**82.7**	**96.2**
NASNet-A (6 @ 4032)	**331×331**	**88.9 M**	**23.8 B**	**82.7**	**96.2**

Motivation: Over-parametrization in Modern Deep Learning

Modern neural networks are severely over-parametrized

- For N training data, $O(N)$ parameters network can achieve zero training error [Yun et al.'19]
  - e.g., 16M parameters are enough for fitting ImageNet dataset perfectly

More parameters

- Better generalization
- Better training accuracy
- Better optimization landscape
- Better convergence speed
- …

Motivation: Over-parametrization in Modern Deep Learning

Modern neural networks are severely over-parametrized

- For N training data, $O(N)$ parameters network can achieve zero training error [Yun et al.'19]
  - e.g., 16M parameters are enough for fitting ImageNet dataset perfectly

More parameters
- Better generalization
- Better training accuracy
- Better optimization landscape
- Better convergence speed
- …

More parameters
- More memory
- More inference time
- More power consumption
- More CO_2
- …

Motivation: Over-parametrization in Modern Deep Learning

Modern neural networks are severely over-parametrized

- For N training data, $O(N)$ parameters network can achieve zero training error [Yun et al.'19]
  - e.g., 16M parameters are enough for fitting ImageNet dataset perfectly

Pruning over-parametrized network

More parameters
- Better generalization
- Better training accuracy
- Better optimization landscape
- Better convergence speed
- …

+

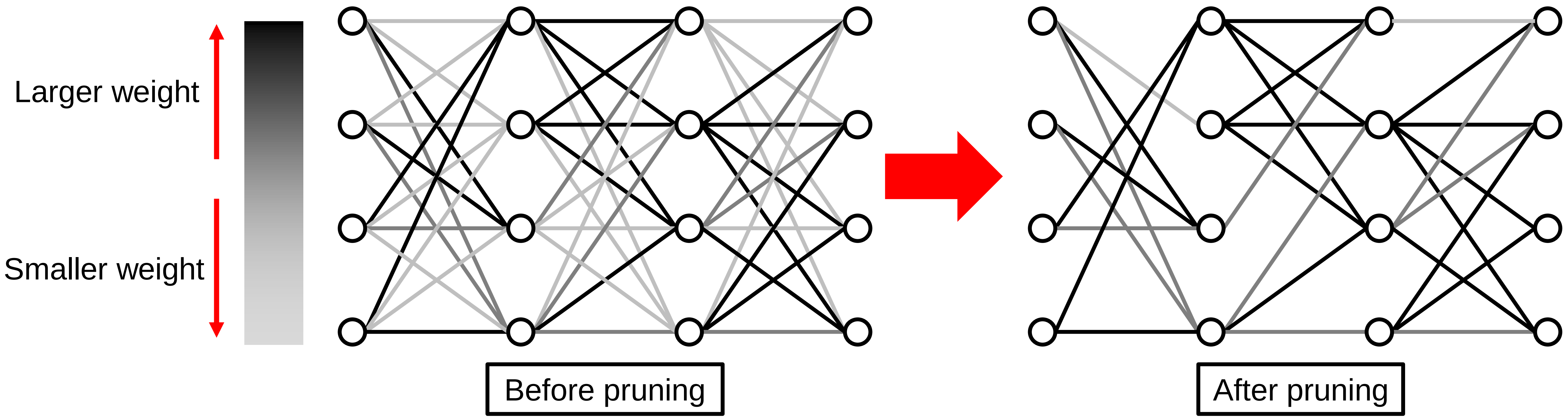
Less parameters
- Less memory
- Less inference time
- Less power consumption
- Less CO_2
- …

Motivation: Magnitude-based Pruning

Magnitude-based pruning (MP) is a popular pruning algorithm, removing small weight edges

Despite its simplicity, MP has been showing remarkable performance in practice

- [Han et al.'15, Han et al.'16, Guo et al.'16, Han et al.'17, Narang et al.'17, Zhu and Gupta'18, Frankle and Carbin'19, Gale et al.'19, Renda et al.'20, Lin et al.'20]



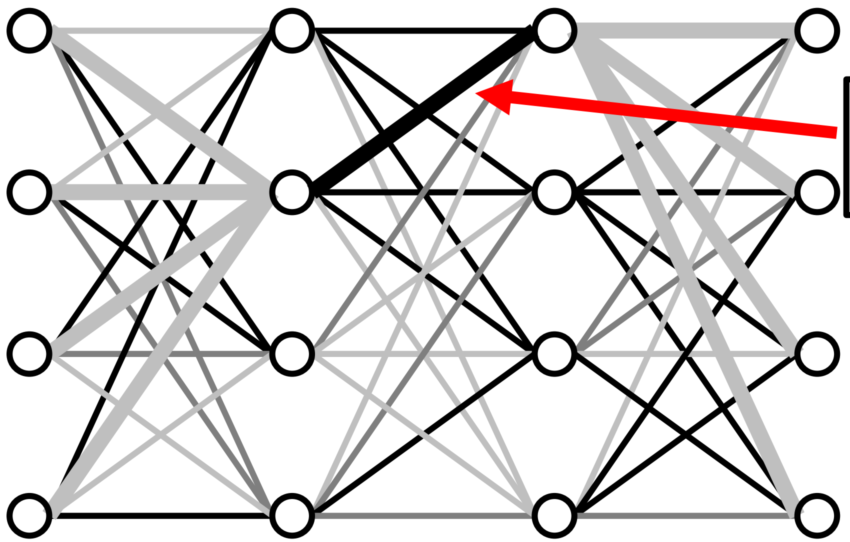
Motivation: Magnitude-based Pruning

Magnitude-based pruning (MP) is a popular pruning algorithm, removing small weight edges

Despite its simplicity, MP has been showing remarkable performance in practice

- [Han et al.'15, Han et al.'16, Guo et al.'16, Han et al.'17, Narang et al.'17, Zhu and Gupta'18, Frankle and Carbin'19, Gale et al.'19, Renda et al.'20, Lin et al.'20]

However, large weight edges may not be important as much as their weights



What if there exists large weight edges connected only to small weight edges?

Motivation: Magnitude-based Pruning

Magnitude-based pruning (MP) is a popular pruning algorithm, removing small weight edges

Despite its simplicity, MP has been showing remarkable performance in practice

We propose a new pruning algorithm by

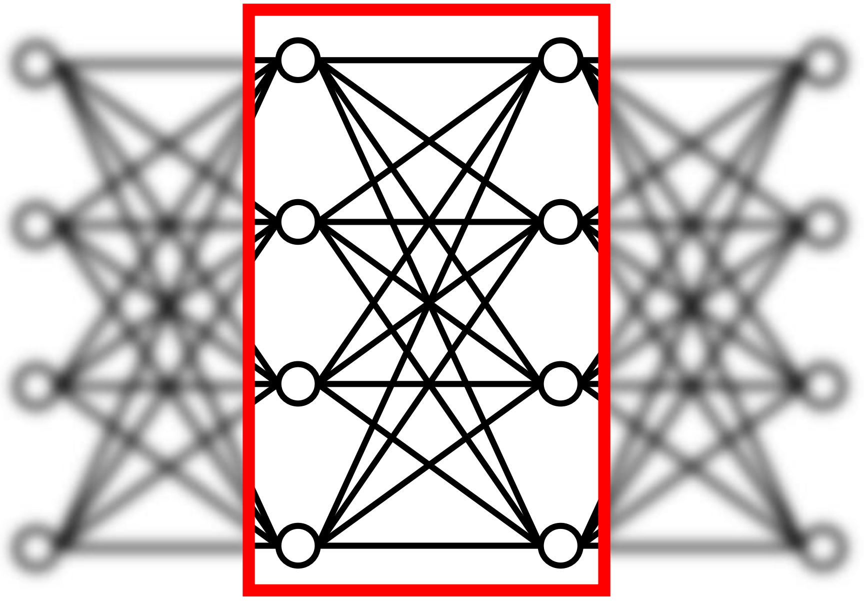
1. Interpreting MP as layerwise approximation
2. Extending it to block approximation

Intuition: Magnitude-based Pruning = Layerwise Approximation

For each layer, MP minimizes Frobenius norm of difference of weight tensors before and after pruning

$$\begin{aligned}\|W_\ell x - \widetilde{W}_\ell x\|_2 &\leq \|W_\ell - \widetilde{W}_\ell\|_2 \cdot \|x\|_2 \\ &\leq \|W_\ell - \widetilde{W}_\ell\|_F \cdot \|x\|_2\end{aligned}$$

x : Input of the layer
W_ℓ: Weight before pruning
$\widetilde{W}_\ell$: Weight after pruning



Intuition: Magnitude-based Pruning = Layerwise Approximation

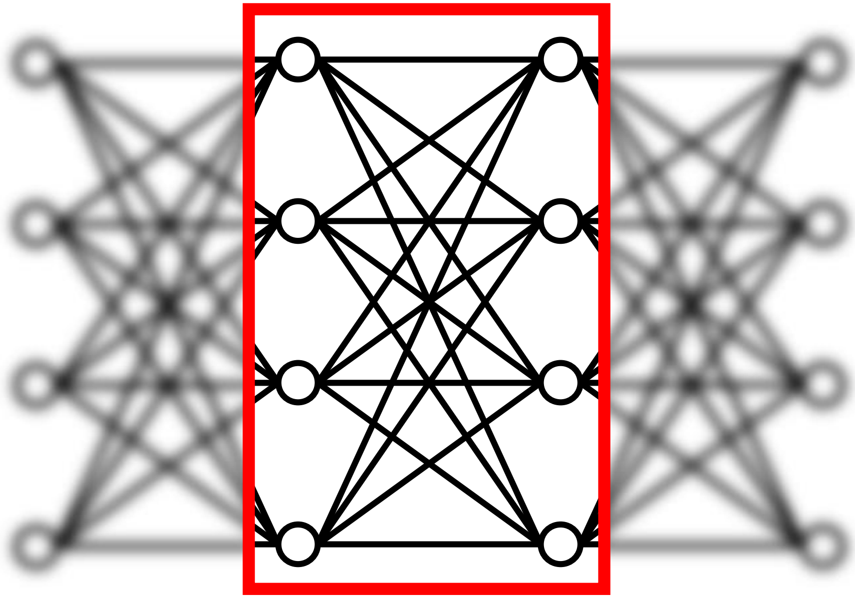
For each layer, MP minimizes Frobenius norm of difference of weight tensors before and after pruning

$$\begin{aligned} \|W_\ell x - \widetilde{W}_\ell x\|_2 &\leq \|W_\ell - \widetilde{W}_\ell\|_2 \cdot \|x\|_2 \\ &\leq \boxed{\|W_\ell - \widetilde{W}_\ell\|_F} \cdot \|x\|_2 \end{aligned}$$

x : Input of the layer

W_ℓ: Weight before pruning

$\widetilde{W}_\ell$: Weight after pruning

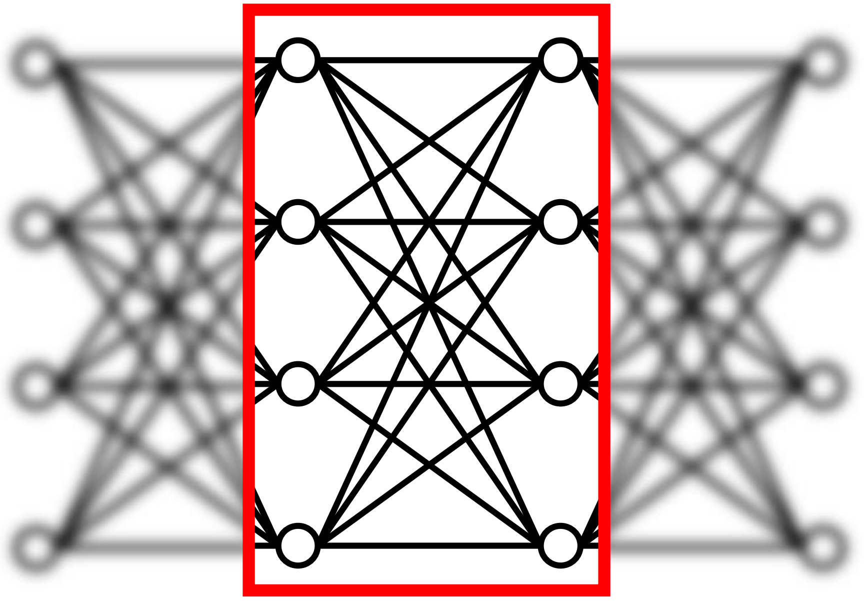


Intuition: Magnitude-based Pruning = Layerwise Approximation

For each layer, MP minimizes Frobenius norm of difference of weight tensors before and after pruning

$$\texttt{score}_{\mathrm{MP}}(i,j;\ell) = \underline{|W_\ell[i,j]|}$$

Pruning edge with smallest MP score minimizes $\|W_\ell - \widetilde{W}_\ell\|_F$



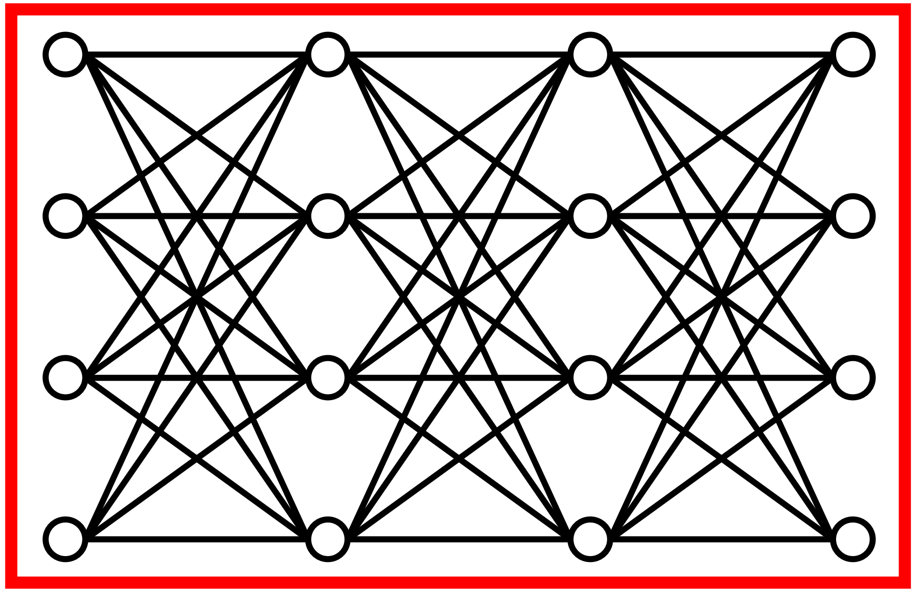
Contribution: Lookahead Pruning = Block Approximation

We propose lookahead pruning (LAP) extending layerwise approximation of MP to block of layers

$$\underline{\|W_{\ell+1}W_\ell W_{\ell-1}x - W_{\ell+1}\widetilde{W}_\ell W_{\ell-1}x\|_2} \leq \|W_{\ell+1}(W_\ell - \widetilde{W}_\ell)W_{\ell-1}\|_2 \cdot \|x\|_2$$

$$\leq \|W_{\ell+1}(W_\ell - \widetilde{W}_\ell)W_{\ell-1}\|_F \cdot \|x\|_2$$

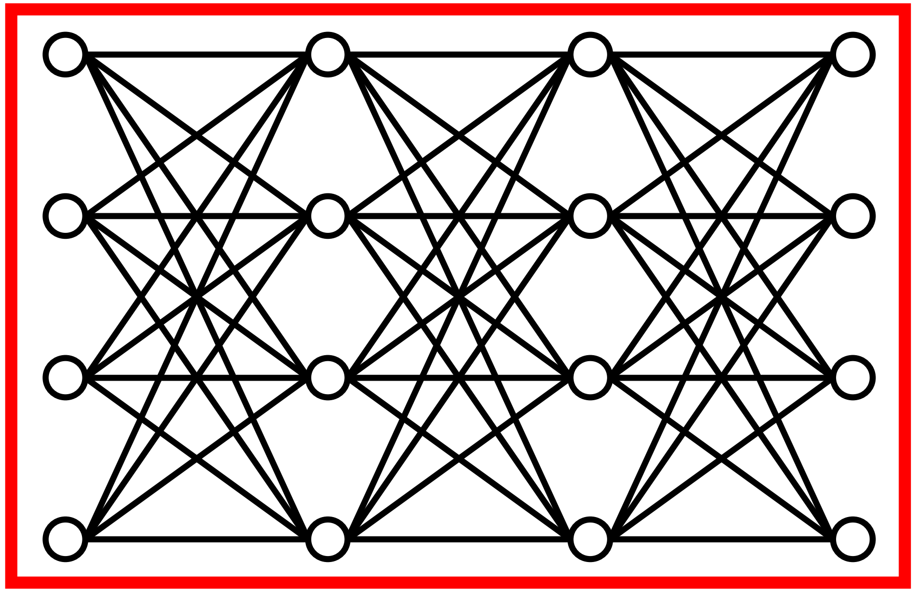
Assume linear activation for now



Contribution: Lookahead Pruning = Block Approximation

We propose lookahead pruning (LAP) extending layerwise approximation of MP to block of layers

$$\begin{aligned}\|W_{\ell+1}W_\ell W_{\ell-1}x - W_{\ell+1}\widetilde{W}_\ell W_{\ell-1}x\|_2 &\le \|W_{\ell+1}(W_\ell - \widetilde{W}_\ell)W_{\ell-1}\|_2 \cdot \|x\|_2 \\ &\le \boxed{\|W_{\ell+1}(W_\ell - \widetilde{W}_\ell)W_{\ell-1}\|_F} \cdot \|x\|_2\end{aligned}$$

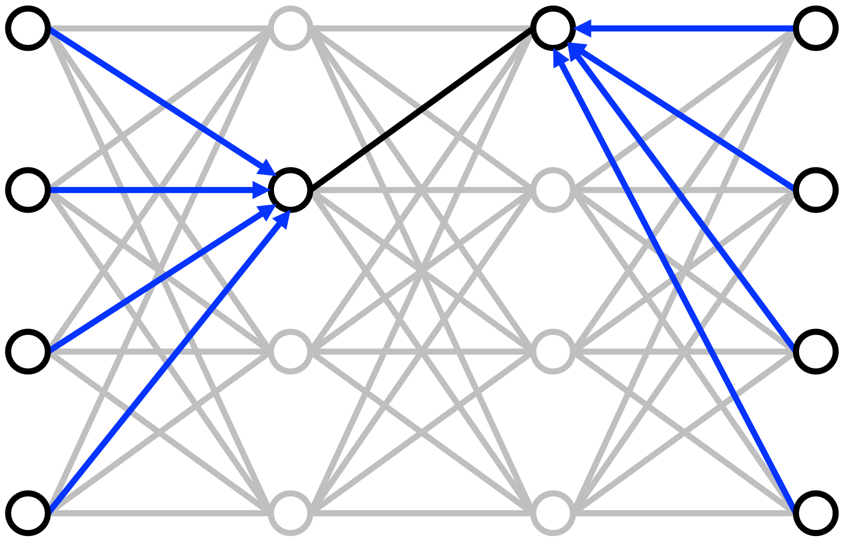


Contribution: Lookahead Pruning = Block Approximation

We propose lookahead pruning (LAP) extending layerwise approximation of MP to block of layers

$$\texttt{score}_{\mathrm{LAP}}(i,j;\ell) = \underline{|W_\ell[i,j]| \cdot \left\|W_{\ell+1}[:,i]\right\|_F \cdot \left\|W_{\ell-1}[j,:]\right\|_F}$$

Pruning edge with smallest LAP score minimizes
$\|W_{\ell+1}(W_\ell - \widetilde{W}_\ell)W_{\ell-1}\|_F$

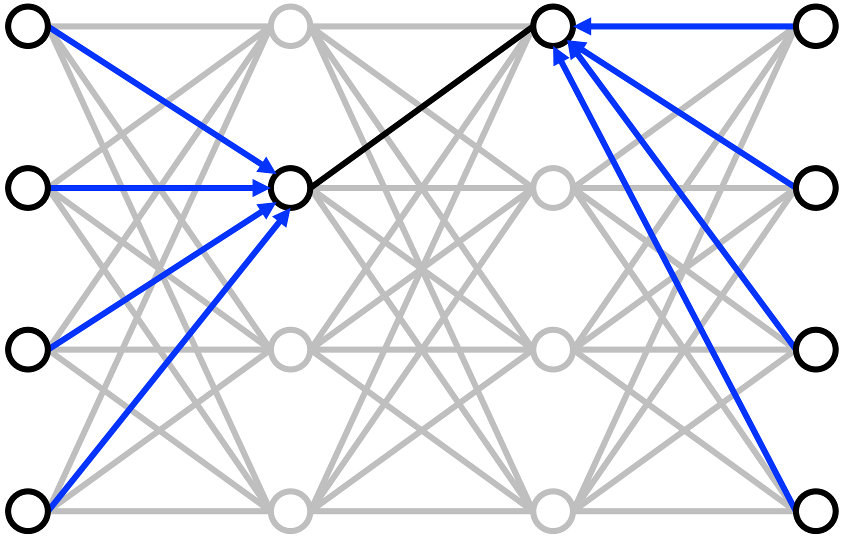


Contribution: Lookahead Pruning for ReLU

LAP for ReLU activation under i.i.d. activation probability

$$\mathbb{E}\left[\|W_{\ell+1}\underline{X_\ell}(W_\ell - \widetilde{W}_\ell)\underline{X_{\ell-1}}W_{\ell-1}x\|_2^2\right]^{\frac{1}{2}} \leq \mathbb{E}\left[\|W_{\ell+1}X_\ell(W_\ell - \widetilde{W}_\ell)X_{\ell-1}W_{\ell-1}\|_F^2\right]^{\frac{1}{2}} \cdot \|x\|_2$$

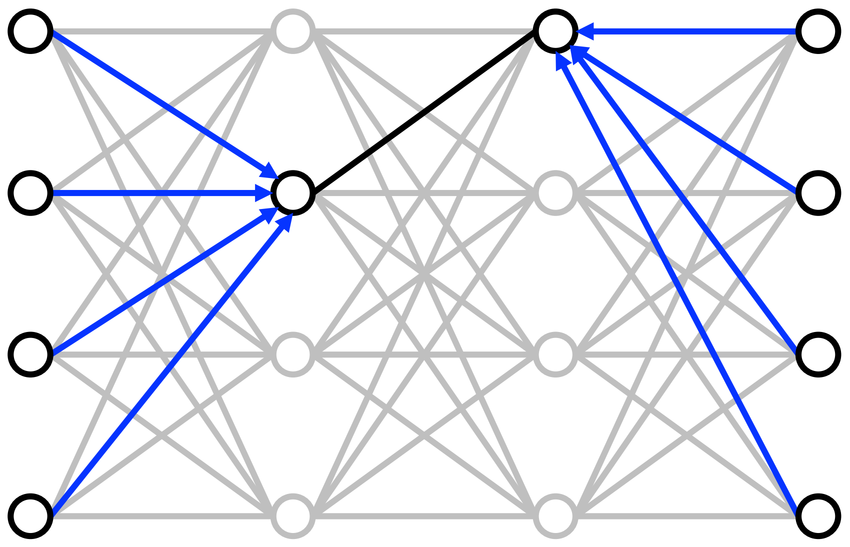
0-1 random diagonal matrix indicating activated neurons



Contribution: Lookahead Pruning for ReLU

LAP for ReLU activation under i.i.d. activation probability

$$\mathbb{E}\left[\|W_{\ell+1}X_\ell(W_\ell - \widetilde{W}_\ell)X_{\ell-1}W_{\ell-1}x\|_2^2\right]^{\frac{1}{2}} \leq \boxed{\mathbb{E}\left[\|W_{\ell+1}X_\ell(W_\ell - \widetilde{W}_\ell)X_{\ell-1}W_{\ell-1}\|_F^2\right]^{\frac{1}{2}}} \cdot \|x\|_2$$



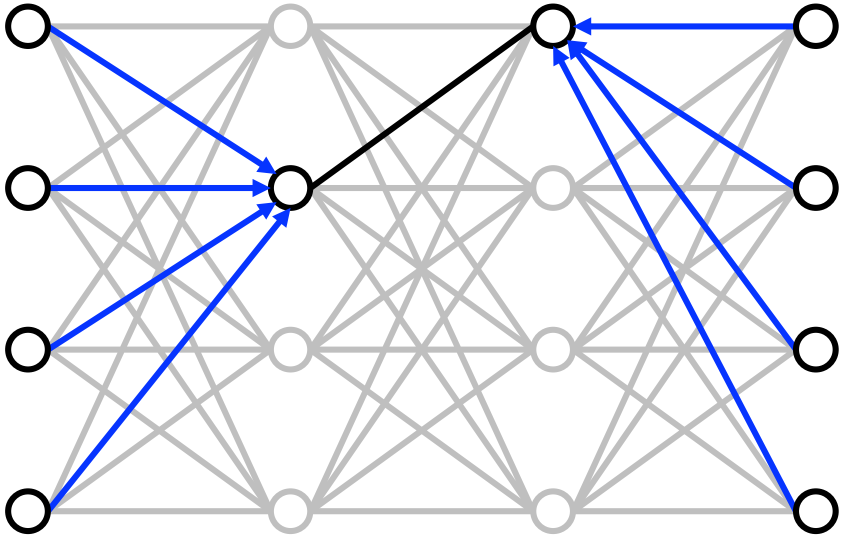
Contribution: Lookahead Pruning for ReLU

LAP for ReLU activation under i.i.d. activation probability

$$\mathtt{score}_{\mathrm{LAP}}(i,j;\ell) = \underline{|W_\ell[i,j]| \cdot \left\|W_{\ell+1}[:,i]\right\|_F \cdot \left\|W_{\ell-1}[j,:]\right\|_F}$$

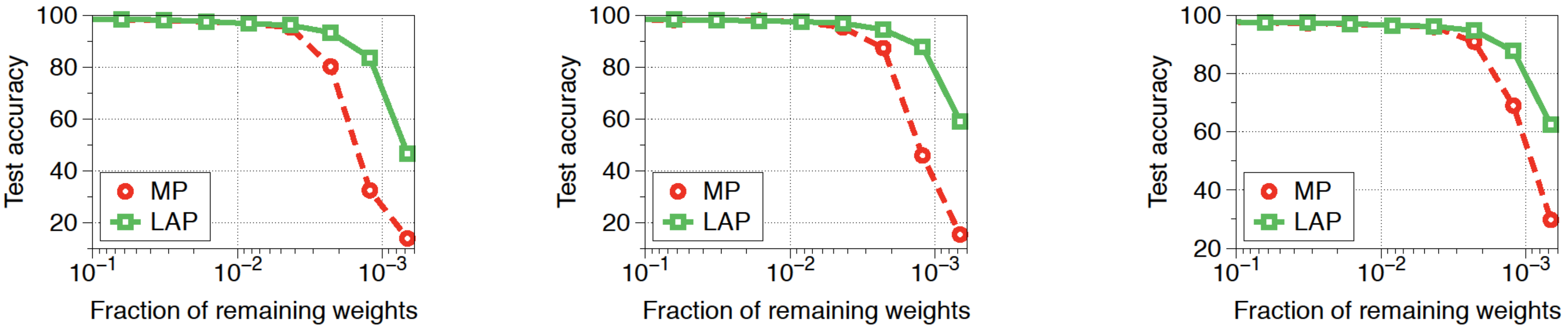
Pruning edge with smallest LAP score minimizes

$$\mathbb{E}\left[\|W_{\ell+1}X_\ell(W_\ell - \widetilde{W_\ell})X_{\ell-1}W_{\ell-1}\|_F^2\right]^{\frac{1}{2}}$$



Experiments: Lookahead Pruning for Other Activations

Empirical evaluation of LAP and MP for MNIST classification task under non-linear activation functions



ReLU | sigmoid | hyperbolic tangent

Experiments: Lookahead Pruning for Modern CNNs

Empirical evaluation of LAP and MP for CIFAR-10 and Tiny-ImageNet classification tasks

Test error rates of VGG-19 on CIFAR-10. Unpruned models have 9.02% error rate.

	12.09%	8.74%	6.31%	4.56%	3.30%	2.38%	1.72%	1.24%
MP	8.99±0.12	9.90±0.09	11.43±0.24	15.62±1.68	29.10±8.78	40.27±11.51	63.27±11.91	77.90±7.94
LAP	**8.89±0.14** (**-1.07%**)	**9.51±0.22** (**-3.96%**)	**10.56±0.28** (**-7.63%**)	**12.11±0.44** (**-22.48%**)	**13.64±0.77** (**-53.13%**)	**16.38±1.47** (**-59.31%**)	**20.88±1.71** (**-67.00%**)	**22.82±0.81** (**-70.71%**)

Test error rates of ResNet-18 on CIFAR-10. Unpruned models have 8.68% error rate.

	10.30%	6.33%	3.89%	2.40%	1.48%	0.92%	0.57%	0.36%
MP	8.18±0.33	**8.74±0.15**	9.82±0.18	**11.28±0.30**	14.31±0.18	18.56±0.36	22.93±0.93	26.77±1.04
LAP	**8.09±0.10** (**-1.08%**)	8.97±0.22 (+2.59%)	**9.74±0.15** (**-0.81%**)	11.35±0.20 (+0.64%)	**13.73±0.24** (**-4.08%**)	**16.29±0.29** (**-12.23%**)	**20.22±0.53** (**-11.82%**)	**22.45±0.64** (**-15.82%**)

Top-5 test error rates of VGG-19 on Tiny-ImageNet. Unpruned models have 36.89% error rate.

	12.16%	10.34%	8.80%	7.48%	6.36%	5.41%	4.61%	3.92%
MP	36.40±1.31	37.37±1.08	38.40±1.30	40.23±1.26	42.68±1.97	45.83±2.76	49.79±2.67	56.15±5.14
LAP	**36.01±1.31** (**-1.07%**)	**37.03±0.90** (**-0.90%**)	**38.20±1.61** (**-0.52%**)	**39.36±1.30** (**-2.16%**)	**40.95±1.46** (**-4.05%**)	**43.14±1.33** (**-5.87%**)	**45.29±1.80** (**-9.02%**)	**48.34±0.30** (**-13.92%**)

Top-5 test error rates of ResNet-50 on Tiny-ImageNet. Unpruned models have 23.19% error rate.

	6.52%	4.74%	3.45%	2.51%	1.83%	1.34%	0.98%	0.72%
MP	23.88±0.27	24.99±0.34	26.84±0.39	29.54±0.58	34.04±0.48	40.19±0.36	45.13±0.57	59.18±16.31
LAP	**23.64±0.40** (**-1.00%**)	**24.91±0.25** (**-0.34%**)	**26.52±0.38** (**-1.17%**)	**28.84±0.43** (**-2.38%**)	**33.71±0.58** (**-0.98%**)	**39.07±0.45** (**-2.79%**)	**43.05±0.97** (**-4.61%**)	**46.16±1.04** (**-22.00%**)

Top-5 test error rates of WRN-16-8 on Tiny-ImageNet. Unpruned models have 25.77% error rate.

	12.22%	8.85%	6.41%	4.65%	3.37%	2.45%	1.77%	1.29%
MP	25.27±0.73	26.79±0.87	28.84±1.04	**31.91±0.80**	37.01±1.42	42.89±2.43	51.10±2.59	59.73±2.85
LAP	**24.99±0.85** (**-1.12%**)	**26.55±1.45** (**-0.87%**)	**28.68±1.17** (**-0.58%**)	32.22±2.51 (+0.98%)	**35.82±2.06** (**-3.22%**)	**41.37±3.07** (**-3.55%**)	**45.43±4.48** (**-11.10%**)	**51.83±1.91** (**-13.22%**)

LAP outperforms MP especially in high-sparsity regime!

Summary

We propose lookahead pruning by extending layerwise approximation of MP to block approximation

Summary

We propose lookahead pruning by extending layerwise approximation of MP to block approximation

In our paper, there are

- More empirical evaluations
- Variants and sequential version of LAP
- LAP for various types of layers
- LAP utilizing real activation probability
- ...

Codes are available at https://github.com/alinlab/lookahead_pruning