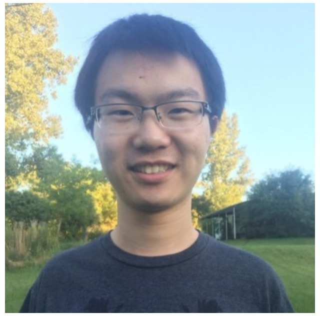
A Simple and Effective Pruning Approach for Large Language Models

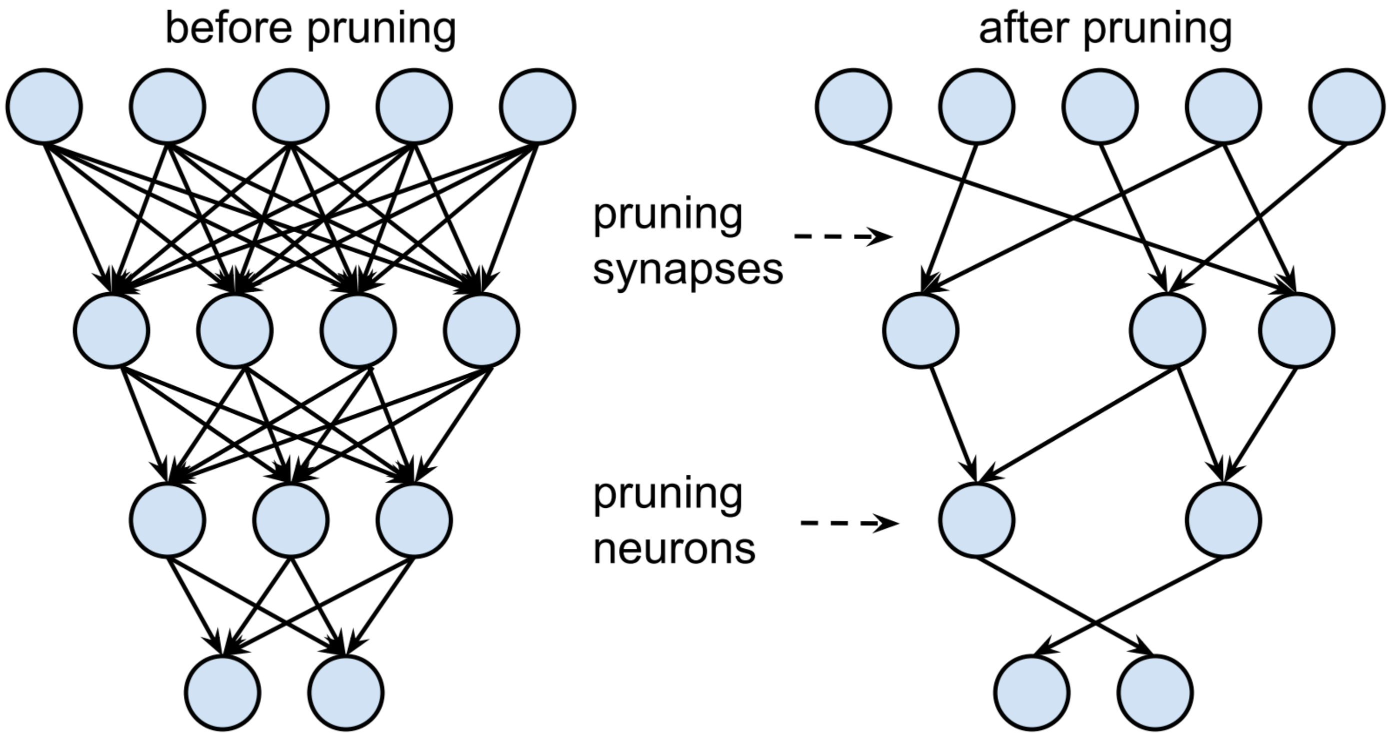
Mingjie Sun
Carnegie Mellon University

Joint work with Zhuang Liu, Anna Bair, Zico Kolter

Network Pruning

A popular approach for compressing neural networks.



Learning both weights and connections for Efficient Neural Networks. Han et al, 2015

Network Pruning

ICLR 2019 best paper award.

Published as a conference paper at ICLR 2019

THE LOTTERY TICKET HYPOTHESIS:
FINDING SPARSE, TRAINABLE NEURAL NETWORKS

Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

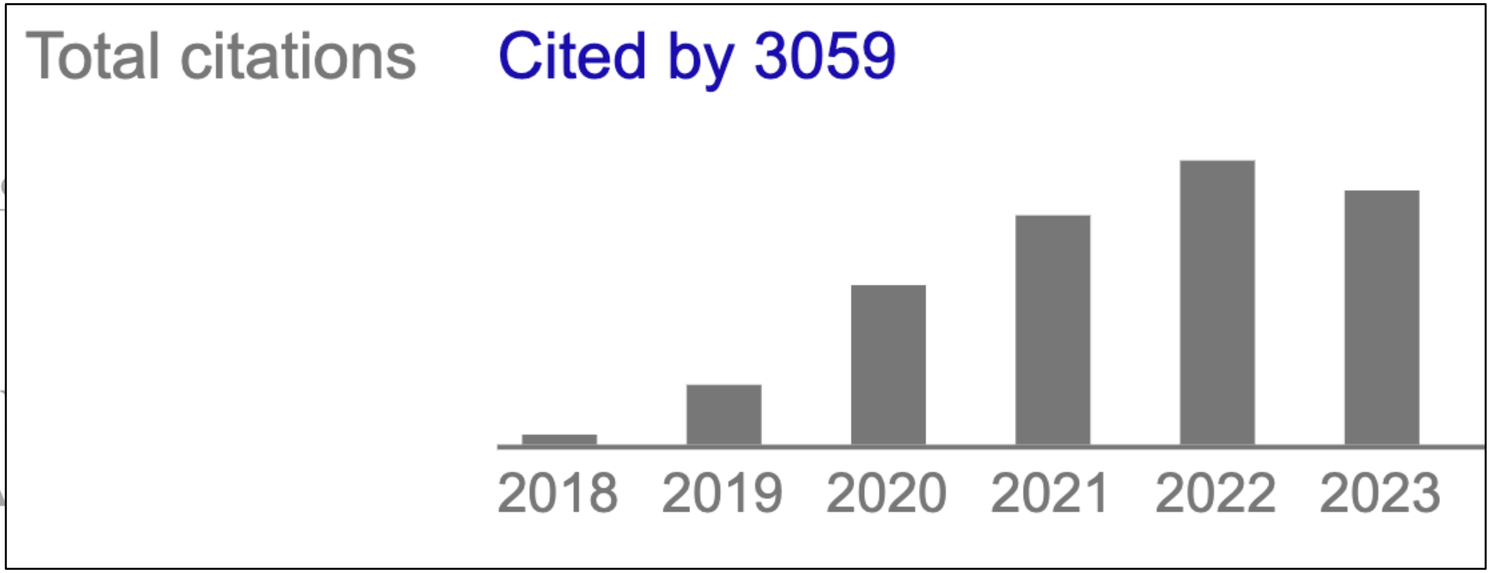
Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

Network Pruning

Huge research interest.

Published as a conferen

THE LOTTER
FINDING SPA



Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

Behind the success

Magnitude Pruning: remove weights with smallest magnitudes.

Behind the success

Magnitude Pruning: remove weights with smallest magnitudes.

A simple but tough to beat baseline

The State of Sparsity in Deep Neural Networks

Trevor Gale[*1†] **Erich Elsen**[*2] **Sara Hooker**[1†]

WHAT IS THE STATE OF NEURAL NETWORK PRUNING?

Davis Blalock[*1] **Jose Javier Gonzalez Ortiz**[*1] **Jonathan Frankle**[1] **John Guttag**[1]

Setting the scope

- Type of pruning:
    - Unstructured Pruning
    - Structured Pruning

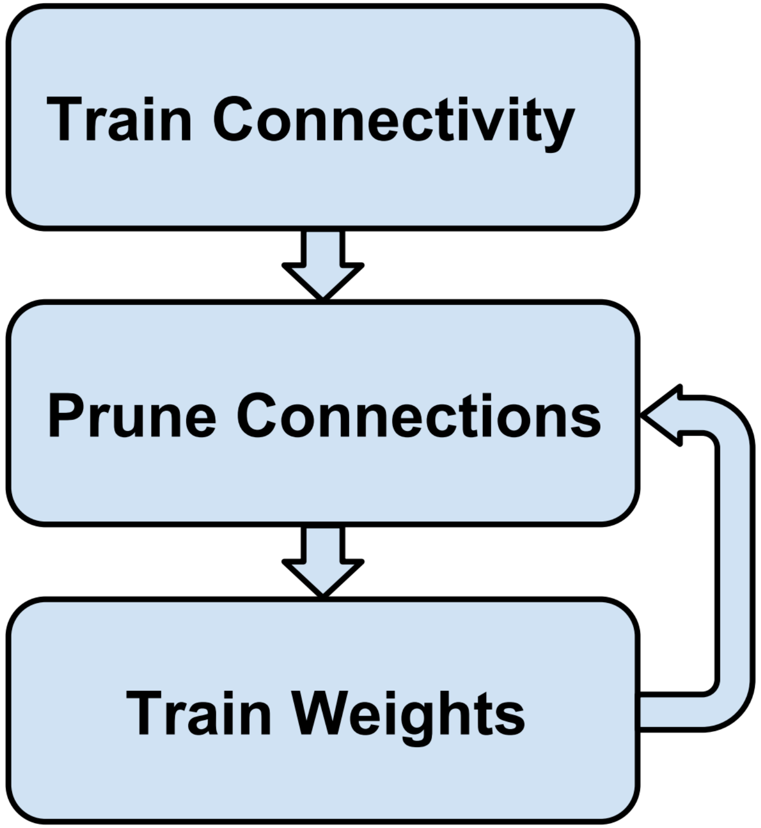
Setting the scope

- Type of pruning:
    - Unstructured Pruning
    - ~~Structured Pruning~~

Setting the scope

- Type of pruning:
  - Unstructured Pruning
  - ~~Structured Pruning~~

- Pruning procedure

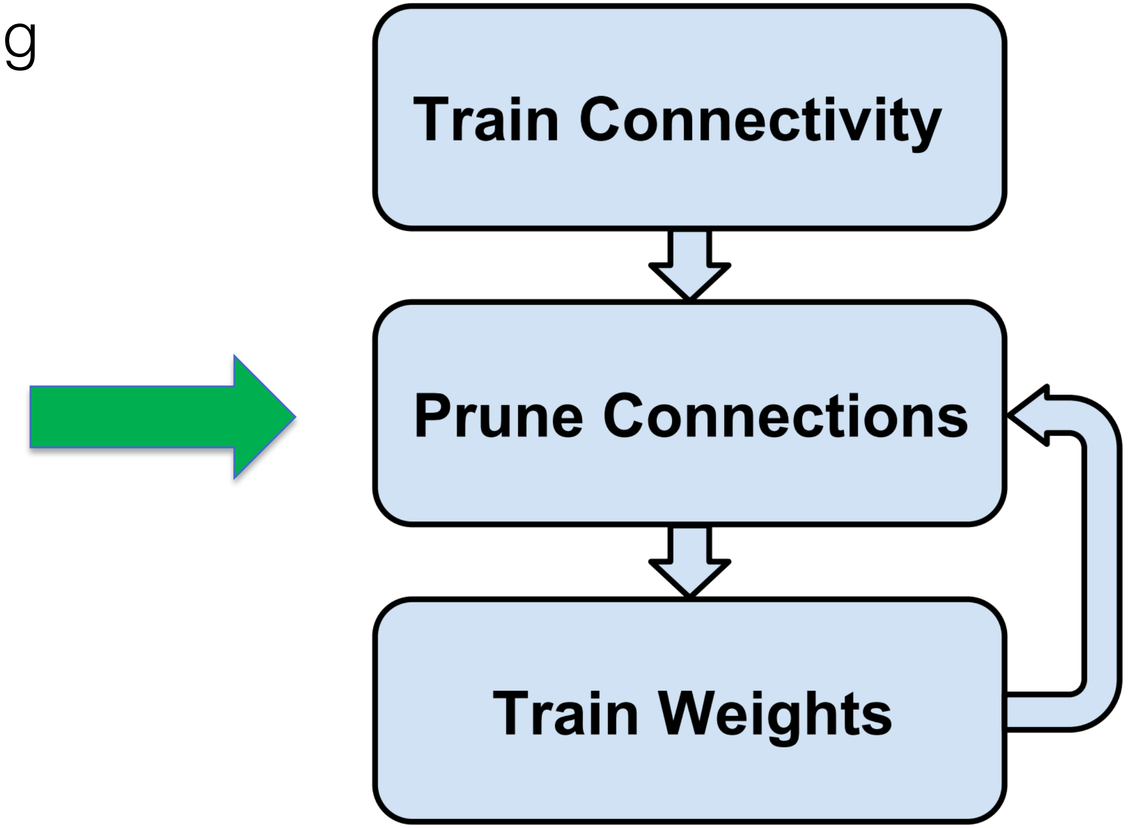


Learning both weights and connections for Efficient Neural Networks. Han et al, 2015

Setting the scope

- Type of pruning:
    - Unstructured Pruning
    - ~~Structured Pruning~~

- Pruning procedure



Magnitude Pruning

W

4	0	1	-1
3	-2	-1	-3
-3	1	0	2

Weights

Magnitude Pruning

$$\mathbf{S} = |\mathbf{W}|$$

Pruning Metric

$$\mathbf{W} \quad \begin{bmatrix} 4 & 0 & 1 & -1 \\ 3 & -2 & -1 & -3 \\ -3 & 1 & 0 & 2 \end{bmatrix} \rightarrow$$

Weights

Magnitude Pruning

$$\mathbf{S} = |\mathbf{W}|$$

Pruning Metric

$\mathbf{W}$

4	0	1	-1
3	-2	-1	-3
-3	1	0	2

Weights

4	0	1	1
3	2	1	3
3	1	0	2

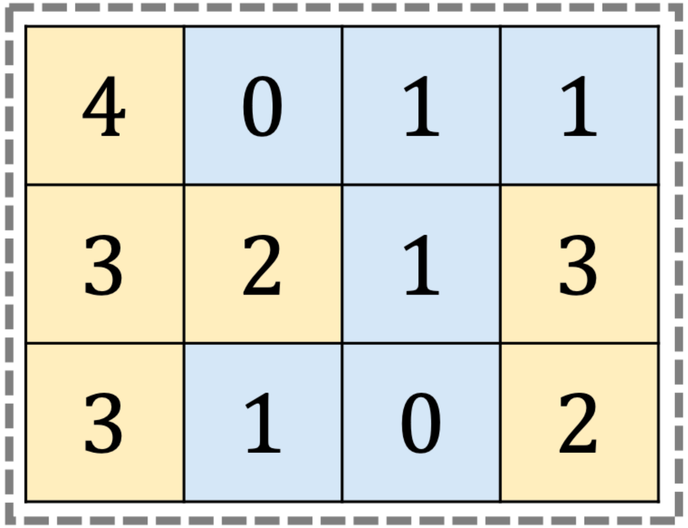
Weight Importance

Magnitude Pruning

4	0	1	1
3	2	1	3
3	1	0	2

Weight Importance

Magnitude Pruning



Weight Importance

Comparison Group *grouped per layer*

Magnitude Pruning

4	0	1	1
3	2	1	3
3	1	0	2

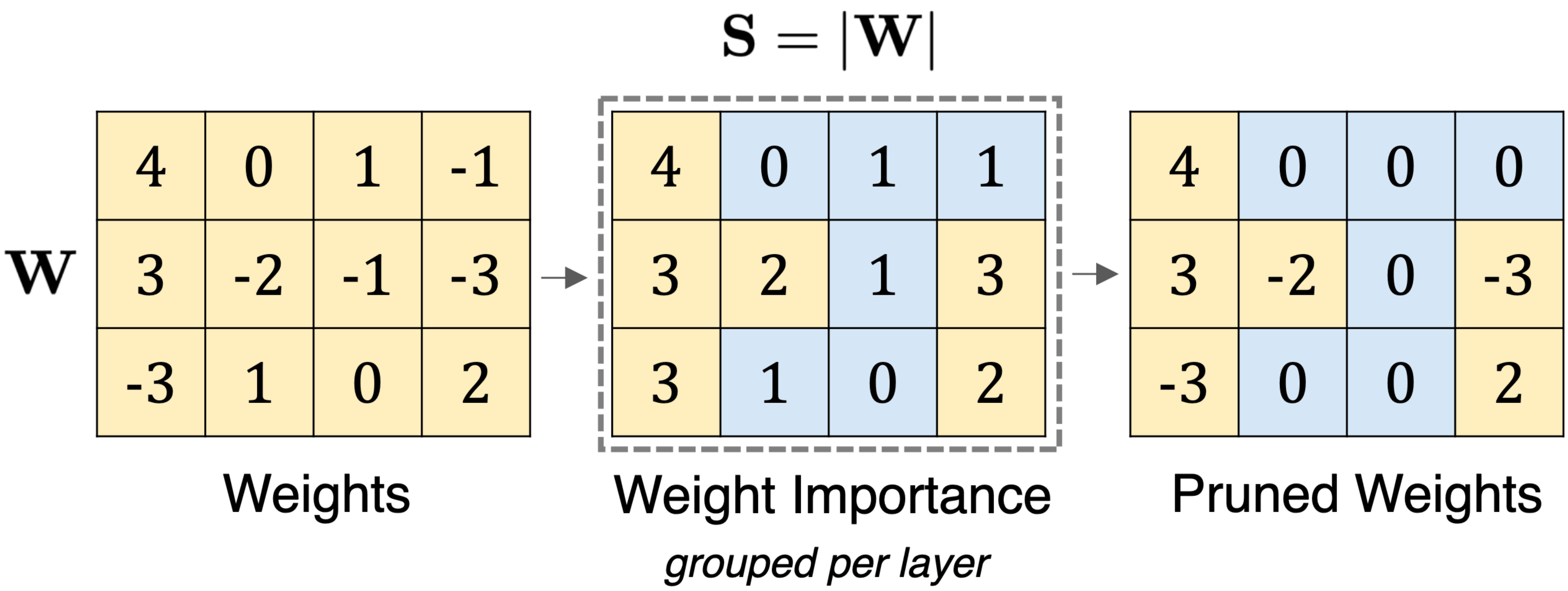
Weight Importance

4	0	0	0
3	-2	0	-3
-3	0	0	2

Pruned Weights

Comparison Group → *grouped per layer*

Magnitude Pruning



A Dilemma for Pruning LLMs

	ImageNet Accuracy	WikiText Perplexity
Magnitude Pruning	ConvNeXt	LLaMA-7B
#Params	89M	7B
Dense	83.8%	5.68
50% sparsity		

A Dilemma for Pruning LLMs

Magnitude Pruning	ImageNet Accuracy ConvNeXt	WikiText Perplexity LLaMA-7B
#Params	89M	7B
Dense	83.8%	5.68
50% sparsity	82.4%	17.29

Significant performance drop.

A Dilemma for Pruning LLMs

WikiText perplexity	Dense	10%	20%
OPT-13B	10.13	14.45	9e3

Explodes at 20% sparsity!!!

A Dilemma for Pruning LLMs

Large language models, despite having 100x or 1000x more parameters, are significantly harder to prune directly.

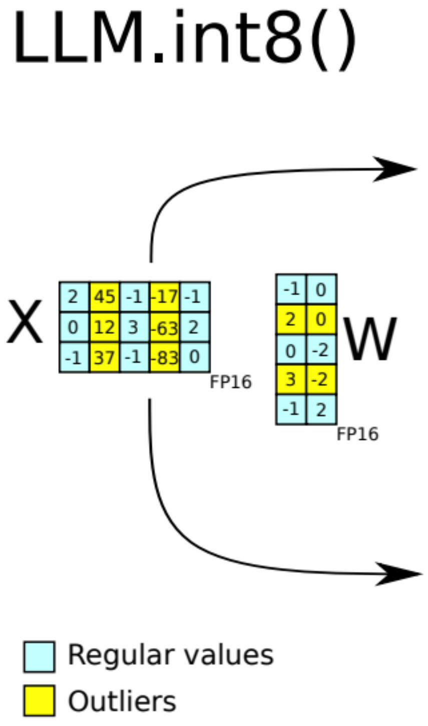
A Dilemma for Pruning LLMs

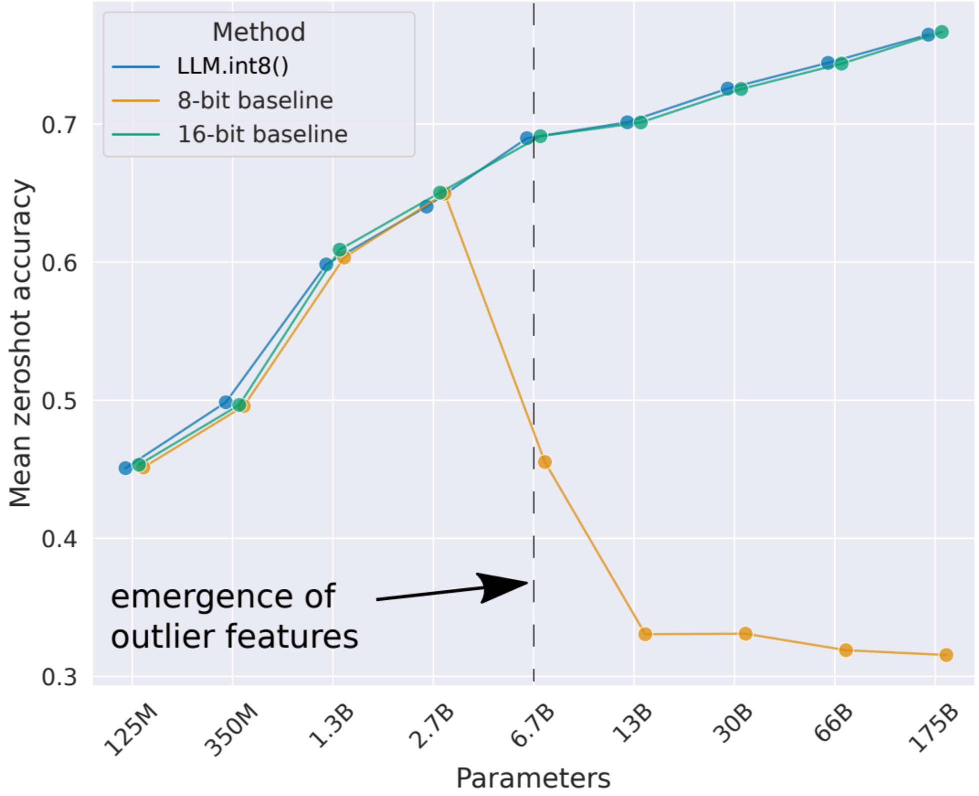
Large language models, despite having 100x or 1000x more parameters, are significantly harder to prune directly.

→ **Another Emergent Property?**

A Missing Ingredient

Outlier features affect quantization performance severely in large language models.

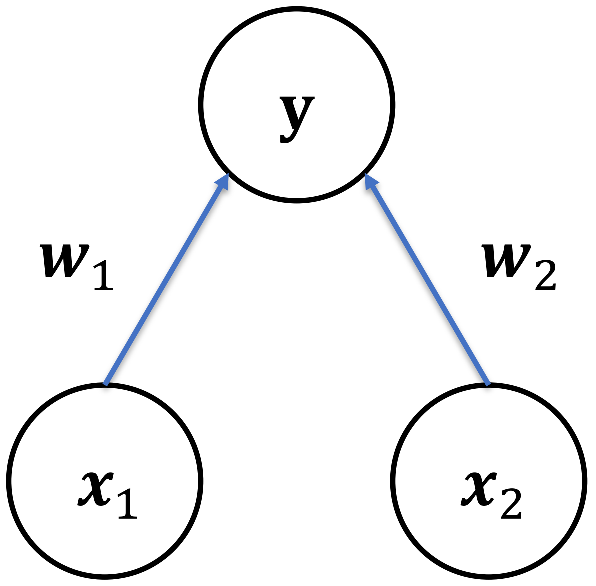




LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. Dettmers et al, 2022.

Activations matter in network pruning

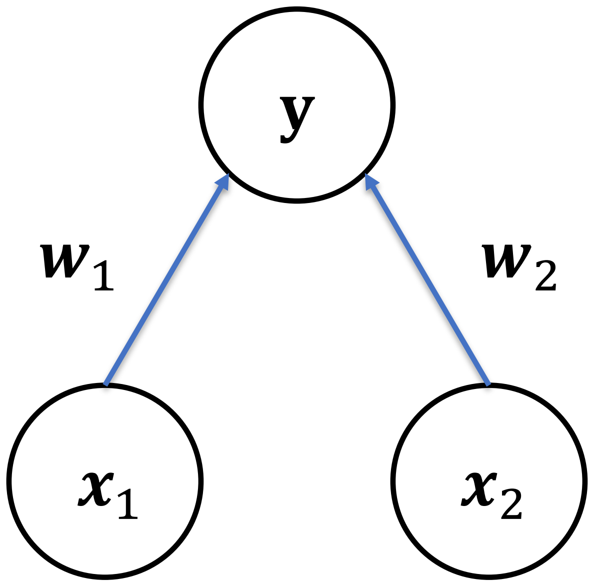
Consider a neuron with two inputs.



Activations matter in network pruning

Magnitude Pruning:

always remove $\boldsymbol{w}_1$, assume $|\boldsymbol{w}_1| < |\boldsymbol{w}_2|$

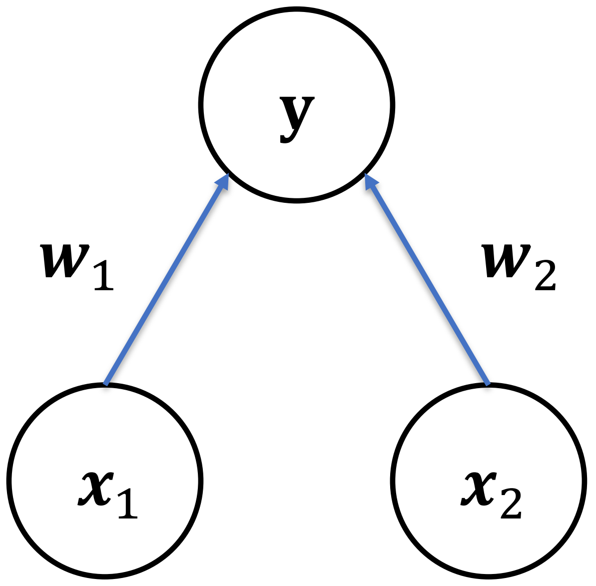


Activations matter in network pruning

Magnitude Pruning:

- always remove $\boldsymbol{w}_1$, assume $|\boldsymbol{w}_1| < |\boldsymbol{w}_2|$

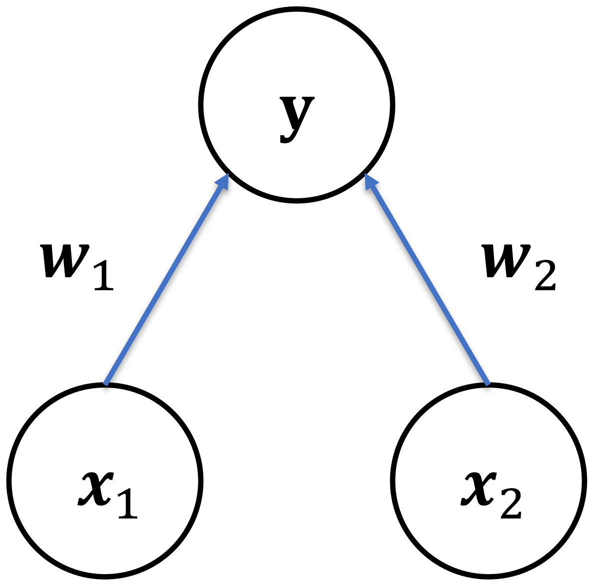
- What if $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ differ significantly in scale?



Limitations of Magnitude Pruning

Limitations of Magnitude Pruning:

No considerations of activations.

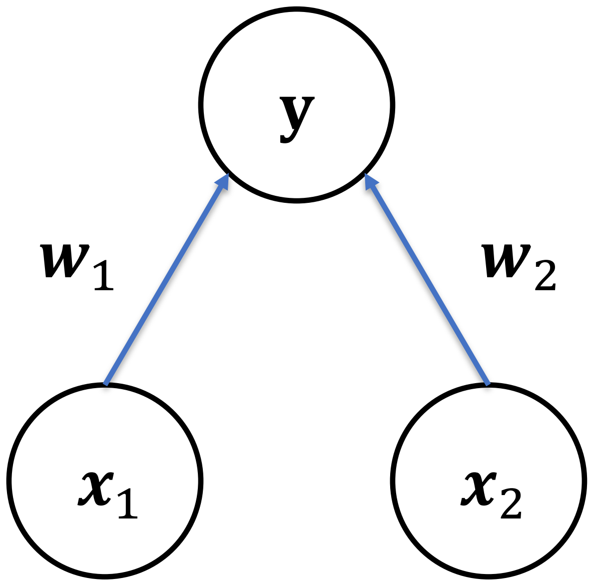


Limitations of Magnitude Pruning

Limitations of Magnitude Pruning:

No considerations of activations.

Activations are just as important as weights.



Our method

$$\mathbf{W} \quad \begin{bmatrix} 4 & 0 & 1 & -1 \\ 3 & -2 & -1 & -3 \\ -3 & 1 & 0 & 2 \end{bmatrix}$$

We propose Wanda: Pruning by **W**eights **and a**ctivations.

Next we show how Wanda would prune this weight.

Weights and Activations

$\mathbf{W}$				
	4	0	1	-1
	3	-2	-1	-3
	-3	1	0	2

$\|\mathbf{X}\|_2$	1	2	8	3

Weights **and a**ctivations

Weights and Activations

Input Dimension

$$\mathbf{W} \quad \begin{bmatrix} 4 & 0 & 1 & -1 \\ 3 & -2 & -1 & -3 \\ -3 & 1 & 0 & 2 \end{bmatrix}$$

Output Dimension

$$\|\mathbf{X}\|_2 \quad \begin{bmatrix} 1 & 2 & 8 & 3 \end{bmatrix}$$

Weights **and** activations

Part 1: Pruning Metric

$$\mathbf{W} = \begin{bmatrix} 4 & 0 & 1 & -1 \\ 3 & -2 & -1 & -3 \\ -3 & 1 & 0 & 2 \end{bmatrix}$$

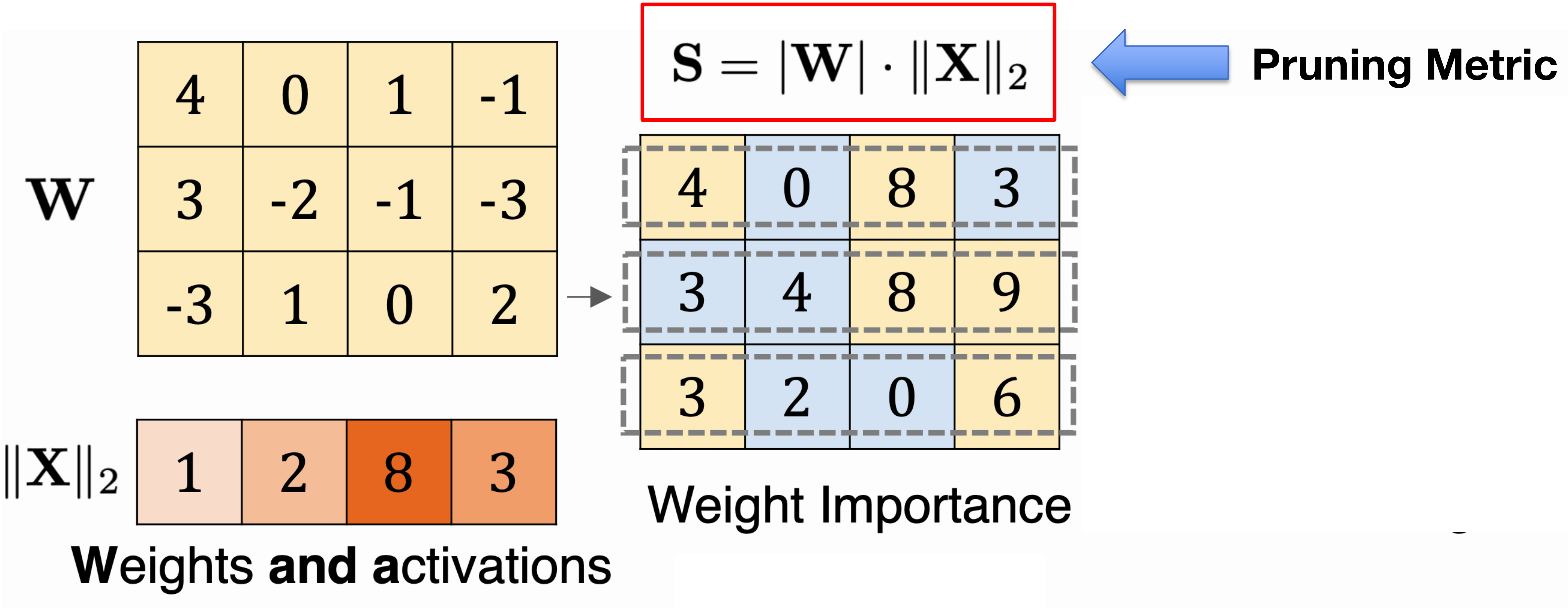
$$\|\mathbf{X}\|_2 = \begin{bmatrix} 1 & 2 & 8 & 3 \end{bmatrix}$$

Weights and activations

$$\boxed{\mathbf{S} = |\mathbf{W}| \cdot \|\mathbf{X}\|_2}$$

Pruning Metric

Part 1: Pruning Metric



Another line of work

Core of GPTQ and SparseGPT:

Layer-wise reconstruction!

GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. Frantar et al, 2023.
SparseGPT: Massive Language Models can be accurately pruned in one-shot. Frantar et al, 2023.

Another line of work

Core of GPTQ and SparseGPT:

Layer-wise reconstruction!

$$\text{argmin}_{\widehat{\mathbf{W}}} \, ||\mathbf{W}\mathbf{X} - \widehat{\mathbf{W}}\mathbf{X}||_2^2$$

Quantized/Sparse weights.

GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. Frantar et al, 2023.
SparseGPT: Massive Language Models can be accurately pruned in one-shot. Frantar et al, 2023.

Another line of work

Effect of removal can be characterized by:

$$\mathbf{S}_{ij} = \left[|\mathbf{W}|^2 / \mathrm{diag}\big((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\big)\right]_{ij}$$

Another line of work

Reduction inspired from Optimal Brain Damage (OBD):

$$\mathbf{S}_{ij} \overset{\lambda=0}{=} \left[|\mathbf{W}|^2 / \mathrm{diag}\Big((\mathbf{X}^T\mathbf{X})^{-1} \Big) \right]_{ij}$$

Another line of work

Reduction inspired from Optimal Brain Damage (OBD):

$$\mathbf{S}_{ij} \overset{\lambda=0}{=} \left[|\mathbf{W}|^2 / \mathrm{diag}\Big((\mathbf{X}^T\mathbf{X})^{-1} \Big) \right]_{ij} \underset{\text{approx.}}{\overset{\text{diagonal}}{=}} \left[|\mathbf{W}|^2 / \Big(\mathrm{diag}(\mathbf{X}^T\mathbf{X}) \Big)^{-1} \right]_{ij} = \left(|\mathbf{W}_{ij}| \cdot \|\mathbf{X}_j\|_2 \right)^2$$

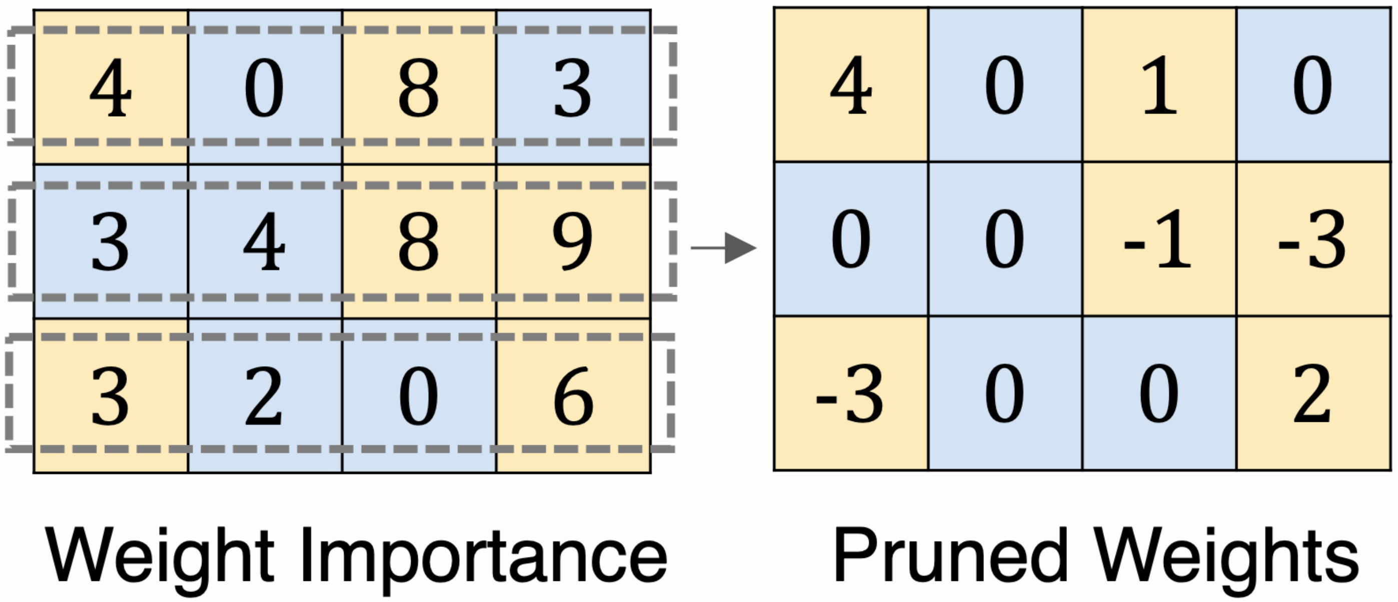
Dropping off-diagonal elements in Hessian.

Part 2: Comparison Group

Compare and remove weights locally inside each output neuron.

Pruning per output

Compare and remove weights for each output neuron.



Comparison Group *grouped per output*

Part 2: Comparison Group

Counter-intuitive.

Better than layer-wise comparisons for LLMs.

		OPT					
Comparison Group	Sparsity	125m	350m	1.3B	2.7B	6.7B	13B
per layer	50%	46.95	38.97	22.20	22.66	15.35	13.54
per output	50%	**38.96**	**36.19**	**19.42**	**14.22**	**11.97**	**11.42**

Part 2: Comparison Group

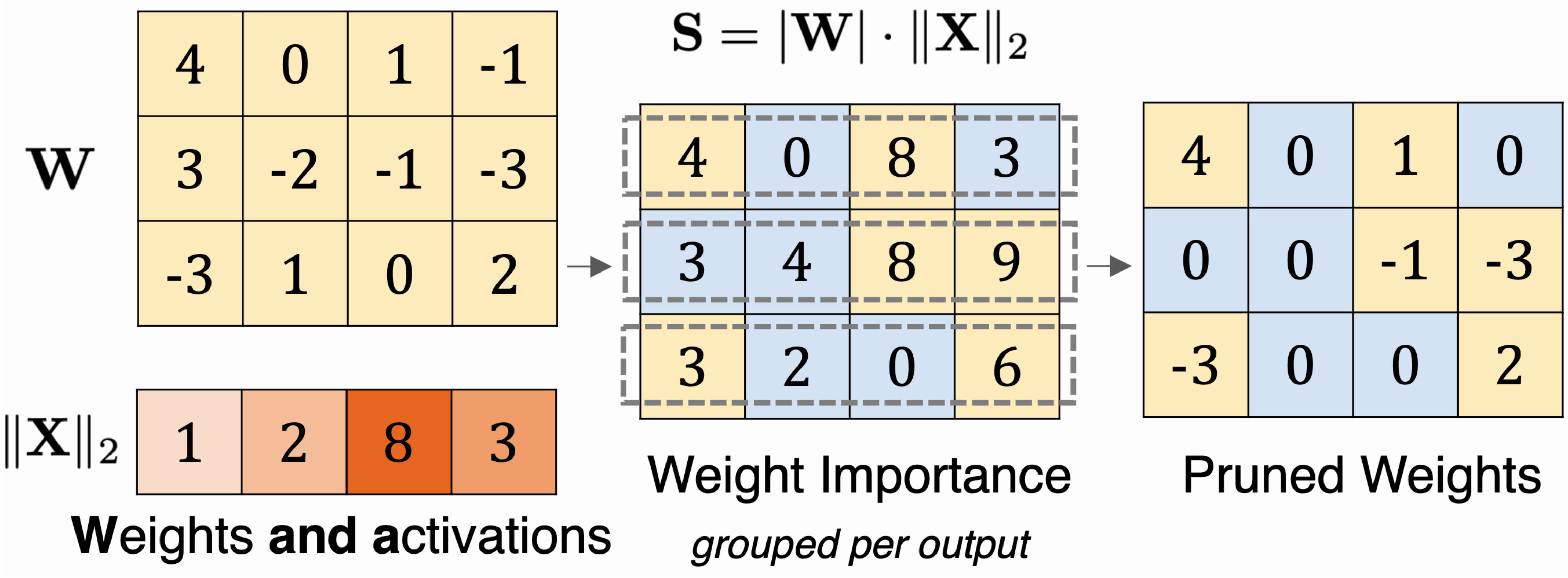
Counter-intuitive.

Better than layer-wise comparisons for LLMs.

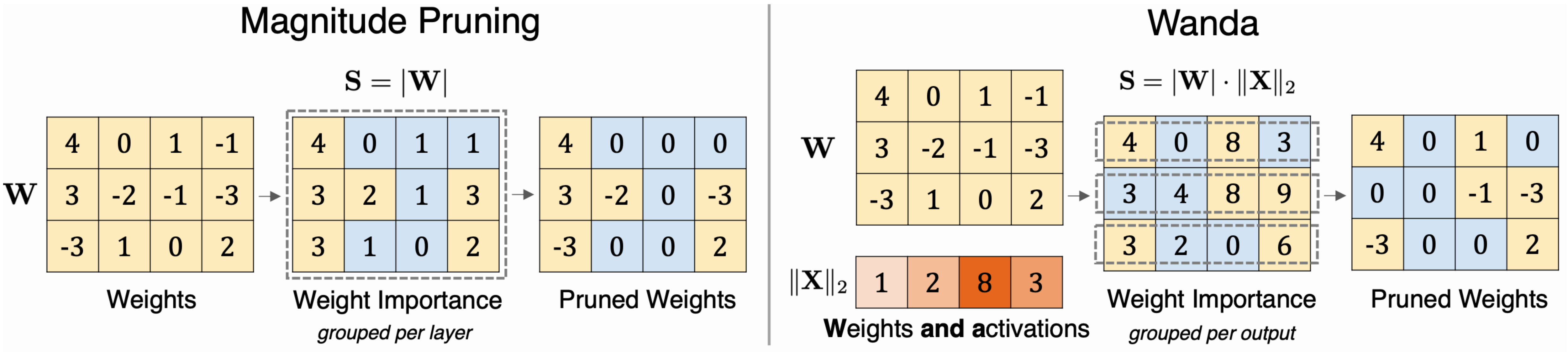
		OPT					
Comparison Group	Sparsity	125m	350m	1.3B	2.7B	6.7B	13B
per layer	50%	46.95	38.97	22.20	22.66	15.35	13.54
per output	50%	**38.96**	**36.19**	**19.42**	**14.22**	**11.97**	**11.42**

→ No idea why!!!

Putting it all together

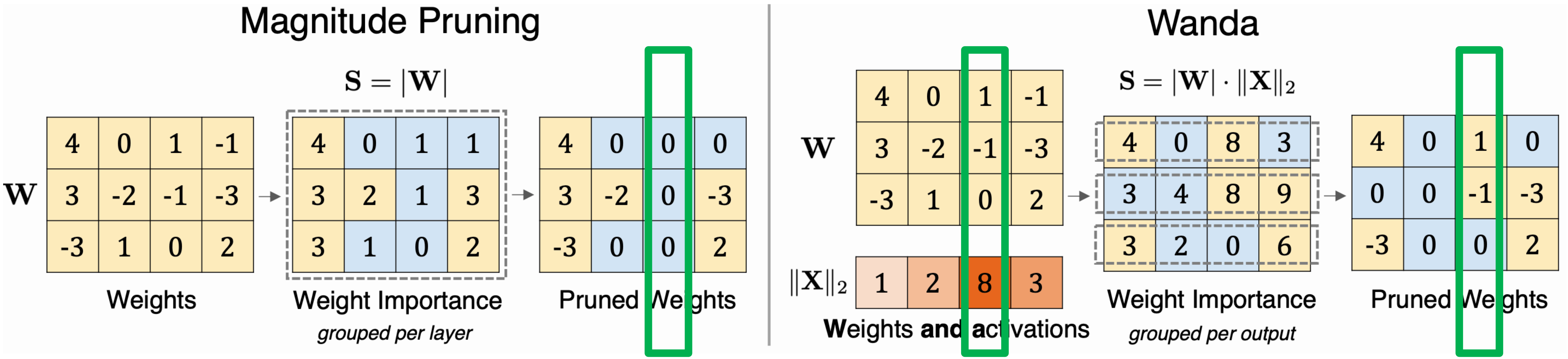


Comparison



Magnitude Pruning
$\mathbf{S} = |\mathbf{W}|$
W
4 0 1 -1
3 -2 -1 -3
-3 1 0 2
Weights
4 0 1 1
3 2 1 3
3 1 0 2
Weight Importance
grouped per layer
4 0 0 0
3 -2 0 -3
-3 0 0 2
Pruned Weights
Wanda
W
4 0 1 -1
3 -2 -1 -3
-3 1 0 2
$\|\mathbf{X}\|_2$
1 2 8 3
Weights and activations
$\mathbf{S} = |\mathbf{W}| \cdot \|\mathbf{X}\|_2$
4 0 8 3
3 4 8 9
3 2 0 6
Weight Importance
grouped per output
4 0 1 0
0 0 -1 -3
-3 0 0 2
Pruned Weights

Comparison



Wanda can preserve outlier features.

In Practice

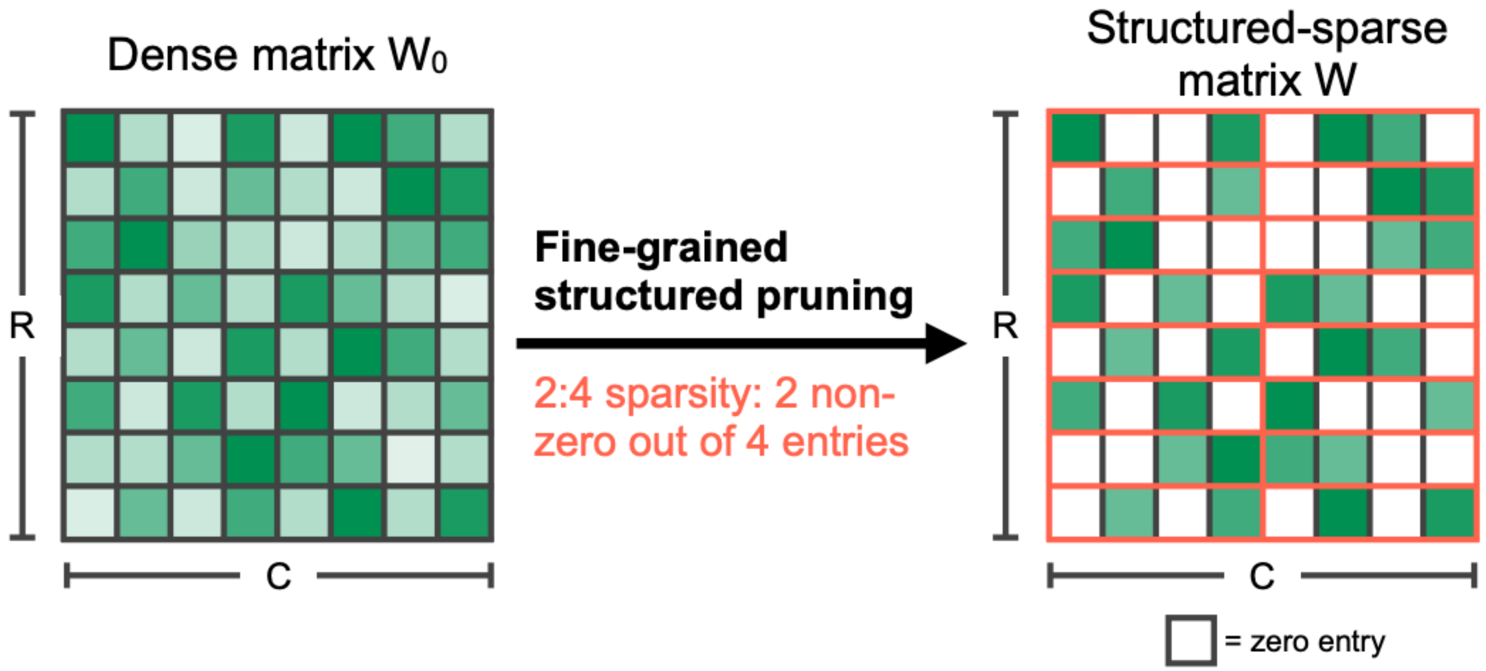
Algorithm 1 PyTorch code for Wanda

```
# W: weight matrix (C_out, C_in);
# X: input matrix (N * L, C_in);
# s: desired sparsity, between 0 and 1;

def prune(W, X, s):
 metric = W.abs() * X.norm(p=2, dim=0)

 _, sorted_idx = torch.sort(metric, dim=1)
 pruned_idx = sorted_idx[:,:int(C_in * s)]
 W.scatter_(dim=1, index=pruned_idx, src=0)

 return W
```

Structured N:M Sparsity

Definition: At most N non-zeros in every contiguous group of M weights.

In practice, 2:4 and 4:8 sparsity.



Accelerating Sparse Deep Neural Networks. Mishra et al, 2021

Zero-Shot

			LLaMA				LLaMA-2		
Method	Weight Update	Sparsity	7B	13B	30B	65B	7B	13B	70B
Dense	-	0%	59.99	62.59	65.38	66.97	59.71	63.03	67.08

Zero-Shot

			LLaMA				LLaMA-2		
Method	Weight Update	Sparsity	7B	13B	30B	65B	7B	13B	70B
Dense	-	0%	59.99	62.59	65.38	66.97	59.71	63.03	67.08
Magnitude	✗	50%	46.94	47.61	53.83	62.74	51.14	52.85	60.93
Wanda	✗	50%	54.21	**59.33**	**63.60**	**66.67**	**56.24**	**60.83**	67.03
Magnitude	✗	4:8	46.03	50.53	53.53	62.17	50.64	52.81	60.28
Wanda	✗	4:8	52.76	**56.09**	**61.00**	**64.97**	52.49	58.75	**66.06**
Magnitude	✗	2:4	44.73	48.00	53.16	61.28	45.58	49.89	59.95
Wanda	✗	2:4	48.53	52.30	**59.21**	**62.84**	48.75	**55.03**	**64.14**

Consistently better than magnitude pruning.

Zero-Shot

			LLaMA				LLaMA-2		
Method	Weight Update	Sparsity	7B	13B	30B	65B	7B	13B	70B
Dense	-	0%	59.99	62.59	65.38	66.97	59.71	63.03	67.08
SparseGPT	✓	50%	**54.94**	58.61	63.09	66.30	**56.24**	60.72	**67.28**
Wanda	✗	50%	54.21	**59.33**	**63.60**	**66.67**	**56.24**	**60.83**	67.03
SparseGPT	✓	4:8	**52.80**	55.99	60.79	64.87	**53.80**	**59.15**	65.84
Wanda	✗	4:8	52.76	**56.09**	**61.00**	**64.97**	52.49	58.75	**66.06**
SparseGPT	✓	2:4	**50.60**	**53.22**	58.91	62.57	**50.94**	54.86	63.89
Wanda	✗	2:4	48.53	52.30	**59.21**	**62.84**	48.75	**55.03**	**64.14**

Wanda performs competitively against SparseGPT.

Perplexity

			LLaMA				LLaMA-2		
Method	Weight Update	Sparsity	7B	13B	30B	65B	7B	13B	70B
Dense	-	0%	5.68	5.09	4.77	3.56	5.12	4.57	3.12
Magnitude	✗	50%	17.29	20.21	7.54	5.90	14.89	6.37	4.98
SparseGPT	✓	50%	**7.22**	6.21	5.31	**4.57**	6.51	5.63	**3.98**
Wanda	✗	50%	7.26	**6.15**	**5.24**	**4.57**	**6.42**	**5.56**	**3.98**
Magnitude	✗	4:8	16.84	13.84	7.62	6.36	16.48	6.76	5.54
SparseGPT	✓	4:8	8.61	**7.40**	6.17	5.38	8.12	6.60	4.59
Wanda	✗	4:8	**8.57**	**7.40**	**5.97**	**5.30**	**7.97**	**6.55**	**4.47**
Magnitude	✗	2:4	42.13	18.37	9.10	7.11	54.59	8.33	6.33
SparseGPT	✓	2:4	**11.00**	**9.11**	7.16	6.28	**10.17**	8.32	5.40
Wanda	✗	2:4	11.53	9.58	**6.90**	**6.25**	11.02	**8.27**	**5.16**

OPT-13B

Method	10%	20%	30%	40%	50%
Magnitude	14.45	9e3	1e4	1e4	1e4
Wanda	10.09	10.07	10.09	10.63	11.42

OPT-13B

Method	10%	20%	30%	40%	50%
Magnitude	14.45	9e3	1e4	1e4	1e4
Wanda	10.09	10.07	10.09	10.63	11.42

→ There exists exact and sparse sub-networks in pre-trained LLMs.

Higher Sparsity

			LLaMA				LLaMA-2		
Method	Weight Update	Sparsity	7B	13B	30B	65B	7B	13B	70B
Dense	-	0%	5.68	5.09	4.77	3.56	5.12	4.57	3.12
Magnitude	✗	80%	1e5	3e4	1e5	2e4	nan	5e4	3e4
SparseGPT	✓	80%	**2e2**	**1e2**	**54.98**	**32.80**	**1e2**	**1e2**	**25.86**
Wanda	✗	80%	5e3	4e3	2e3	2e3	5e3	2e3	1e2

→ Weight update can be helpful in high sparsity regime.

Fine-tuning

Evaluation	Dense	Fine-tuning	50%	4:8	2:4
Zero-Shot	59.99	✗	54.21	52.76	48.53
		LoRA	56.53	54.87	54.46
		Full	**58.15**	**56.65**	**56.19**
Perplexity	5.68	✗	7.26	8.57	11.53
		LoRA	6.84	7.29	8.24
		Full	**5.98**	**6.63**	**7.02**

Pruning Configuration

Pruning Metric	Comparison Group				
	layer	(input, 1)	(input, 128)	(output, 1)	(output, 128)
Magnitude: $\lvert \mathbf{W}_{ij} \rvert$	17.29	**8.86**	16.82	13.41	17.47
SparseGPT: $\left[\lvert \mathbf{W} \rvert^2 / \mathrm{diag}(\mathbf{H}^{-1}) \right]_{ij}$	7.91	8.86	8.02	**7.41**	7.74
Wanda: $\lvert \mathbf{W}_{ij} \rvert \cdot \lVert \mathbf{X}_j \rVert$	7.95	8.86	8.12	**7.26**	7.71

Wanda's pruning configuration is optimal.

Pruning Efficiency

	LLaMA			
Method	7B	13B	30B	65B
SparseGPT	203.1	339.0	810.3	1353.4
Wanda	**0.54**	**0.91**	**2.9**	**5.6**

Pruning Efficiency

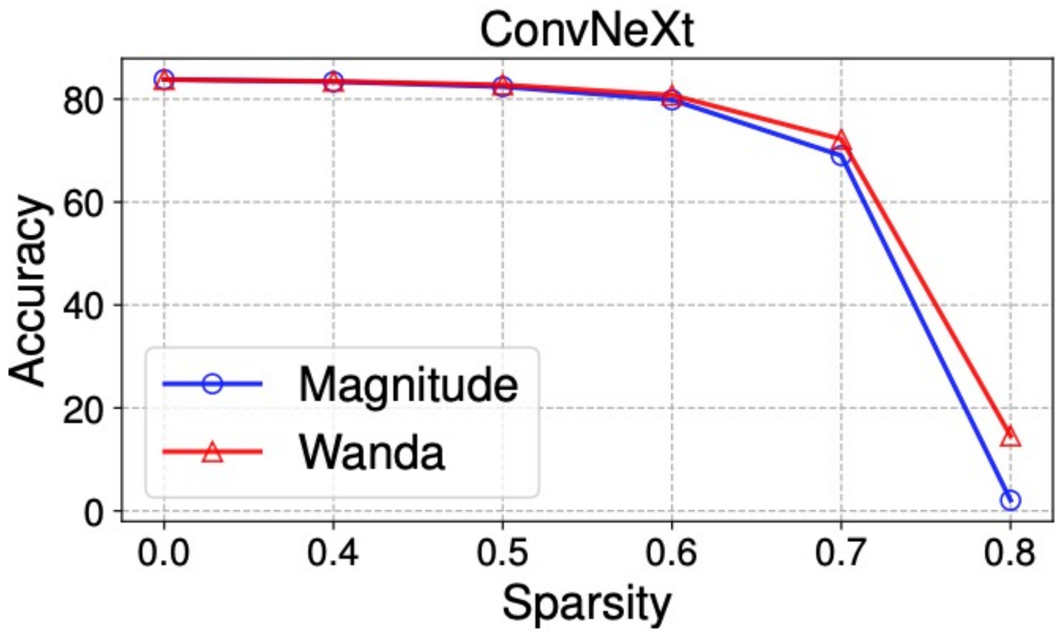
	LLaMA			
Method	7B	13B	30B	65B
SparseGPT	203.1	339.0	810.3	1353.4
Wanda	**0.54**	**0.91**	**2.9**	**5.6**

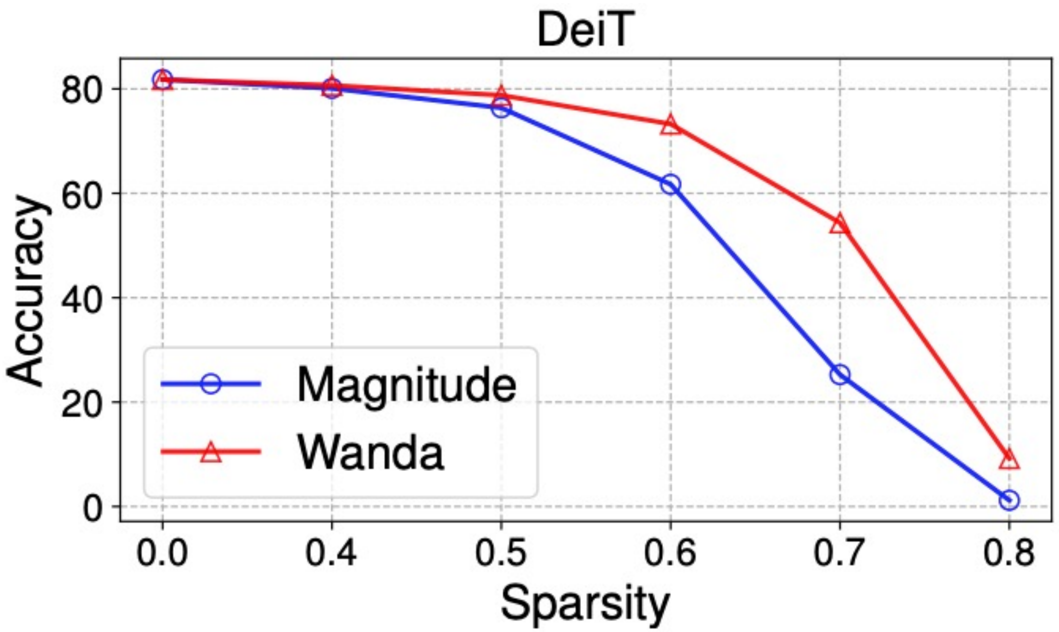
→ Critical when pruning needs to be performed real-time.

A general pruning method?

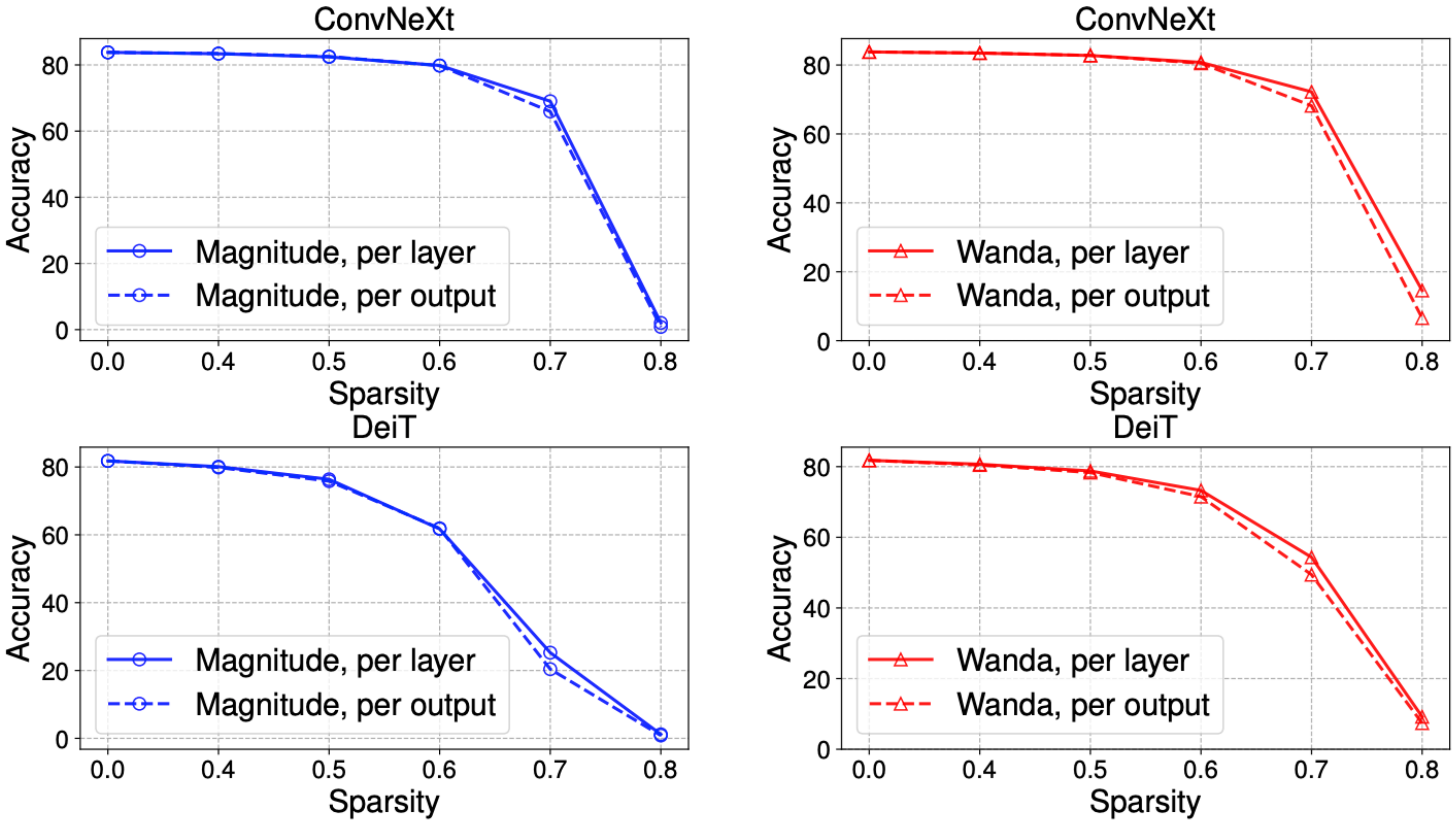
ImageNet Classification.

ConvNeXt and DeiT.





Wanda’s pruning metric is consistently better than weight magnitude.



Our observation on pruning per output does not hold in general.

Summary

Activations are just as important as weights for network pruning.

Summary

Activations are just as important as weights for network pruning.

We demonstrate this on pruning large language models.

Weights are pruned according to two principles:

- magnitude multiplied by input activation norms
- comparing weights on a *per output* basis.

It can find effective *exact* sparse networks in pretrained LLMs.