Edge Computing

Lecture 06: Quantization and Pruning

Recap

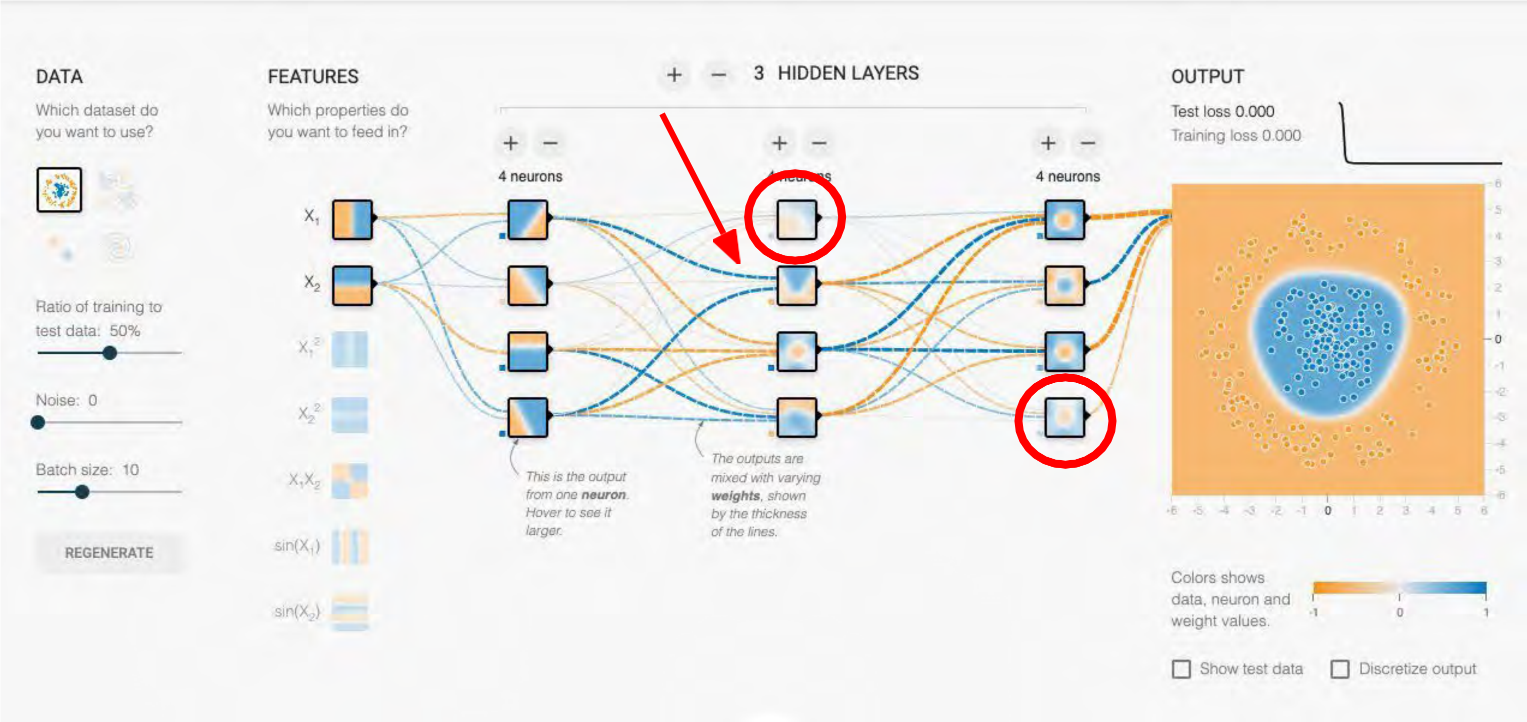
- Neural network: terminology
  - Neuron, Synapse, Activation, Weight, etc.
- Common building block: layer
  - FC, Conv, Conv2D, Depthwise Conv
  - Receptive field, padding, strides
  - Pooling, Normalization
- Convolution neural network
  - Alexnet, VGG16, MobileNetv2
  - Tensorflow, Tensorflow Lite

Agenda

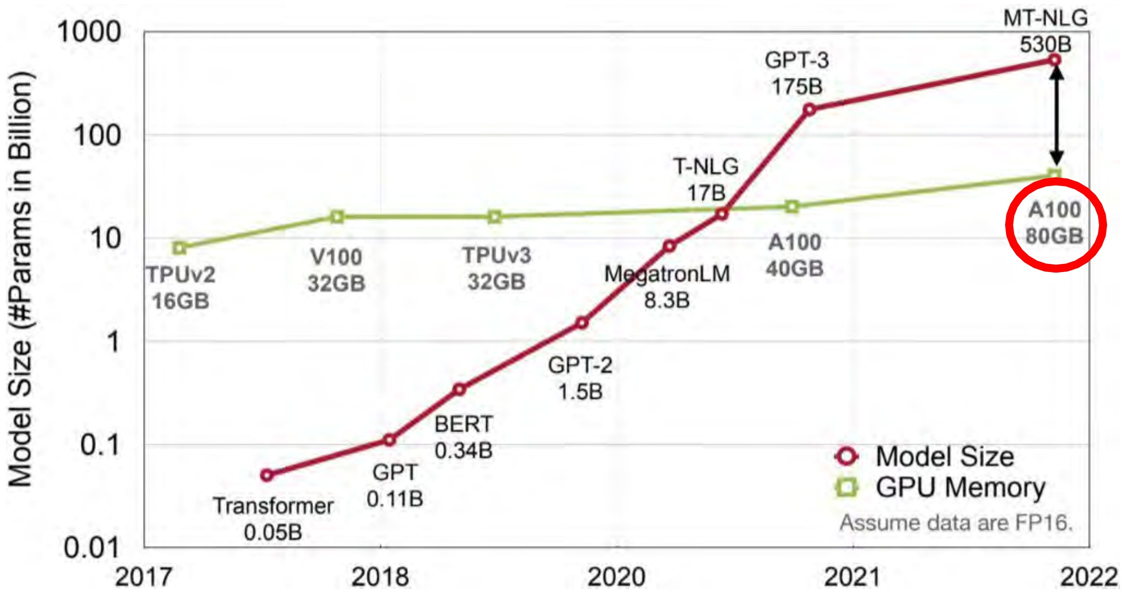
- Pruning
  - What and why?
  - How? Pruning criterion
  - Fine-tune/retrain acceleration
- Quantization
  - What and why?
  - Linear quantization
  - Quantization granularity
  - Calibration and Clipping

What is Pruning?

- Shrinking models by removing synapses and neurons.



Why pruning?



https://hanlab.mit.edu/courses/2023-fall-65940

Why pruning? Model Footprint

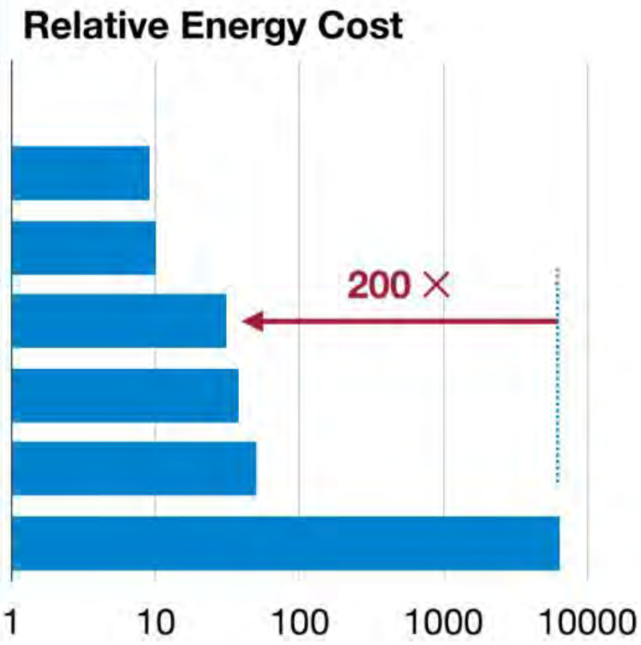
- Nvidia Jetson
    - Custom boards vs Dev boards

https://connecttech.com/jetson/jetson-module-comparison/ & Song, Han, EfficientML.ai

Why pruning? Energy

Operation	Energy [pJ]
32 bit int ADD	0.1
32 bit float ADD	0.9
32 bit Register File	1
32 bit int MULT	3.1
32 bit float MULT	3.7
32 bit SRAM Cache	5
32 bit DRAM Memory	640

Rough Energy Cost For Various Operations in 45nm 0.9V



Computing's Energy Problem (and What We Can Do About it) [Horowitz, M., IEEE ISSCC 2014]

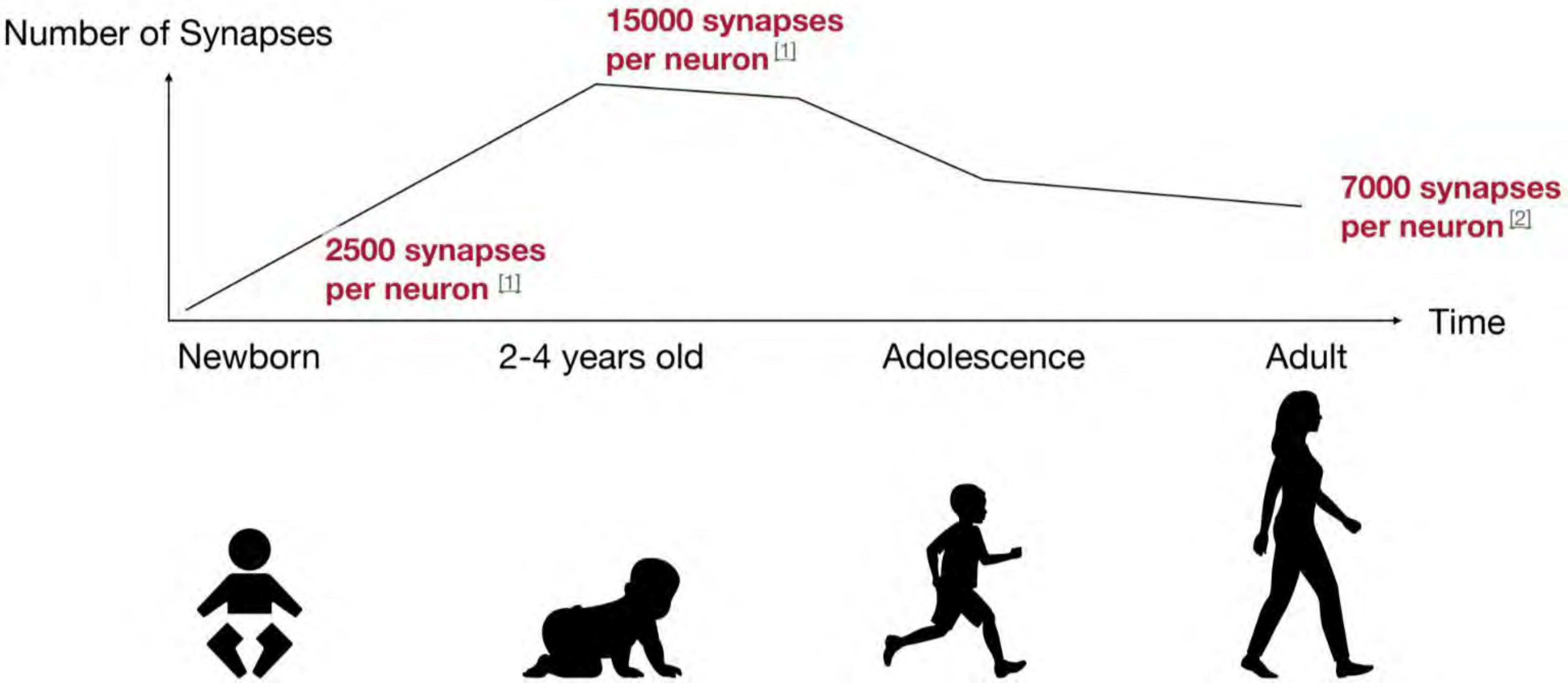
Smaller network, will it work?

- MAC, Multiply-ACCumulation operations ~ FLOPS

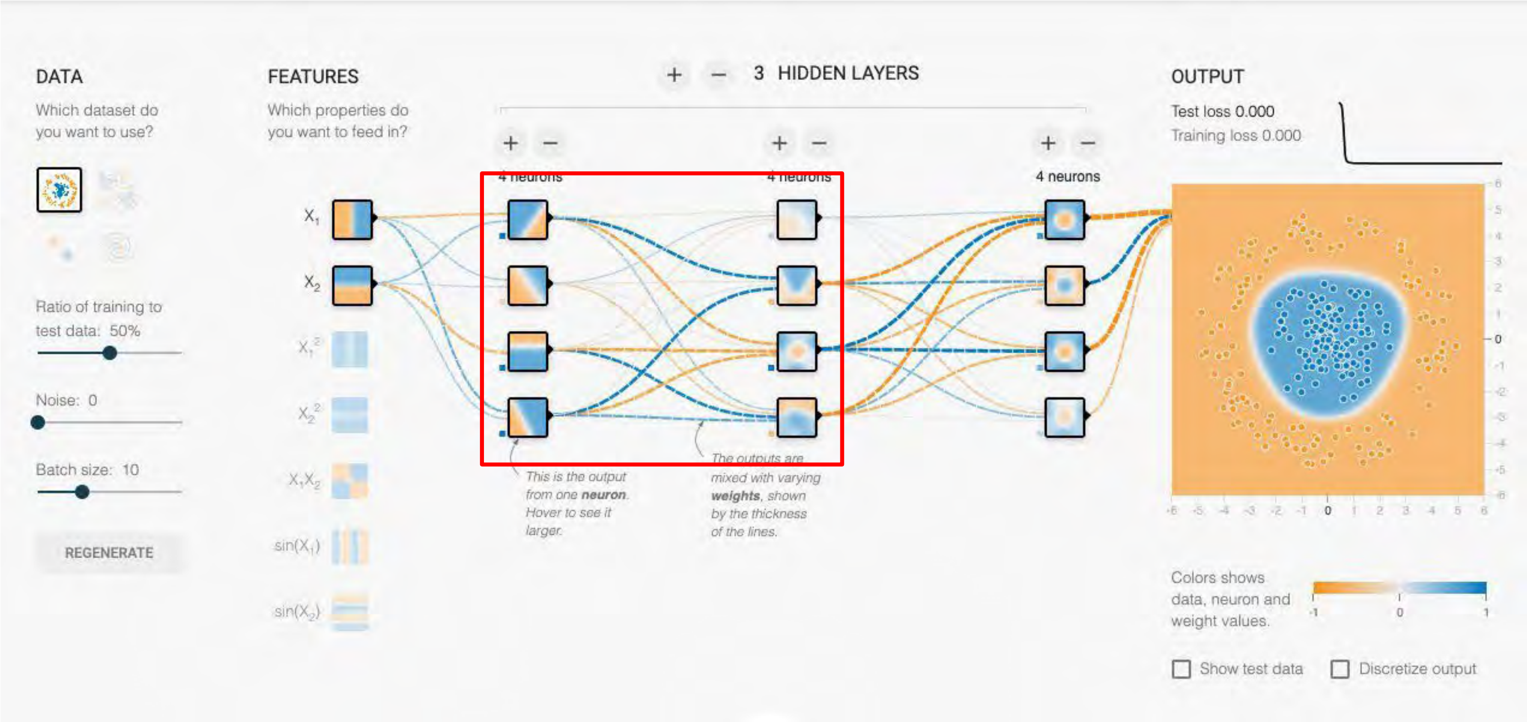
Neural Network	#Parameters			MACs
	Before Pruning	After Pruning	Reduction	Reduction
AlexNet	61 M	6.7 M	**9** ×	3 ×
VGG-16	138 M	10.3 M	**12** ×	5 ×
GoogleNet	7 M	2.0 M	**3.5** ×	5 ×
ResNet50	26 M	7.47 M	**3.4** ×	6.3 ×
SqueezeNet	1 M	0.38 M	**3.2** ×	3.5 ×

https://hanlab.mit.edu/courses/2023-fall-65940

Human brain prunes too!

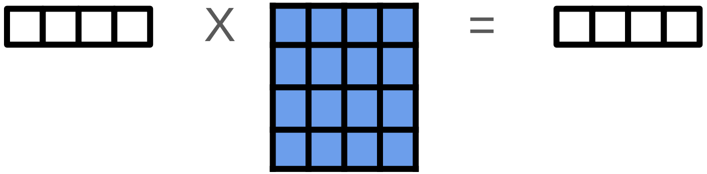


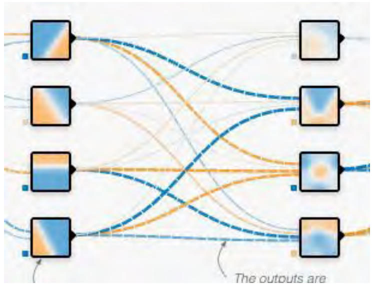
How to prune?



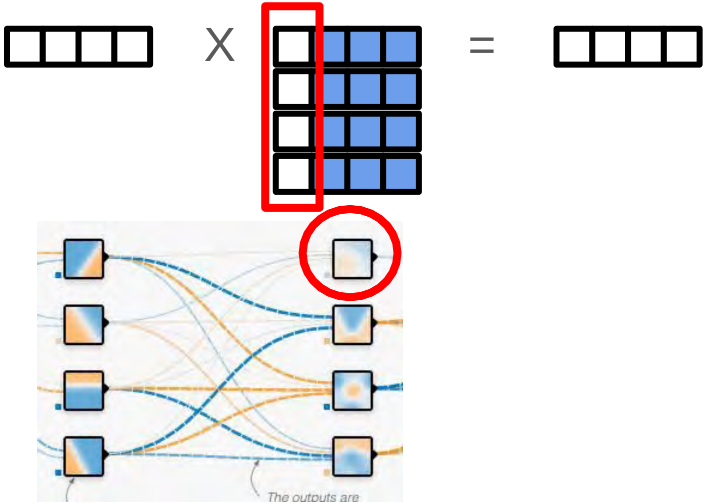
DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE
FEATURES
Which properties do you want to feed in?
3 HIDDEN LAYERS
4 neurons
4 neurons
4 neurons
This is the output from one **neuron**. Hover to see it larger.
The outputs are mixed with varying **weights**, shown by the thickness of the lines.
OUTPUT
Test loss 0.000
Training loss 0.000
Colors shows data, neuron and weight values.
Show test data
Discretize output

How to prune?

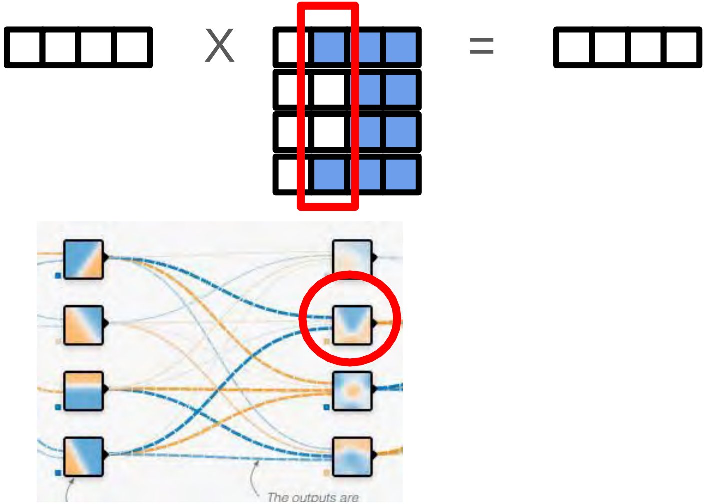




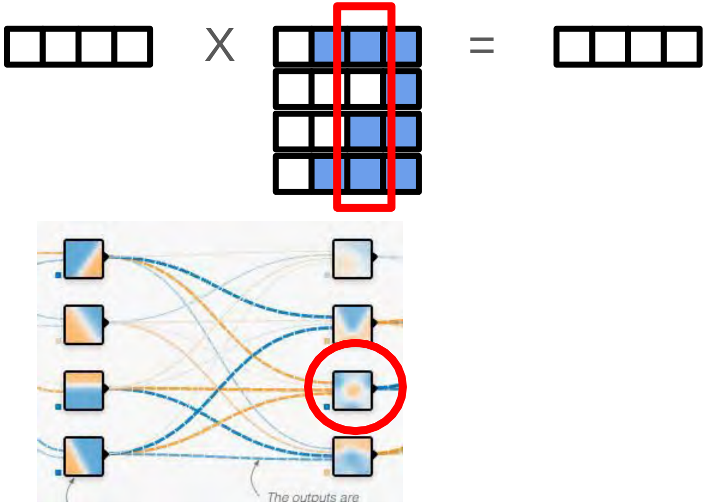
How to prune?



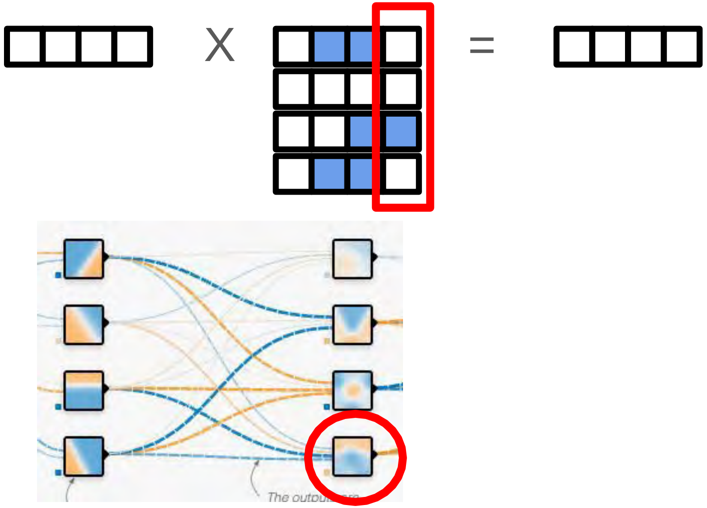
How to prune?



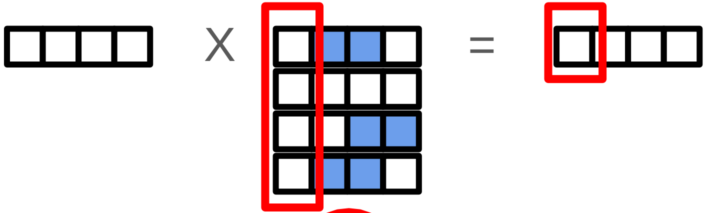
How to prune?

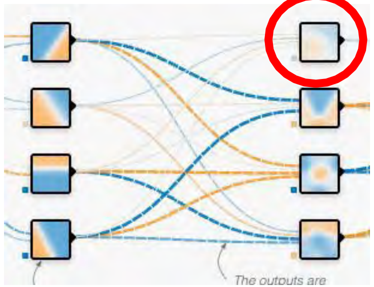


How to prune?

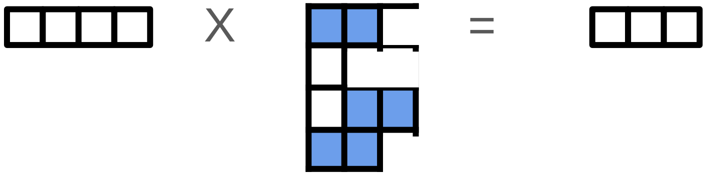


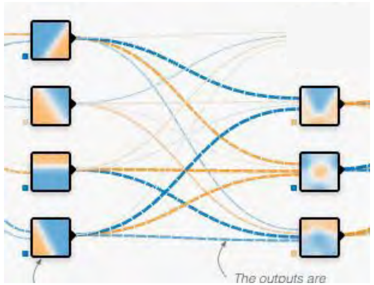
How to prune?



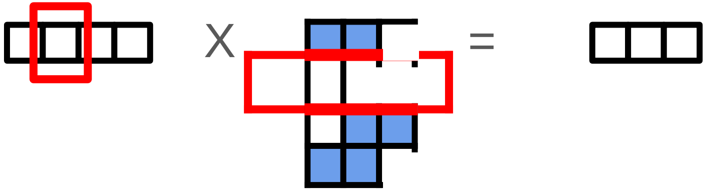


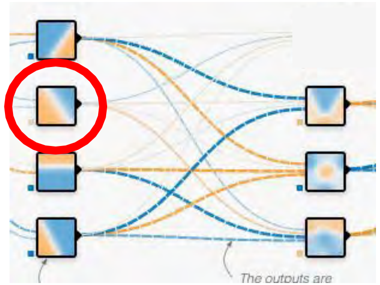
How to prune?



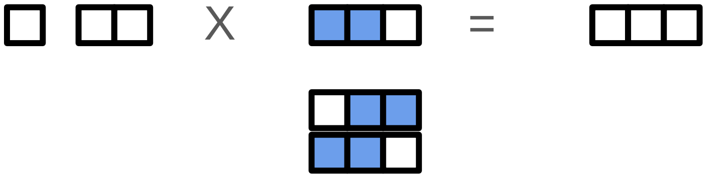


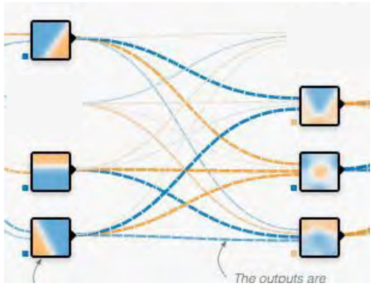
How to prune?



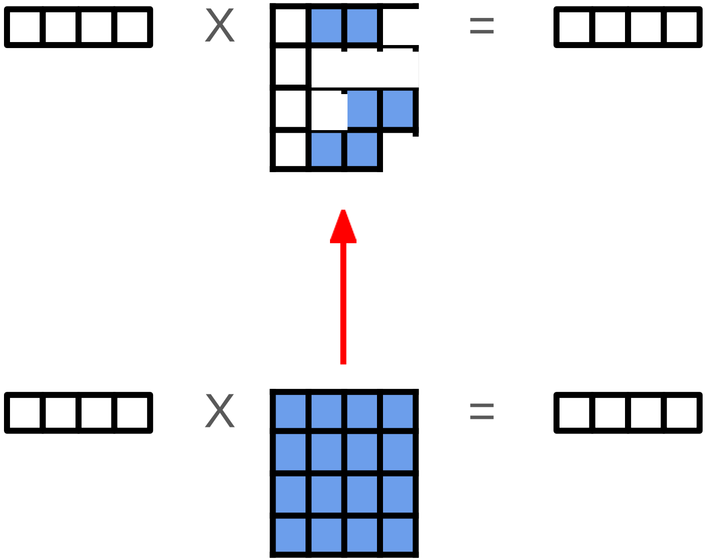


How to prune?

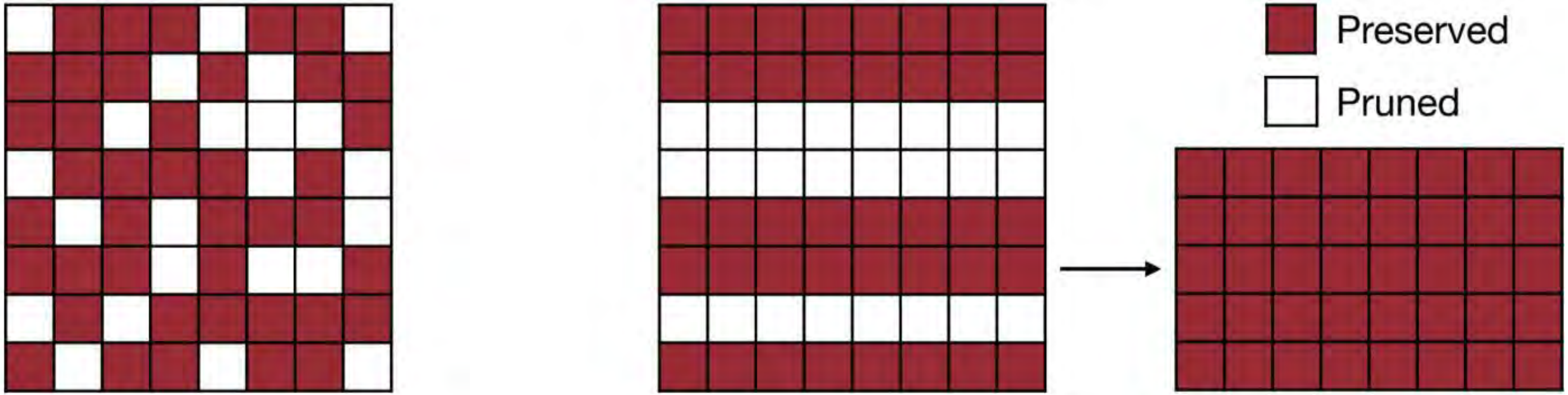




How to prune?



Pruning Granularity



Fine-grained/Unstructured

- More flexible pruning index choice
- Hard to accelerate (irregular)

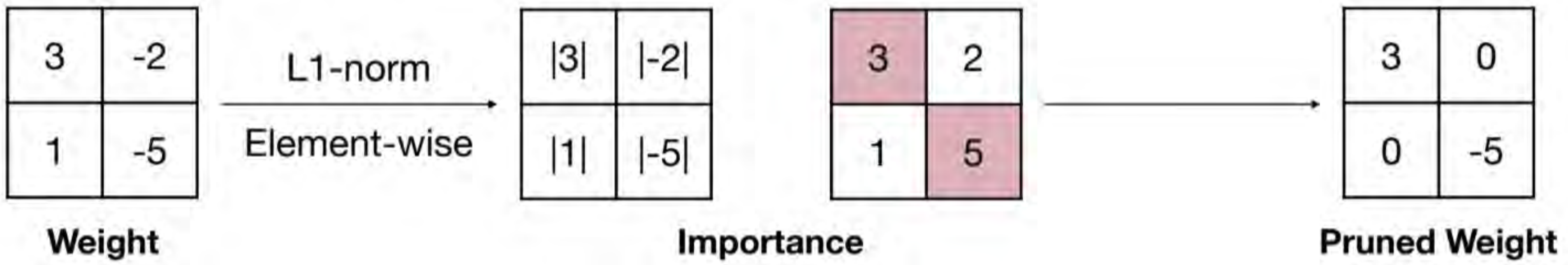
Coarse-grained/Structured

- Less flexible pruning index choice (a subset of the fine-grained case)
- Easy to accelerate (just a smaller matrix!)

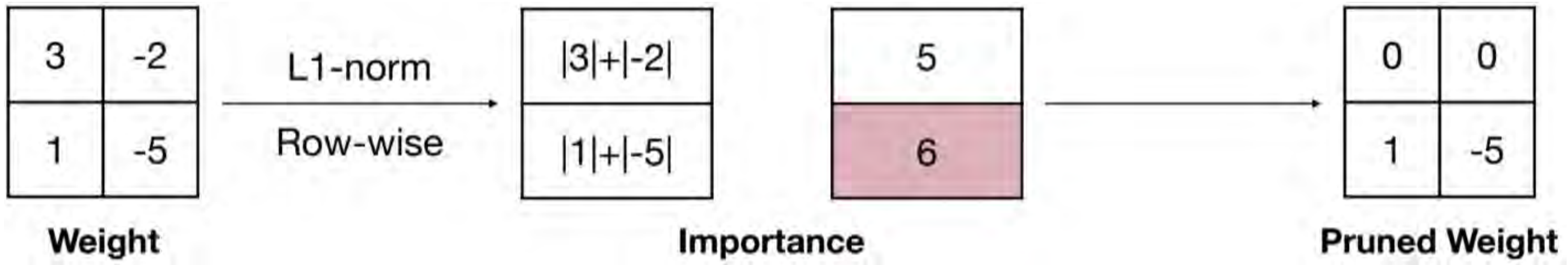
https://hanlab.mit.edu/courses/2023-fall-65940

Magnitude-based Pruning

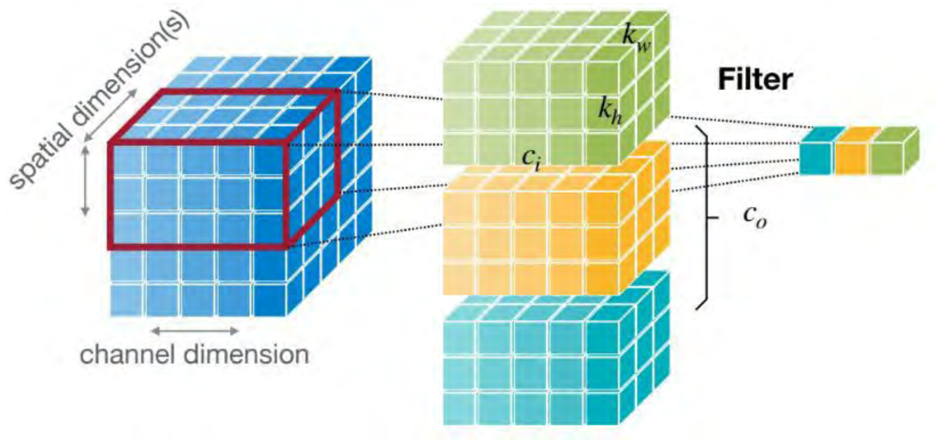
- Fine-grained



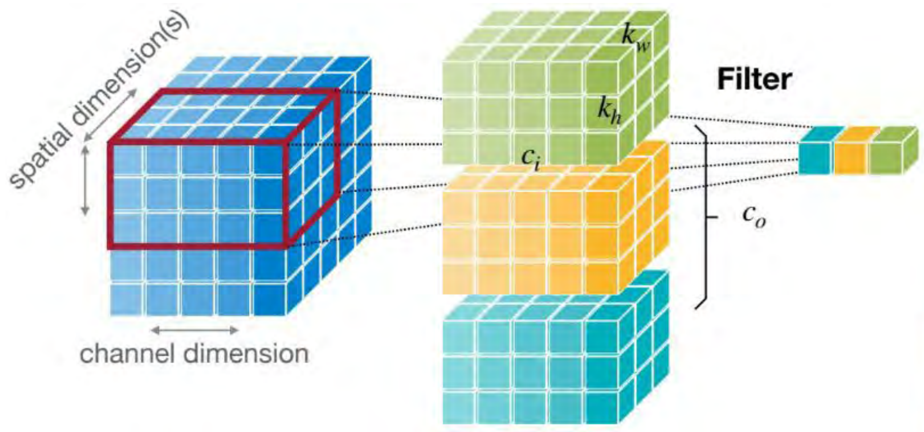
- Coarse-grained / structured

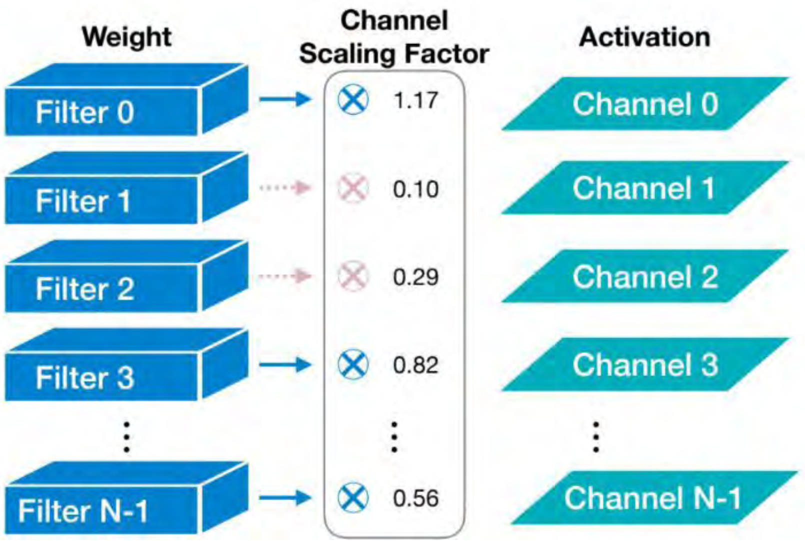


How to prune CNN?



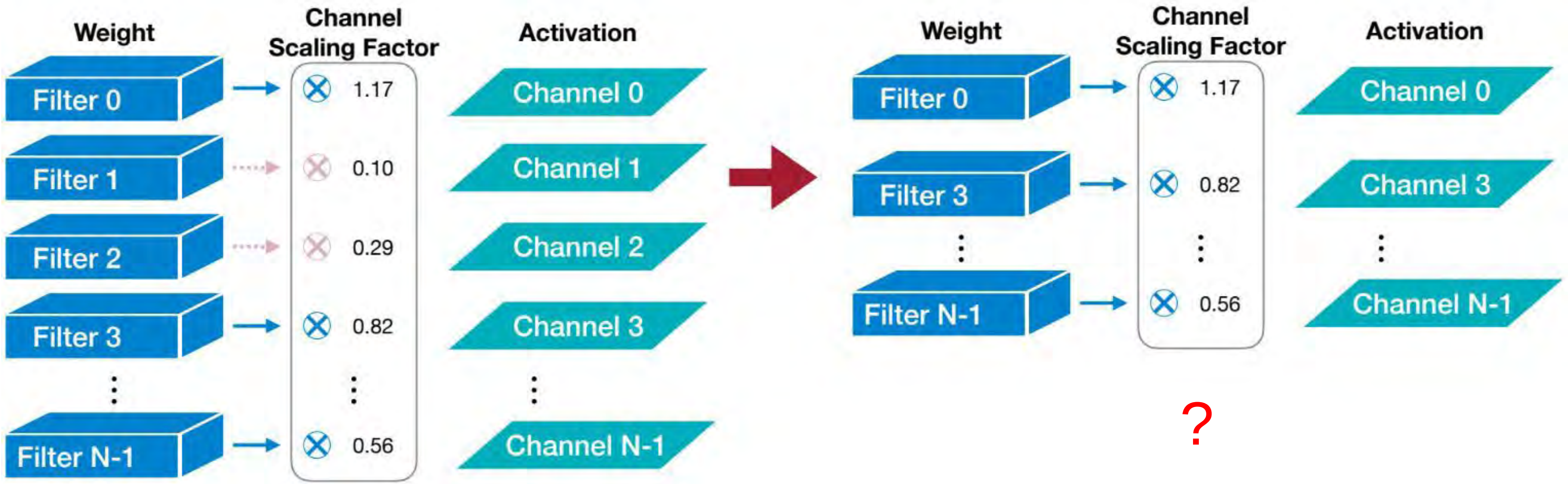
Scaling-based Pruning





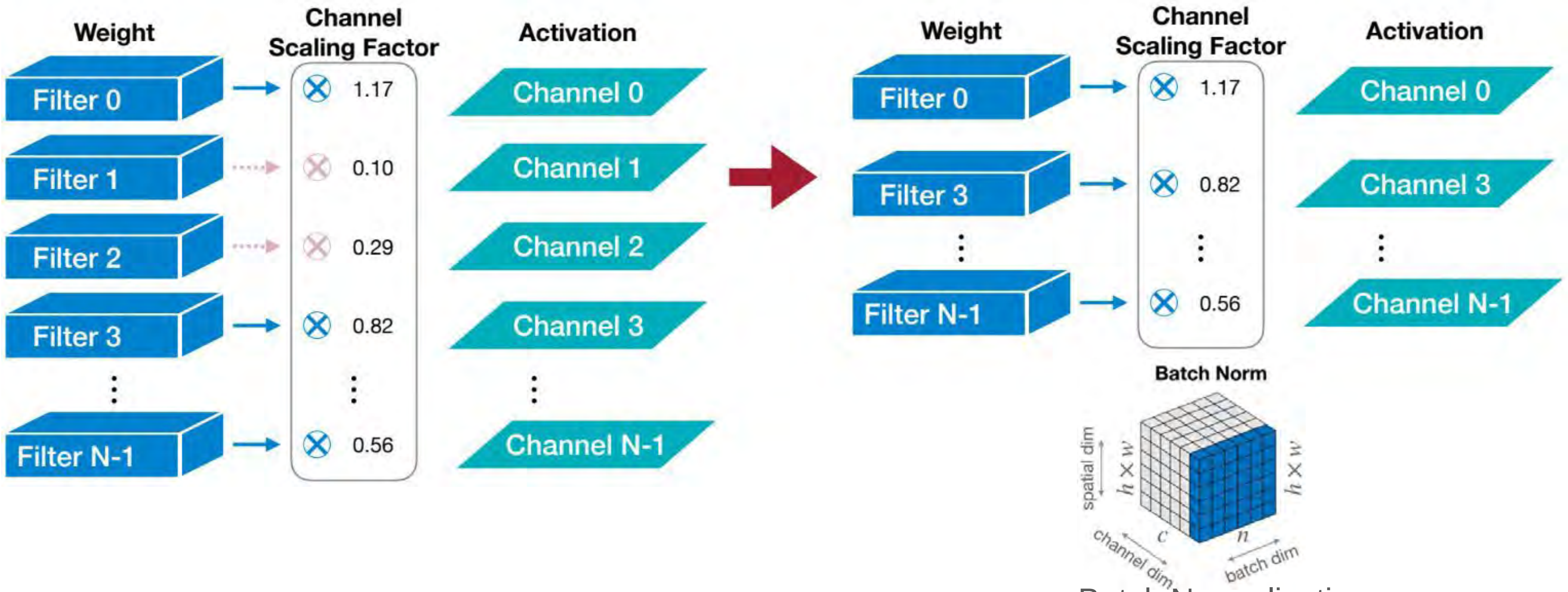
Learning Efficient Convolutional Networks through Network Slimming [Liuet al.,ICCV 2017]

Scaling-based Pruning



Learning Efficient Convolutional Networks through Network Slimming [Liuet al.,ICCV 2017]

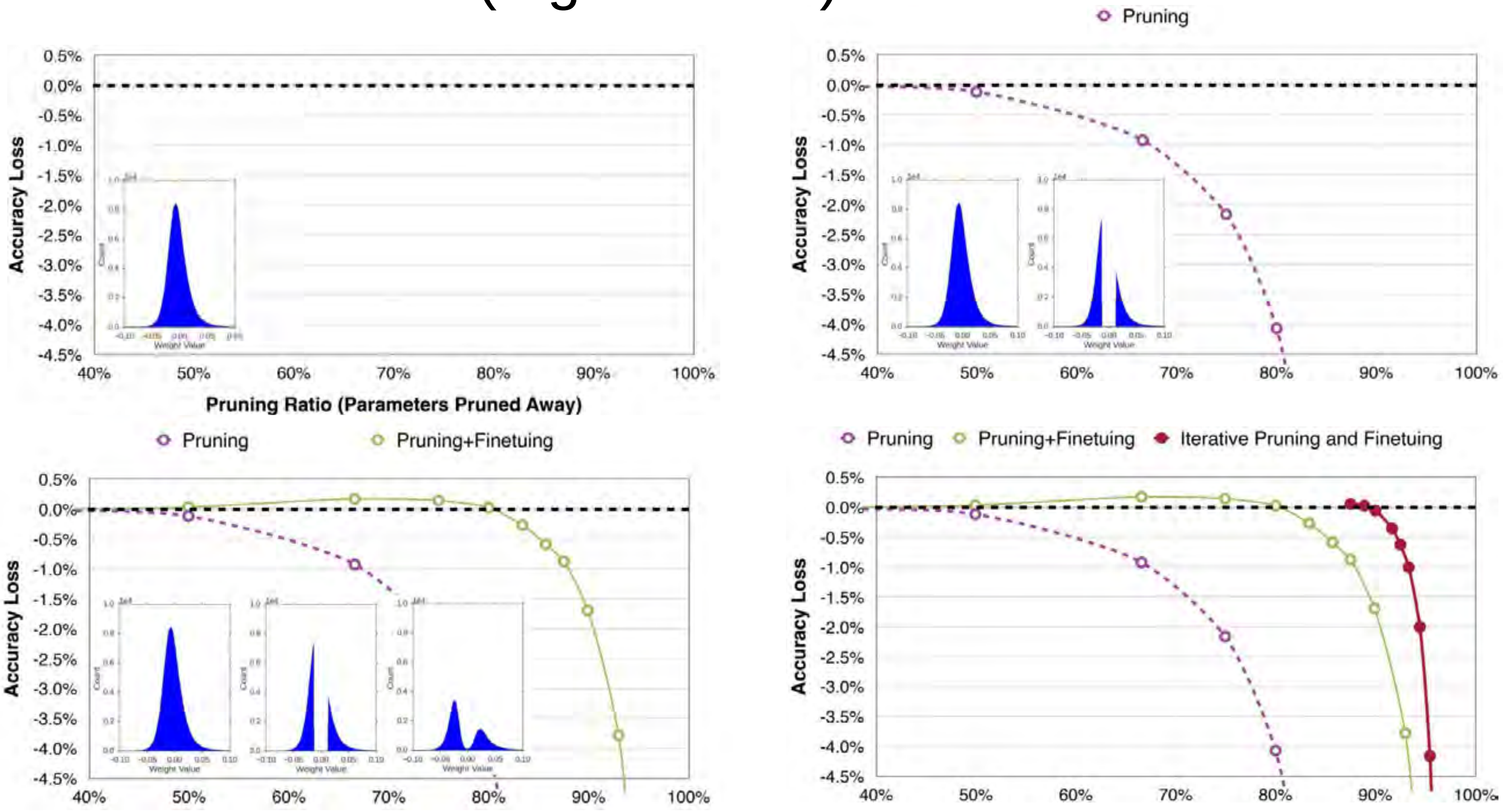
Scaling-based Pruning



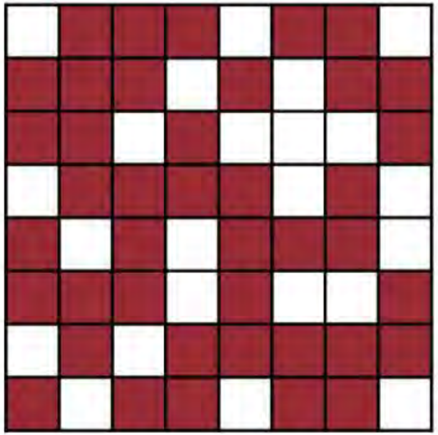
Batch Normalization scaling factor

Learning Efficient Convolutional Networks through Network Slimming [Liuet al.,ICCV 2017]

Fine-tune/Retrain (e.g. AlexNet)



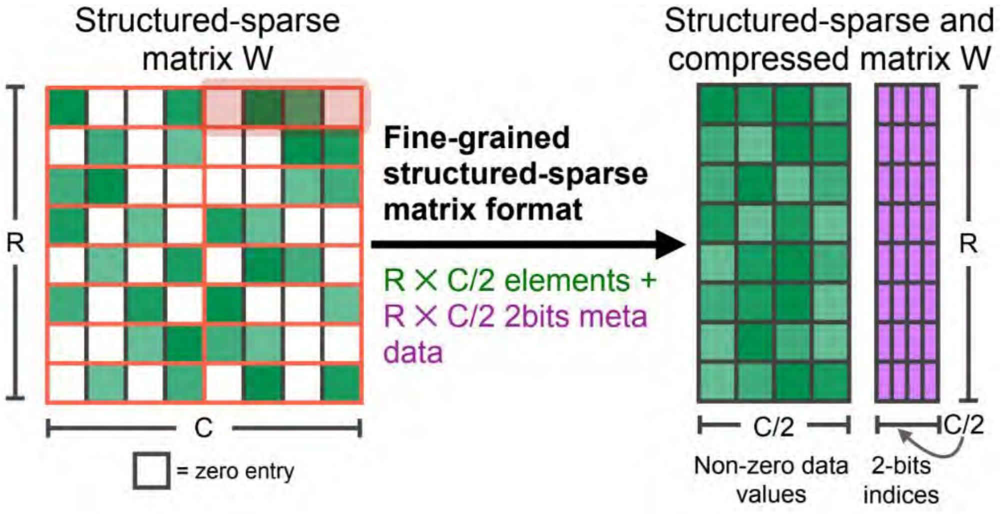
Q: How to accelerate irregular pruning?



Fine-grained/Unstructured

- More flexible pruning index choice
- Hard to accelerate (irregular)

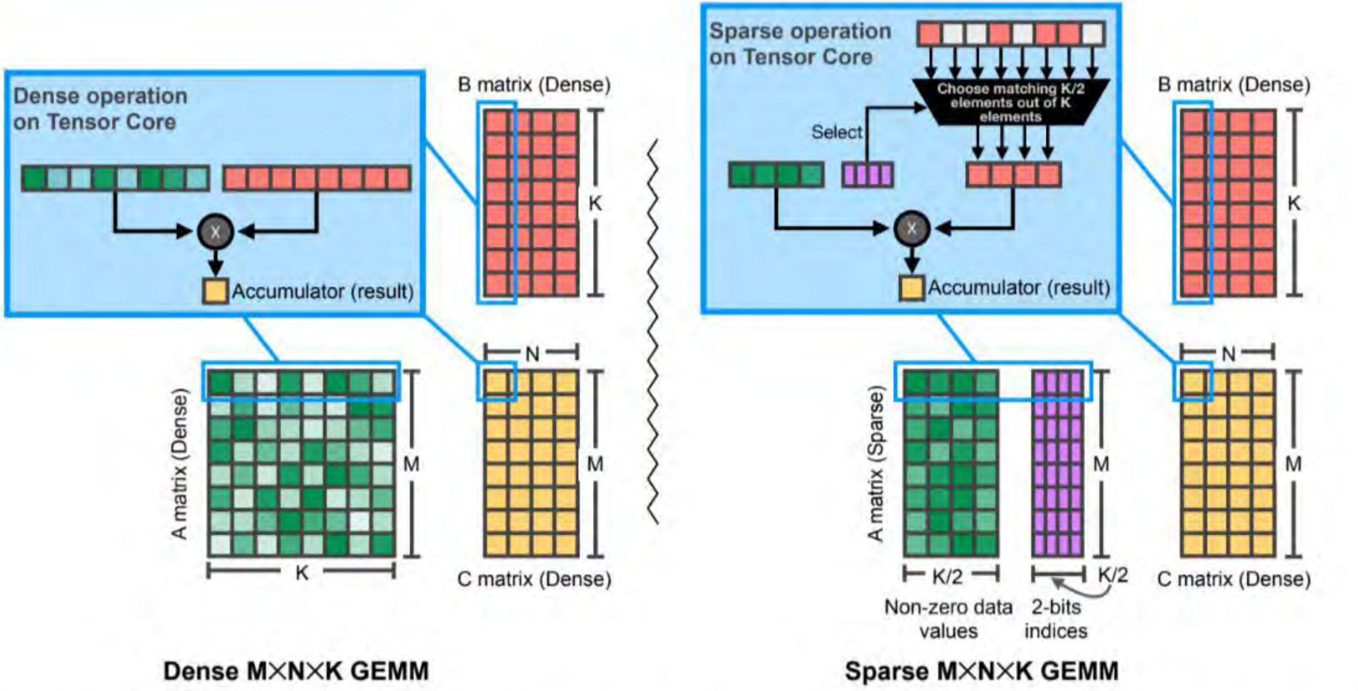
M:N Sparsity



Two weights are nonzero out of four consecutive weights (2:4 sparsity).

Accelerating Sparse Deep Neural Networks[Mishraet al.,arXiv 2021

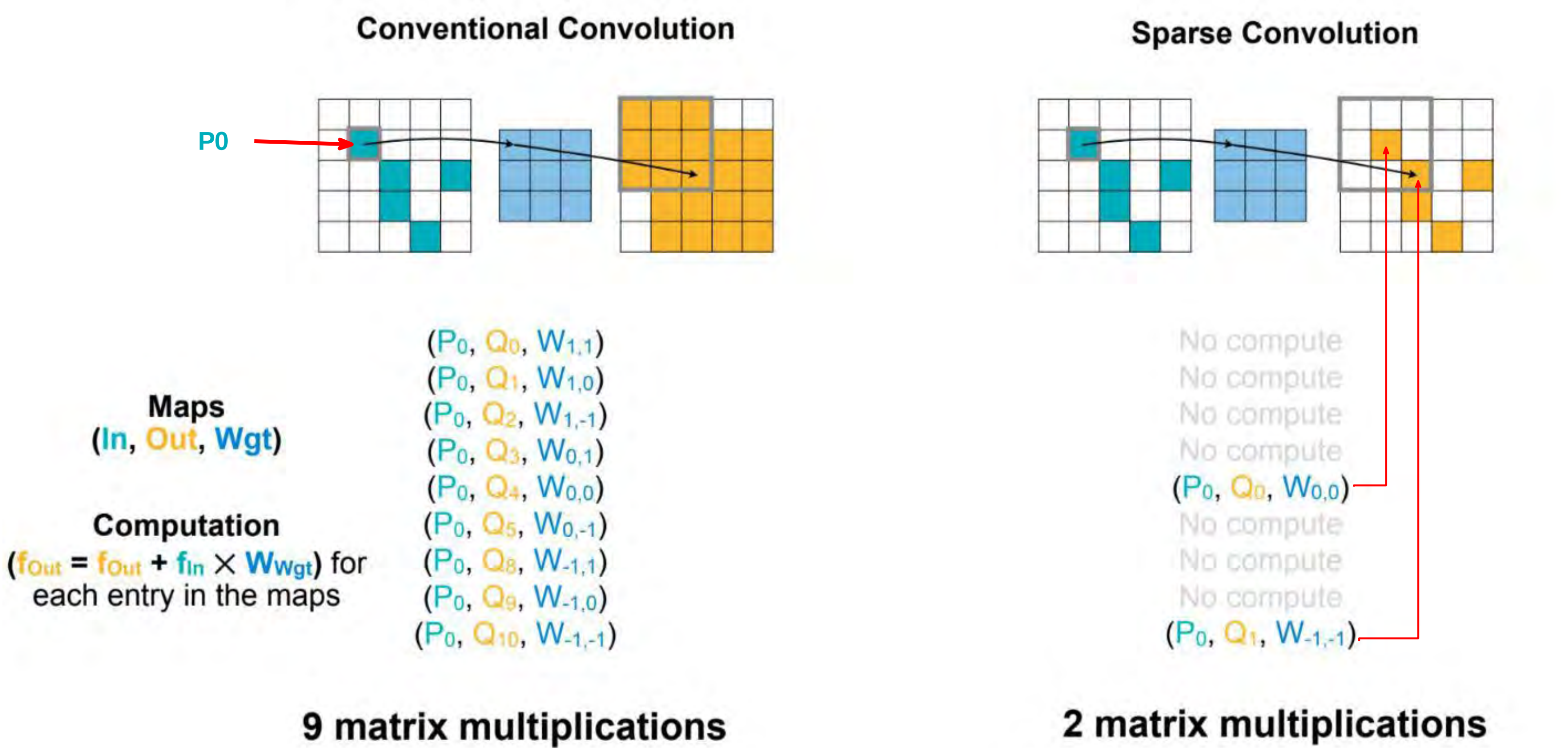
M:N Sparsity



The indices are used to mask out the inputs. Only 2 multiplications will be done out of four.

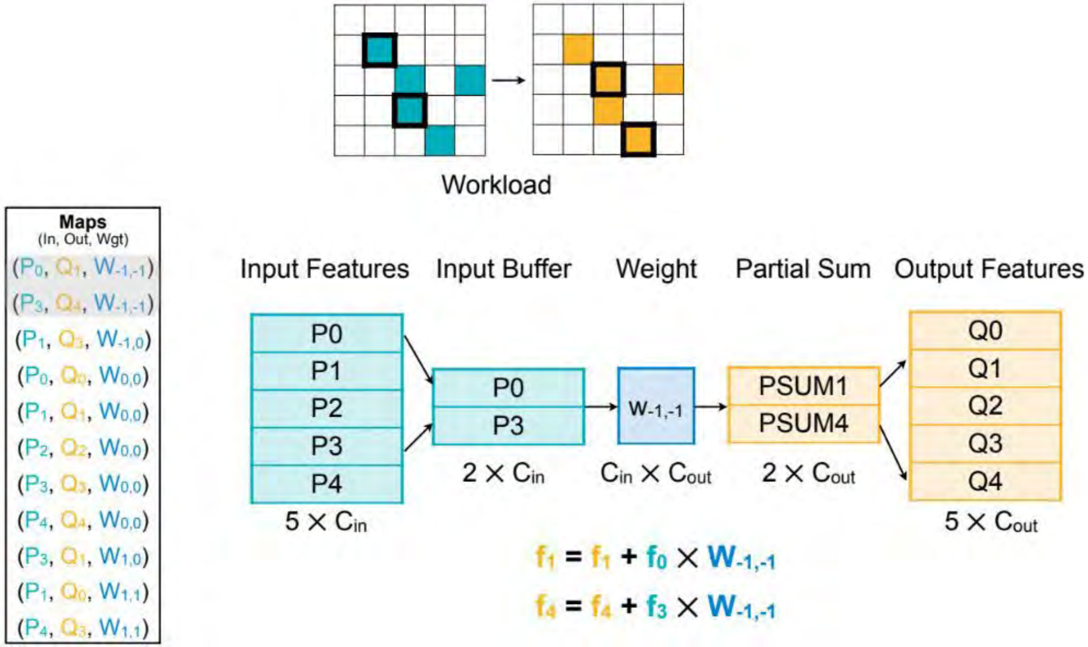
Accelerating Sparse Deep Neural Networks[Mishraet al.,arXiv 2021

Sparse Conv



TorchSparse: Efficient Point Cloud Inference Engine [Tanget al., MLSys 2022]

Sparse Conv



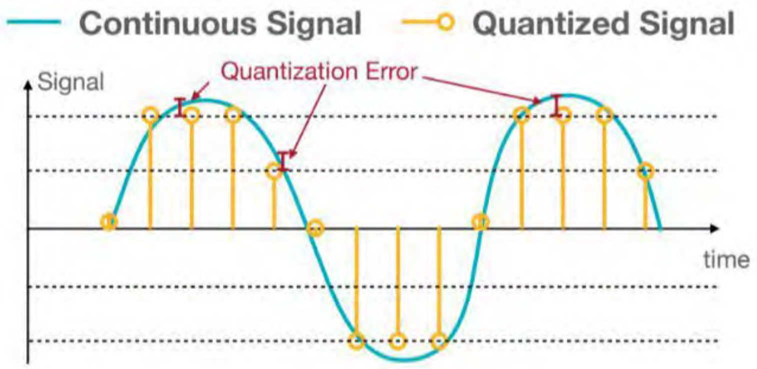
TorchSparse: Efficient Point Cloud Inference Engine [Tanget al., MLSys 2022]

Agenda

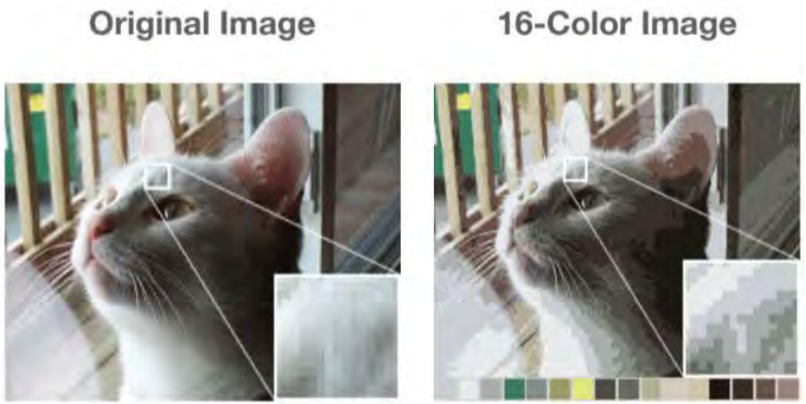
- Pruning
    - What and why?
    - How? Pruning criterion
    - Fine-tune/retrain acceleration
- Quantization
    - What and why?
    - Linear quantization
    - Quantization granularity
    - Calibration and Clipping

What is quantization?

- Quantization is the process of constraining an input from a continuous or otherwise large set of values to a discrete set.



The difference between an input value and its quantized value is referred to as quantization error.



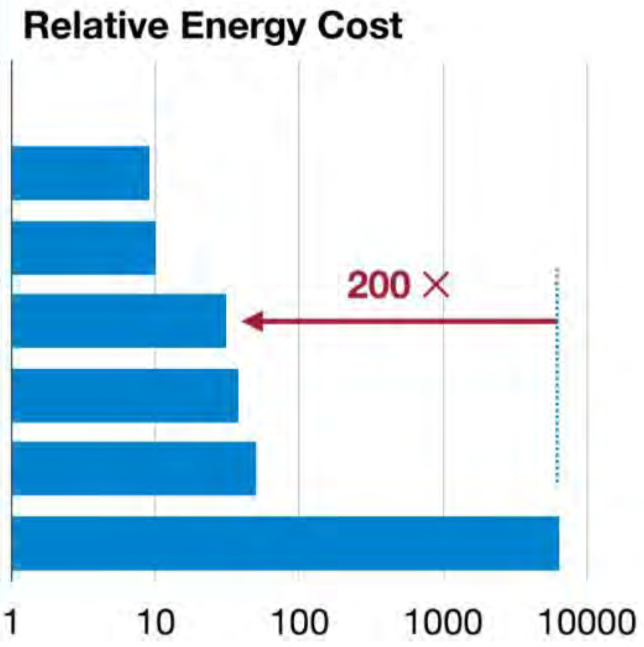
Images are in the public domain.

"Palettization"

Why quantization? Small footprint & low energy

Operation	Energy [pJ]
32 bit int ADD	0.1
32 bit float ADD	0.9
32 bit Register File	1
32 bit int MULT	3.1
32 bit float MULT	3.7
32 bit SRAM Cache	5
32 bit DRAM Memory	640

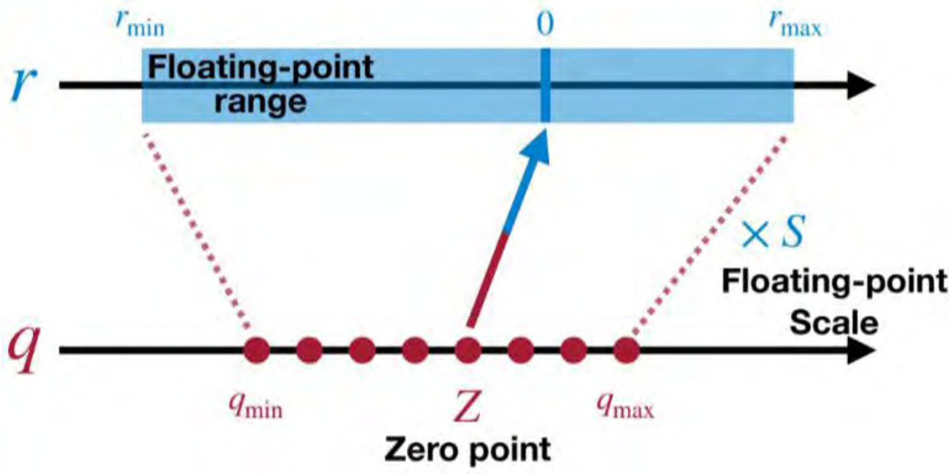
Rough Energy Cost For Various Operations in 45nm 0.9V



Computing's Energy Problem (and What We Can Do About it) [Horowitz, M., IEEE ISSCC 2014]

Linear Quantization

- An affine mapping of integers to real numbers
  - R = S(q-Z)



$$S = \frac{r_{max} - r_{min}}{q_{max} - q_{min}}$$

$$Z = \text{round}\left(q_{min} - \frac{r_{min}}{S}\right)$$

$$r_{max} = S\left(q_{max} - Z\right)$$

$$r_{min} = S\left(q_{min} - Z\right)$$

Linear Quantization: Example

2.09	-0.98	1.48	0.09
0.05	-0.14	-1.08	2.12
-0.91	1.92	0	-1.03
1.87	0	1.53	1.49

Bit Width	q_{min}	q_{max}
2	-2	1
3	-4	3
4	-8	7
N	-2^{N-1}	$2^{N-1}-1$

$$r_{\max} = S\left(q_{\max} - Z\right)$$

$$r_{\min} = S\left(q_{\min} - Z\right)$$

$$S = \frac{r_{\max} - r_{\min}}{q_{\max} - q_{\min}}$$

$$Z = \mathrm{round}\left(q_{\min} - \frac{r_{\min}}{S}\right)$$

Linear Quantization: Example

2.09	-0.98	1.48	0.09
0.05	-0.14	-1.08	2.12
-0.91	1.92	0	-1.03
1.87	0	1.53	1.49

S = (2.12 - (-1.08)) / (1 - (-2))
= 1.07

Z = round (-2 - -1.08 / 1.07)
= -1

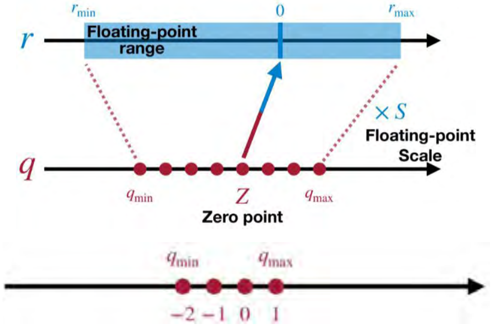
$$r_{\max} = S\left(q_{\max} - Z\right)$$

$$r_{\min} = S\left(q_{\min} - Z\right)$$

$$S = \frac{r_{\max} - r_{\min}}{q_{\max} - q_{\min}}$$

$$Z = \text{round}\left(q_{\min} - \frac{r_{\min}}{S}\right)$$

Bit Width	q_{min}	q_{max}
2	-2	1
3	-4	3
4	-8	7
N	-2^{N-1}	$2^{N-1}-1$



Linear Quantization

weights (32-bit float)

2.09	-0.98	1.48	0.09
0.05	-0.14	-1.08	2.12
-0.91	1.92	0	-1.03
1.87	0	1.53	1.49

⇨ (quantized weights (2-bit signed int) − zero point (2-bit signed int)) × scale (32-bit float) = reconstructed weights (32-bit float)

quantized weights (2-bit signed int)

1	-2	0	-1
-1	-1	-2	1
-2	1	-1	-2
1	-1	0	0

zero point (2-bit signed int): -1

scale (32-bit float): 1.07

reconstructed weights (32-bit float)

2.14	-1.07	1.07	0
0	0	-1.07	2.14
-1.07	2.14	0	-1.07
2.14	0	1.07	1.07

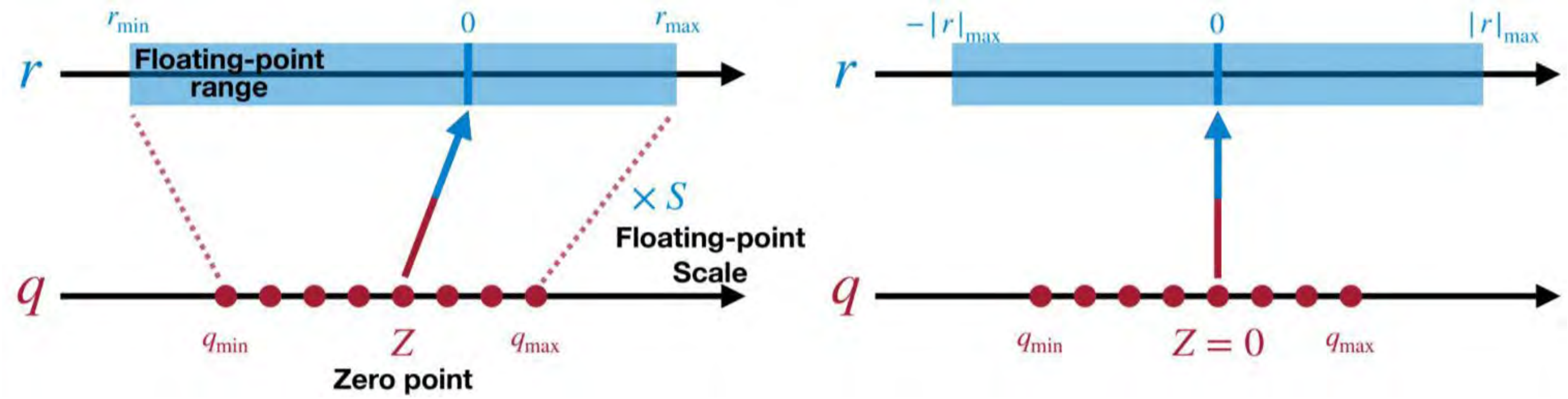
quantization error

-0.05	0.09	0.41	0.09
0.05	-0.14	-0.01	-0.02
0.16	-0.22	0	0.04
-0.27	0	0.46	0.42

$q = \text{round}(r / S + Z)$

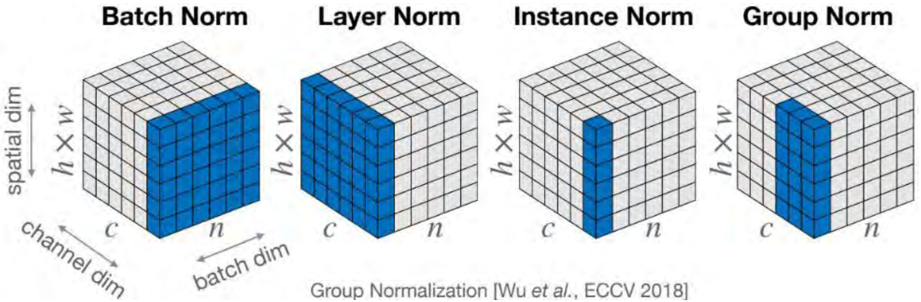
Binary	Decimal
01	1
00	0
11	-1
10	-2

Symmetric Linear Quantization



Quantization Granularity

- Per-tensor quantization
- Per-channel quantization
- Group quantization



Group Normalization [Wu *et al.*, ECCV 2018]

Per-channel vs Per-tensor

$$
\begin{array}{c|cccc}
 & & ic & & \\
\hline
 & 2.09 & -0.98 & 1.48 & 0.09 \\
oc & 0.05 & -0.14 & -1.08 & 2.12 \\
 & -0.91 & 1.92 & 0 & -1.03 \\
 & 1.87 & 0 & 1.53 & 1.49
\end{array}
$$

Per-Channel Quantization

$|r|_{\max} = 2.09 \quad S_0 = 2.09$

$|r|_{\max} = 2.12 \quad S_1 = 2.12$

$|r|_{\max} = 1.92 \quad S_2 = 1.92$

$|r|_{\max} = 1.87 \quad S_3 = 1.87$

1	0	1	0
0	0	-1	1
0	1	0	-1
1	0	1	1

Quantized

2.09	0	2.09	0
0	0	-2.12	2.12
0	1.92	0	-1.92
1.87	0	1.87	1.87

Reconstructed

$$\|\mathbf{W} - \mathbf{S} \odot \mathbf{q_W}\|_F = 2.08$$

<

Per-Tensor Quantization

$$|r|_{\max} = 2.12$$

$$S = \frac{|r|_{\max}}{q_{\max}} = \frac{2.12}{2^{2-1} - 1} = 2.12$$

1	0	1	0
0	0	-1	1
0	1	0	0
1	0	1	1

Quantized

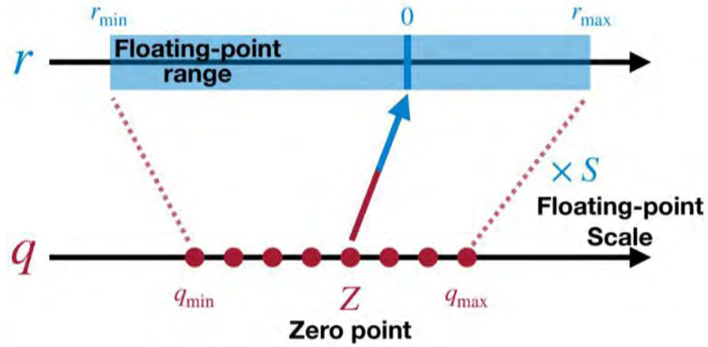
2.12	0	2.12	0
0	0	-2.12	2.12
0	2.12	0	0
2.12	0	2.12	2.12

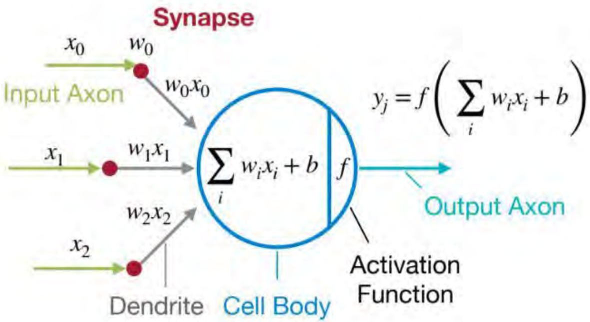
Reconstructed

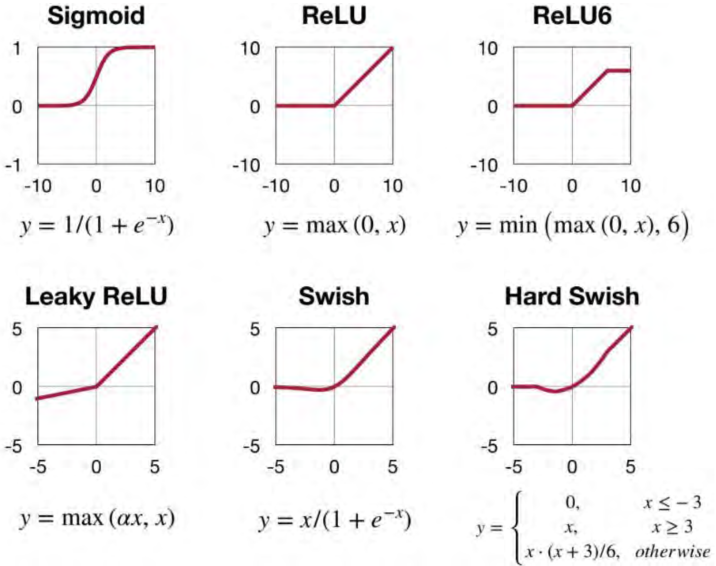
$$\|\mathbf{W} - S\mathbf{q_W}\|_F = 2.28$$

Non-static Dynamic Range

- Weights has [r_min,r_max] fixed range
- Activation value has unknown range
    - depends on input
- How to determine r_min and r_max?

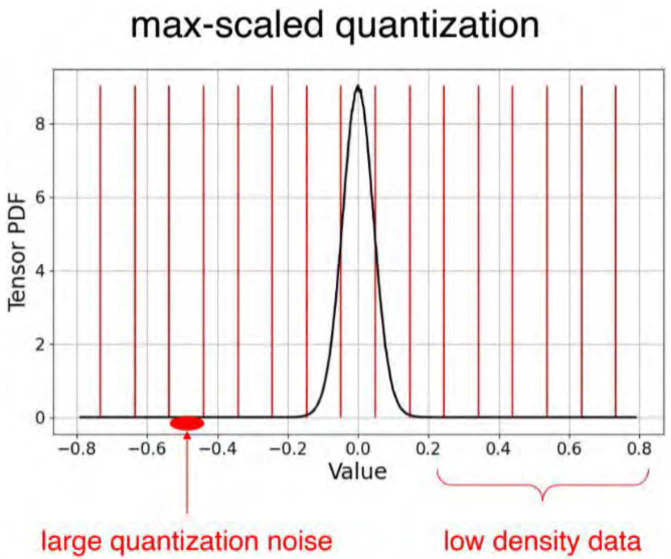


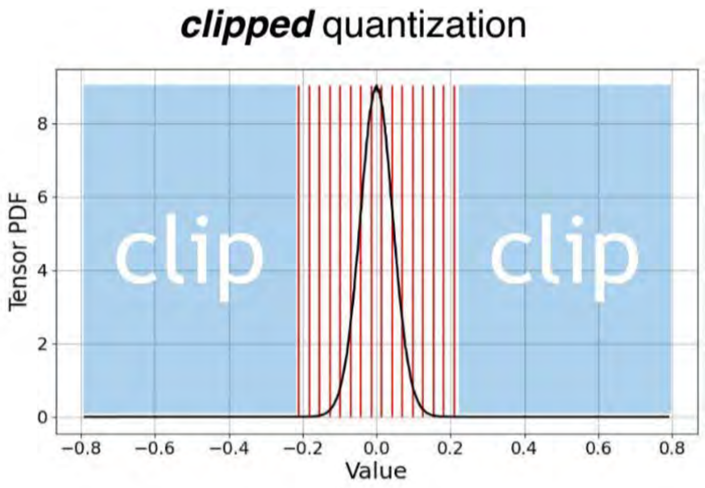




Dynamic Range: Calibration & Clipping

- Run a batch of samples
- Get the statistical distribution of activations
- Get rid of outliers

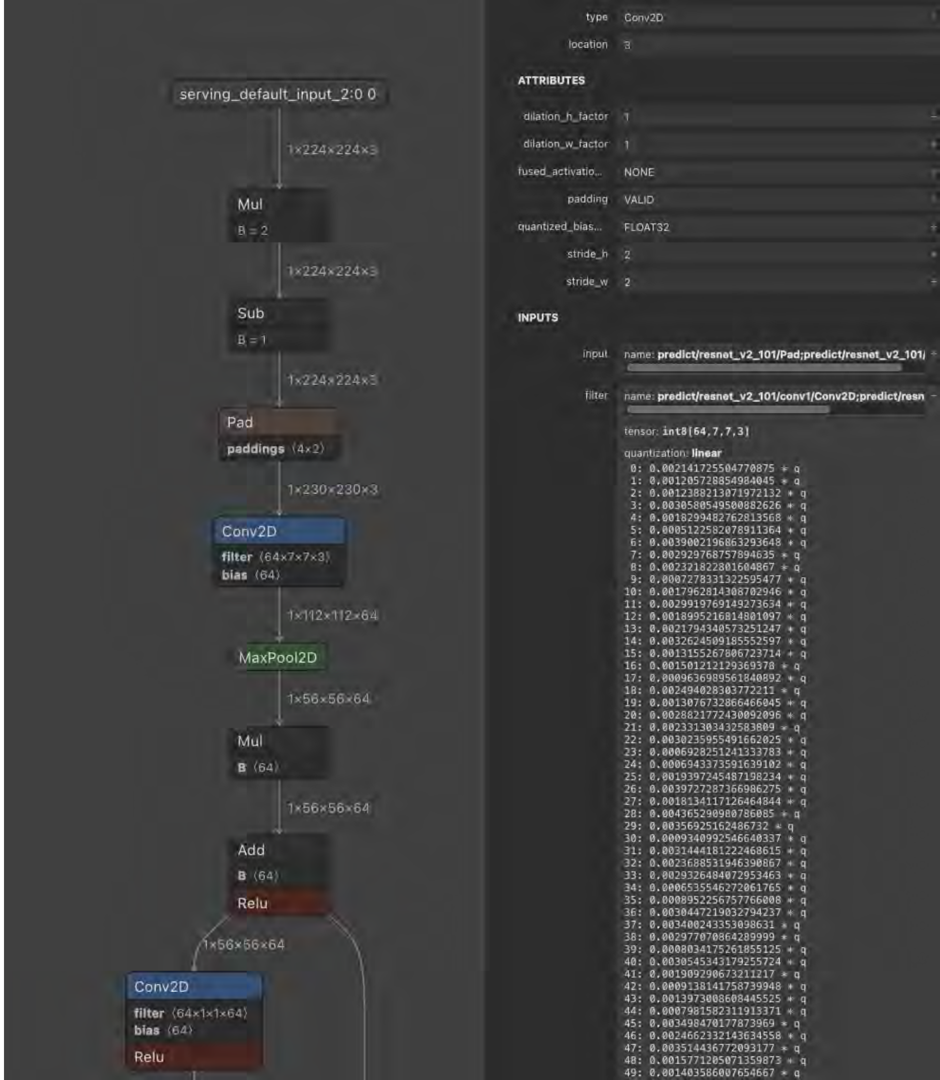




https://hanlab.mit.edu/courses/2023-fall-65940

Tutorial: TF Lite quantization

- [Post-training dynamic range quantization | TensorFlow Lite]()
- Visualize quantized vs unquantized model in Netron.app



Summary

- Pruning
  - What and why?
  - How? Pruning criterion
    - Magnitude-based, Scaling-based
  - Fine-tune/retrain acceleration
    - M:N sparsity, SparseConv
- Quantization
  - What and why?
  - Linear quantization
    - Scale, zero point
  - Quantization granularity
    - Per-tensor, per-channel
  - Calibration and Clipping

Next Lectures

- Example paper presentation: ML-Exray
- Hardware architecture
- Special accelerators
- Quiz on Lecture 05-06