NETWORK COMPRESSION

Hung-yi Lee 李宏毅

Smaller Model

Less parameters

Deploying ML models in resource-constrained environments

Lower latency, Privacy, etc.

Outline

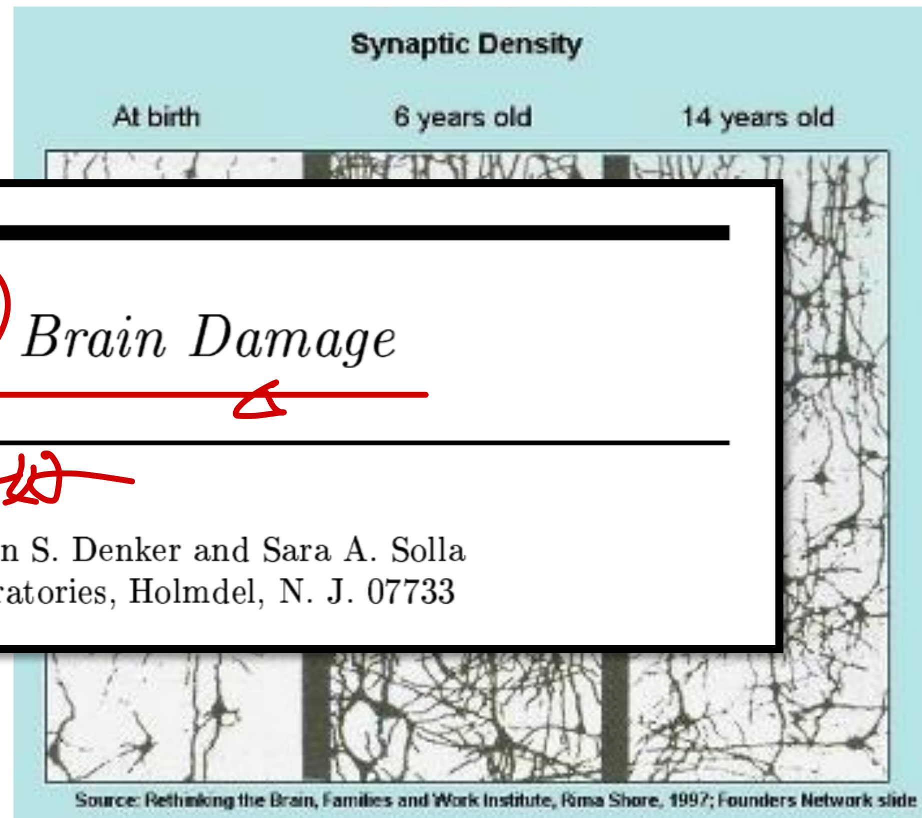
- Network Pruning
- Knowledge Distillation
- Parameter Quantization
- Architecture Design
- Dynamic Computation

We will not talk about hard-ware solution today.

Network Pruning

Network can be pruned 减

- Networks are typically over-parameterized (there is significant redundant weights or neurons)
- Prune them!



Optimal Brain Damage

最好

Yann Le Cun, John S. Denker and Sara A. Solla
AT&T Bell Laboratories, Holmdel, N. J. 07733

(NIPS, 1989)

Network Pruning

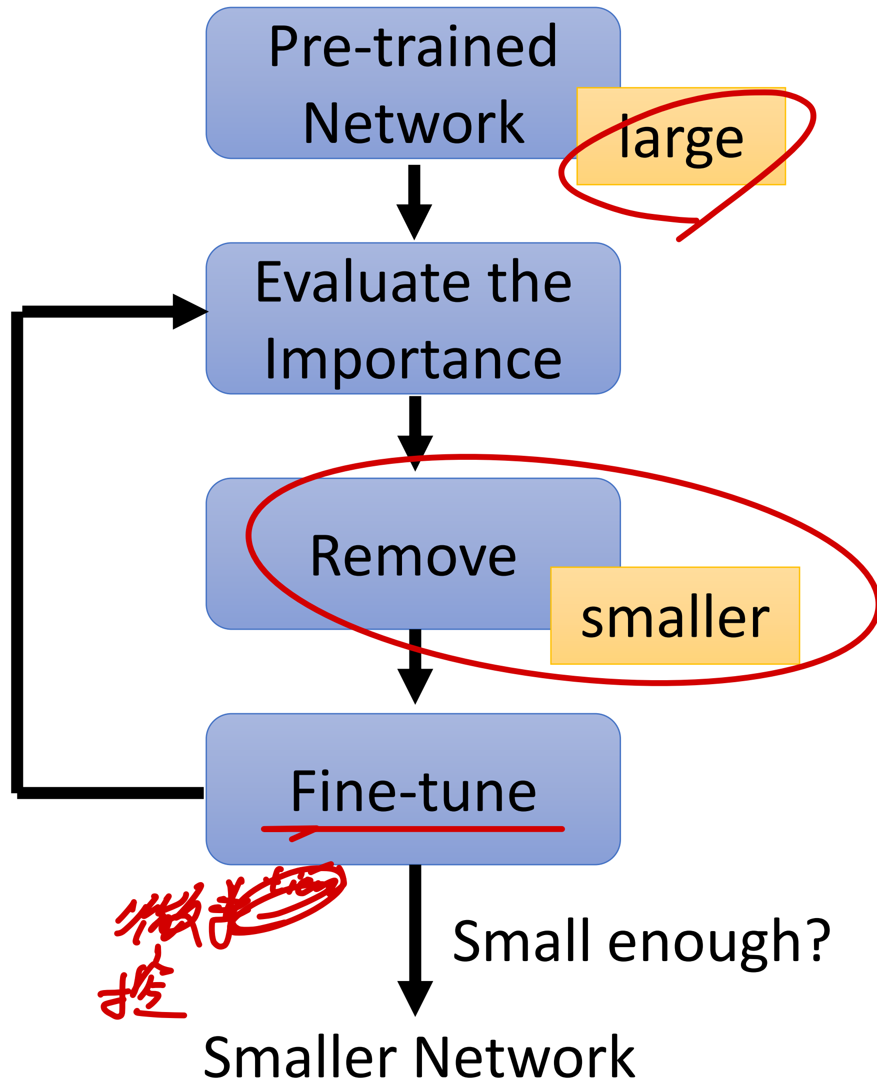
- Importance of a weight:

  absolute values, life long …

- Importance of a neuron:

  the number of times it wasn't zero on a given data set ……

- After pruning, the accuracy will drop (hopefully not too much)
- Fine-tuning on training data for recover
- Don't prune too much at once, or the network won't recover.

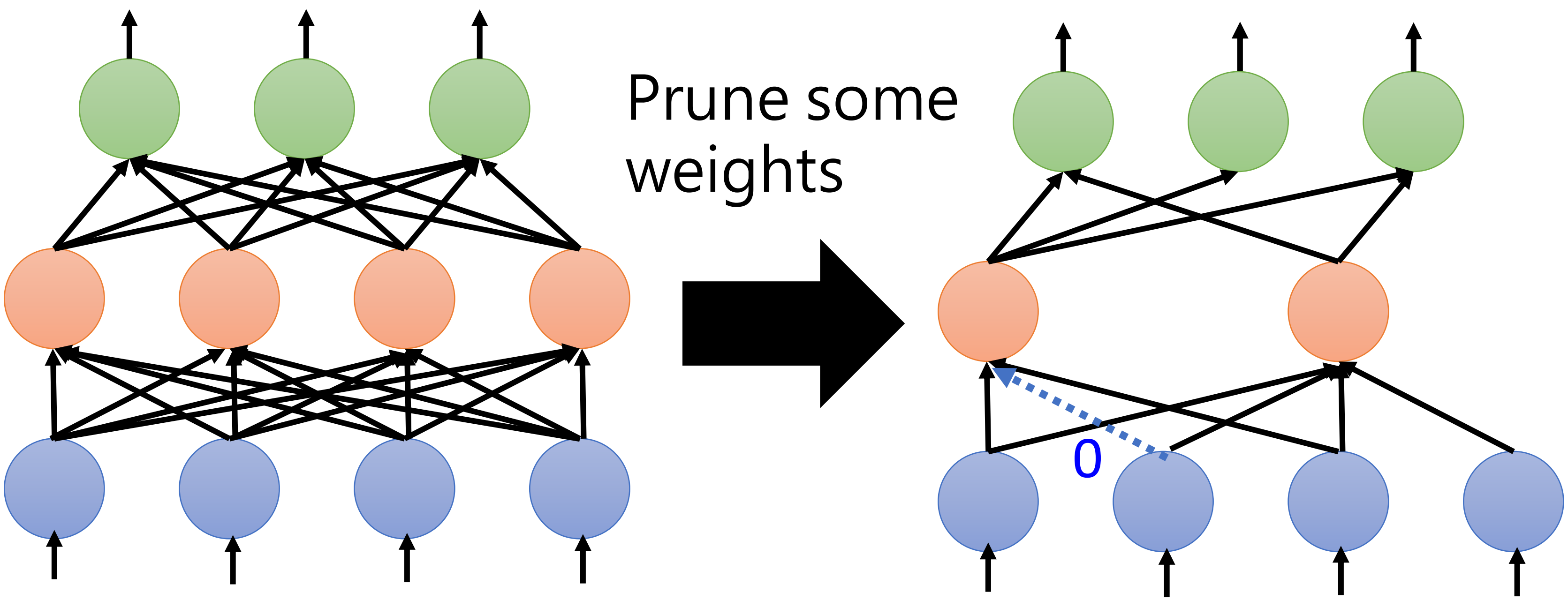


Network Pruning - Practical Issue

以 par 为单位

- Weight pruning

The network architecture becomes irregular.



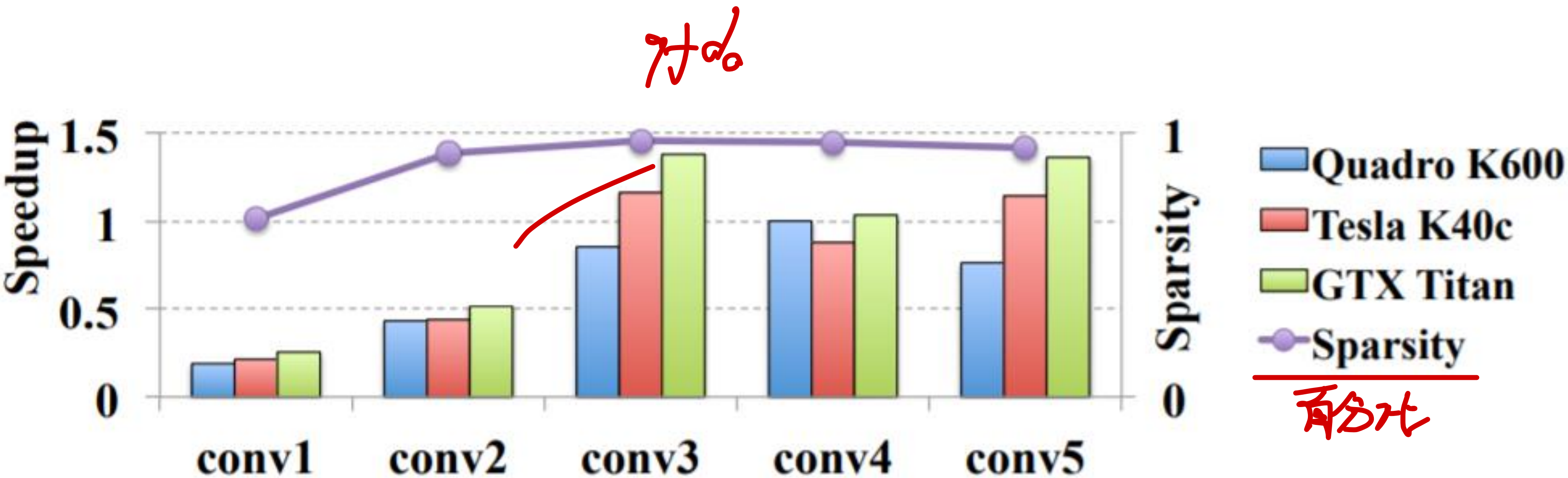
Hard to implement, hard to speedup ......

不能加速

形态不好

Network Pruning - Practical Issue

- Weight pruning

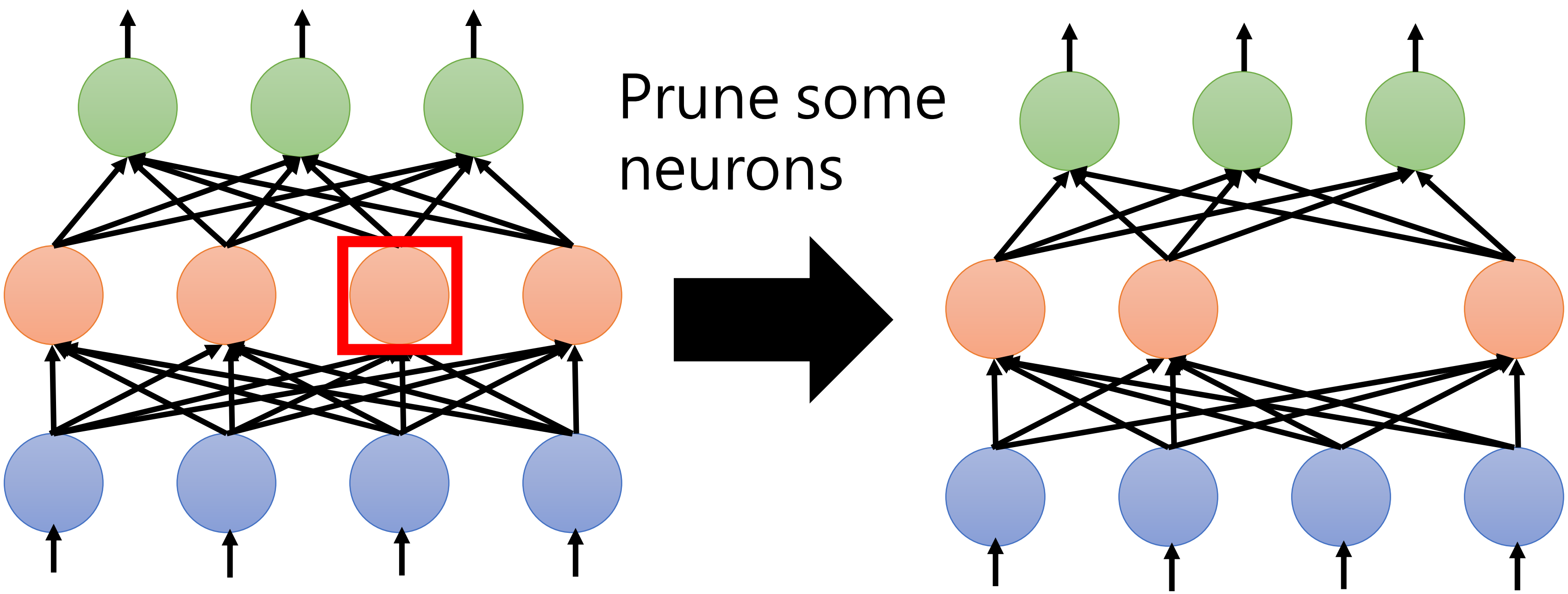


https://arxiv.org/pdf/1608.03665.pdf

Network Pruning - Practical Issue

- Neuron pruning

The network architecture is regular.



Easy to implement, easy to speedup ……

Why Pruning?

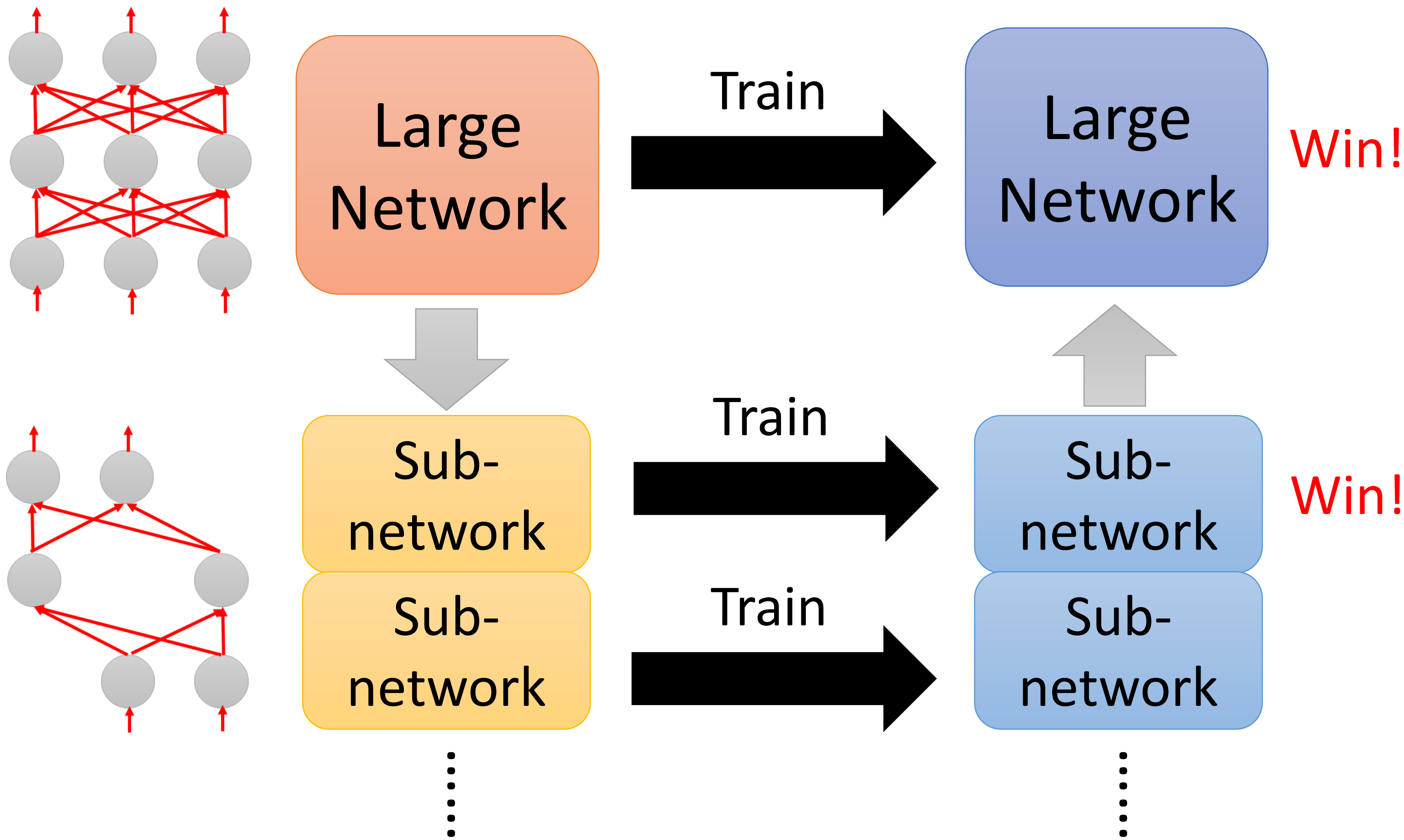
- How about simply train a smaller network?
- It is widely known that smaller network is more difficult to learn successfully.
    - Larger network is easier to optimize? https://www.youtube.com/watch?v=_VuWvQUMQVk
- Lottery Ticket Hypothesis

  https://arxiv.org/abs/1803.03635

大乐透

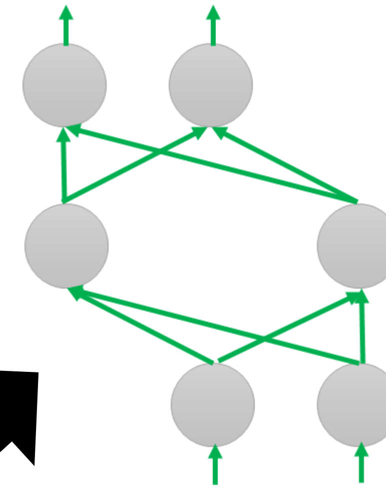
Why Pruning?

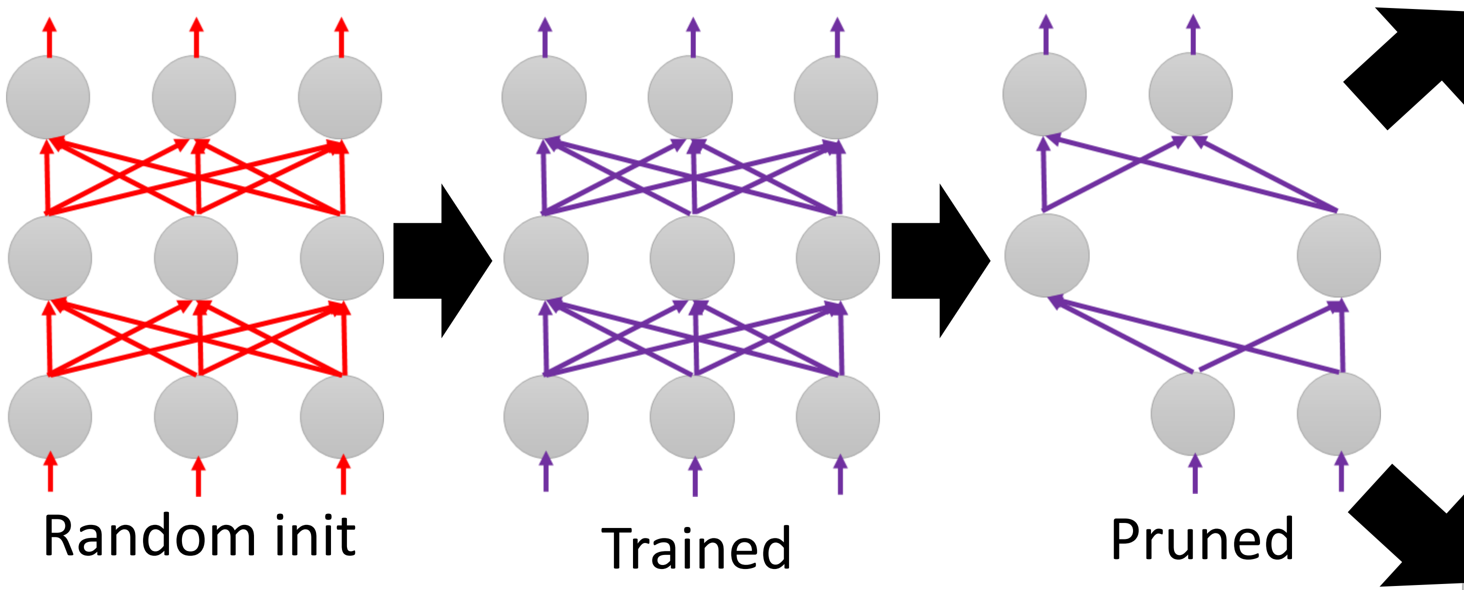
Lottery Ticket Hypothesis

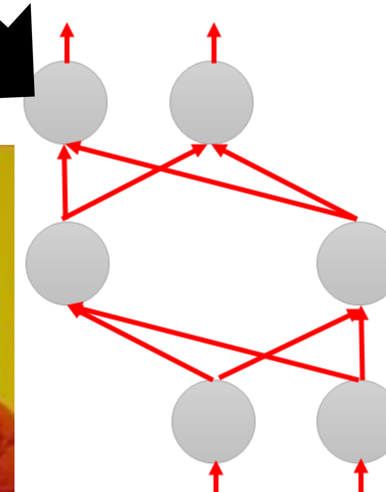


Why Pruning?

Lottery Ticket Hypothesis







Random Init weights

Trained weight

Another random Init weights

Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask

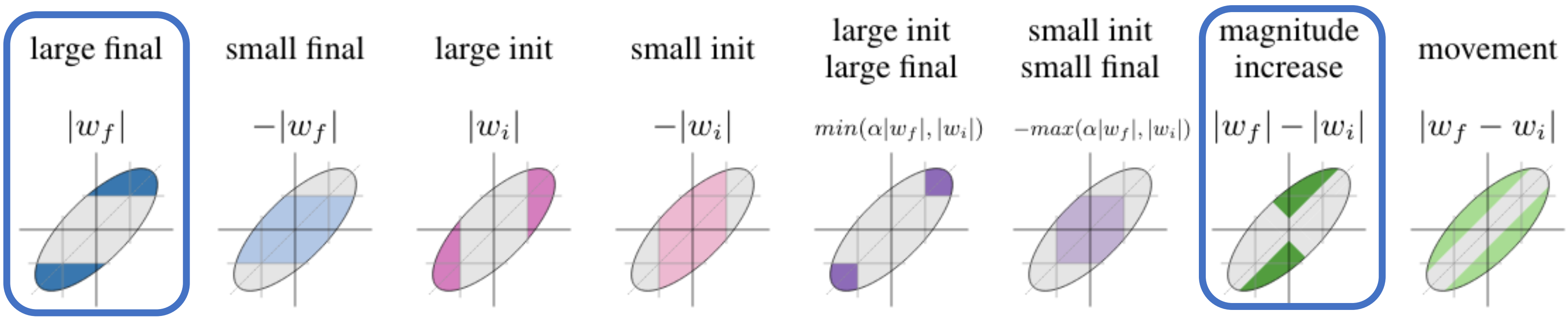
https://arxiv.org/abs/1905.01067

解 gòu

Why Pruning?

Lottery Ticket Hypothesis

- Different pruning strategy



- "sign-ificance" of initial weights: Keeping the sign is critical

0.9, 3.1, -9.1, 8.5 …… ➡ $+\alpha$, $+\alpha$, - α, $+\alpha$ ……

- Pruning weights from a network with random weights

Weight Agnostic Neural Networks https://arxiv.org/abs/1906.04358

Why Pruning?

https://arxiv.org/abs/1810.05270

- Rethinking the Value of Network Pruning

Dataset	Model	Unpruned	Pruned Model	Fine-tuned	Scratch-E	Scratch-B
CIFAR-10	VGG-16	93.63 (±0.16)	VGG-16-A	93.41 (±0.12)	93.62 (±0.11)	**93.78** (±0.15)
	ResNet-56	93.14 (±0.12)	ResNet-56-A	92.97 (±0.17)	92.96 (±0.26)	**93.09** (±0.14)
			ResNet-56-B	92.67 (±0.14)	92.54 (±0.19)	**93.05** (±0.18)
	ResNet-110	93.14 (±0.24)	ResNet-110-A	93.14 (±0.16)	**93.25** (±0.29)	93.22 (±0.22)
			ResNet-110-B	92.69 (±0.09)	92.89 (±0.43)	**93.60** (±0.25)
ImageNet	ResNet-34	73.31	ResNet-34-A	72.56	72.77	**73.03**
			ResNet-34-B	72.29	72.55	**72.91**

- **New** random initialization, not **original** random initialization in “Lottery Ticket Hypothesis”
- Limitation of “Lottery Ticket Hypothesis” (small lr, unstructured)

Knowledge Distillation

Knowledge Distillation

Knowledge Distillation
https://arxiv.org/pdf/1503.02531.pdf
Do Deep Nets Really Need to be Deep?
https://arxiv.org/pdf/1312.6184.pdf

Learning target

“1”: 0.7, “7”: 0.2, “9”: 0.1

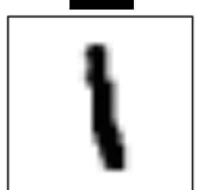
Cross-entropy minimization

?

Teacher Net (Large)

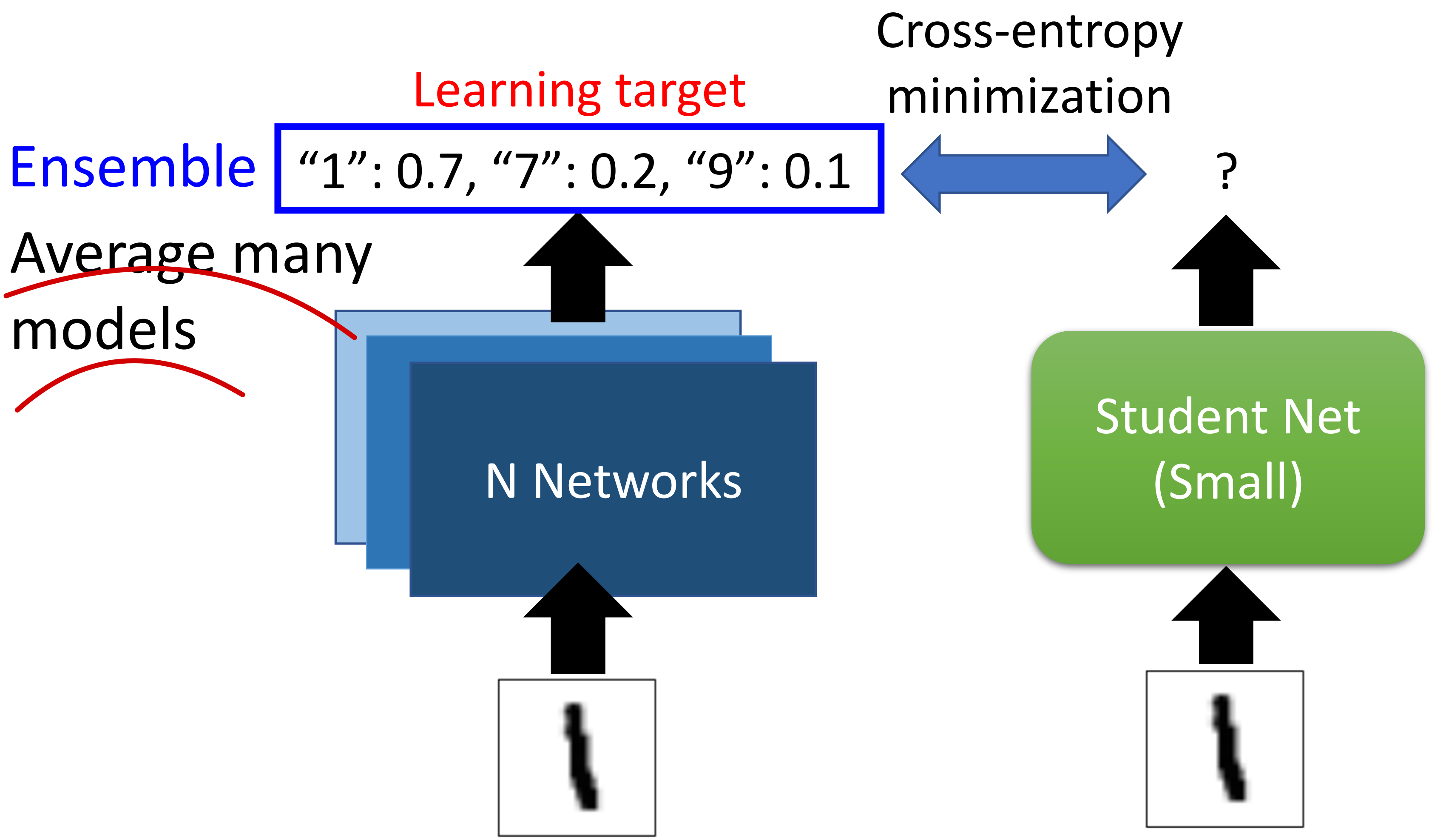
Student Net (Small)

Providing the information that “1” is similar to “7”



Knowledge Distillation

Knowledge Distillation
https://arxiv.org/pdf/1503.02531.pdf
Do Deep Nets Really Need to be Deep?
https://arxiv.org/pdf/1312.6184.pdf



Knowledge Distillation

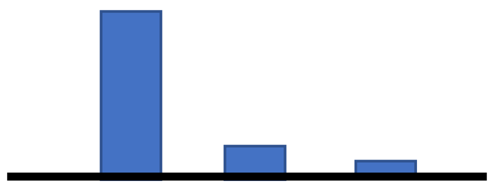
T

S

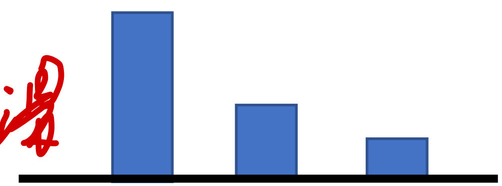
- Temperature for softmax

Hyperpara

$$y_i' = \frac{exp(y_i)}{\sum_j exp(y_j)} \quad \xrightarrow[T = 100]{} \quad y_i' = \frac{exp(y_i/T)}{\sum_j exp(y_j/T)}$$



平滑



$y_1 = 100$	$y_1' = 1$	$y_1/T = 1$	$y_1' = 0.56$
$y_2 = 10$	$y_2' \approx 0$	$y_2/T = 0.1$	$y_2' = 0.23$
$y_3 = 1$	$y_3' \approx 0$	$y_3/T = 0.01$	$y_3' = 0.21$

Parameter Quantization

Parameter Quantization

16 — 8

- 1. Using less bits to represent a value
- 2. Weight clustering

weights in a network

0.5	1.3	4.3	-0.1
0.1	-0.2	-1.2	0.3
1.0	3.0	-0.4	0.1
-0.5	-0.1	-3.4	-5.0

Clustering

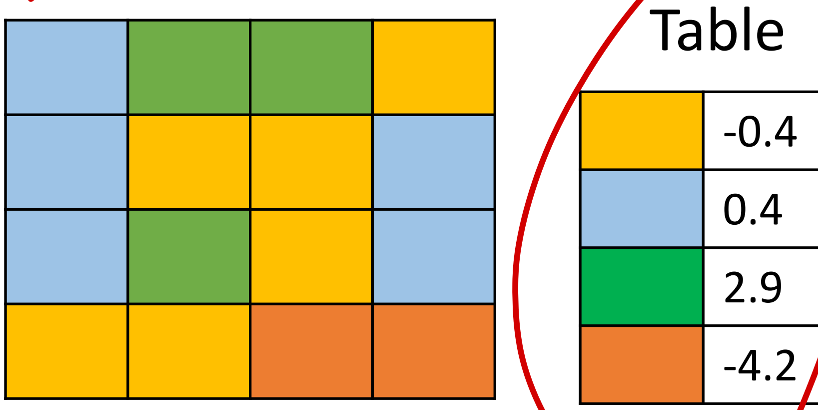
Parameter Quantization

- 1. Using less bits to represent a value
- 2. Weight clustering

weights in a network

0.5	1.3	4.3	-0.1
0.1	-0.2	-1.2	0.3
1.0	3.0	-0.4	0.1
-0.5	-0.1	-3.4	-5.0

Clustering



Only needs 2 bits

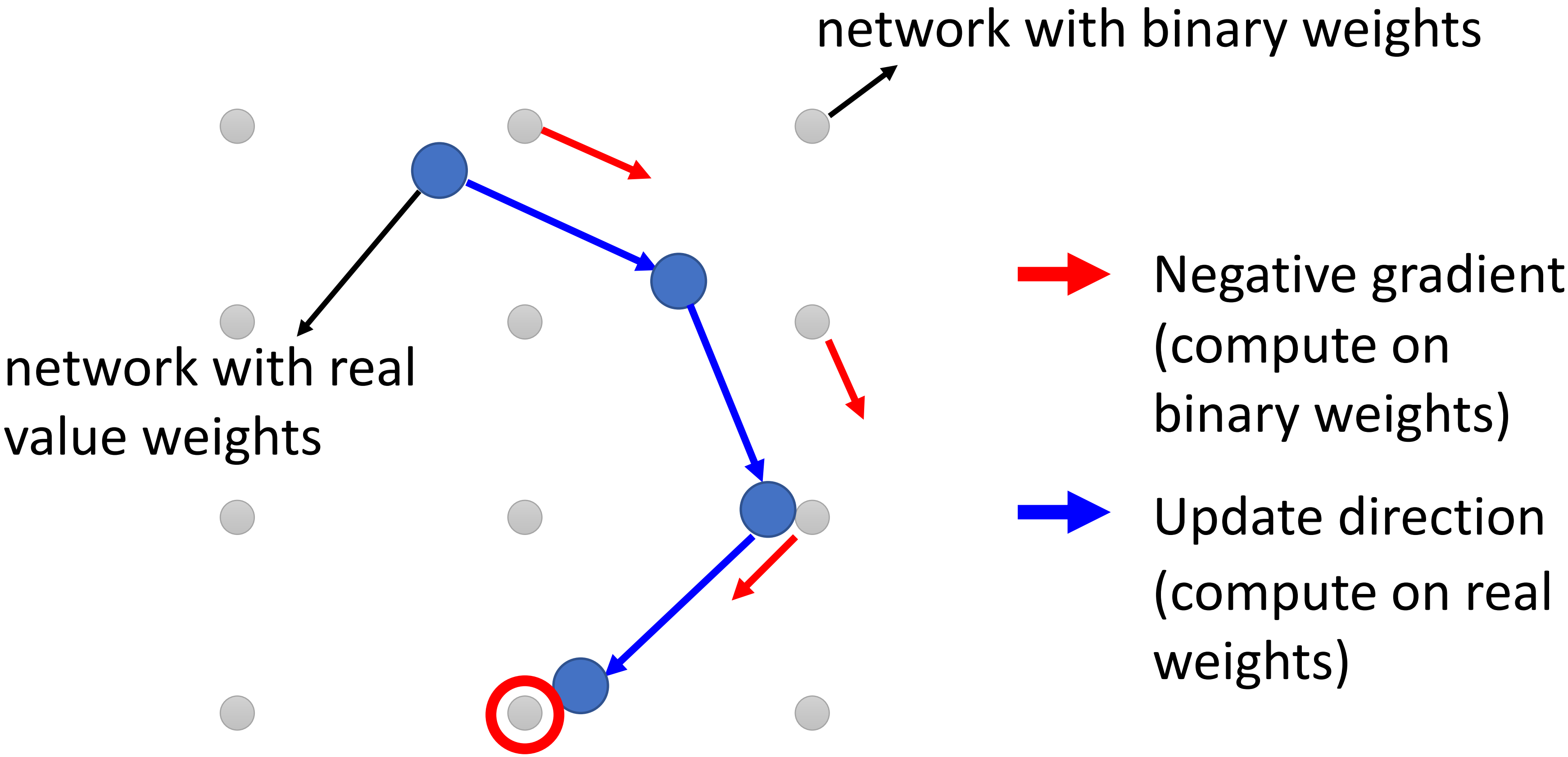
- 3. Represent frequent clusters by less bits, represent rare clusters by more bits
  - e.g. Huffman encoding

Binary Weights

Your weights are always +1 or -1

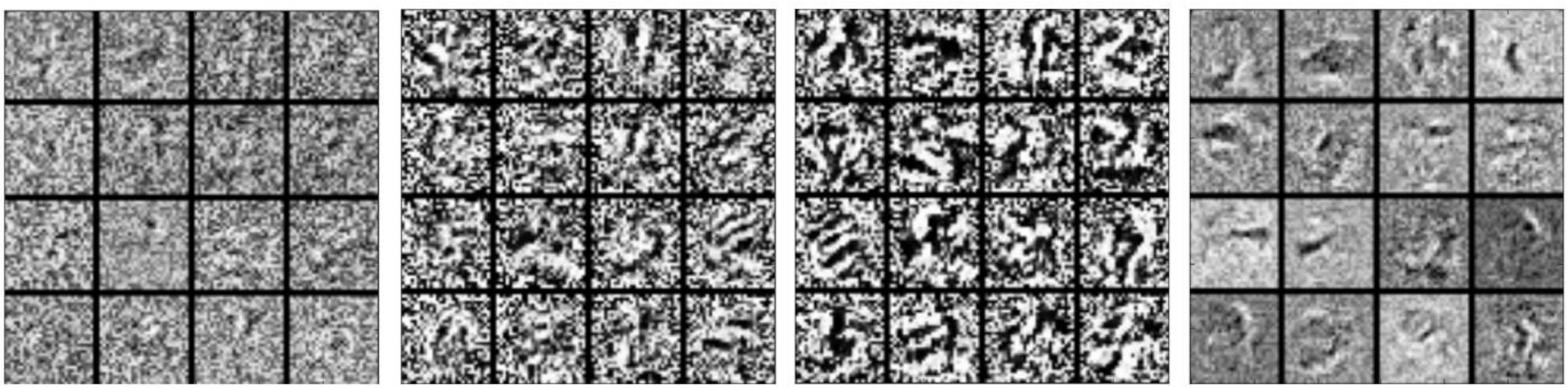
- Binary Connect

Binary Connect:
https://arxiv.org/abs/1511.00363
Binary Network:
https://arxiv.org/abs/1602.02830
XNOR-net:
https://arxiv.org/abs/1603.05279



Binary Weights

Method	MNIST	CIFAR-10	SVHN
No regularizer	$1.30 \pm 0.04\%$	10.64%	2.44%
BinaryConnect (det.)	$1.29 \pm 0.08\%$	9.90%	2.30%
BinaryConnect (stoch.)	$1.18 \pm 0.04\%$	**8.27%**	2.15%
50% Dropout	$1.01 \pm 0.04\%$		



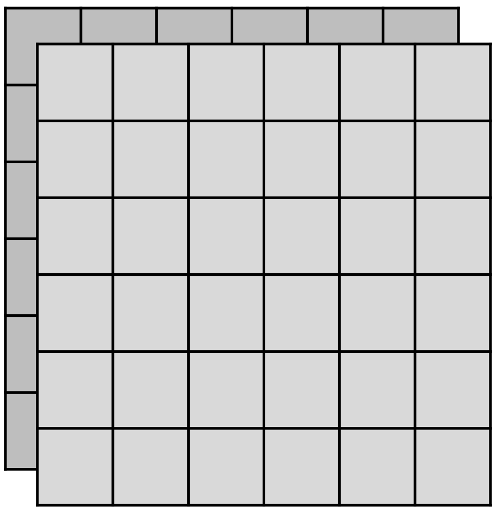
https://arxiv.org/abs/1511.00363

Architecture Design

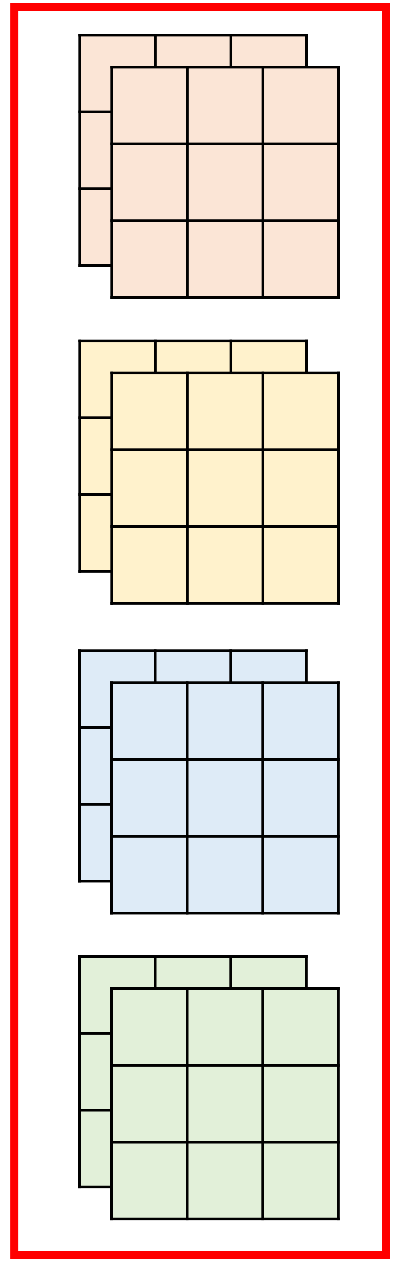
Depthwise Separable Convolution

Review: Standard CNN

Input feature map

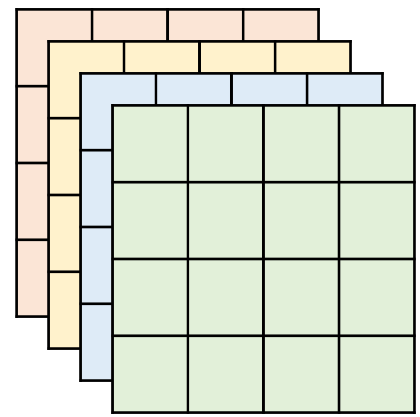


2 channels



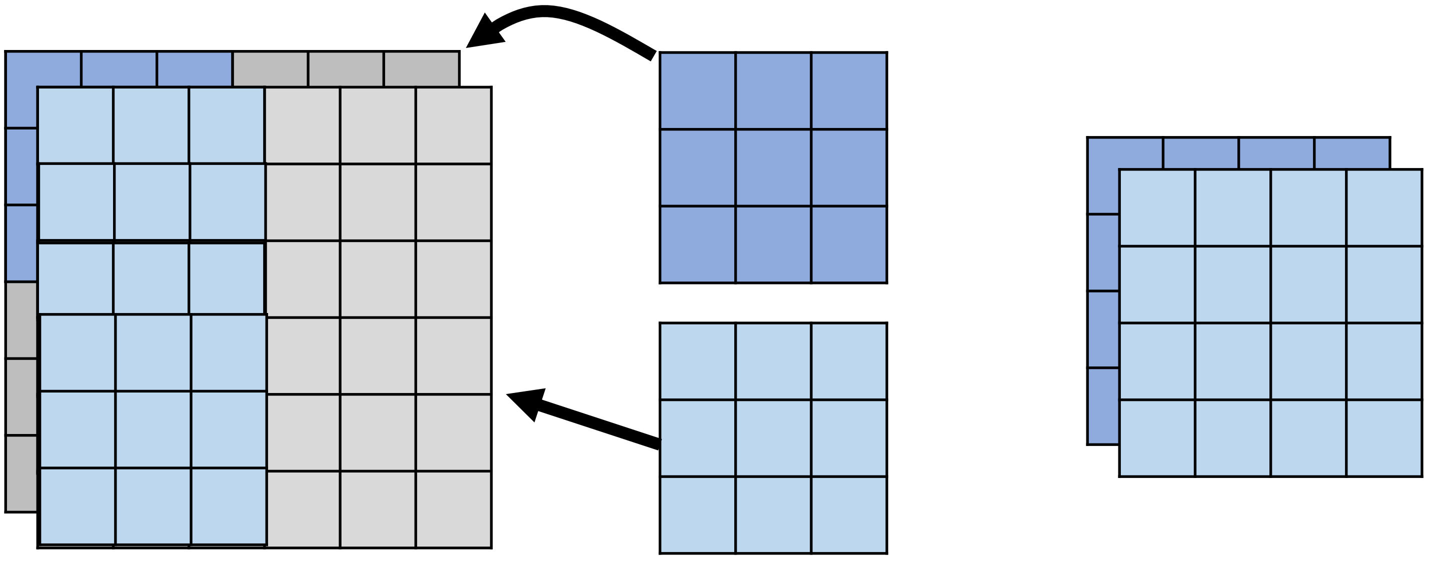
$$3 \times 3 \times 2 \times 4 = 72$$

parameters



Depthwise Separable Convolution

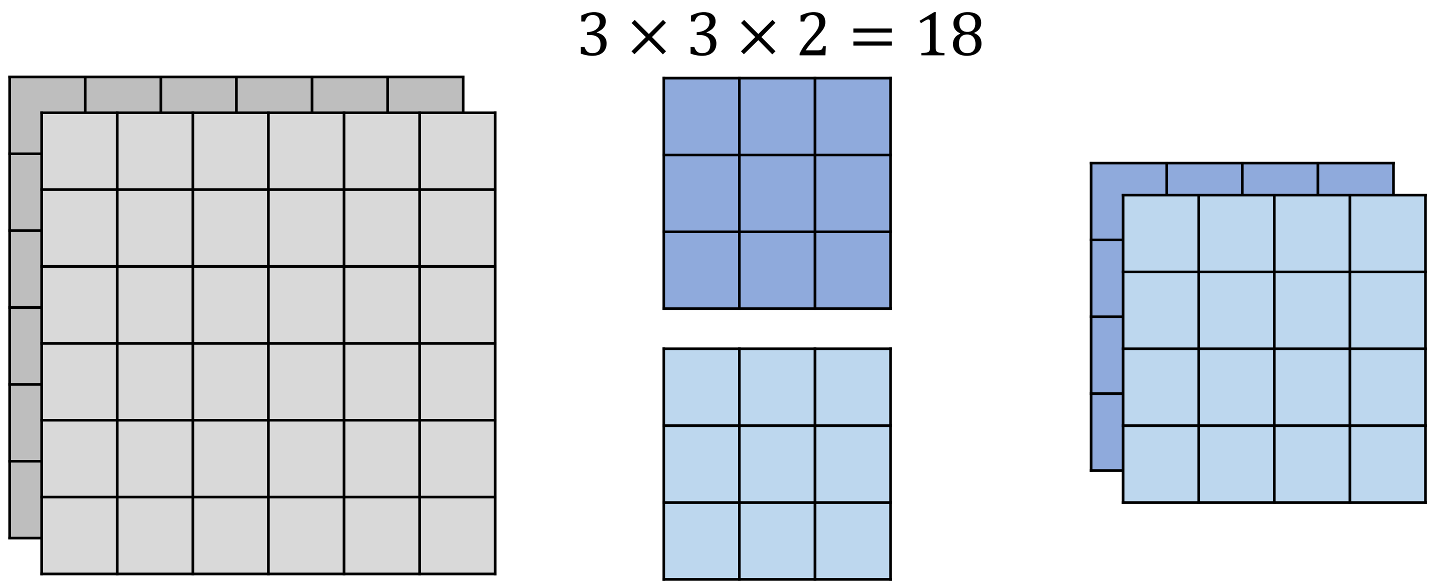
1. Depthwise Convolution



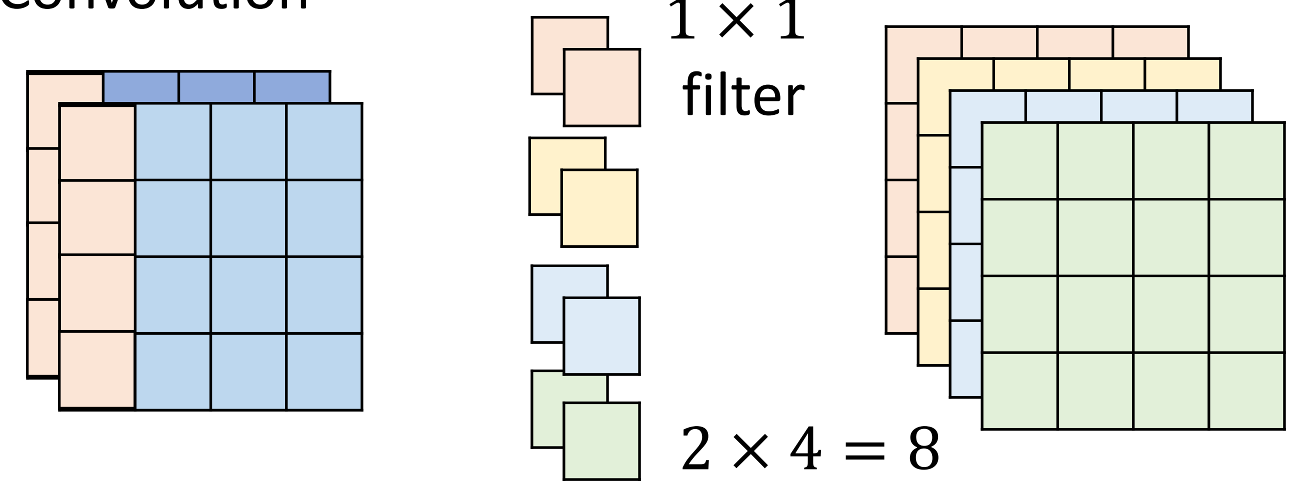
- Filter number = Input channel number
- Each filter only considers one channel.
- The filters are $k \times k$ matrices
- There is no interaction between channels.

Depthwise Separable Convolution

1. Depthwise Convolution



2. Pointwise Convolution



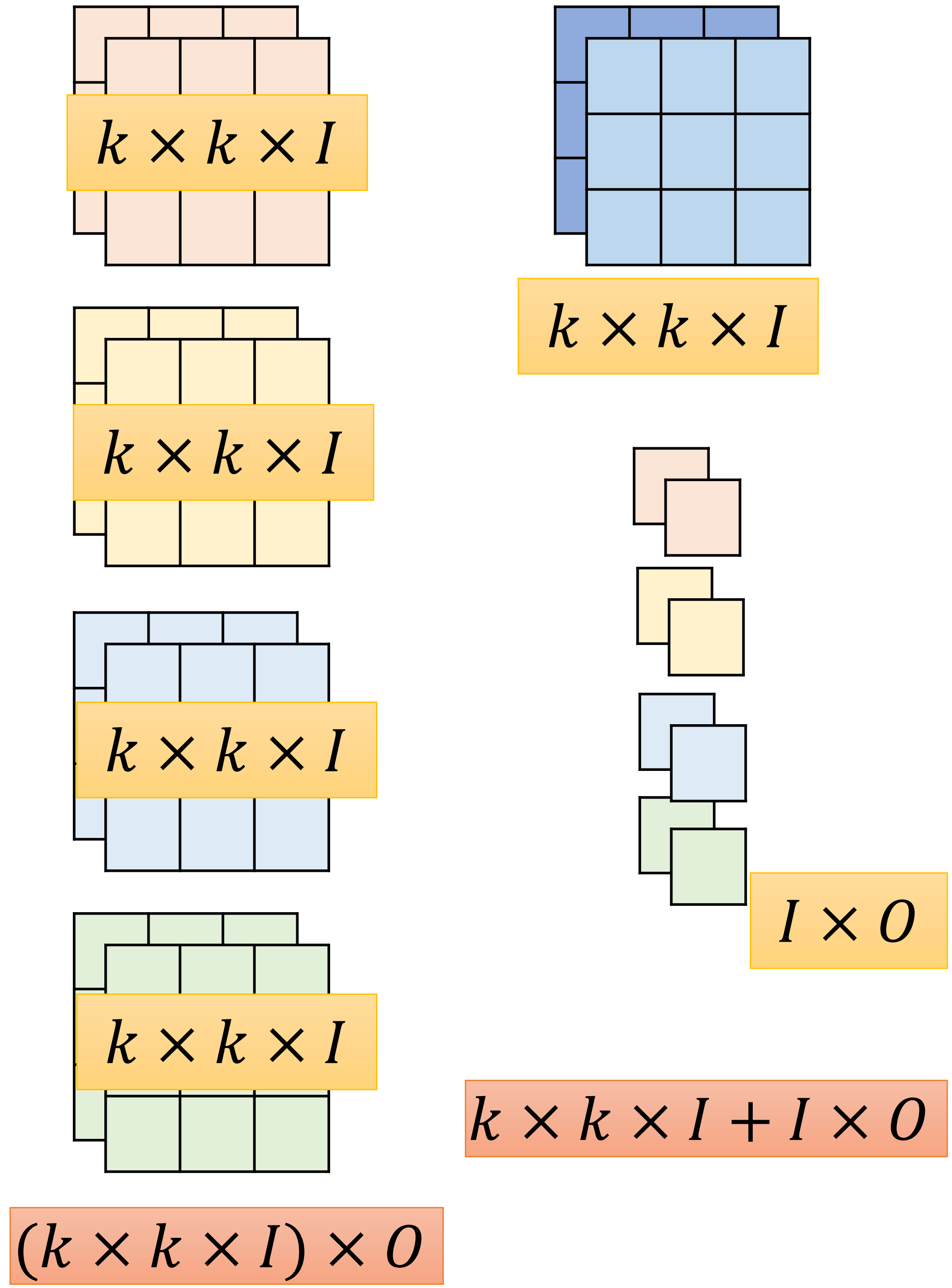
I: number of input channels

O: number of output channels

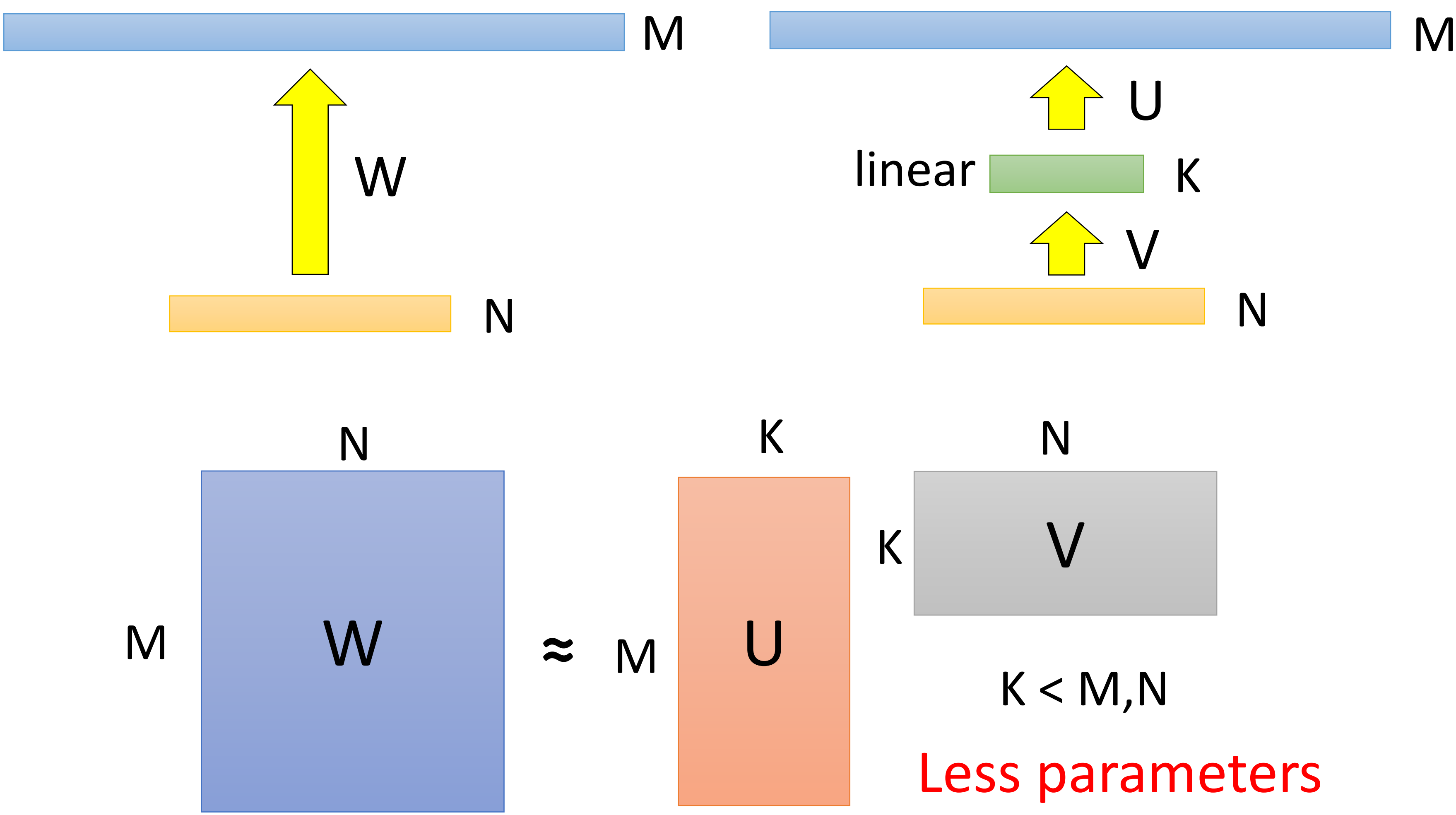
$k \times k$: kernel size

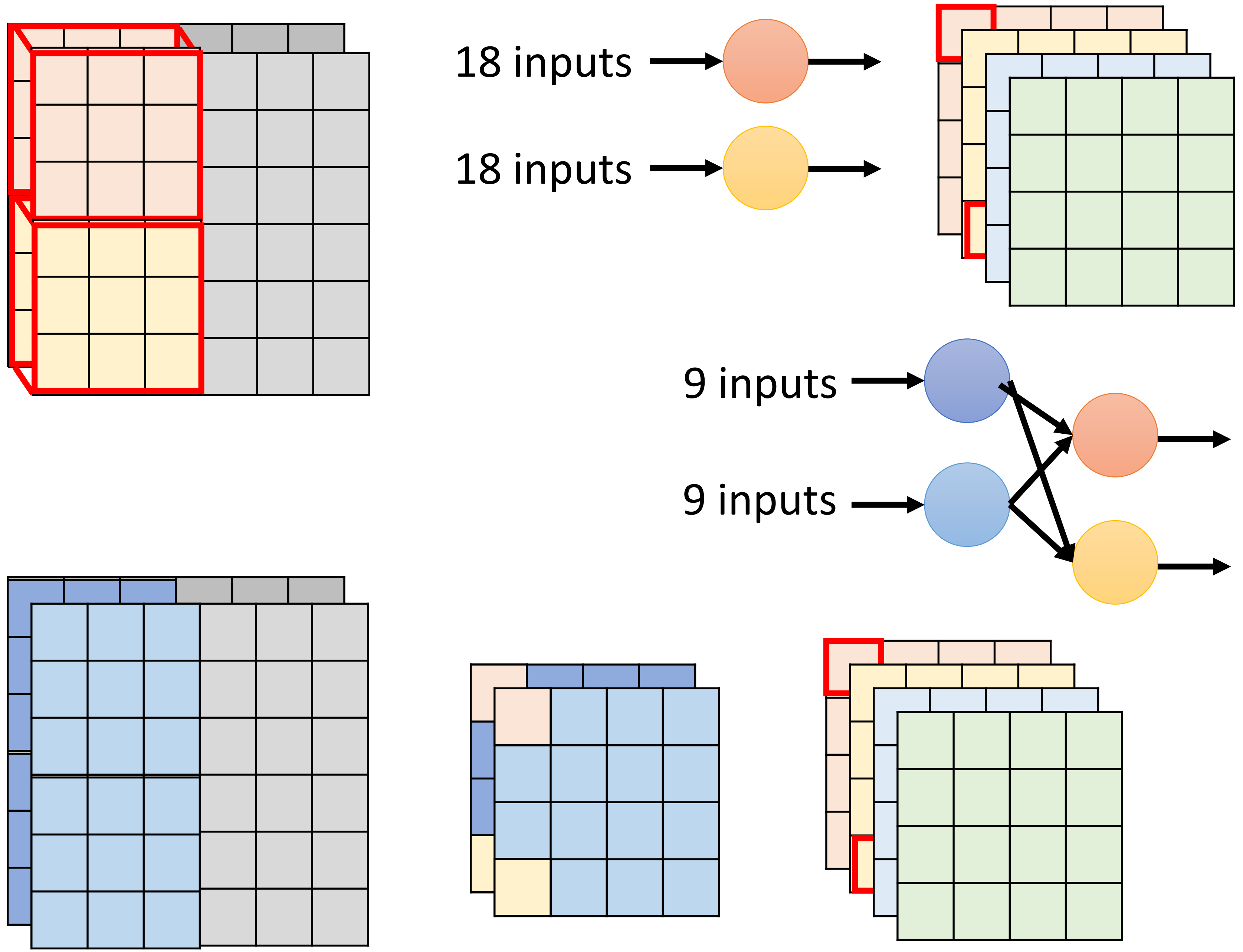
$$\frac{k \times k \times I + I \times O}{k \times k \times I \times O}$$

$$= \frac{1}{O} + \boxed{\frac{1}{k \times k}}$$



Low rank approximation




18 inputs
18 inputs
9 inputs
9 inputs

To learn more ……

- SqueezeNet
  - https://arxiv.org/abs/1602.07360
- MobileNet
  - https://arxiv.org/abs/1704.04861
- ShuffleNet
  - https://arxiv.org/abs/1707.01083
- Xception
  - https://arxiv.org/abs/1610.02357
- GhostNet
  - https://arxiv.org/abs/1911.11907

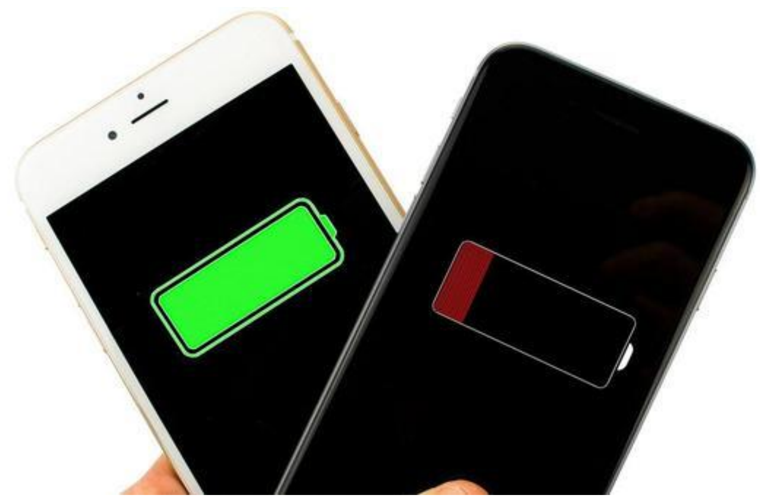
Dynamic Computation

Dynamic Computation

- The network adjusts the computation it need.

Different devices

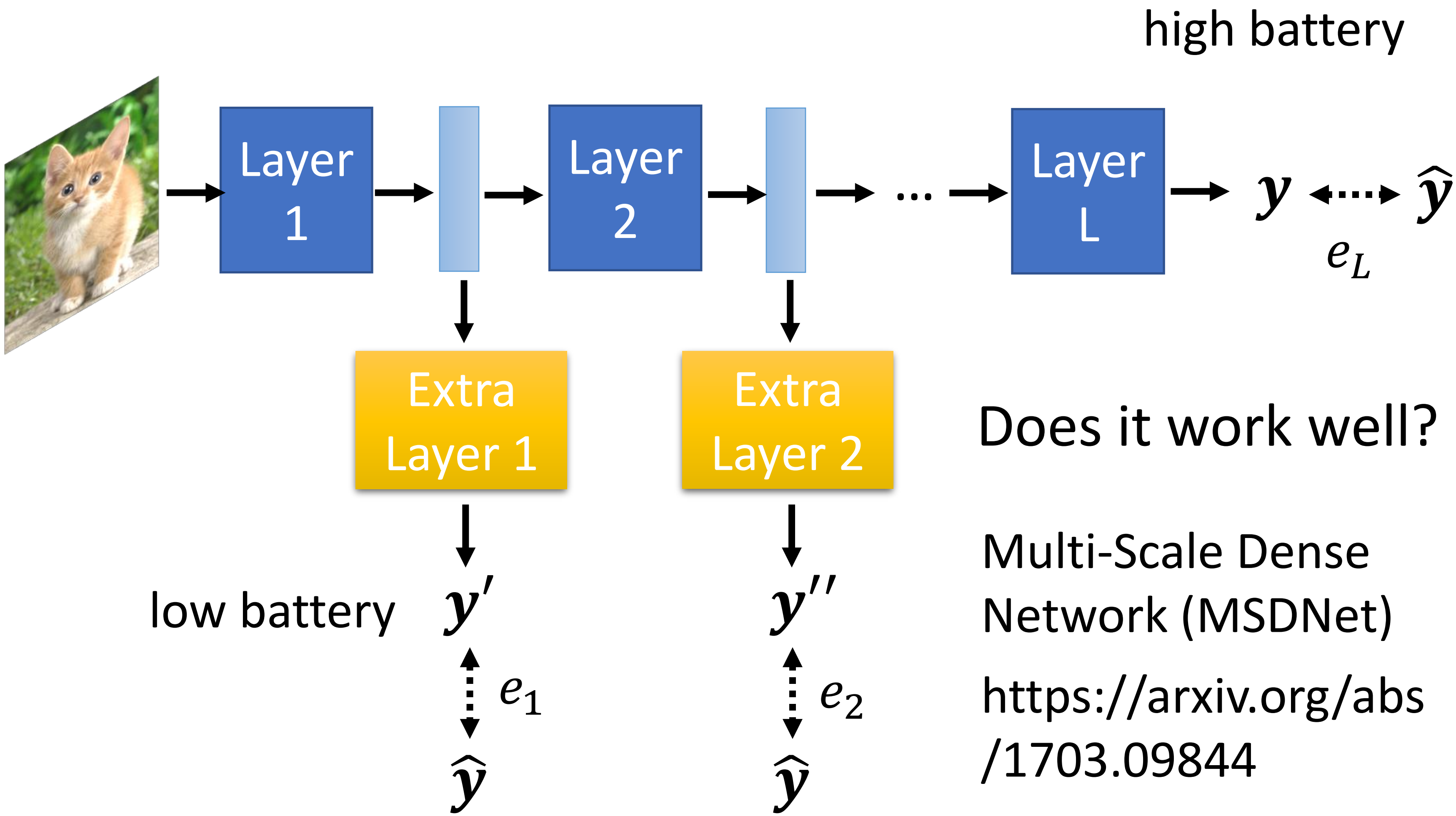
high/low battery



- Why don’t we prepare a set of models?

Dynamic Depth

$$L = e_1 + e_2 + \cdots + e_L$$



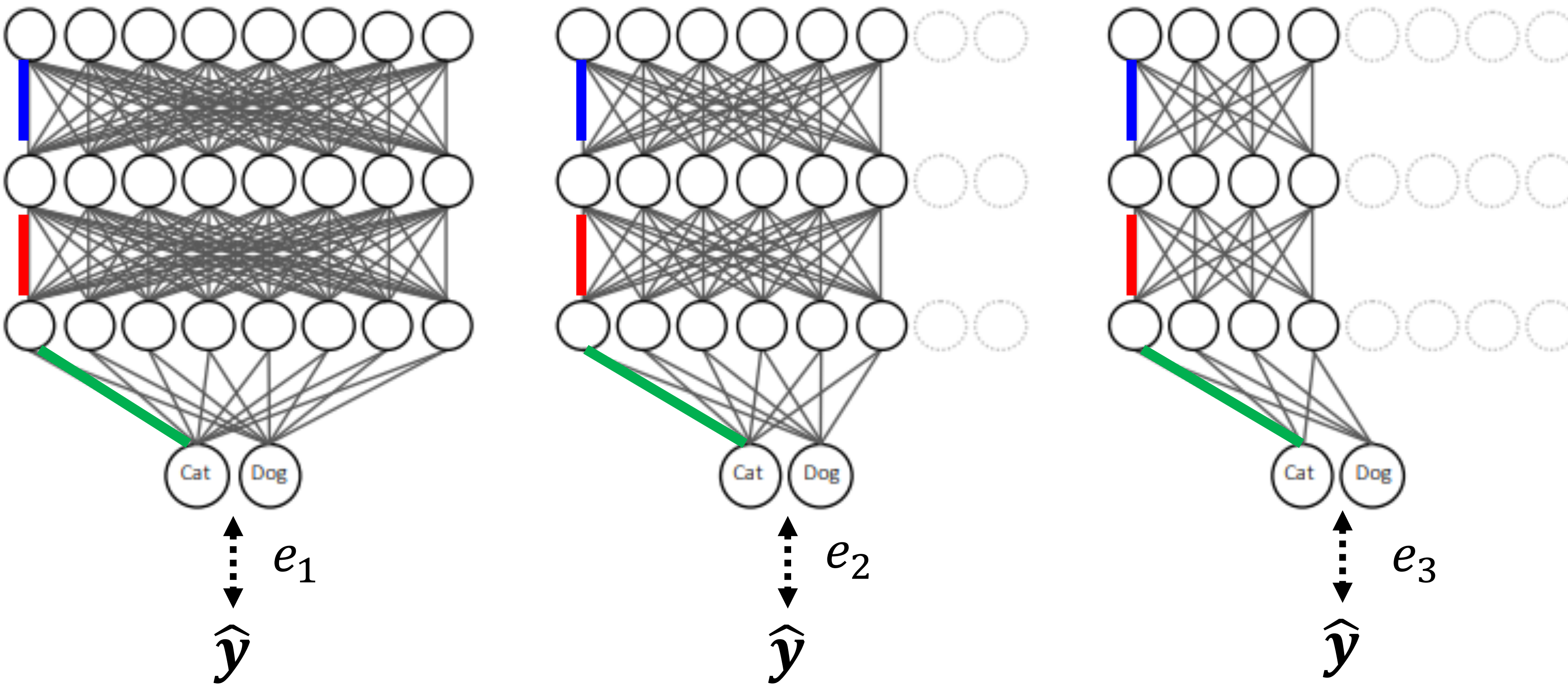
Does it work well?

Multi-Scale Dense Network (MSDNet)

https://arxiv.org/abs/1703.09844

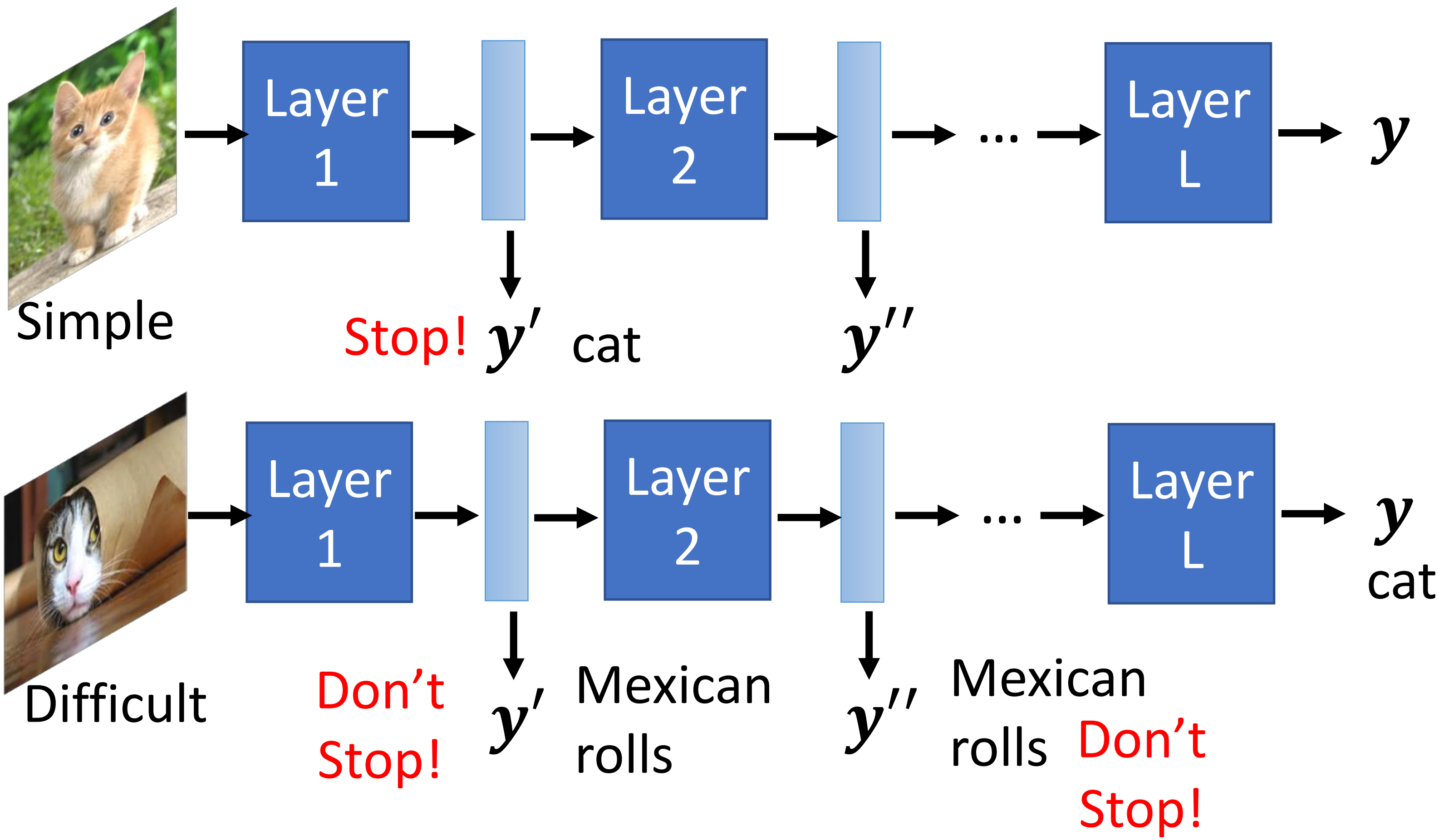
Dynamic Width

$$L = e_1 + e_2 + e_3$$



Slimmable Neural Networks
https://arxiv.org/abs/1812.08928

Computation based on Sample Difficulty



- SkipNet: Learning Dynamic Routing in Convolutional Networks
- Runtime Neural Pruning
- BlockDrop: Dynamic Inference Paths in Residual Networks

Concluding Remarks

- Network Pruning
- Knowledge Distillation
- Parameter Quantization
- Architecture Design
- Dynamic Computation