NETWORK COMPRESSION

Hung-yi Lee 李宏毅

Resource-limited Devices

Limited memory space, limited computing power, etc.

Outline

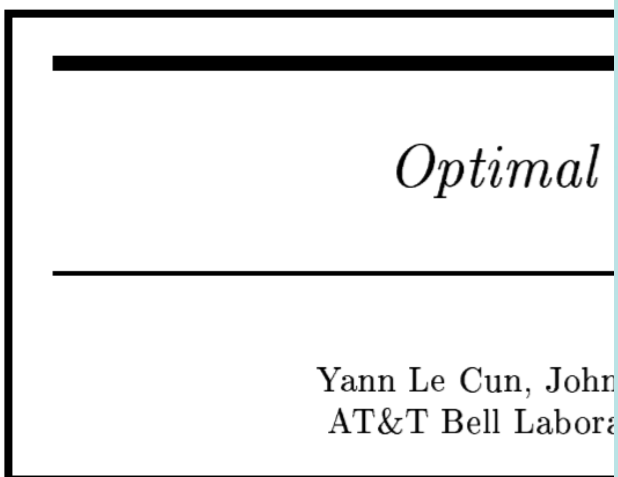
- Network Pruning
- Knowledge Distillation
- Parameter Quantization
- Architecture Design
- Dynamic Computation

We will not talk about hard-ware solution today.

Network Pruning

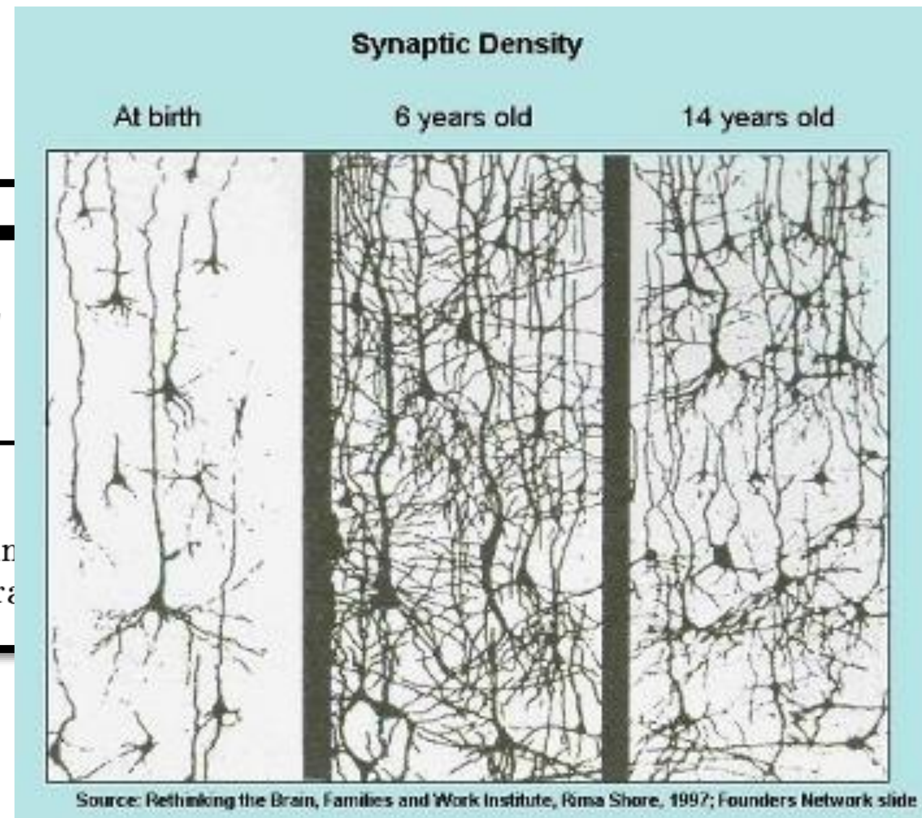
Network can be pruned

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



Optimal

Yann Le Cun, John
AT&T Bell Labora



Network Pruning

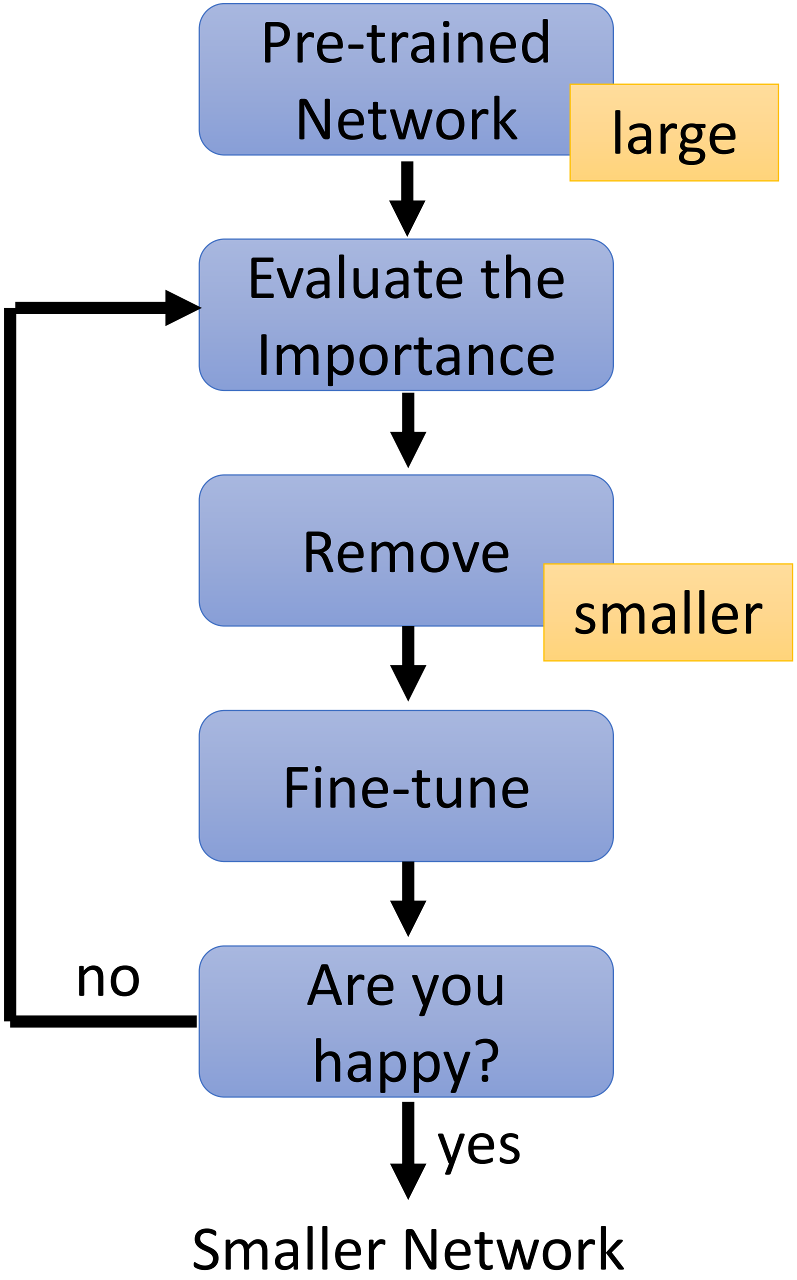
- Importance of a weight:

  L1, L2 ……

- 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.



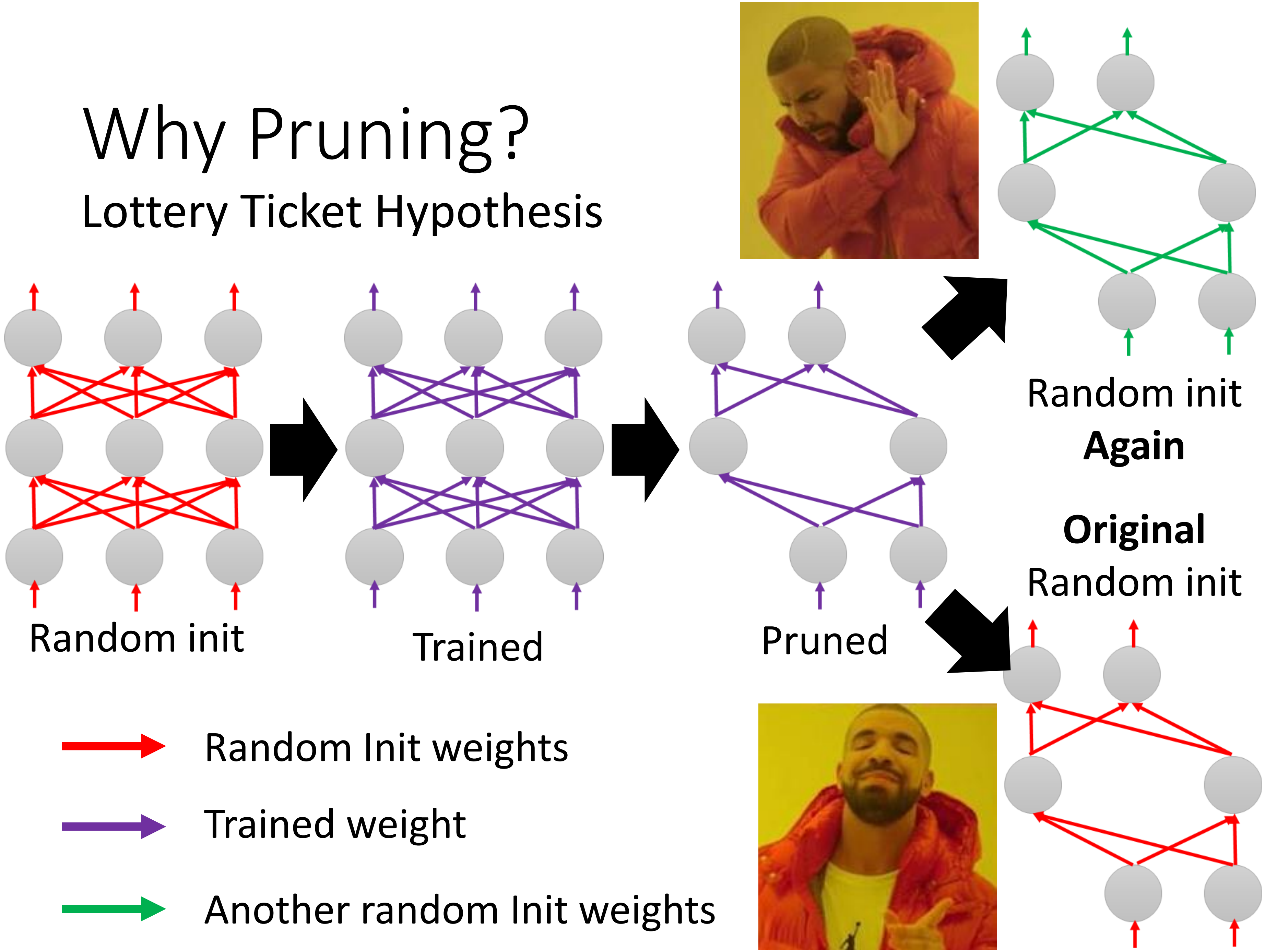
Why Pruning?

- 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

Why Pruning?

Lottery Ticket Hypothesis



Why Pruning?

- Rethinking the Value of Network Pruning
  - https://arxiv.org/abs/1810.05270

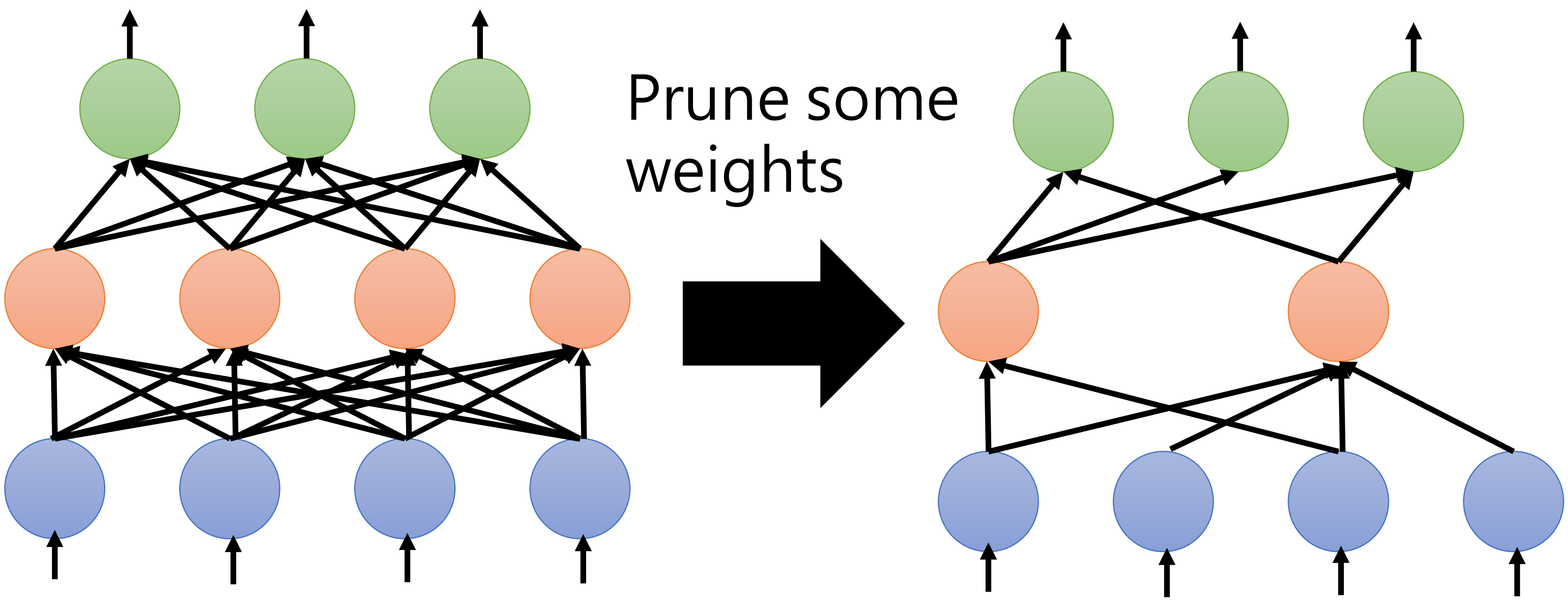
Dataset	Model	Unpruned	Pruned Model	Fine-tuned	Scratch-E	Scratch-B
CIFAR-10	VGG-16	93.63 (±0.16)	VGG-16-A	93.41 (±0.12)	93.62 (±0.11)	**93.78** (±0.15)
	ResNet-56	93.14 (±0.12)	ResNet-56-A	92.97 (±0.17)	92.96 (±0.26)	**93.09** (±0.14)
			ResNet-56-B	92.67 (±0.14)	92.54 (±0.19)	**93.05** (±0.18)
	ResNet-110	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**

- Real random initialization, not original random initialization in "Lottery Ticket Hypothesis"
- Pruning algorithms could be seen as performing network architecture search

Network Pruning - Practical Issue

- Weight pruning

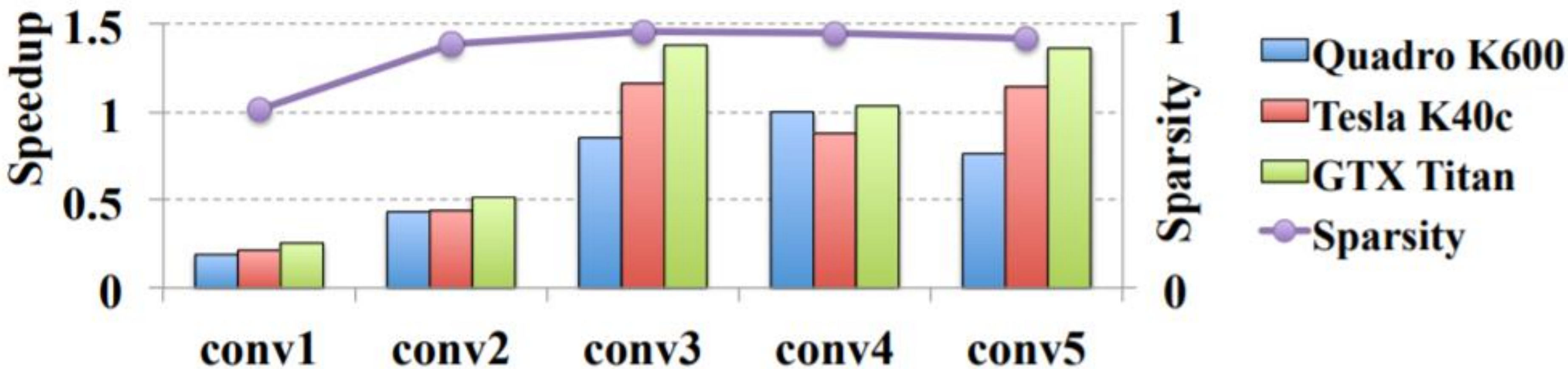
The network architecture becomes irregular.



Hard to implement, hard to speedup ……

Network Pruning - Practical Issue

- Weight pruning

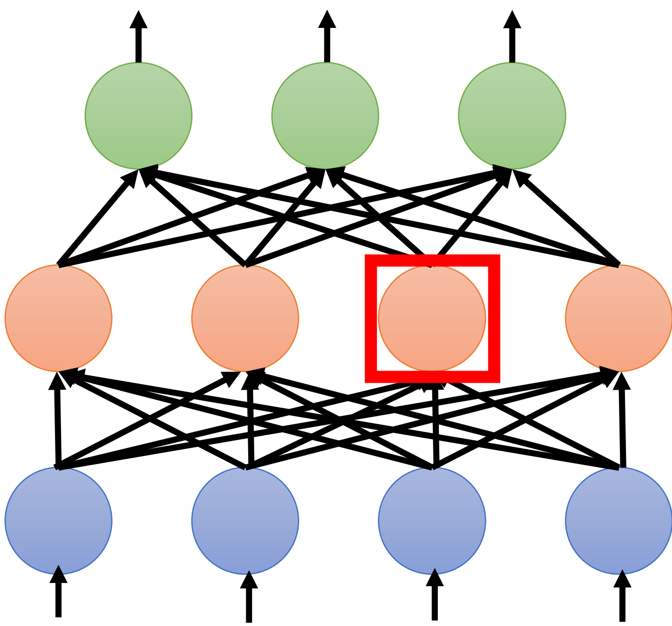


https://arxiv.org/pdf/1608.03665.pdf

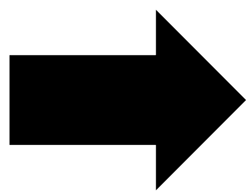
Network Pruning - Practical Issue

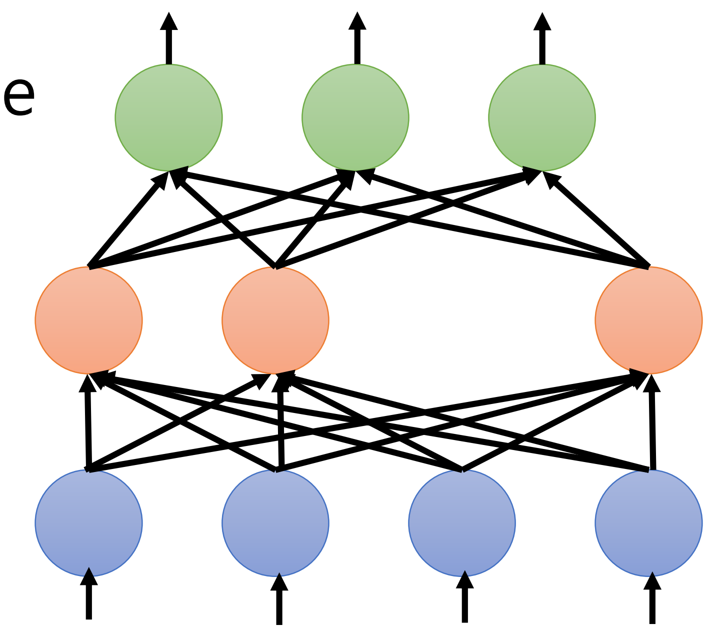
- Neuron pruning

The network architecture is regular.



Prune some neurons



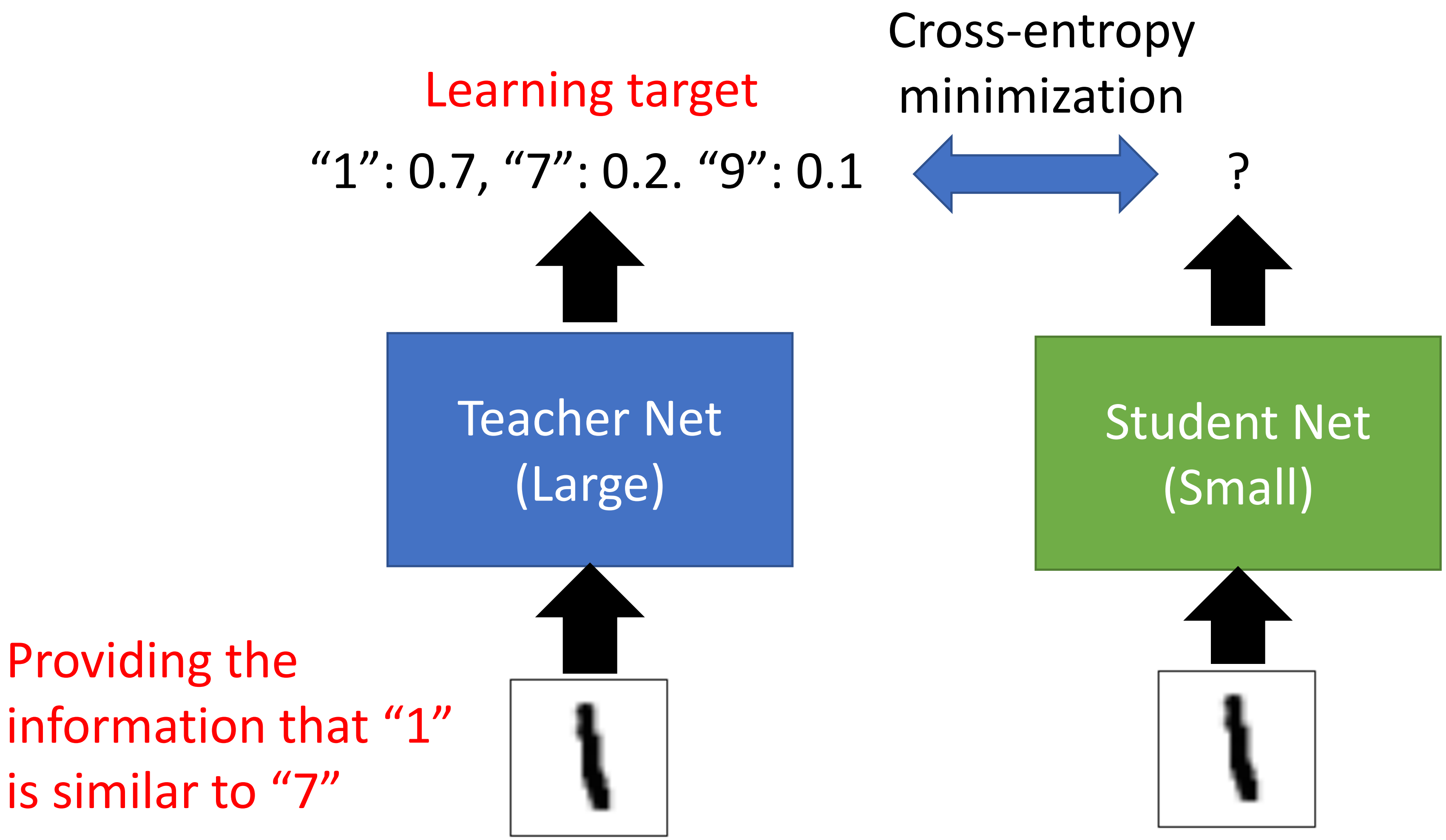


Easy to implement, easy to speedup ……

Knowledge Distillation

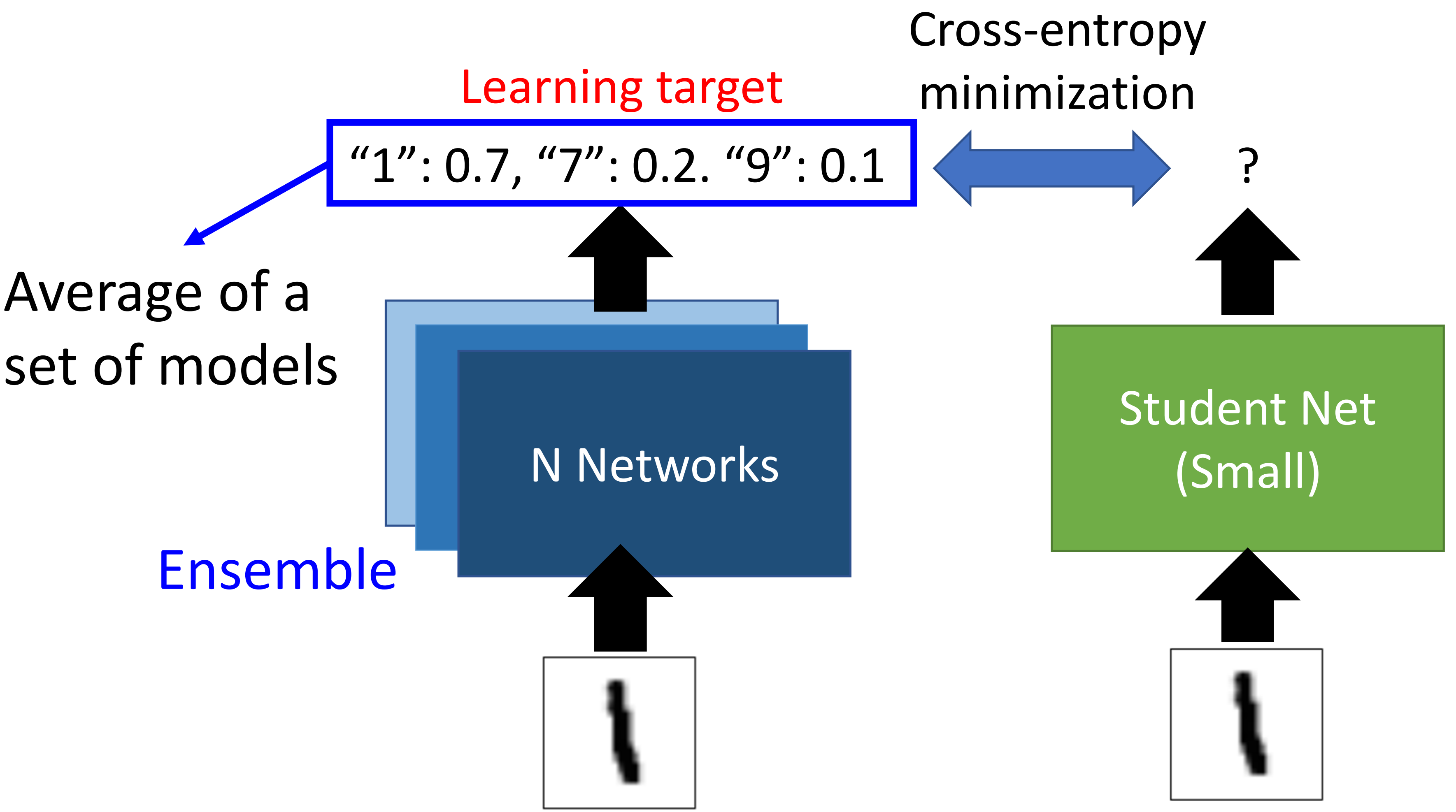
Knowledge Distillation

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



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

- Temperature

$$y_i = \frac{exp(x_i)}{\sum_j exp(x_j)} \xrightarrow[T = 100]{} y_i = \frac{exp(x_i/T)}{\sum_j exp(x_j/T)}$$

$x_1 = 100$ $\quad y_1 = 1$ $\qquad x_1/T = 1$ $\quad y_1 = 0.56$

$x_2 = 10$ $\quad y_2 \approx 0$ $\qquad x_2/T = 0.1$ $\quad y_2 = 0.23$

$x_3 = 1$ $\quad y_3 \approx 0$ $\qquad x_3/T = 0.01$ $\quad y_3 = 0.21$

Parameter Quantization

Parameter Quantization

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

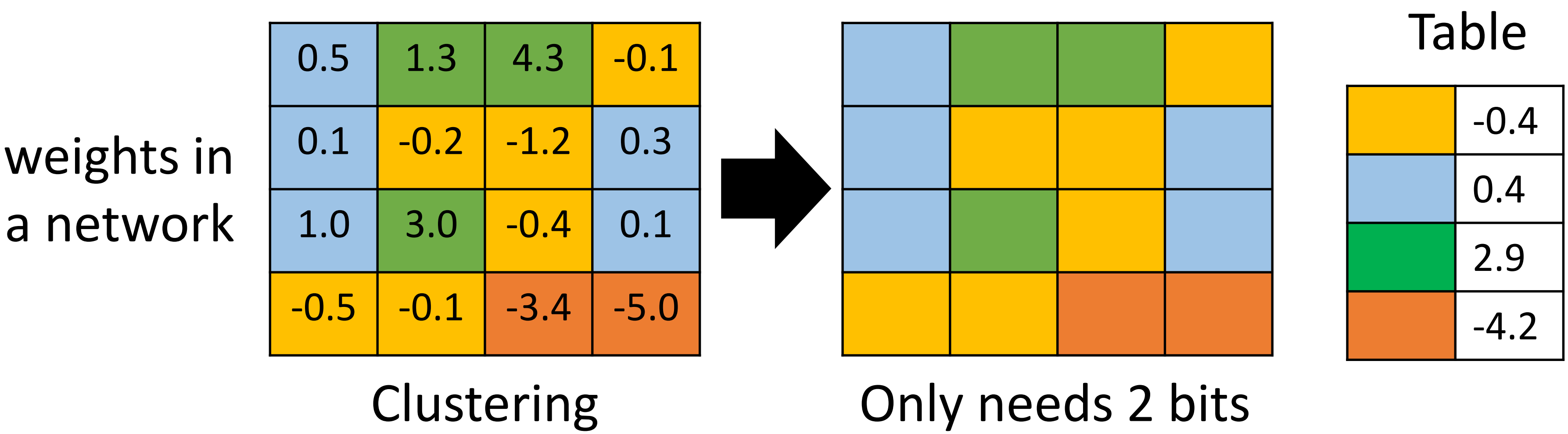
weights in a network

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

Parameter Quantization

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



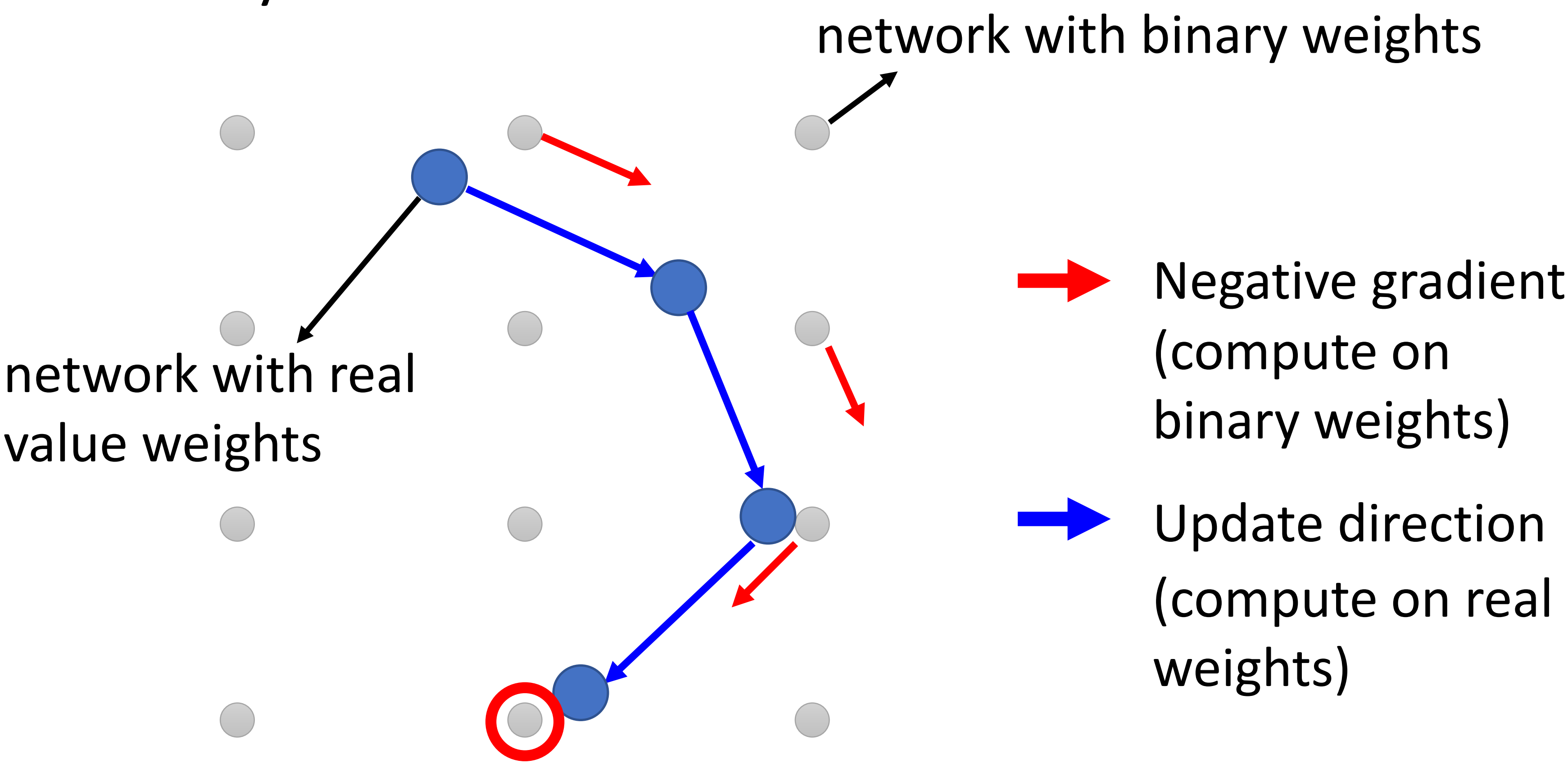
- 3. Represent frequent clusters by less bits, represent rare clusters by more bits
    - e.g. Huffman encoding

Binary Weights

Your weights are always +1 or -1

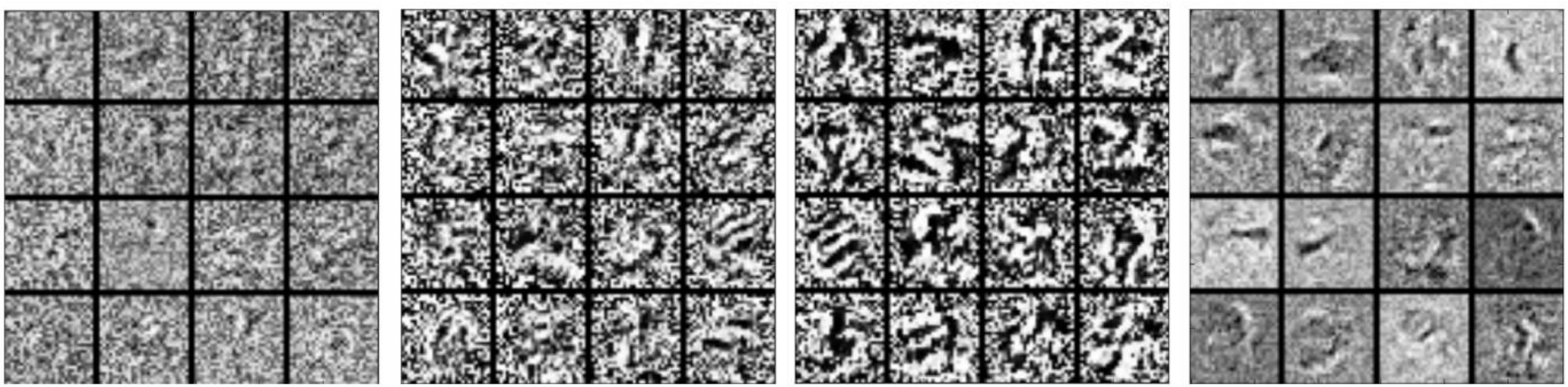
- Binary Connect

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 Connect

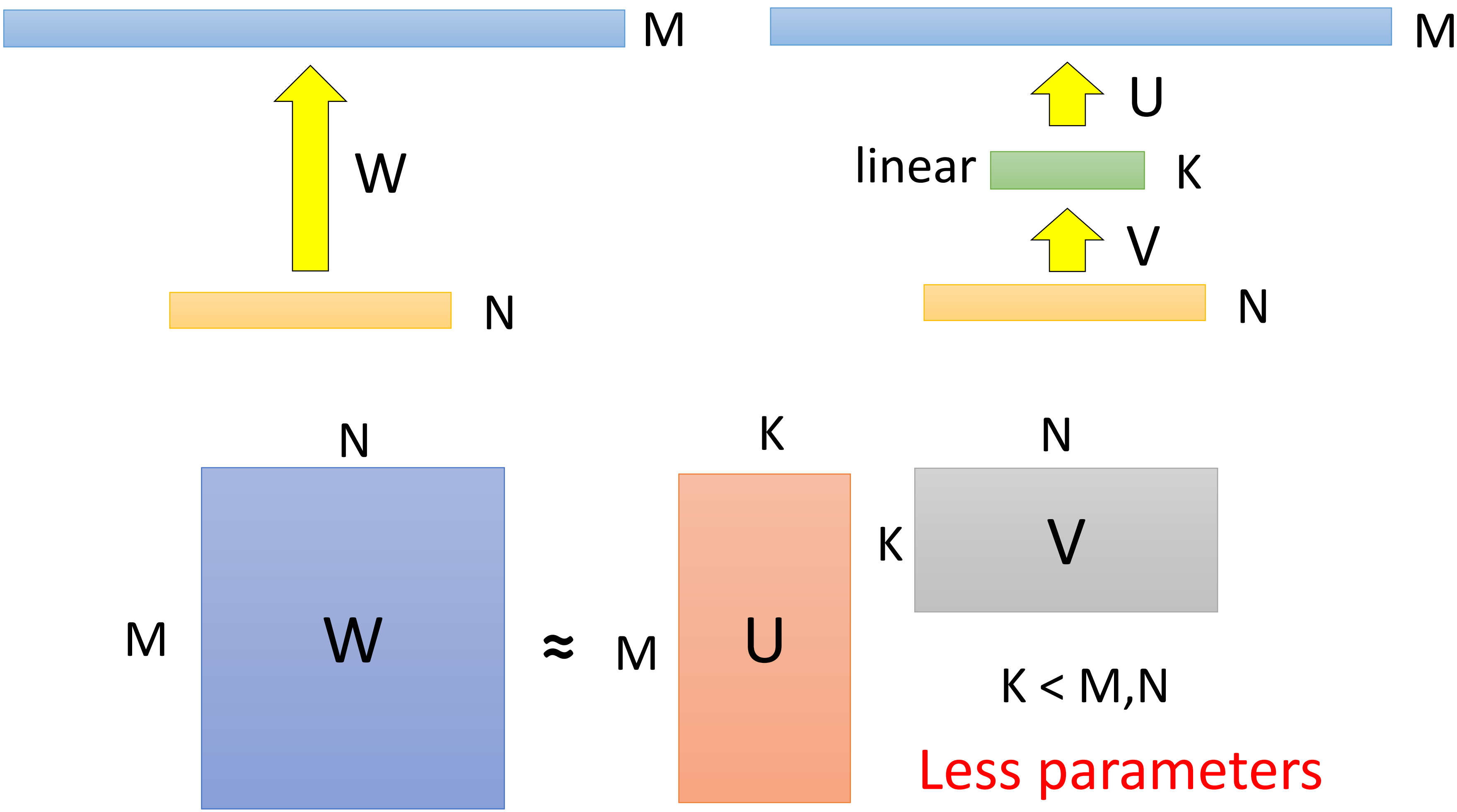
Method	MNIST	CIFAR-10	SVHN
No regularizer	$1.30 \pm 0.04\%$	10.64%	2.44%
BinaryConnect (det.)	$1.29 \pm 0.08\%$	9.90%	2.30%
BinaryConnect (stoch.)	$1.18 \pm 0.04\%$	**8.27%**	2.15%
50% Dropout	$1.01 \pm 0.04\%$		



https://arxiv.org/abs/1511.00363

Architecture Design

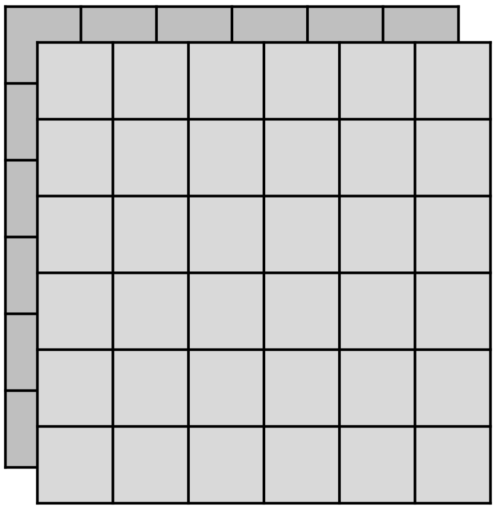
Low rank approximation



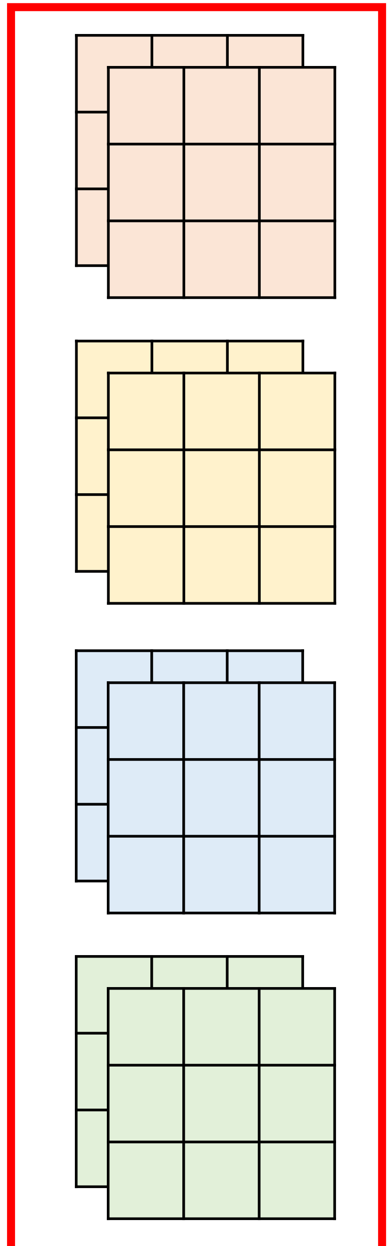
Less parameters

Review: Standard CNN

Input feature map

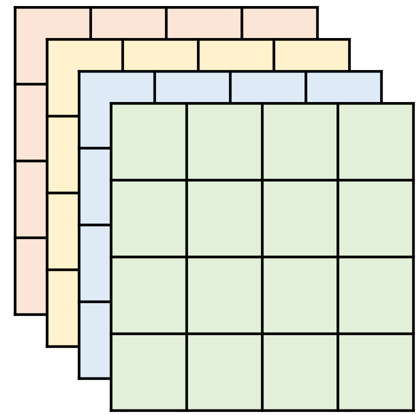


2 channels



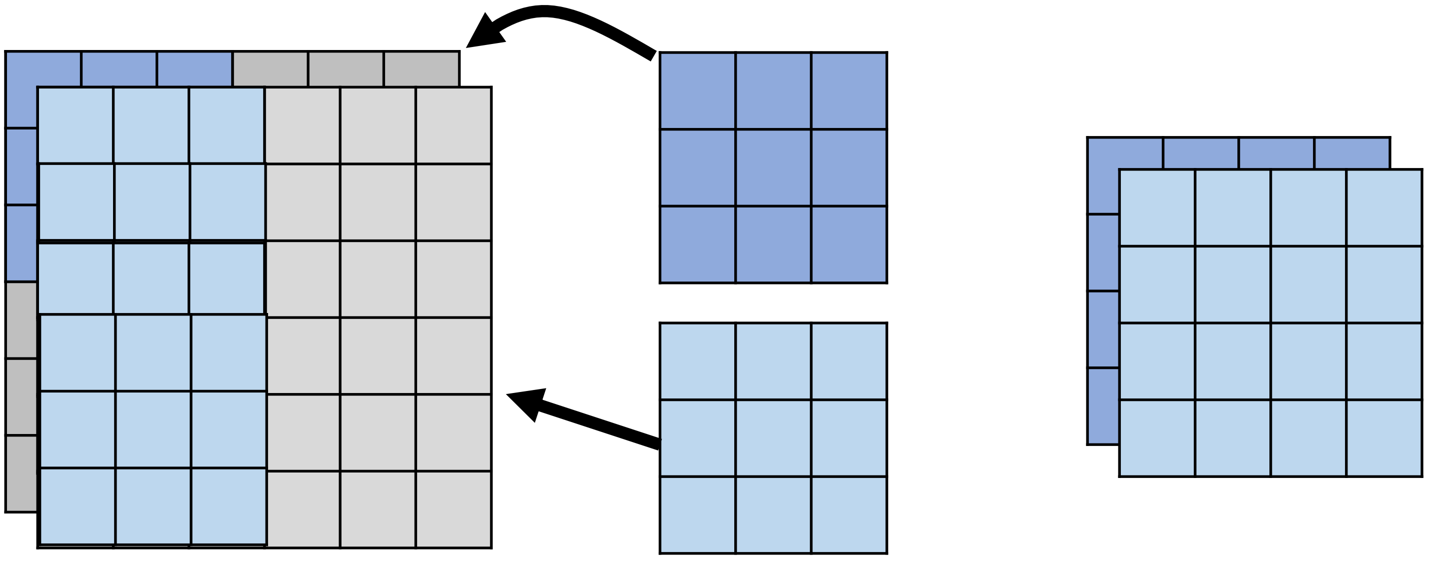
$$3 \times 3 \times 2 \times 4 = 72$$

parameters



Depthwise Separable Convolution

1. Depthwise Convolution

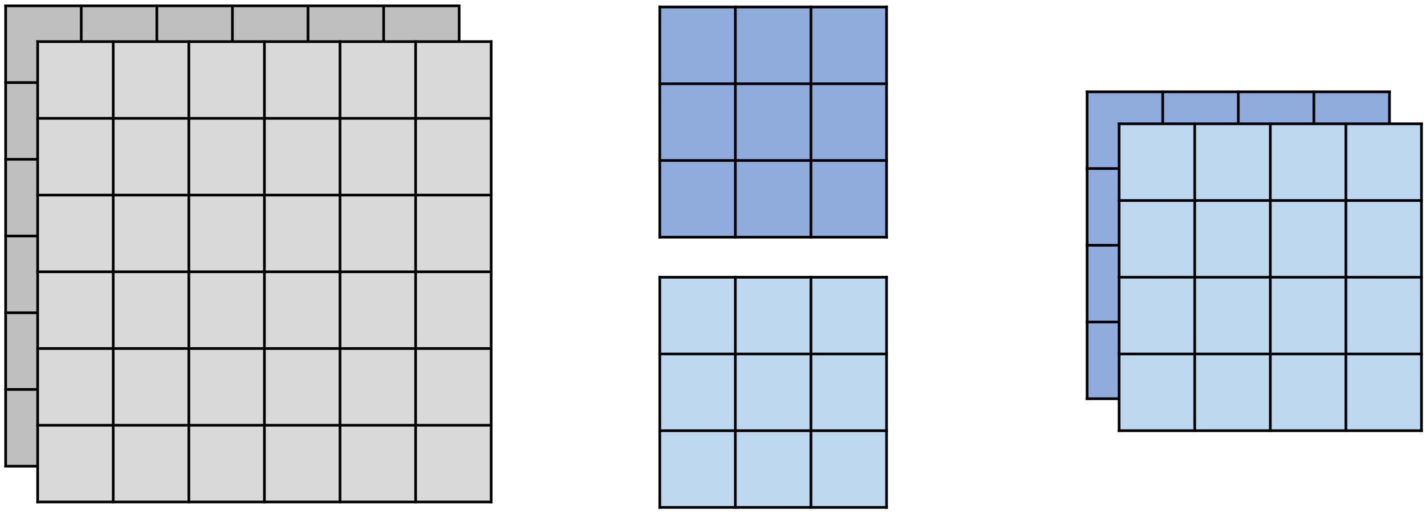


- Filter number = Input channel number
- Each filter only considers one channel.
- The filters are $k \times k$ matrices
- There is no interaction between channels.

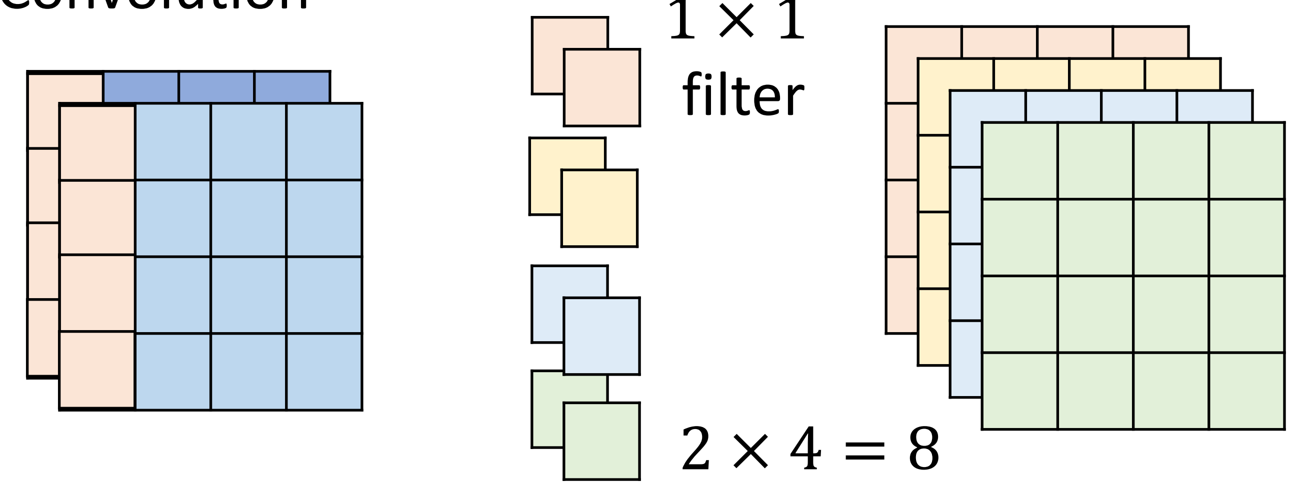
Depthwise Separable Convolution

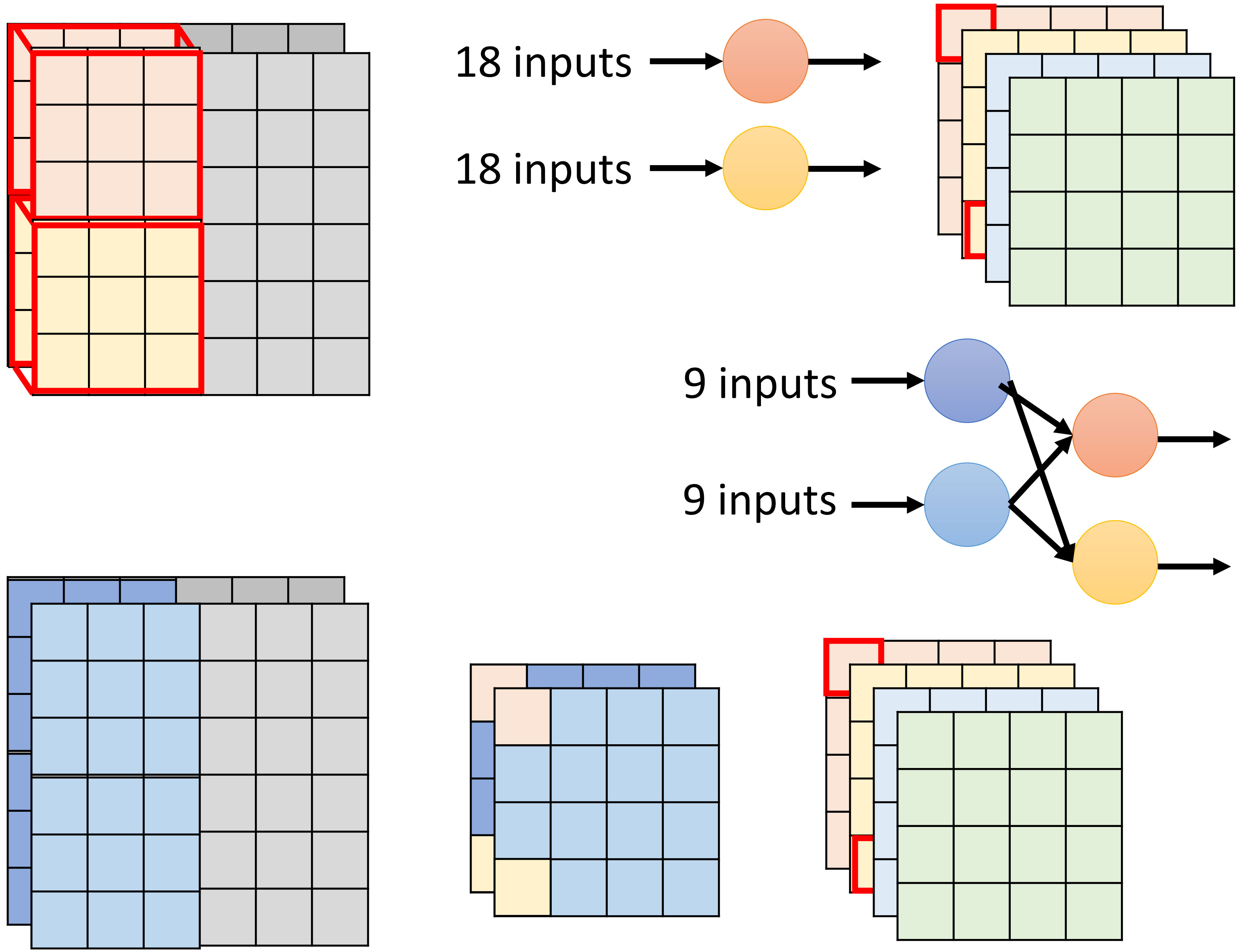
1. Depthwise Convolution

$$3 \times 3 \times 2 = 18$$



2. Pointwise Convolution



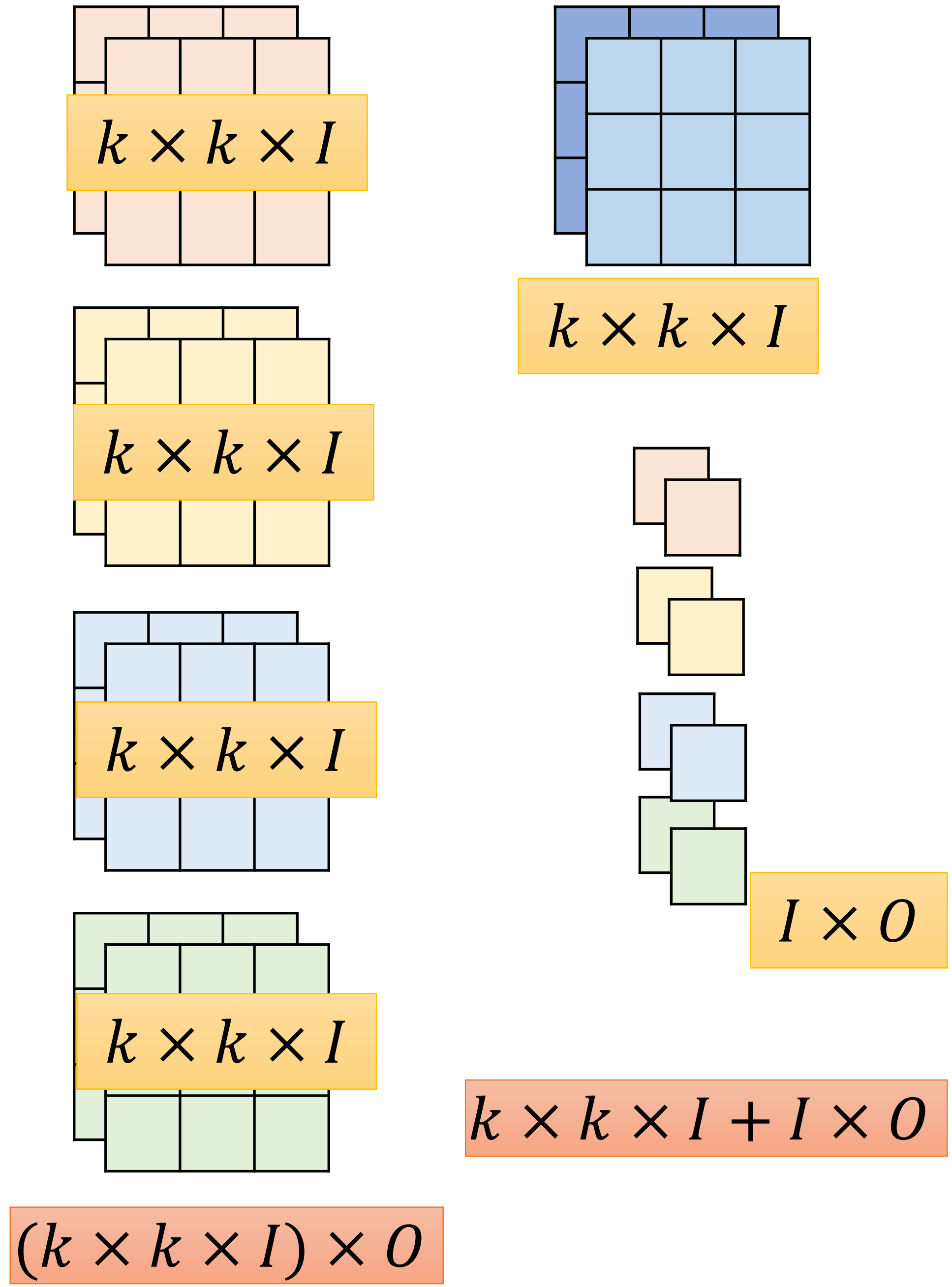

18 inputs
18 inputs
9 inputs
9 inputs

I: number of input channels

O: number of output channels

$k \times k$: kernel size

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



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

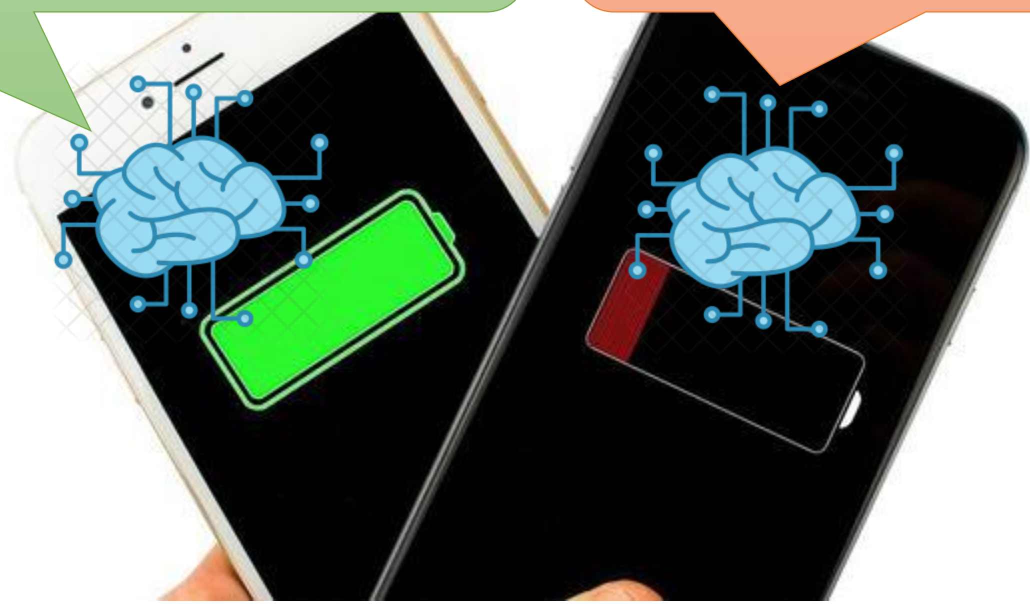
Dynamic Computation

Dynamic Computation

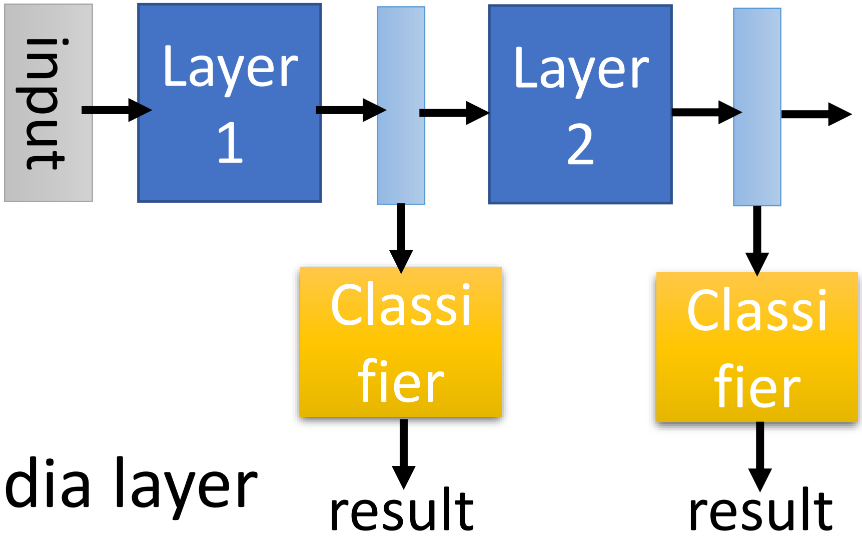
- Can network adjust the computation power it need?

資源充足，那麼就做到最好

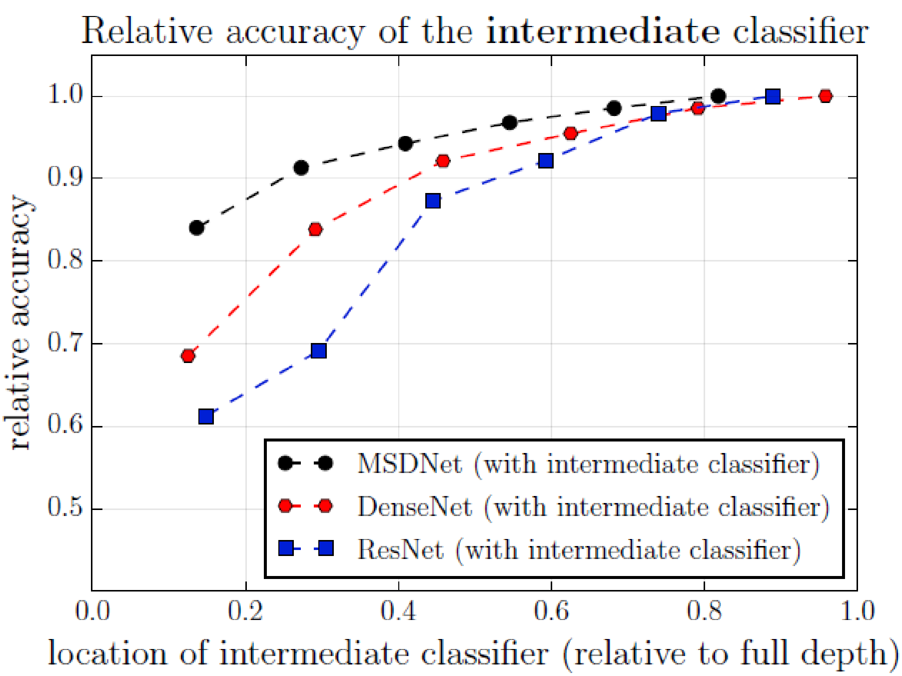
減少運算量，先求有再求好
(但也不要太差)

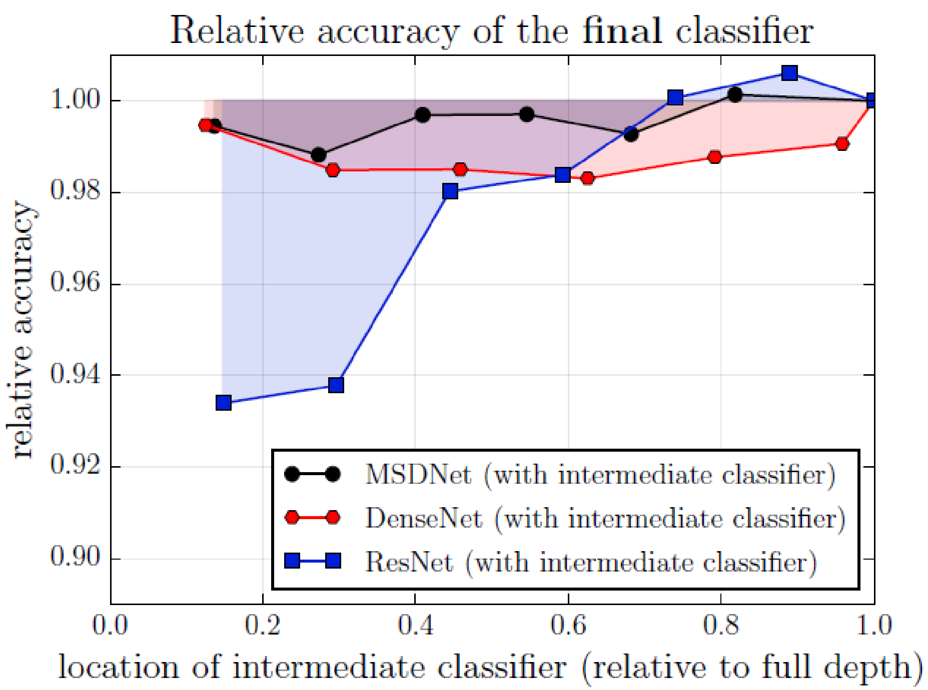


Possible Solutions



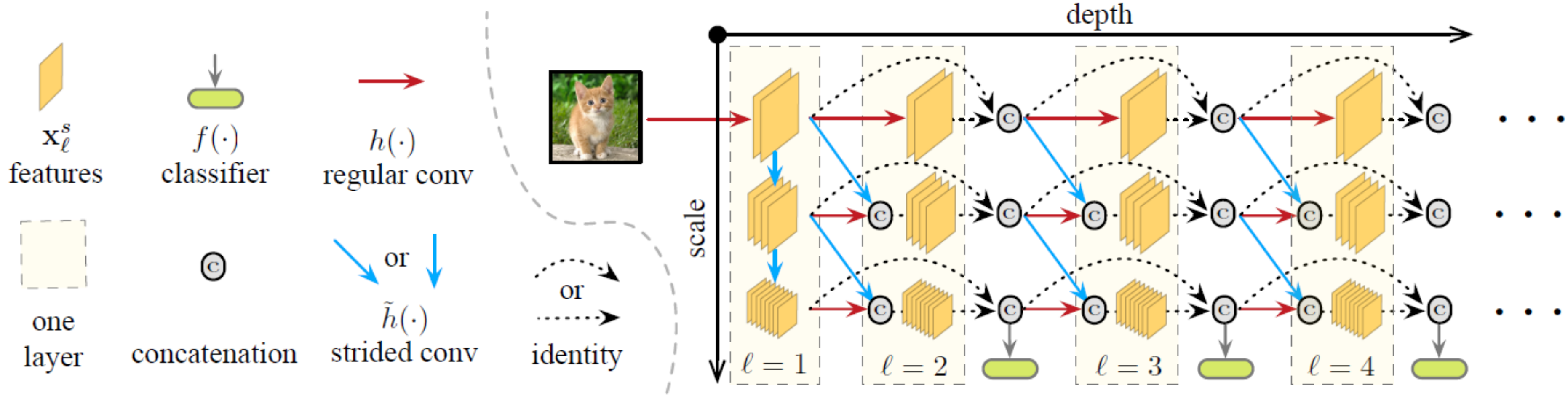
- 1. Train multiple classifiers
- 2. Classifiers at the intermedia layer

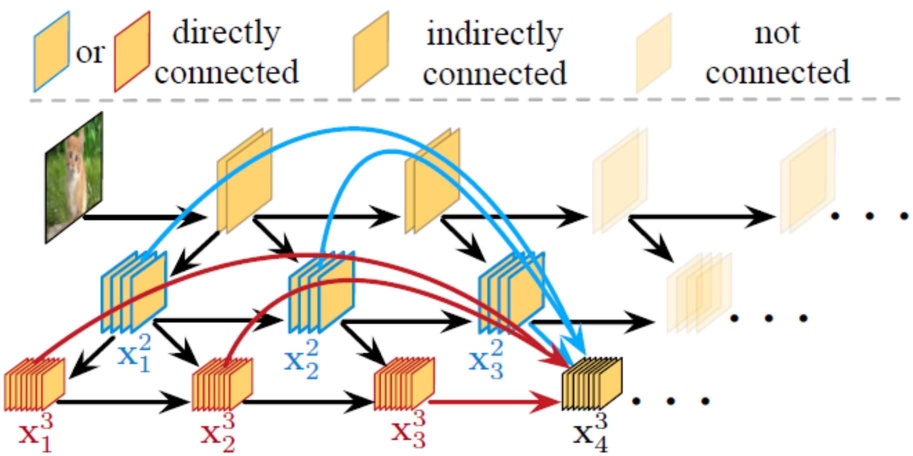




https://arxiv.org/abs/1703.09844

Multi-Scale Dense Networks





x_l^s	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$
$s = 1$	$h_1^1(x_0^1)$	$h_2^1(x_1^1)$	$h_3^1(x_1^1, x_2^1)$	$h_4^1(x_1^1, x_2^1, x_3^1)$
$s = 2$	$\tilde{h}_1^2(x_1^1)$	$\begin{bmatrix} \tilde{h}_2^2([x_1^1]) \\ h_2^2([x_1^2]) \end{bmatrix}$	$\begin{bmatrix} \tilde{h}_3^2([x_1^1, x_2^1]) \\ h_3^2([x_1^2, x_2^2]) \end{bmatrix}$	$\begin{bmatrix} \tilde{h}_4^2([x_1^1, x_2^1, x_3^1]) \\ h_4^2([x_1^2, x_2^2, x_3^2]) \end{bmatrix}$
$s = 3$	$\tilde{h}_1^3(x_1^2)$	$\begin{bmatrix} \tilde{h}_2^3([x_1^2]) \\ h_2^3([x_1^3]) \end{bmatrix}$	$\begin{bmatrix} \tilde{h}_3^3([x_1^2, x_2^2]) \\ h_3^3([x_1^3, x_2^3]) \end{bmatrix}$	$\begin{bmatrix} \tilde{h}_4^3([x_1^2, x_2^2, x_3^2]) \\ h_4^3([x_1^3, x_2^3, x_3^3]) \end{bmatrix}$

https://arxiv.org/abs/1703.09844

Concluding Remarks

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