NETWORK COMPRESSION

Hung-yi Lee 李宏毅

Smaller Model

Less parameters

Deploying ML models in resource-constrained environments

Lower latency, Privacy, etc.

Outline

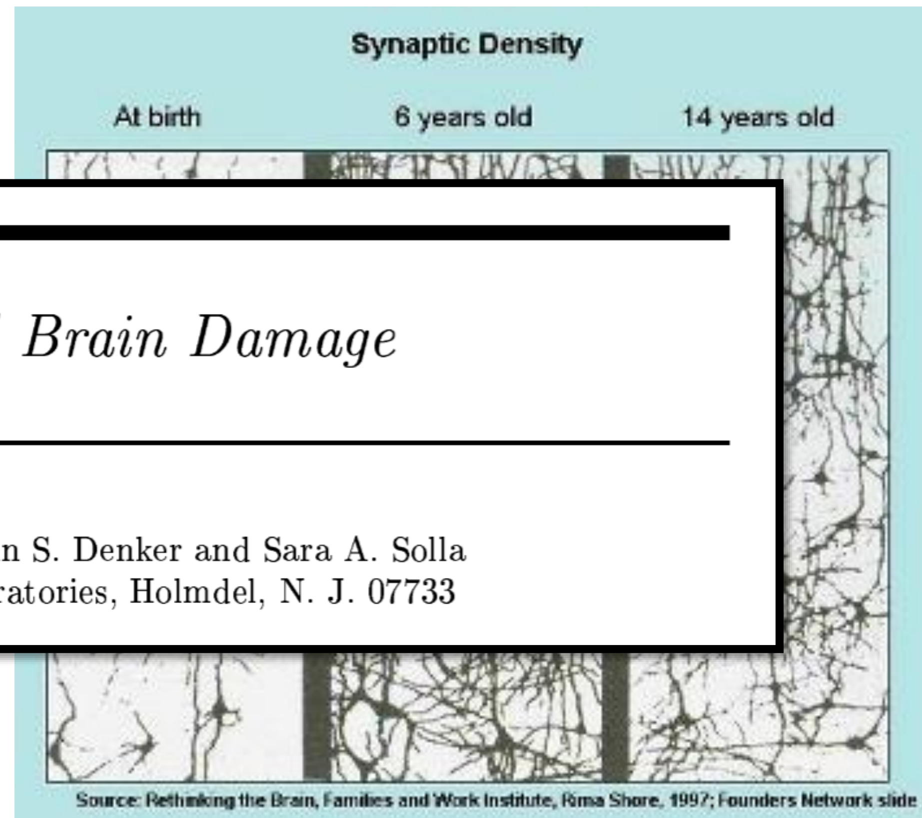
- Network Pruning
- Knowledge Distillation
- Parameter Quantization
- Architecture Design
- Dynamic Computation

We will not talk about hard-ware solution today.

Network Pruning

Network can be pruned

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



Optimal Brain Damage

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

(NIPS, 1989)

Network Pruning

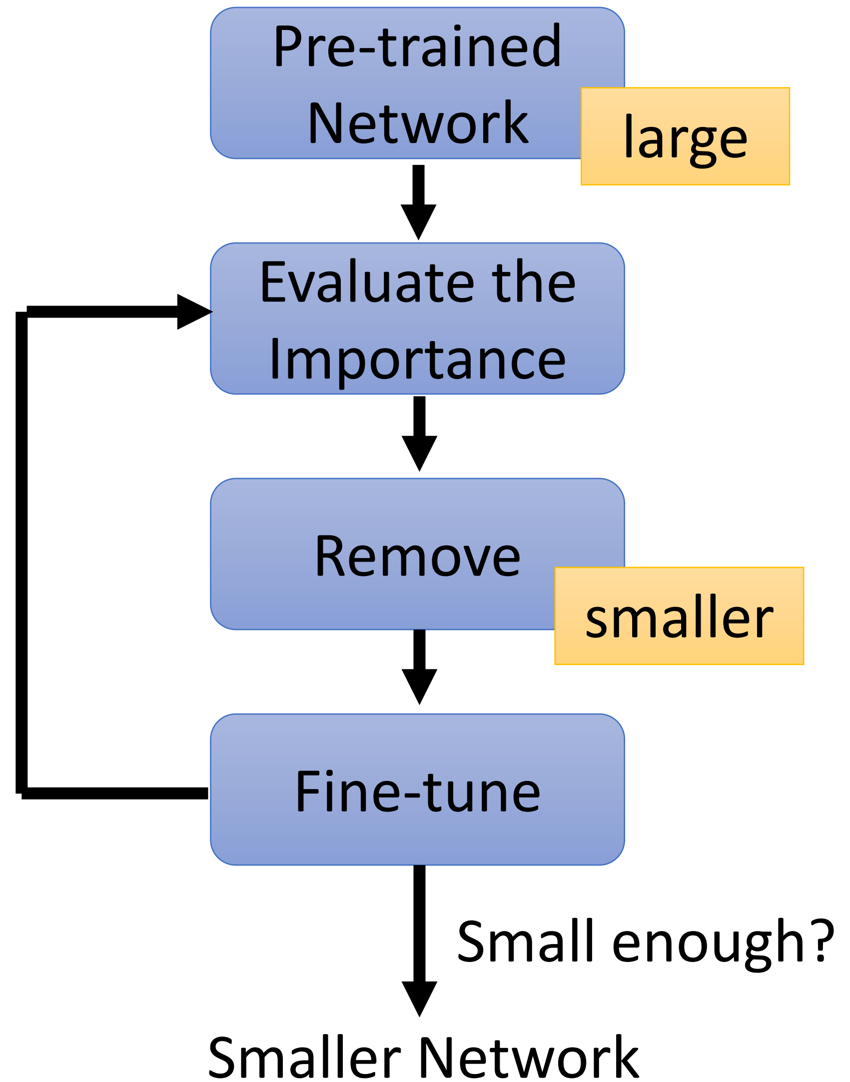
- Importance of a weight:

  absolute values, life long ...

- Importance of a neuron:

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

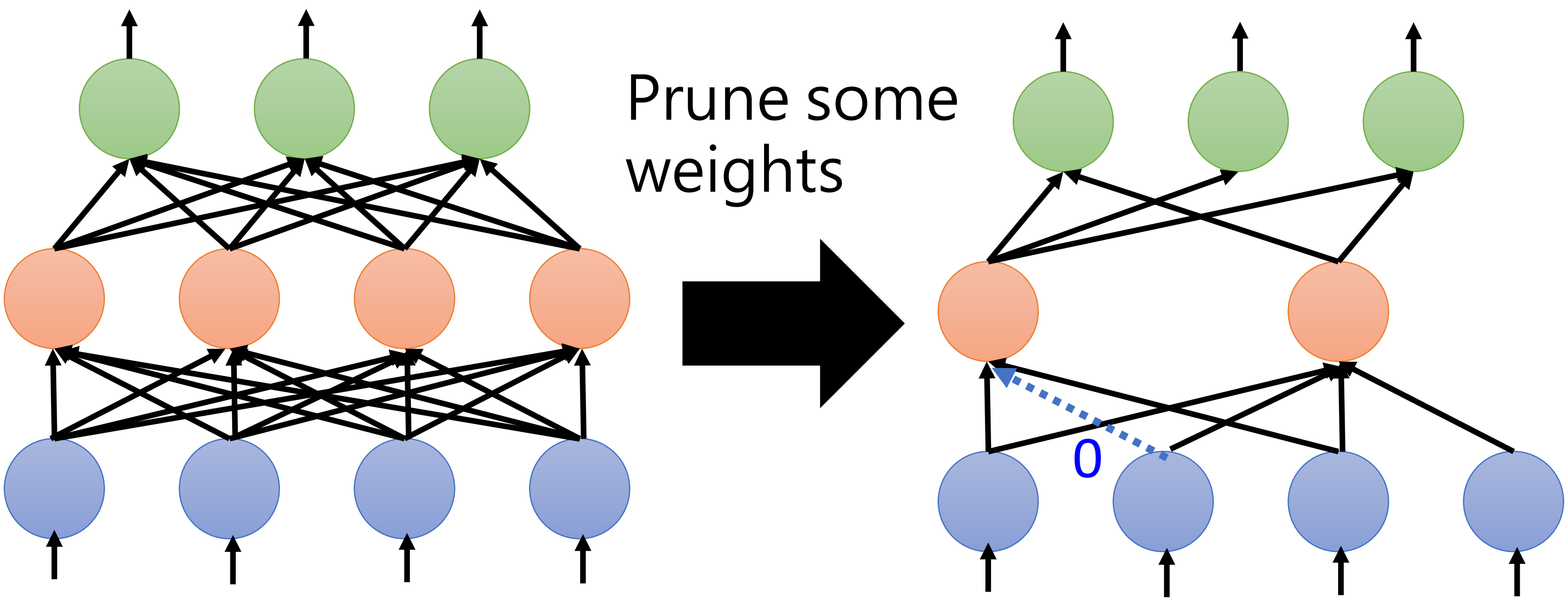
- After pruning, the accuracy will drop (hopefully not too much)
- Fine-tuning on training data for recover
- Don't prune too much at once, or the network won't recover.



Network Pruning - Practical Issue

- Weight pruning

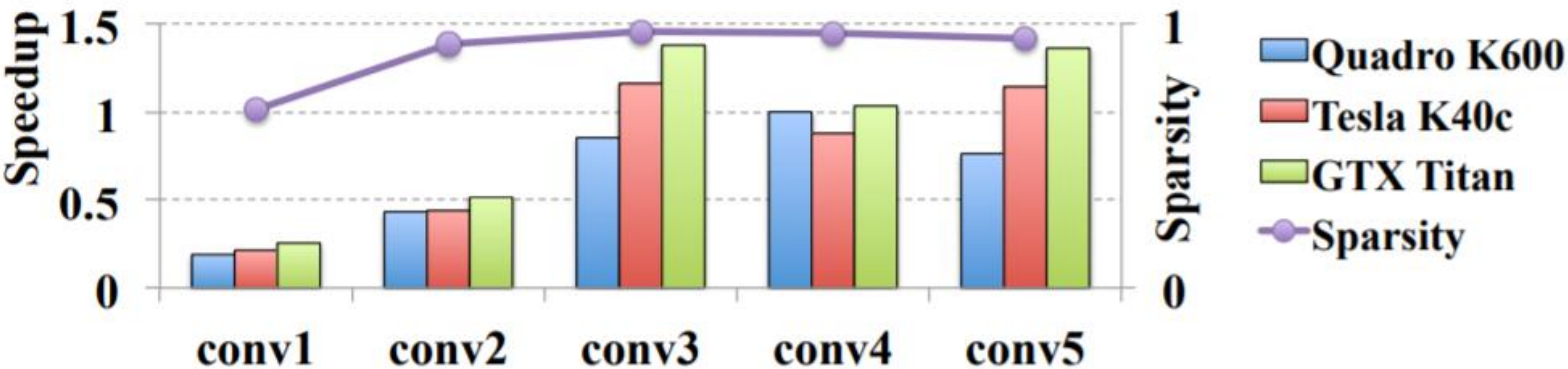
The network architecture becomes irregular.



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

Network Pruning - Practical Issue

- Weight pruning

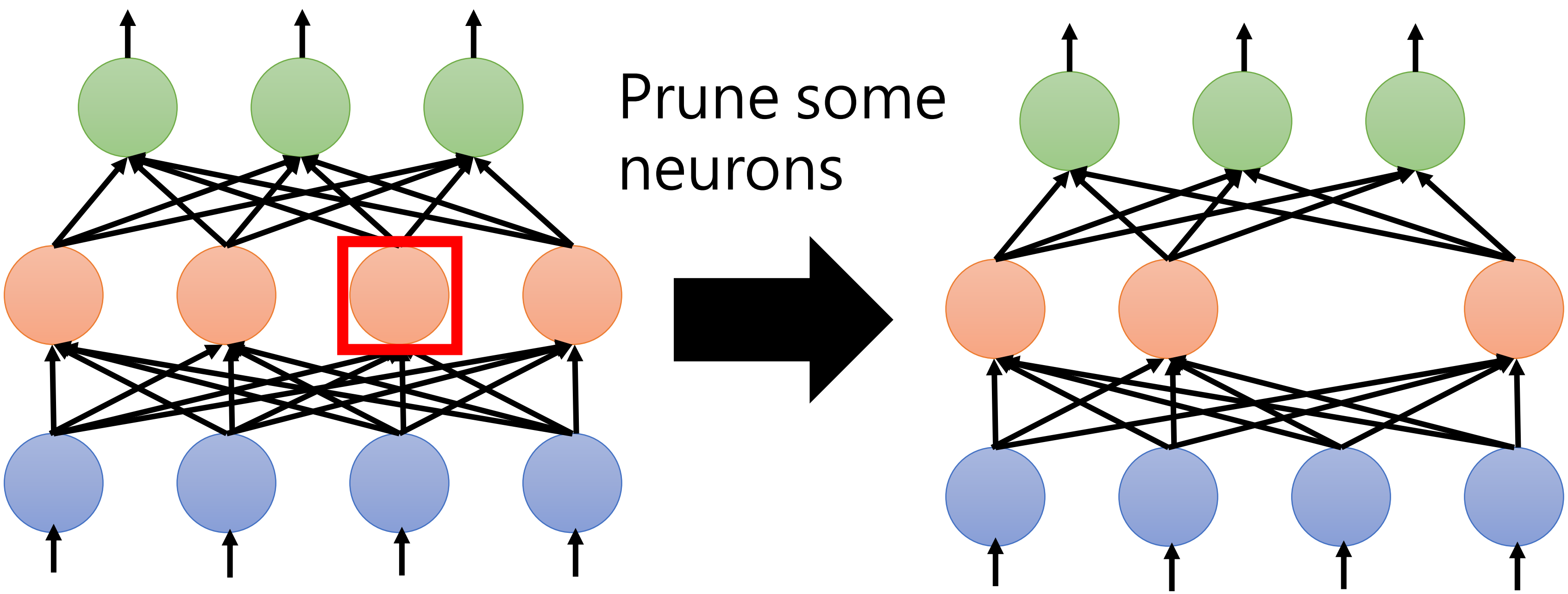


https://arxiv.org/pdf/1608.03665.pdf

Network Pruning - Practical Issue

- Neuron pruning

The network architecture is regular.



Easy to implement, easy to speedup ……

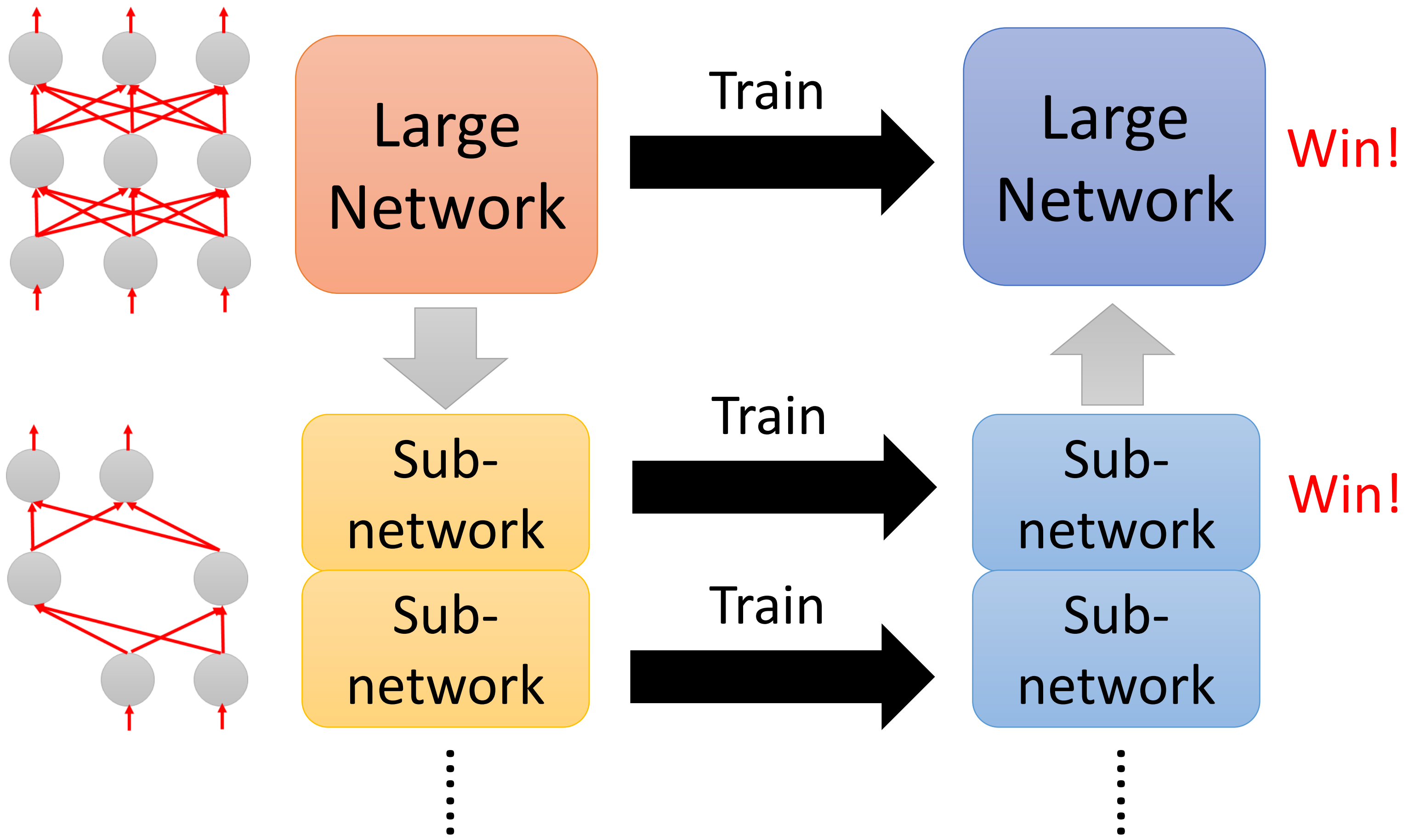
Why Pruning?

- 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

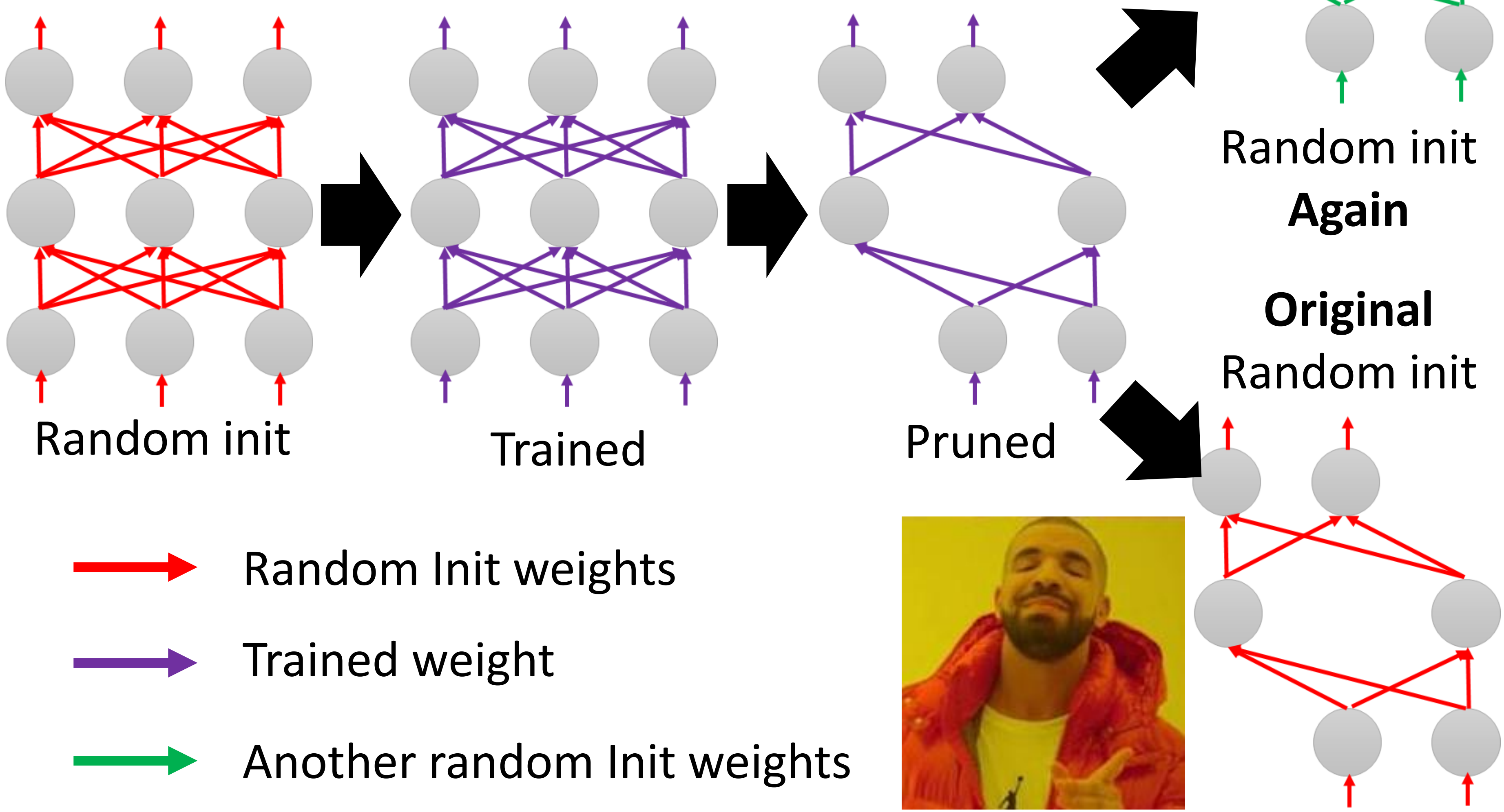
Why Pruning?

Lottery Ticket Hypothesis



Why Pruning?

Lottery Ticket Hypothesis

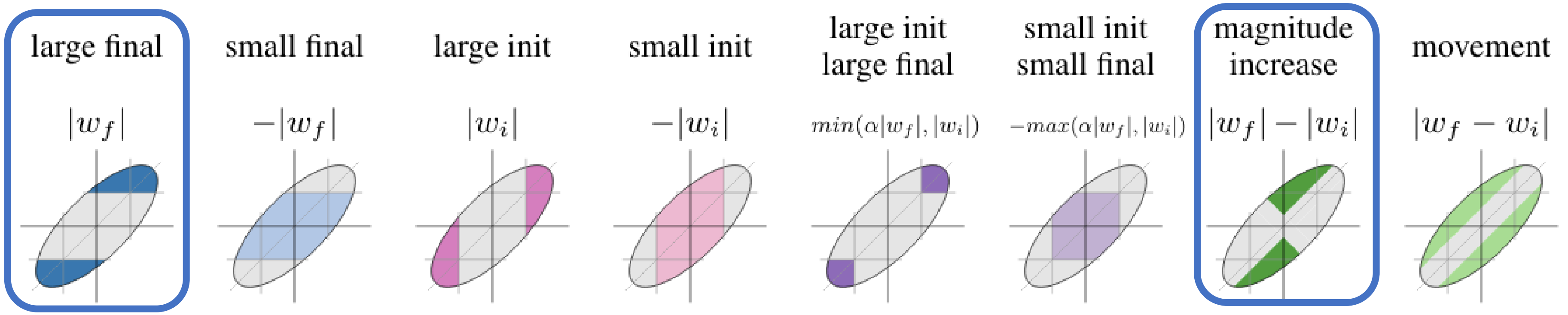


Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask
https://arxiv.org/abs/1905.01067

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

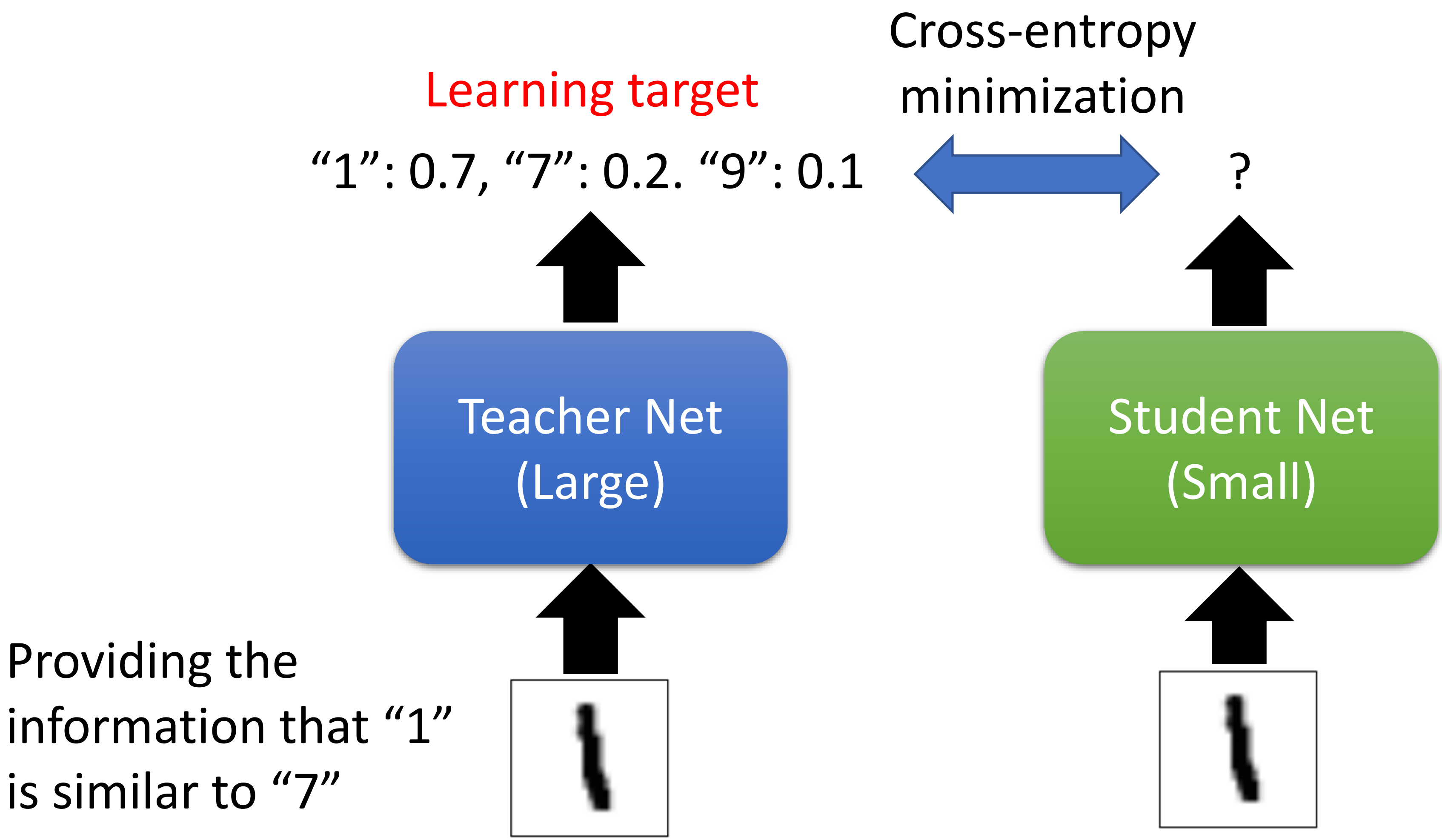
Dataset	Model	Unpruned	Pruned Model	Fine-tuned	Scratch-E	Scratch-B
CIFAR-10	VGG-16	93.63 ($\pm$0.16)	VGG-16-A	93.41 ($\pm$0.12)	93.62 ($\pm$0.11)	**93.78** ($\pm$0.15)
	ResNet-56	93.14 ($\pm$0.12)	ResNet-56-A	92.97 ($\pm$0.17)	92.96 ($\pm$0.26)	**93.09** ($\pm$0.14)
			ResNet-56-B	92.67 ($\pm$0.14)	92.54 ($\pm$0.19)	**93.05** ($\pm$0.18)
	ResNet-110	93.14 ($\pm$0.24)	ResNet-110-A	93.14 ($\pm$0.16)	**93.25** ($\pm$0.29)	93.22 ($\pm$0.22)
			ResNet-110-B	92.69 ($\pm$0.09)	92.89 ($\pm$0.43)	**93.60** ($\pm$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

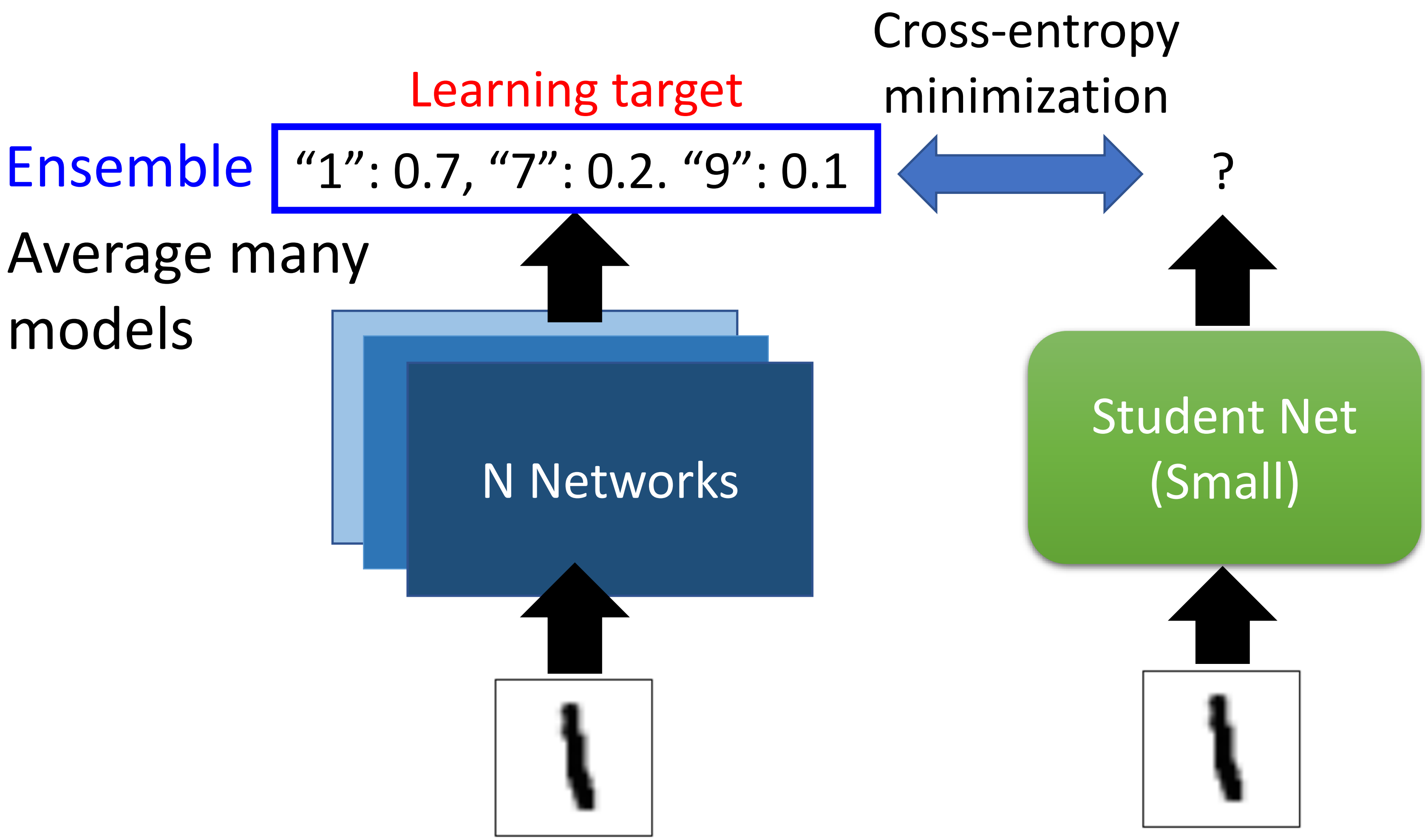
Knowledge Distillation

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



Knowledge Distillation

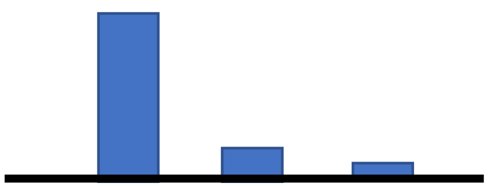
Knowledge Distillation
https://arxiv.org/pdf/1503.02531.pdf
Do Deep Nets Really Need to be Deep?
https://arxiv.org/pdf/1312.6184.pdf

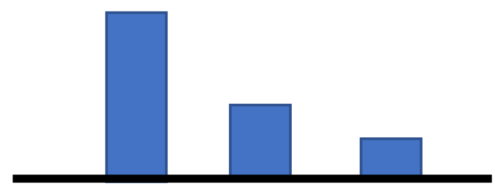


Knowledge Distillation

- Temperature for softmax

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





$y_1 = 100 \qquad y_1' = 1$

$y_2 = 10 \qquad y_2' \approx 0$

$y_3 = 1 \qquad y_3' \approx 0$

$y_1/T = 1 \qquad y_1' = 0.56$

$y_2/T = 0.1 \qquad y_2' = 0.23$

$y_3/T = 0.01 \qquad y_3' = 0.21$

Parameter Quantization

Parameter Quantization

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

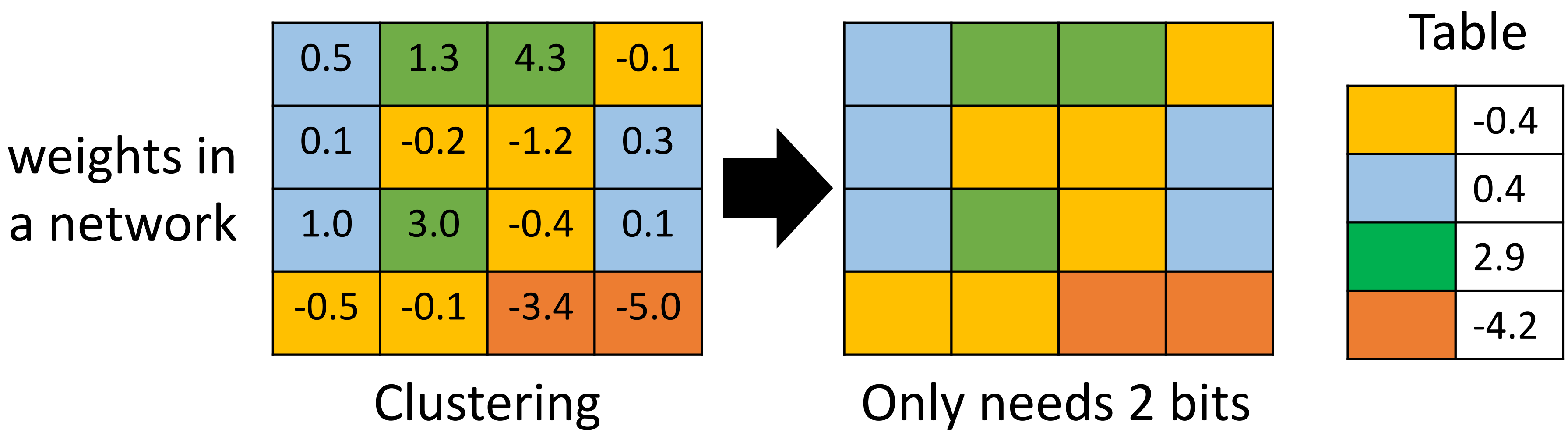
weights in a network

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



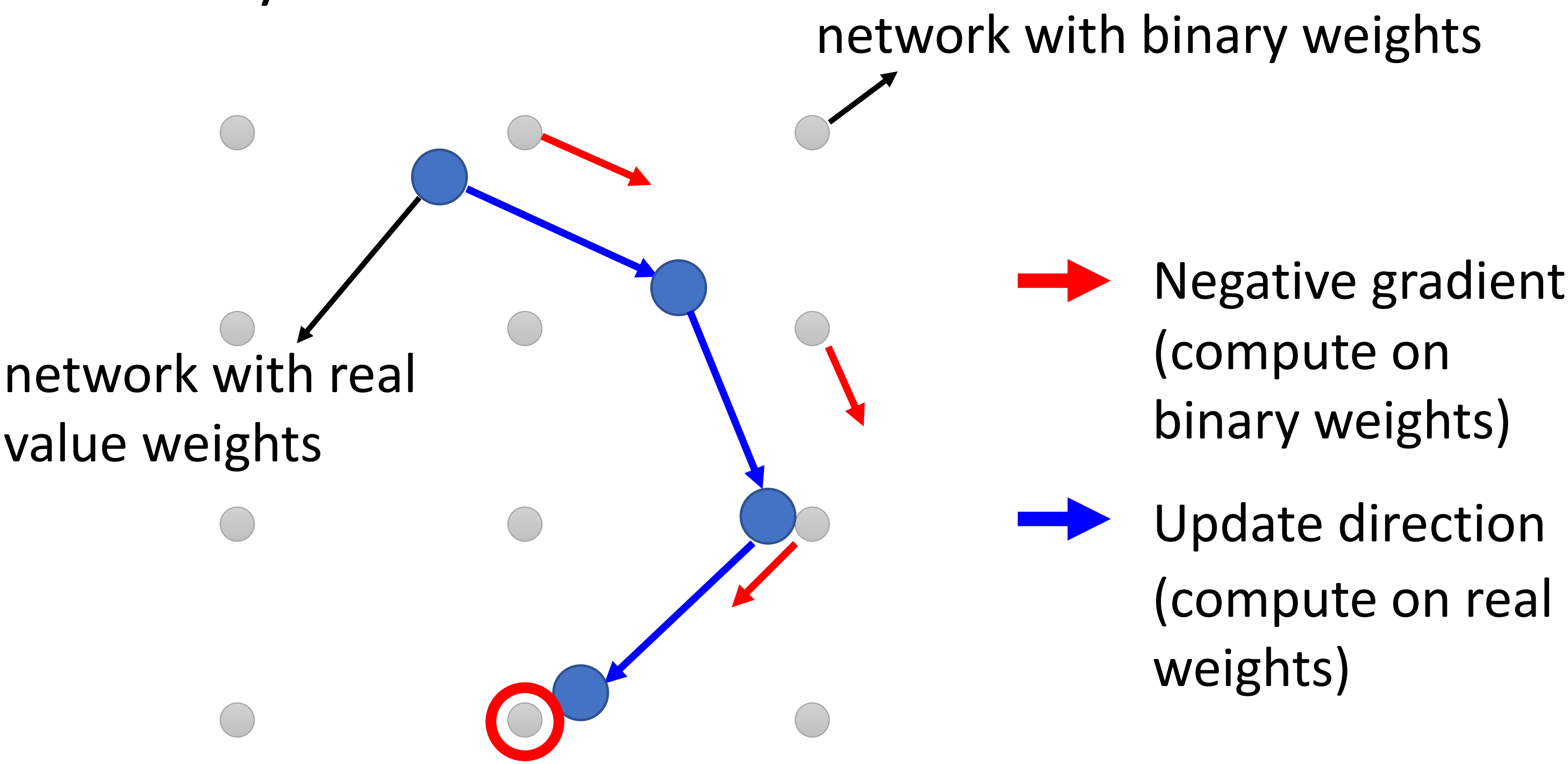
- 3. Represent frequent clusters by less bits, represent rare clusters by more bits
  - e.g. Huffman encoding

Binary Weights

Your weights are always +1 or -1

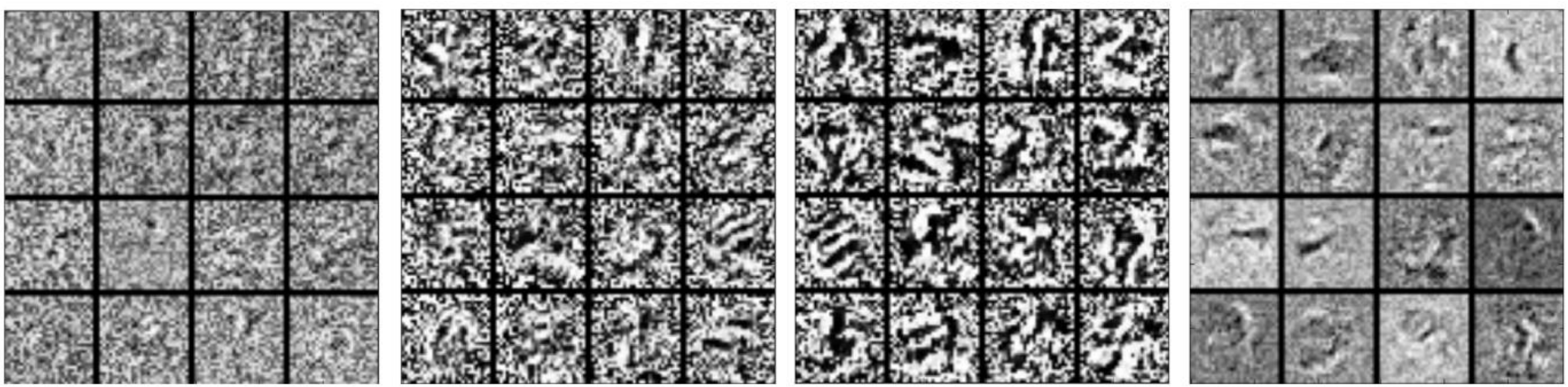
Binary Connect:
https://arxiv.org/abs/1511.00363
Binary Network:
https://arxiv.org/abs/1602.02830
XNOR-net:
https://arxiv.org/abs/1603.05279

- Binary Connect



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

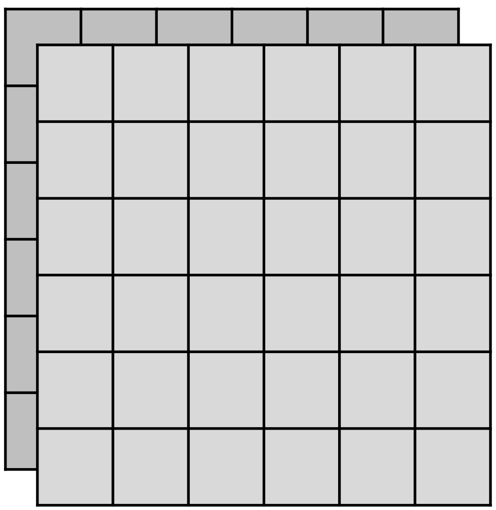
Depthwise Separable Convolution

Review: Standard CNN

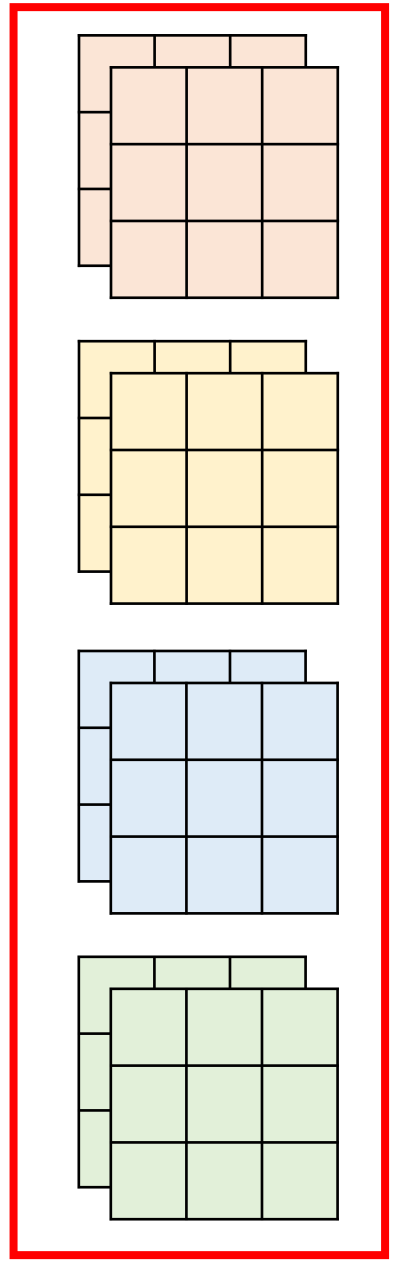
$$3 \times 3 \times 2 \times 4 = 72$$

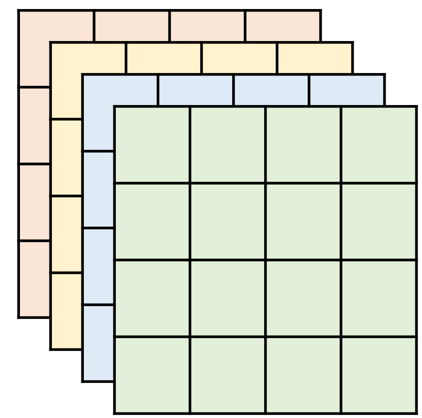
parameters

Input feature map



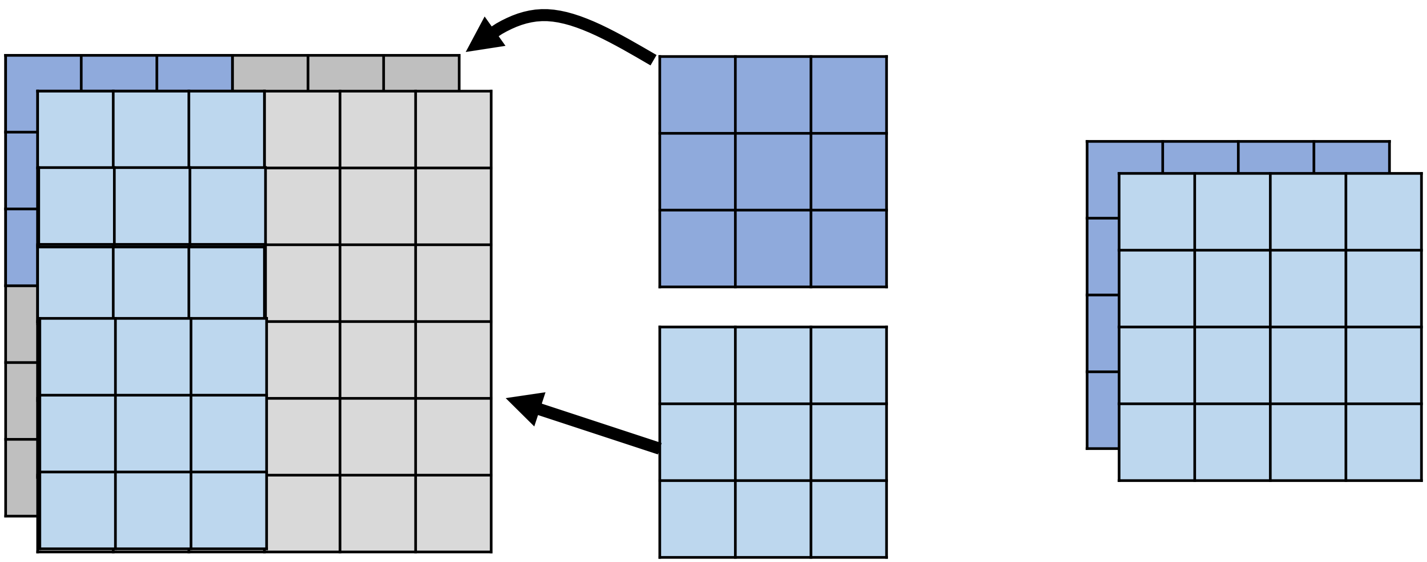
2 channels





Depthwise Separable Convolution

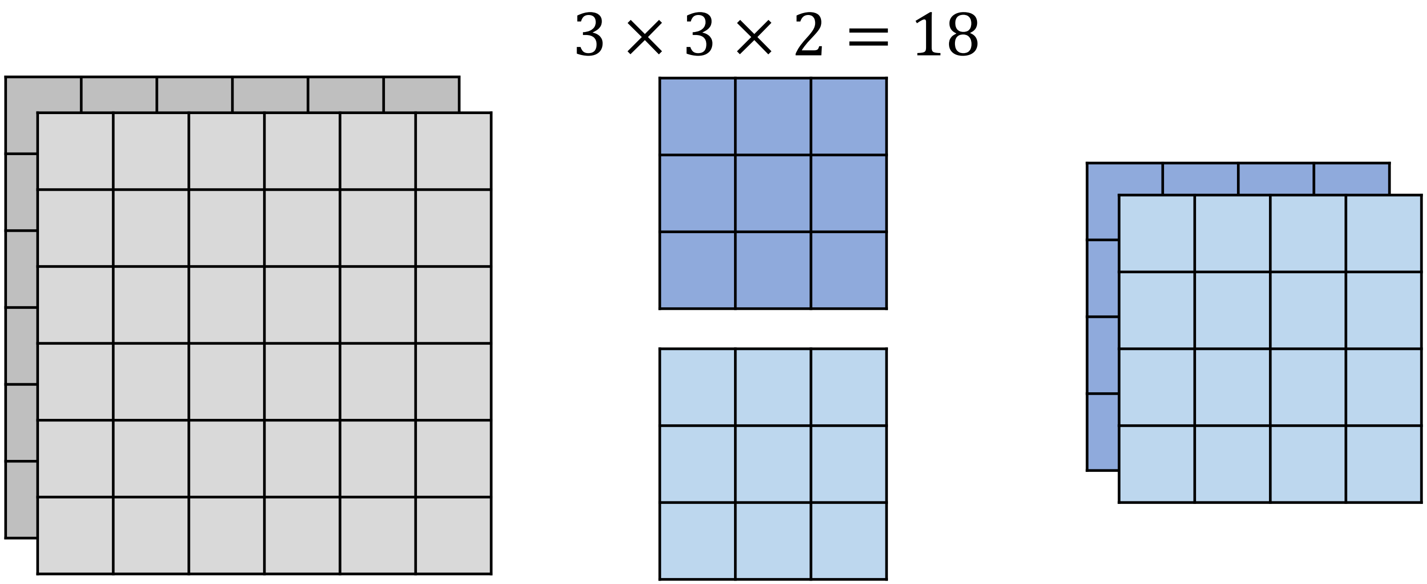
1. Depthwise Convolution



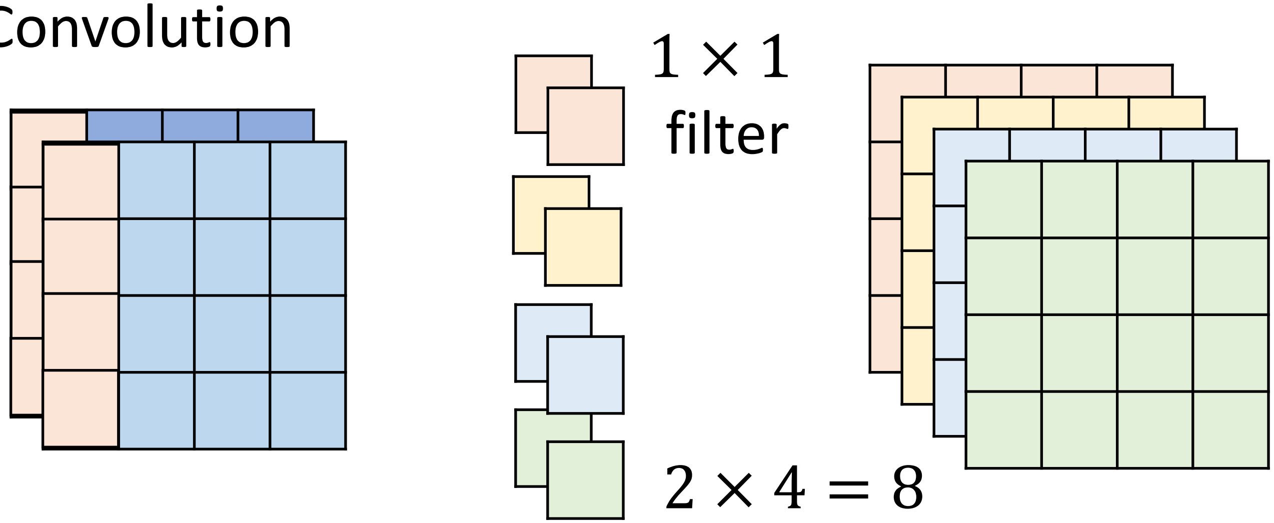
- Filter number = Input channel number
- Each filter only considers one channel.
- The filters are $k \times k$ matrices
- There is no interaction between channels.

Depthwise Separable Convolution

1. Depthwise Convolution



2. Pointwise Convolution

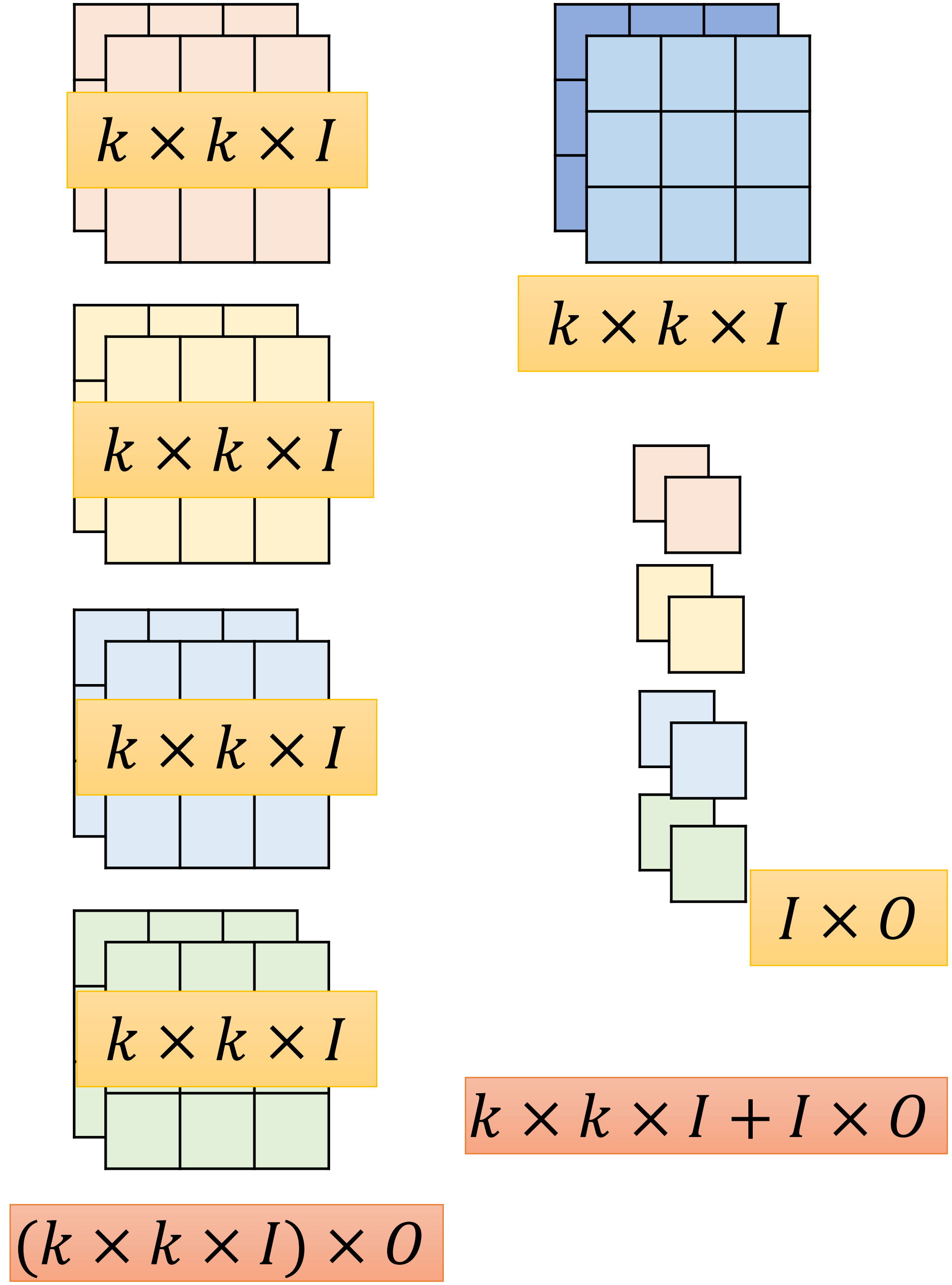


I: number of input channels

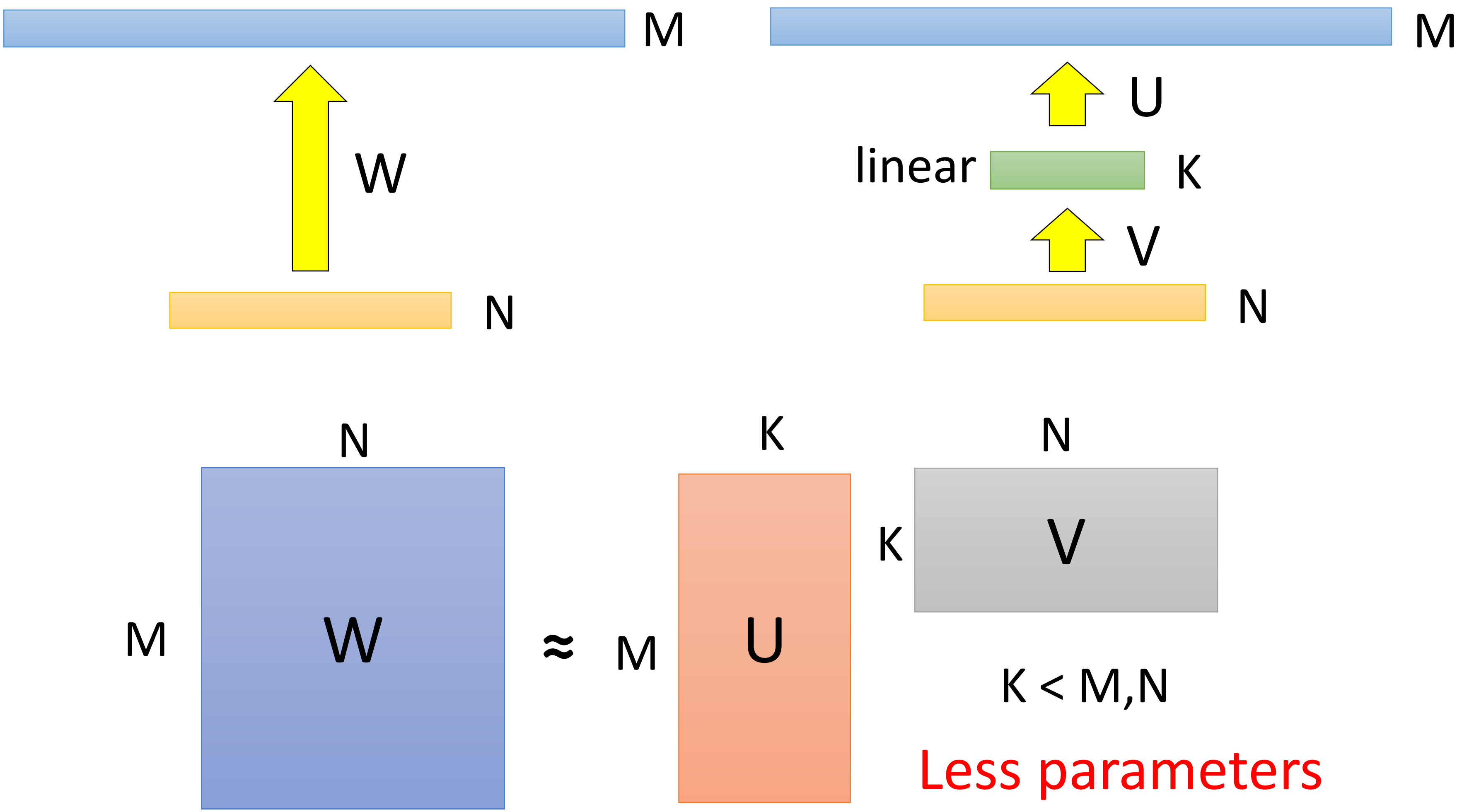
O: number of output channels

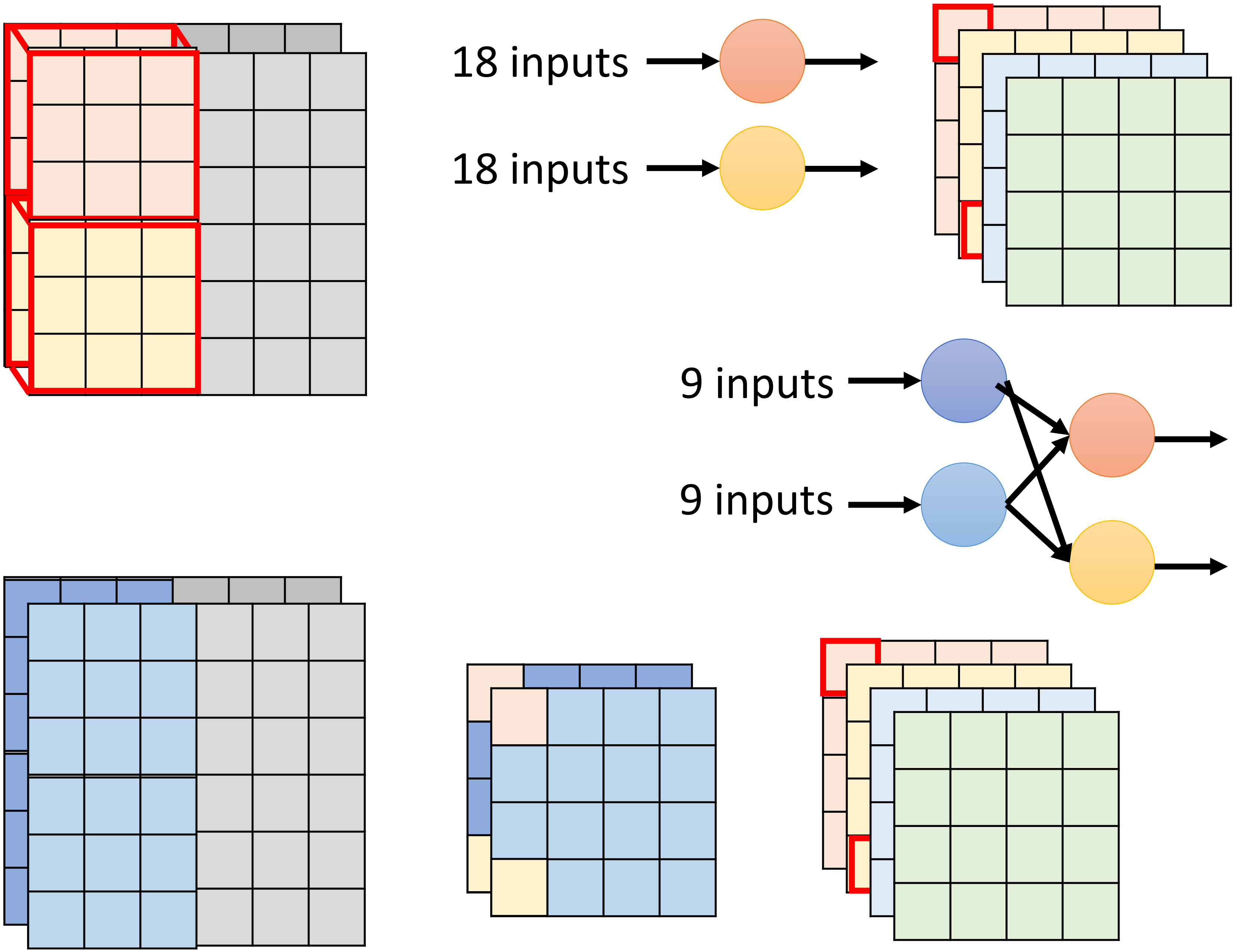
$k \times k$: kernel size

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



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

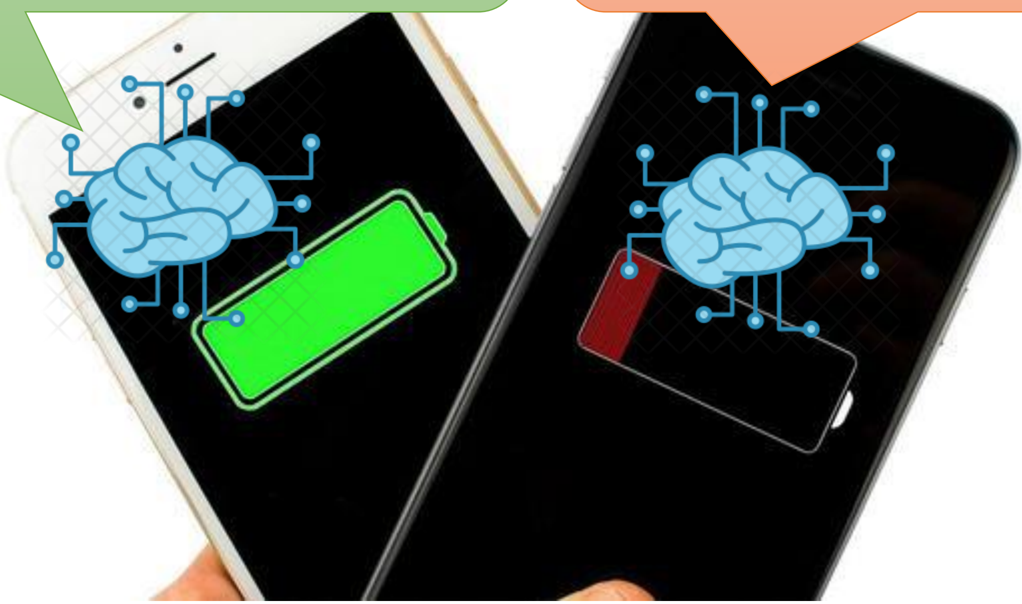
Dynamic Computation

Dynamic Computation

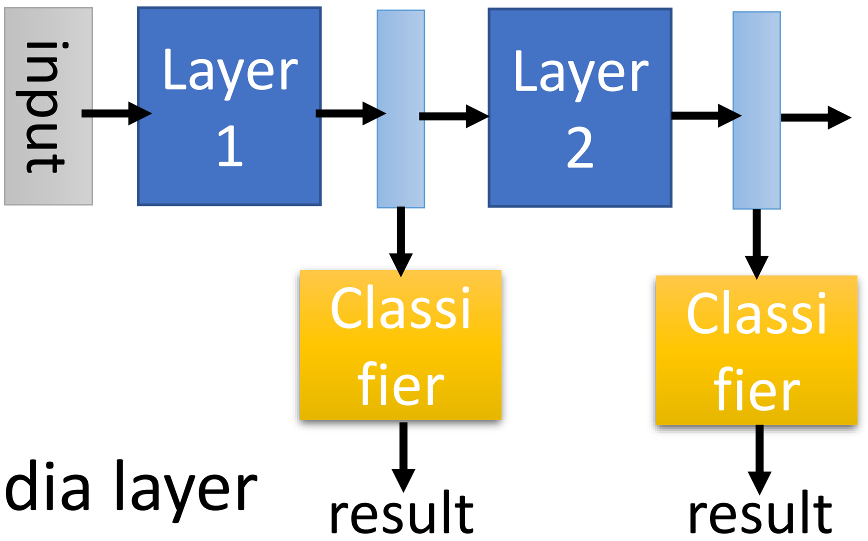
- Can network adjust the computation power it need?

資源充足，那麼就做到最好

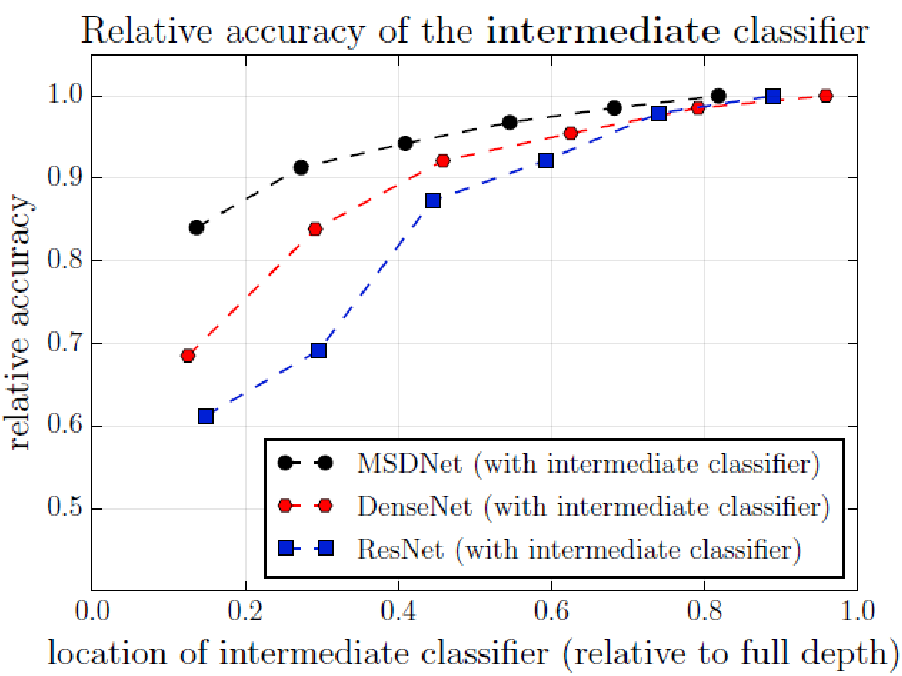
減少運算量，先求有再求好
(但也不要太差)

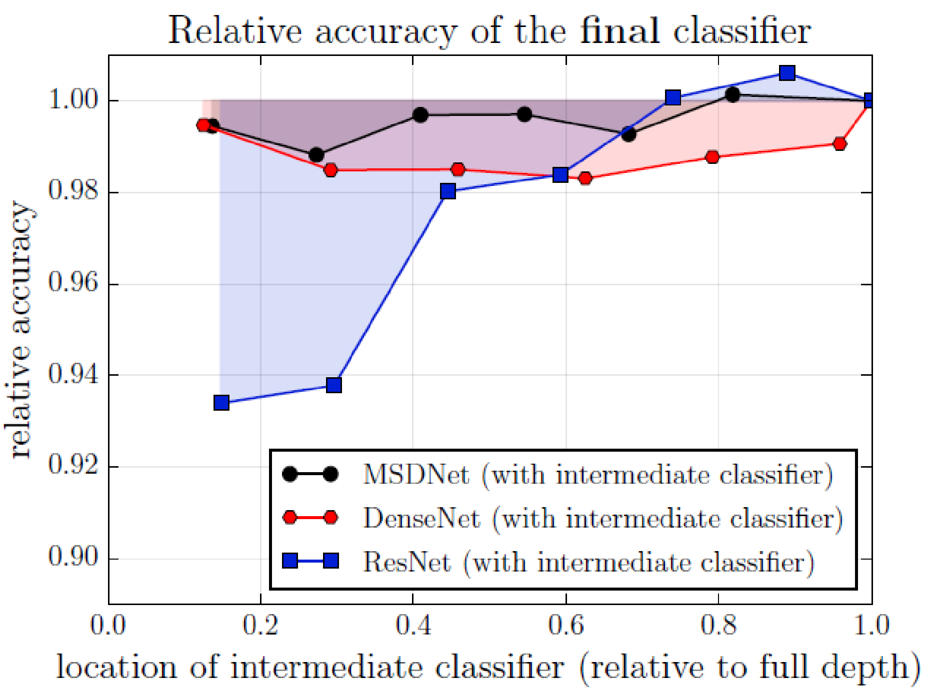


Possible Solutions



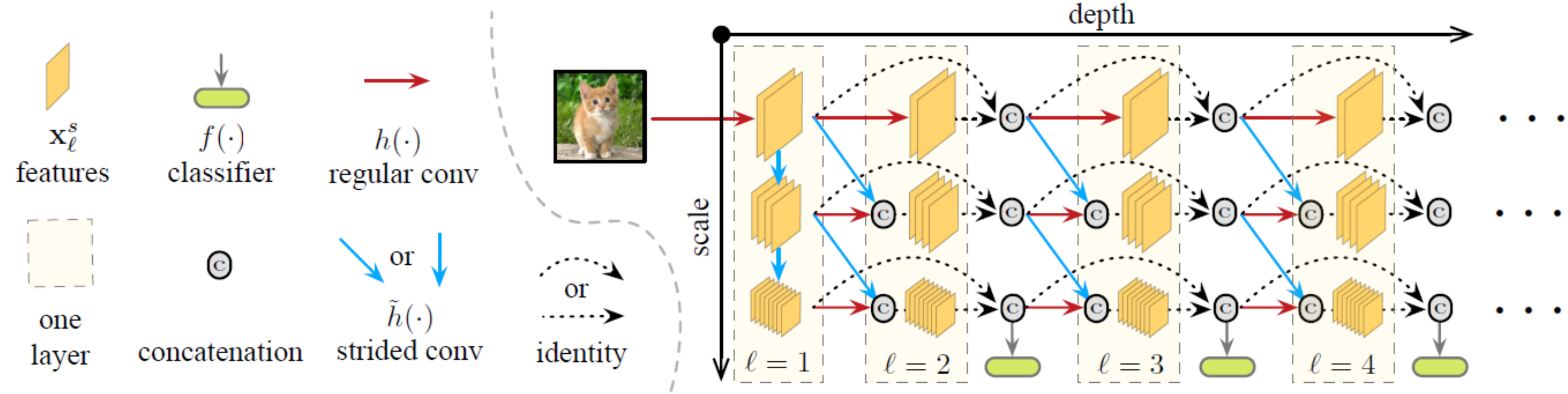
- 1. Train multiple classifiers
- 2. Classifiers at the intermedia layer

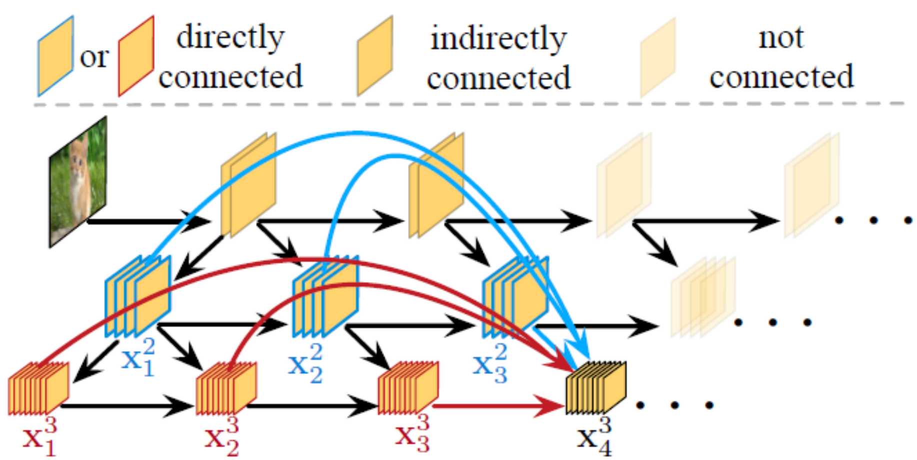




https://arxiv.org/abs/1703.09844

Multi-Scale Dense Networks





x_l^s	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$
$s = 1$	$h_1^1(\mathrm{x}_0^1)$	$h_2^1\left(\mathrm{x}_1^1\right)$	$h_3^1\left(\mathrm{x}_1^1, \mathrm{x}_2^1\right)$	$h_4^1\left(\mathrm{x}_1^1, \mathrm{x}_2^1, \mathrm{x}_3^1\right)$
$s = 2$	$\tilde{h}_1^2\left(\mathrm{x}_1^1\right)$	$\begin{bmatrix} \tilde{h}_2^2\left(\left[\mathrm{x}_1^1\right]\right) \\ h_2^2\left(\left[\mathrm{x}_1^2\right]\right) \end{bmatrix}$	$\begin{bmatrix} \tilde{h}_3^2\left(\left[\mathrm{x}_1^1, \mathrm{x}_2^1\right]\right) \\ h_3^2\left(\left[\mathrm{x}_1^2, \mathrm{x}_2^2\right]\right) \end{bmatrix}$	$\begin{bmatrix} \tilde{h}_4^2\left(\left[\mathrm{x}_1^1, \mathrm{x}_2^1, \mathrm{x}_3^1\right]\right) \\ h_4^2\left(\left[\mathrm{x}_1^2, \mathrm{x}_2^2, \mathrm{x}_3^2\right]\right) \end{bmatrix}$
$s = 3$	$\tilde{h}_1^3\left(\mathrm{x}_1^2\right)$	$\begin{bmatrix} \tilde{h}_2^3\left(\left[\mathrm{x}_1^2\right]\right) \\ h_2^3\left(\left[\mathrm{x}_1^3\right]\right) \end{bmatrix}$	$\begin{bmatrix} \tilde{h}_3^3\left(\left[\mathrm{x}_1^2, \mathrm{x}_2^2\right]\right) \\ h_3^3\left(\left[\mathrm{x}_1^3, \mathrm{x}_2^3\right]\right) \end{bmatrix}$	$\begin{bmatrix} \tilde{h}_4^3\left(\left[\mathrm{x}_1^2, \mathrm{x}_2^2, \mathrm{x}_3^2\right]\right) \\ h_4^3\left(\left[\mathrm{x}_1^3, \mathrm{x}_2^3, \mathrm{x}_3^3\right]\right) \end{bmatrix}$

https://arxiv.org/abs/1703.09844

Concluding Remarks

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