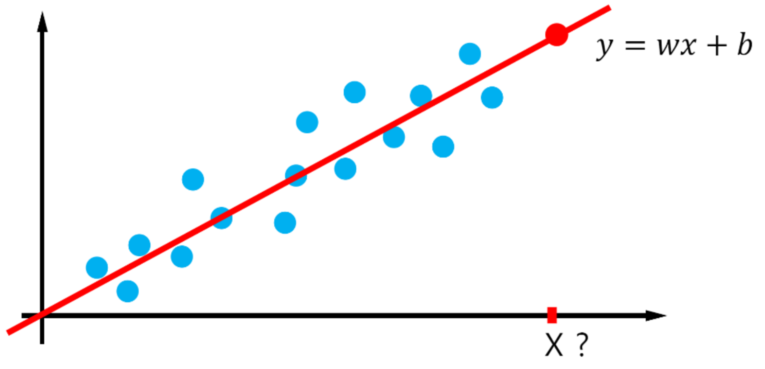
NeuralNet 101

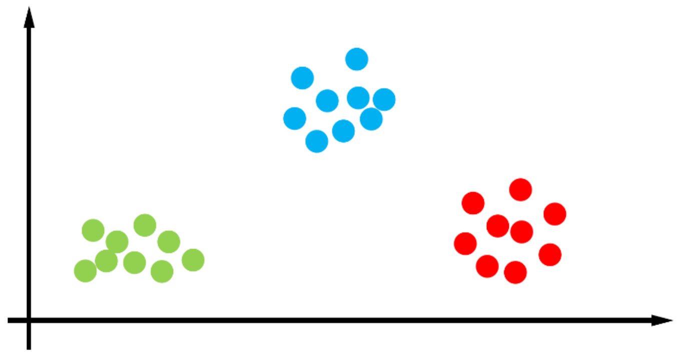
6. CNN

We have a problem.. (again)

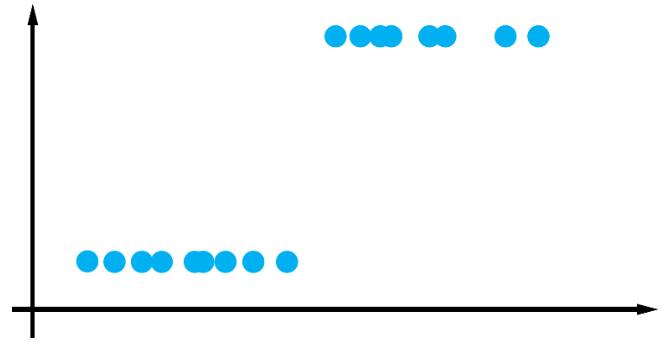
Linear Function



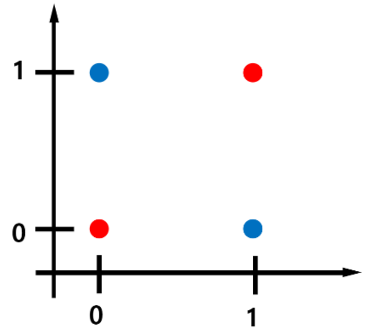
We have a problem..



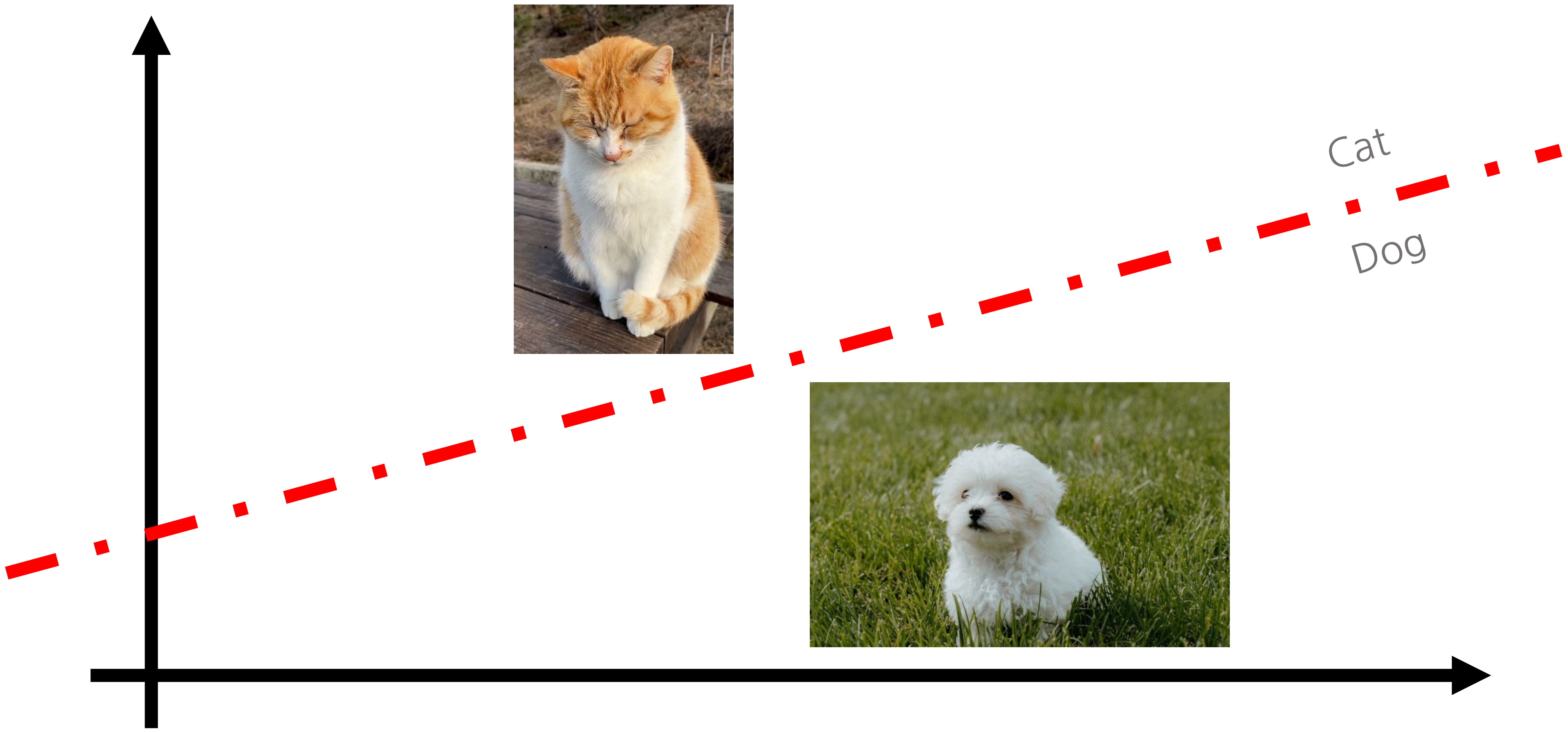
We have a problem..



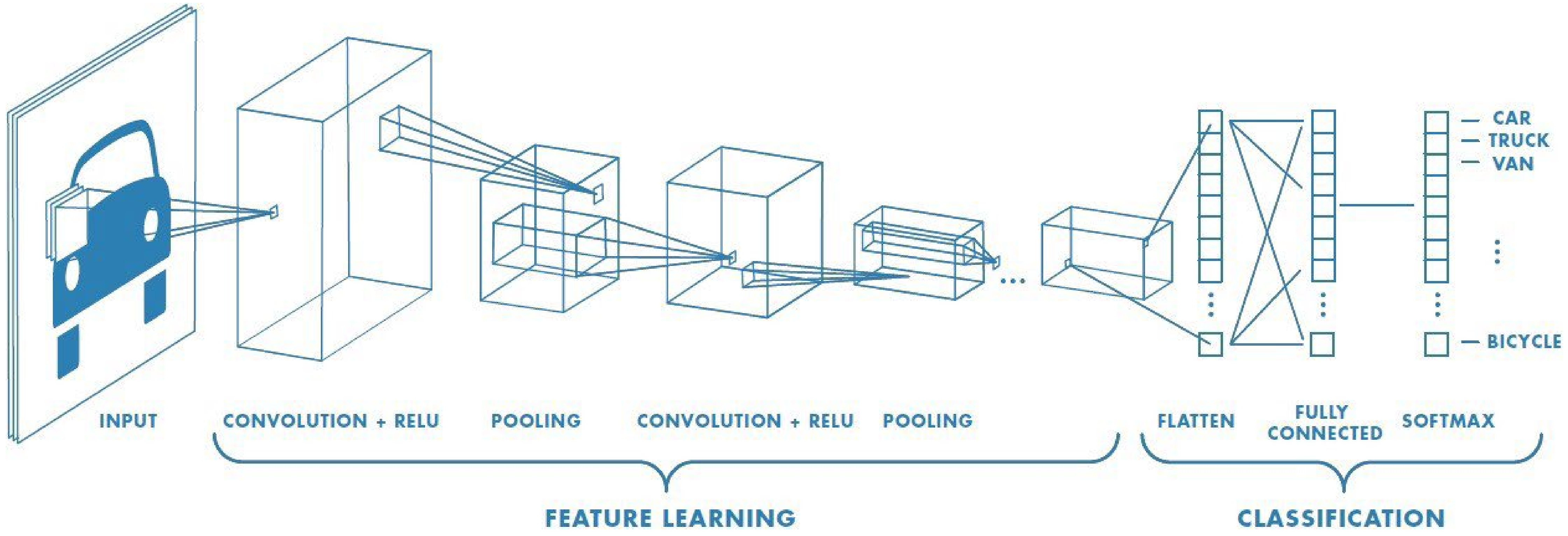
We have a problem..



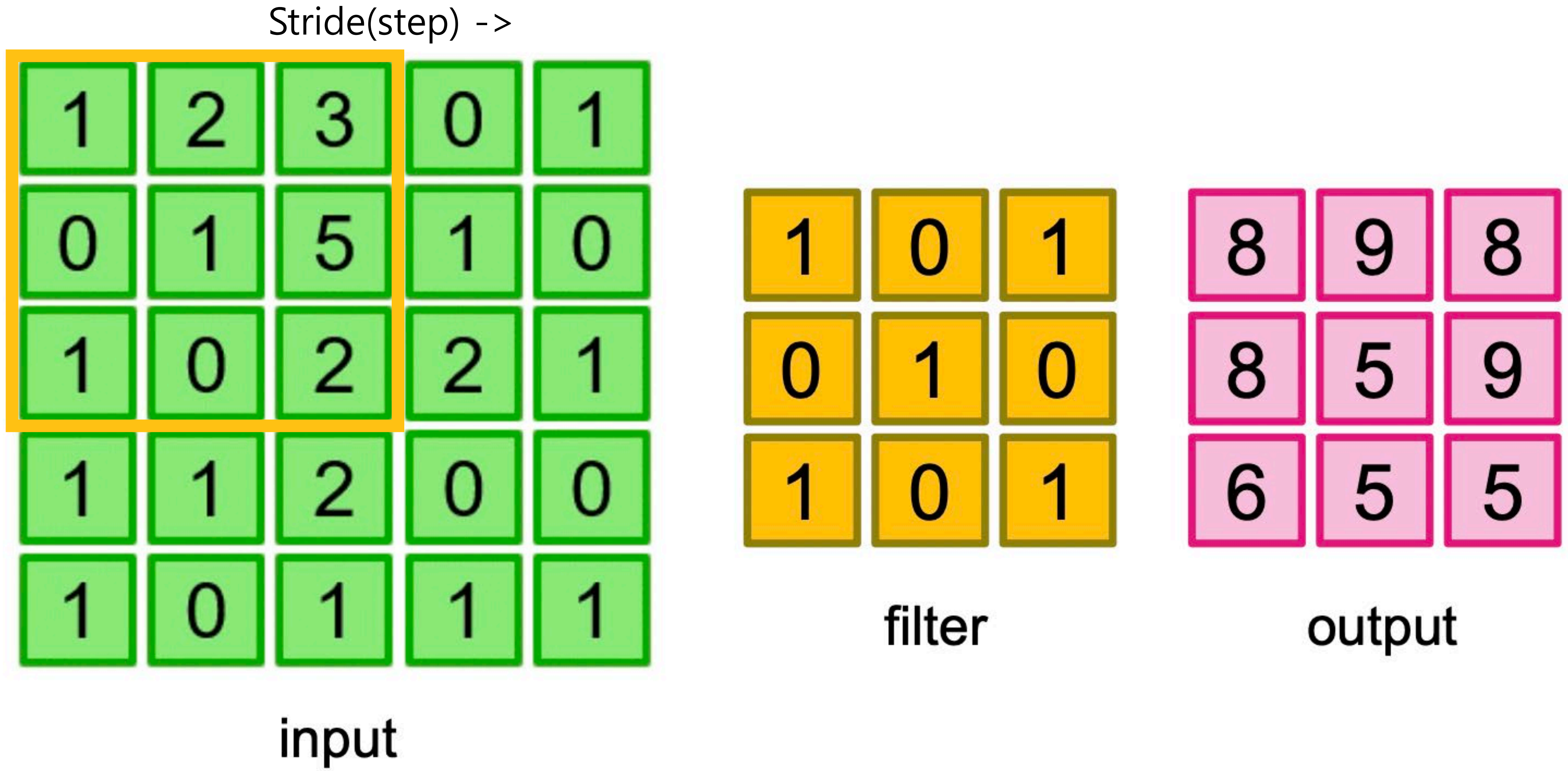
How to train for image data?



CNN (step-by-step)

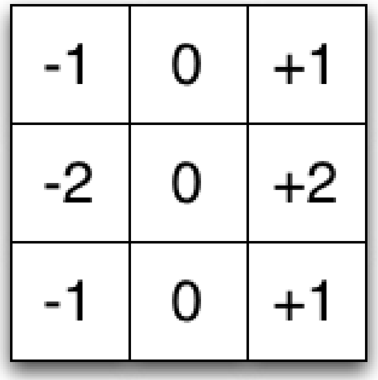


Convolution



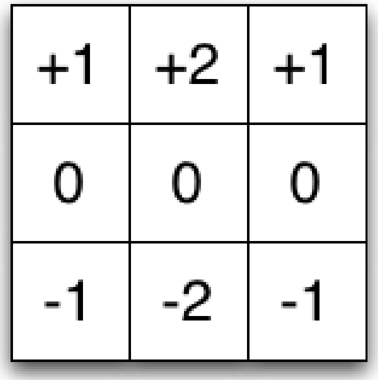
Convolution Filter (Kernel)

ex) Sobel X/Y: Edge Filter



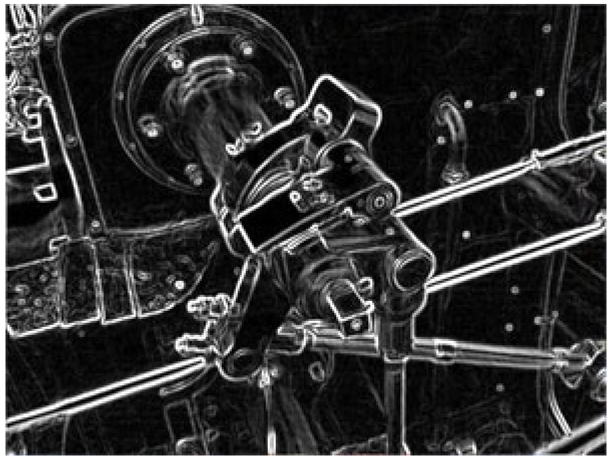
-1	0	+1
-2	0	+2
-1	0	+1

x filter



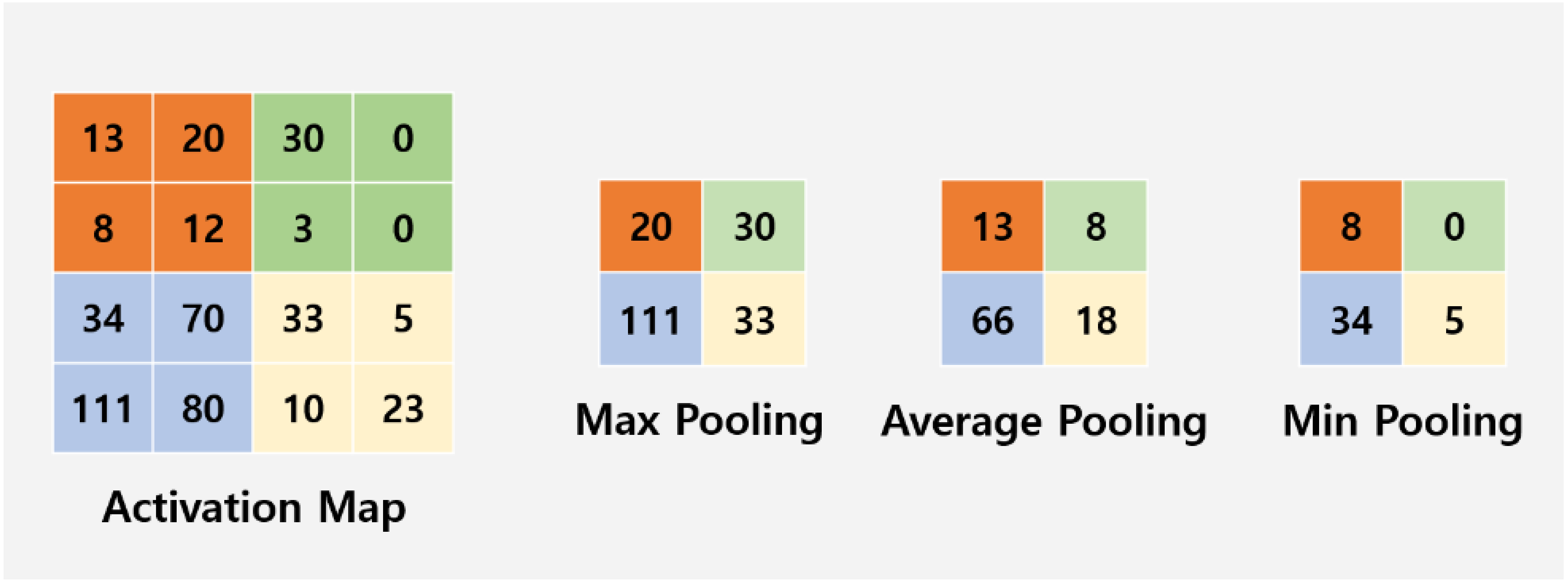
+1	+2	+1
0	0	0
-1	-2	-1

y filter



Appropriate filters automatically trained in CNN, to extract feature

Pooling (sub sampling)



13	20	30	0
8	12	3	0
34	70	33	5
111	80	10	23

Activation Map

20	30
111	33

Max Pooling

13	8
66	18

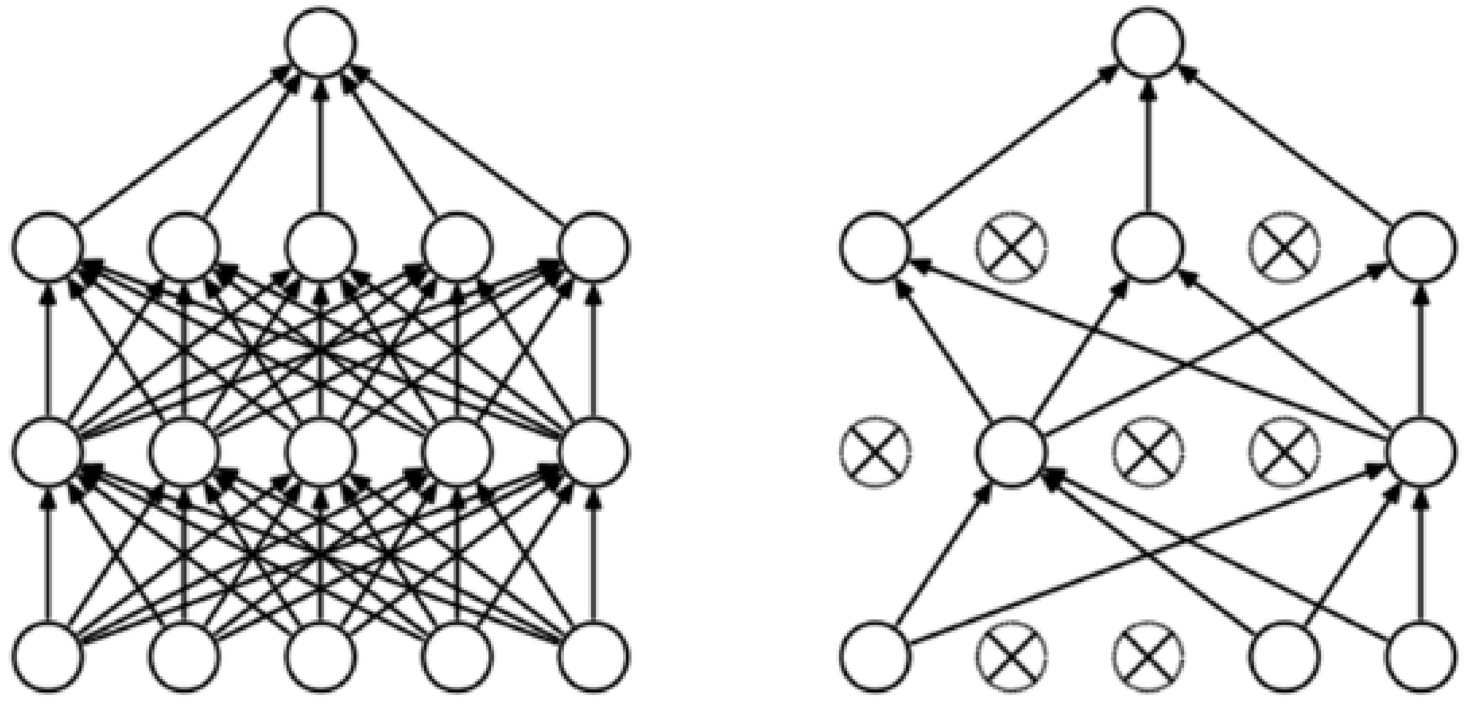
Average Pooling

8	0
34	5

Min Pooling

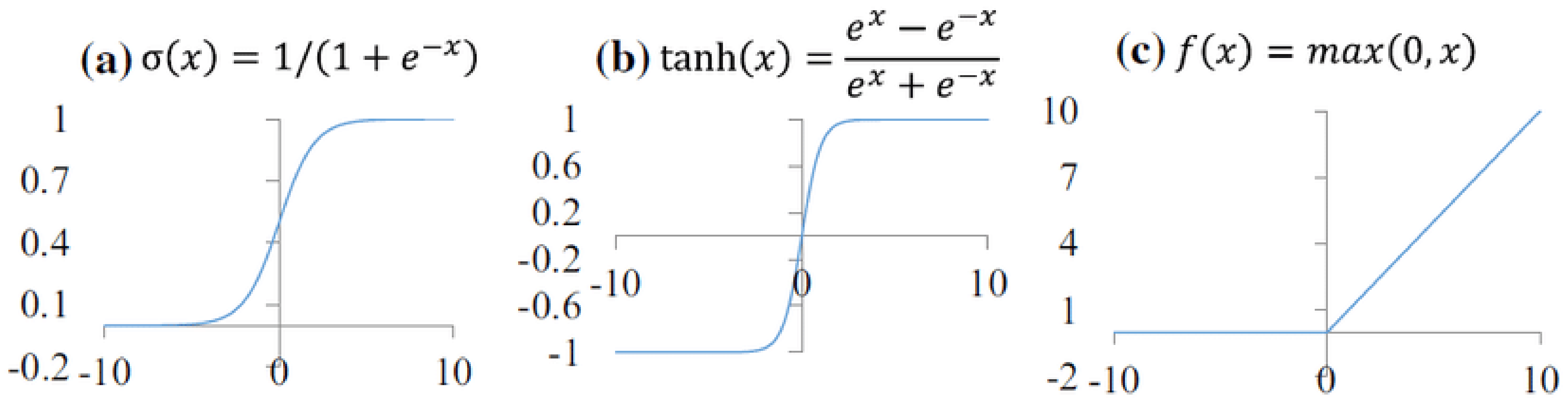
Reduce data size for less calculation and less overfitting

Dropout


(a) Standard Neural Net (b) After applying dropout.

Random dropout prevents overfitting and co-adaptation
-> better feature!

Activation Function

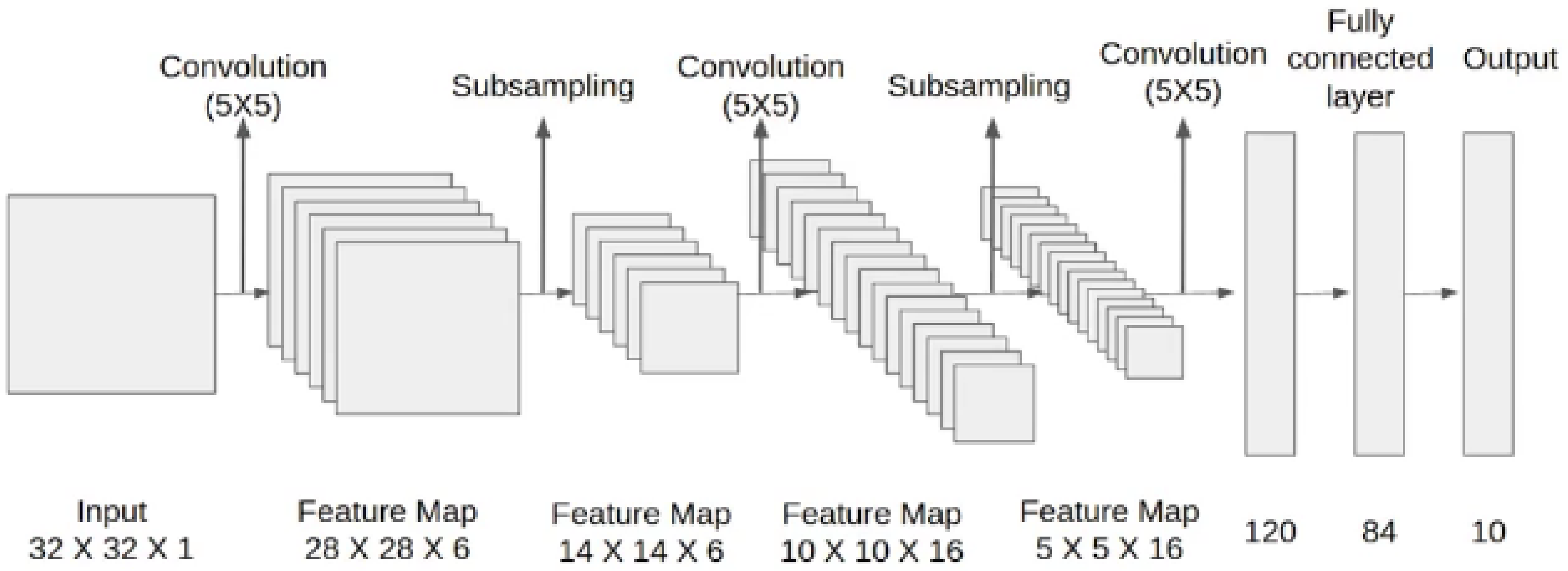


tanh: better slope, back-prop direction not biased
ReLU: faster training, no saturation for extreme values

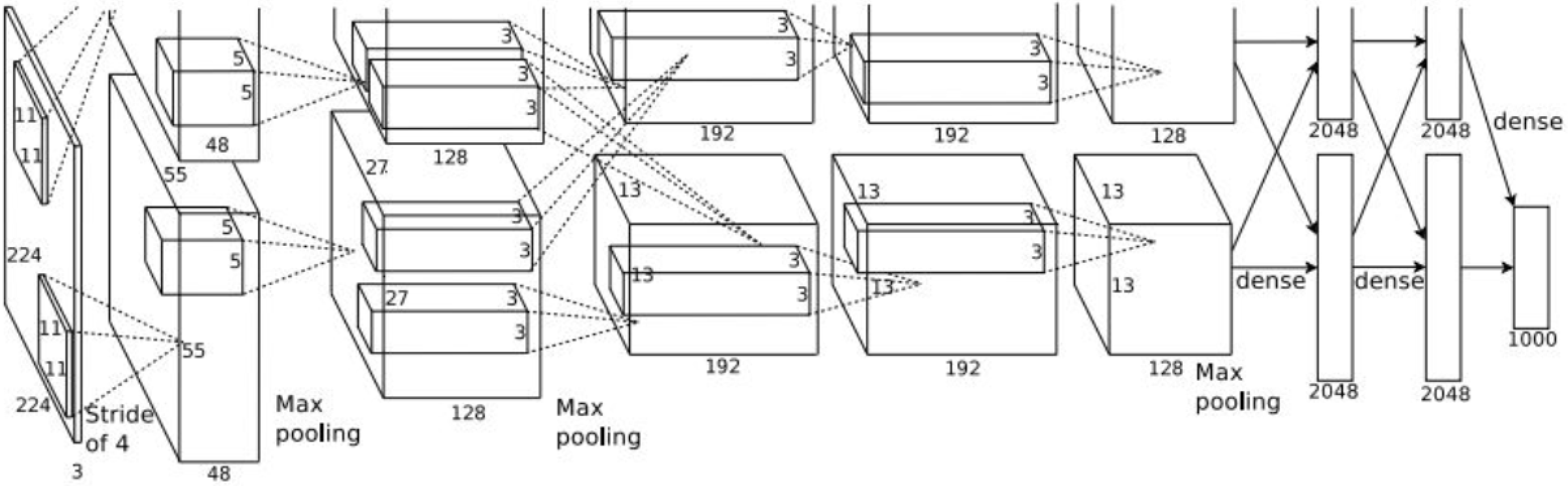
CNN Architecture Overview

LeNet -> AlexNet/ZFNet ->
VGG/GoogLeNet -> ResNet -> …

LeNet-5

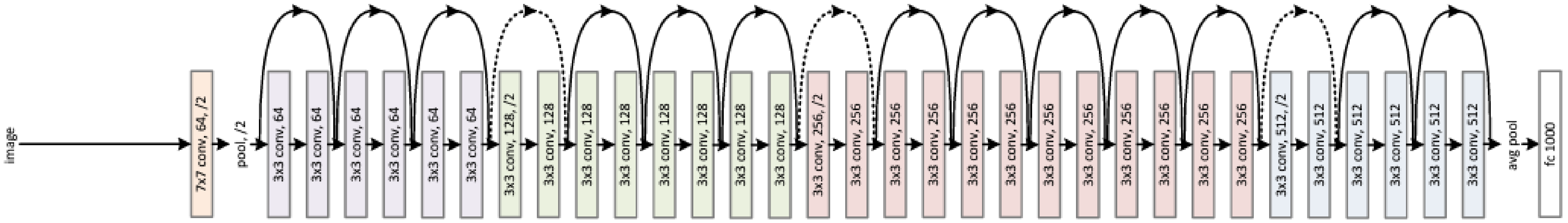

Convolution (5X5)
Subsampling
Convolution (5X5)
Subsampling
Convolution (5X5)
Fully connected layer
Output
Input 32 X 32 X 1
Feature Map 28 X 28 X 6
Feature Map 14 X 14 X 6
Feature Map 10 X 10 X 16
Feature Map 5 X 5 X 16
120
84
10

AlexNet

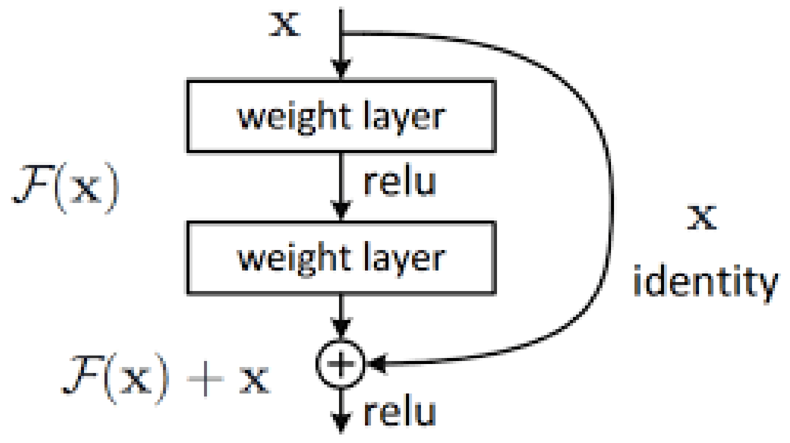


224
224
3
11
11
11
11
Stride of 4
5
5
5
5
55
55
48
48
Max pooling
3
3
3
3
3
3
3
3
27
27
128
128
Max pooling
13
13
13
192
192
192
192
128
128
Max pooling
dense
dense
dense
2048
2048
2048
2048
1000

ResNet



skip connection:
  residual mapping
-> deep layer performance!


x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu