digi**podium**

Machine Learning

Lecture 18

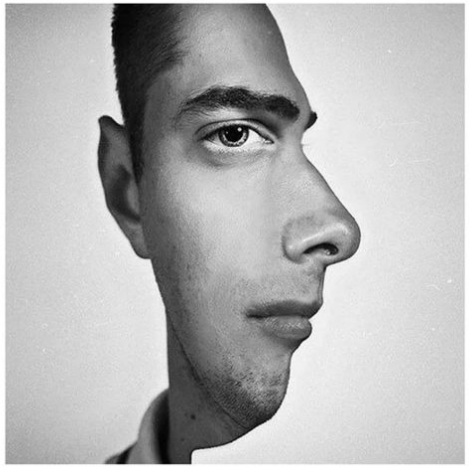
Deep learning

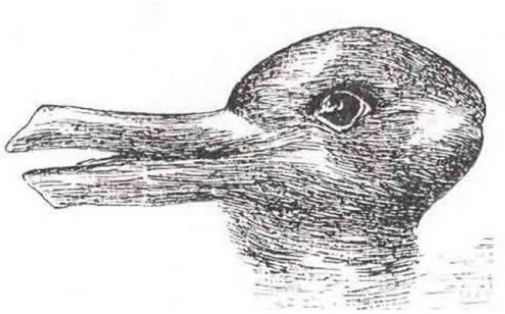
Convolutional Neural Networks

- What are Convolutional Neural Networks?
- Step 1 - Convolution Operation
- Step 1(b) - ReLU Layer
- Step 2 - Pooling
- Step 3 - Flattening
- Step 4 - Full Connection
- Summary

- EXTRA: Softmax & Cross-Entropy

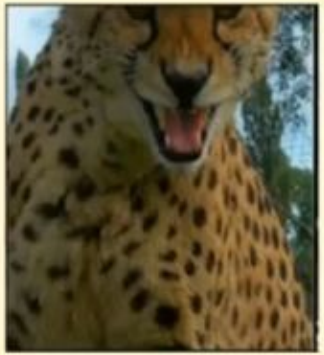
Convolutional Neural Networks





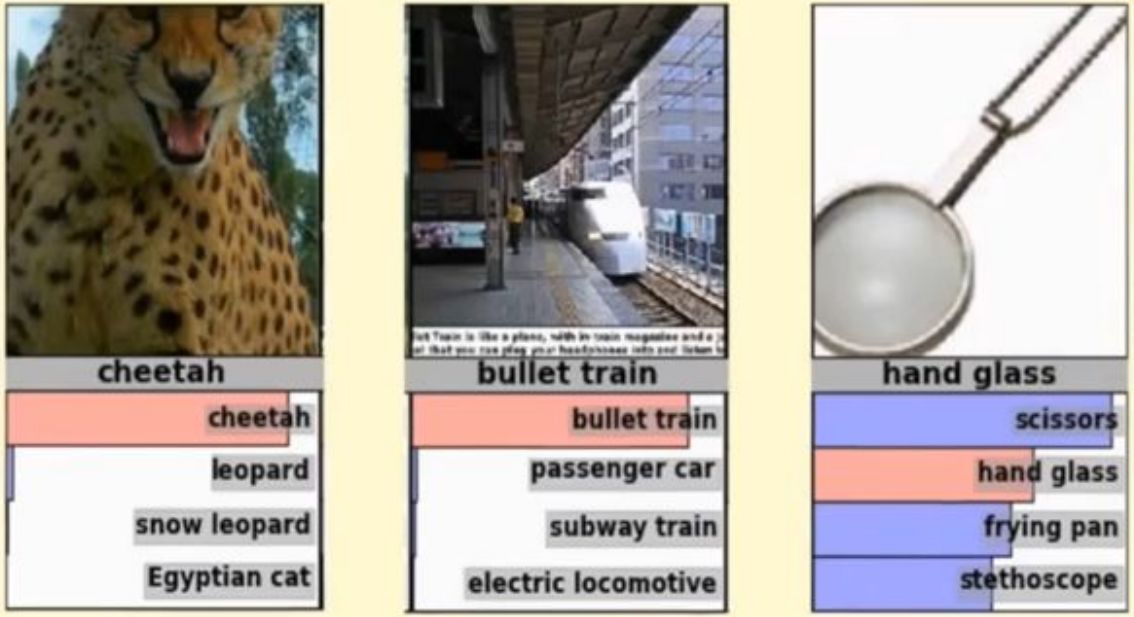
WHAT DO YOU SEE HERE?

Examples from the test set
(with the network's guesses)

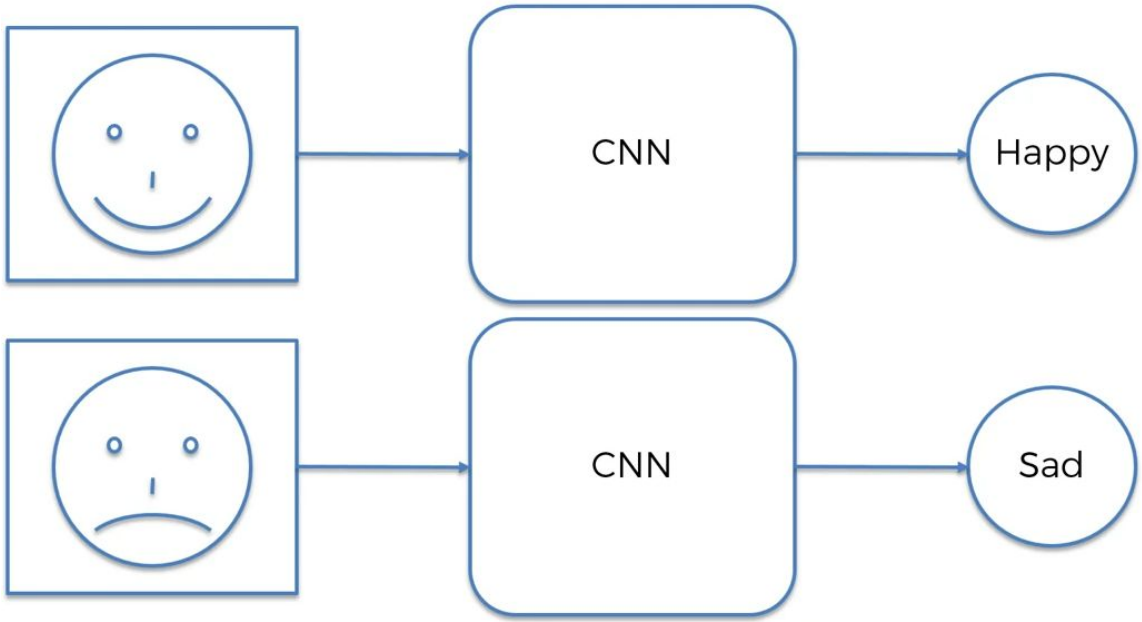


Convolutional Neural Networks

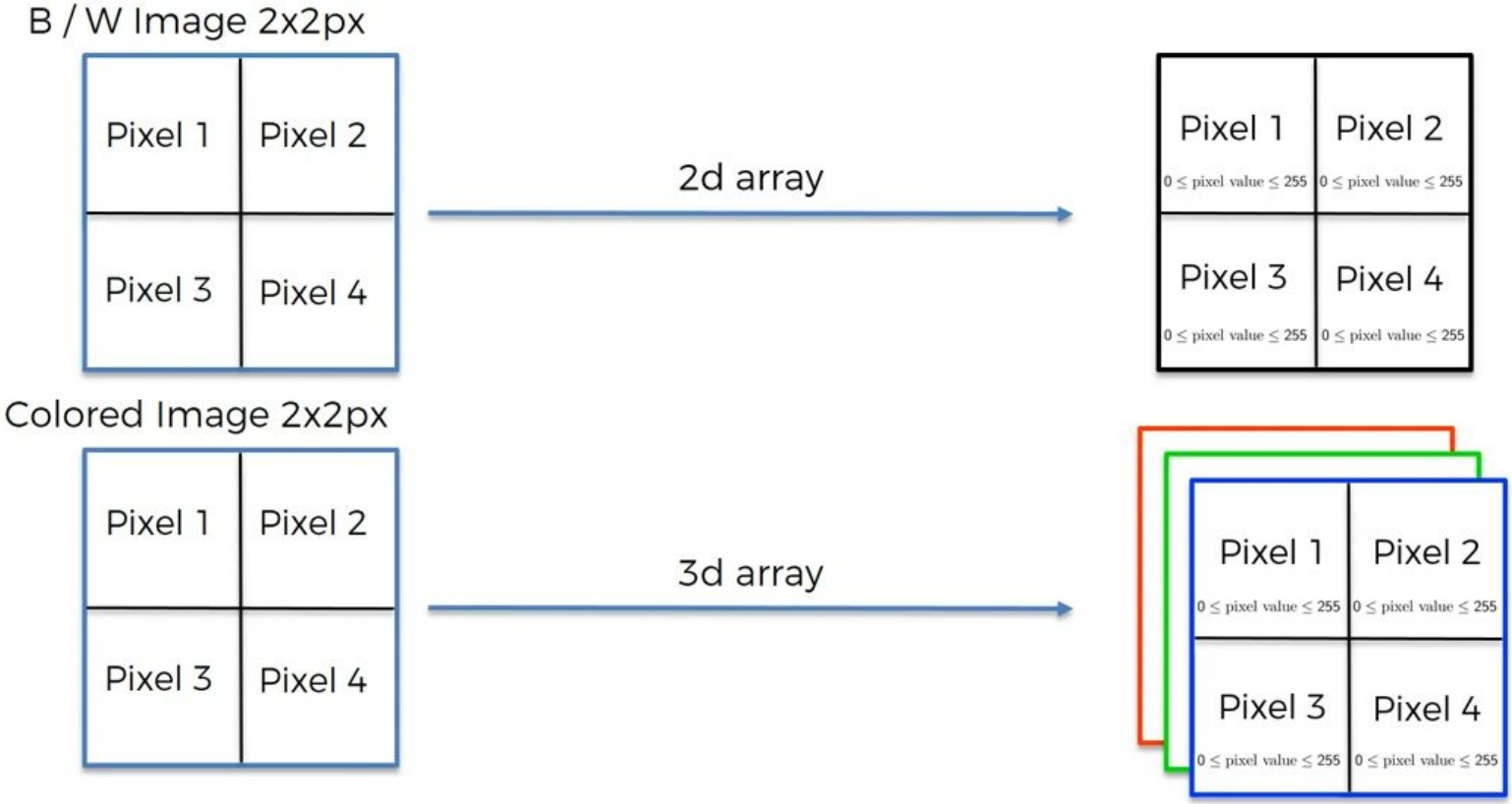
Examples from the test set
(with the network's guesses)



Convolutional Neural Networks

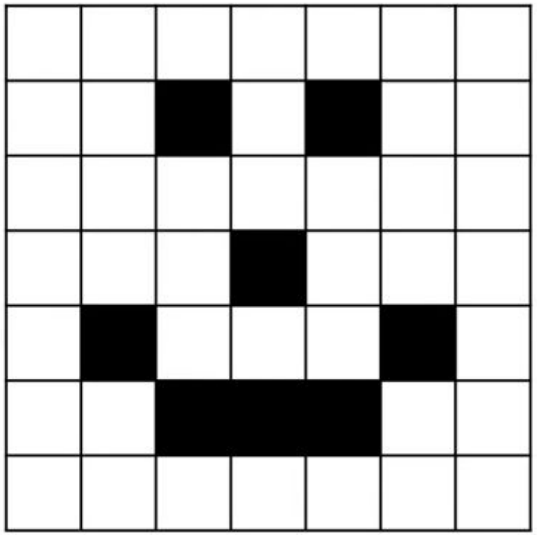


Convolutional Neural Networks

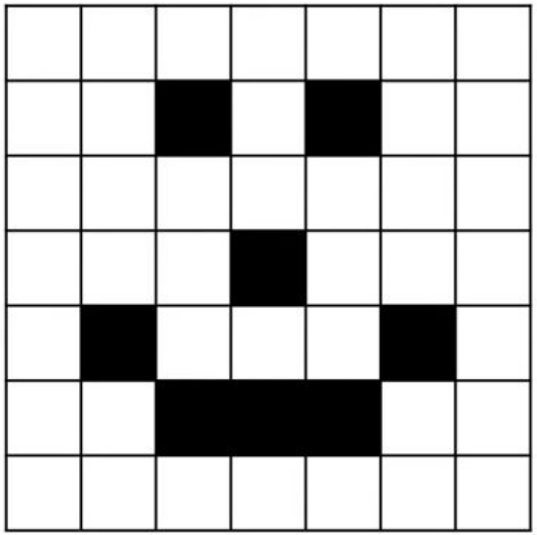


Convolutional Neural Networks

Convolutional Neural Networks



Convolutional Neural Networks



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Convolutional Neural Networks

STEP 1: Convolution

STEP 2: Max Pooling

STEP 3: Flattening

STEP 4: Full Connection

Step 1 – Convolution

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

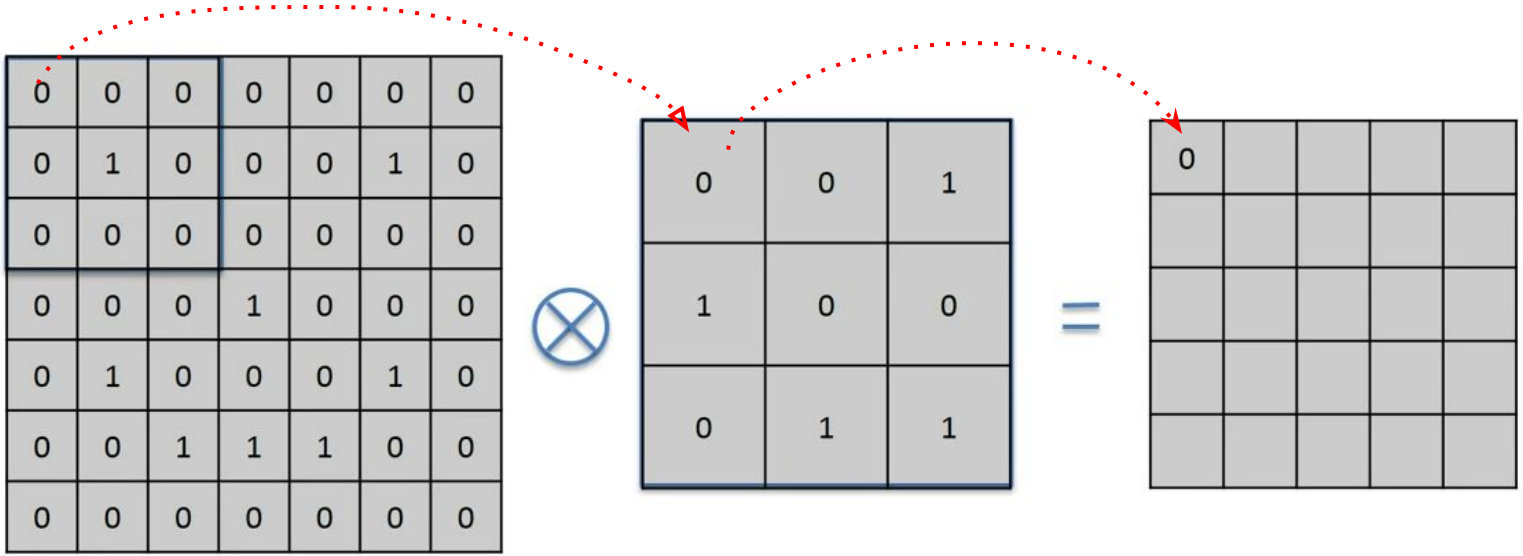
Input Image

0	0	1
1	0	0
0	1	1

Feature
Detector

Step 1 – Convolution

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$



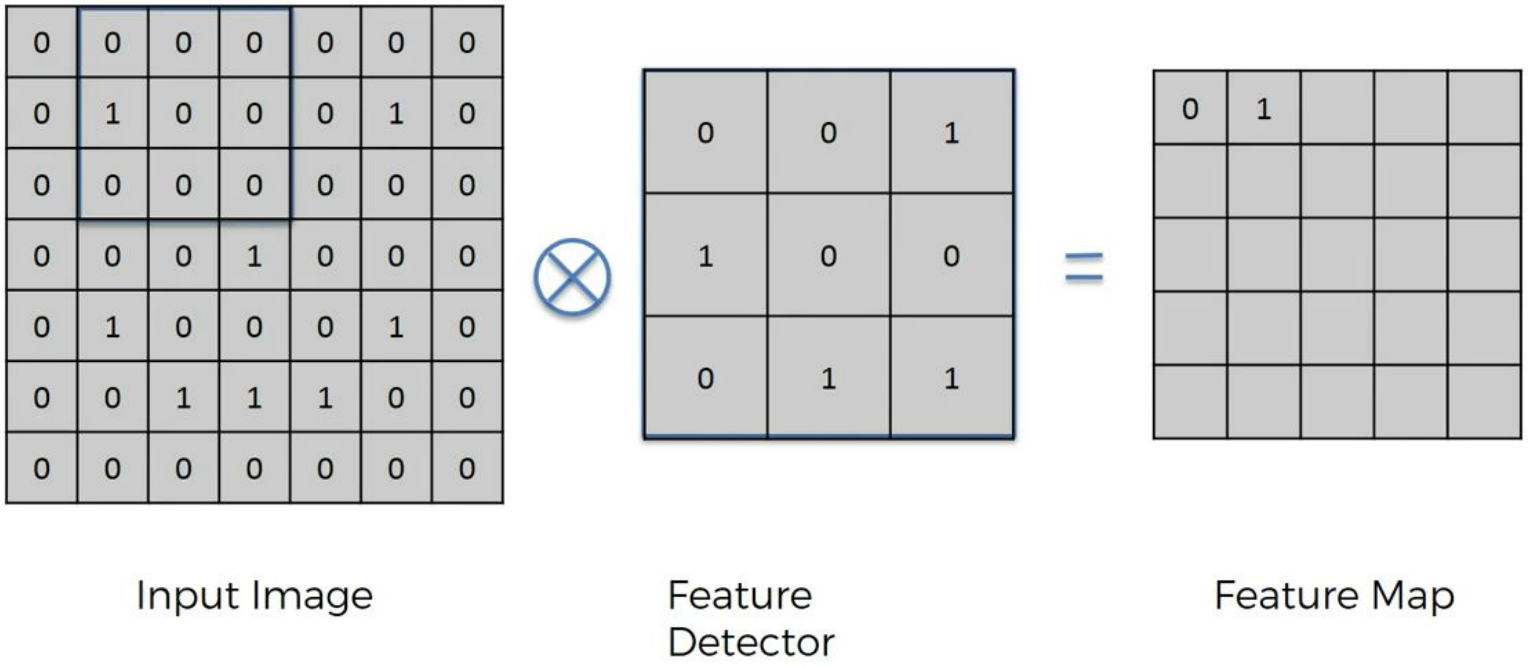
Input Image

Feature Detector

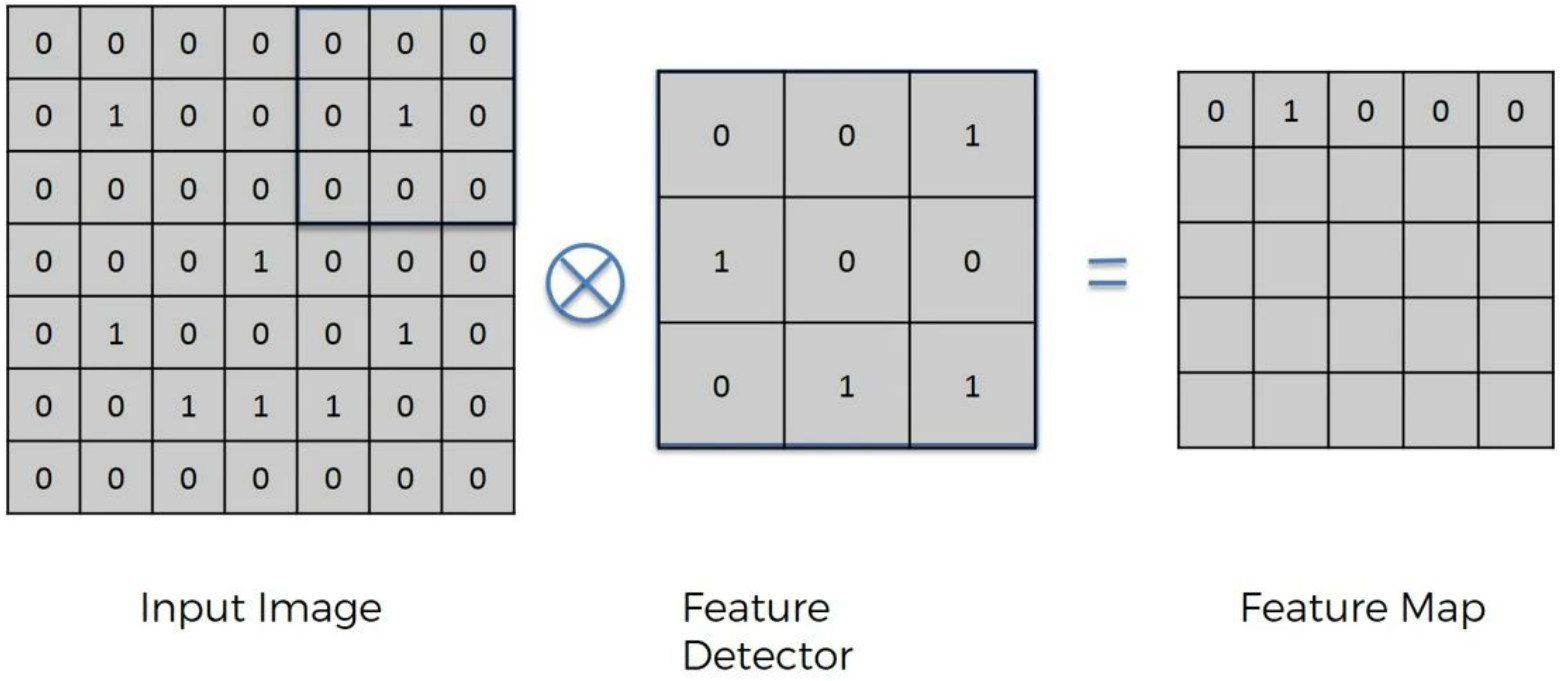
Feature Map

Step 1 – Convolution

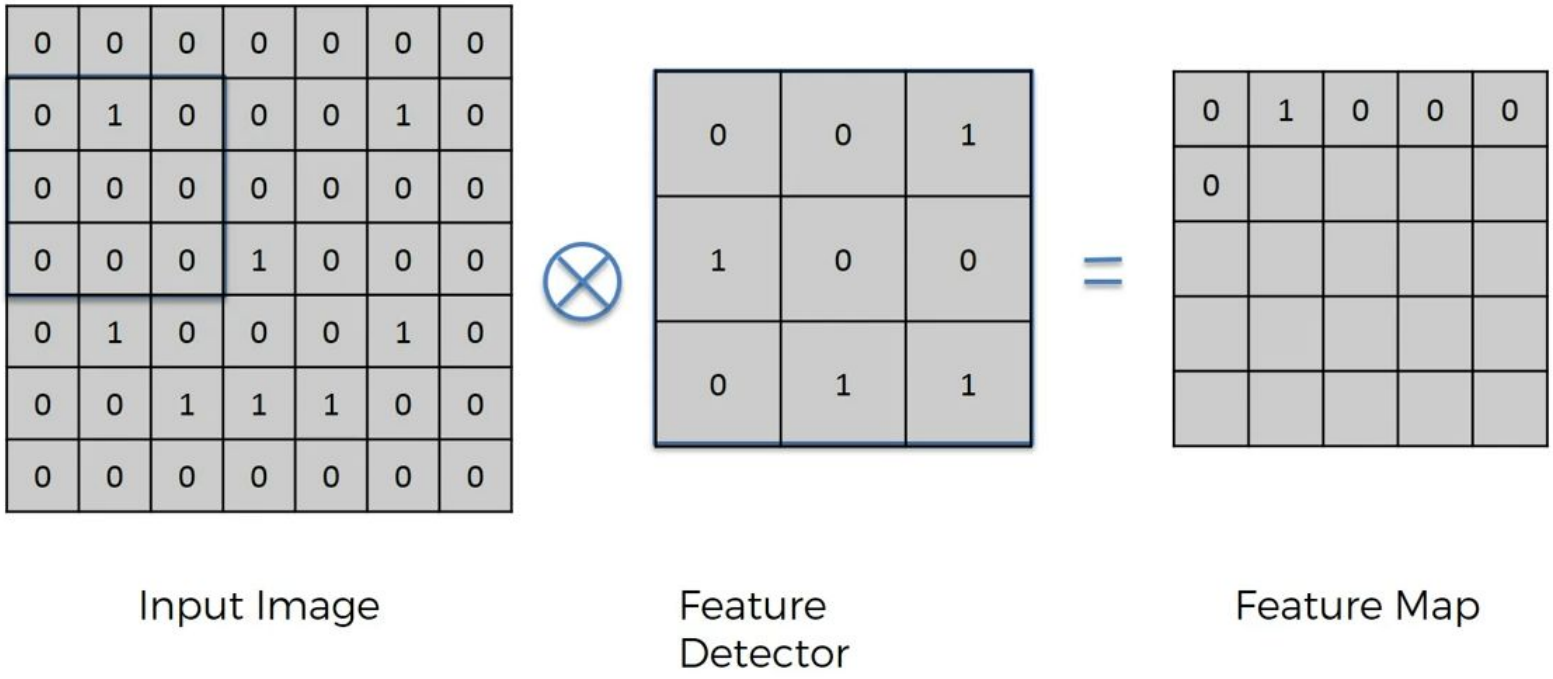
Step 1 – Convolution



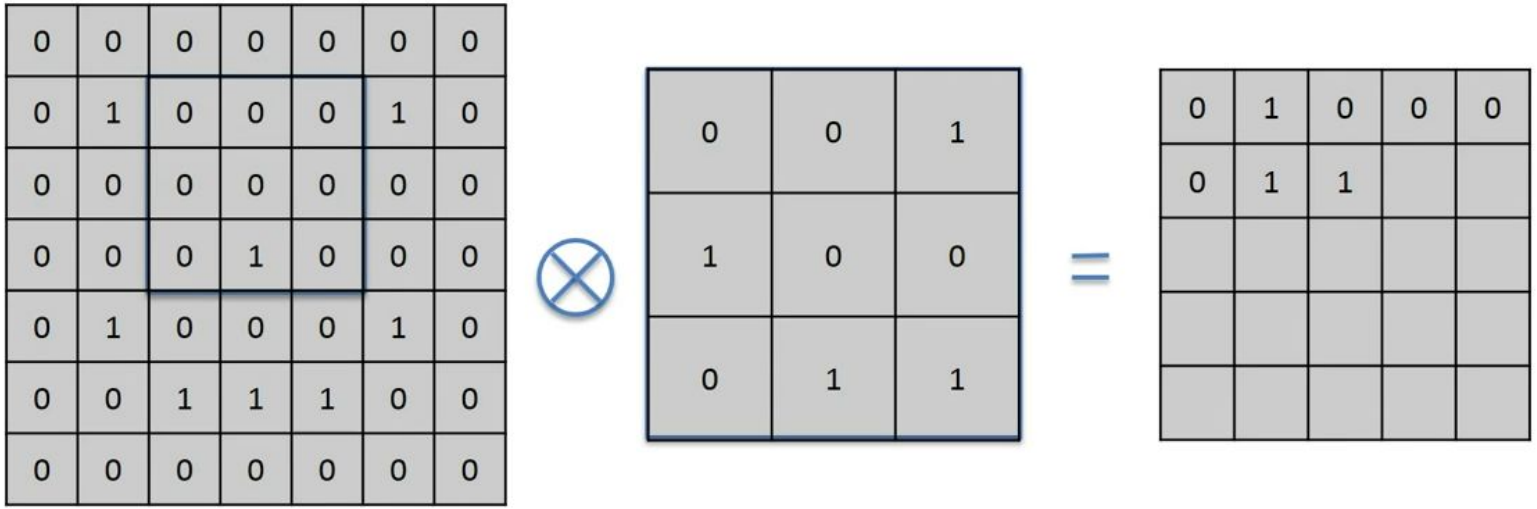
Step 1 – Convolution



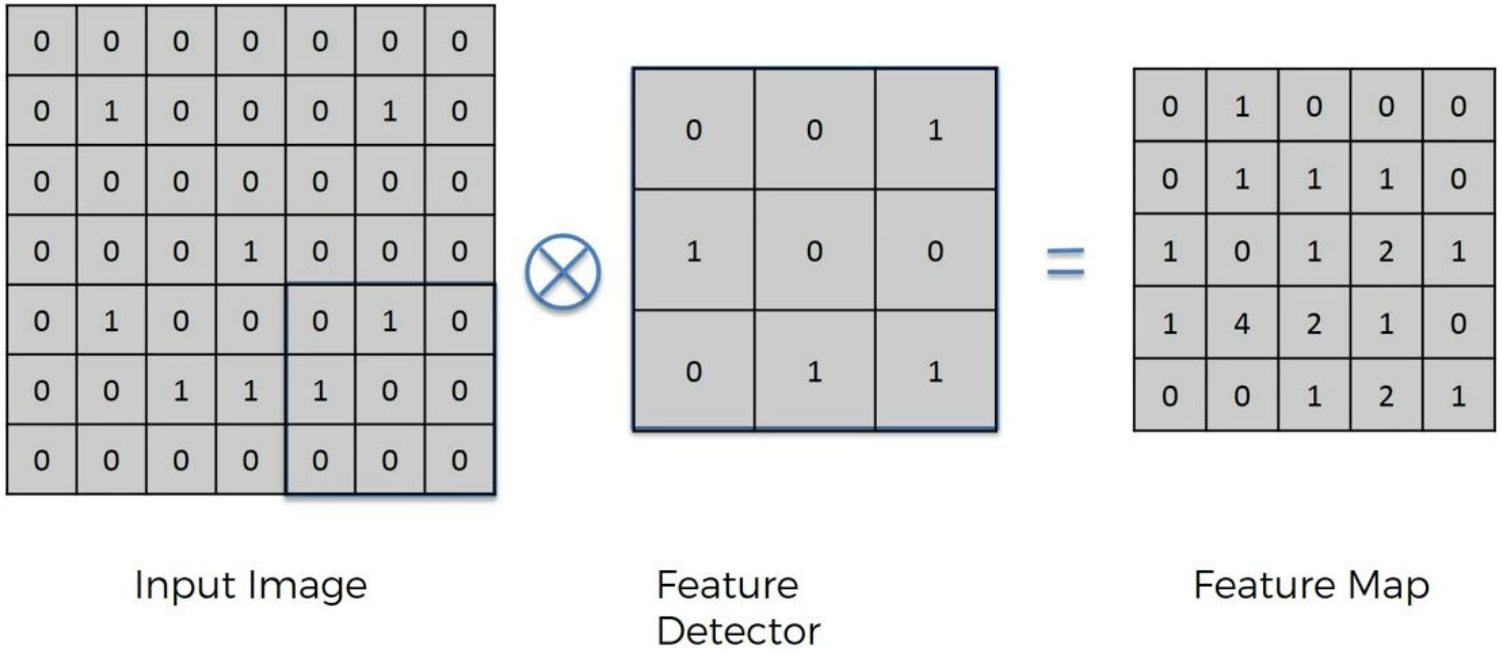
Step 1 – Convolution



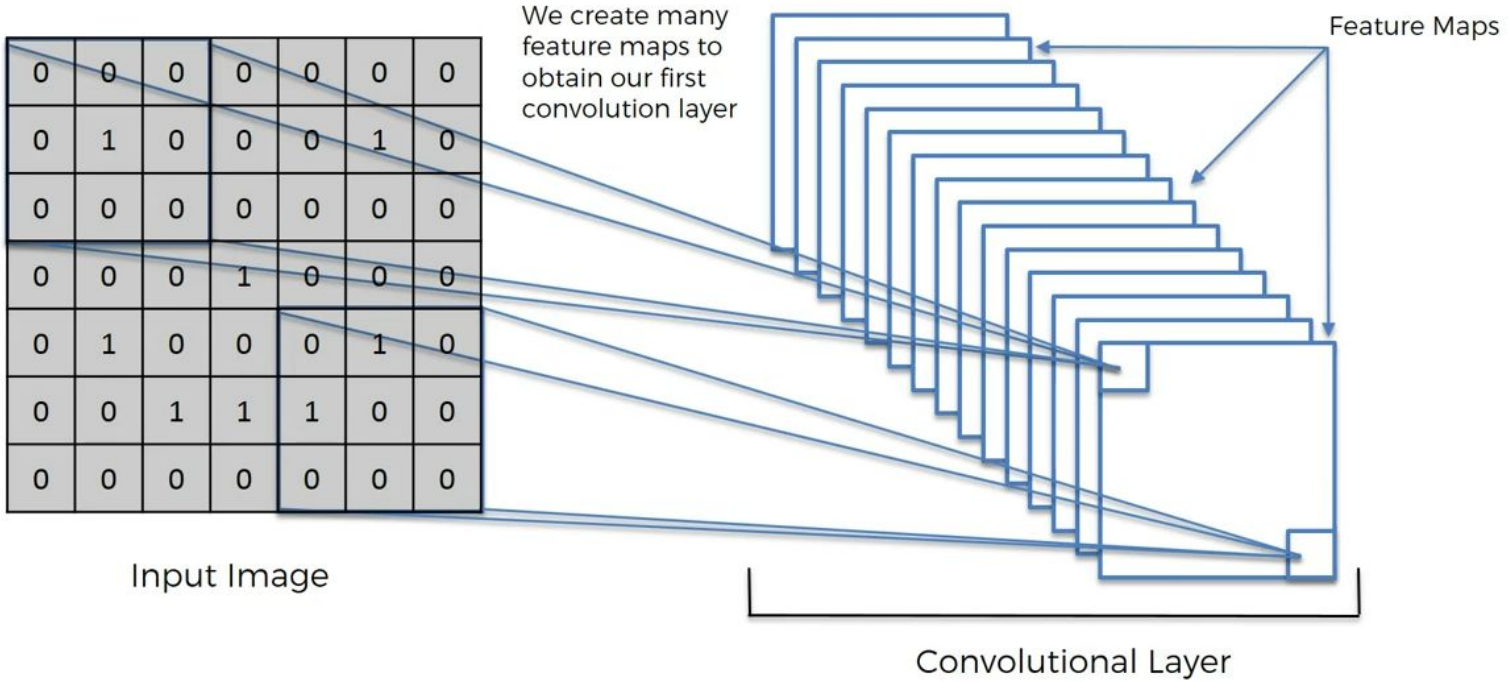
Step 1 – Convolution

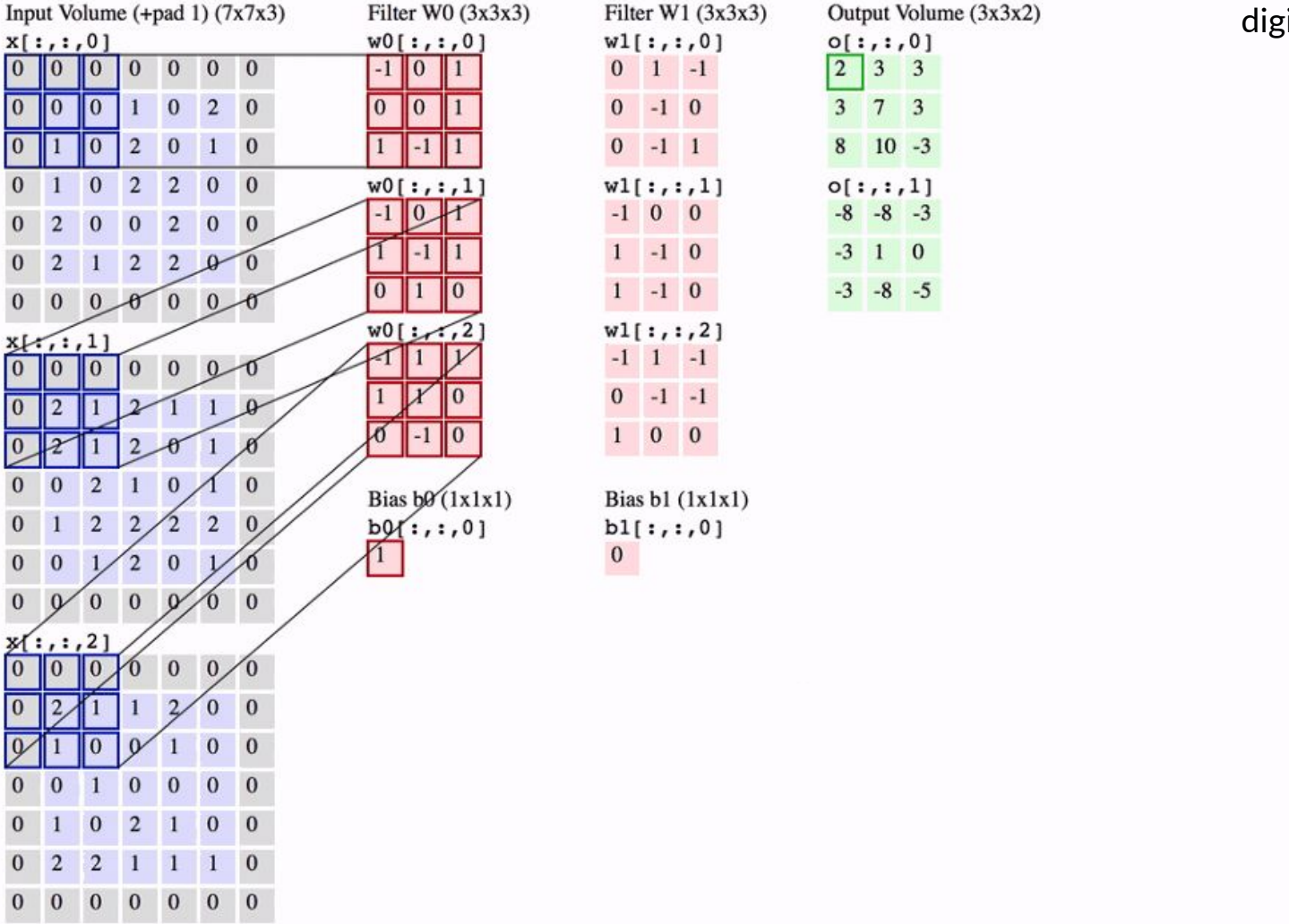


Step 1 – Convolution



Step 1 – Convolution




Input Volume (+pad 1) (7x7x3)
x[:,:,0]
0 0 0 0 0 0 0
0 0 0 1 0 2 0
0 1 0 2 0 1 0
0 1 0 2 2 0 0
0 2 0 0 2 0 0
0 2 1 2 2 0 0
0 0 0 0 0 0 0
x[:,:,1]
0 0 0 0 0 0 0
0 2 1 2 1 1 0
0 2 1 2 0 1 0
0 0 2 1 0 1 0
0 1 2 2 2 2 0
0 0 1 2 0 1 0
0 0 0 0 0 0 0
x[:,:,2]
0 0 0 0 0 0 0
0 2 1 1 2 0 0
0 1 0 0 1 0 0
0 0 1 0 0 0 0
0 1 0 2 1 0 0
0 2 2 1 1 1 0
0 0 0 0 0 0 0
Filter W0 (3x3x3)
w0[:,:,0]
-1 0 1
0 0 1
1 -1 1
w0[:,:,1]
-1 0 1
1 -1 1
0 1 0
w0[:,:,2]
-1 1 1
1 1 0
0 -1 0
Bias b0 (1x1x1)
b0[:,:,0]
1
Filter W1 (3x3x3)
w1[:,:,0]
0 1 -1
0 -1 0
0 -1 1
w1[:,:,1]
-1 0 0
1 -1 0
1 -1 0
w1[:,:,2]
-1 1 -1
0 -1 -1
1 0 0
Bias b1 (1x1x1)
b1[:,:,0]
0
Output Volume (3x3x2)
o[:,:,0]
2 3 3
3 7 3
8 10 -3
o[:,:,1]
-8 -8 -3
-3 1 0
-3 -8 -5

Step 1 – Convolution

Sharpen:

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

Blur:

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Edge Enhance:

	0	0	0	
	-1	1	0	
	0	0	0	

Edge Detect:

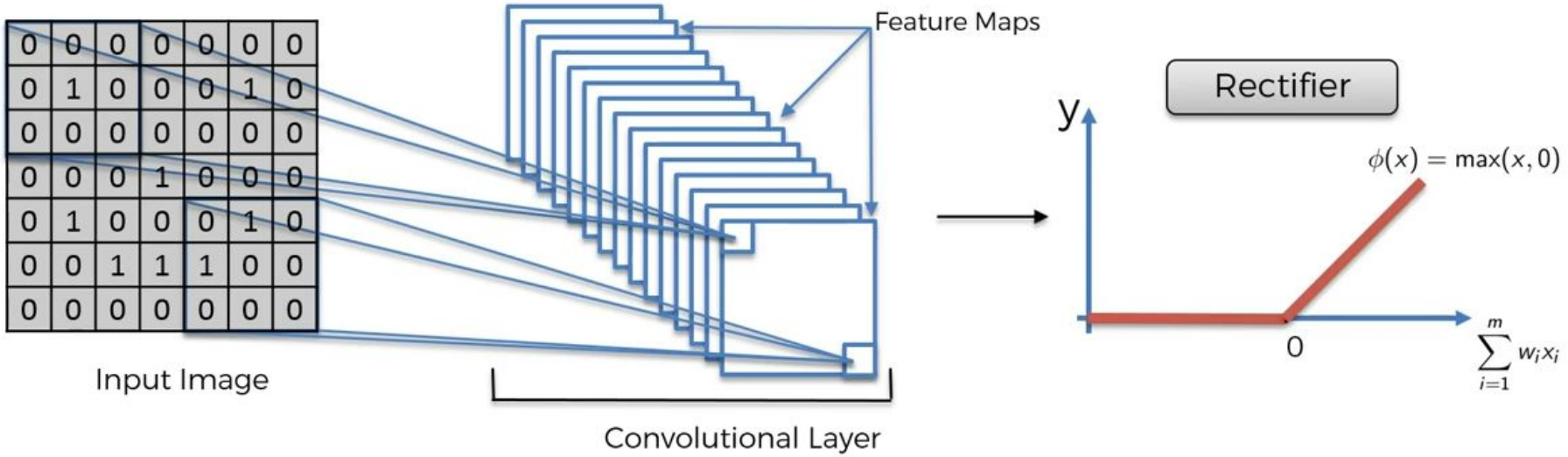
	0	1	0	
	1	-4	1	
	0	1	0	

Step 1 – Convolution

*

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

Step 1(B) – ReLU Layer



Step 1 (B) – ReLU Layer

Normal image

Convolution filter

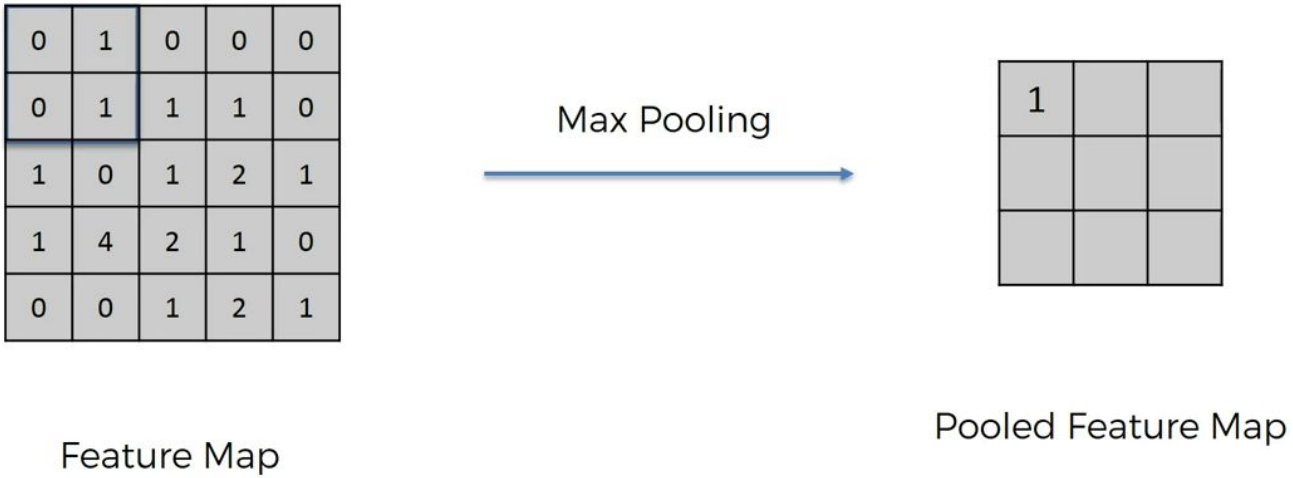
Relu function

Step 2 – Max Pooling

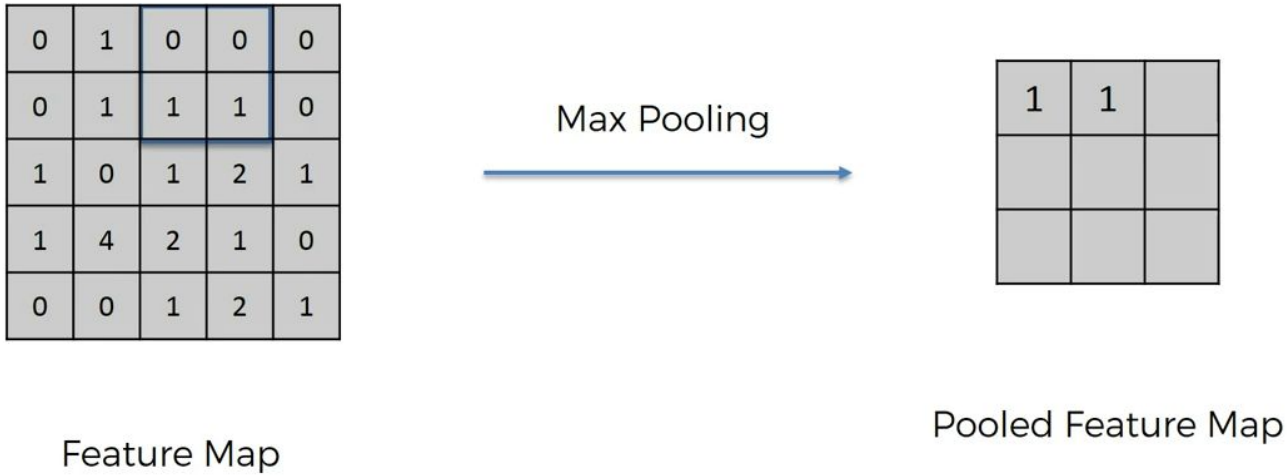
Step 2 – Max Pooling

Image Source: Wikipedia

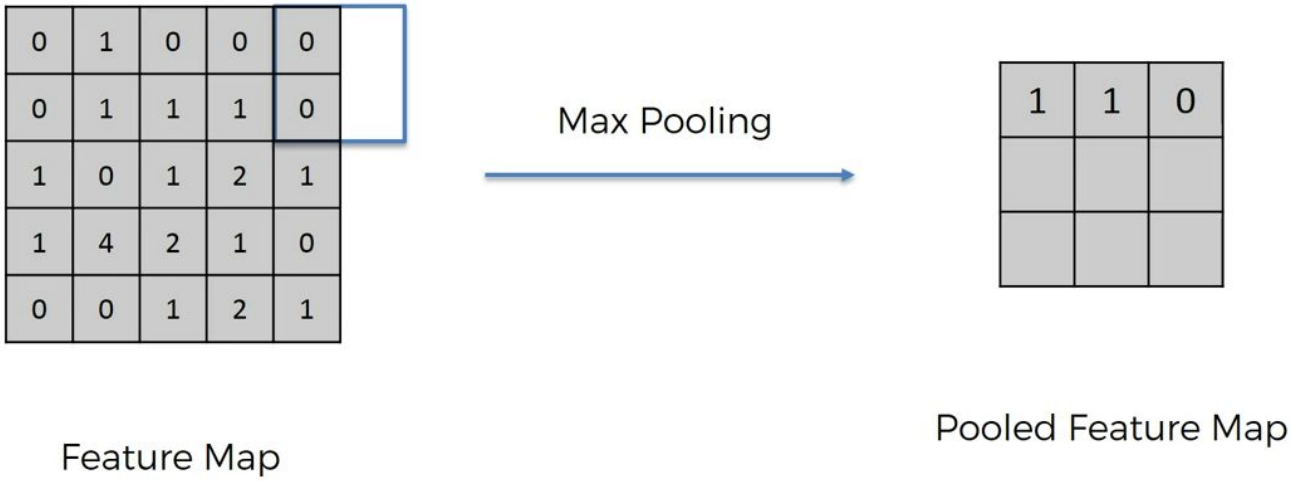
Step 2 – Max Pooling



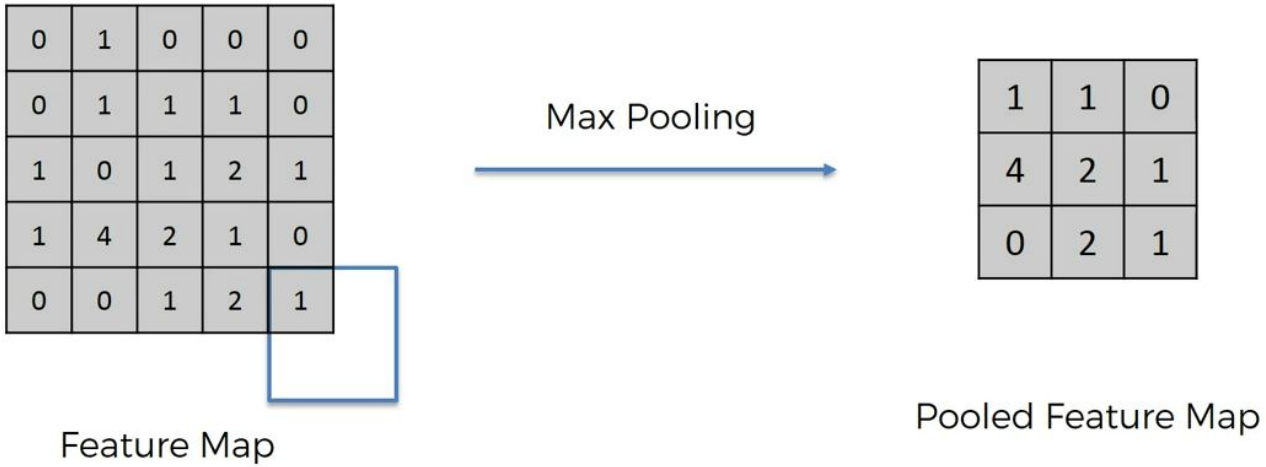
Step 2 – Max Pooling



Step 2 – Max Pooling



Step 2 – Max Pooling



Step 2 – Max Pooling

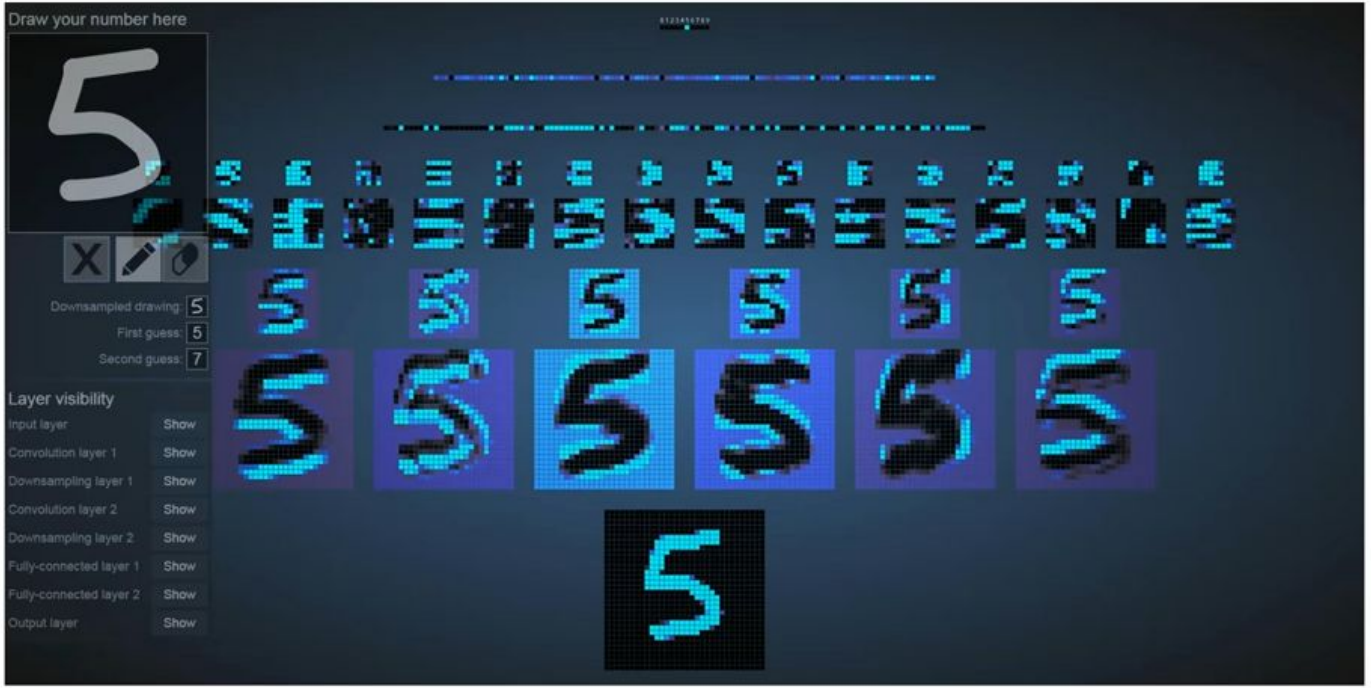
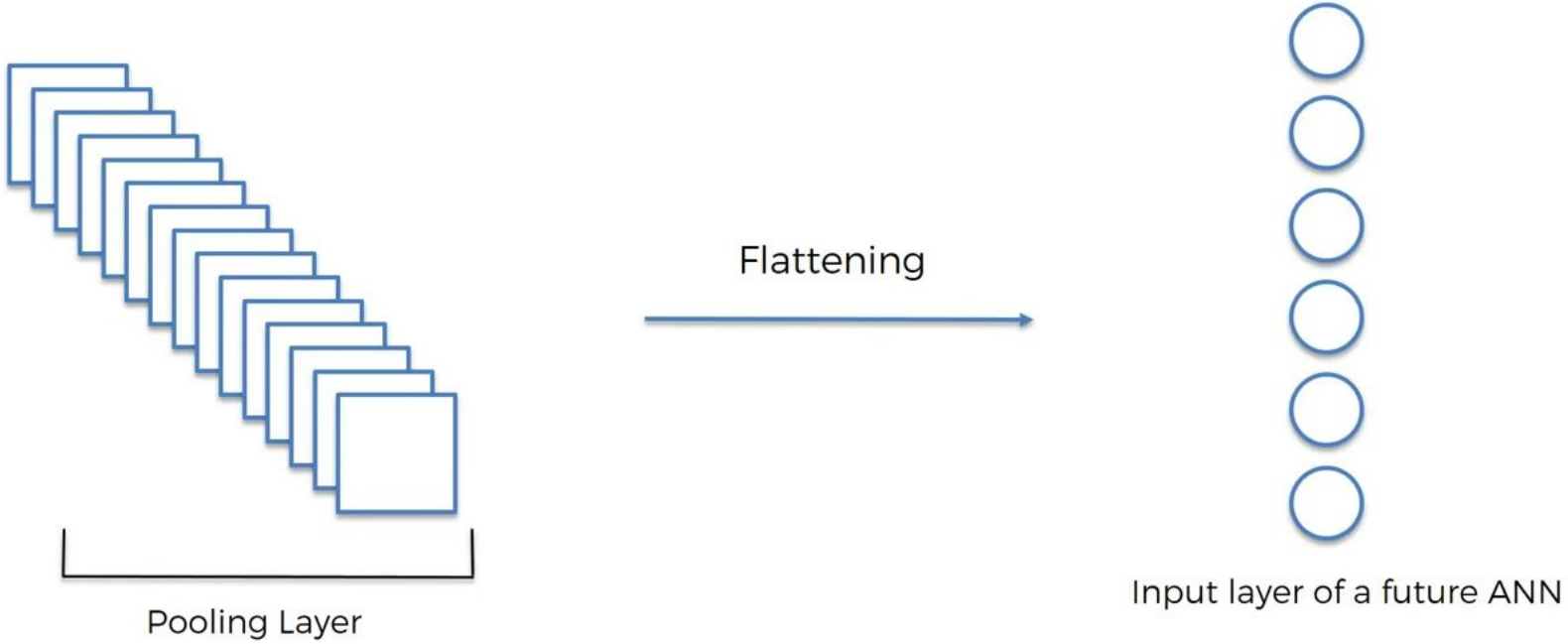


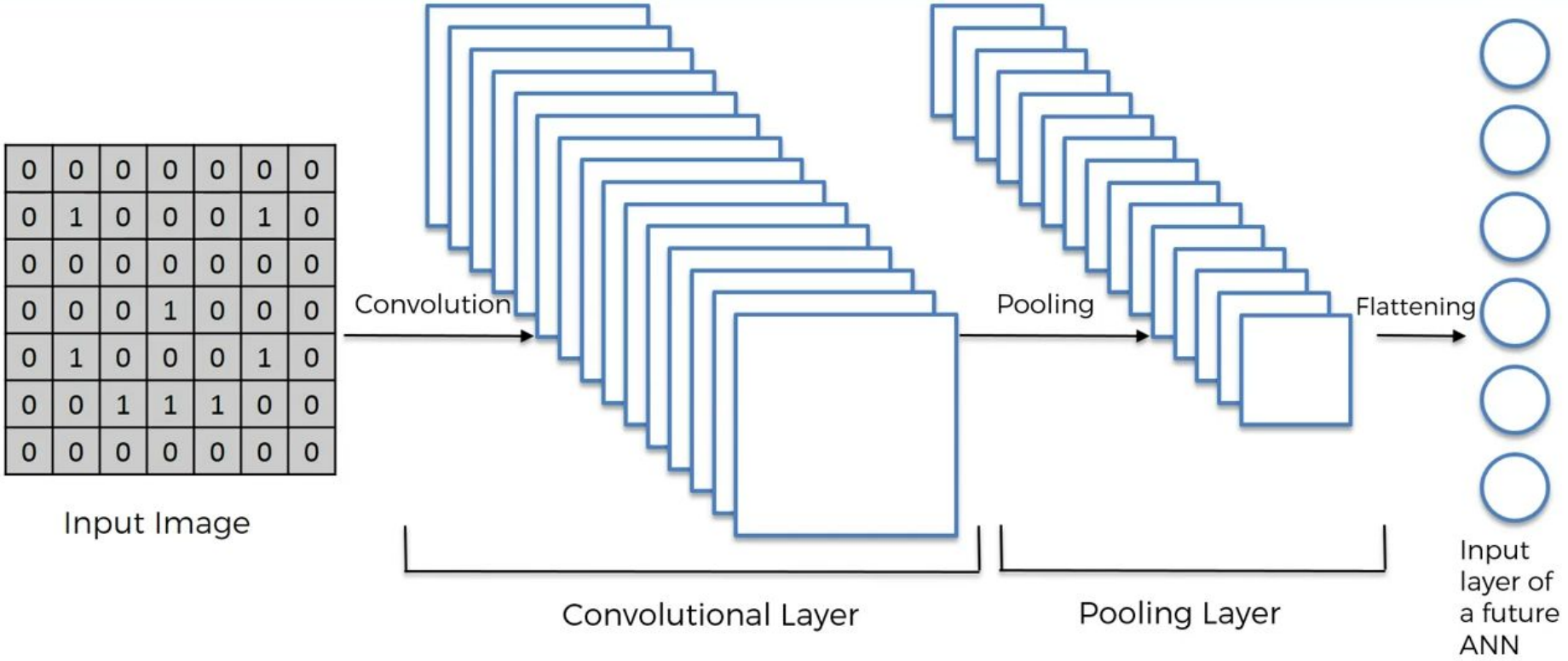
Image Source: scs.ryerson.ca/~aharley/vis/conv/flat.html

Step 3 – Flattening

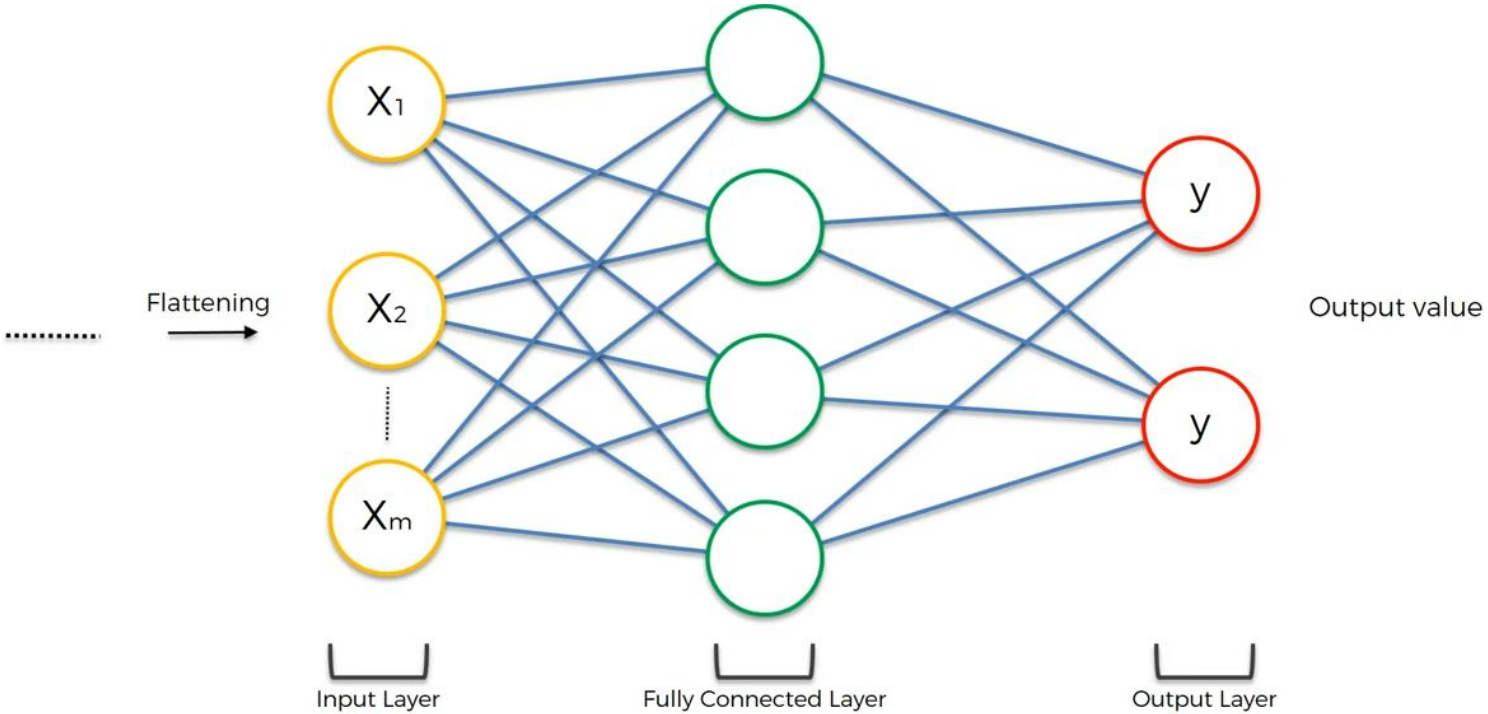
Step 3 – Flattening



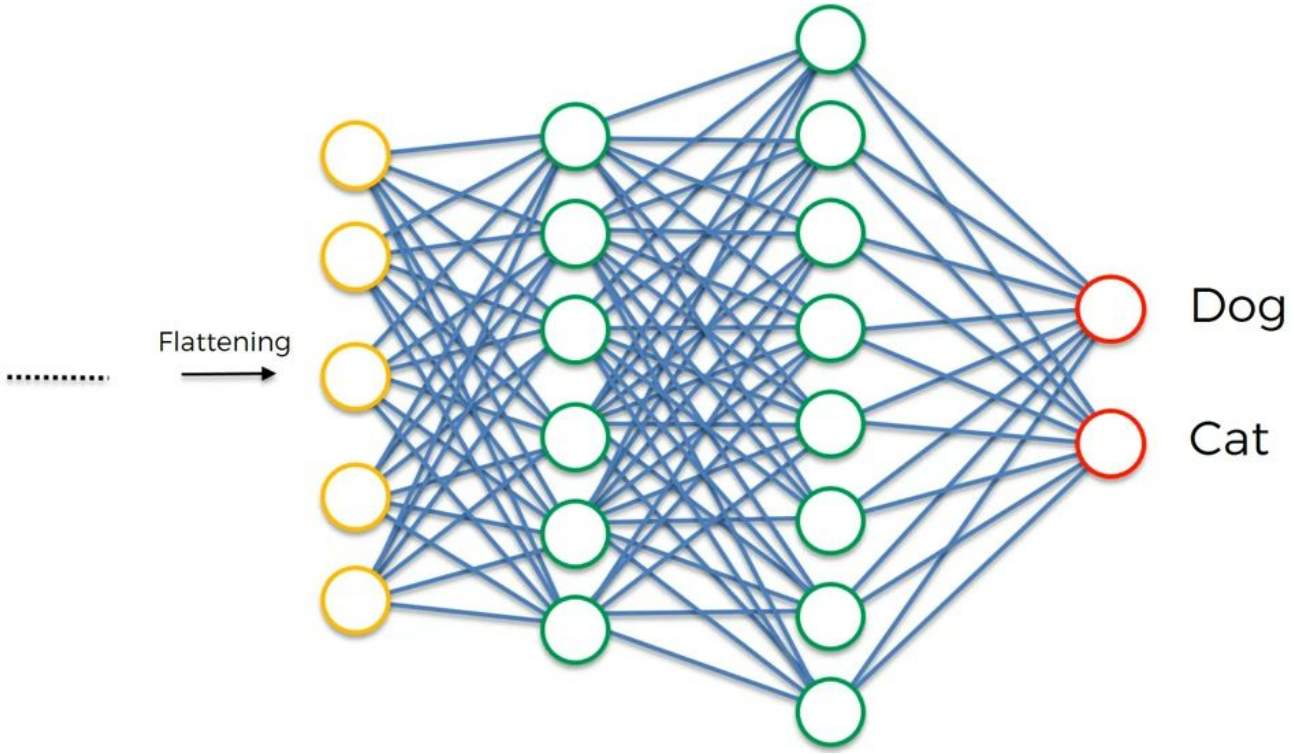
Step 3 – Flattening



Step 4 – Full Connection

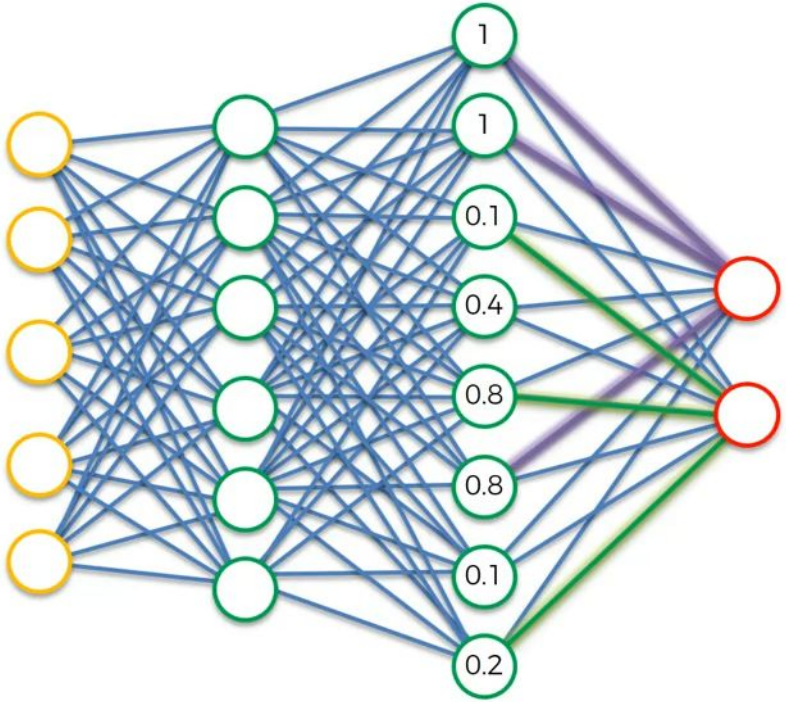


Step 4 – Full Connection

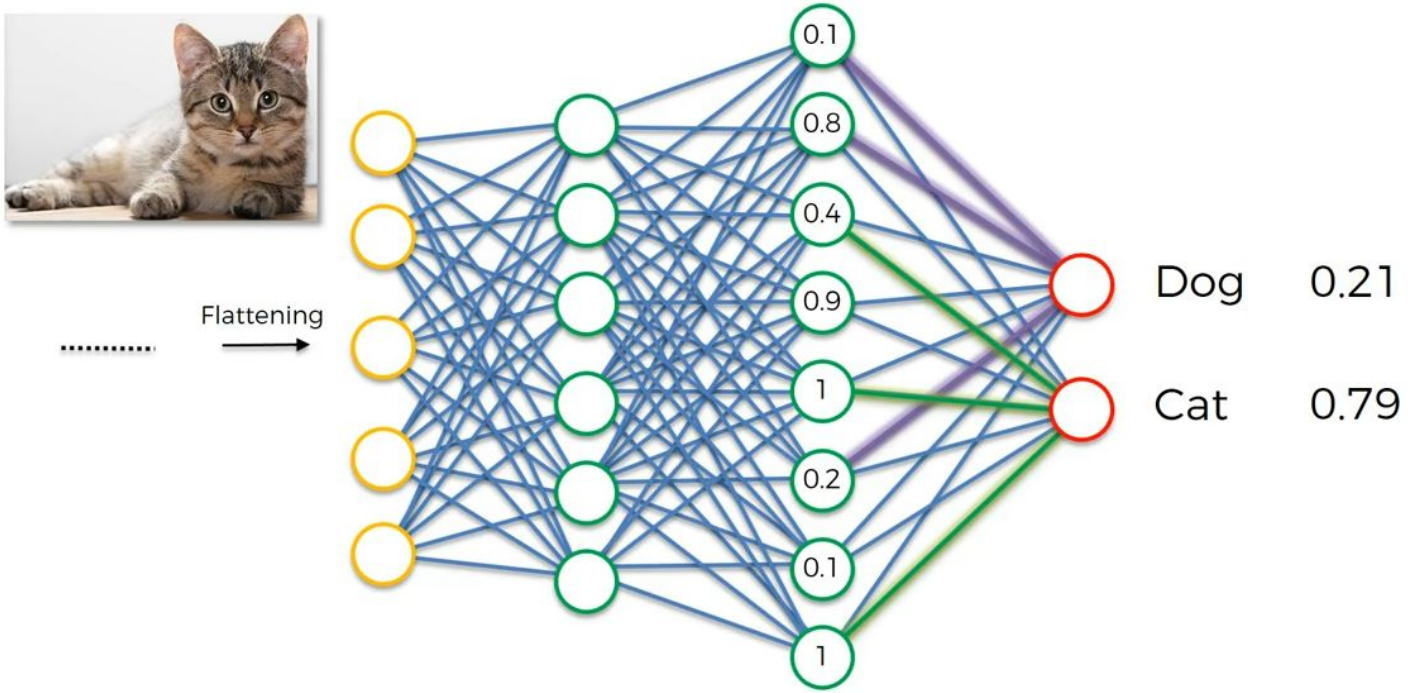


Step 4 – Full Connection

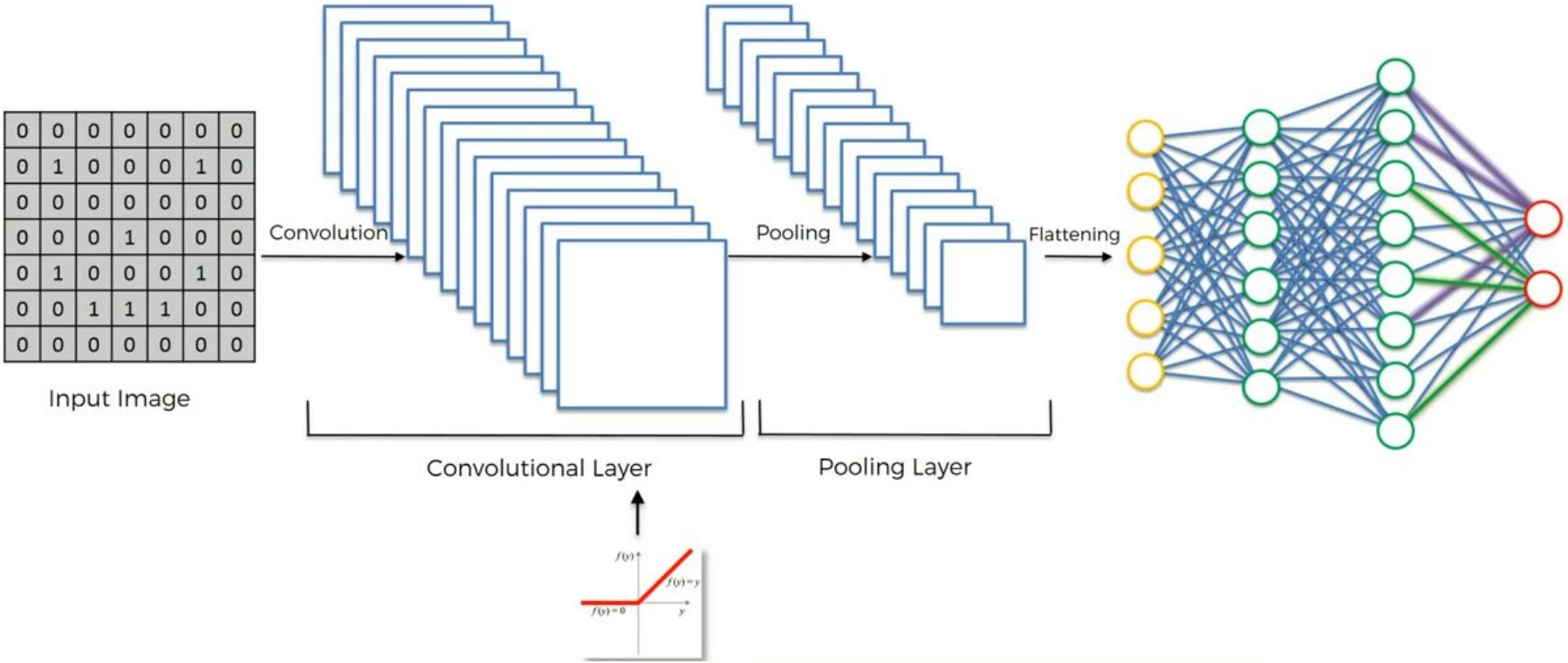
Flattening



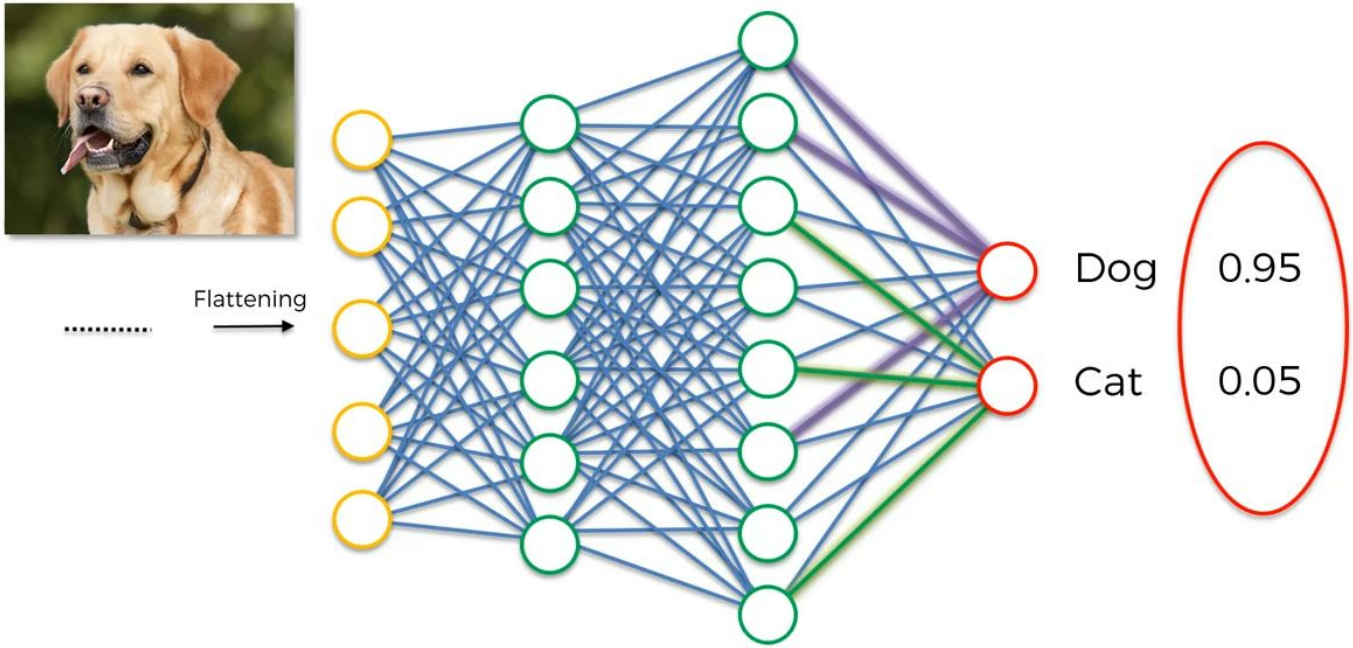
Step 4 – Full Connection



Summary

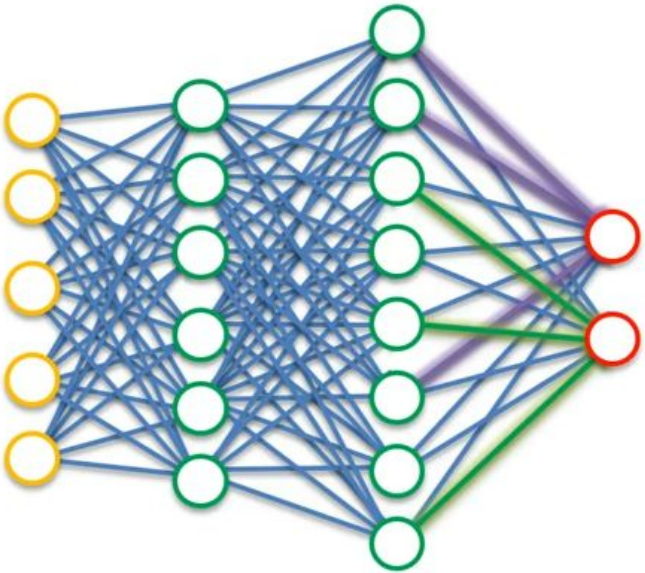

0 0 0 0 0 0 0
0 1 0 0 0 1 0
0 0 0 0 0 0 0
0 0 0 1 0 0 0
0 1 0 0 0 1 0
0 0 1 1 1 0 0
0 0 0 0 0 0 0
Input Image
Convolution
Pooling
Flattening
Convolutional Layer
Pooling Layer
f(y) = y
f(y) = 0
y

Softmax & Cross-Entropy



Softmax & Cross-Entropy

Flattening



$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

Dog → z_1 → 0.95

Cat → z_2 → 0.05

Softmax & Cross-Entropy

Dog 0.9

Cat 0.1

$$H(p,q) = -\sum_{x} p(x) \log q(x)$$

1
0

Softmax & Cross-Entropy

Dog 0.9

Cat 0.1

$$H(p,q) = -\sum_{x} p(x) \log q(x)$$

1
0

Softmax & Cross-Entropy

		NN1	NN2
Dog	1	0.9	0.6
Cat	0	0.1	0.4

		NN1	NN2
Dog	0	0.1	0.3
Cat	1	0.9	0.7

		NN1	NN2
Dog	1	0.4	0.1
Cat	0	0.6	0.9

Softmax & Cross-Entropy

NN1

Row	Dog^	Cat^	Dog	Cat
#1	0.9	0.1	1	0
#2	0.1	0.9	0	1
#3	0.4	0.6	1	0

NN2

Row	Dog^	Cat^	Dog	Cat
#1	0.6	0.4	1	0
#2	0.3	0.7	0	1
#3	0.1	0.9	1	0

Classification Error

1/3 = 0.33

1/3 = 0.33

Cross-Entropy

0.38

1.06

Lower is better

Let's Code