Convolutional Neural Networks

Unique Origin Unique Future

Convolutional Neural Networks

- **Reduction of Model Complexity**
  - Partially connected & shared weights
  - Convolutional Neural Network (CNN)
  - Specialized to sequential data
    - Images, Speech, Text

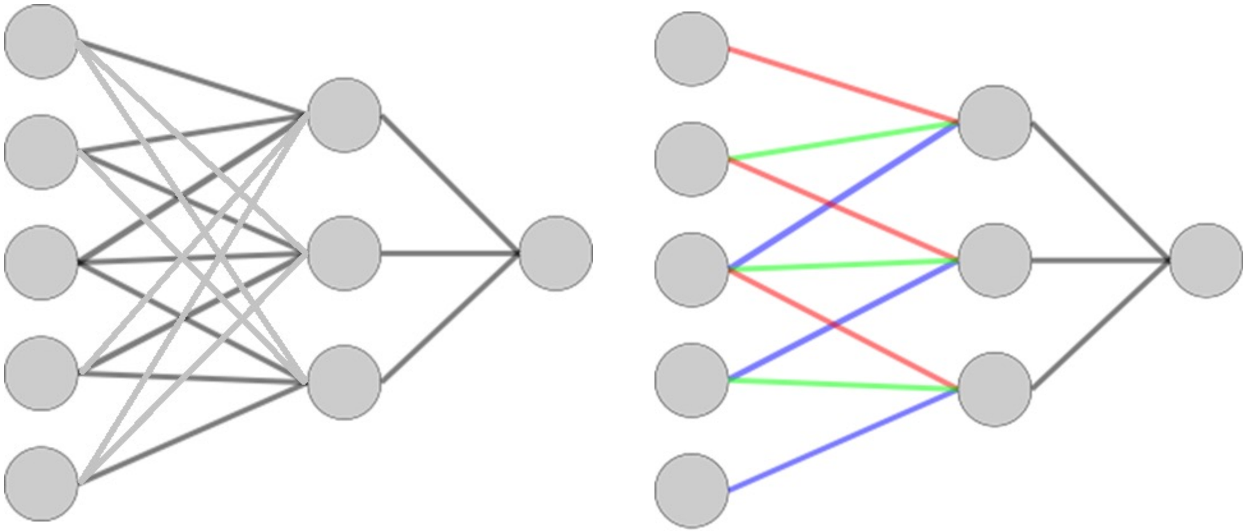
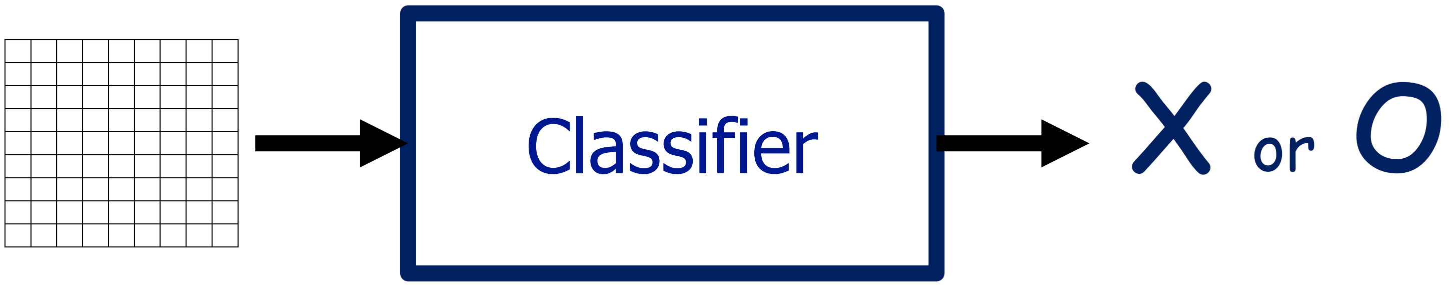


Image Classification

- **X, O Classification**

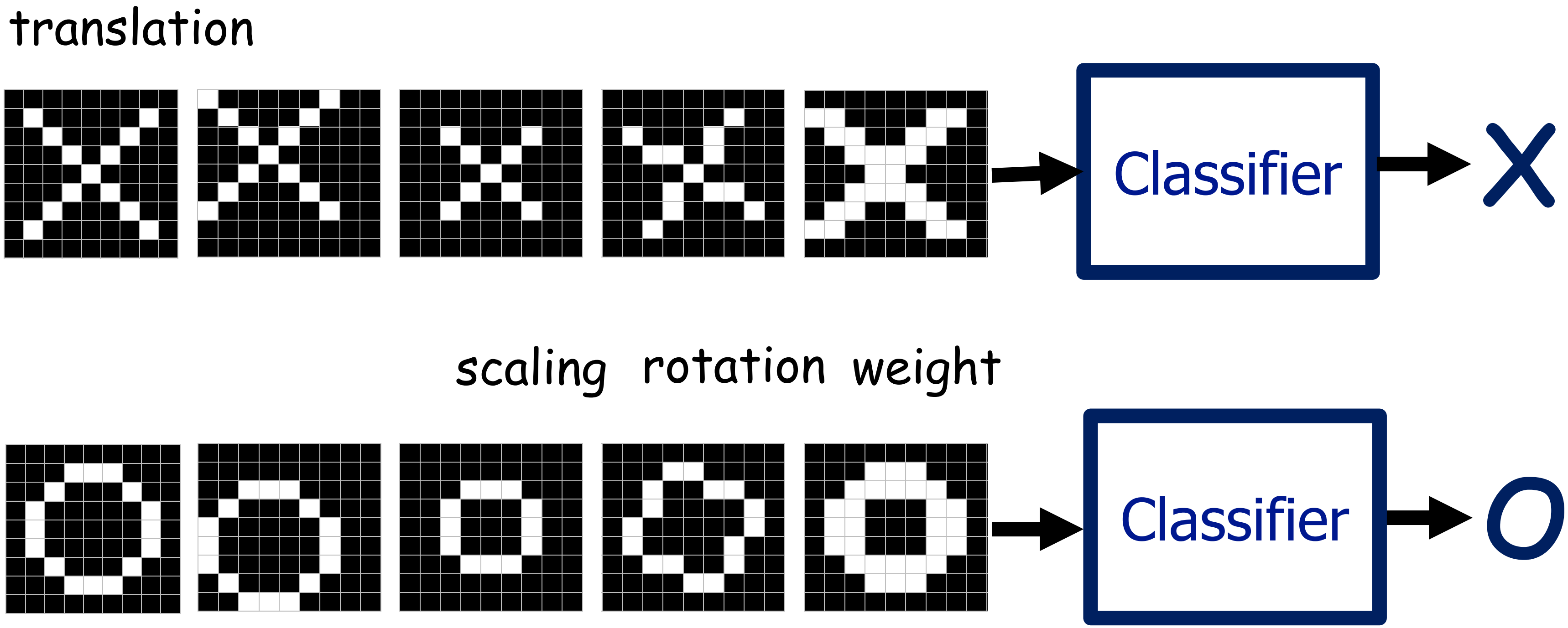


Classifier

X or O

A two-dimensional array of pixels

Image Classification



translation

Classifier

X

scaling rotation weight

Classifier

O

Image Classification

- **Same?**
  - How to determine both are the same?

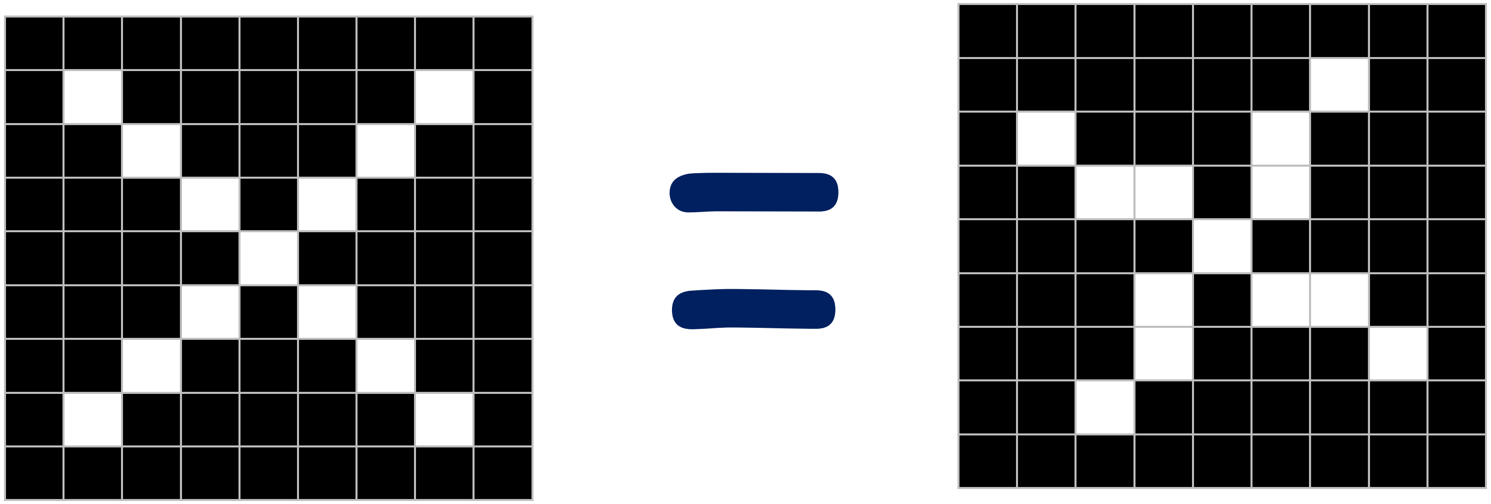


Image Classification

- **Both are partially matching**
  - Critical LOCAL features are the same

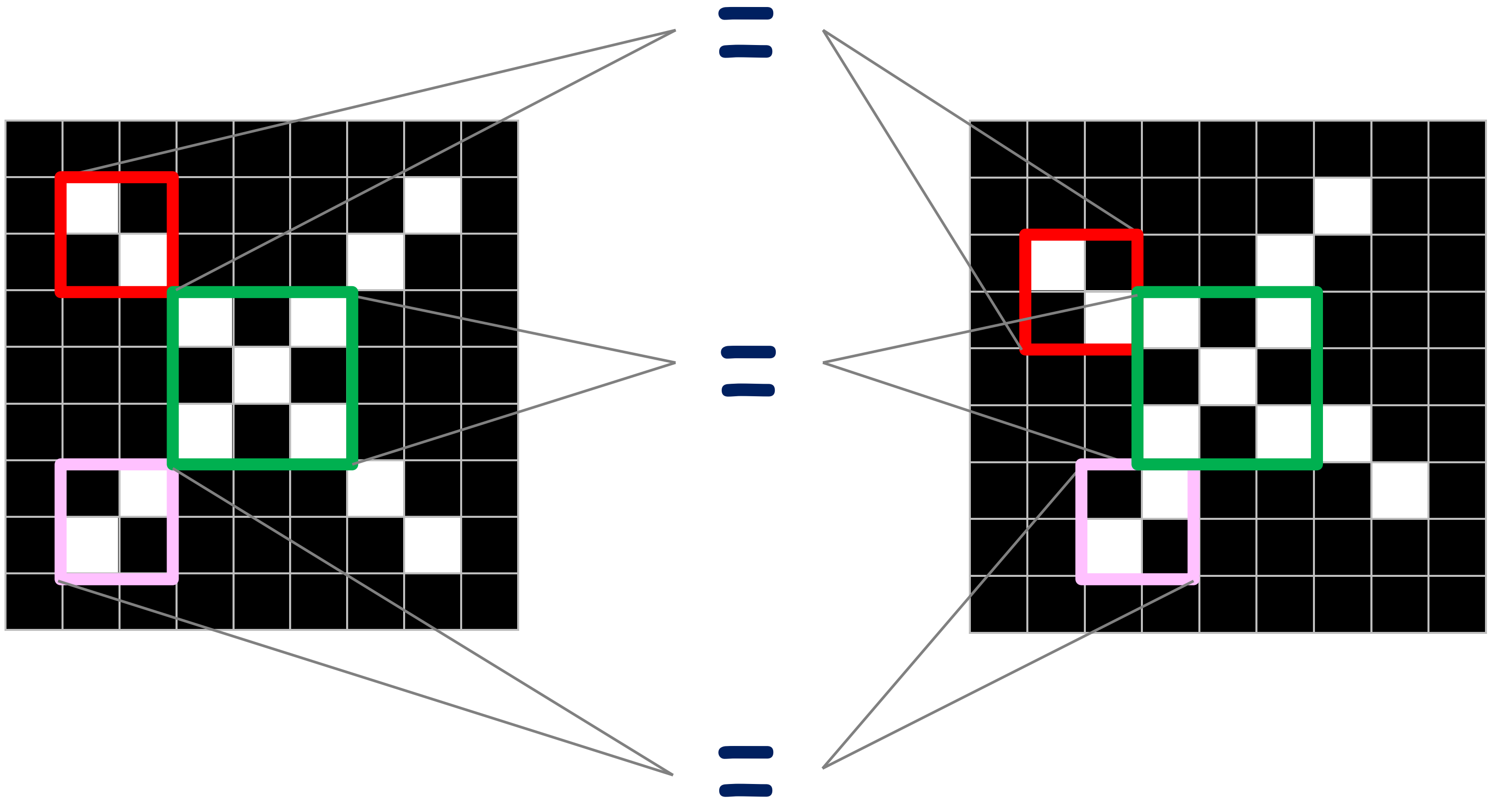
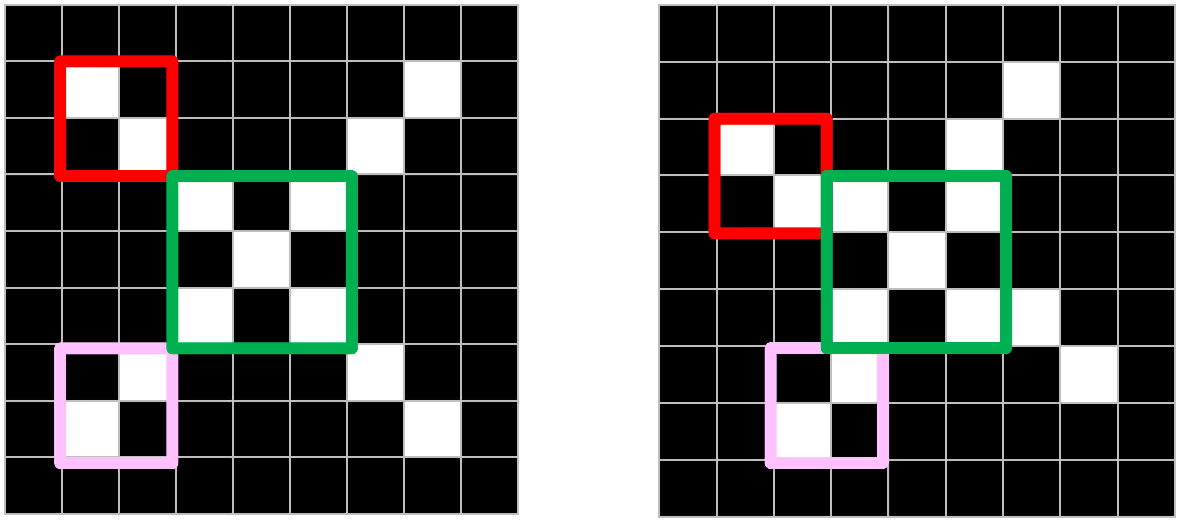


Image Classification

- **Local Features**
  - How about finding local features, and
  - Matching them



Feature Extraction

- **Convolution**
  - A way to find out local features

$$\begin{array}{|c|c|c|}\hline 1 & 0 & 1 \\ \hline 0 & 1 & 1 \\ \hline 1 & 1 & 0 \\ \hline \end{array} \; * \; \begin{array}{|c|c|c|}\hline 1 & 0 & 1 \\ \hline 0 & 1 & 0 \\ \hline 1 & 0 & 1 \\ \hline \end{array} \; = \; 4$$

$$I * K = \sum K_{ij} \times I_{ij}$$

Feature Extraction

- **Convolution**

-1	-1	1
-1	1	-1
1	-1	-1

$$(I * K)_{xy} = \sum_{i=1}^{w}\sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y+j-1}$$

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

	-1						

Feature Extraction

- **Convolution**

-1	-1	1
-1	1	-1
1	-1	-1

$$(I * K)_{xy} = \sum_{i=1}^{w}\sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y+j-1}$$

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

	-1						
	-2						

Feature Extraction

- **Convolution**

-1	-1	1
-1	1	-1
1	-1	-1

$$(I * K)_{xy} = \sum_{i=1}^{w}\sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y+j-1}$$

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

	-1						
	-2						
	1						

Feature Extraction

- **Convolution**

-1	-1	1
-1	1	-1
1	-1	-1

$$(I * K)_{xy} = \sum_{i=1}^{w} \sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y+j-1}$$

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

	-1	-2	0	-2	-1	1	
	-2	3	-2	-2	-2	-2	
	1	-1	-2	0	-4	-1	
	-1	-3	0	-2	0	1	
	-2	-1	-4	0	2	-2	
	1	-2	0	0	-3	-1	

Feature Extraction

- **Convolution**

-1	-1	1
-1	1	-1
1	-1	-1

$$(I * K)_{xy} = \sum_{i=1}^{w}\sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y+j-1}$$

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

0	0	-1	-2	0	1	0	0
0	-1	-2	0	-2	-1	1	0
-1	-2	3	-2	-2	-2	-2	1
-2	1	-1	-2	0	-4	-1	0
0	-1	-3	0	-2	0	1	-2
1	-2	-1	-4	0	2	-2	-1
0	1	-2	0	0	-3	-1	0
0	0	1	0	-2	-1	0	0

Feature Extraction

- **Threshold**

-1	-1	1
-1	1	-1
1	-1	-1

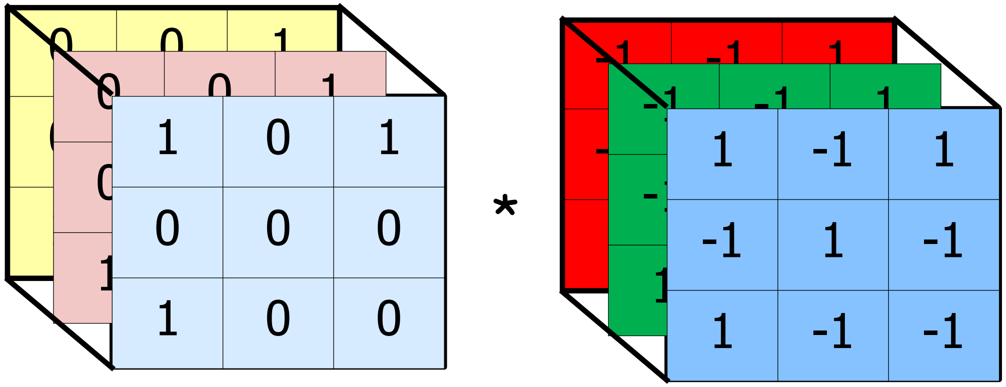
$$(I * K)_{xy} = \sum_{i=1}^{w}\sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y+j-1}$$

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

0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	3	0	0	0	0	1
0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0
1	0	0	0	0	2	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0

Feature Extraction

- **Convolution (3D)**



1	0	1
0	0	0
1	0	0

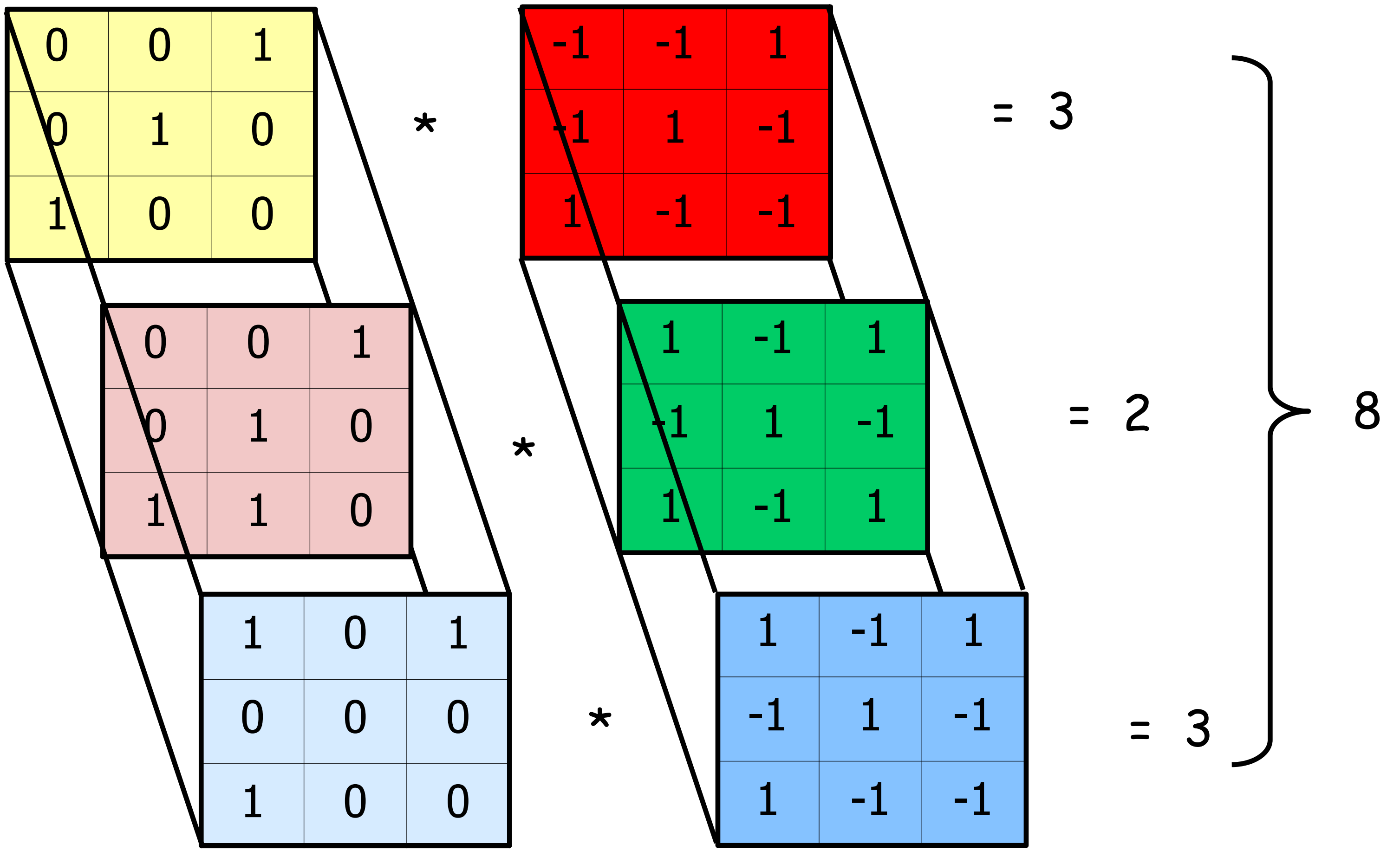
1	-1	1
-1	1	-1
1	-1	-1

= 3 +2 +3 = 8

$$I * K = \sum_{channel} \sum_{i,j} K_{ij}^{channel} \times I_{ij}^{channel}$$

Feature Extraction

- **Convolution (3D)**



$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} -1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix} = 3$$

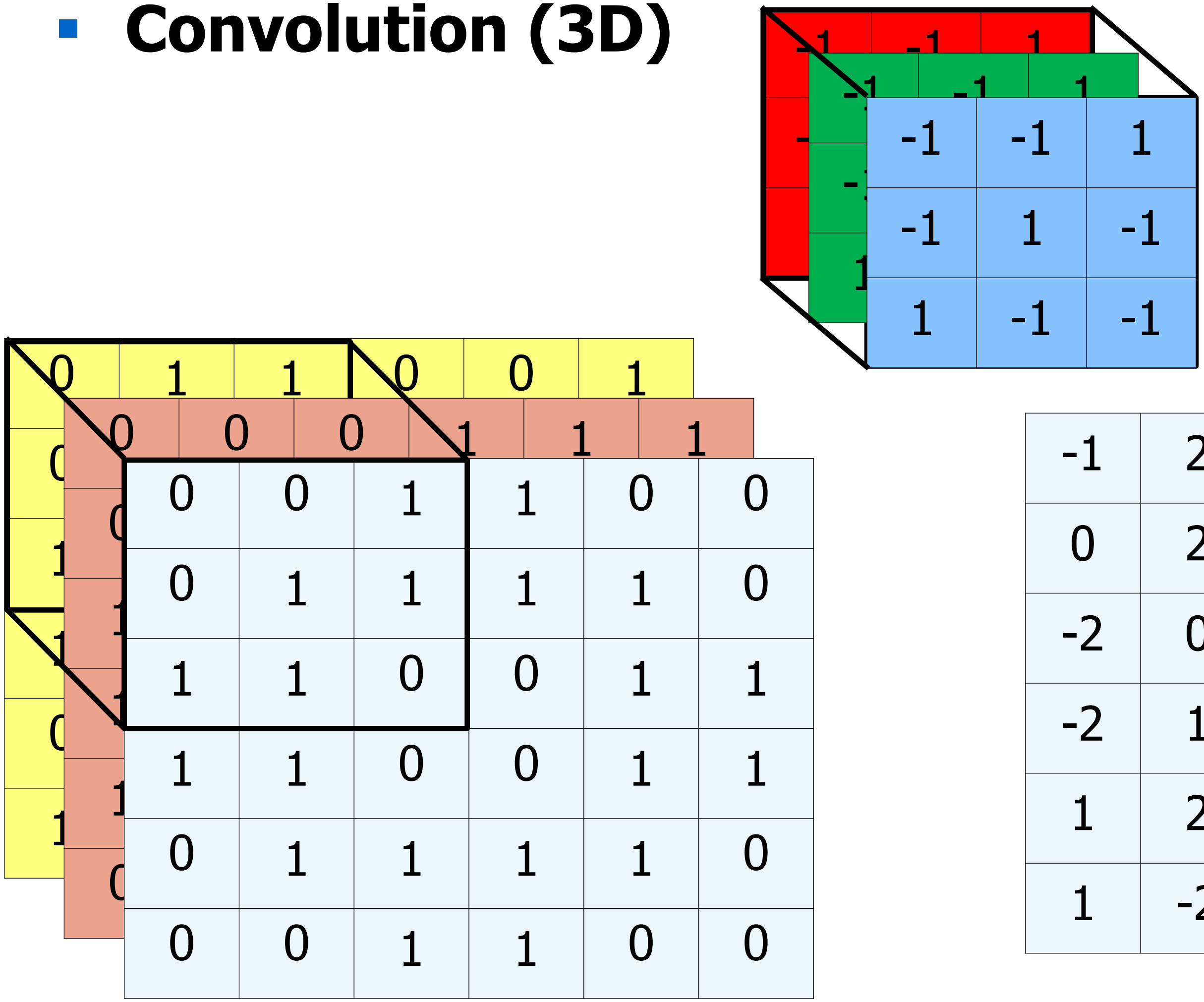
$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} = 2$$

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix} = 3$$

$$\left.\begin{matrix} 3 \\ 2 \\ 3 \end{matrix}\right\} 8$$

Feature Extraction

- **Convolution (3D)**



-1	-1	1
-1	1	-1
1	-1	-1

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

-1	2	4	-2	-1	-4
0	2	5	1	3	2
-2	0	4	2	-6	1
-2	1	0	-2	3	-3
1	2	-4	4	2	-2
1	-2	-4	3	2	1

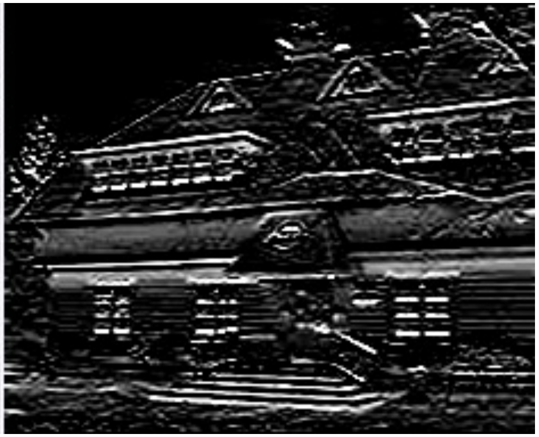
Feature Extraction

Examples of Convolution

1	1	1
0	0	0
-1	-1	-1

Horizontal Line

Input



Output

(Feature Map)

Feature Extraction

- **Pooling**
  - Subsampling from *m* by *m* pixels into 1 pixels
  - Max, averaging or L^p pooling

0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	3	0	0	0	0	1
0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0
1	0	0	0	0	2	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0

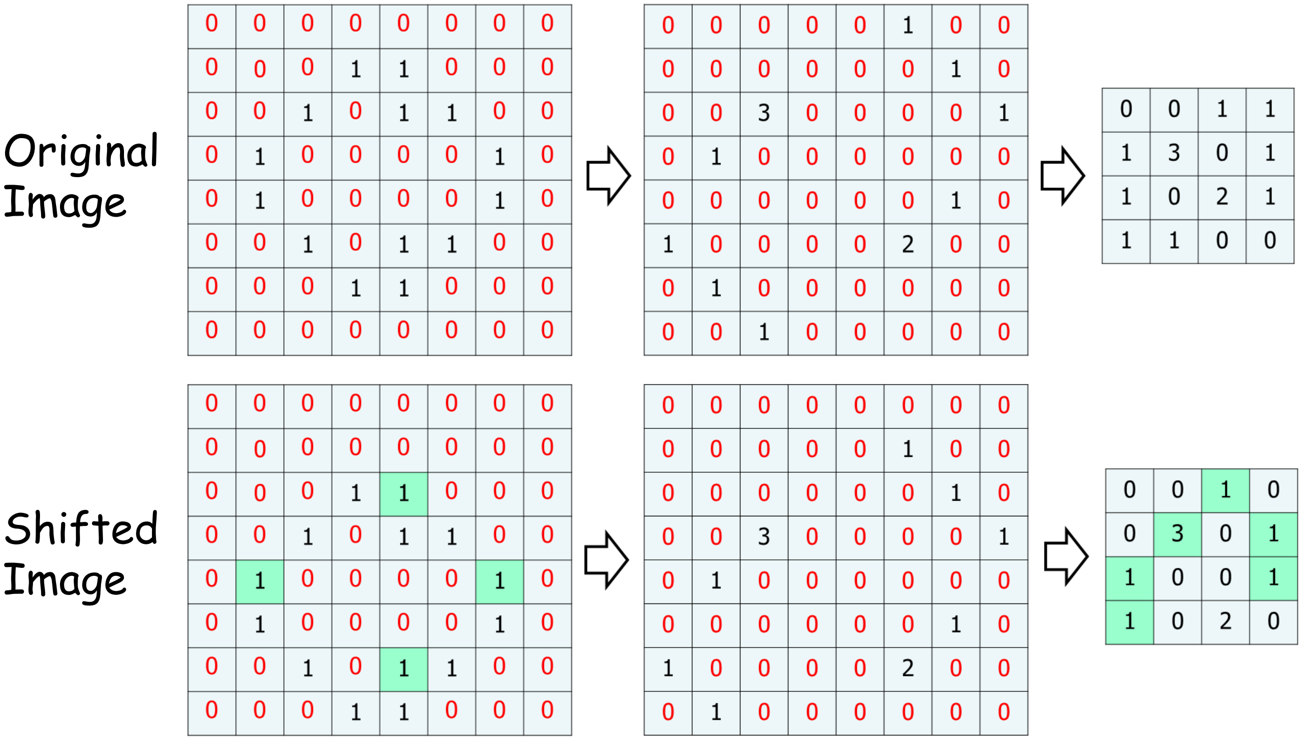
Feature map

0	0	1	1
1	3	0	1
1	0	2	1
1	1	0	0

Subsampled
feature map

Feature Extraction

- **Advantage of Pooling**
  - Reducing the number of parameters
  - Generating more robust feature maps: Shift Invariant



Original Image

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

0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	3	0	0	0	0	1
0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0
1	0	0	0	0	2	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0

0	0	1	1
1	3	0	1
1	0	2	1
1	1	0	0

Shifted Image

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

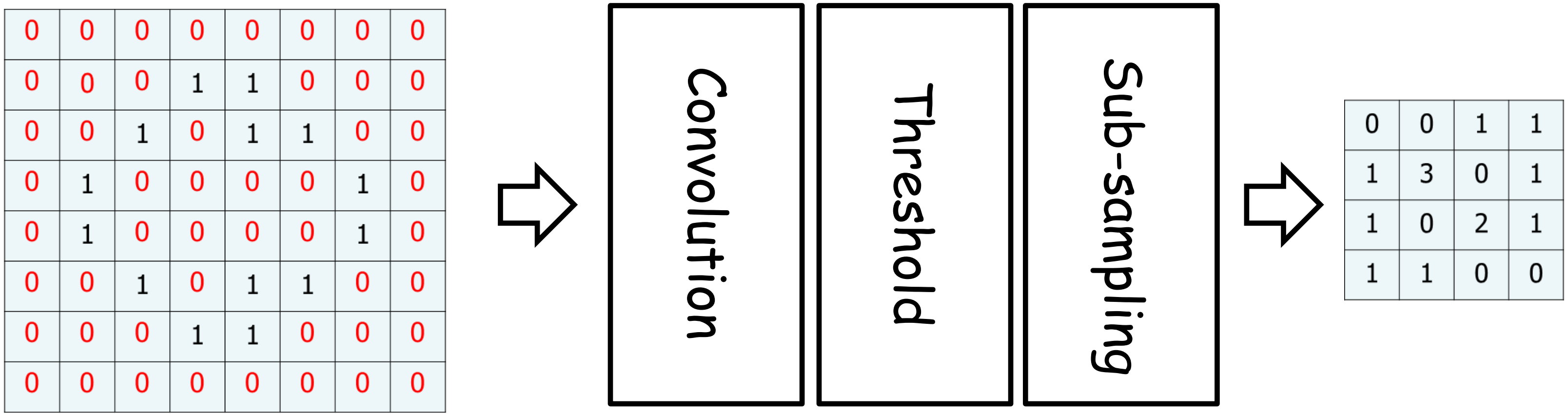
0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	3	0	0	0	0	1
0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0
1	0	0	0	0	2	0	0
0	1	0	0	0	0	0	0

0	0	1	0
0	3	0	1
1	0	0	1
1	0	2	0

More
non-zero values
are matched !!

Convolutional Neural Networks

- **Feature Extraction**



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

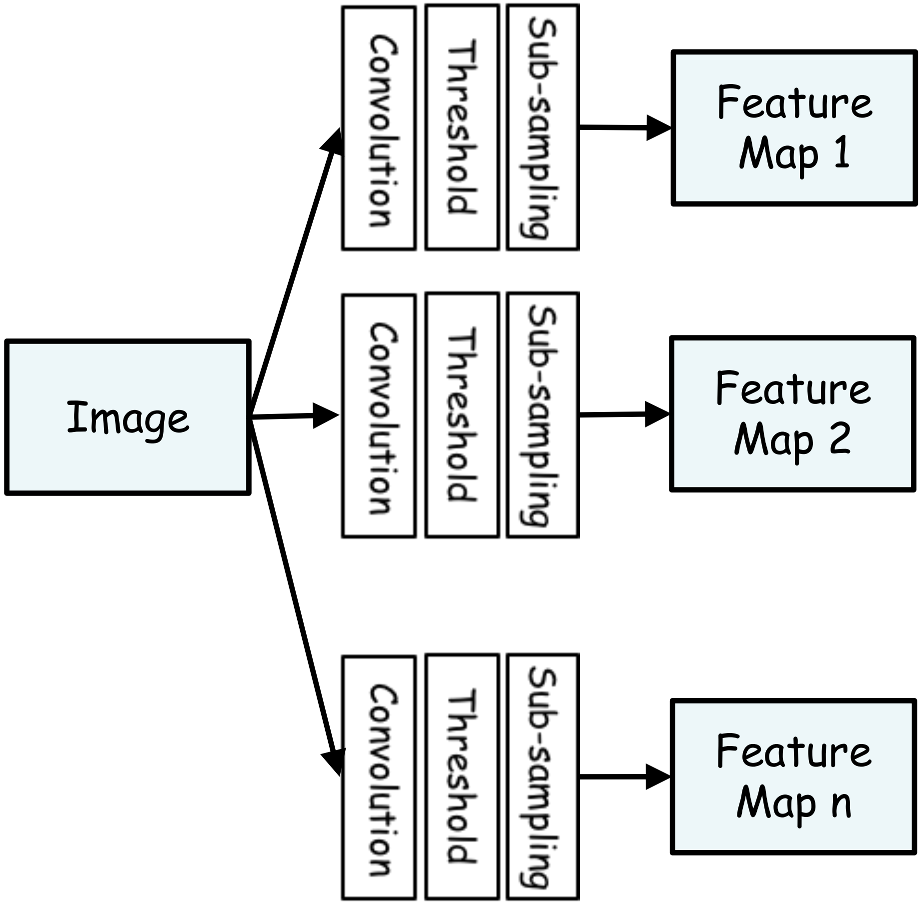
Image

0	0	1	1
1	3	0	1
1	0	2	1
1	1	0	0

Feature map

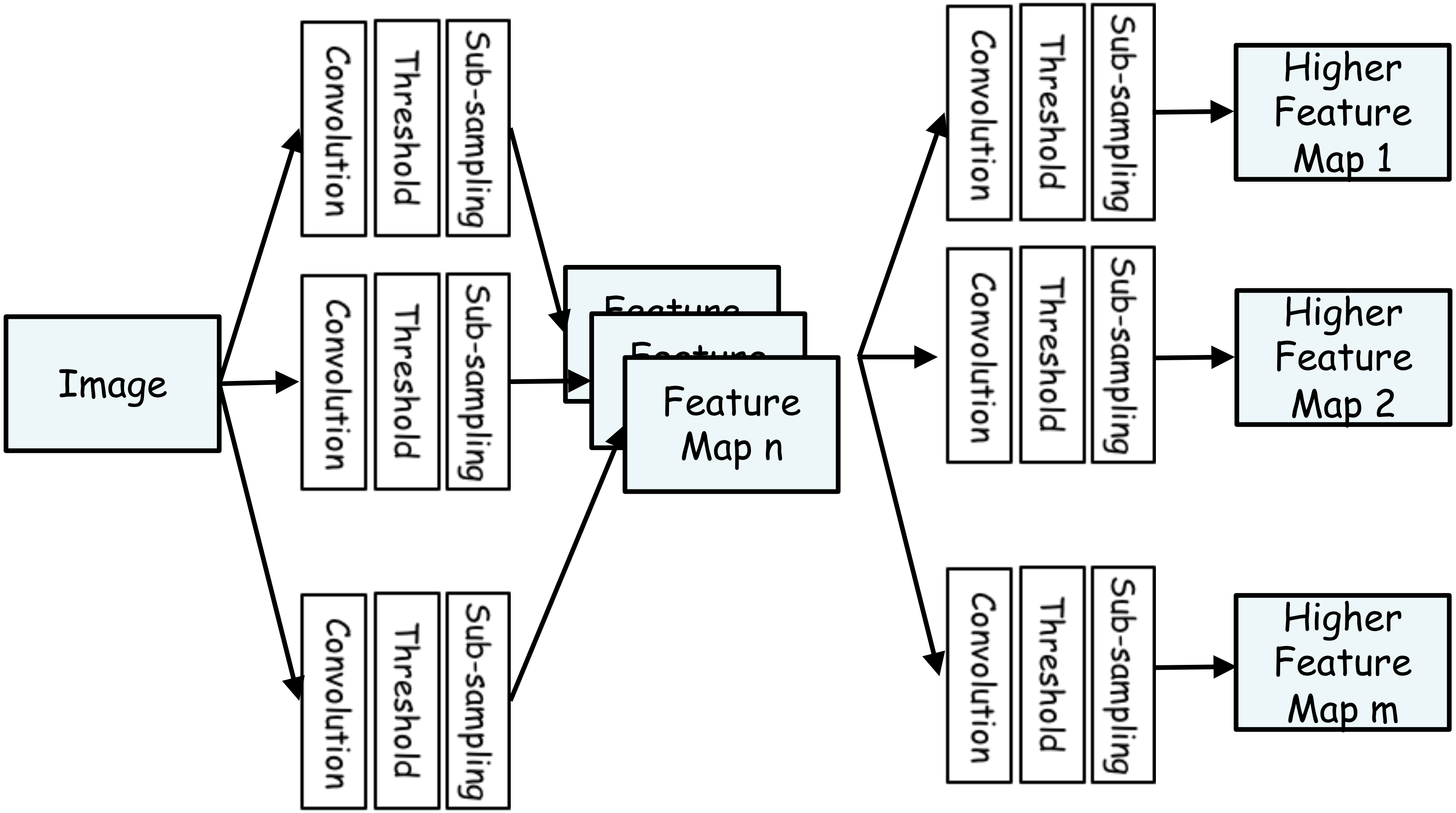
Convolutional Neural Networks

- **Structure**



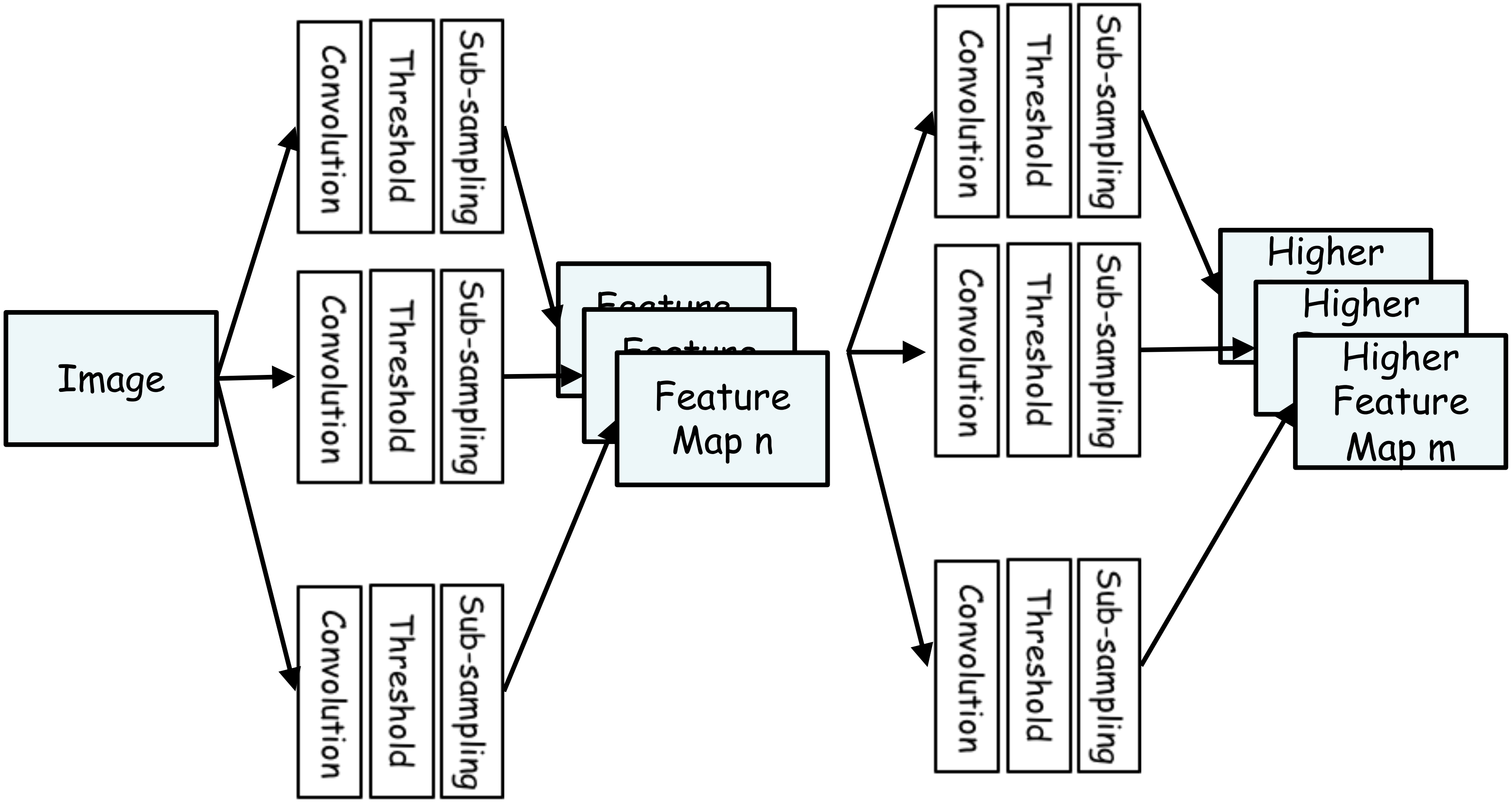
Convolutional Neural Networks

- **Structure**



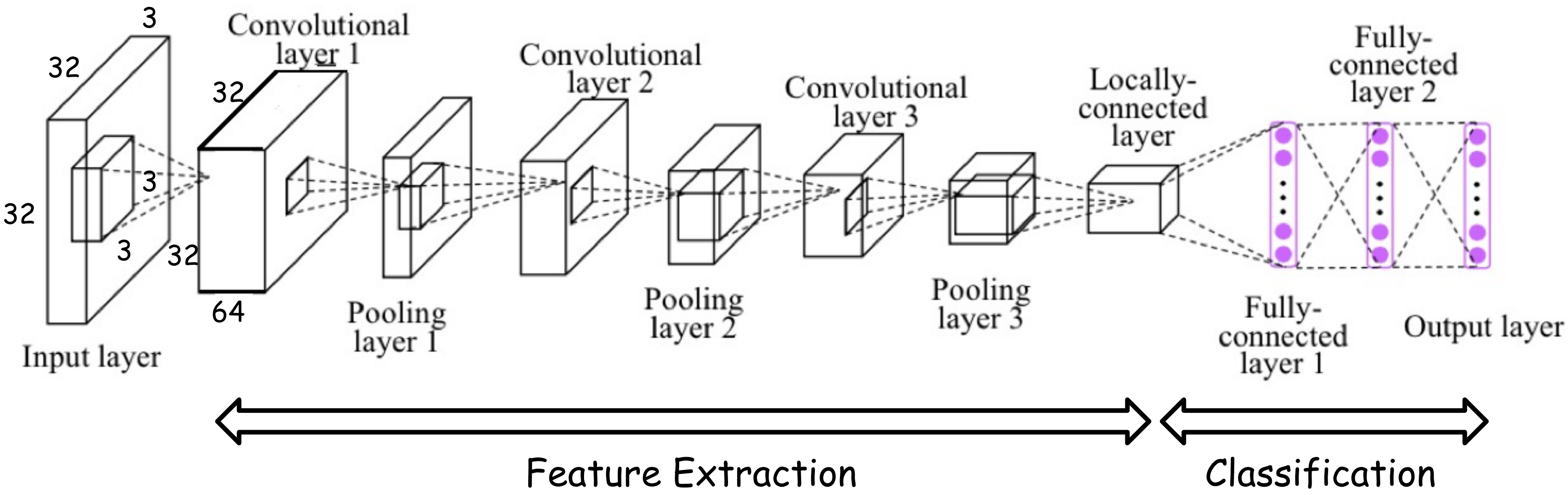
Convolutional Neural Networks

- **Structure**



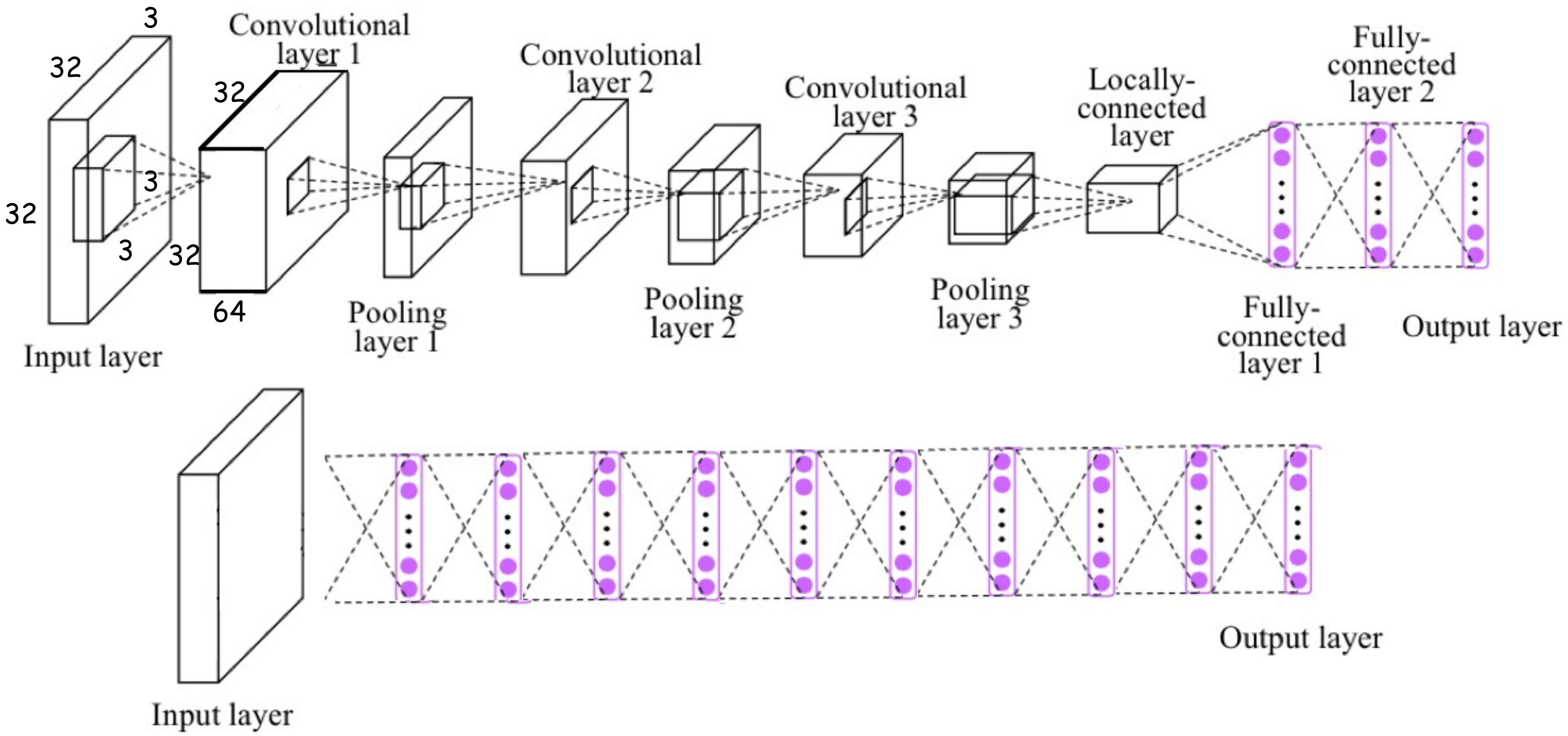
Convolutional Neural Networks

- **Graphical Representation**



Neural Network Representation

- **Convolutional Neural Networks**

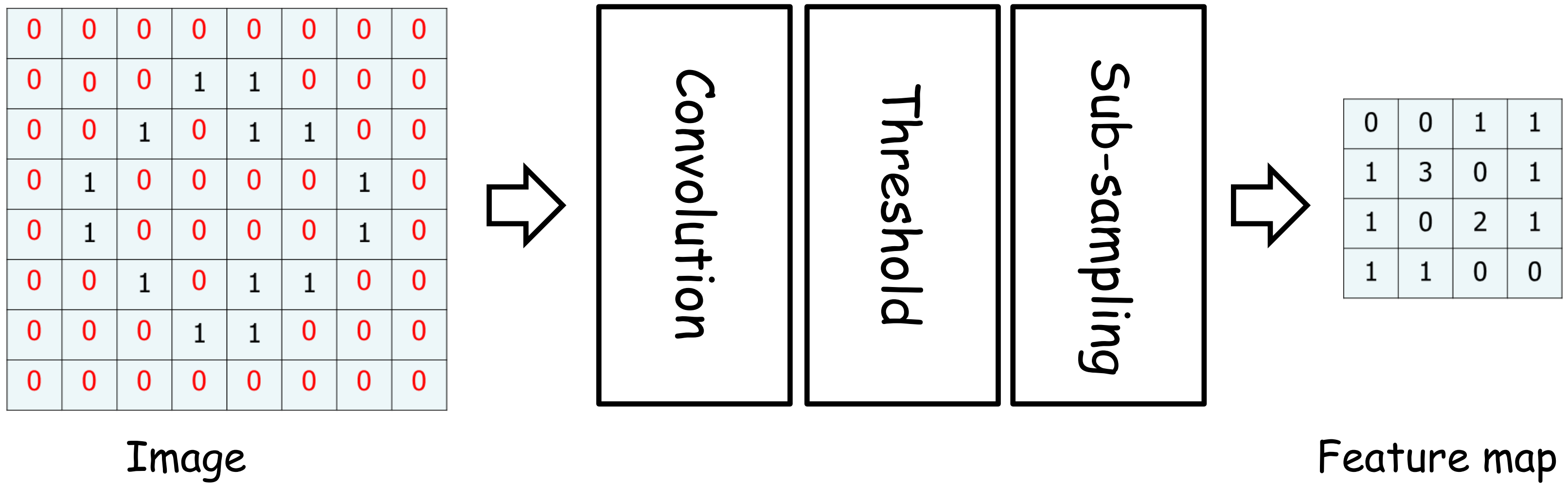


Convolutional Neural Networks

- **Who determines convolution masks?**
  - designed by EXPERTS!!
- **In CNN**
  - CNNs can be converted into neural networks
  - Convolution masks are converted into connection weights
  - Masks are found with gradient descent methods

Neural Network Representation

- **Feature Extraction**



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

Image

Convolution

Threshold

Sub-sampling

0	0	1	1
1	3	0	1
1	0	2	1
1	1	0	0

Feature map

Neural Network Representation

- **Convolution & Threshold**
  - Values in Kernel
    => Connection weights
  - Most of them are zeros

input

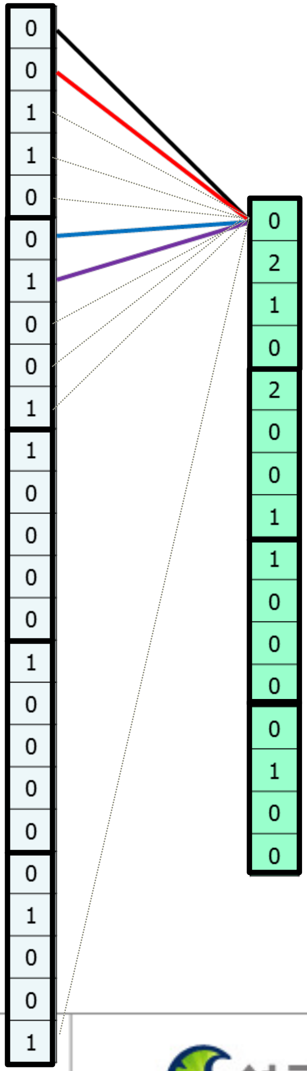
0	0	1	1	0
0	1	0	0	1
1	0	0	0	0
1	0	0	0	0
0	1	0	0	1

0	**1**
1	**0**

Kernel

output

0	2	1	0
2	0	0	1
1	0	0	0
0	1	0	0



Neural Network Representation

- **Convolution & Threshold**
  - Values in Kernel
    => Connection weights
  - Most of them are zeros

input

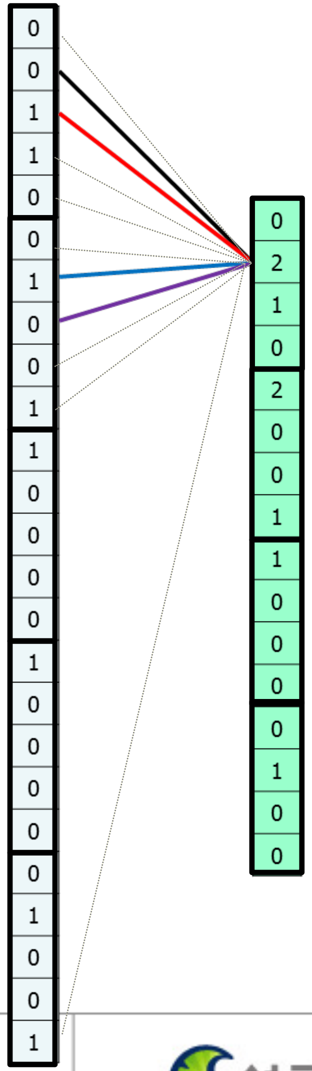
0	0	1	1	0
0	1	0	0	1
1	0	0	0	0
1	0	0	0	0
0	1	0	0	1

0	**1**
1	**0**

Kernel

output

0	2	1	0
2	0	0	1
1	0	0	0
0	1	0	0



Neural Network Representation

- **Convolution & Threshold**
  - Values in Kernel
    => Connection weights
  - Most of them are zeros

input

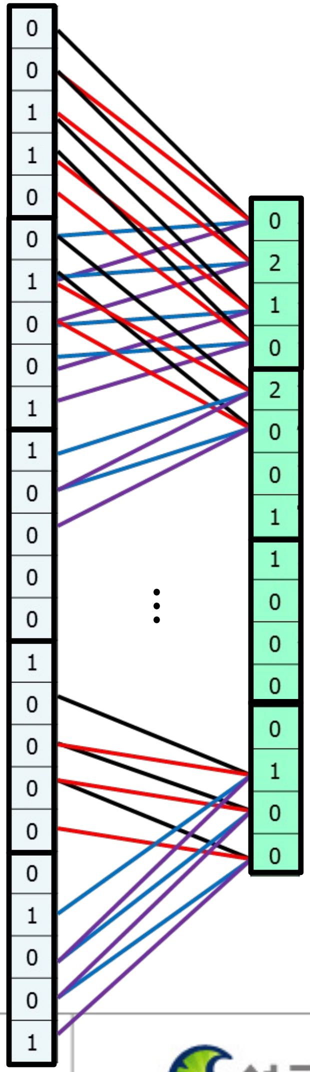
0	0	1	1	0
0	1	0	0	1
1	0	0	0	0
1	0	0	0	0
0	1	0	0	1

0	**1**
1	**0**

Kernel

output

0	2	1	0
2	0	0	1
1	0	0	0
0	1	0	0



Neural Network Representation

- **Pooling**
  - Weight are fixed to 1
  - Activation:

$$y = \max(w_1 x_1, w_2 x_2, w_3 x_3, w_4 x_4)$$

input

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

0	**1**
1	**0**

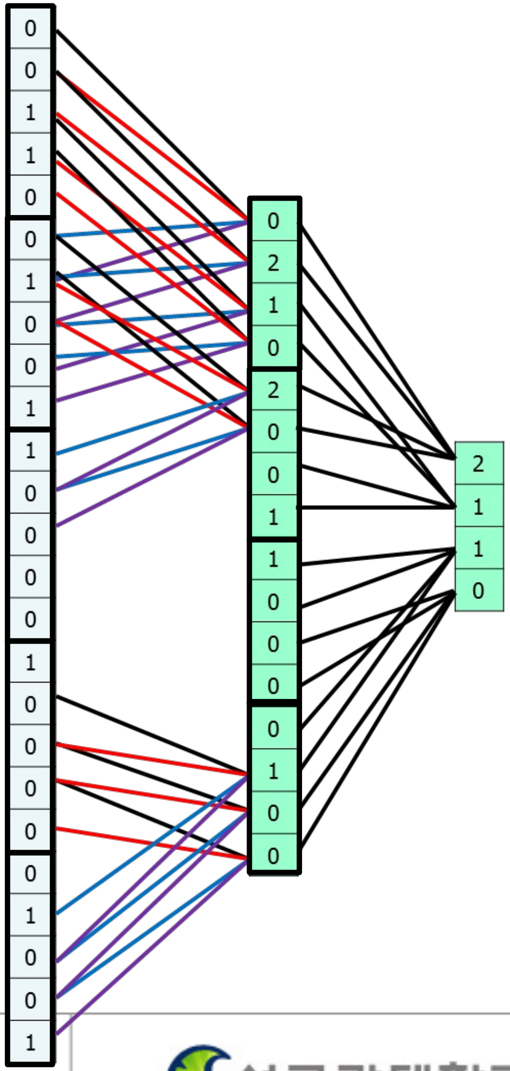
Kernel

output

0	2	1	0
2	0	0	1
1	0	0	0
0	1	0	0

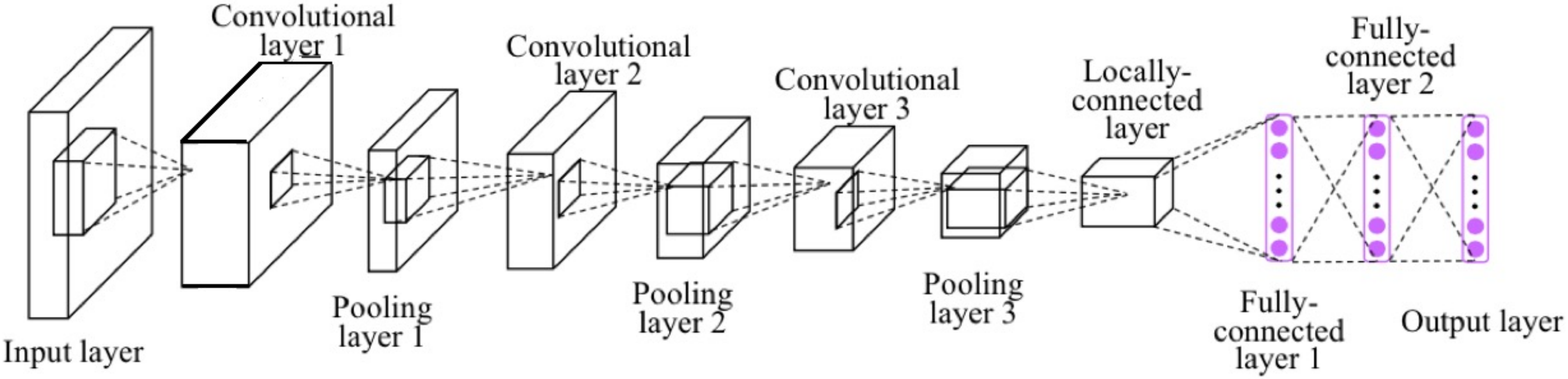
2	1
1	0

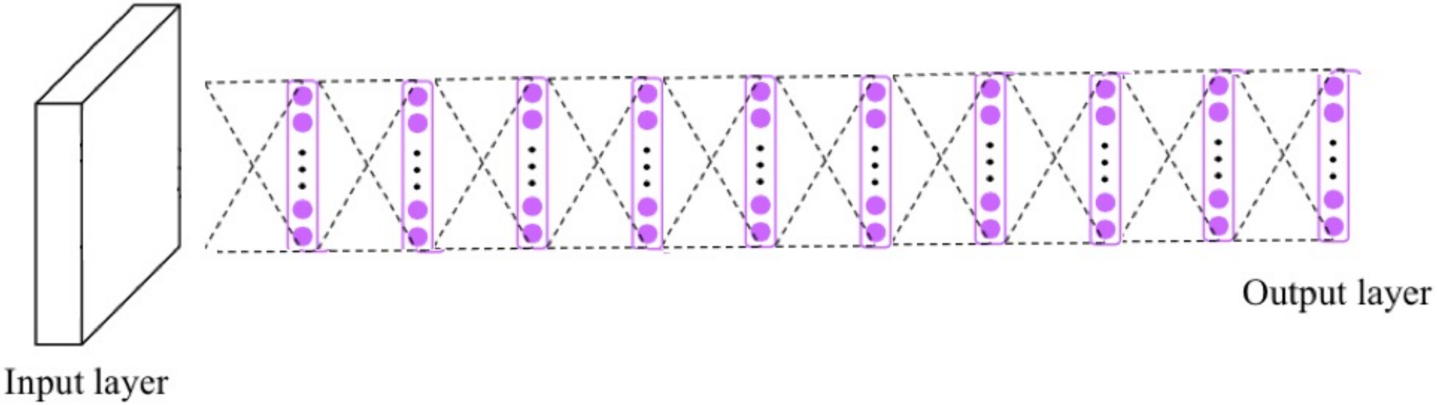
Subsampled output



Neural Network Representation

- **Structure**


Convolutional layer 1
Convolutional layer 2
Convolutional layer 3
Locally-connected layer
Fully-connected layer 2
Input layer
Pooling layer 1
Pooling layer 2
Pooling layer 3
Fully-connected layer 1
Output layer


Output layer
Input layer