Convolutional Neural Networks
CS-477 Computer Vision

Dr. Mohsin Kamal
Associate Professor
dr.mohsinkamal@seecs.edu.pk

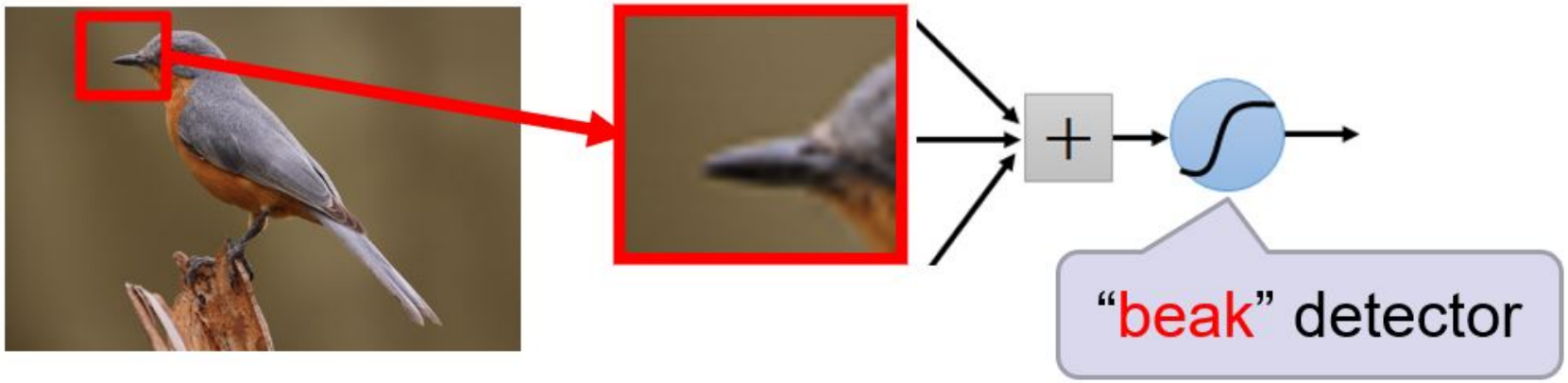
School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST), Pakistan

- A convolutional neural network (CNN) is a type of artificial neural network used primarily for image recognition and processing, due to its ability to recognize patterns in images.
- FNN could not scale up to image and video processing tasks.
- CNN specifically tailored for image and video processing tasks.

Consider learning an image:

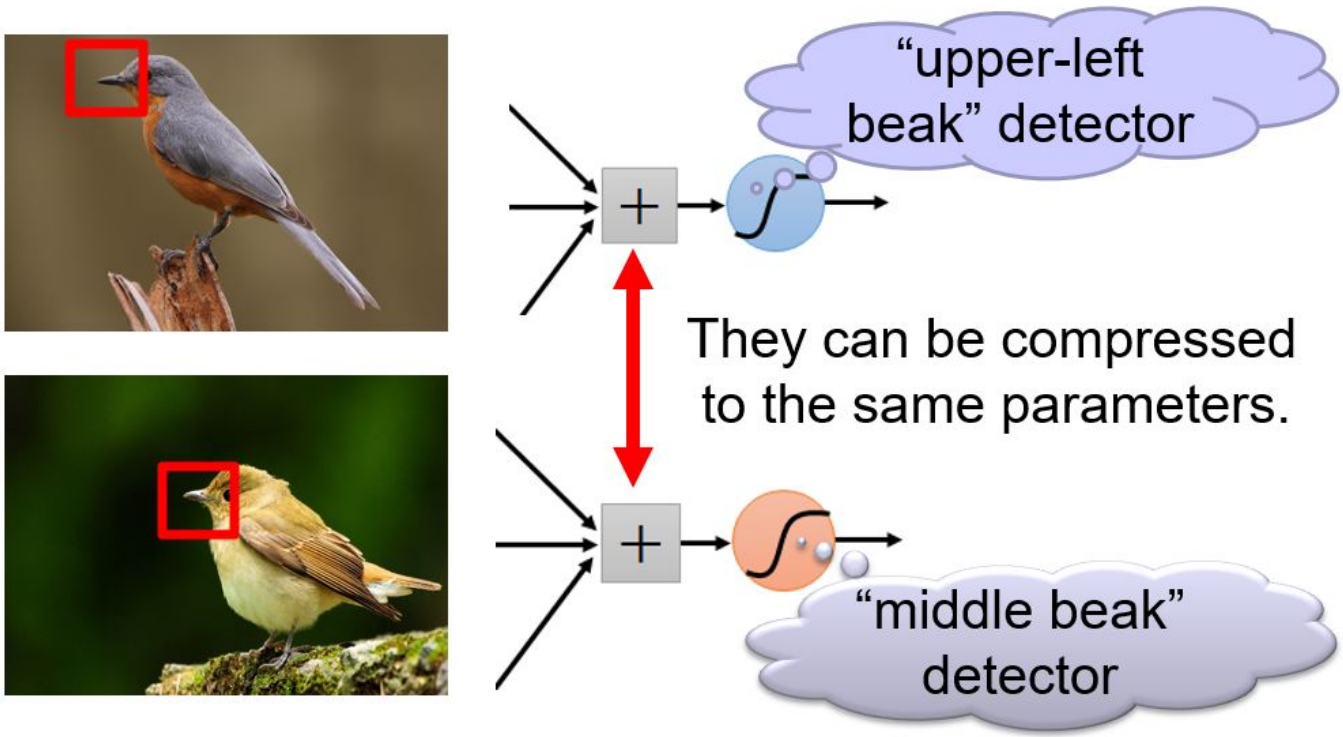
Some patterns are much smaller than the whole image

Can represent a small region with fewer parameters

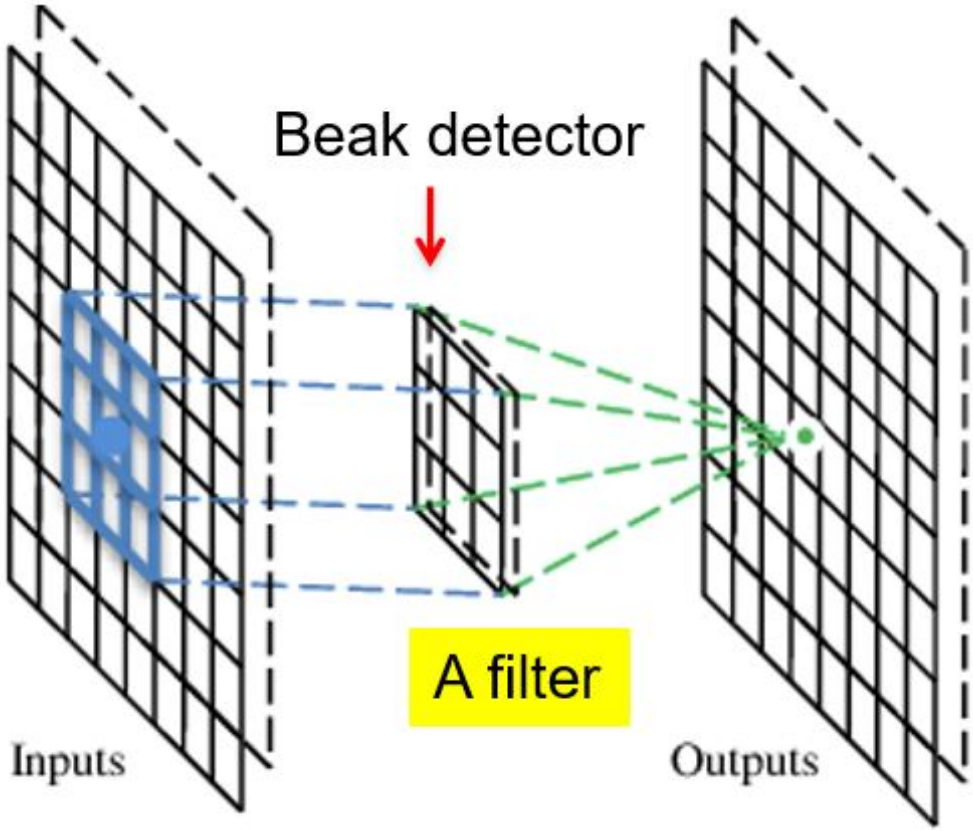


Same pattern appears in different places:

- They can be compressed!
- What about training a lot of such "small" detectors and each detector must "move around".



A CNN is a neural network with some convolutional layers (and some other layers). A convolutional layer has a number of filters that does convolutional operation.

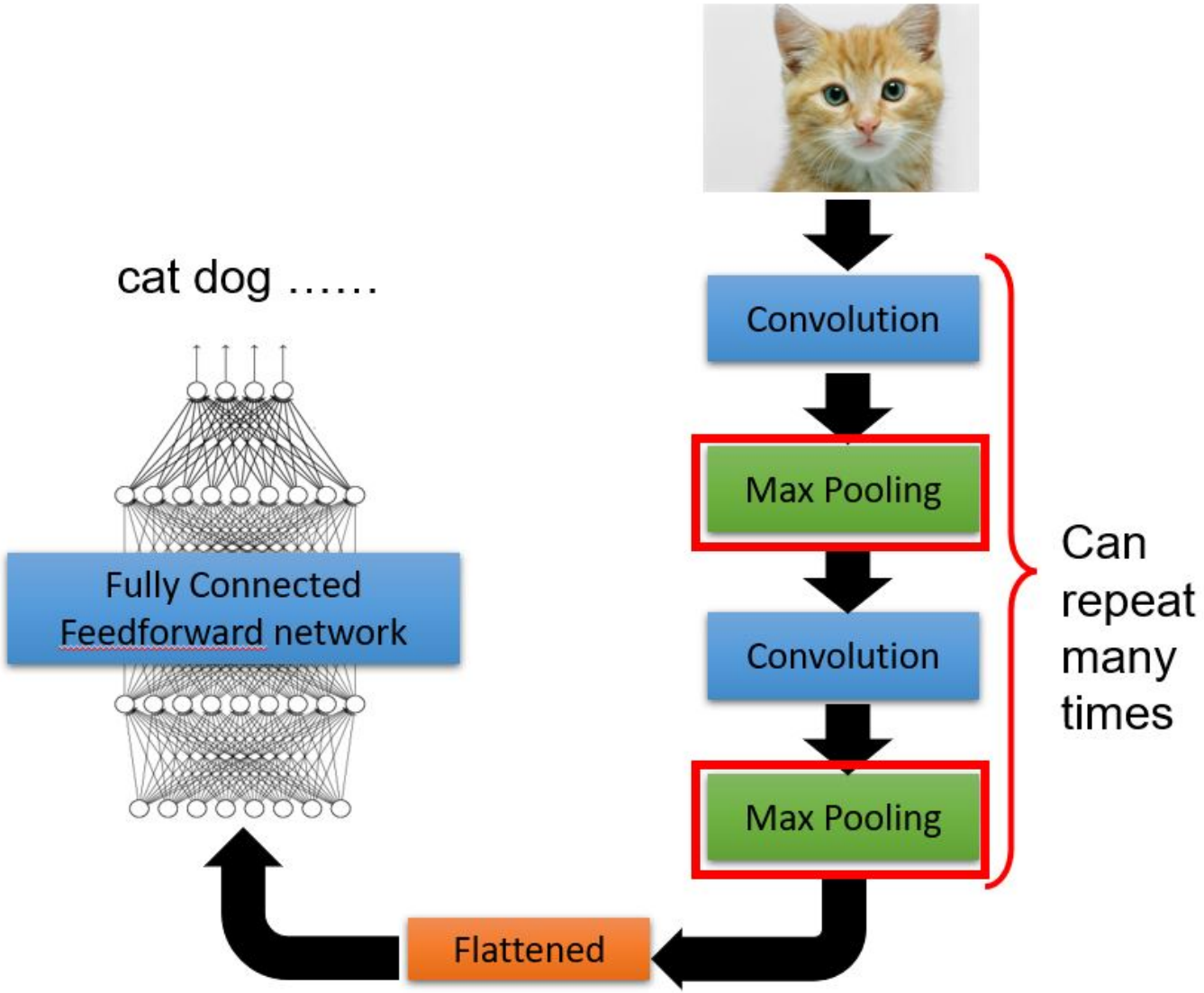


Architecture of CNN

A typical CNN has 4 layers

- Input layer
- Convolution layer
- Pooling layer
- Fully connected layer

Architecture of CNN



Input layer

- Example input a 28 pixel by 28 pixel grayscale image
- Unlike FNN, we do not "flatten" the input to a 1D vector
  - Input is presented to network in 2D as 28 x 28 matrix

Convolution

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

6 x 6 image

These are the network parameters to be learned.

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

Filter 1

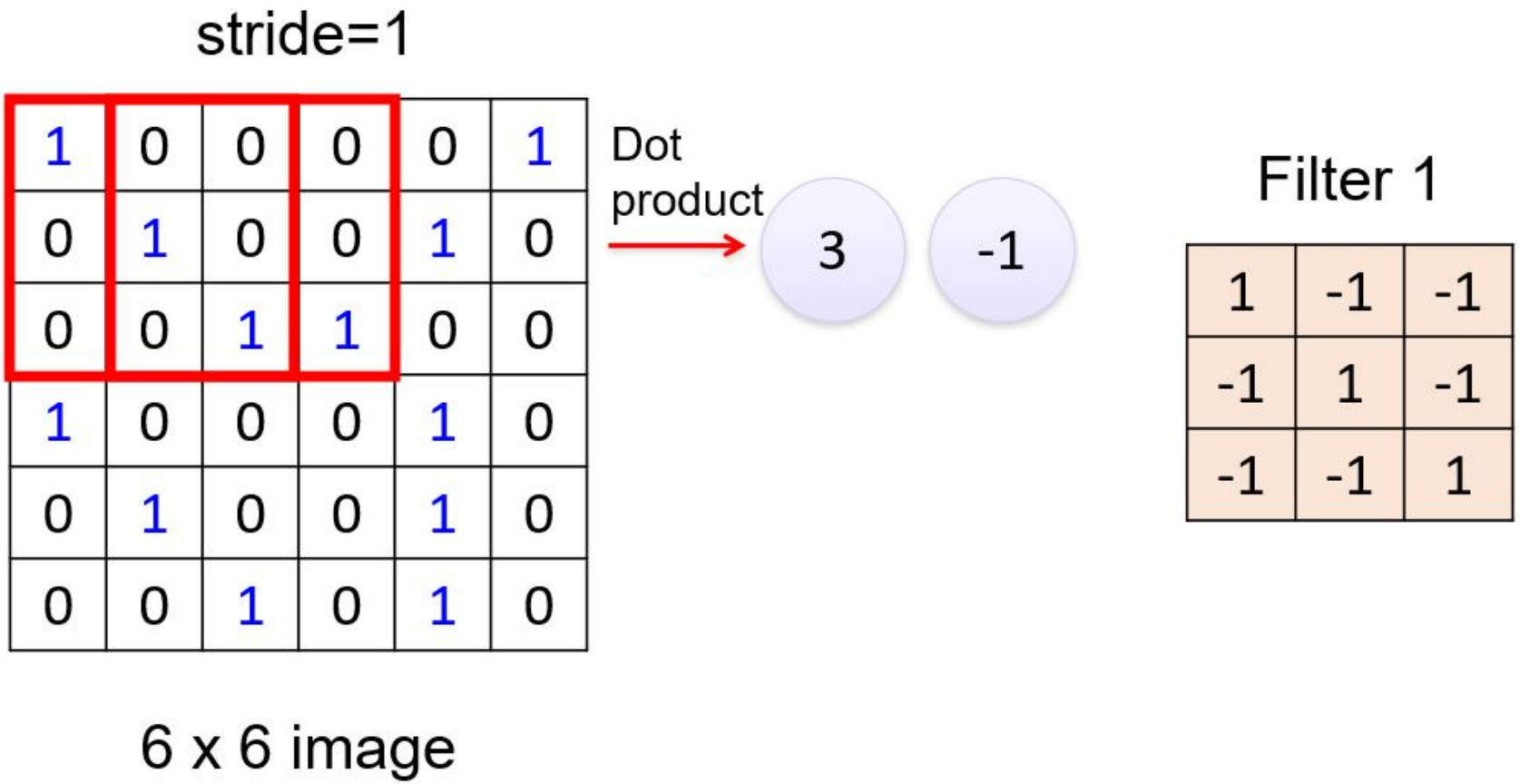
-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

⋮ ⋮

Each filter detects a small pattern (3 x 3).

Convolution



Convolution

If stride=2

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

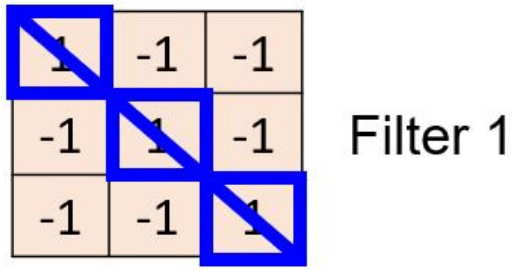
6 x 6 image

3 -3

Filter 1

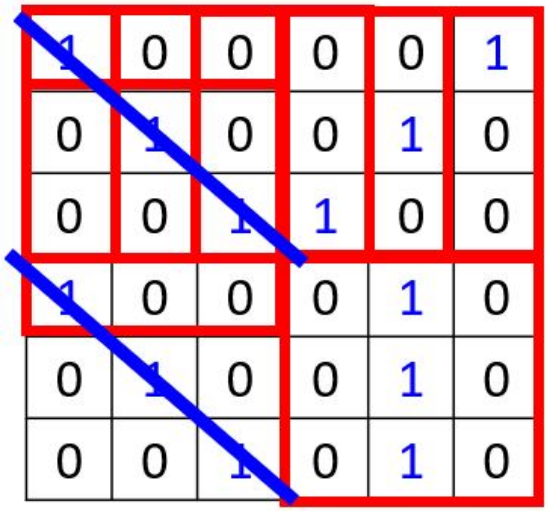
1	-1	-1
-1	1	-1
-1	-1	1

Convolution

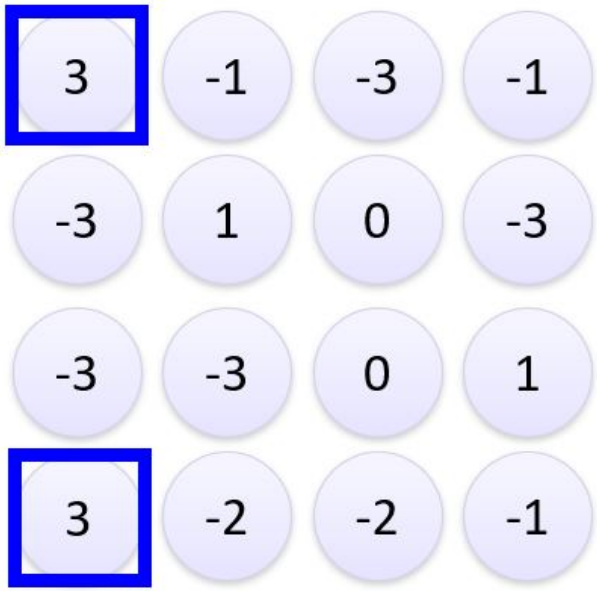


Filter 1

stride=1



6 x 6 image



Convolution

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

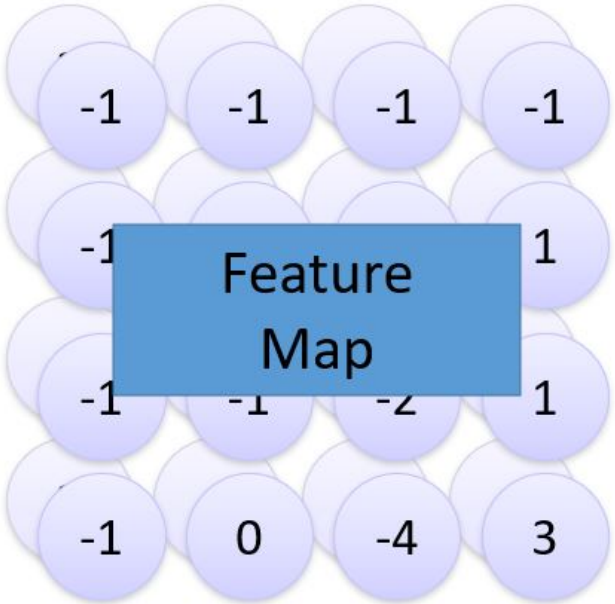
Filter 2

stride=1

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

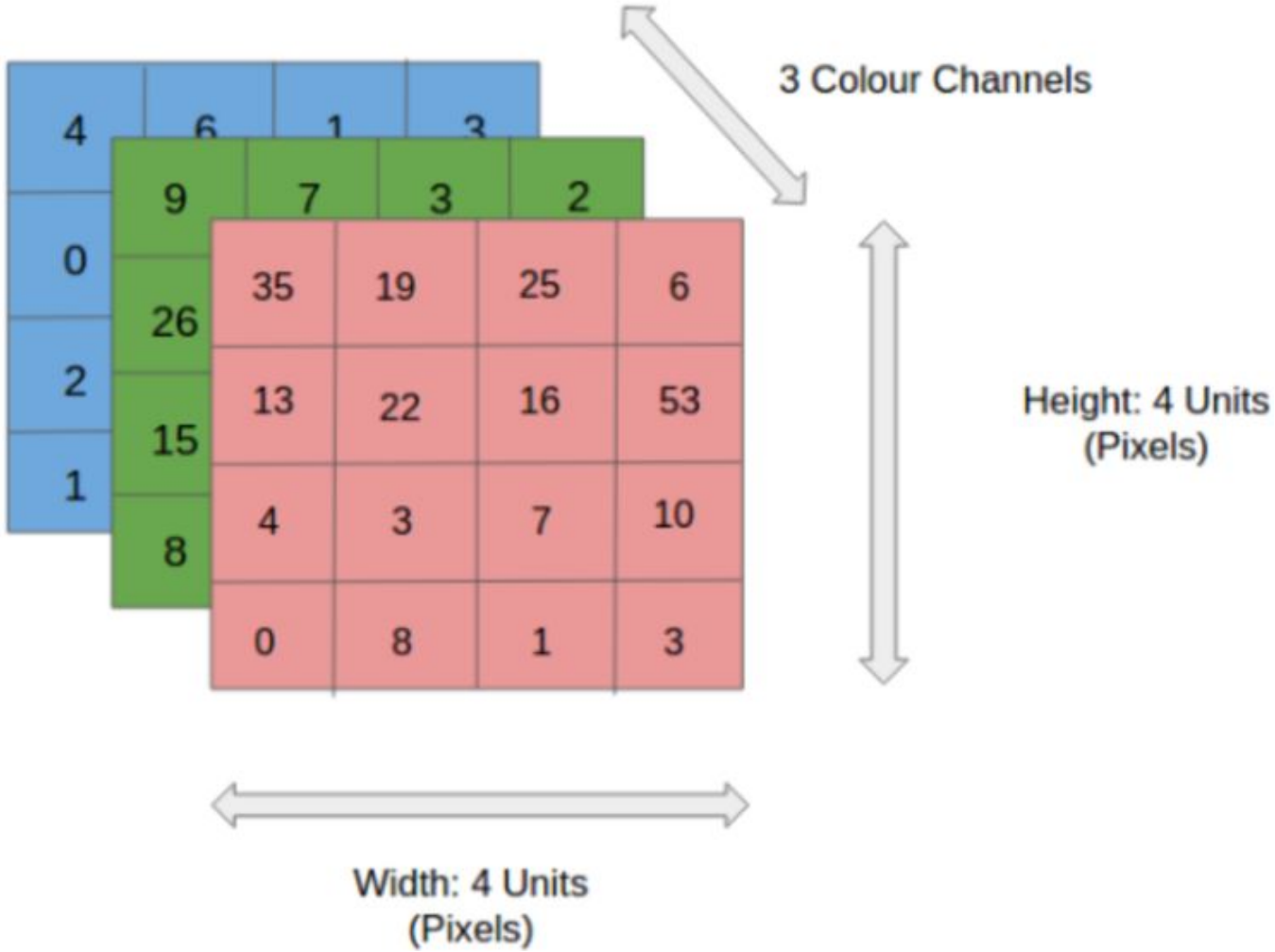
6 x 6 image

Repeat this for each filter

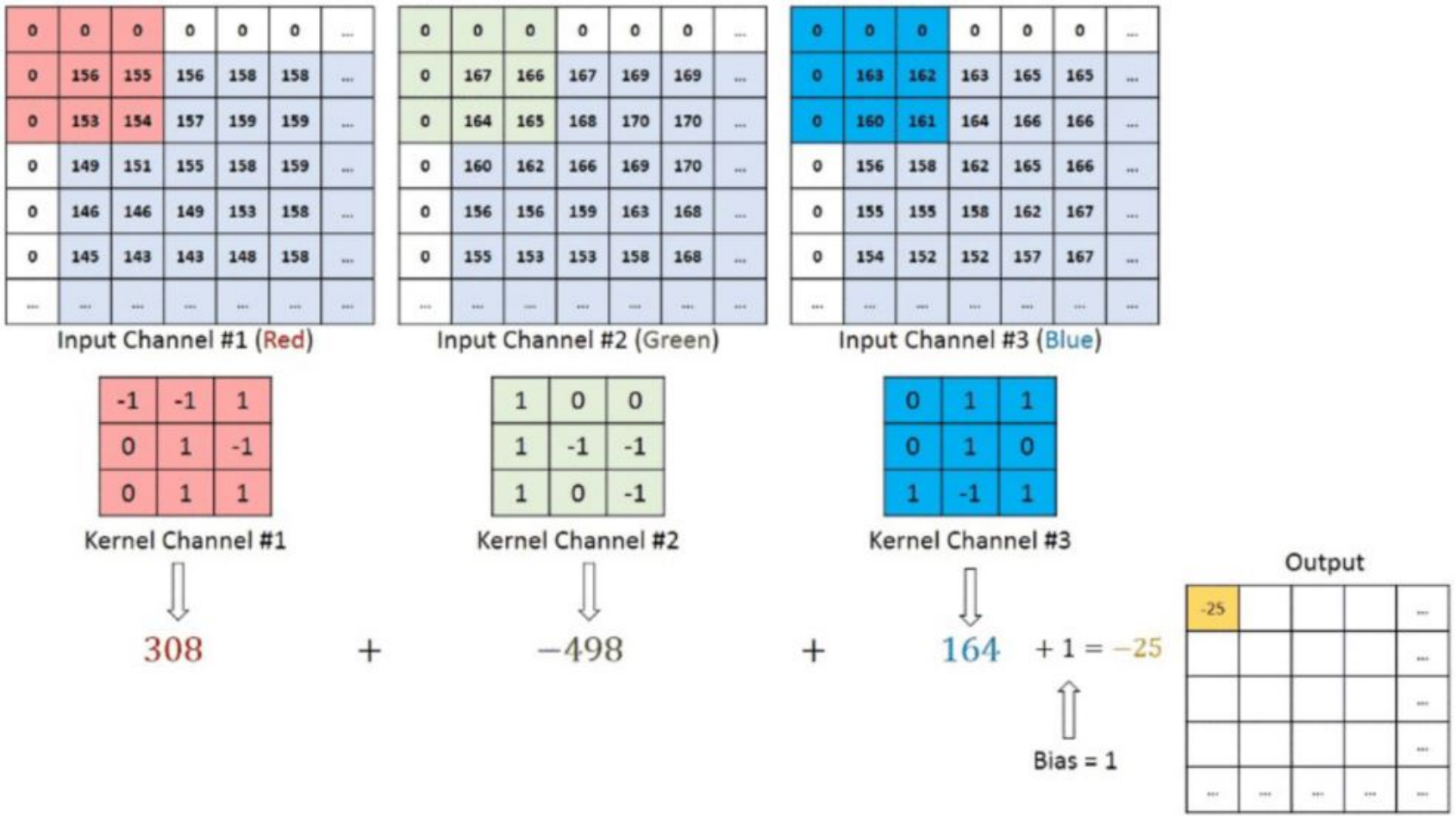


Convolution in RGB

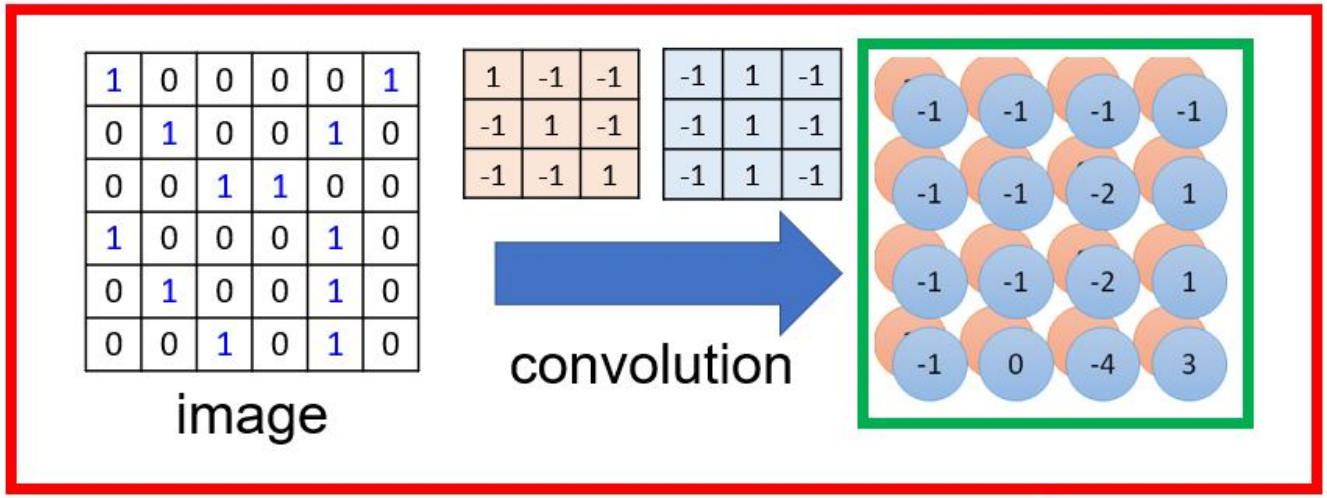
An RGB image is of the form

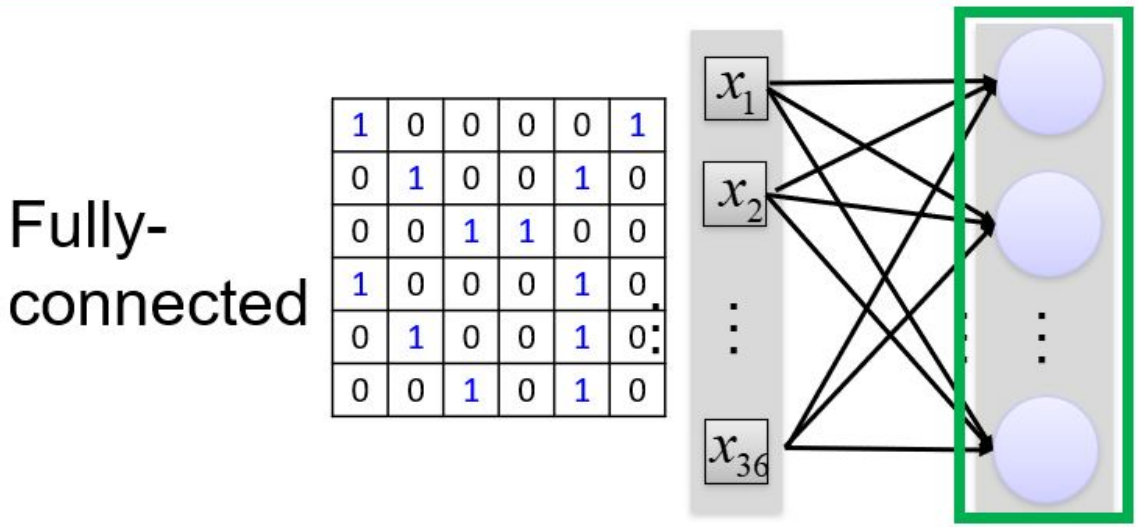


Convolution in RGB

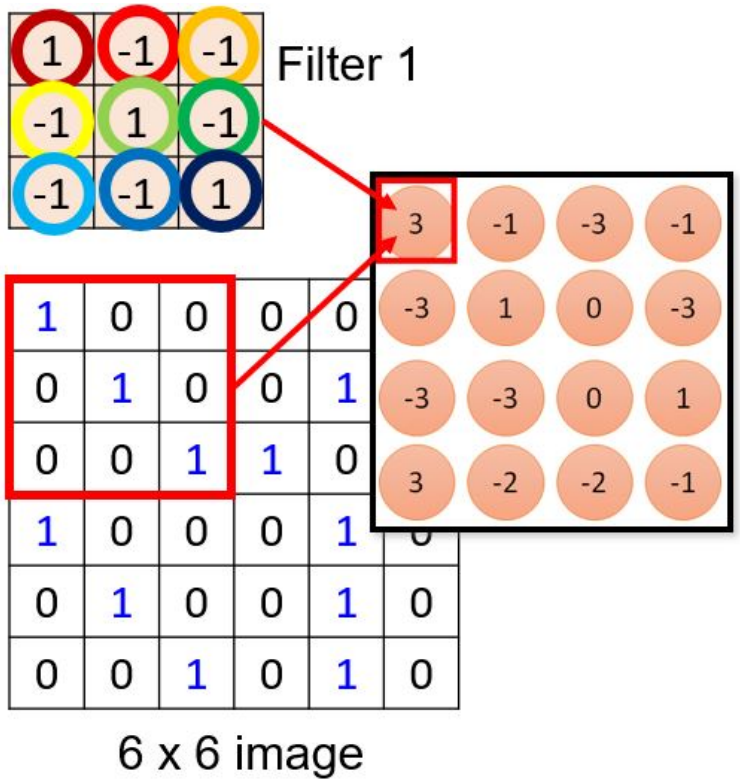


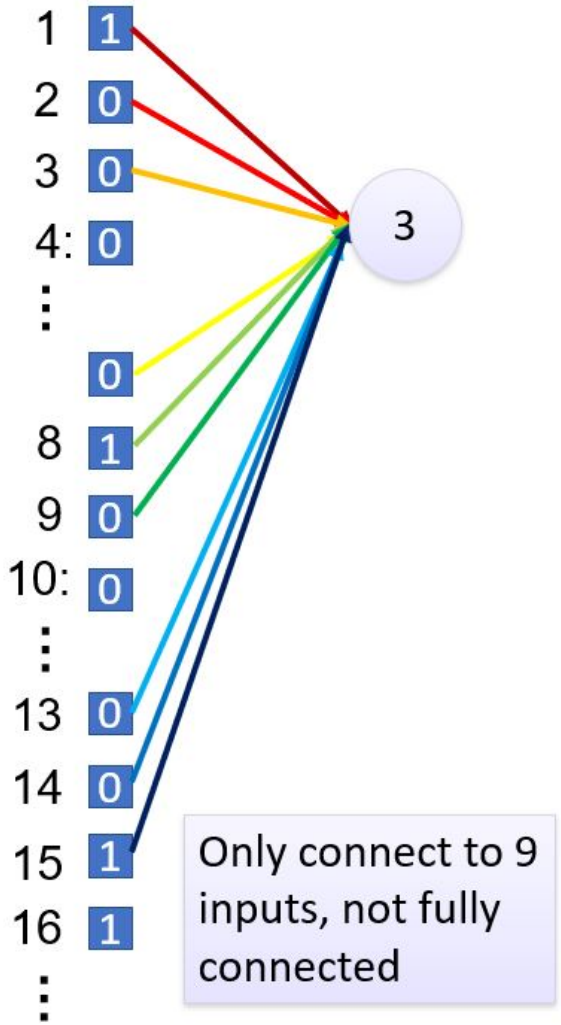
Convolution vs. Fully Connected NN



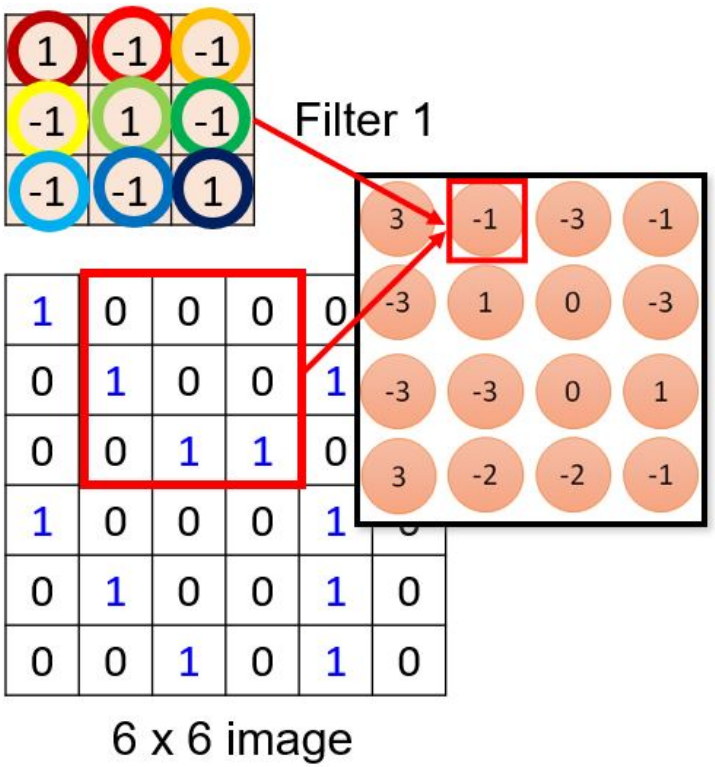


Convolution vs. Fully Connected NN



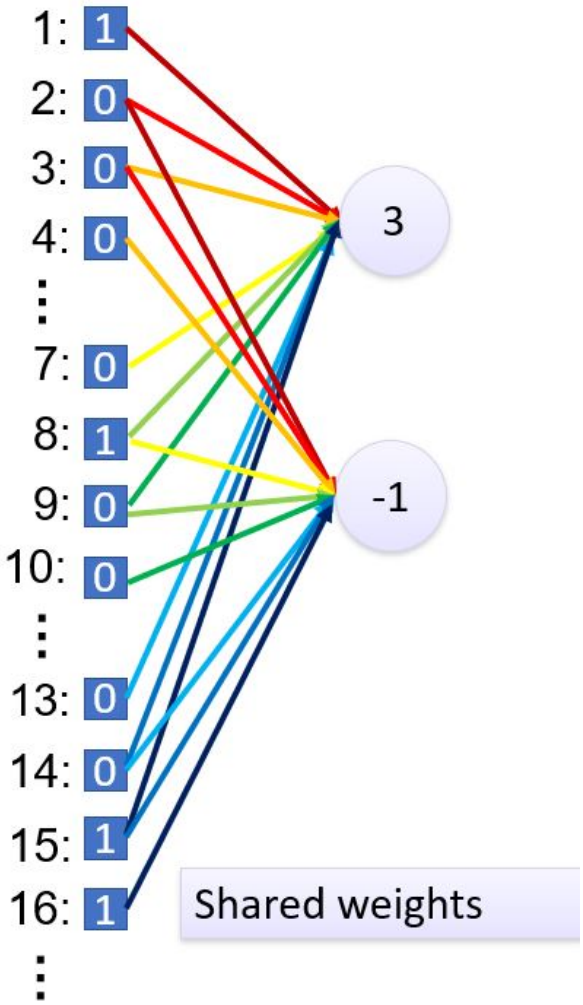


Convolution vs. Fully Connected NN



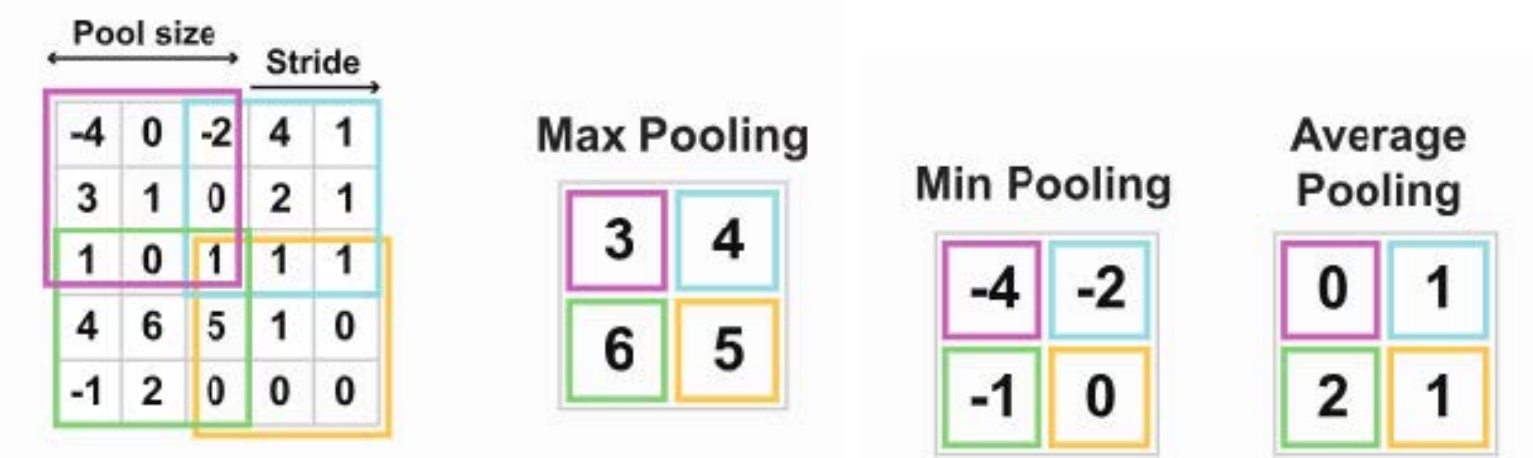
Fewer parameters

Even fewer parameters



Pooling

- In convolutional neural networks (CNNs), pooling is a down-sampling operation commonly used to reduce the spatial dimensions of the input volume.
- The most common type of pooling is called max pooling and average pooling.



Max pooling

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

Filter 1

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

Filter 2

3	-1
-3	1

-3	-1
0	-3

-3	-3
3	-2

0	1
-2	-1

-1	-1
-1	-1

-1	-1
-2	1

-1	-1
-1	0

-2	1
-4	3

Why pooling

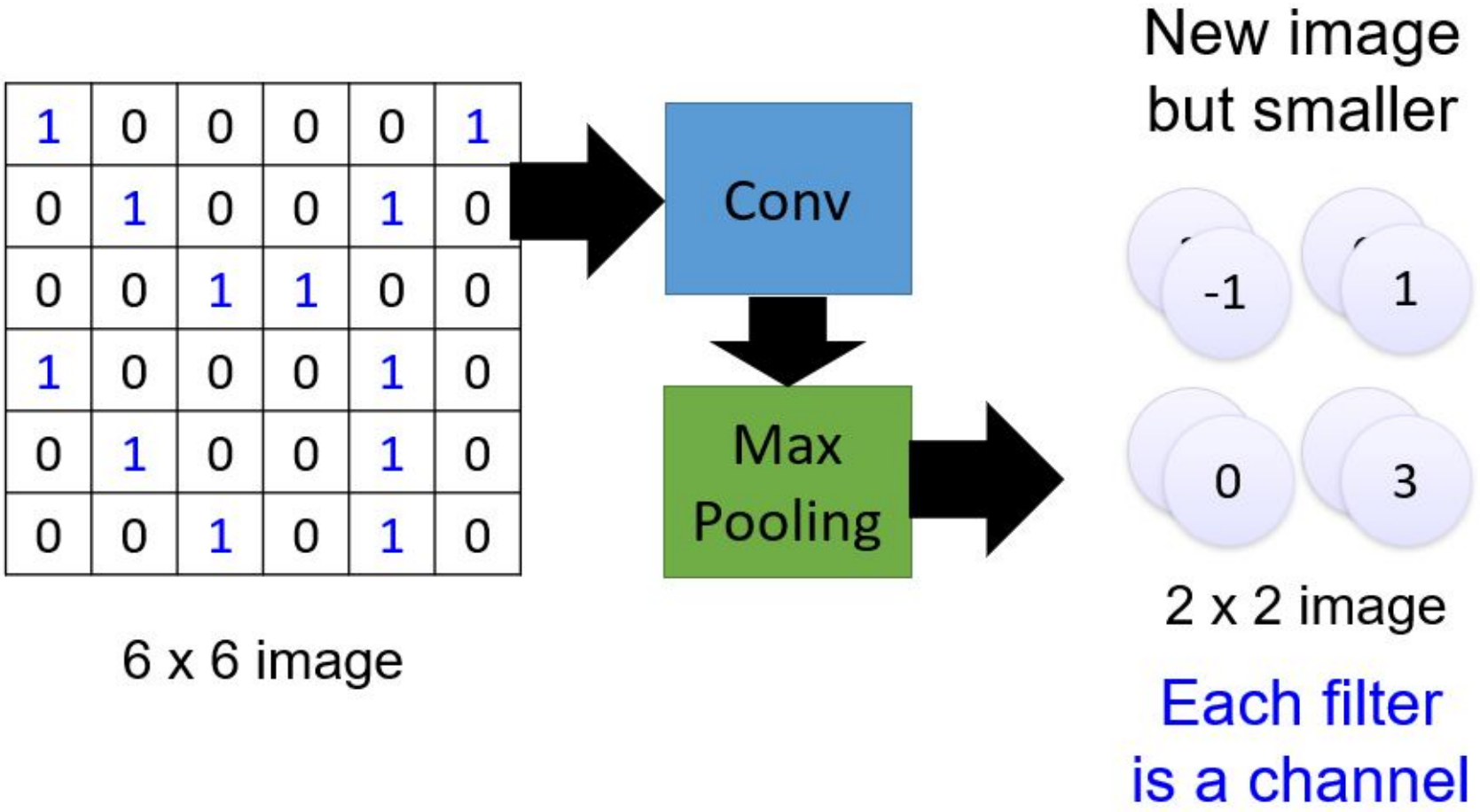
- Subsampling pixels will not change the object
- We can subsample the pixels to make image smaller
- fewer parameters to characterize the image

Max pooling

A CNN compresses a fully connected network in two ways:

- Reducing number of connections
- Shared weights on the edges
- Max pooling further reduces the complexity

Max pooling


1 0 0 0 0 1
0 1 0 0 1 0
0 0 1 1 0 0
1 0 0 0 1 0
0 1 0 0 1 0
0 0 1 0 1 0
6 x 6 image
Conv
Max Pooling
New image but smaller
-1
1
0
3
2 x 2 image
Each filter is a channel

The whole CNN

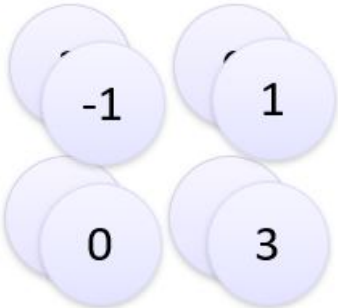
Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

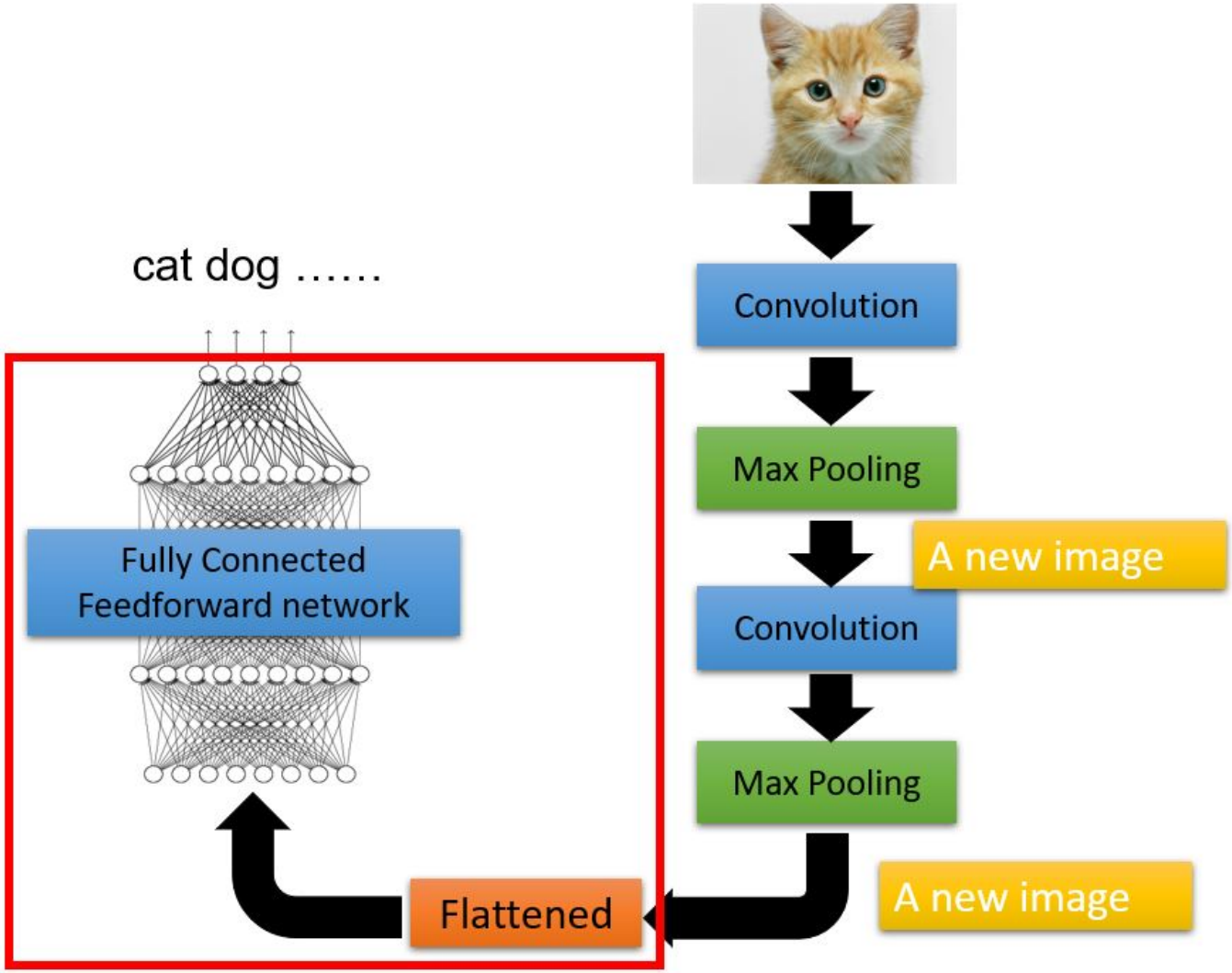


A new image

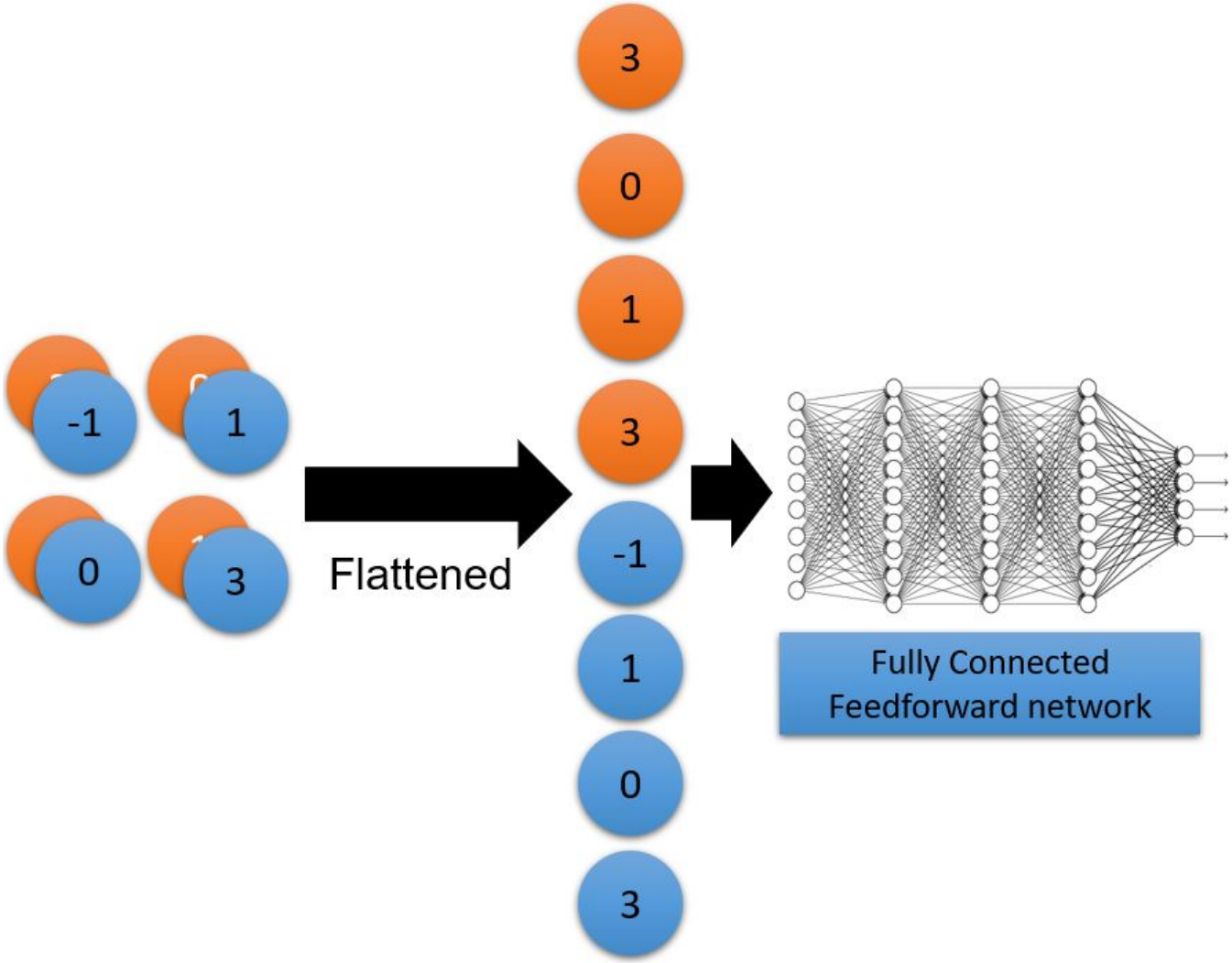
Smaller than the original image

The number of channels is the number of filters

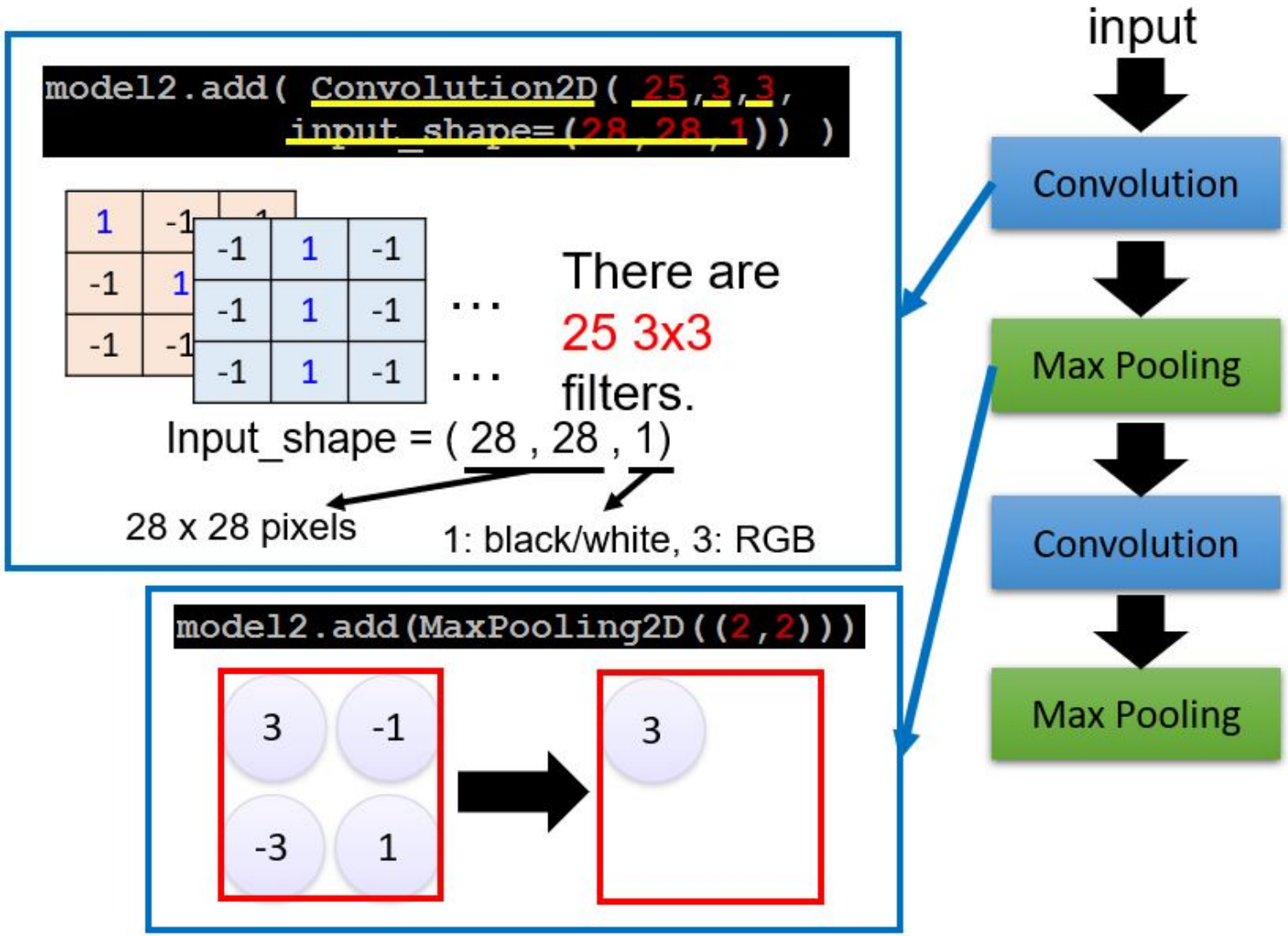
The whole CNN



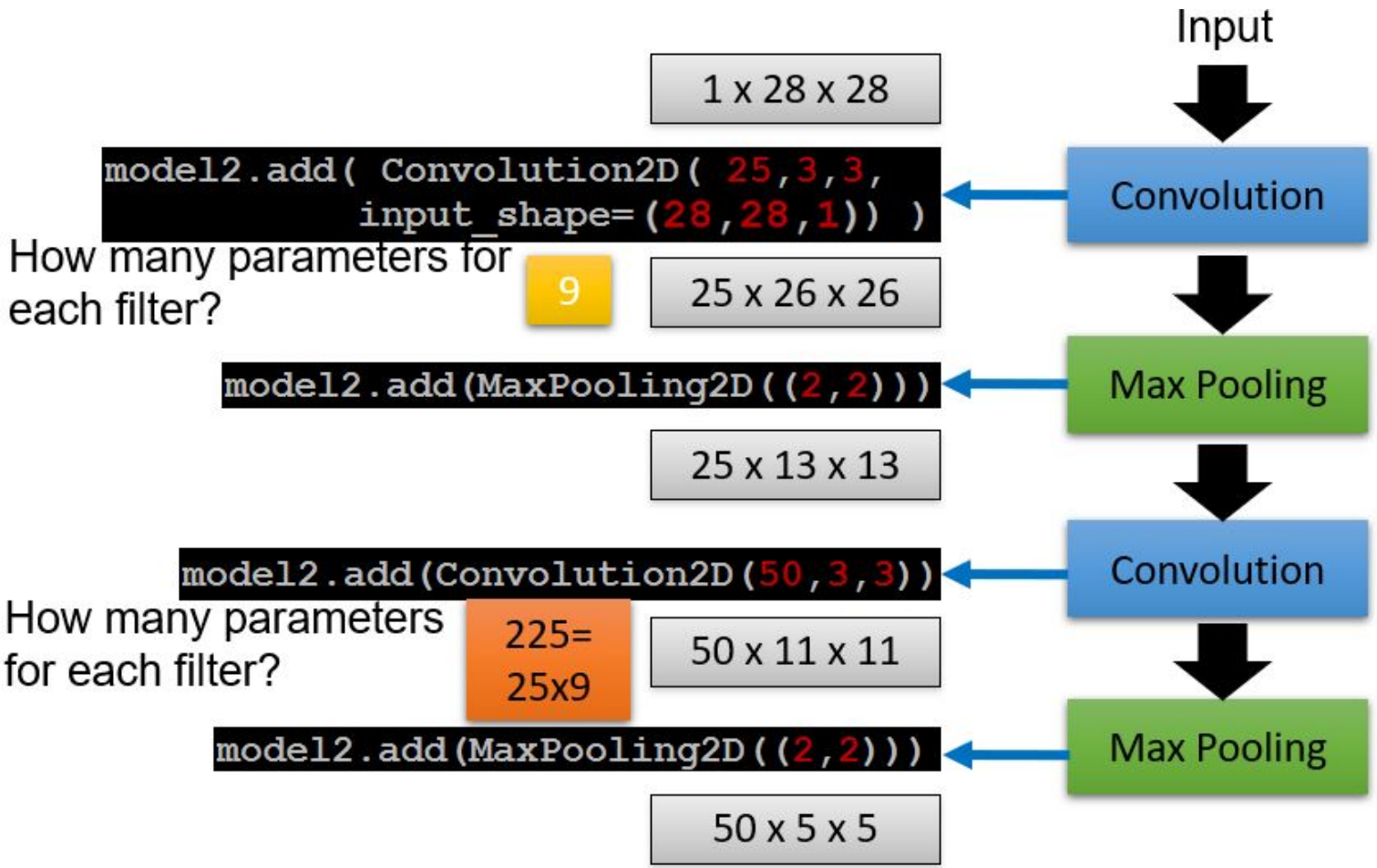
Flattening


-1
1
0
3
Flattened
3
0
1
3
-1
1
0
3
Fully Connected
Feedforward network

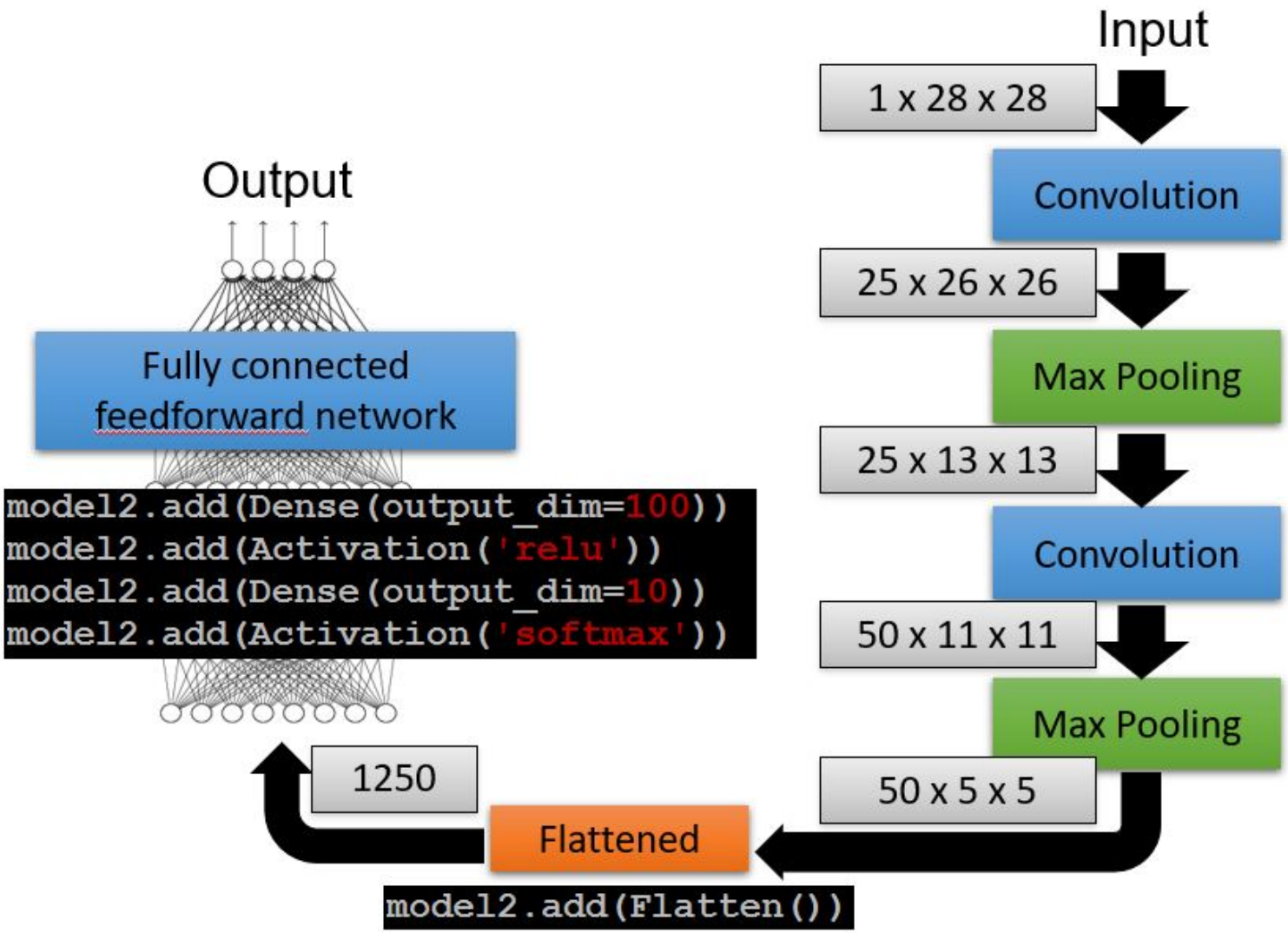
CNN in Keras


model2.add(Convolution2D(25,3,3,
input_shape=(28,28,1)))
There are
25 3x3
filters.
Input_shape = (28 , 28 , 1)
28 x 28 pixels
1: black/white, 3: RGB
model2.add(MaxPooling2D((2,2)))
3
-1
-3
1
3
input
Convolution
Max Pooling
Convolution
Max Pooling

CNN in Keras


Input
1 x 28 x 28
model2.add(Convolution2D(25,3,3, input_shape=(28,28,1)))
Convolution
How many parameters for each filter?
9
25 x 26 x 26
model2.add(MaxPooling2D((2,2)))
Max Pooling
25 x 13 x 13
model2.add(Convolution2D(50,3,3))
Convolution
How many parameters for each filter?
225= 25x9
50 x 11 x 11
model2.add(MaxPooling2D((2,2)))
Max Pooling
50 x 5 x 5

CNN in Keras


Input
1 x 28 x 28
Convolution
25 x 26 x 26
Max Pooling
25 x 13 x 13
Convolution
50 x 11 x 11
Max Pooling
50 x 5 x 5
Flattened
model2.add(Flatten())
1250
Fully connected
feedforward network
Output
model2.add(Dense(output_dim=100))
model2.add(Activation('relu'))
model2.add(Dense(output_dim=10))
model2.add(Activation('softmax'))