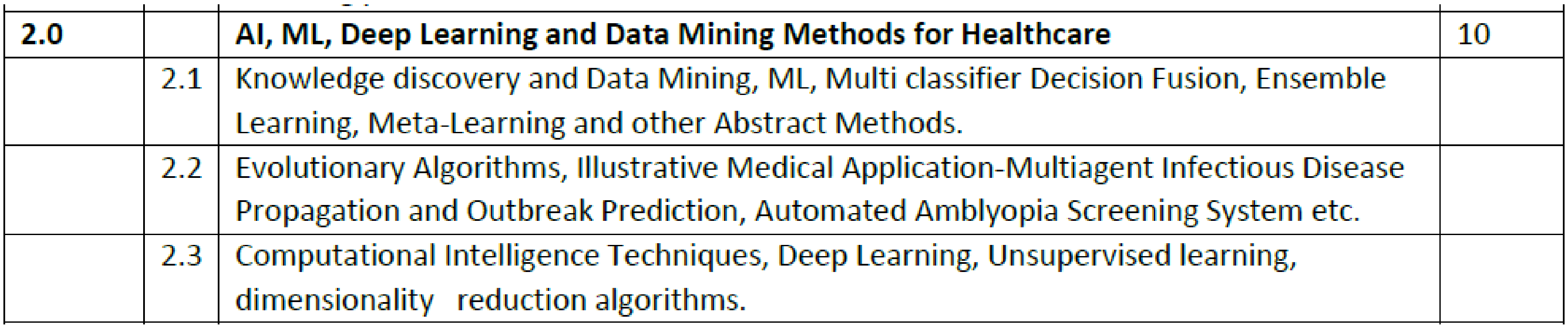
HAIMLC701 AI & ML in Healthcare

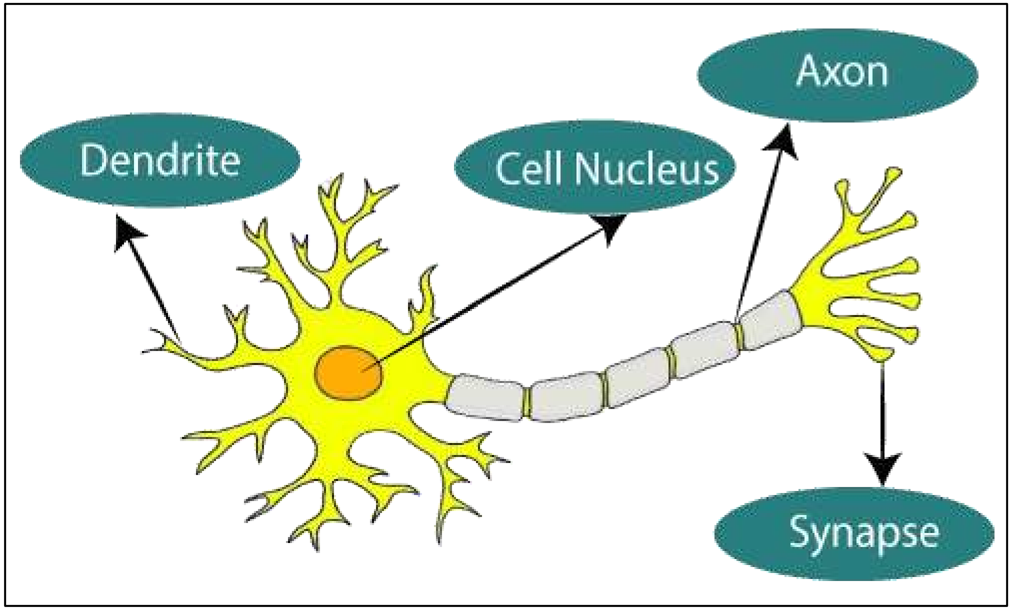


2.0		**AI, ML, Deep Learning and Data Mining Methods for Healthcare**	10
	2.1	Knowledge discovery and Data Mining, ML, Multi classifier Decision Fusion, Ensemble Learning, Meta-Learning and other Abstract Methods.	
	2.2	Evolutionary Algorithms, Illustrative Medical Application-Multiagent Infectious Disease Propagation and Outbreak Prediction, Automated Amblyopia Screening System etc.	
	2.3	Computational Intelligence Techniques, Deep Learning, Unsupervised learning, dimensionality reduction algorithms.	

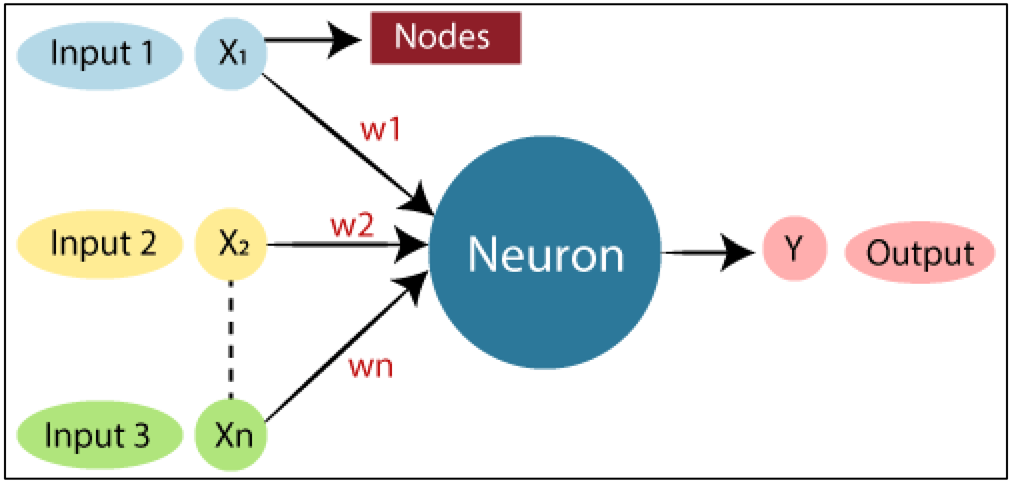
What is Deep learning?

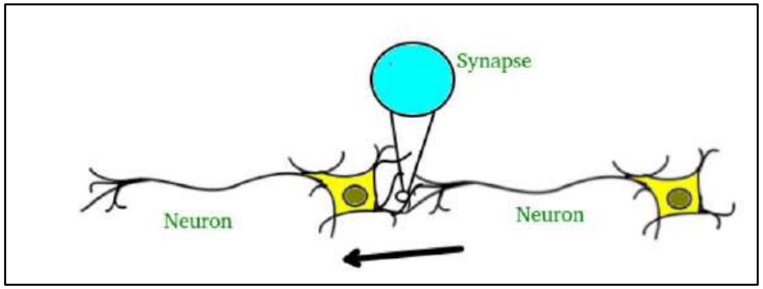
- Subfield of machine learning focuses on learning data representations as successive layers of increasingly meaningful representations
- **What are Neural Networks?**
- Used to mimic the basic functioning of the human brain and are inspired by how the human brain interprets information
- Used to solve various real-time tasks because of its ability to perform computations quickly and its fast responses

Biological Neuron vs Artificial Neuron

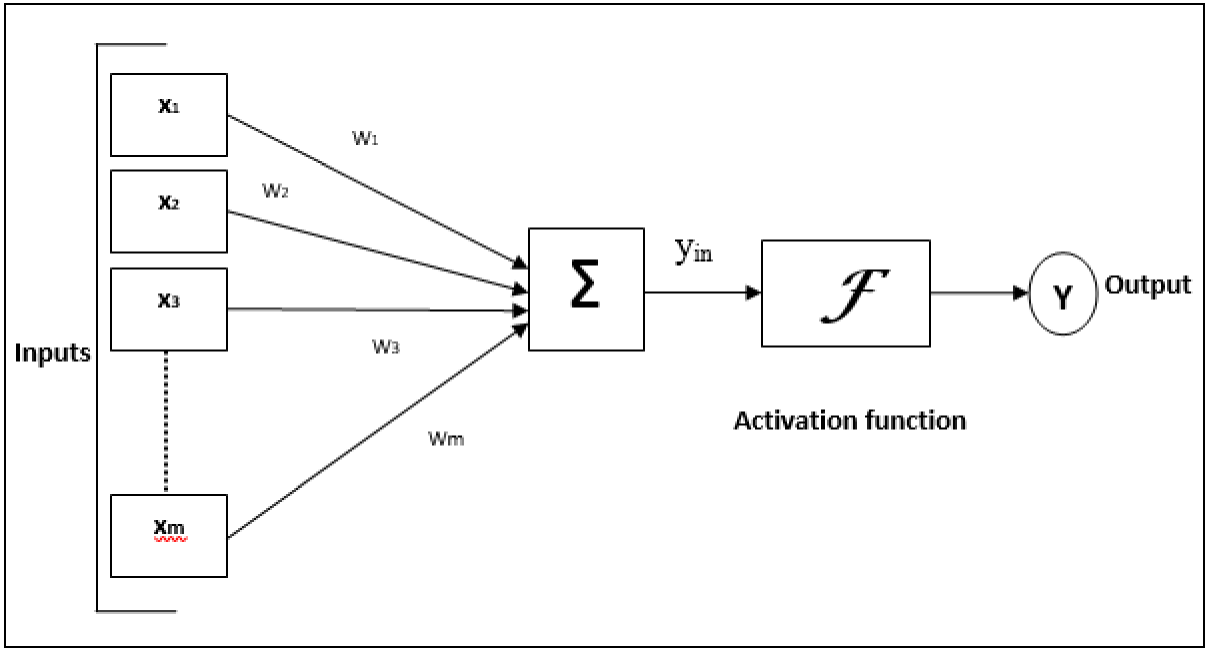


Dendrites –Inputs
cell nucleus – Nodes
synapse –Weights
Axon -Output





ANN



- For the above general model of ANN, the net input can be calculated as

$$y_{in} = x_1 . w_1 + x_2 . w_2 + x_3 . w_3 \dots x_m . w_m$$

i.e., Net input $y_{in} = \sum_i^m x_i . w_i$

The output can be calculated by applying the activation function over the net input.

$$Y = F(y_{in})$$

Types of Neural Networks

- **ANN-** Artificial neural network
  - It is a feed-forward neural network because the inputs are sent in the forward direction
  - It can also contain hidden layers which can make the model even denser
  - They have a fixed length as specified by the programmer
  - It is used for Textual Data or Tabular Data
  - A widely used real-life application is Facial Recognition
  - It is comparatively less powerful than CNN and RNN

Types of Neural Networks

- **CNN–** Convolutional Neural Networks
  - Mainly used for Image Data in Computer Vision
  - Some of the real-life applications are object detection in autonomous vehicles
  - Contains a combination of convolutional layers and neurons
  - More powerful than both ANN and RNN

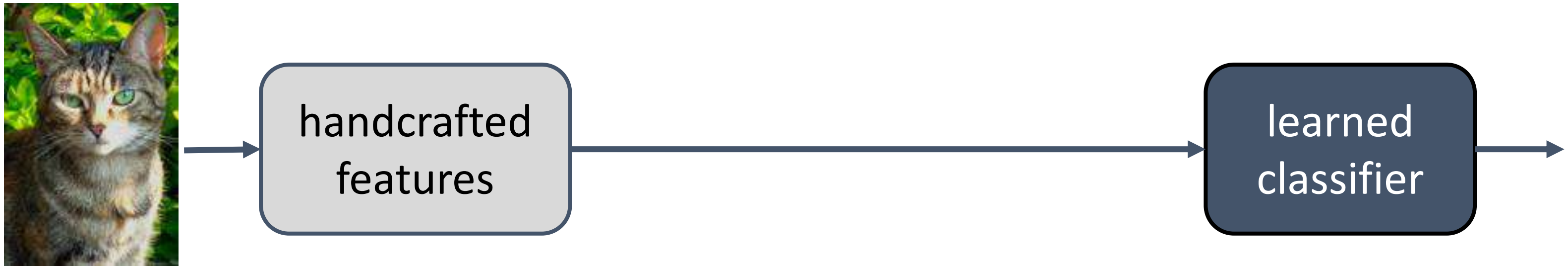
Types of Neural Networks

- **RNN–**Recurrent Neural Networks
  - It is used to process and interpret time series data
  - In this type of model, the output from a processing node is fed back into nodes in the same or previous layers
  - The most known types of RNN are **LSTM** (Long Short Term Memory) Networks

Learning in ANN can be classified into three categories namely supervised learning, unsupervised learning, and reinforcement learning.

Machine Learning Vs Deep Learning

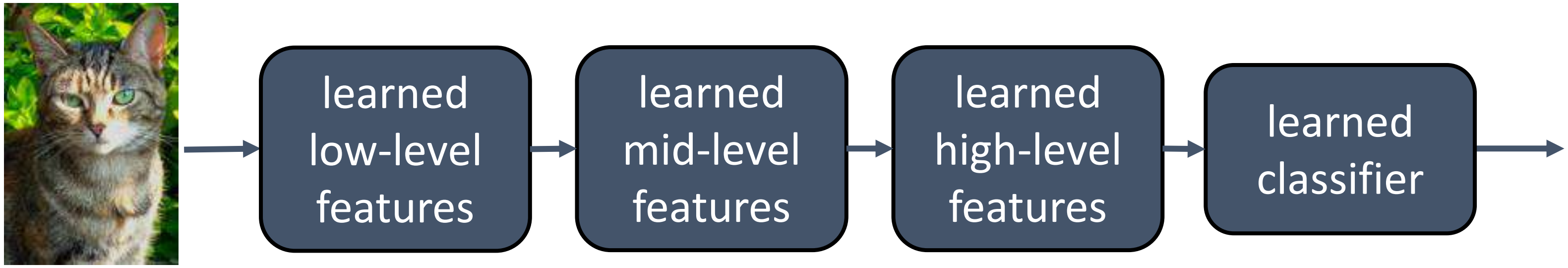
“Traditional” machine learning:



handcrafted features

learned classifier

Deep, “end-to-end” learning:



learned low-level features

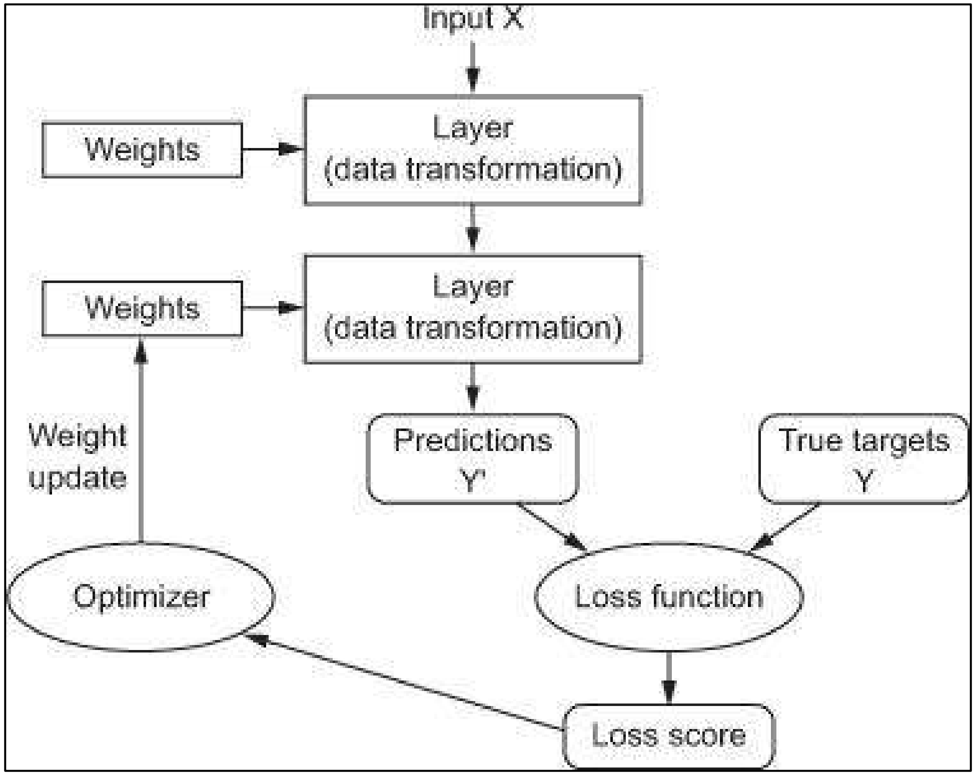
learned mid-level features

learned high-level features

learned classifier

Anatomy of a deep neural network

- Layers
- Input data and targets
- Loss function
- Optimizer

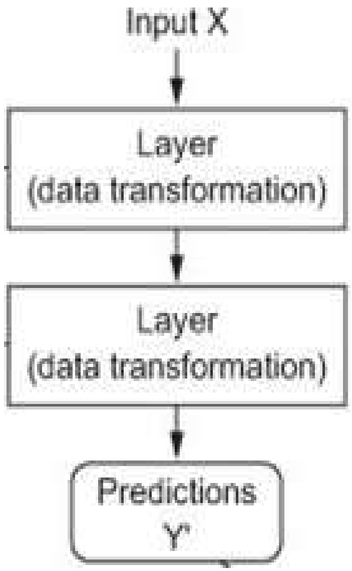


Layers

- Data processing modules
- Many different kinds exist
  - densely connected
  - convolutional
  - recurrent
  - pooling, flattening, merging, normalization, etc.
- Input: one or more tensors
  output: one or more tensors
- Usually have a state, encoded as **weights**
  - learned, initially random
- When combined, form a **network** or a **model**

Input data and targets

- The network maps the input data X to predictions Y’
- During training, the predictions Y’ are compared to true targets Y using the loss function

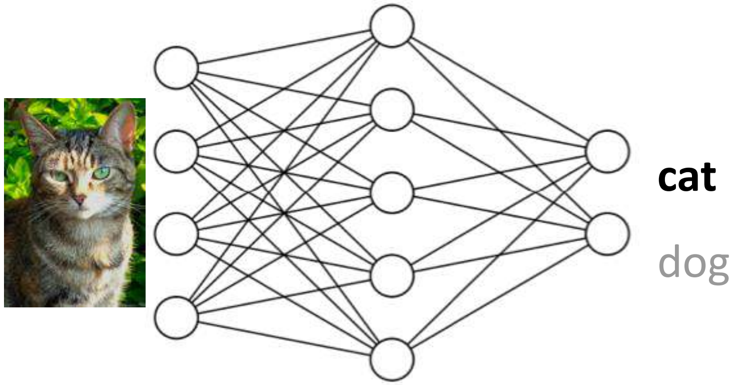


Input X

Layer
(data transformation)

Layer
(data transformation)

Predictions
Y'

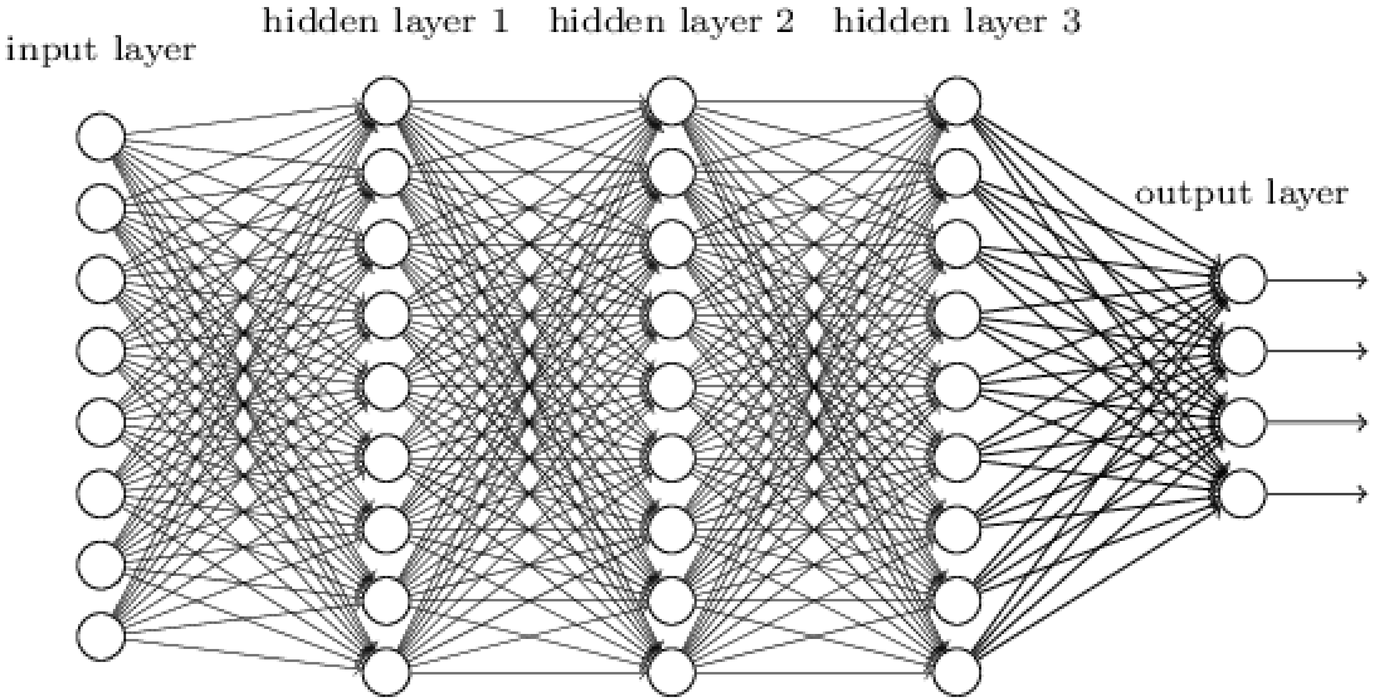


Loss function

- The quantity to be minimized (optimized) during training
  - the only thing **the network** cares about
  - there might also be other metrics **you** care about
- Common tasks have “standard” loss functions:
  - *mean squared error* for regression
  - *binary cross-entropy* for two-class classification
  - *categorical cross-entropy* for multi-class classification
  - etc.

CNN-Convolutional Neural Network

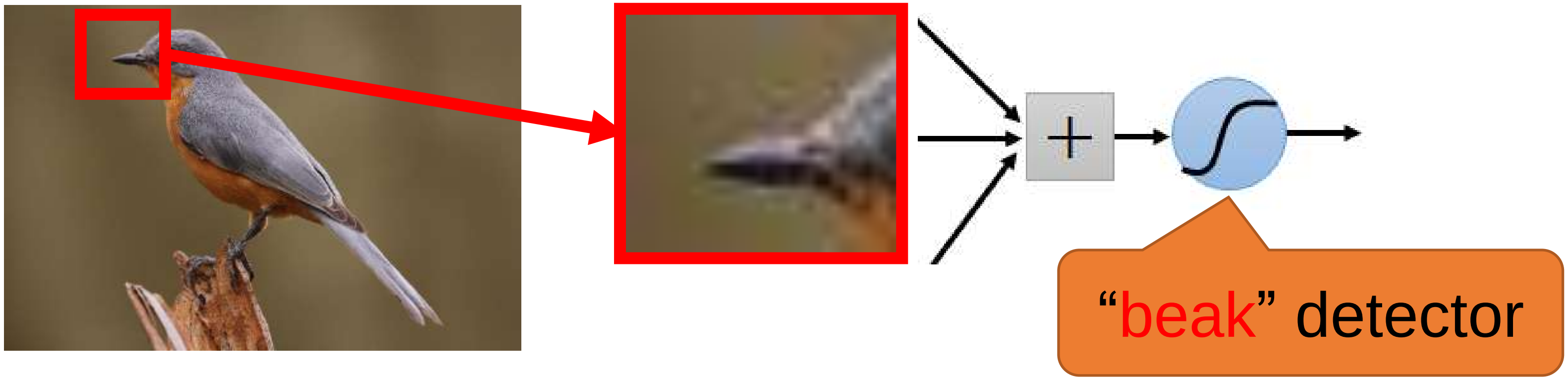
- Class of ANN most commonly applied to analyze visual imagery
- CNNs use a mathematical operation called convolution in place of general matrix multiplication in at least one of their layers



Consider learning an image:

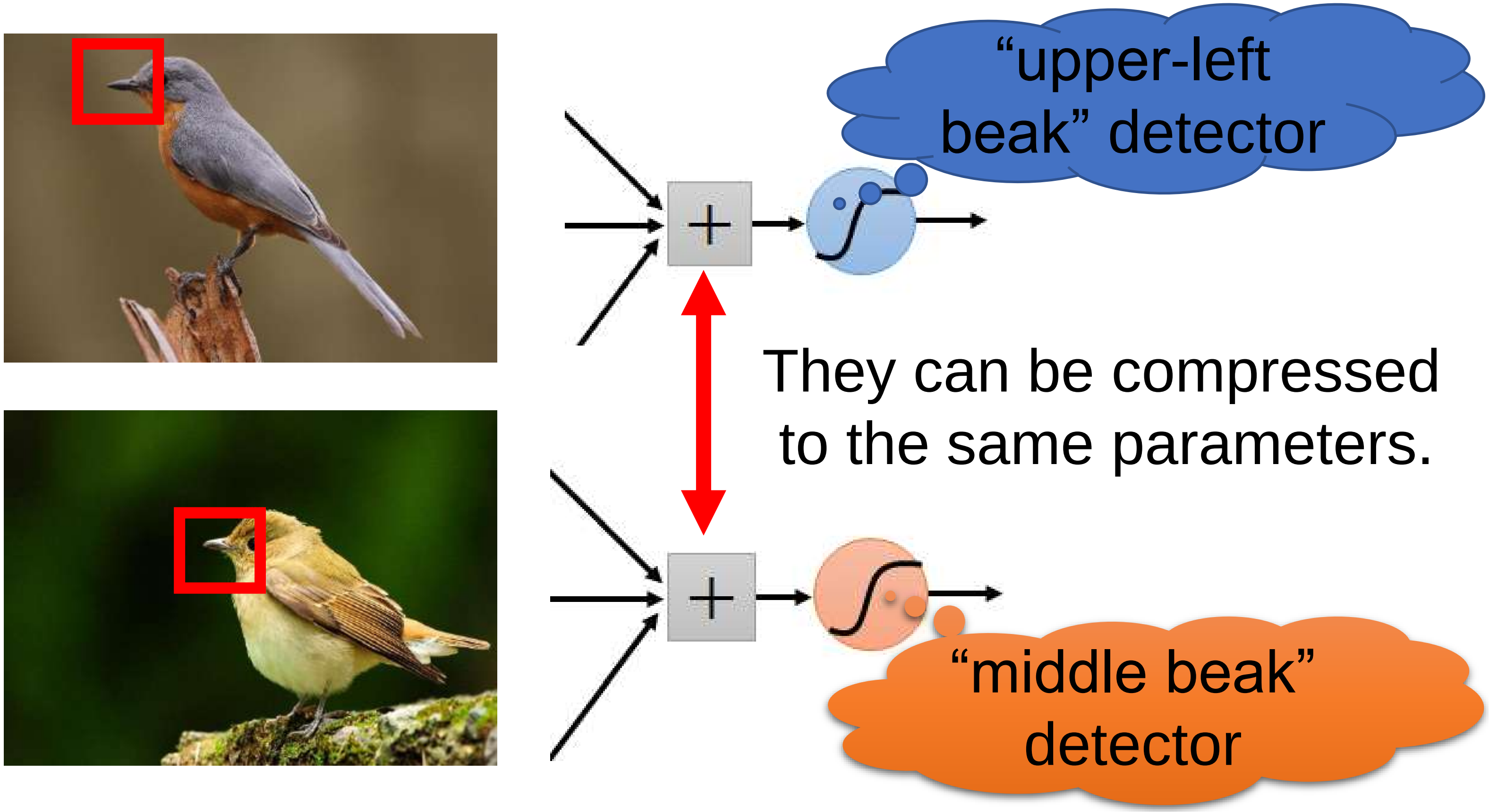
- Some patterns are much smaller than the whole image

Can represent a small region with fewer parameters



“beak” detector

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”.



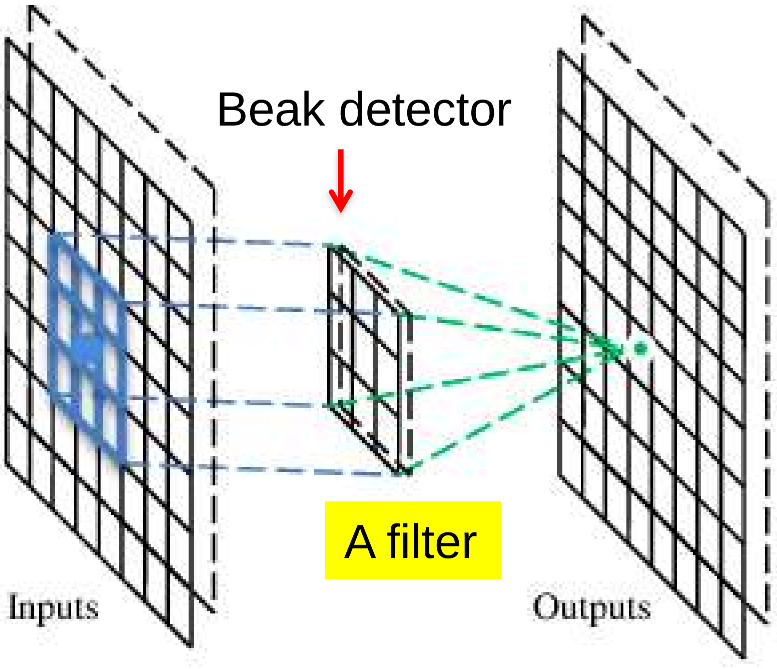
“upper-left beak” detector

They can be compressed to the same parameters.

“middle beak” detector

A convolutional layer

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



Convolution

These are the network parameters to be learned.

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

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

Filter 1

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

Filter 2

⋮ ⋮

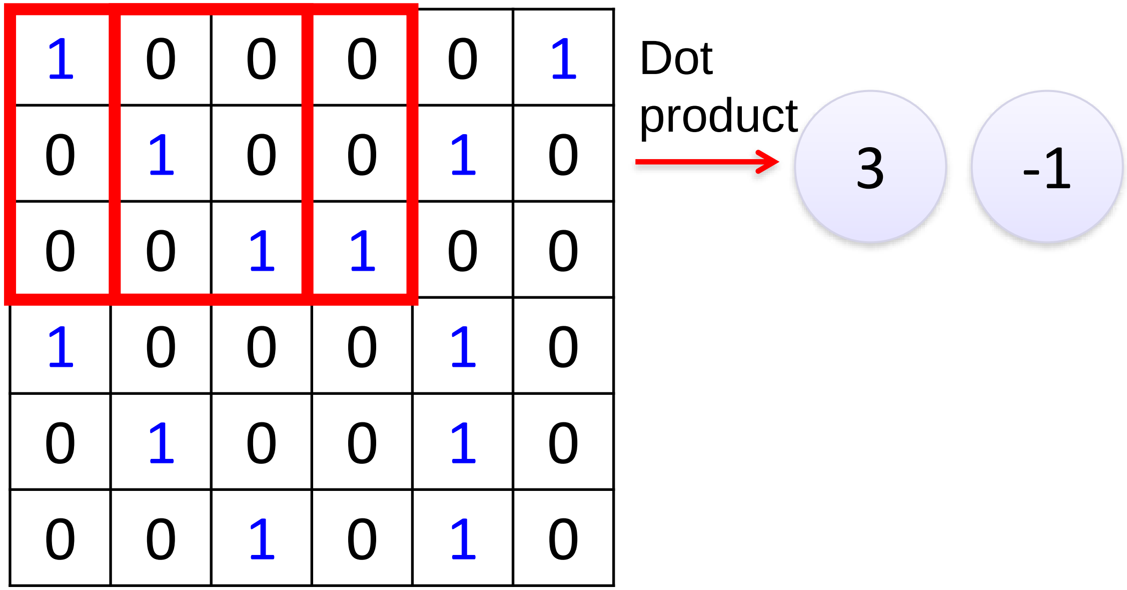
Each filter detects a small pattern (3 x 3).

Convolution

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

Filter 1

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

Convolution

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

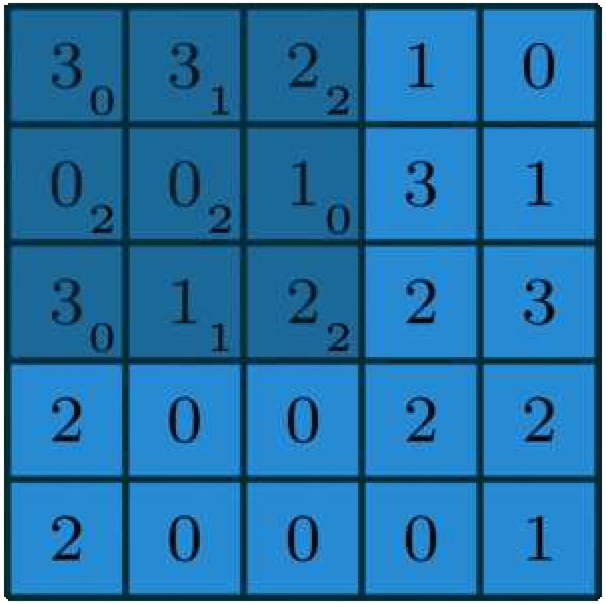
Filter 1

If stride=2

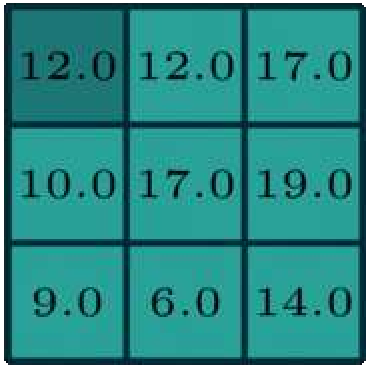
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

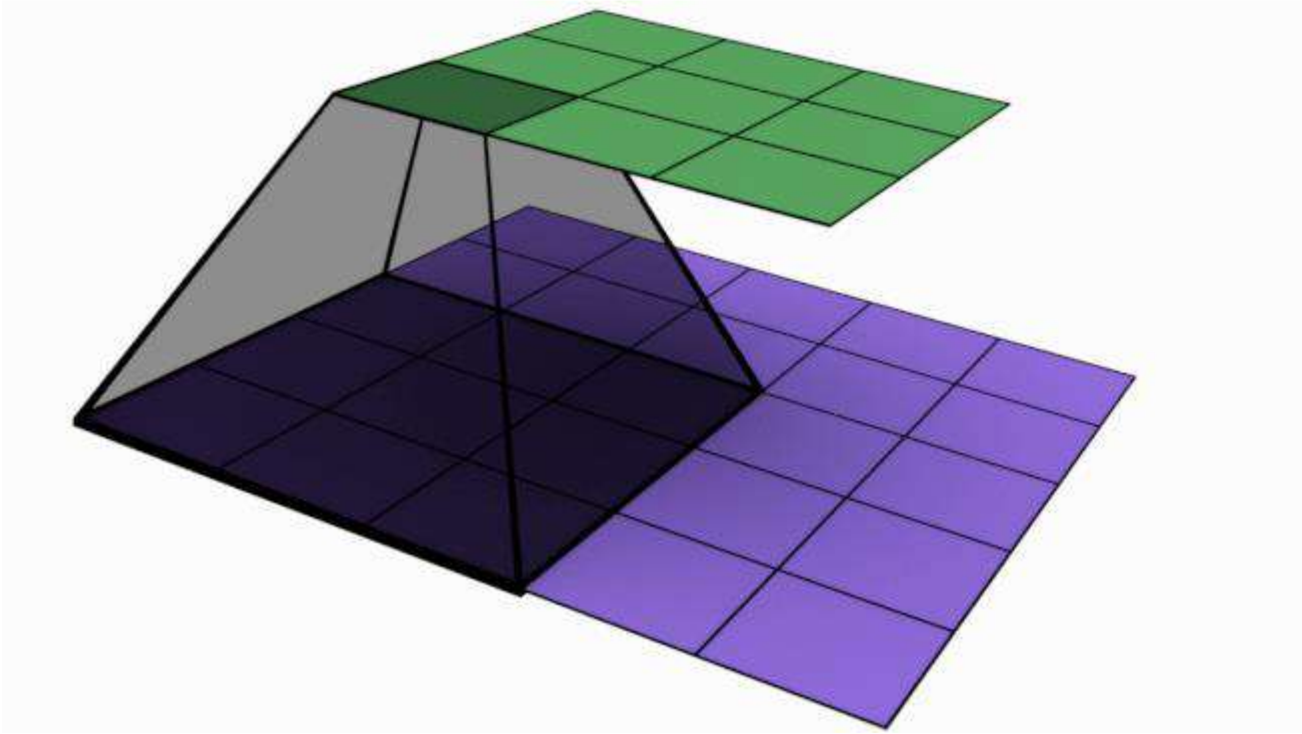
3 -3



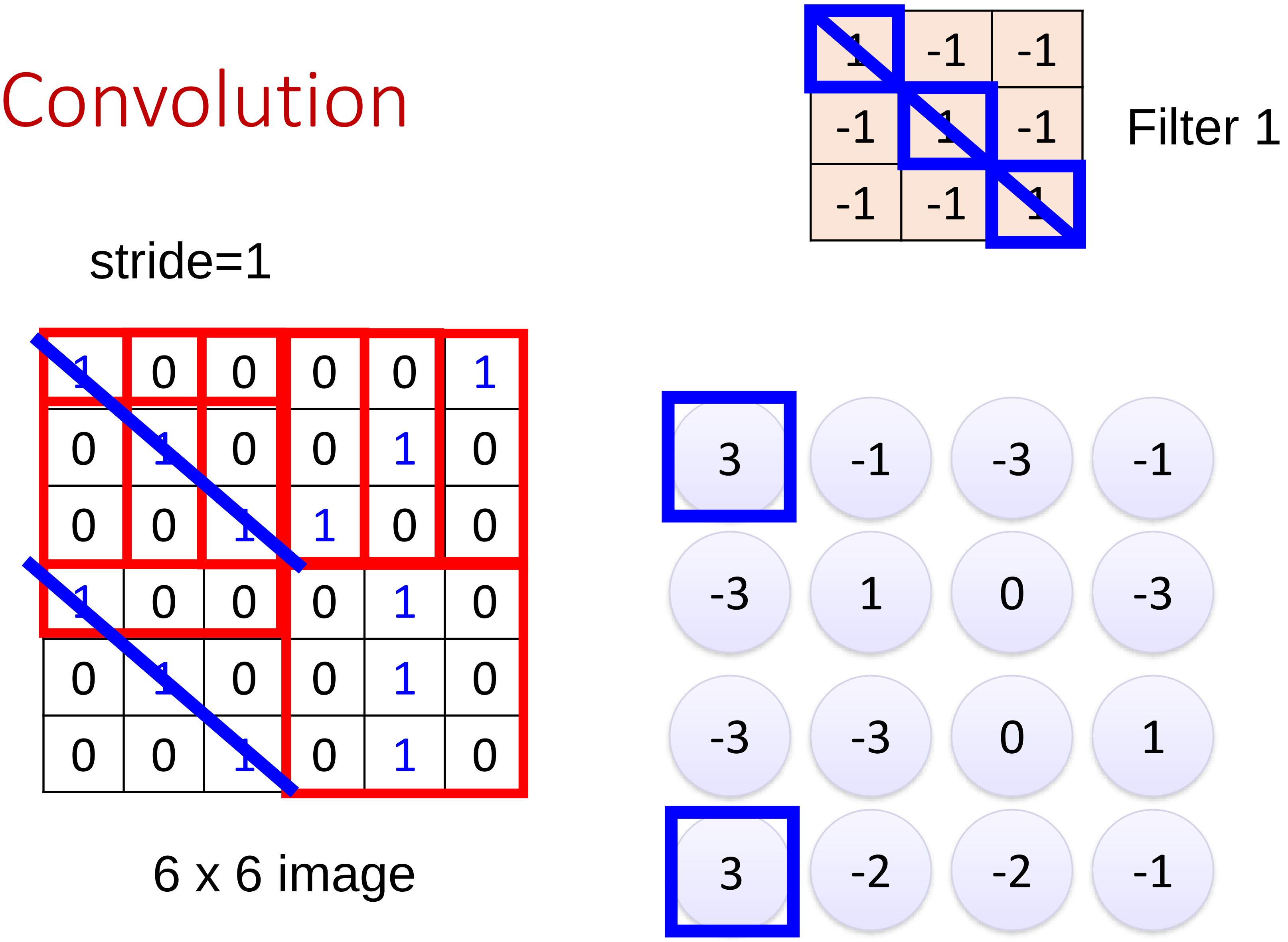
3_0	3_1	2_2	1	0
0_2	0_2	1_0	3	1
3_0	1_1	2_2	2	3
2	0	0	2	2
2	0	0	0	1



12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0



Convolution



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

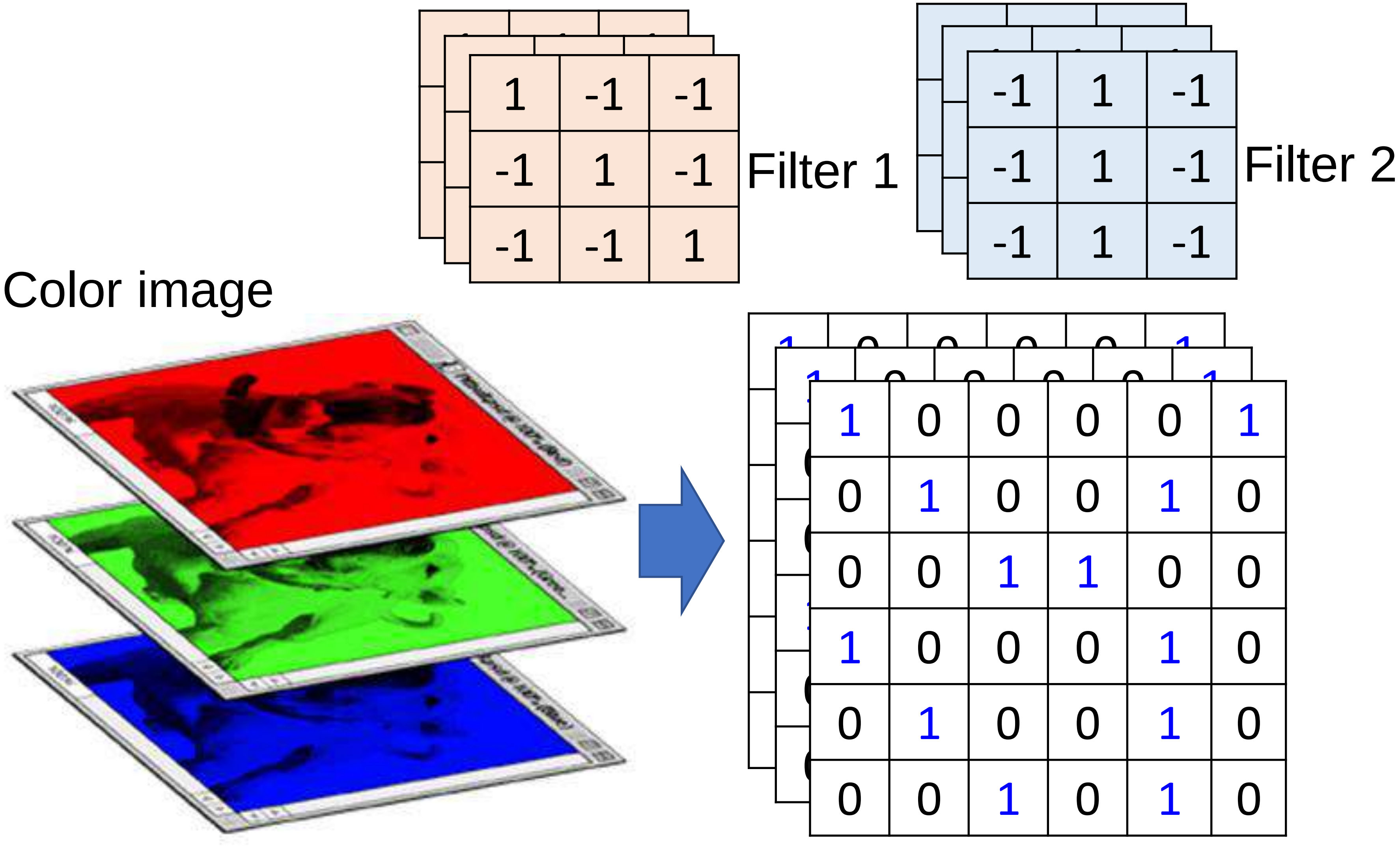
6 x 6 image

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

Filter 1

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

Color image: RGB 3 channels



Filter 1

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

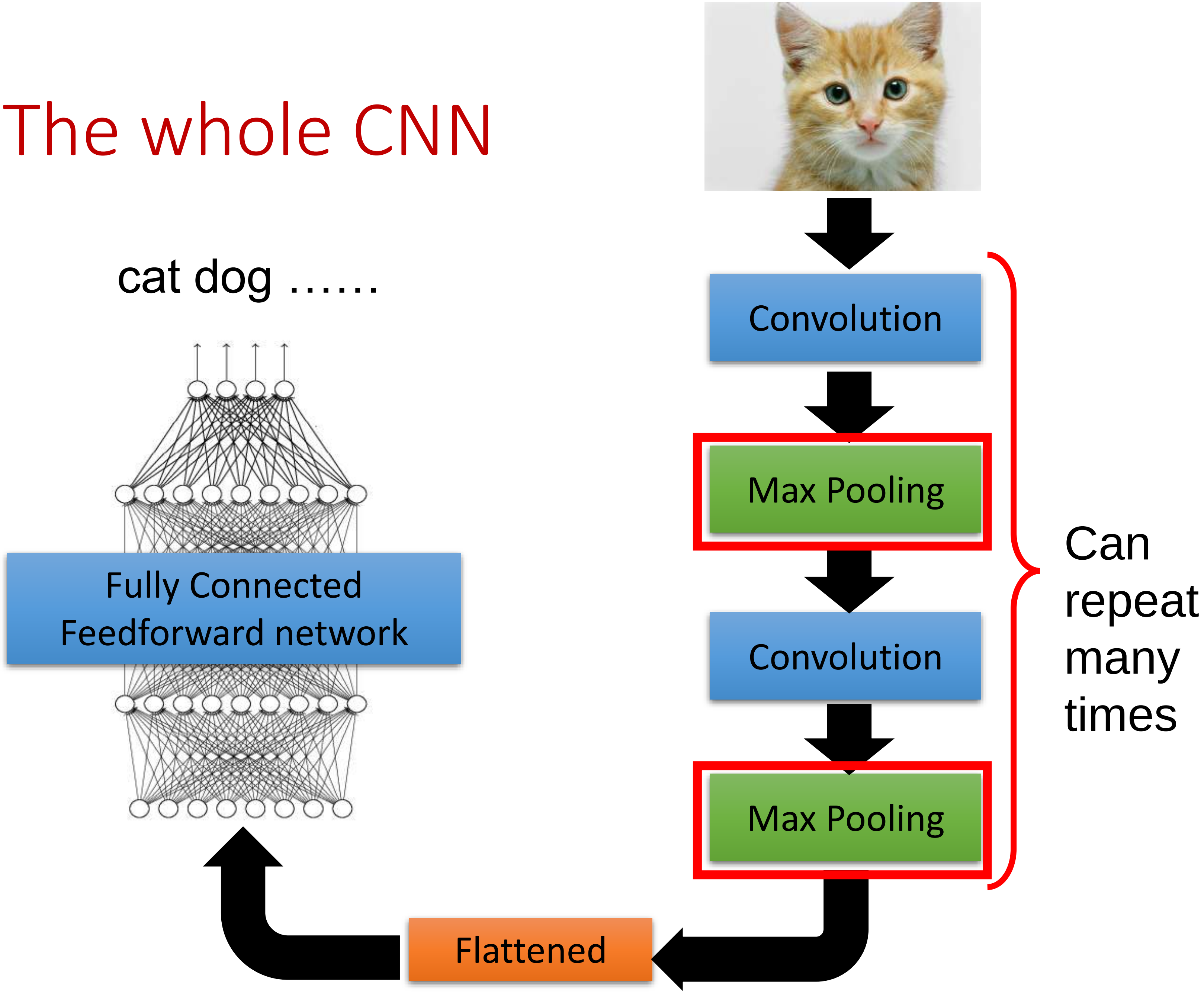
Filter 2

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

Color image

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

The whole CNN



Max Pooling

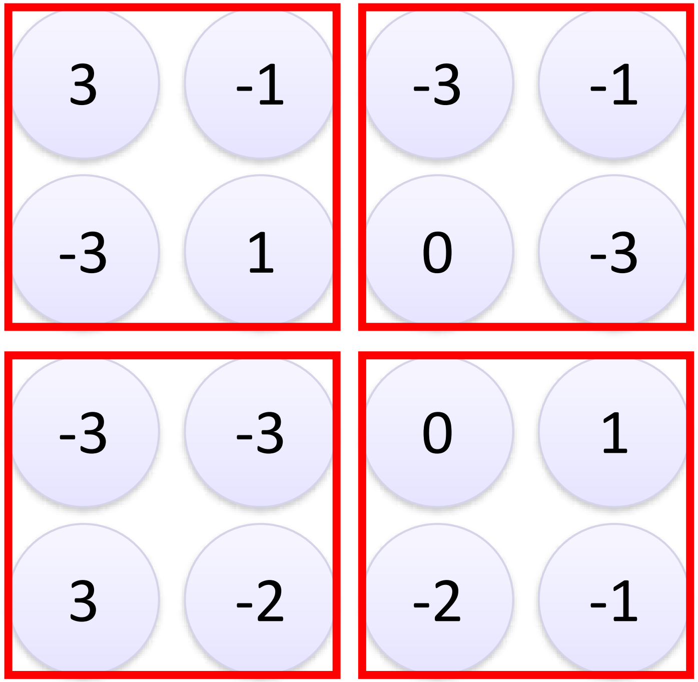
Size reduces after Max Pooling

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

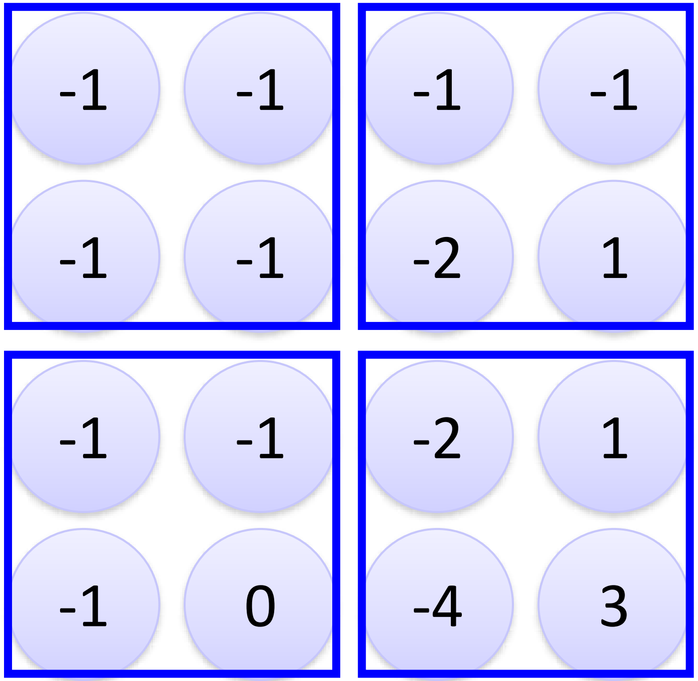
Filter 1

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

Filter 2



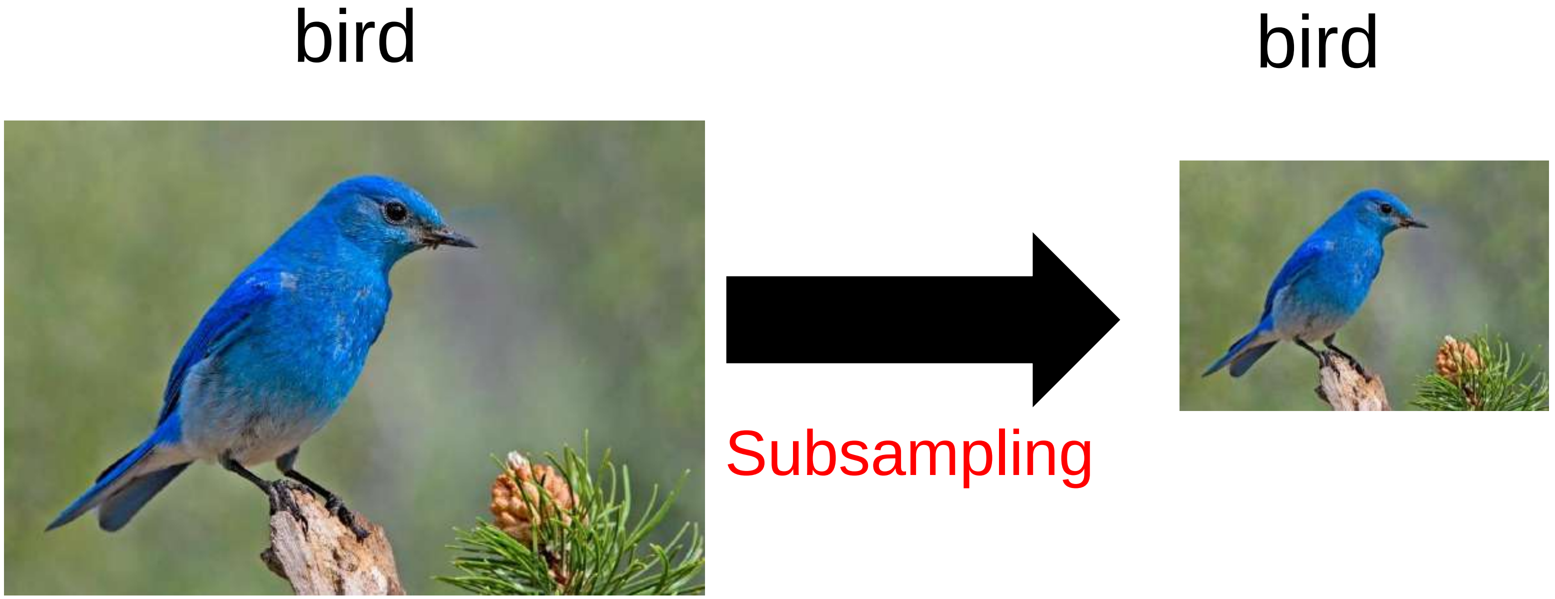
3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1



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

Why Pooling

- Subsampling pixels will not change the object



bird

Subsampling

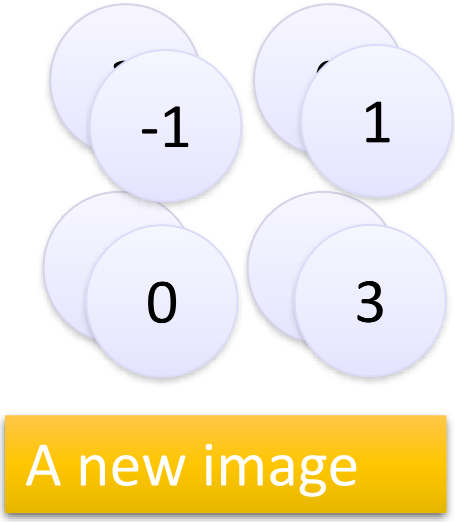
bird

- We can subsample the pixels to make image smaller
- fewer parameters to characterize the image

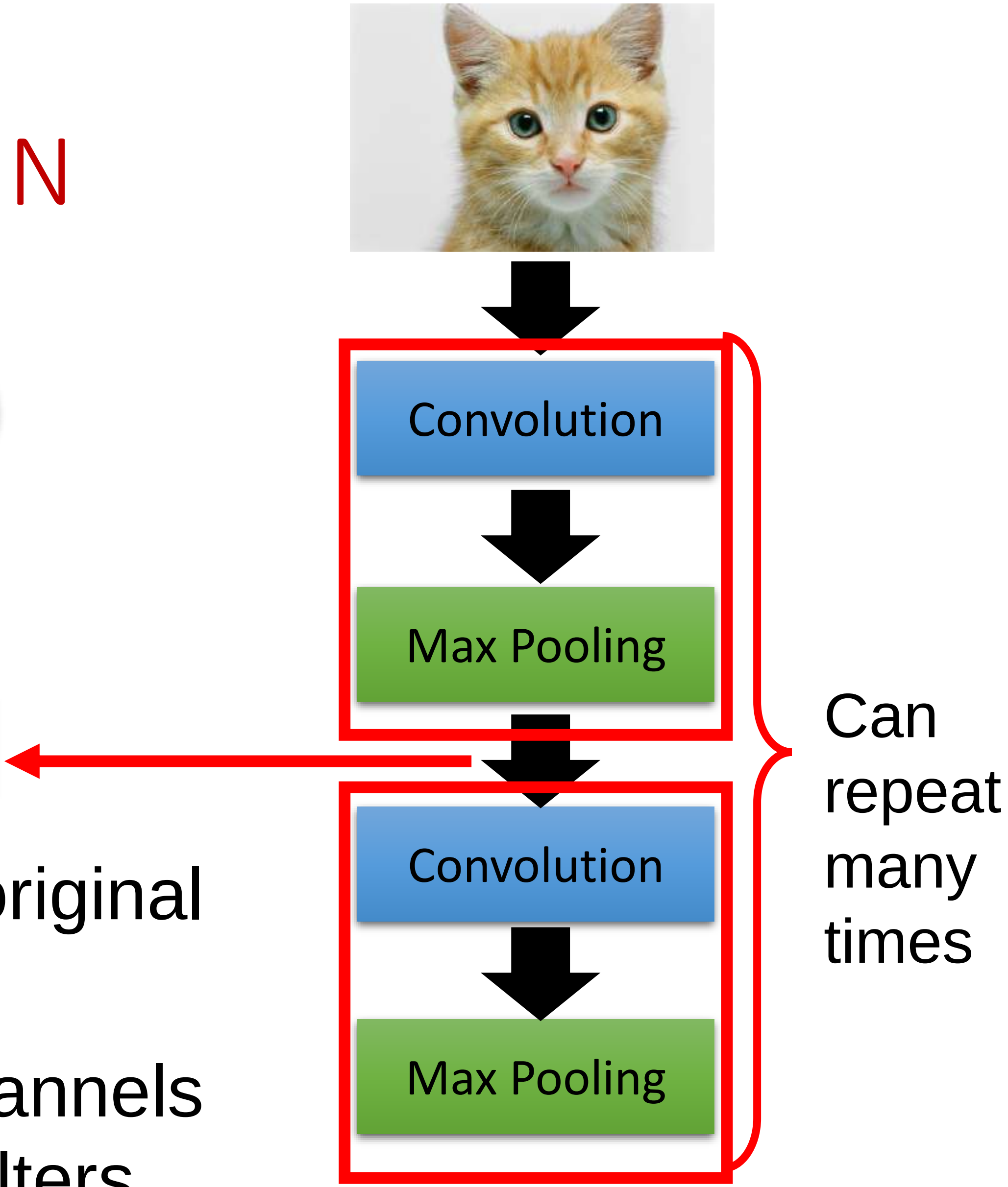
A CNN compresses a fully connected network in two ways:

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

The whole CNN



A new image



Convolution

Max Pooling

Convolution

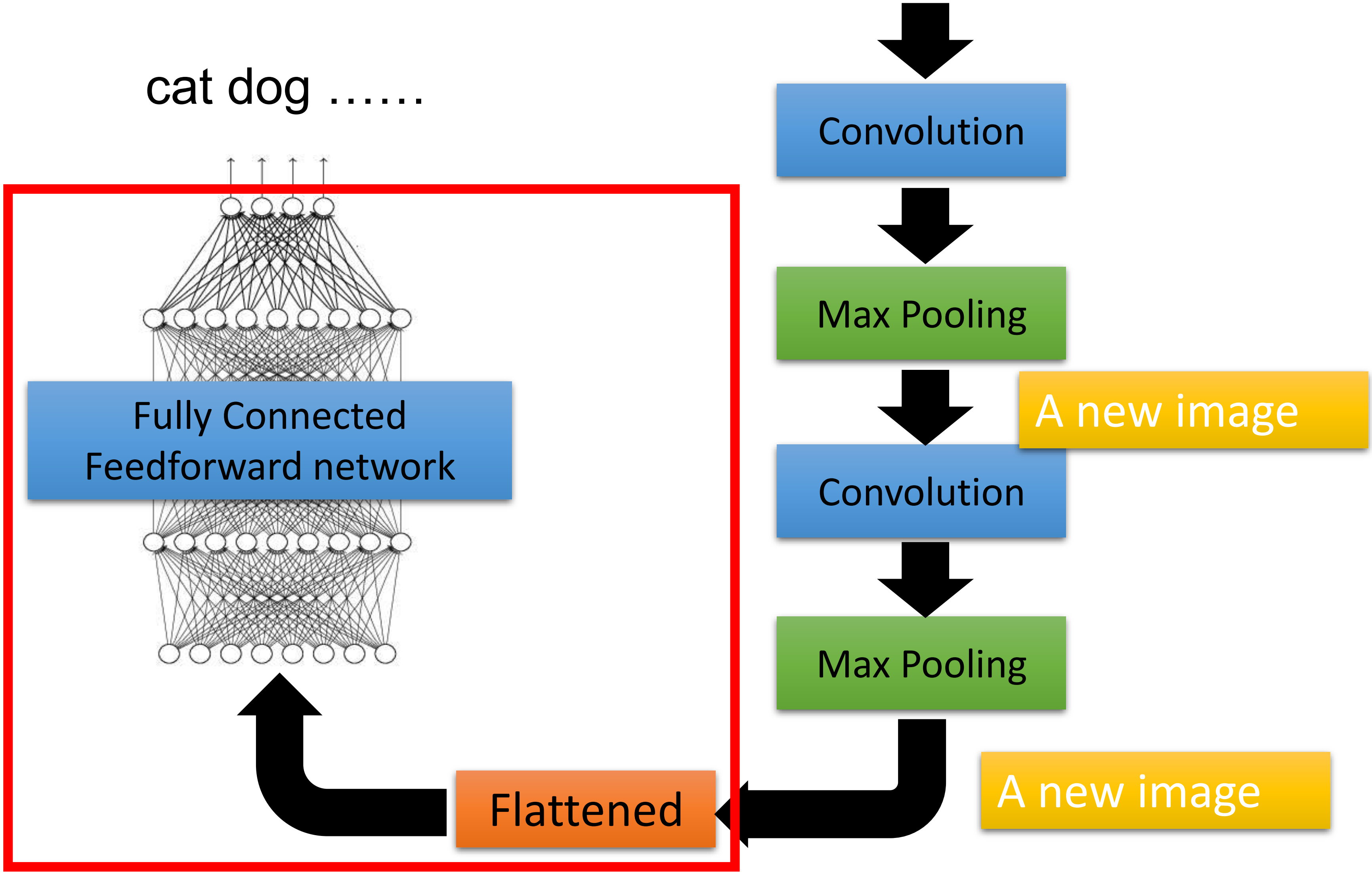
Max Pooling

Can repeat many times

Smaller than the original image

The number of channels is the number of filters

The whole CNN



cat dog ……

Convolution

Max Pooling

A new image

Convolution

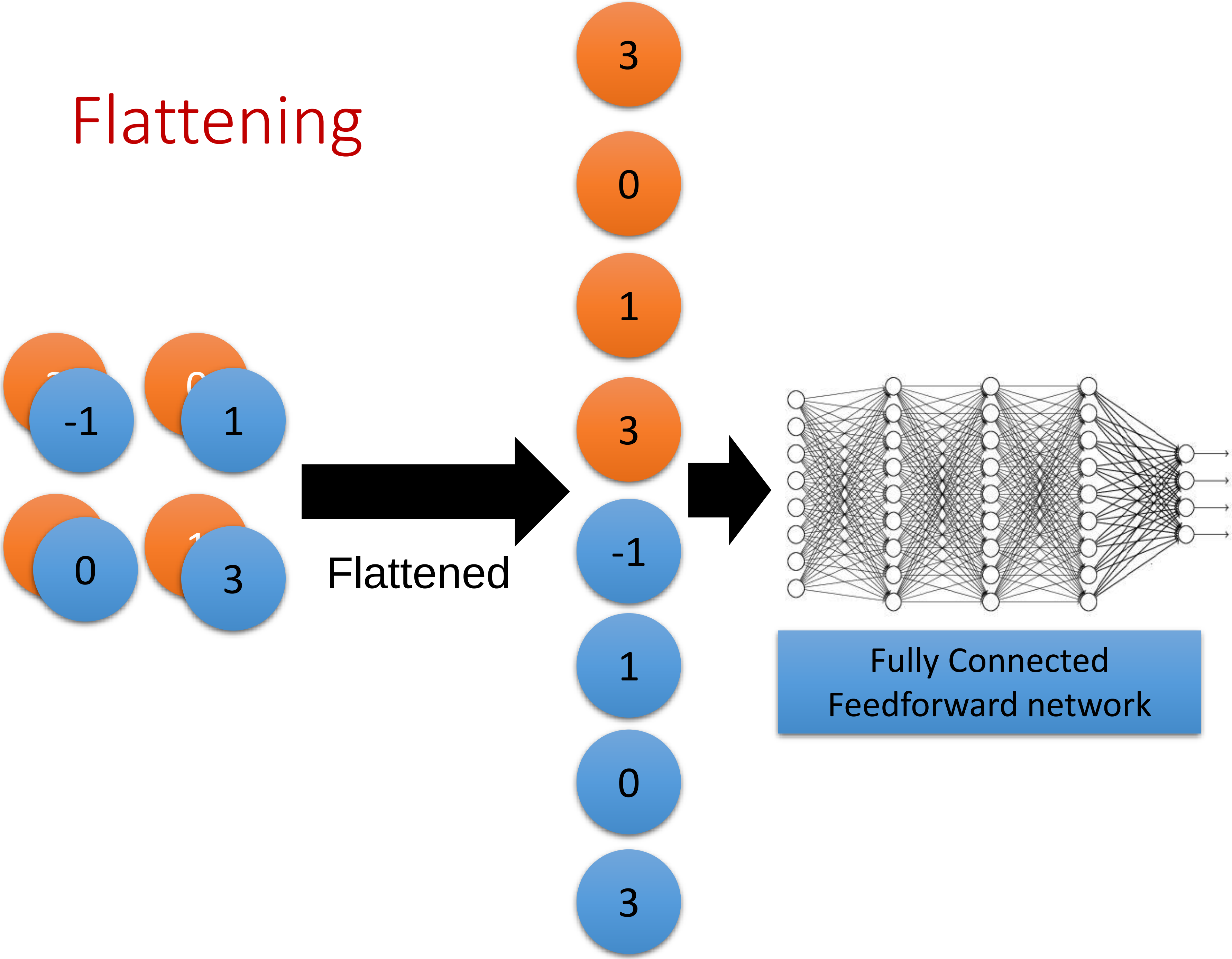
Max Pooling

A new image

Flattened

Fully Connected Feedforward network

Flattening



Flattened

Fully Connected Feedforward network

Second Example

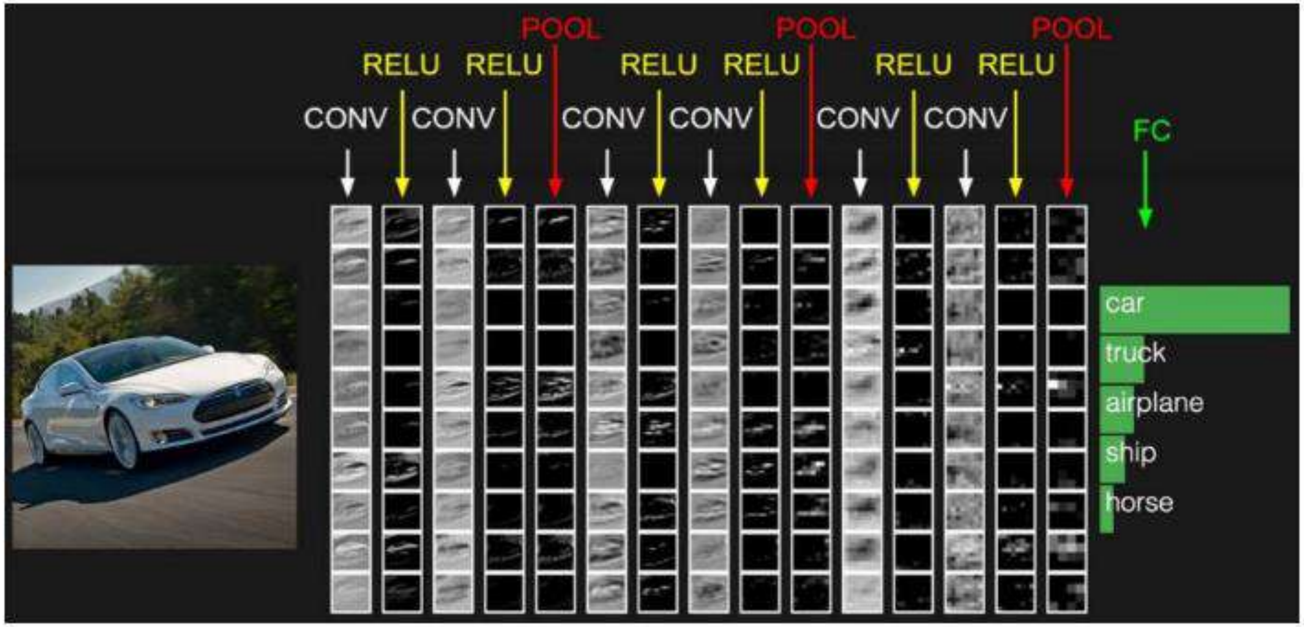
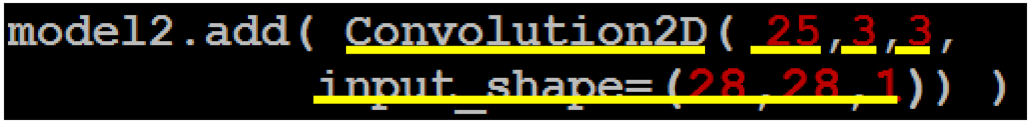


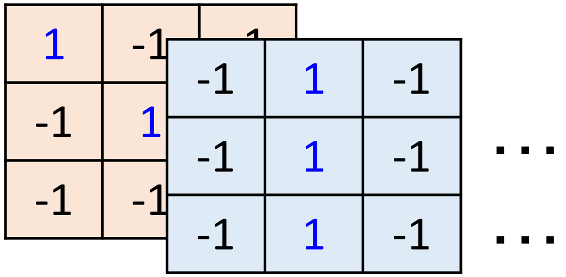
Image source: cs231n.stanford.edu

CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D tensor)***



```
model2.add( Convolution2D( 25,3,3,
                input_shape=(28,28,1)) )
```

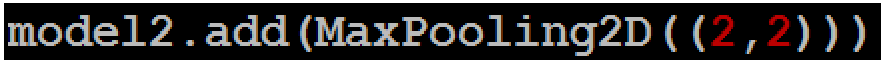


There are 25 3x3 filters.

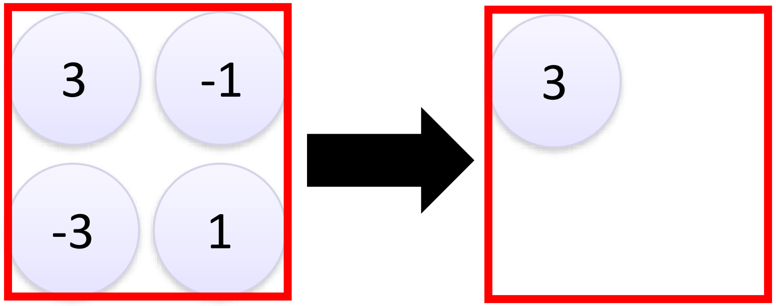
Input_shape = (28 , 28 , 1)

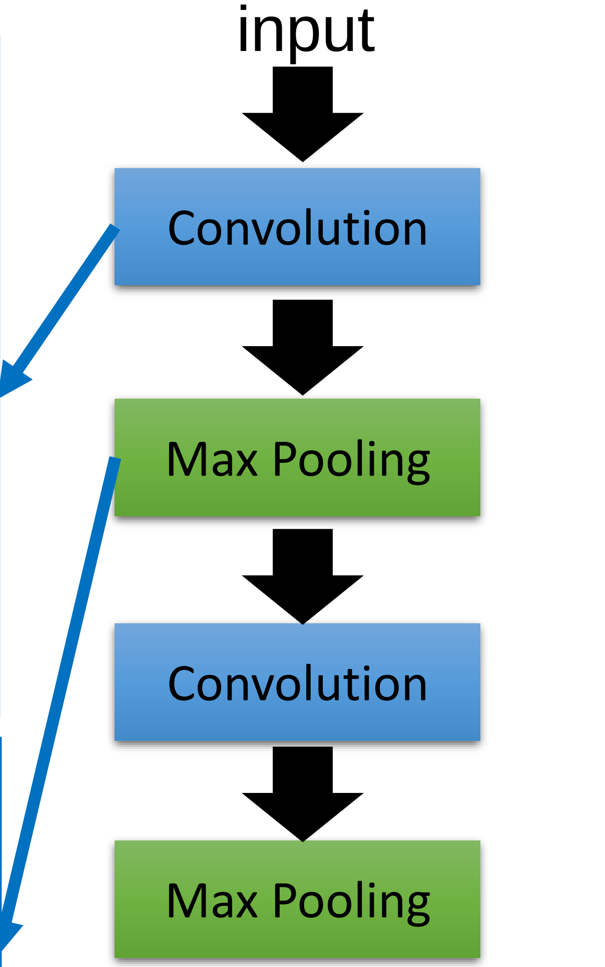
28 x 28 pixels

1: black/white, 3: RGB



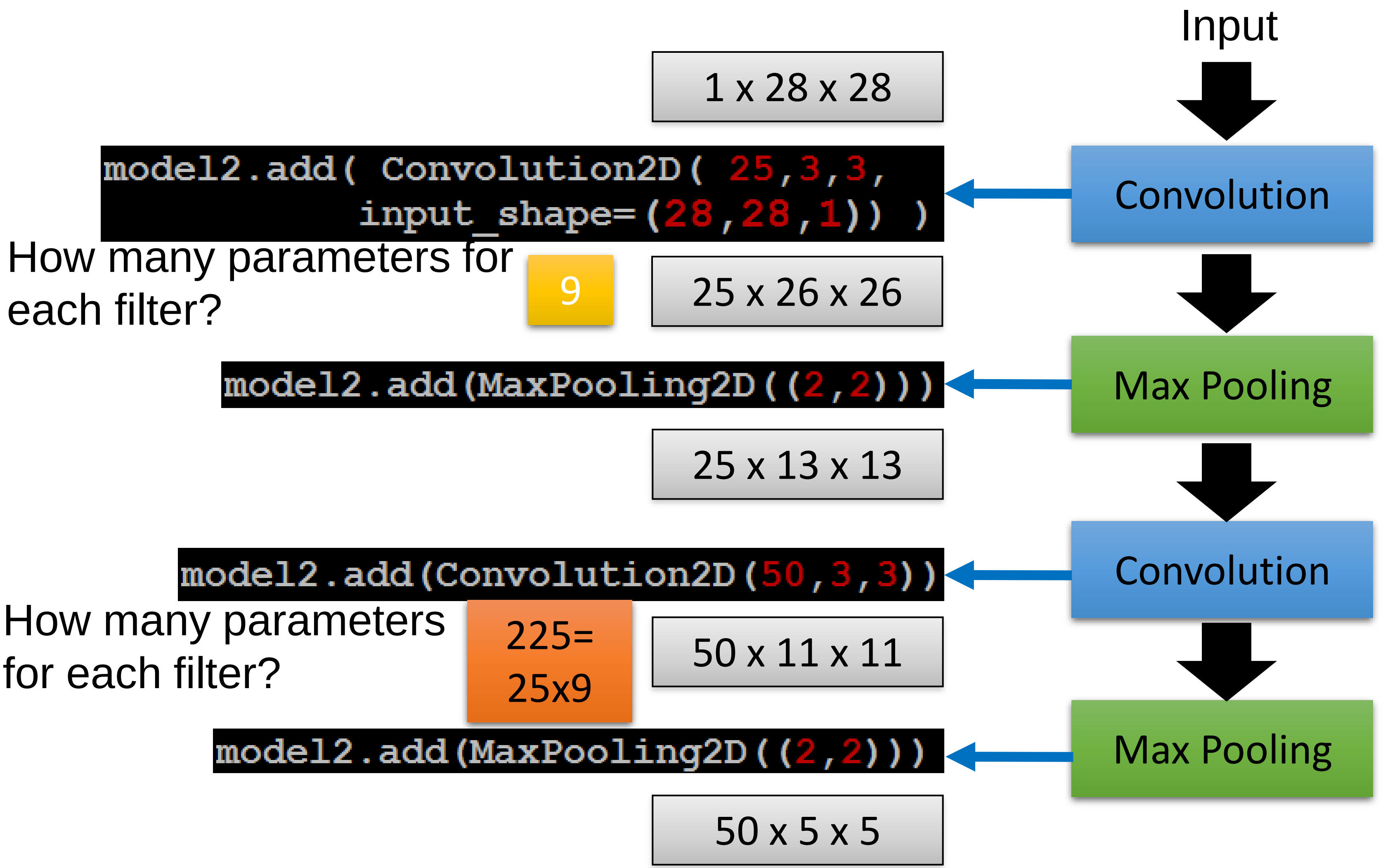
```
model2.add(MaxPooling2D((2,2)))
```





CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D array)***



1 x 28 x 28

```
model2.add( Convolution2D( 25,3,3,
            input_shape=(28,28,1)) )
```

How many parameters for each filter? 9

25 x 26 x 26

```
model2.add(MaxPooling2D((2,2)))
```

25 x 13 x 13

```
model2.add(Convolution2D(50,3,3))
```

How many parameters for each filter? 225= 25x9

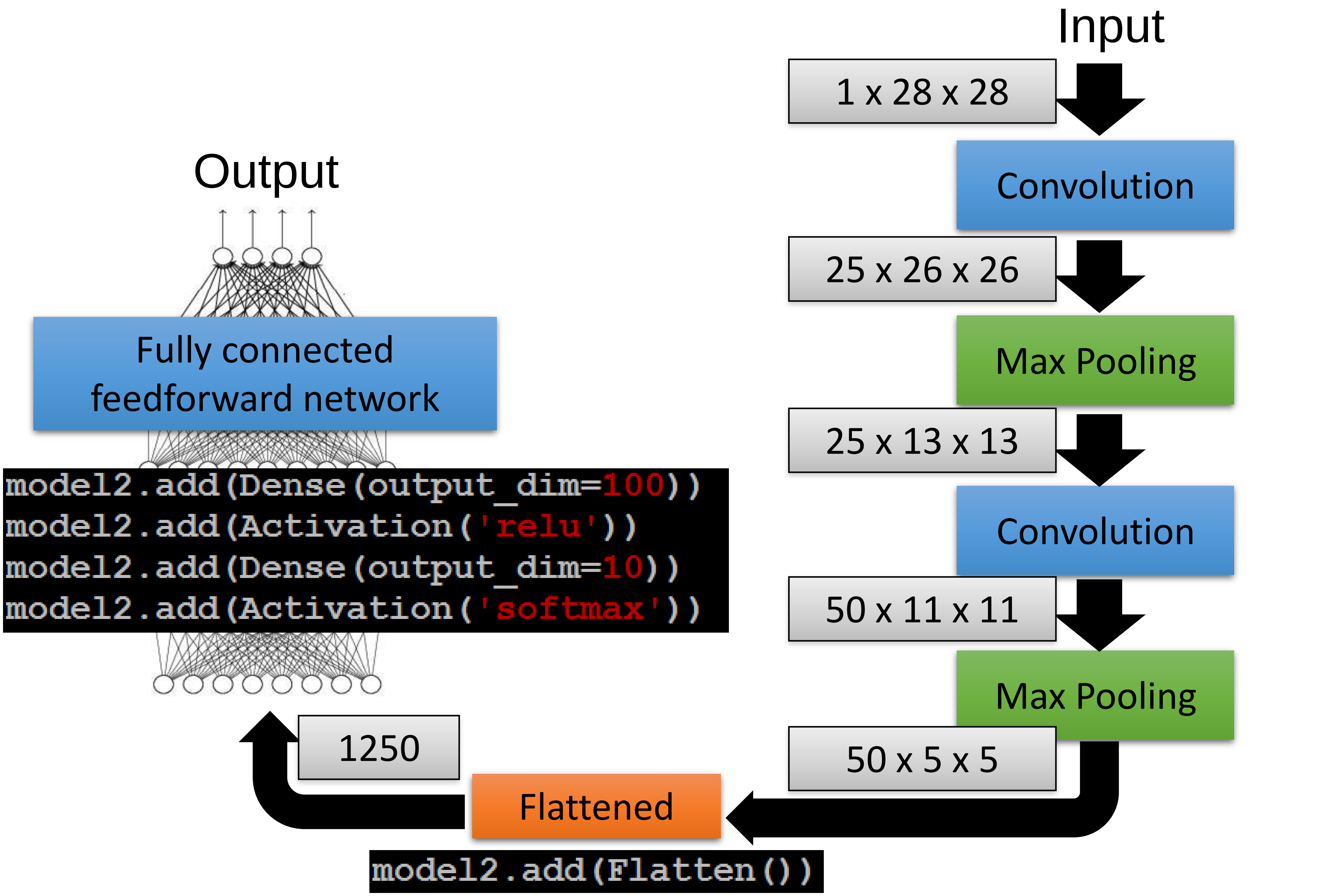
50 x 11 x 11

```
model2.add(MaxPooling2D((2,2)))
```

50 x 5 x 5

CNN in Keras

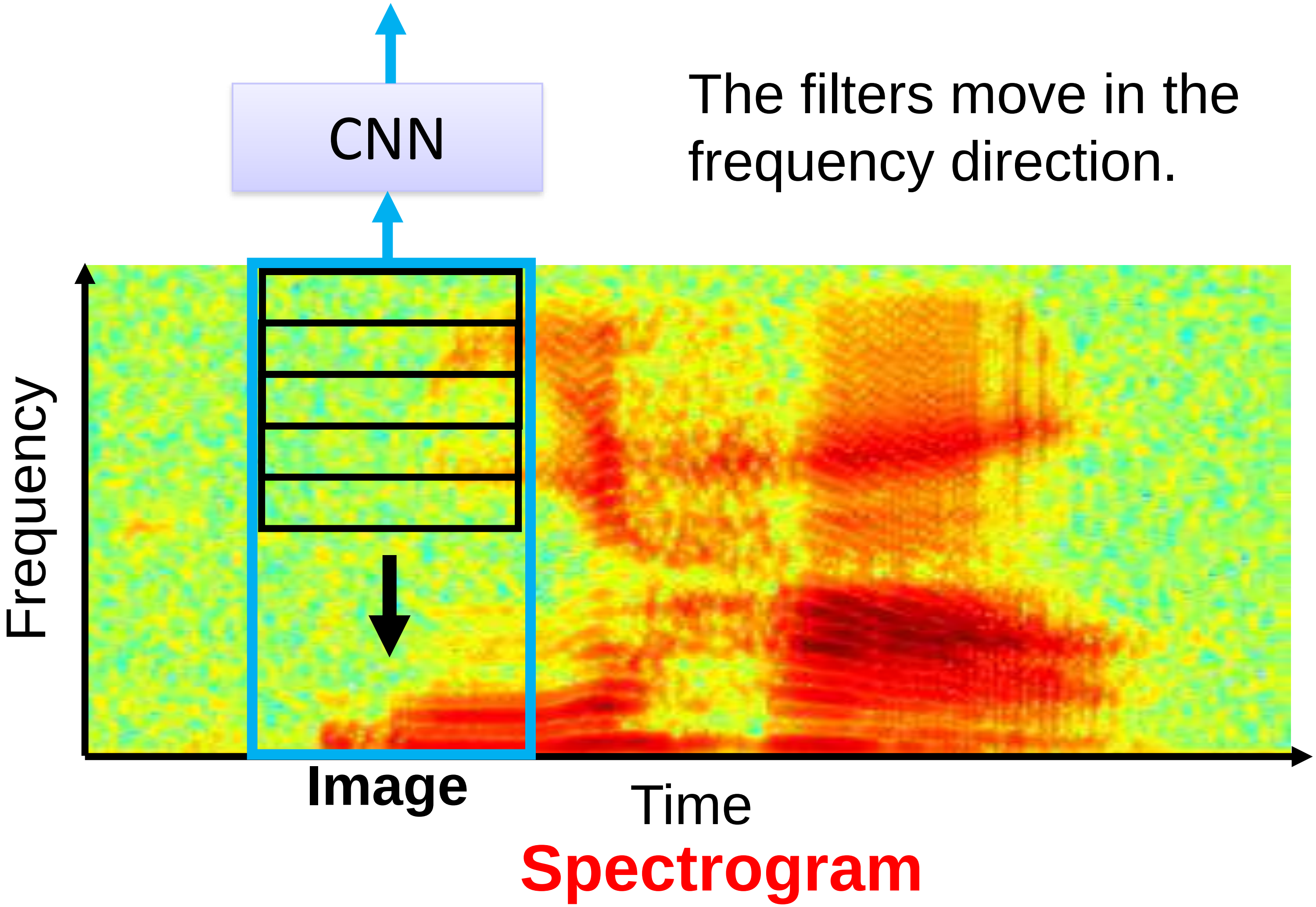
Only modified the ***network structure*** and ***input format (vector -> 3-D array)***



```
model2.add(Dense(output_dim=100))
model2.add(Activation('relu'))
model2.add(Dense(output_dim=10))
model2.add(Activation('softmax'))
```

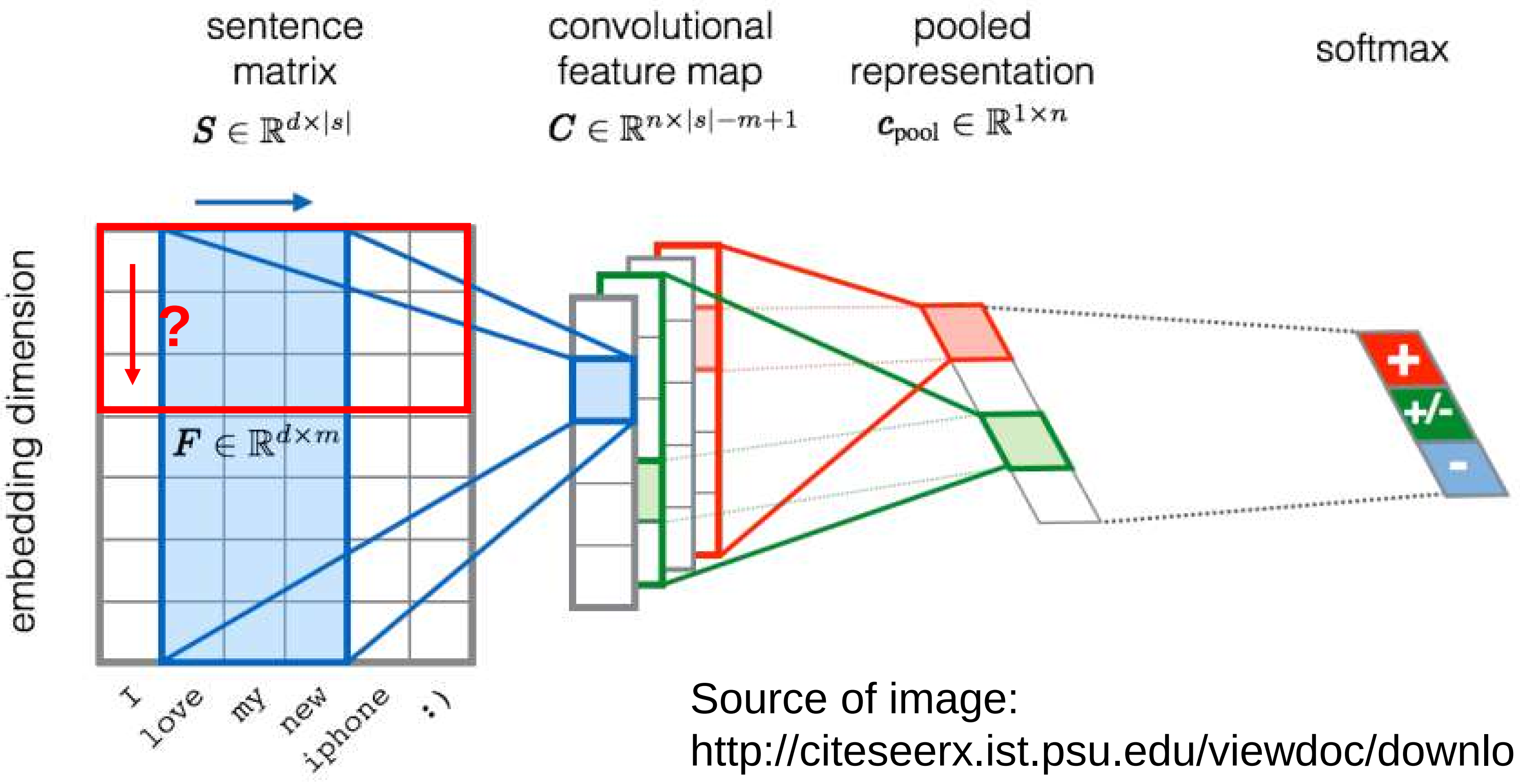
```
model2.add(Flatten())
```

CNN in speech recognition



The filters move in the frequency direction.

CNN in text classification


sentence matrix
$S \in \mathbb{R}^{d \times |s|}$
convolutional feature map
$C \in \mathbb{R}^{n \times |s|-m+1}$
pooled representation
$c_{\text{pool}} \in \mathbb{R}^{1 \times n}$
softmax
embedding dimension
?
$F \in \mathbb{R}^{d \times m}$
+
+/-
-
I
love
my
new
iphone
:)

Source of image: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.703.6858&rep=rep1&type=pdf