The week ahead

- **Quiz 11:** means is 89% and average completion time 5 min.
- **Quiz 12, Friday, Nov 20th 6am until Nov 22nd 11:59pm (midnight)**
  - Practical advice
  - This quiz will have 10 questions (15min)
  - If you get 5 questions correctly, you get a full grade on it
- **Touch-point 3**: deliverables due **Nov 22nd**, live-event Mon, Nov 23rd
  - Single-slide presentation outlining progress highlights and current challenges
  - Three-minute pre-recorded presentation with your progress and current challenges
- **Project final report due Dec 7th 11:59pm (midnight)**
  - GitHub page with all of the results you have achieved utilizing both unsupervised learning and supervised learning
  - Final seven-minute long pre-recorded presentation

CS4641B Machine Learning

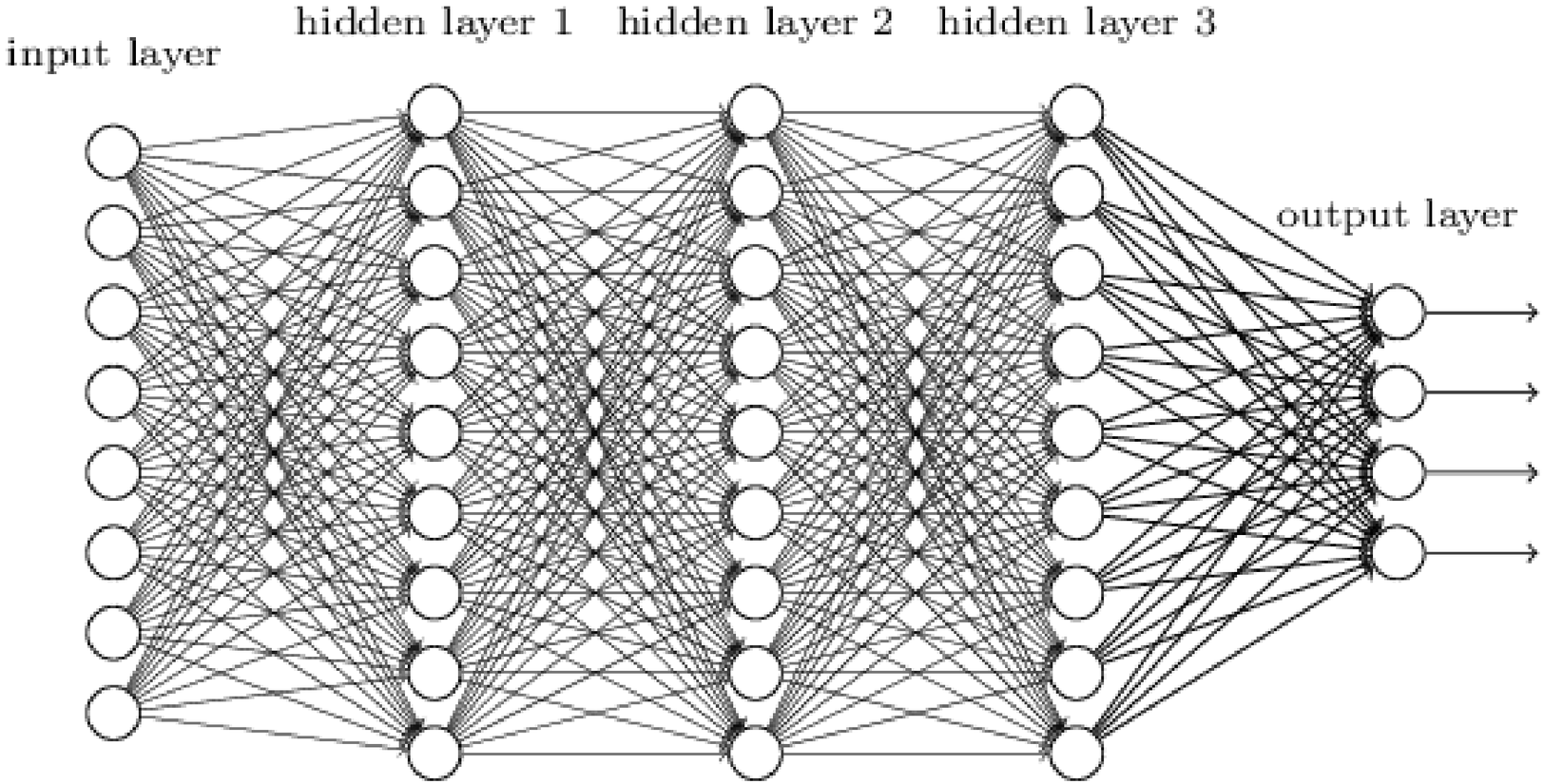
Lecture 23: Convolutional neural networks

Rodrigo Borela ▸ rborelav@gatech.edu

Slides adapted from Ming Li (University of Waterloo) and Mahdi Roozbahani

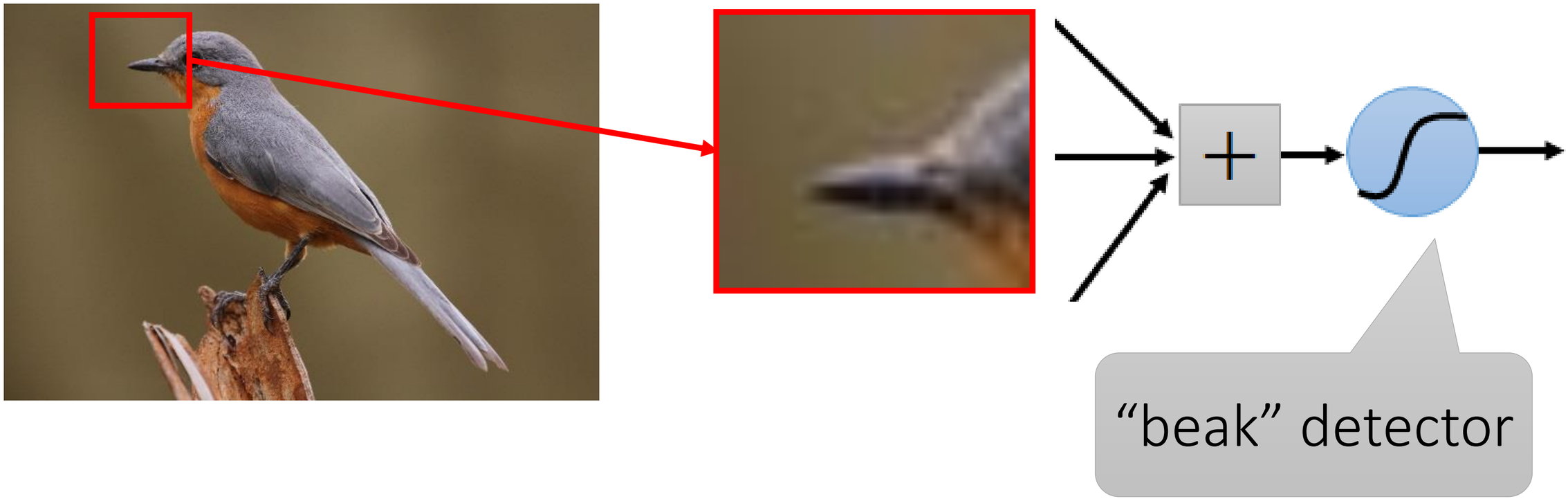
Learning smaller network models

- We know it is good to learn a small model.
- From this fully connected model, do we really need all the edges?
- Can we exploit correlation between features in any way?



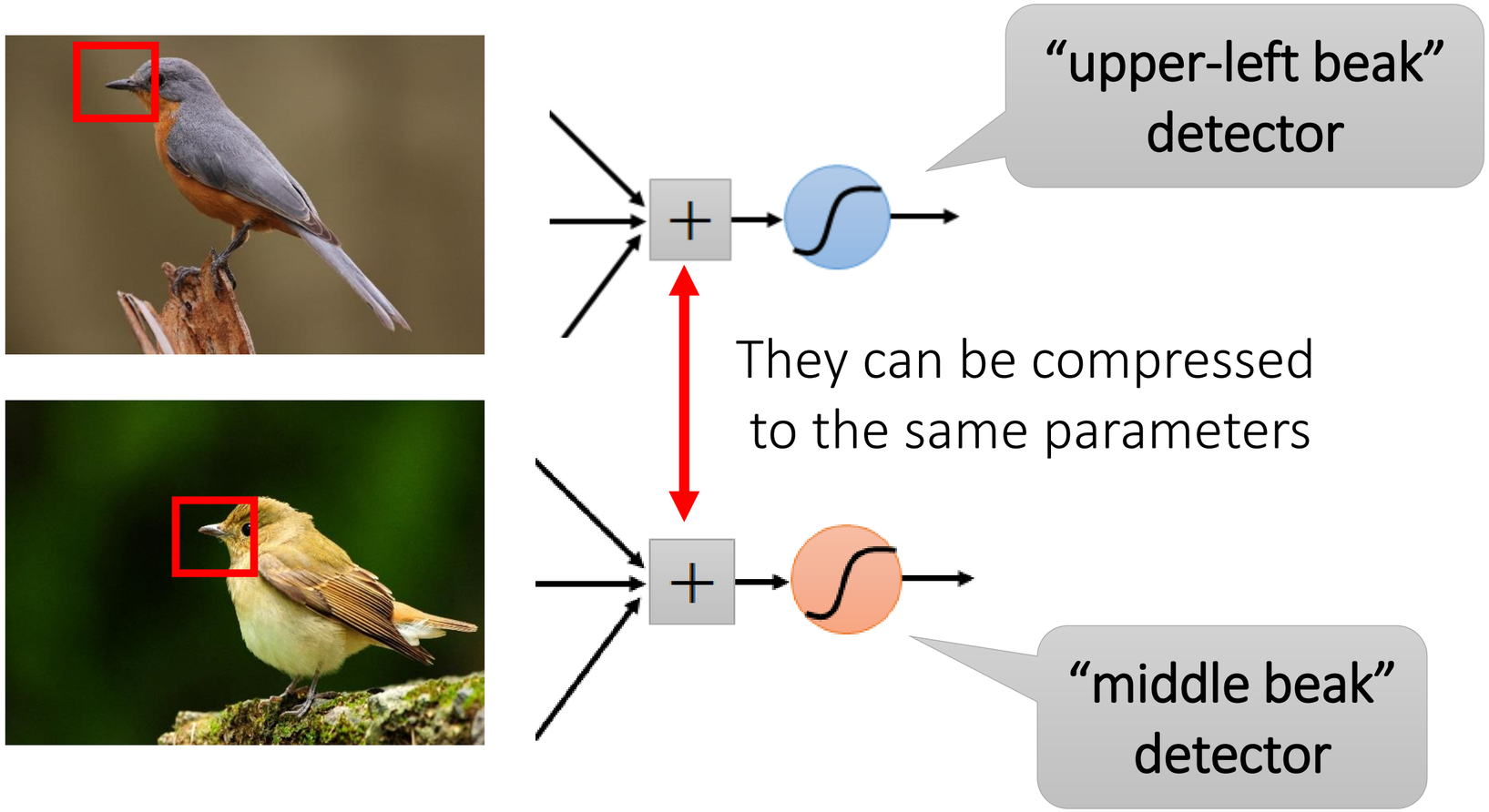
Consider learning an image classifier

- Some patterns are much smaller than the whole image
- Can represent a small region with fewer parameters



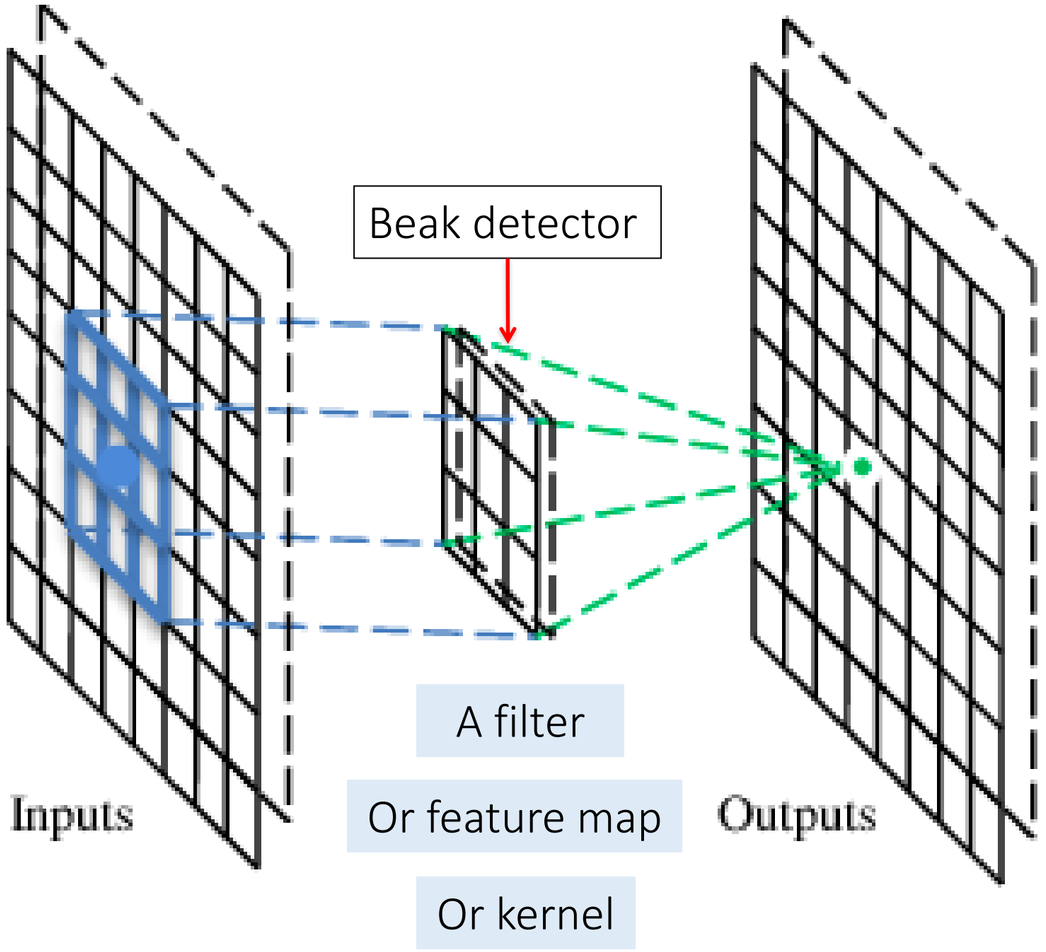
Building blocks

- Same pattern appears in different places: They can be compressed!
- What about training a lot of such “small” detectors and making them spatially independent (allowing them to “move around”.)



Convolutional layers

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



Convolution operation

- Example:

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 operation

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

dot product

3 -1

6 x 6 image

Convolution operation

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

Convolution operation

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

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

Convolution operation

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

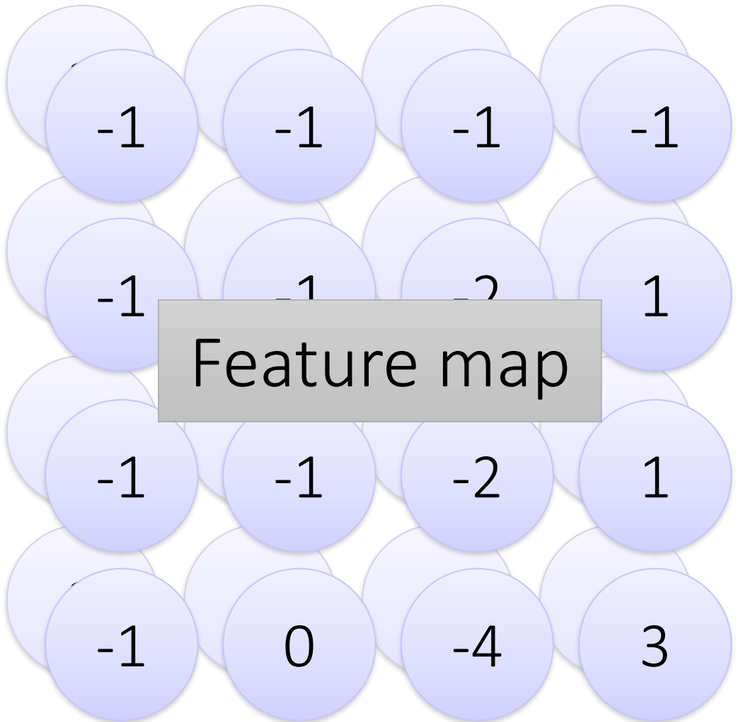
Filter 2

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

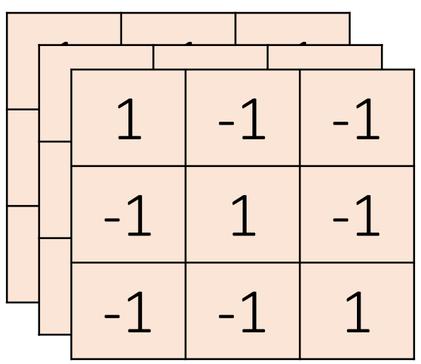
6 x 6 image

Repeat this for each filter

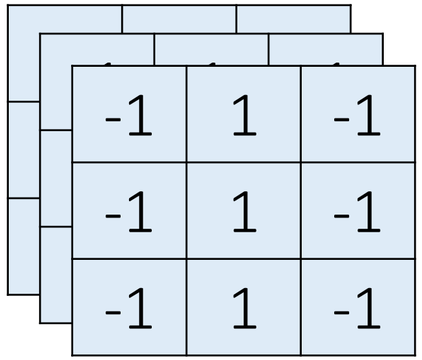


Two 4 x 4 images
Forming 2 x 4 x 4 matrix

Color image: RGB 3 channels

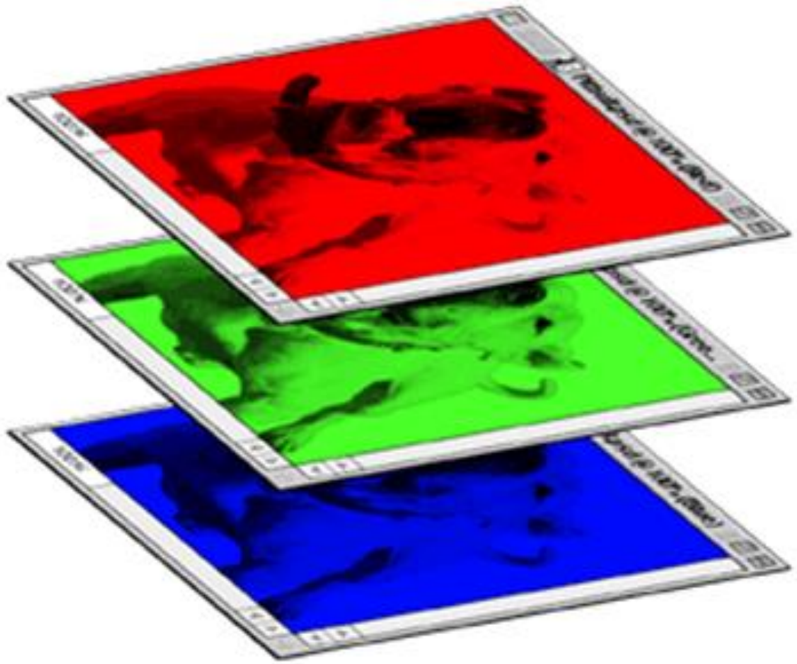


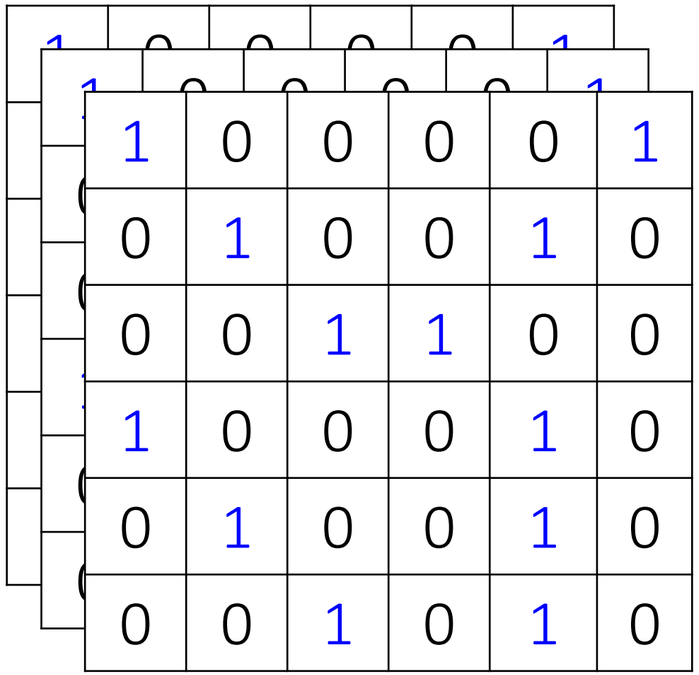
Filter 1



Filter 2

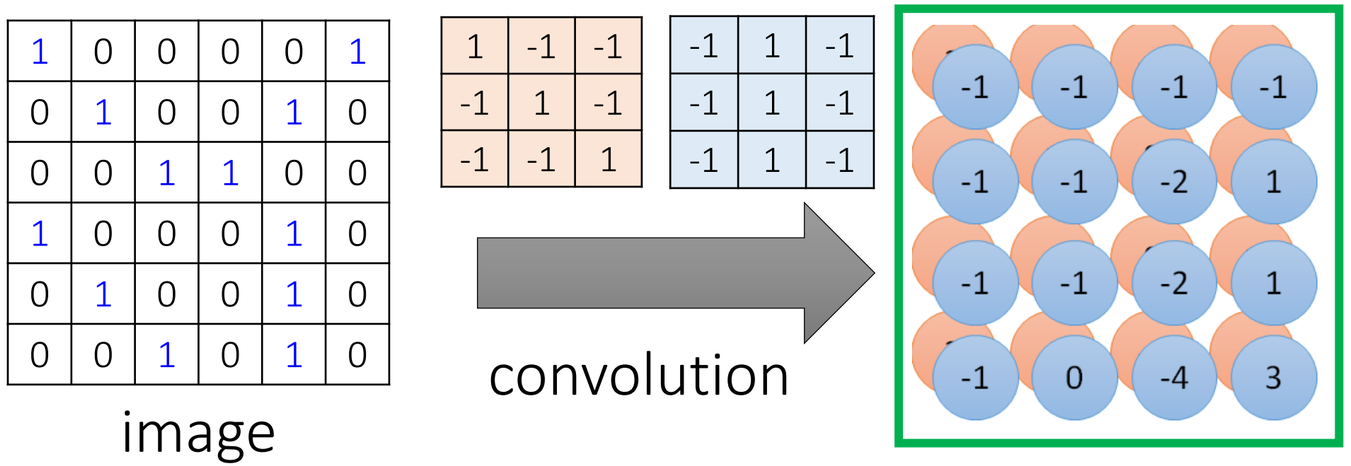
Color image



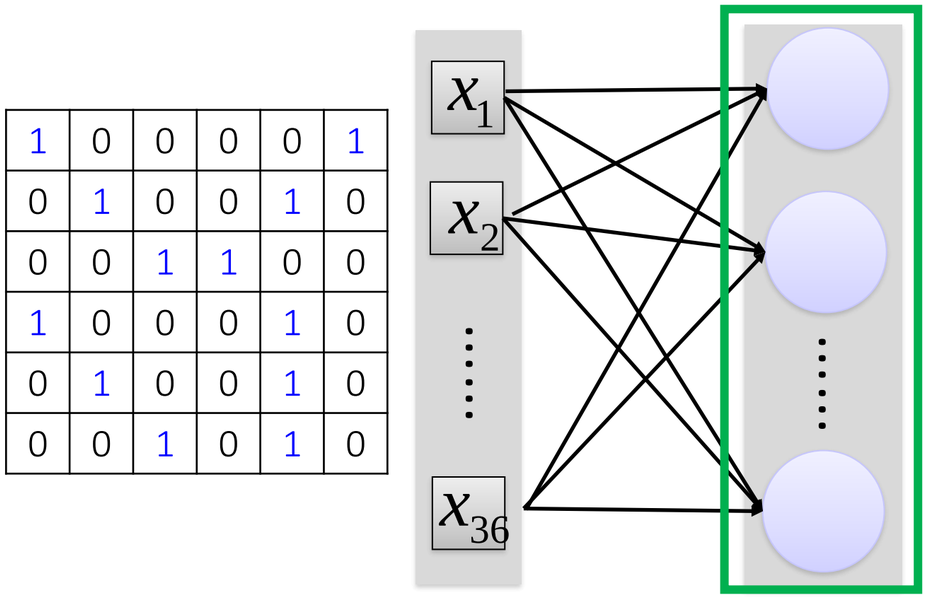


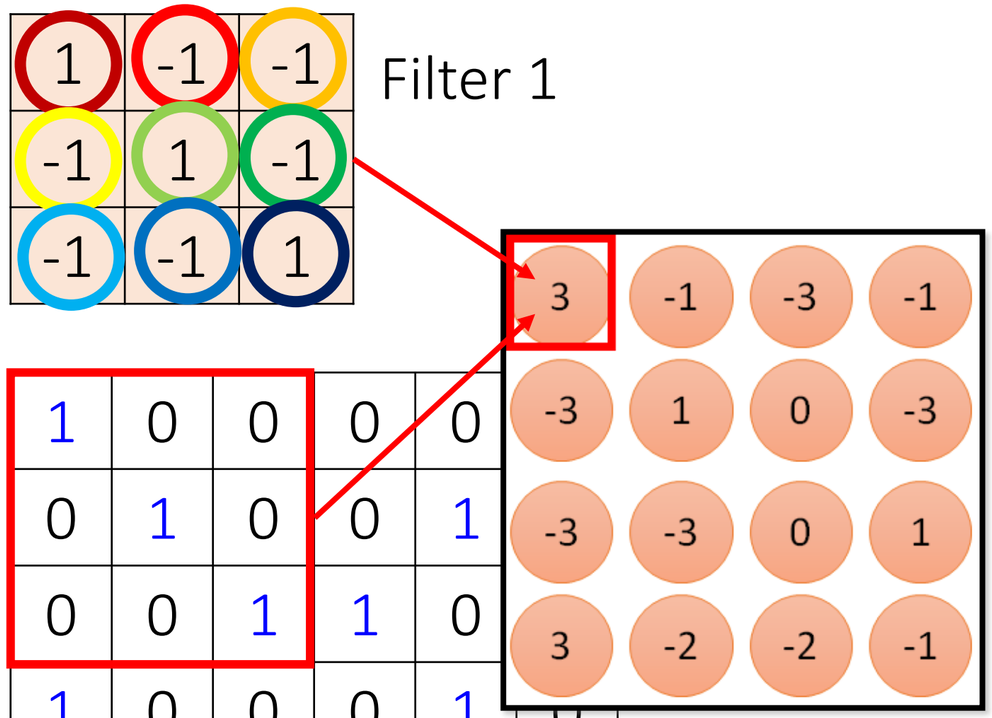
Convolution vs. fully connected

Convolutional

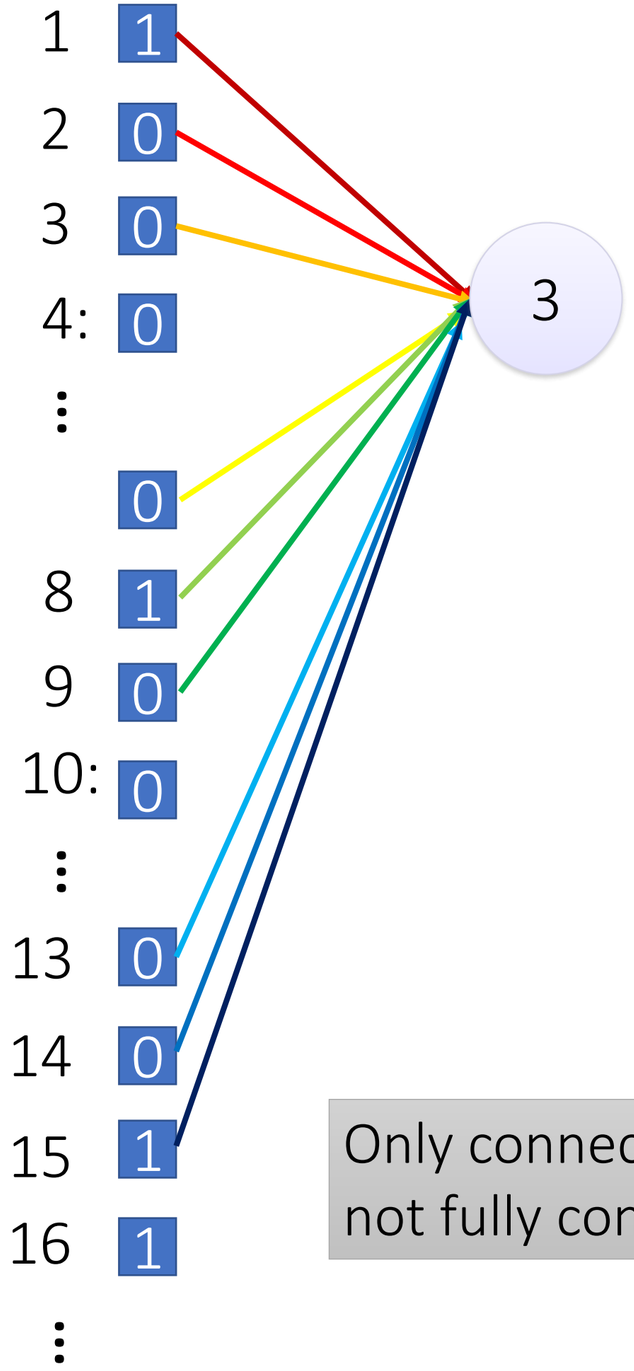


Fully-connected

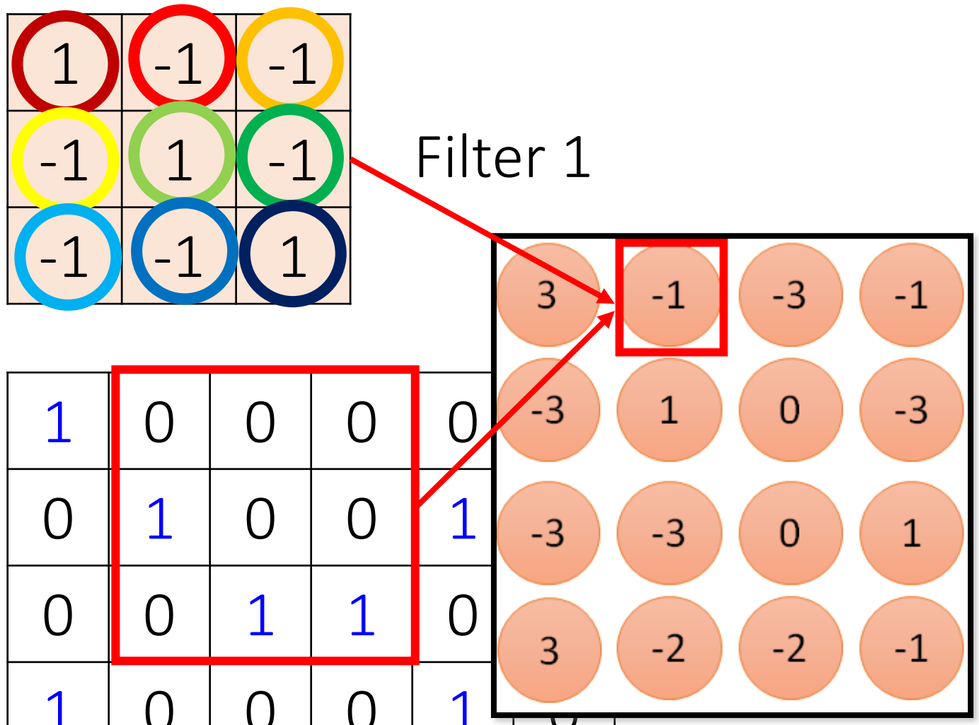


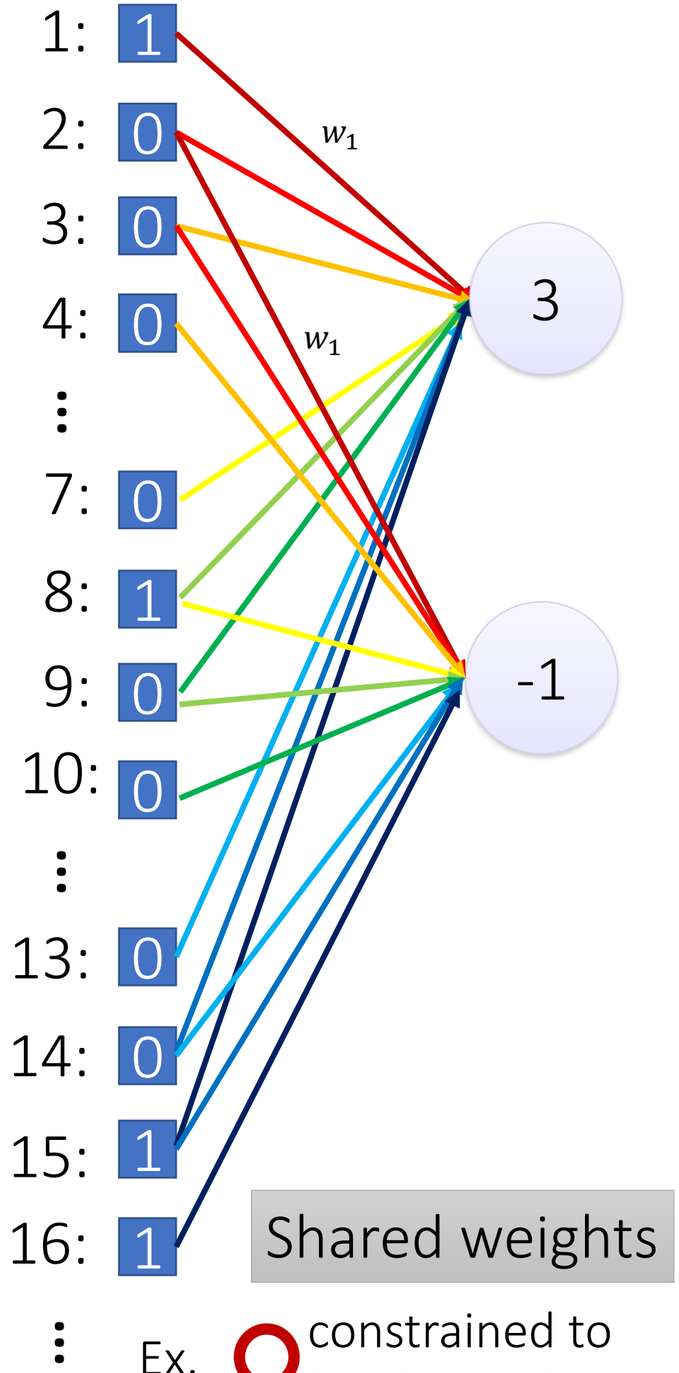


fewer parameters!



Only connect to 9 inputs, not fully connected





Even fewer parameters!

Shared weights

Ex. O constrained to be identical

The complete CNN

Convolution

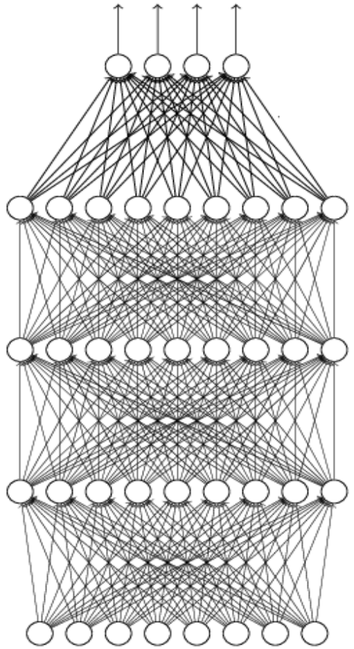
Max Pooling

Convolution

Max Pooling

Can repeat many times

Flattened



cat dog ……

Fully connected Feedforward network

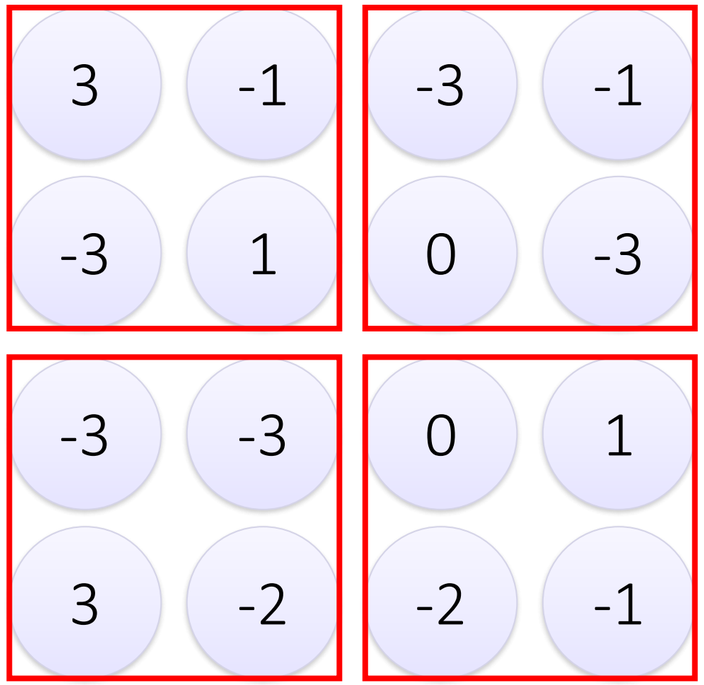
Max pooling

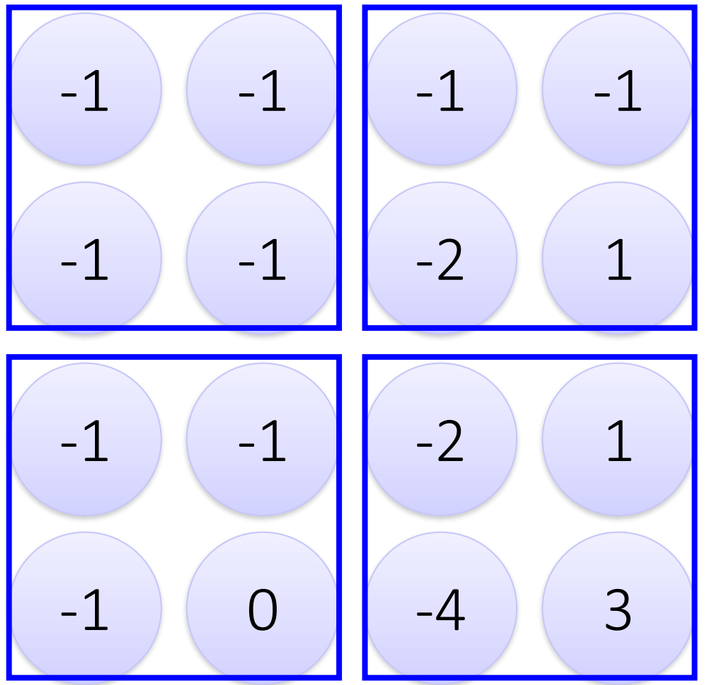
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

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

Filter 2





Why pooling?

- Subsampling pixels will not change the object

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

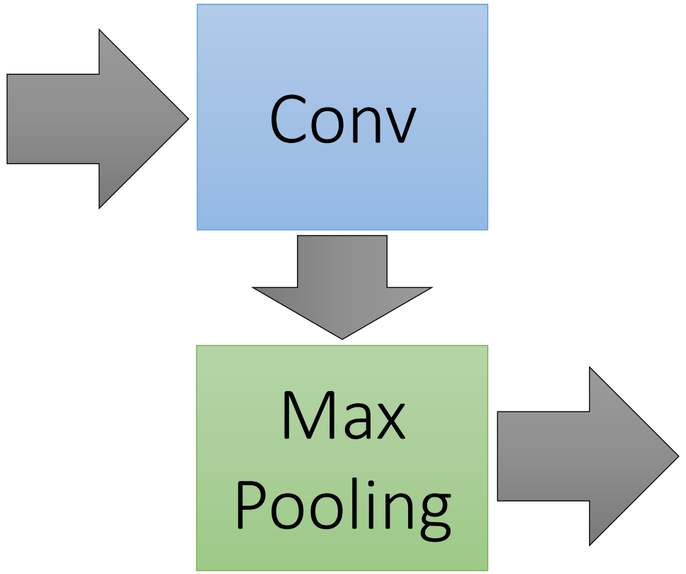
How does a CNN compress a fully connected network?

- 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



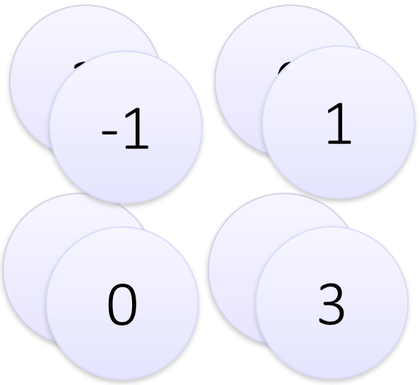
New image but smaller

-1	1
0	3

2 x 2 image

Each filter is a channel

The complete 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 complete CNN

Convolution

Max Pooling

A new image

Convolution

Max Pooling

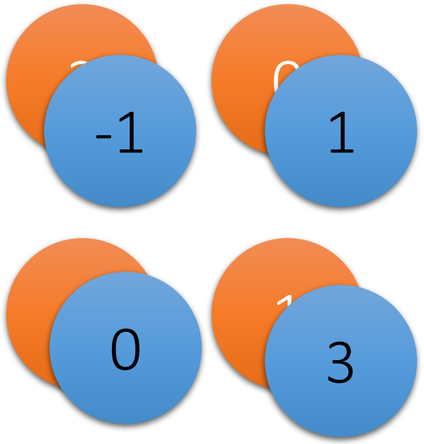
A new image

Flattened

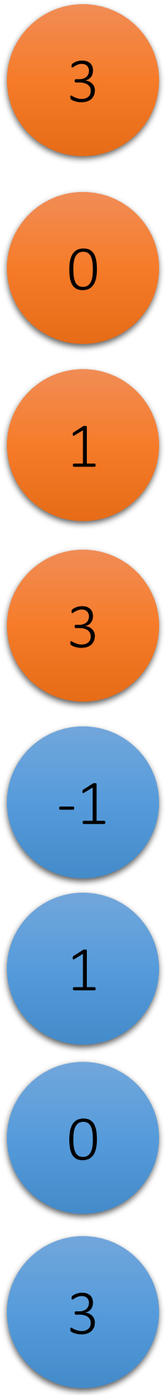
Fully connected Feedforward network

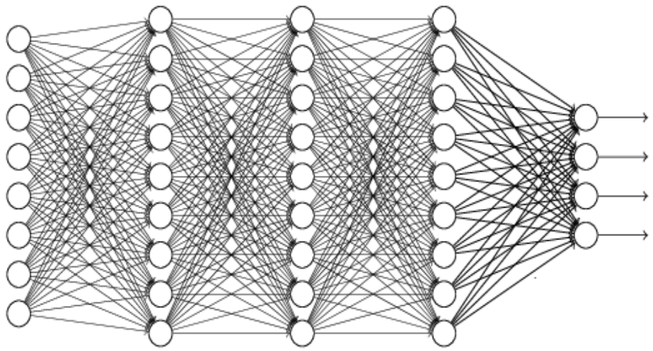
cat dog ……

Flattening



Flattened



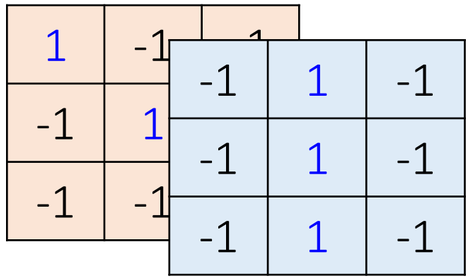


Fully connected
Feedforward network

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)) )
```



……

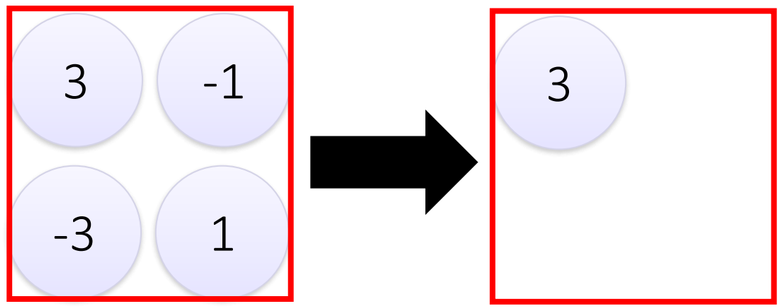
There are 25 3x3 filters.

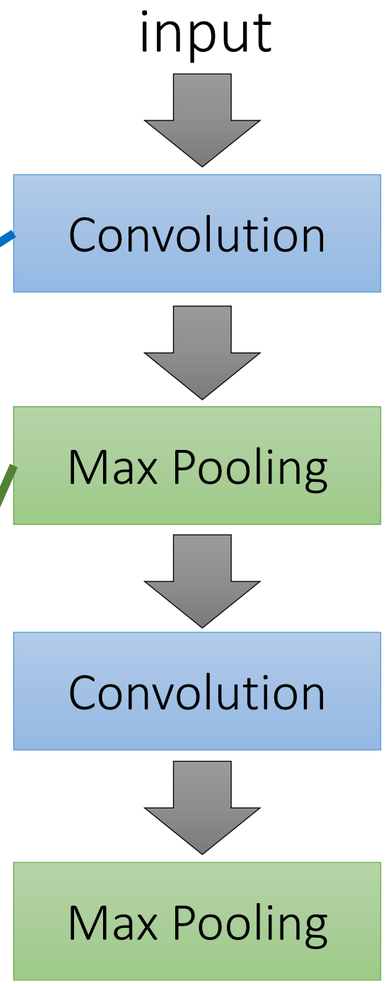
Input_shape = (28 , 28 , 1)

28 x 28 pixels

1: black/white, 3: RGB

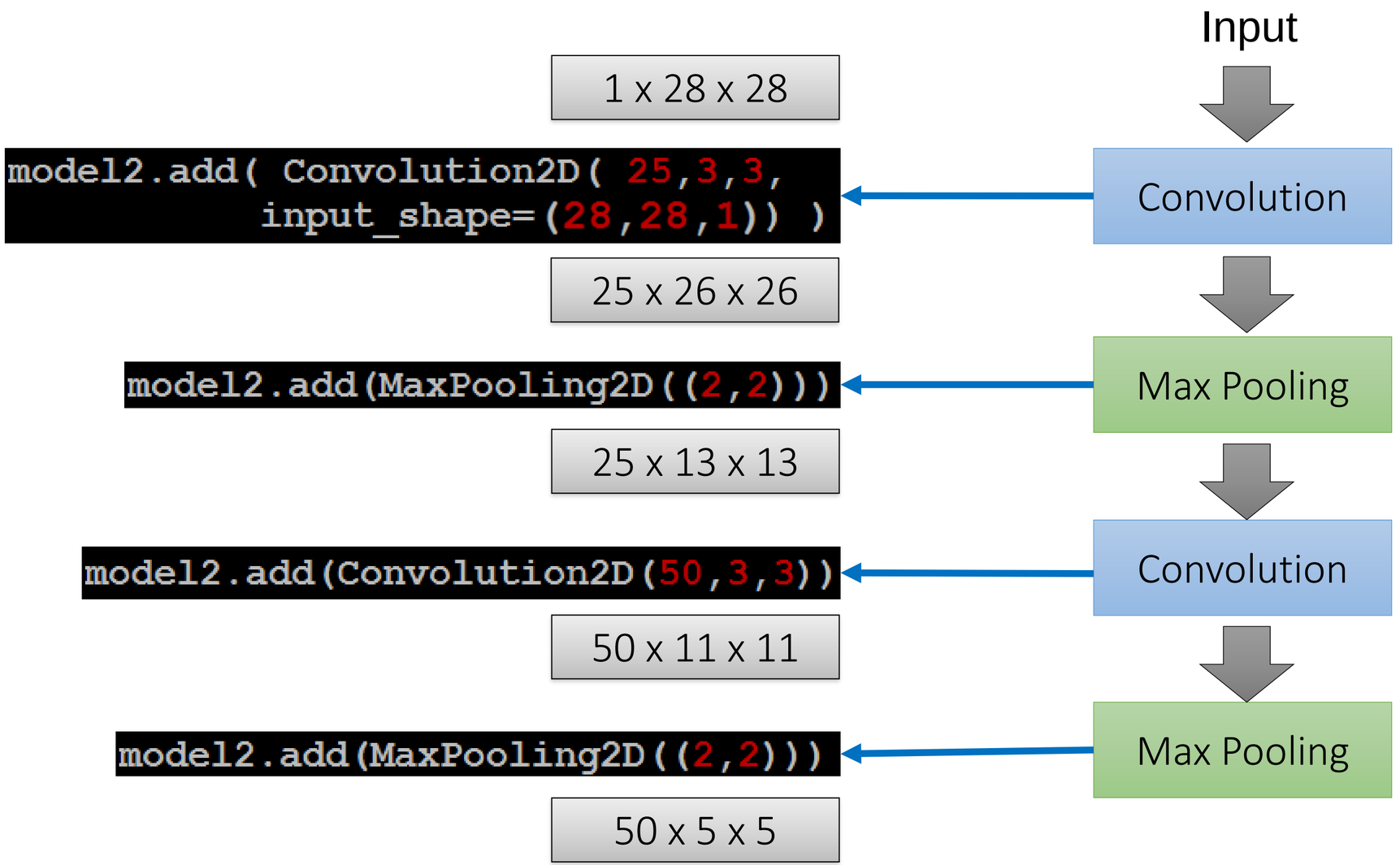
```
model2.add(MaxPooling2D((2,2)))
```





CNN in Keras

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



CNN in Keras

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

Input

1 x 28 x 28

Convolution + ReLU

25 x 26 x 26

Max Pooling

25 x 13 x 13

Convolution + ReLU

50 x 11 x 11

Max Pooling

50 x 5 x 5

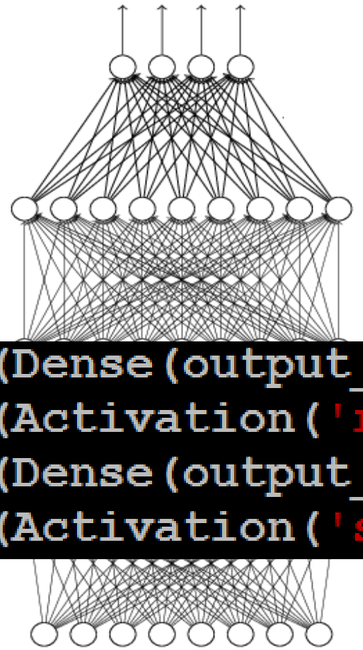
Flattened

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

1250

Output

Fully connected Feedforward network



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

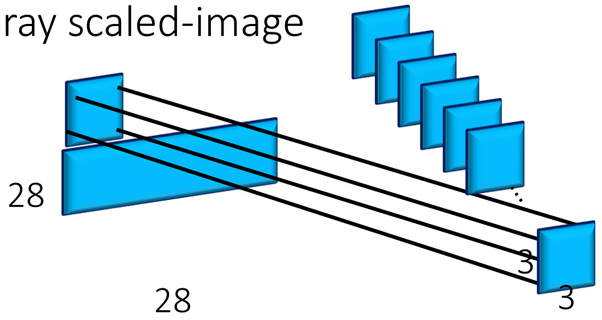
Number of parameters

$25 \times 3 \times 3 + 25$ parameters

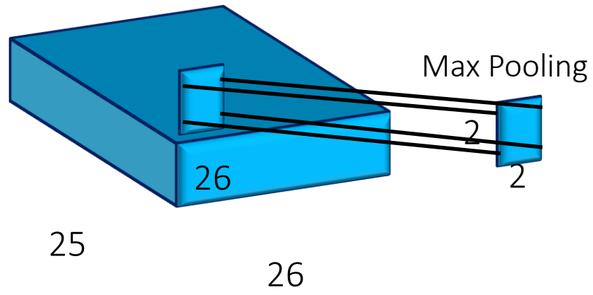
25 filters - Conv1

$25: 3 \times 3$

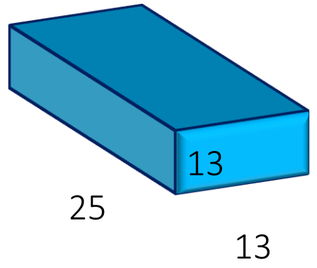
Gray scaled-image

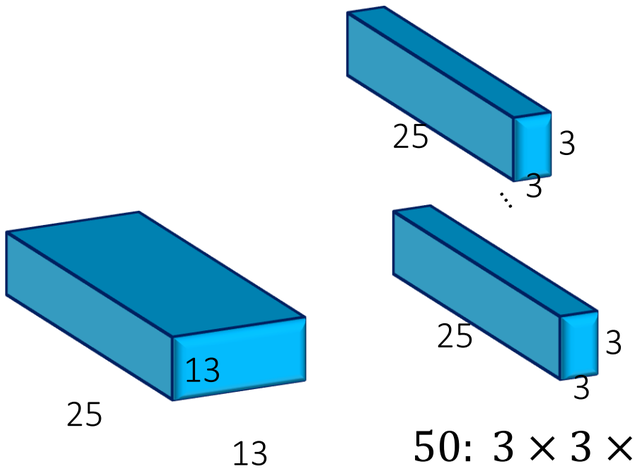


Convolved result for 25 filters



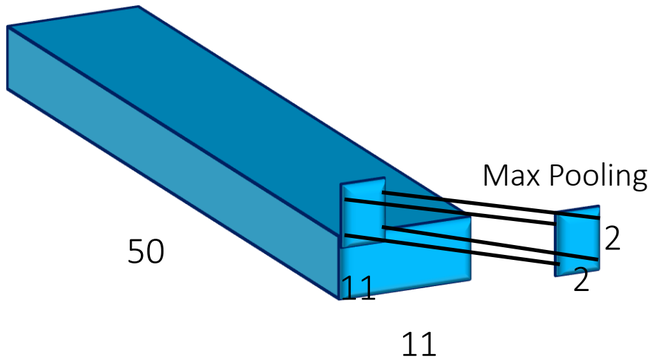
Result after Max Pooling





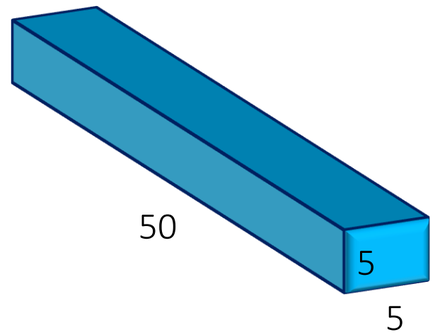
$50: 3 \times 3 \times 25$

50 filters – Conv2



Convolved result for 50 filters

Result after Max Pooling



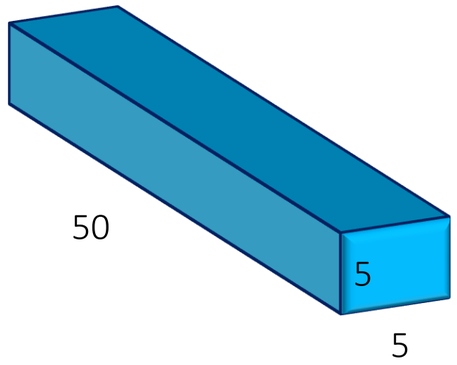
$50 \times 3 \times 3 \times 25 + 50$ parameters

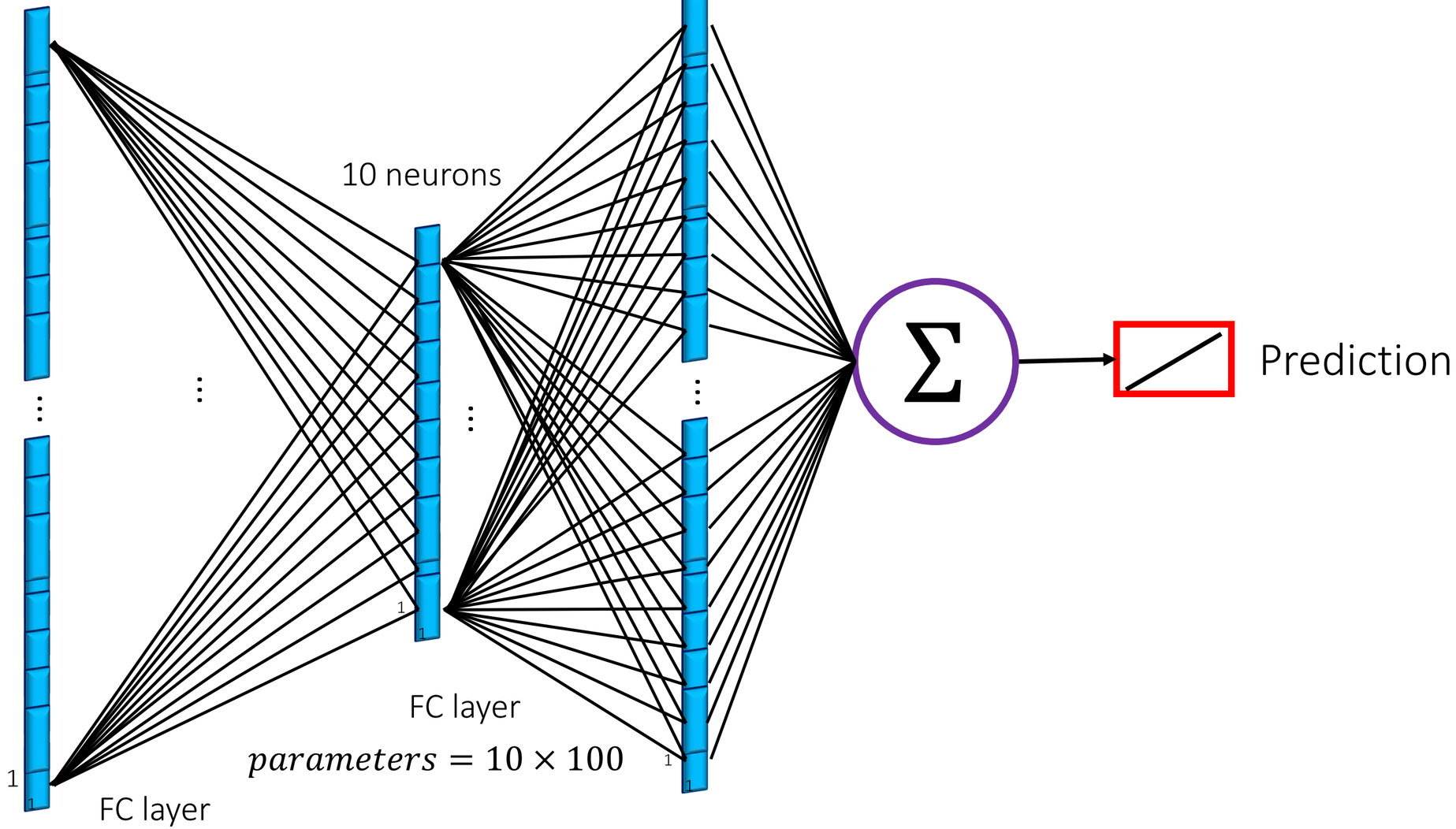
neurons after flattening: $50 \times 5 \times 5 = 1{,}250$

100 neurons

10 neurons

Flattening





FC layer

$parameters = 1{,}250 \times 10$

FC layer

$parameters = 10 \times 100$

10 CNN Architecture

Example

https://www.cs.ryerson.ca/~aharley/vis/conv/