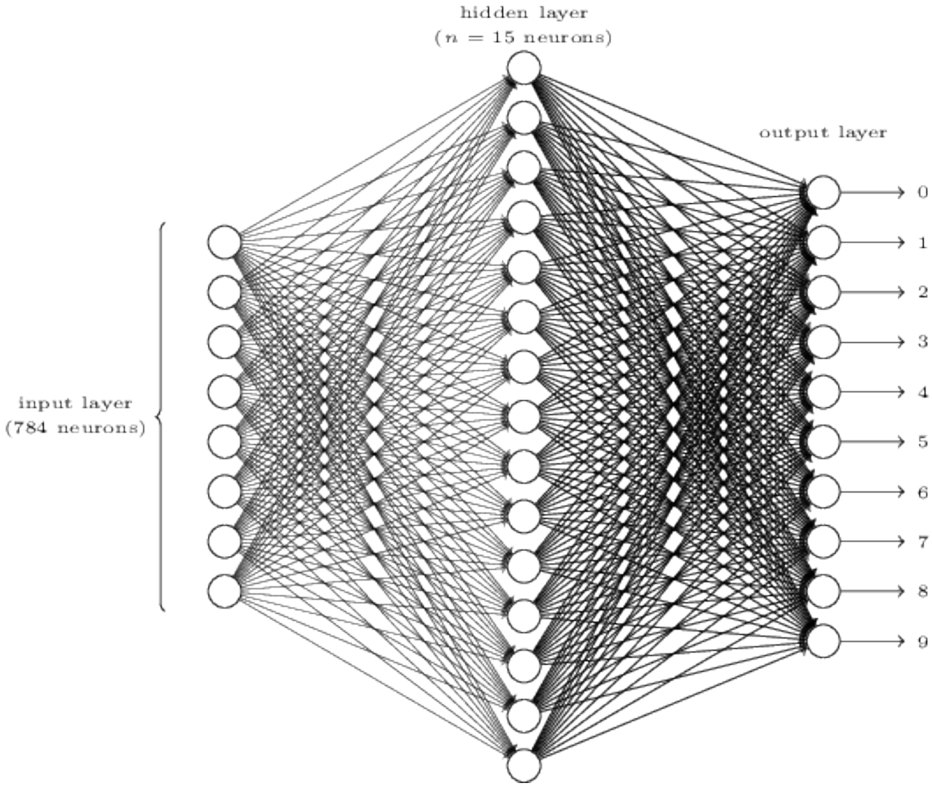
Convolutional Neural Networks

Nail Ibrahimli

Recognizing Digits with Neural Nets.

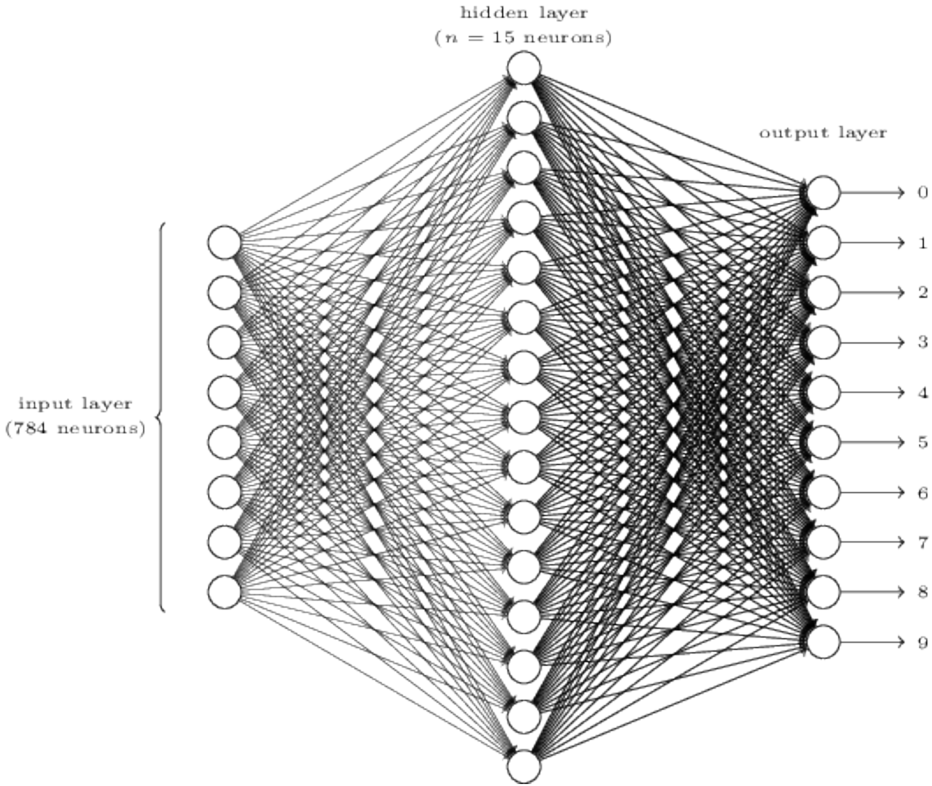


For x representing digit 6:

$$y(x) = (0,0,0,0,0,0,1,0,0,0)^T$$

$$C(w,b) \equiv \frac{1}{2n}\sum_x \|y(x) - a\|^2$$

Complexity of the World:



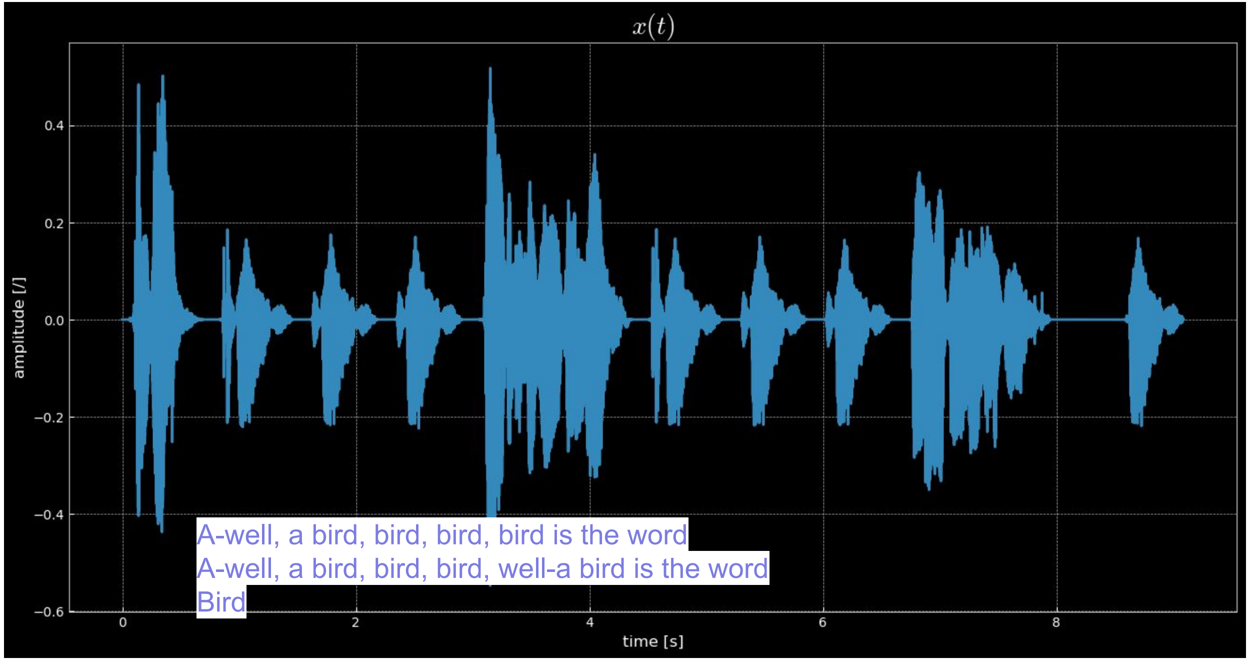
Images have much higher resolutions and input dimensions.

Number of the categories and classes are usually much more than 10.

Sound signal

Sound signal

Sound signal



A-well, a bird, bird, bird, bird is the word
A-well, a bird, bird, bird, well-a bird is the word
Bird

Image signals - locality

Image signals - locality

Image signals - locality

Image signals - locality

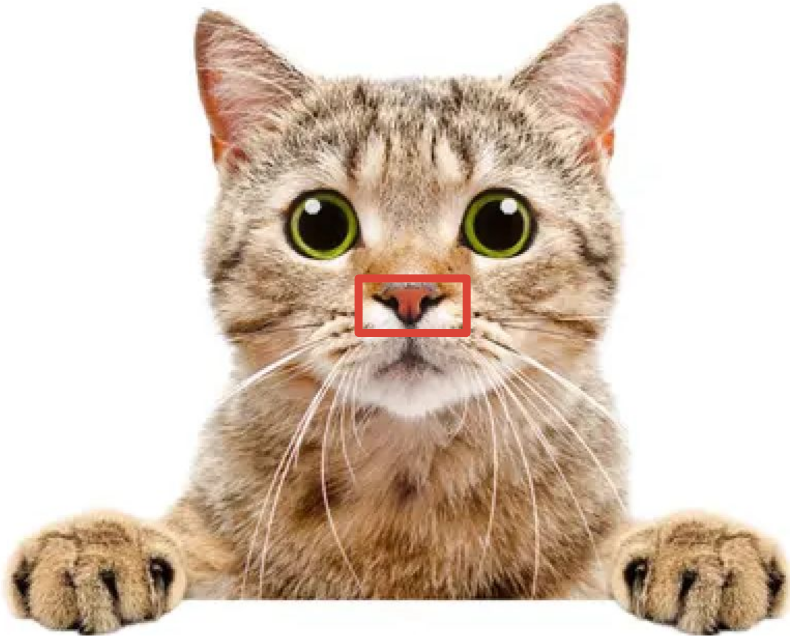


Image signals - stationarity

Image credits: Andy Warhol

Image signals - stationarity

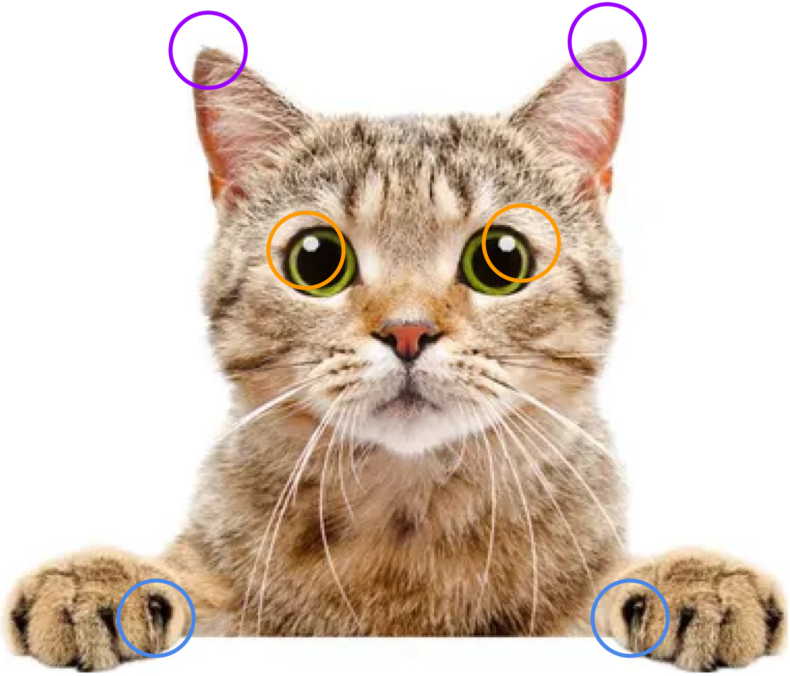


Image signals - compositionality

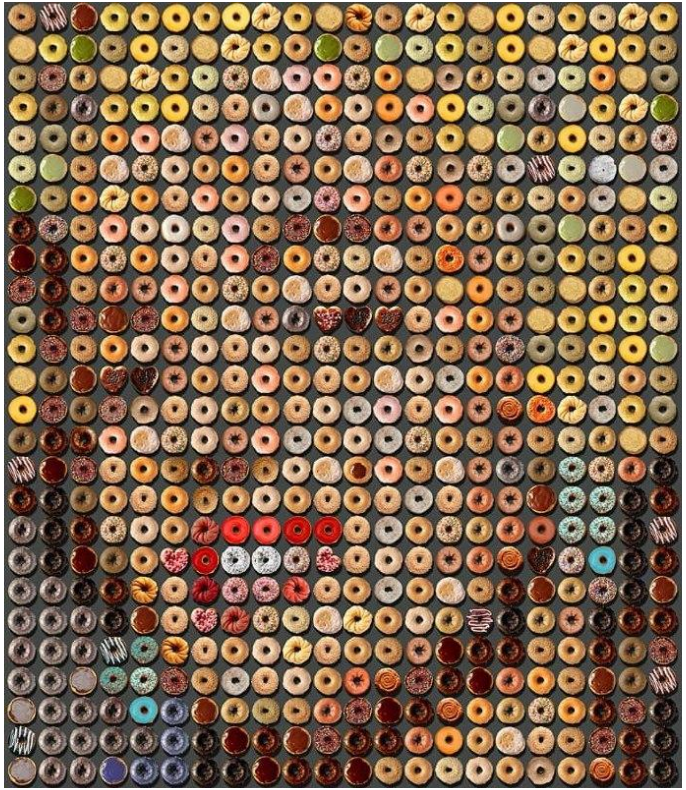



Image signals - compositionality

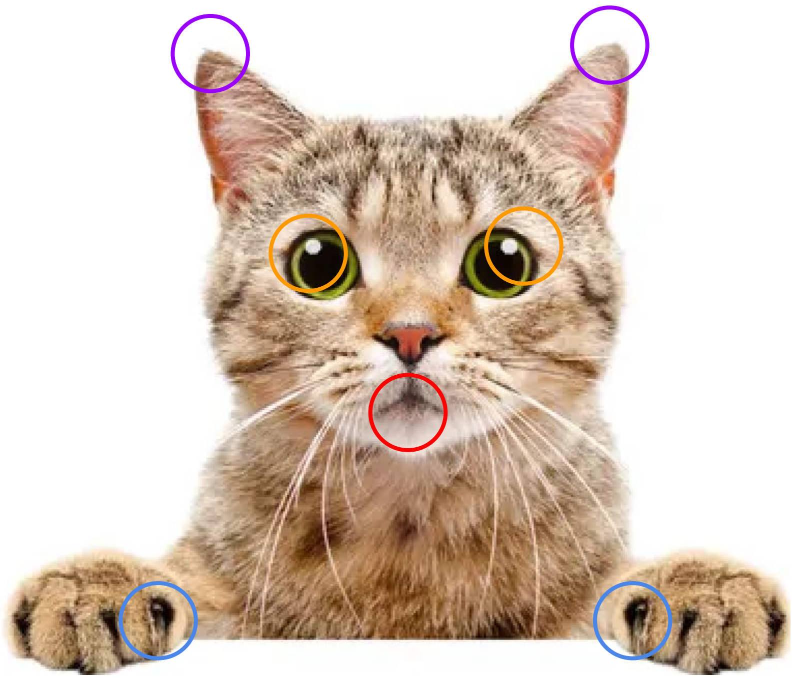


Image signals properties

- Locality - neighboring pixels are correlated

- Stationarity - similar features can occur multiple times in different positions in the image plane

- Compositionality - natural images are composed of features

One dimensional convolution

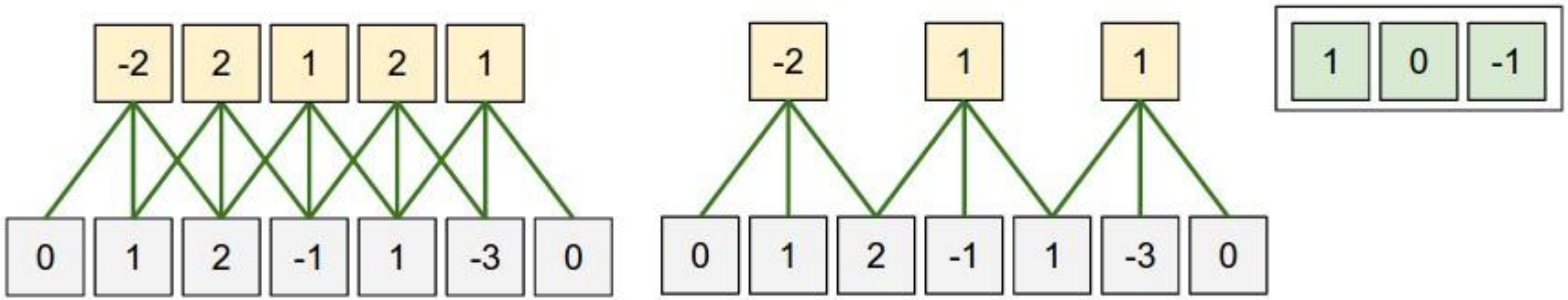
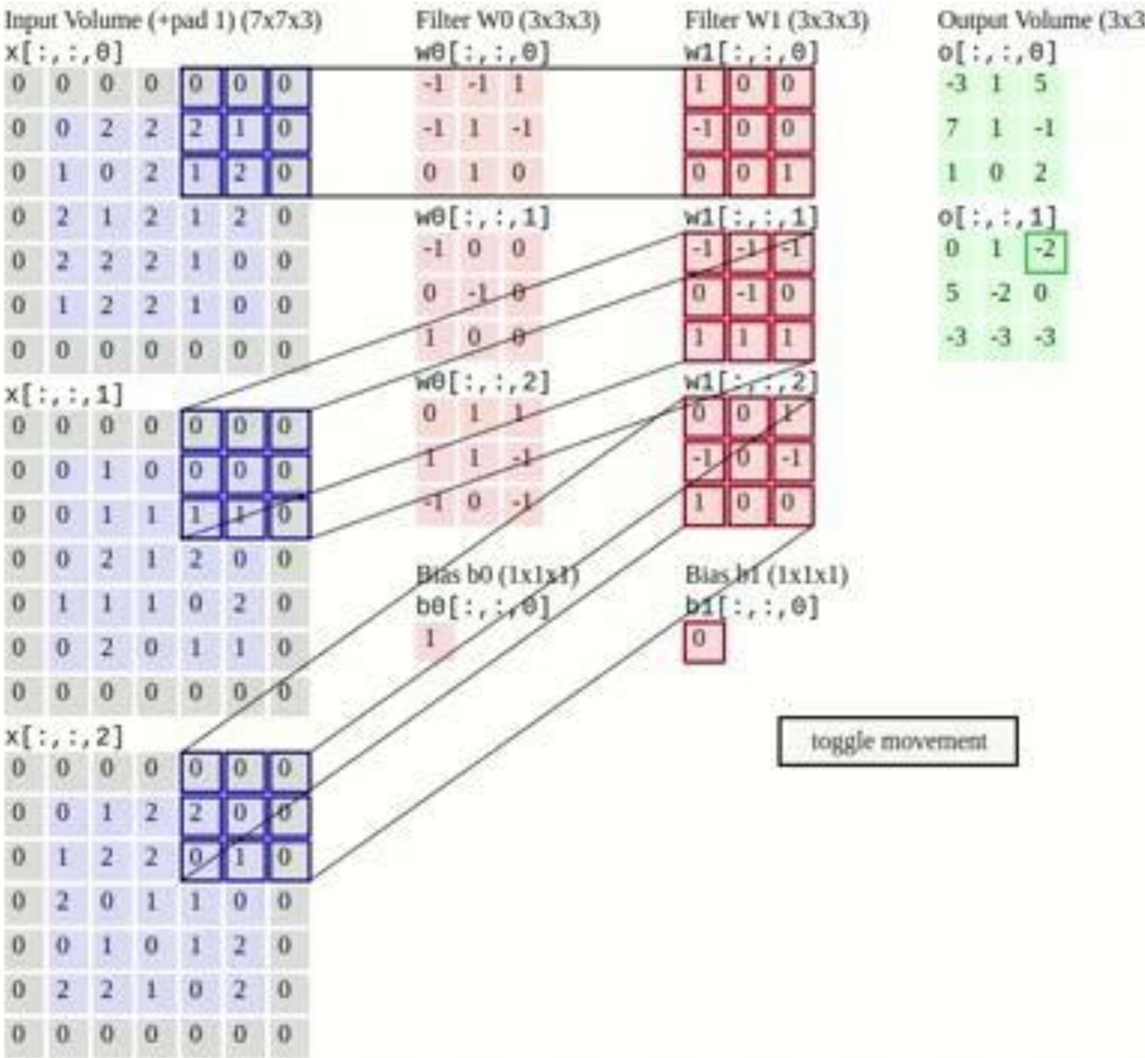


Image credits: Stanford CS 231n

Image convolution



Kernels

Input image

Convolution Kernel

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Feature map

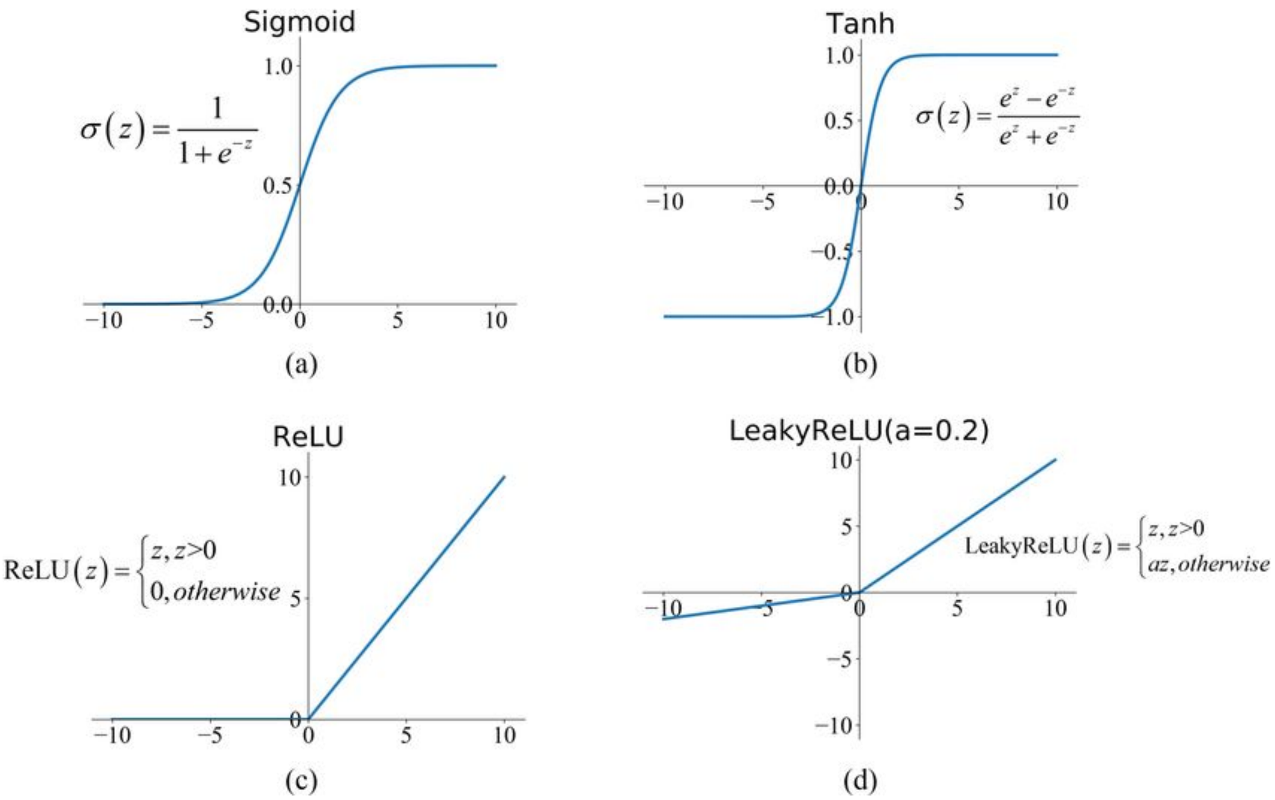
Input image

Kernel

$$\begin{pmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{pmatrix}$$

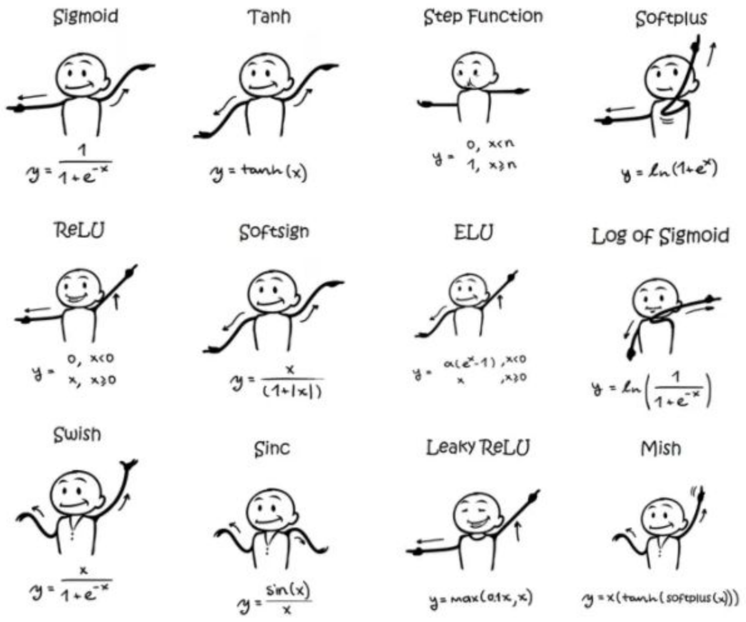
Feature map

Activations

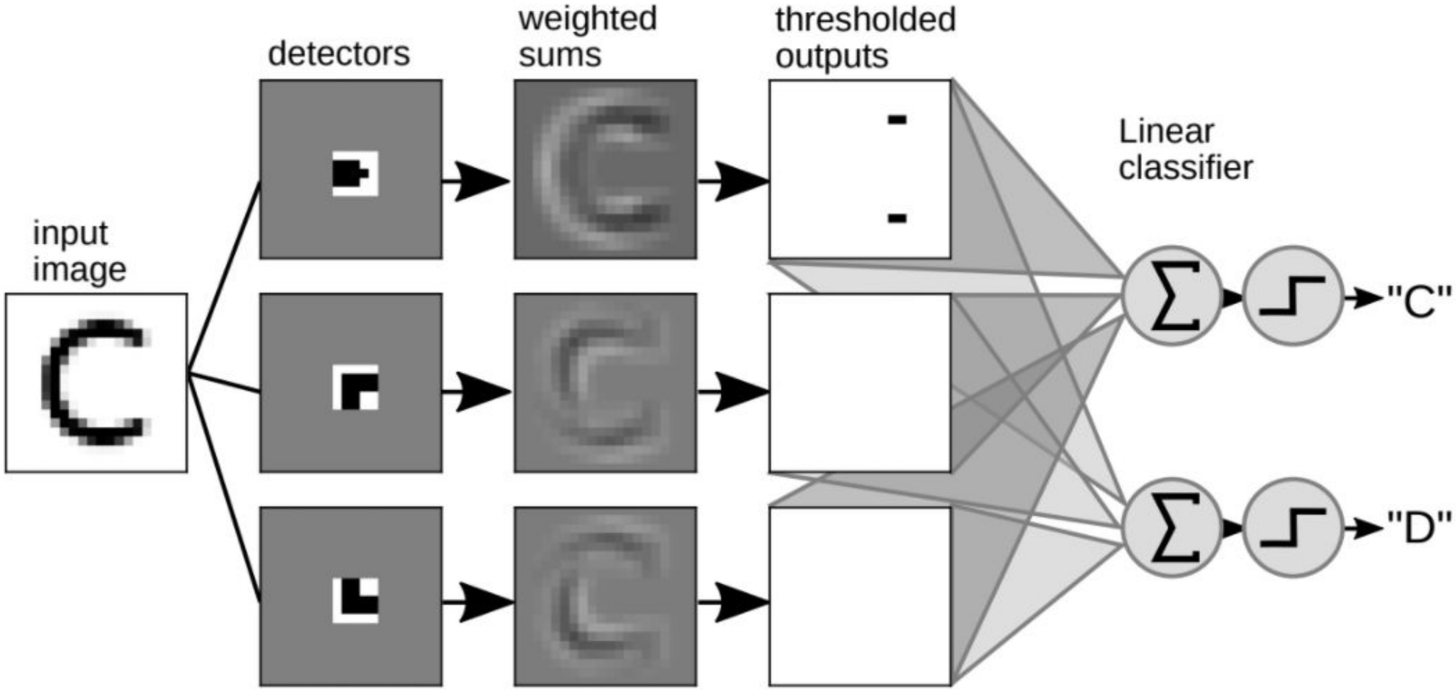


Activations

[PyTorch activation functions]

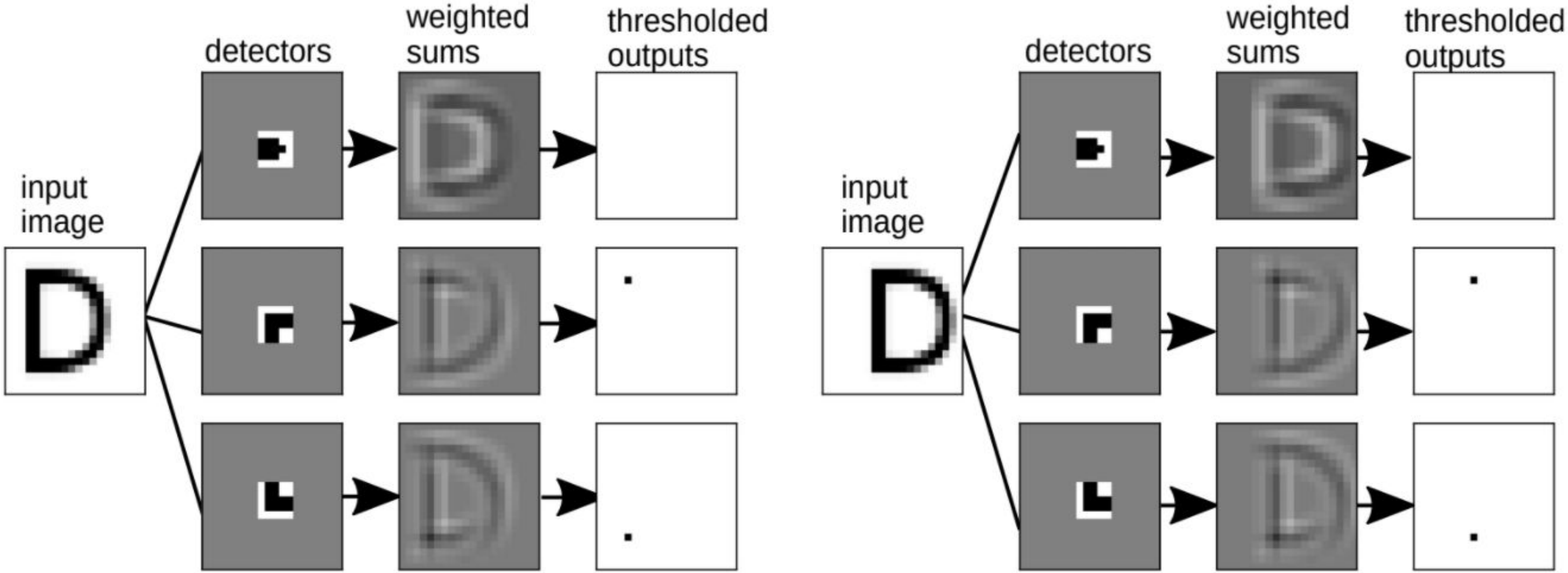


Convolution motivation



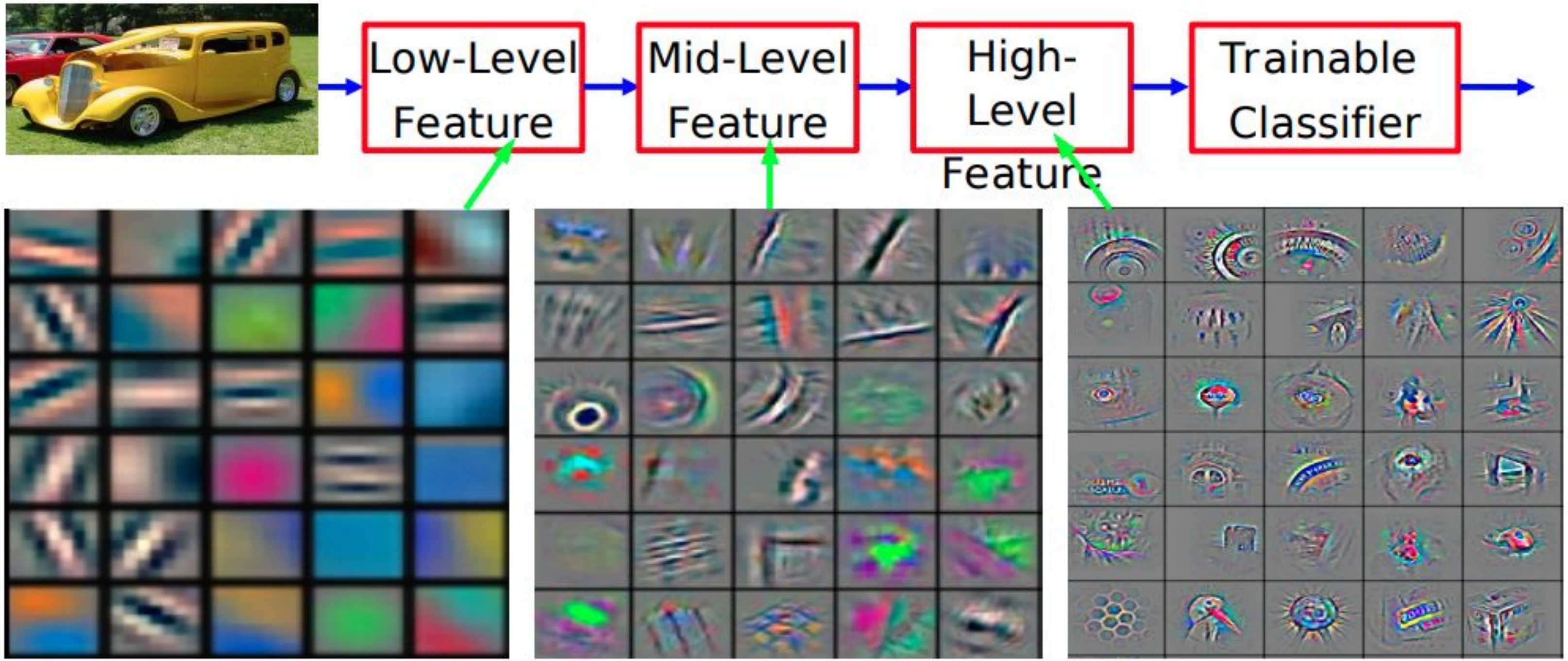
Slide credit: Yann Lecun

Convolution motivation



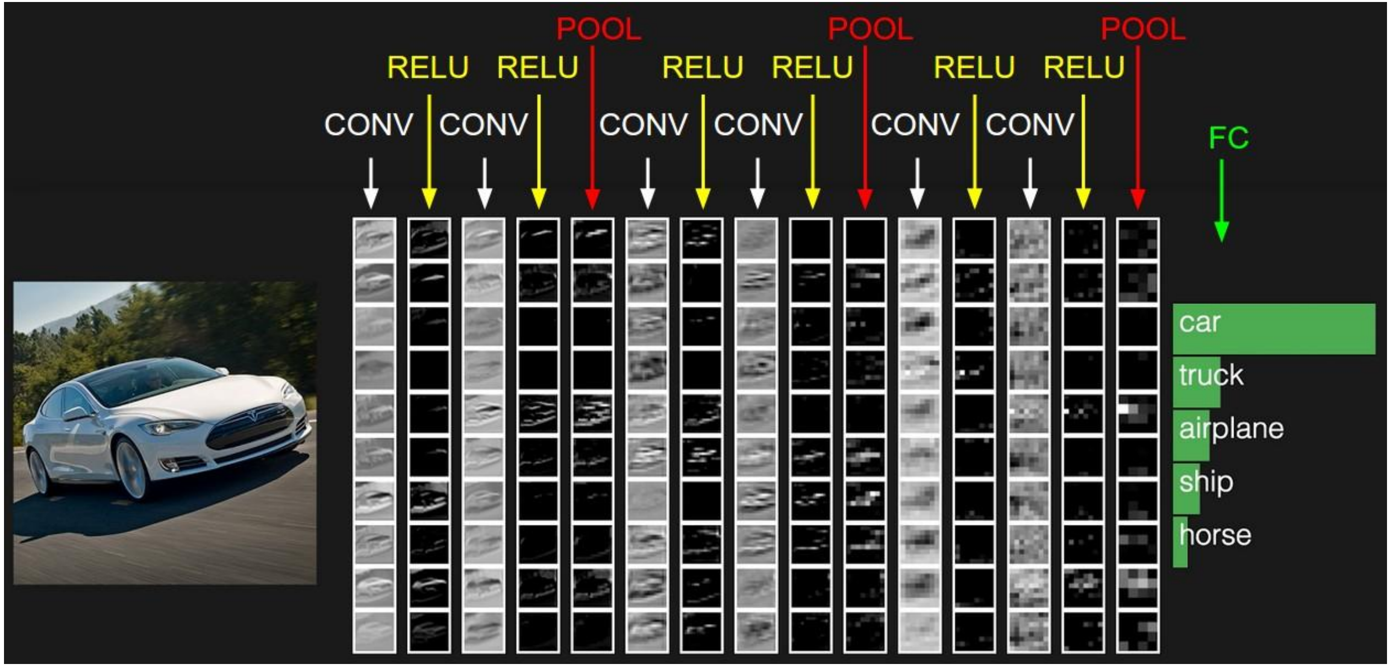
Slide credit: Yann Lecun

Convolutional features



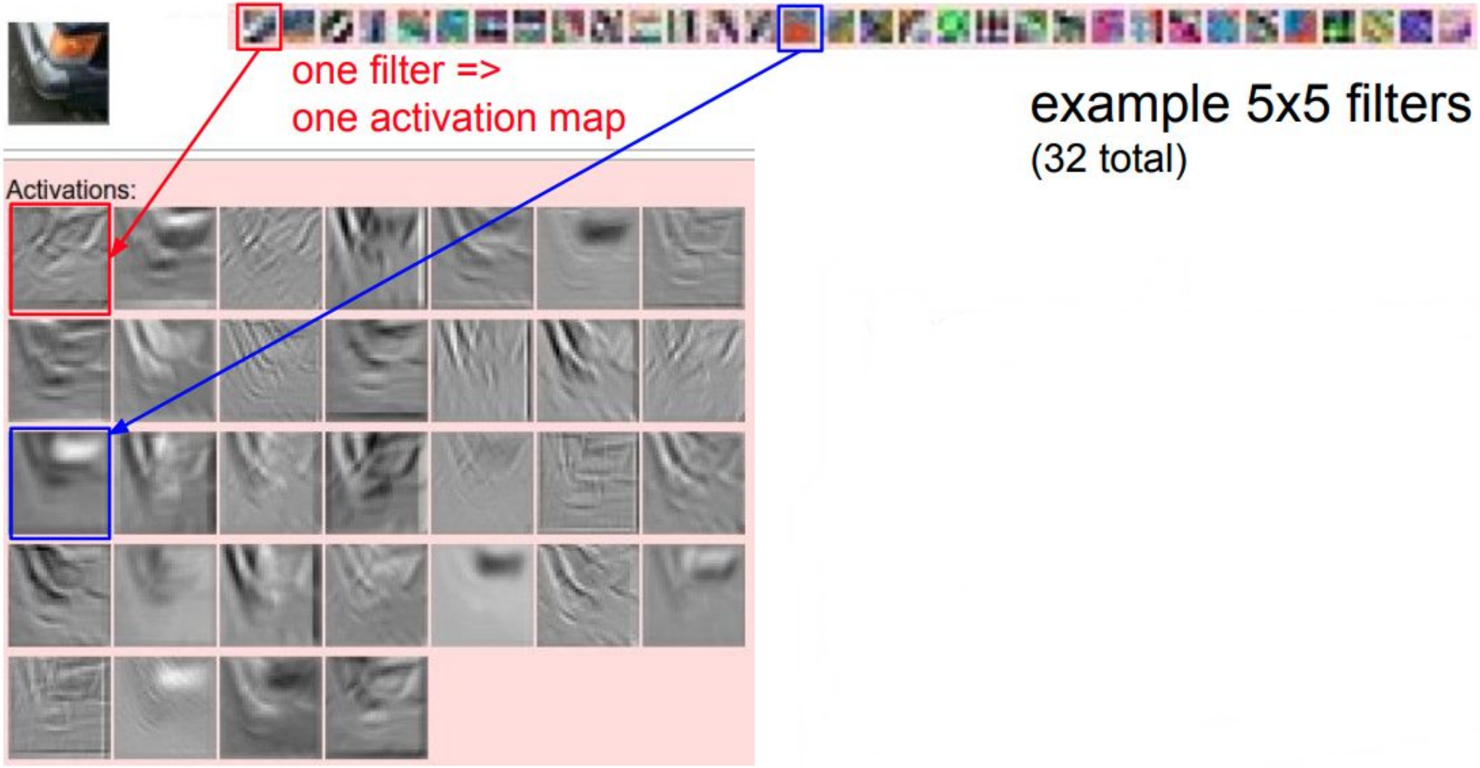
Slide credit: Yann Lecun
Image credit: Visualizing and Understanding Convolutional Networks (Zeiler & Fergus, 2013)

Convolutional features



Slide credit: Andrej Karpathy

Convolutional kernels



example 5x5 filters
(32 total)

Image credits: Stanford CS 231n

Convolutional low-level features

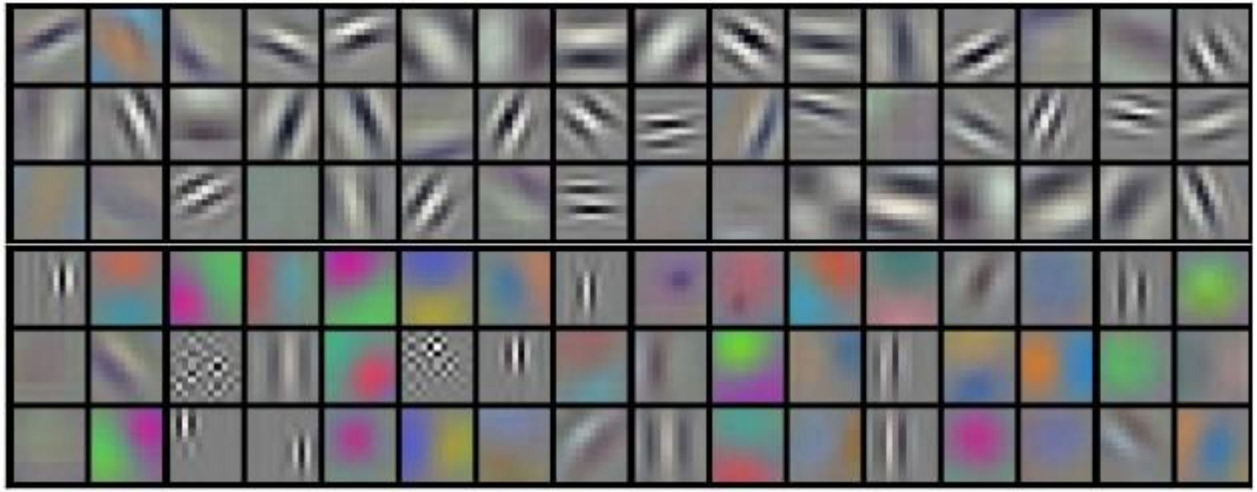
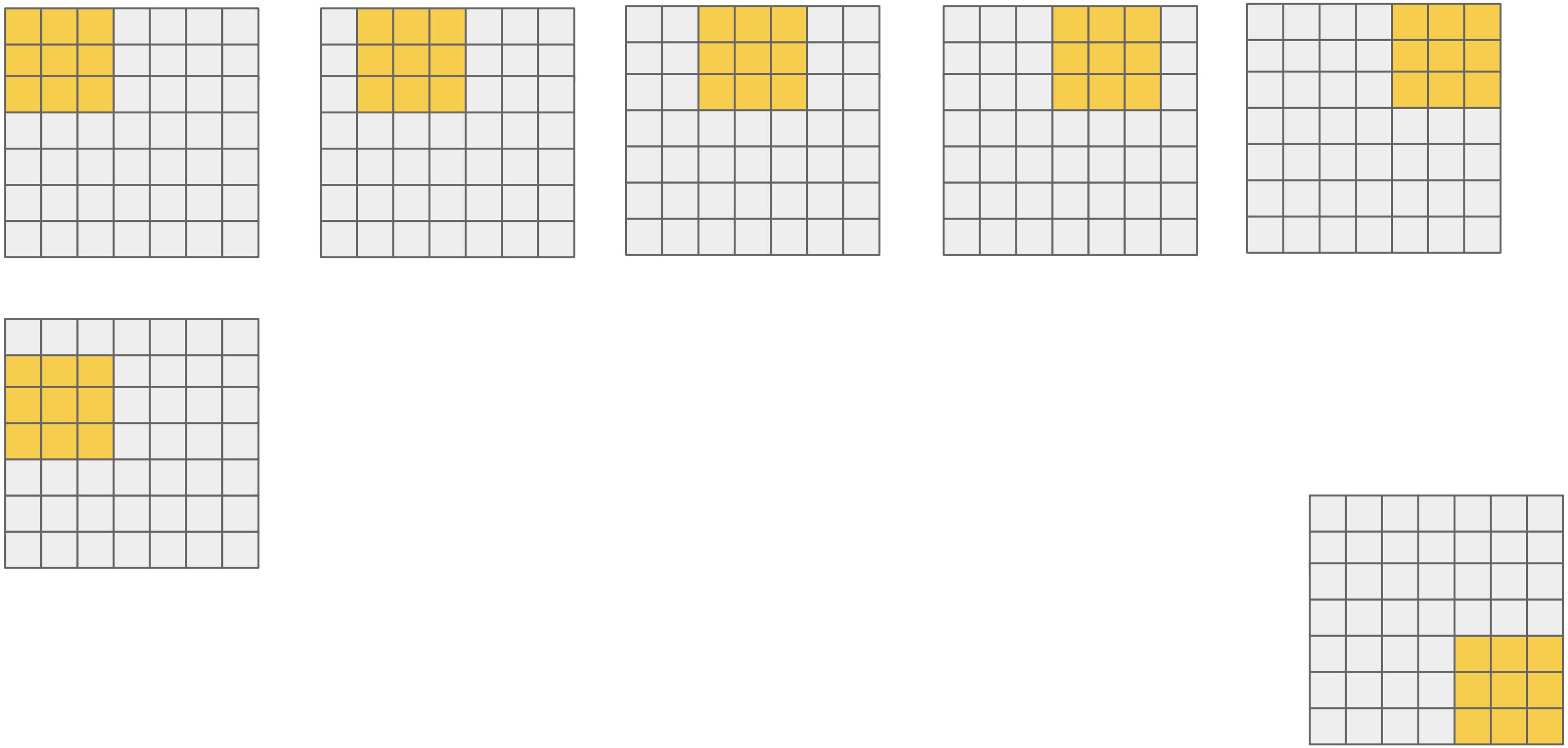


Image credit: Stanford CS231n

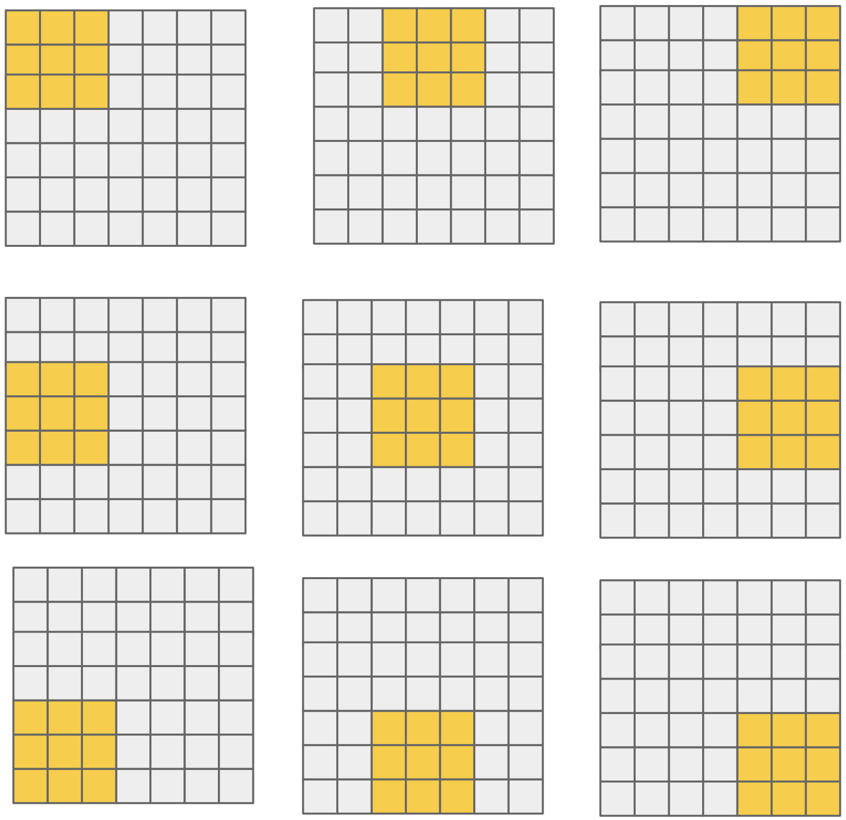
Convolution operation

N=7,F=3,S=1



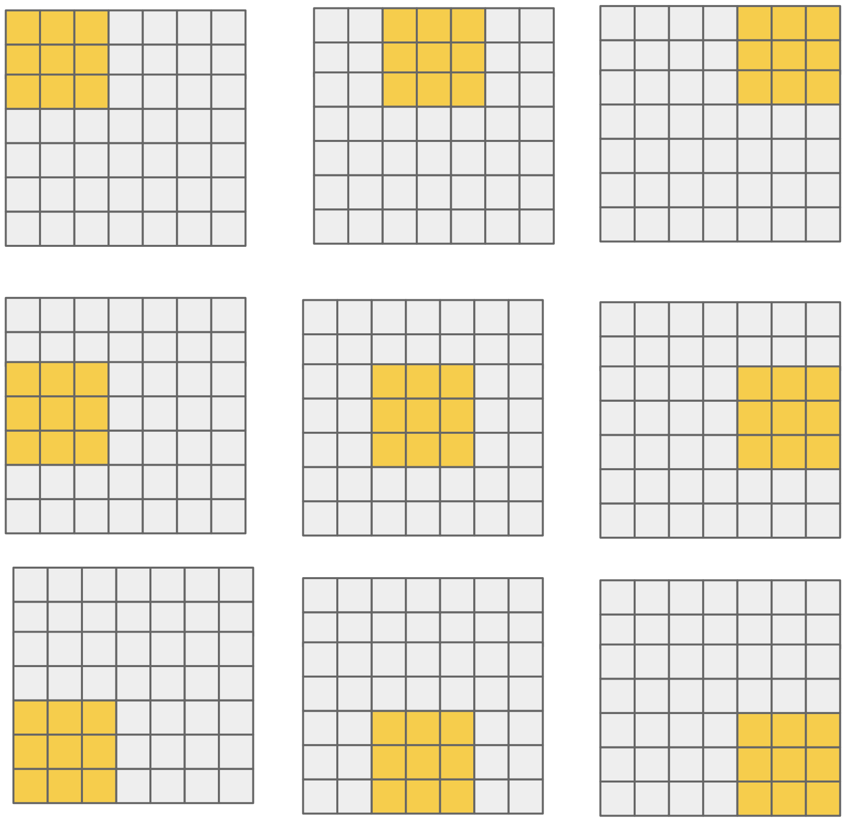
Convolution operation

N=7,F=3,S=2



Convolution operation

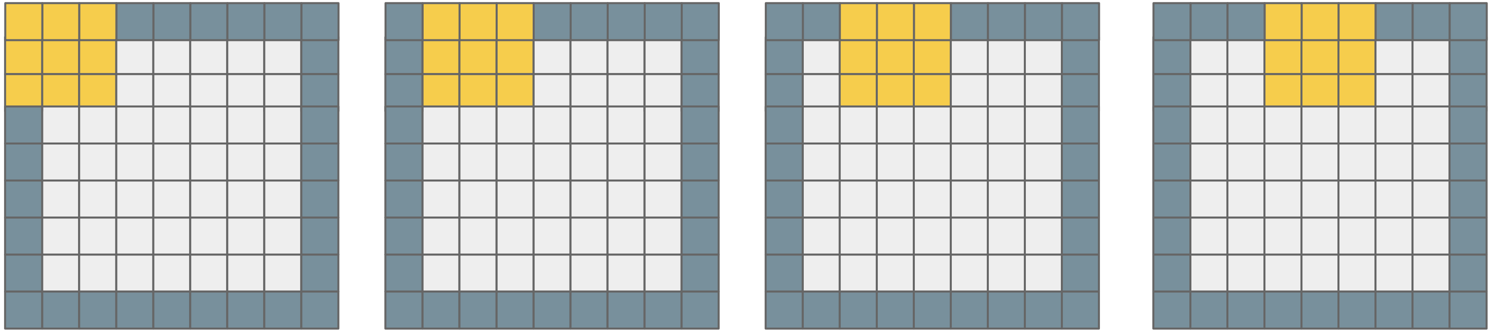
N=7,F=3,S=2

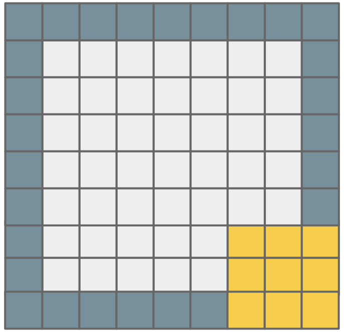


Output = (N-F)/S+1

Convolution operation

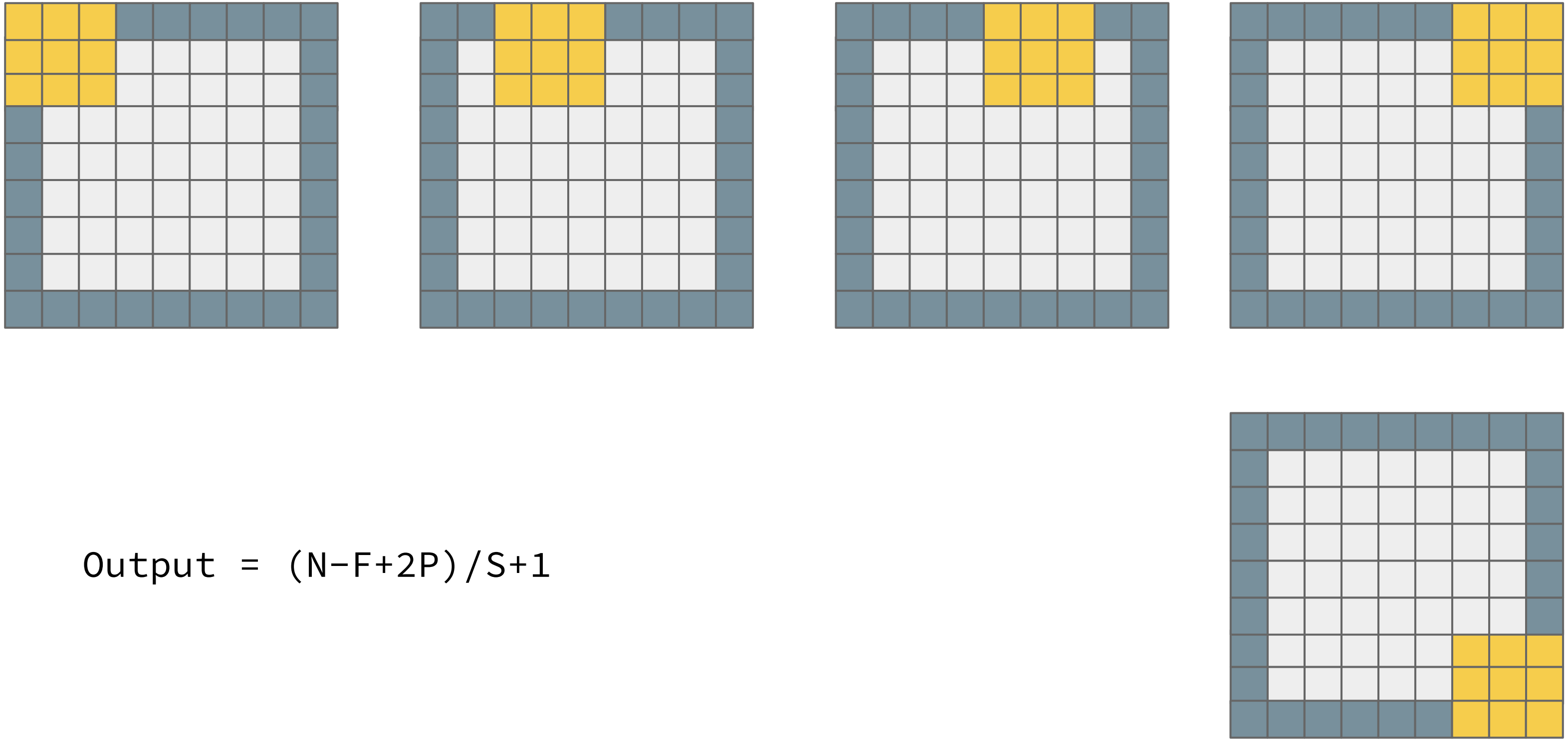
N=7, F=3, S=1, P=1





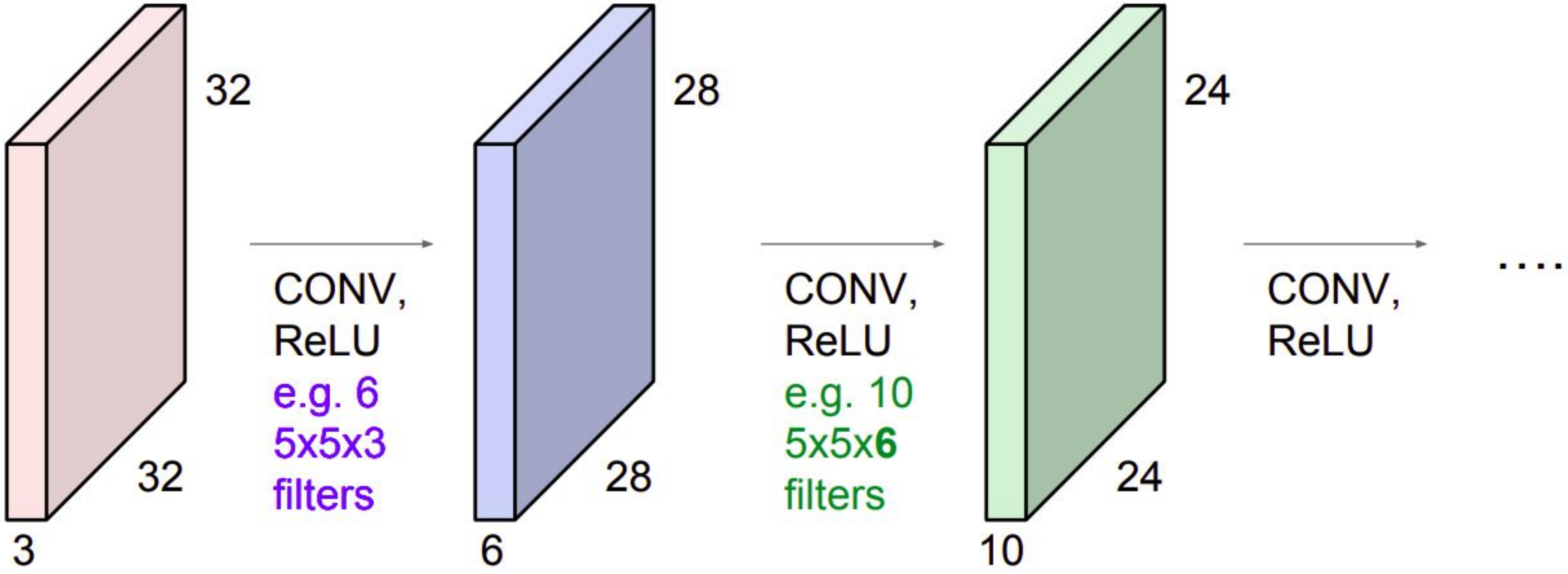
Convolution operation

N=7 F=3, S=2, P=1

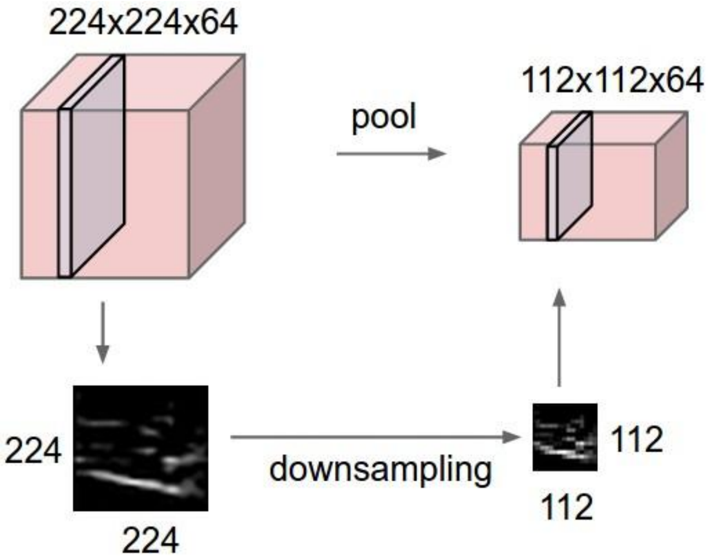


Output = (N-F+2P)/S+1

Number of parameters

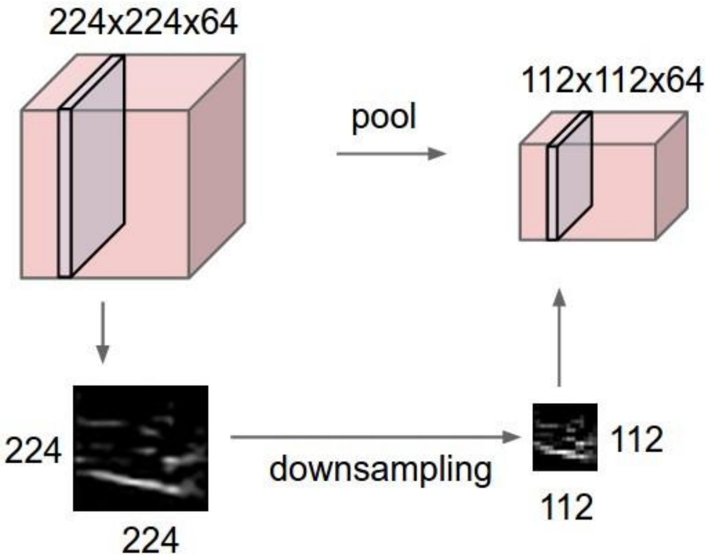


Pooling layer



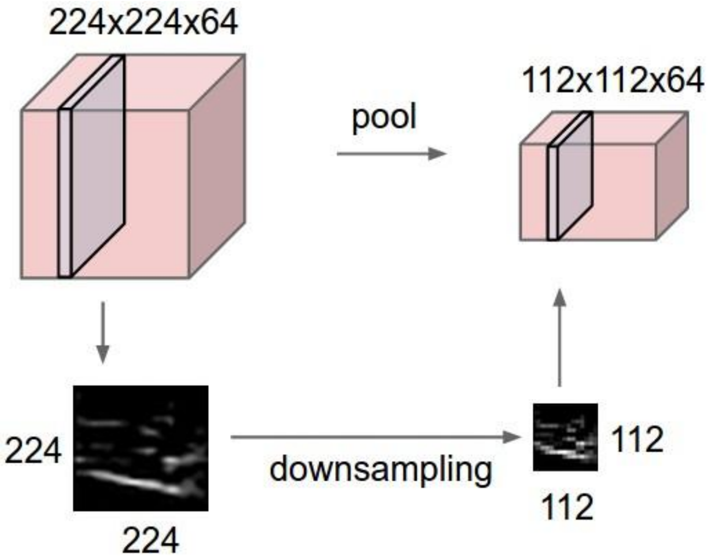
Advantages?

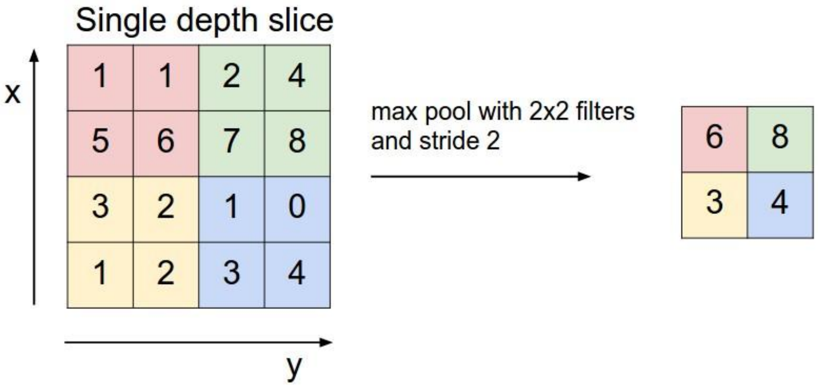
Pooling layer



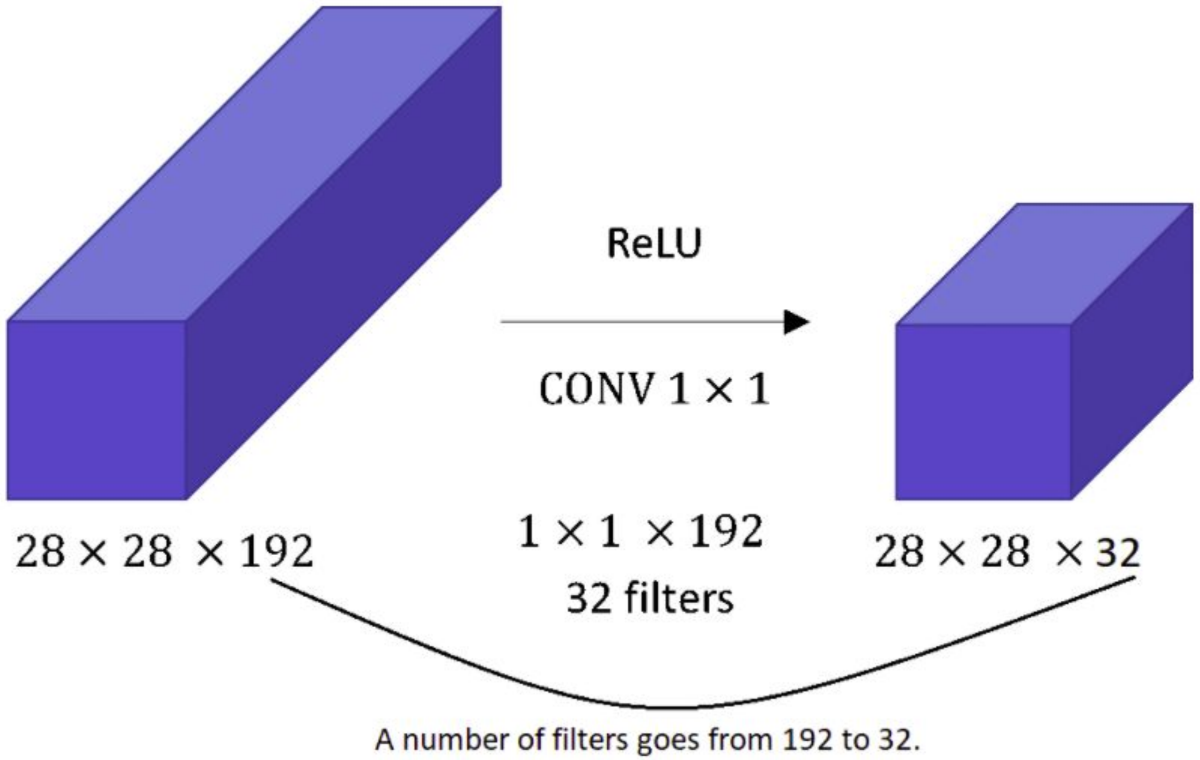
- reduce the spatial size of the representation
- control overfitting.

Pooling layer (Maxpool)

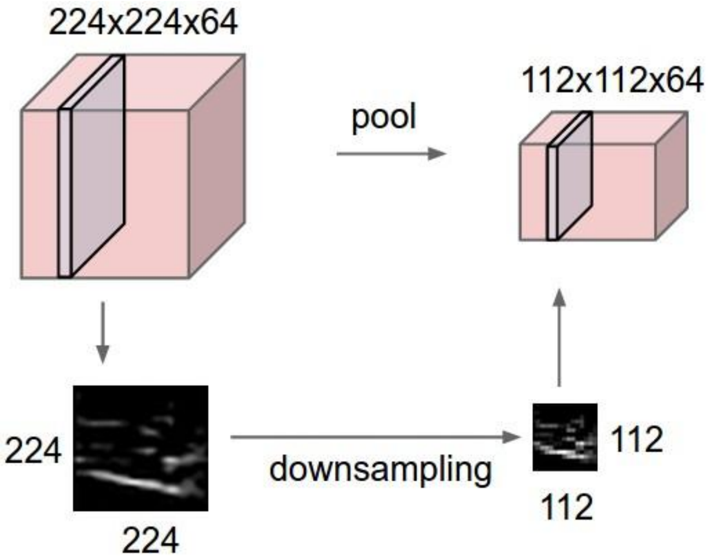


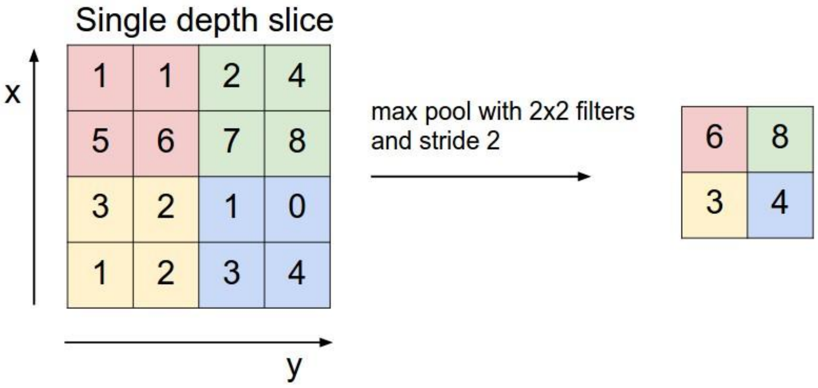


1x1 Convolutions

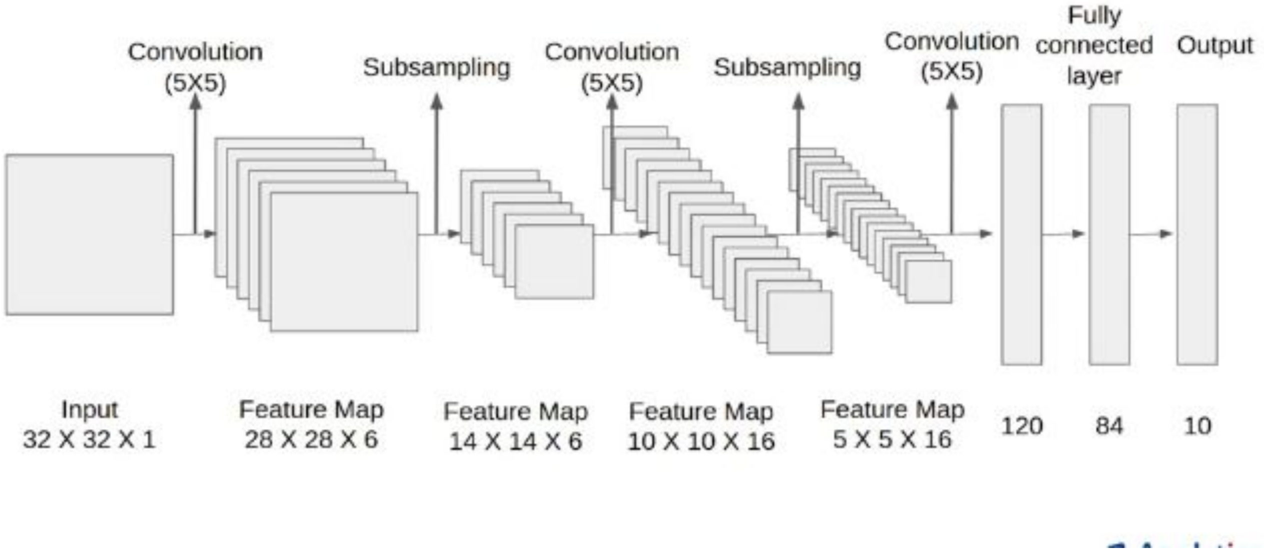


Pooling layer (Maxpool)





LeNet5



Convolution (5X5)
Subsampling
Convolution (5X5)
Subsampling
Convolution (5X5)
Fully connected layer
Output
Input 32 X 32 X 1
Feature Map 28 X 28 X 6
Feature Map 14 X 14 X 6
Feature Map 10 X 10 X 16
Feature Map 5 X 5 X 16
120
84
10

Lenet5

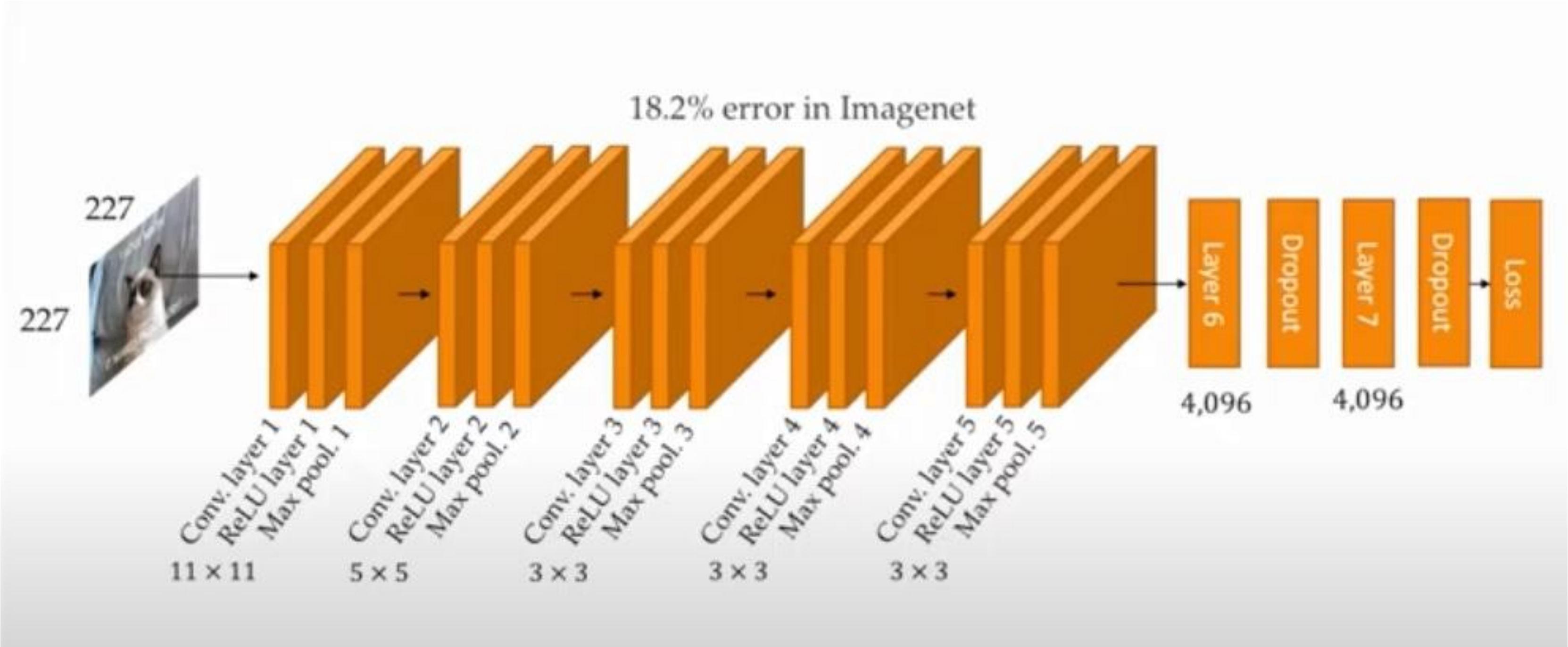
Credit: Yann Lecun

Lenet5


```python
class LeNet5(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(1, 20, 5, 1)
        self.conv2 = nn.Conv2d(20, 20, 5, 1)
        self.fc1 = nn.Linear(4*4*20, 500)
        self.fc2 = nn.Linear(500, 10)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = F.max_pool2d(x, 2, 2)
        x = F.relu(self.conv2(x))
        x = F.max_pool2d(x, 2, 2)
        x = x.view(-1, 4*4*20)
        x = F.relu(self.fc1)
        x = self.fc2(x)
        return F.logsoftmax(x, dim=1)
```

AlexNet



ImageNet2021