PyTorch

卷积神经网络

主讲人：龙良曲

Convolution

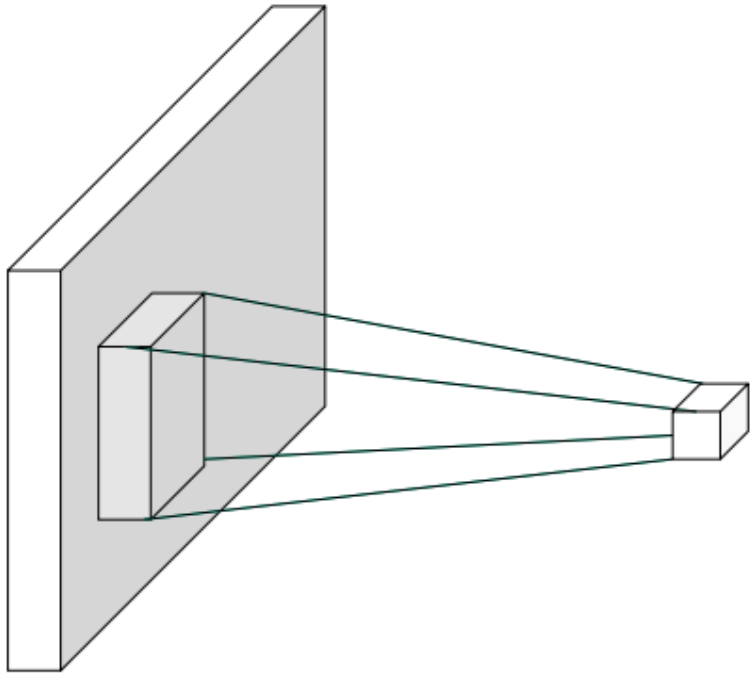
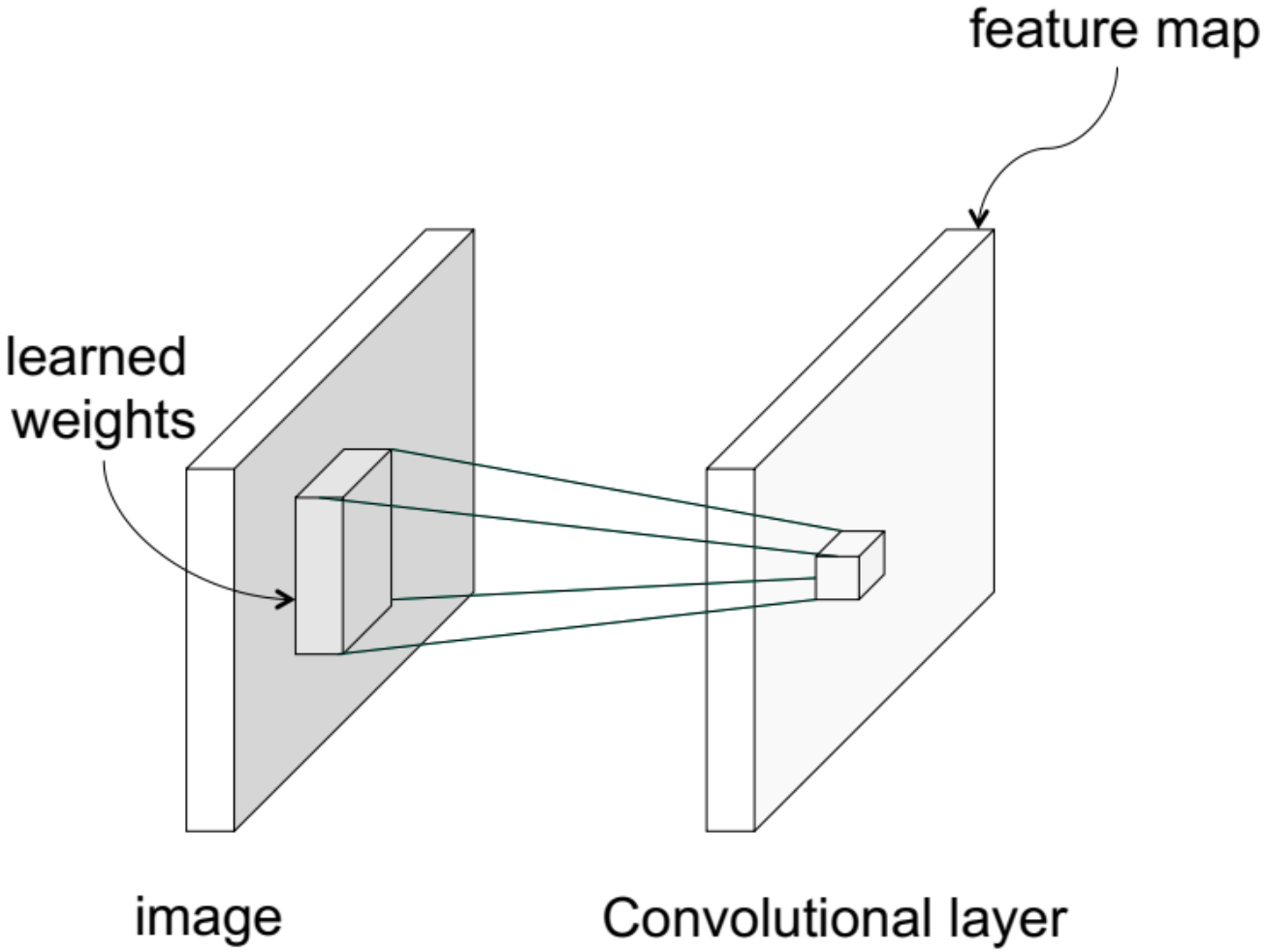
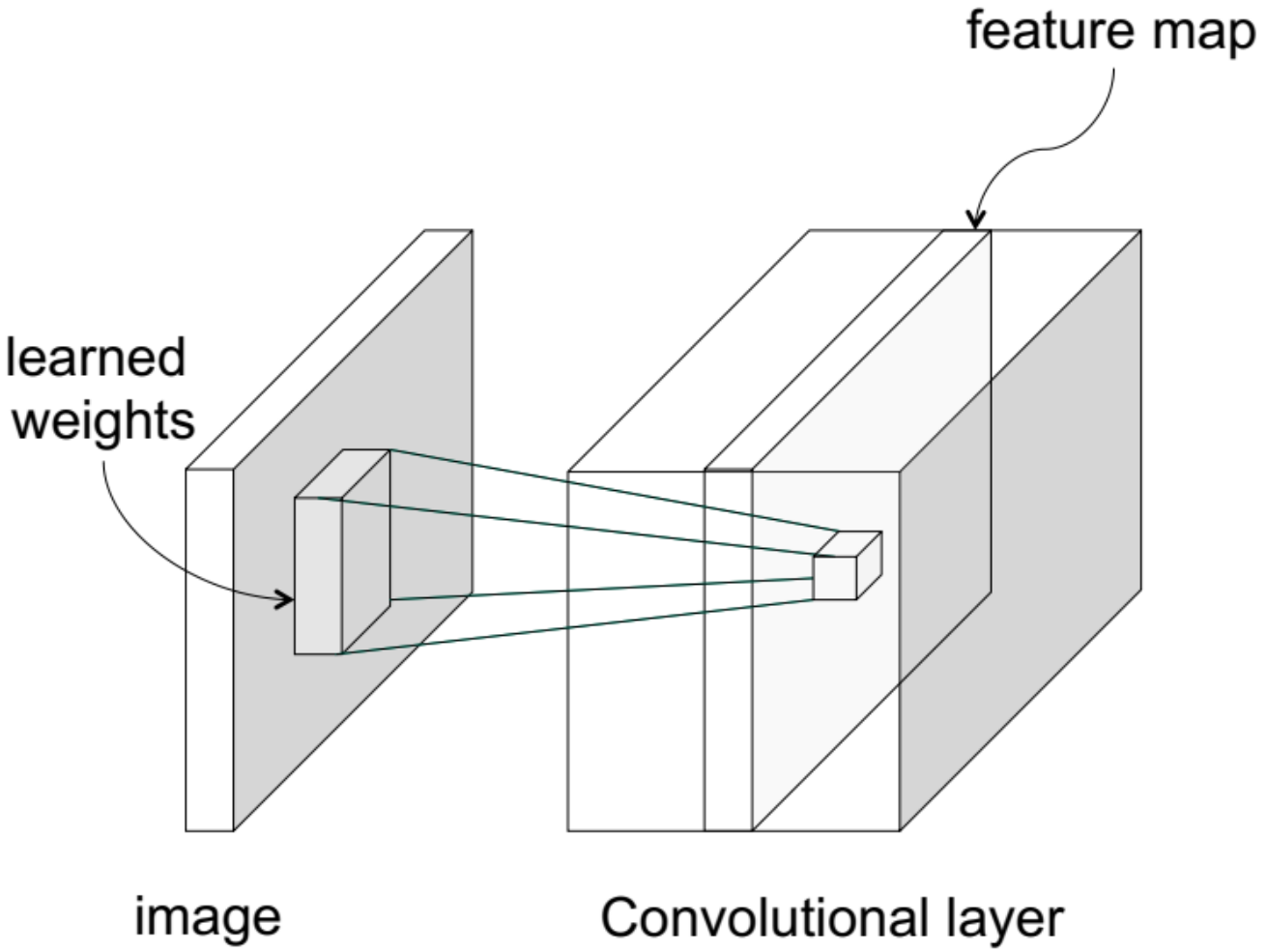


image Convolutional layer

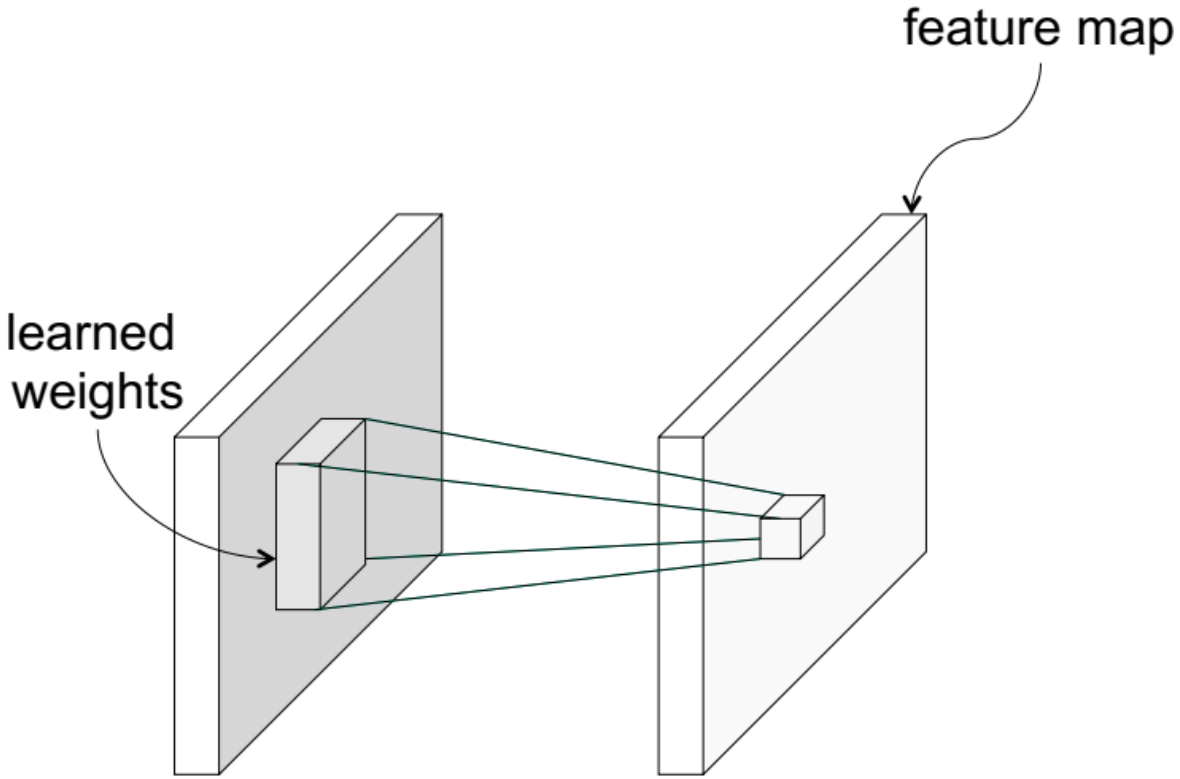
Moving window



Several kernels



Animation

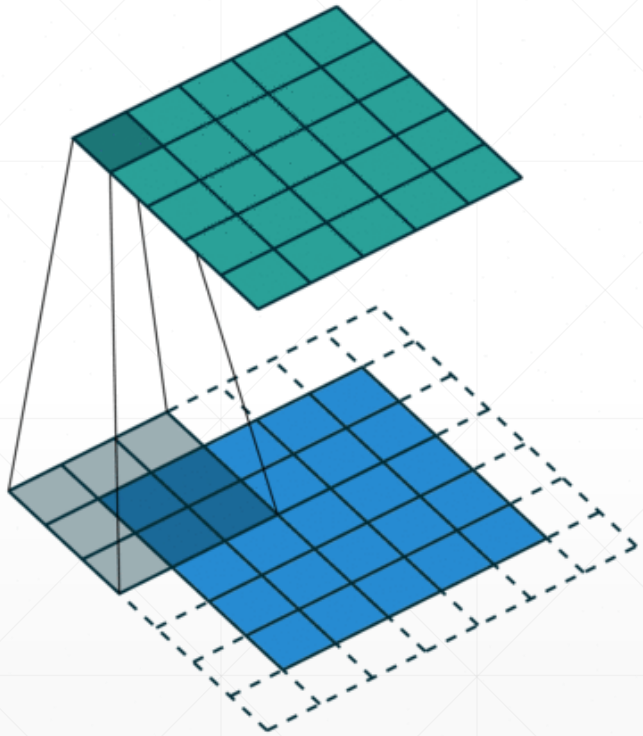


1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		

https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050

Notation



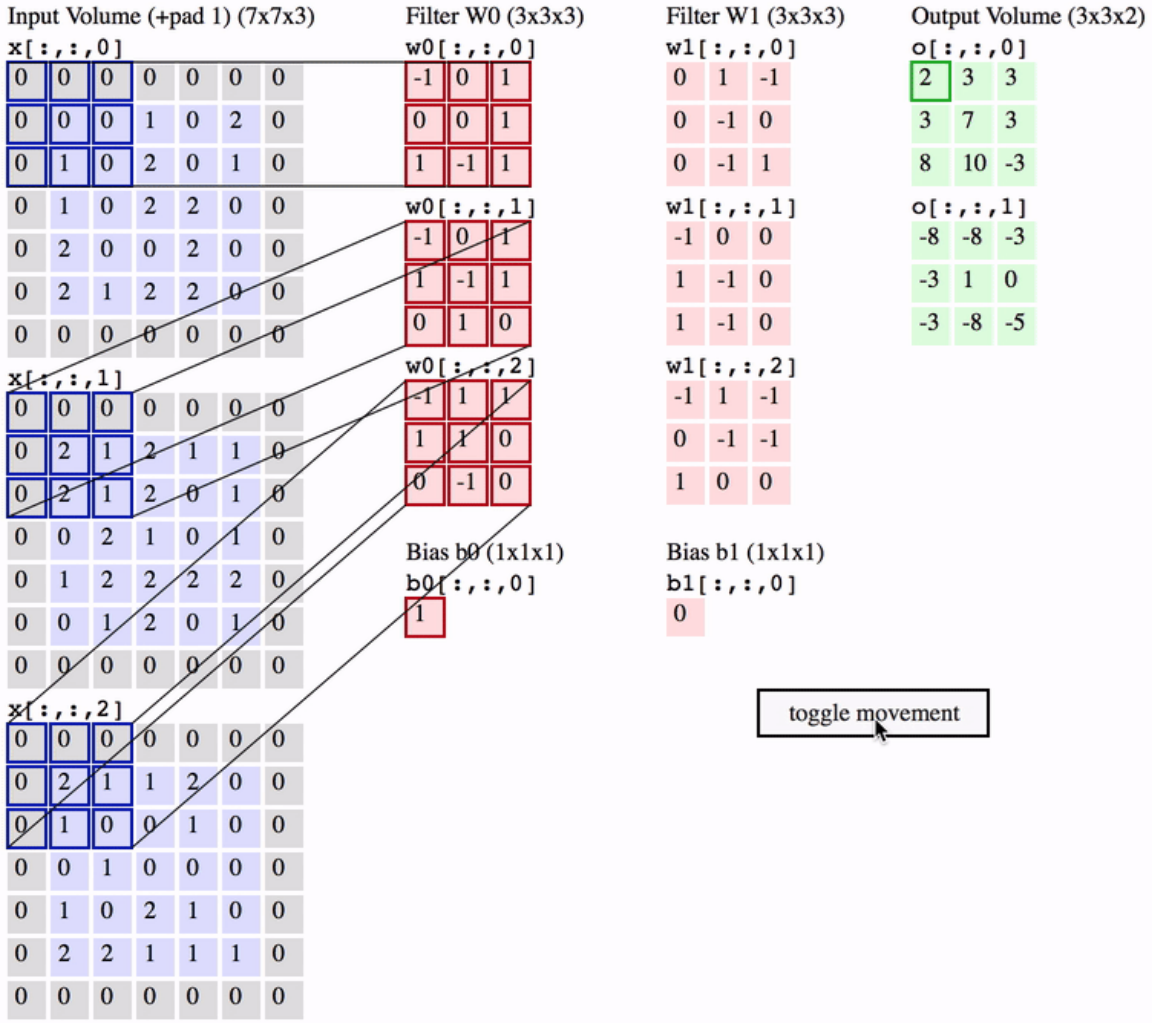
Input_channels:
Kernel_channels: 2 ch
Kernel_size:
Stride:
Padding:

- Input: $(N, C_{in}, H_{in}, W_{in})$
- Output: $(N, C_{out}, H_{out}, W_{out})$ where

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times \text{padding}[0] - \text{dilation}[0] \times (\text{kernel\_size}[0] - 1) - 1}{\text{stride}[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times \text{padding}[1] - \text{dilation}[1] \times (\text{kernel\_size}[1] - 1) - 1}{\text{stride}[1]} + 1 \right\rfloor$$

Multi-Kernels



x: [b, 3, 28, 28]

one k: [3, 3, 3]

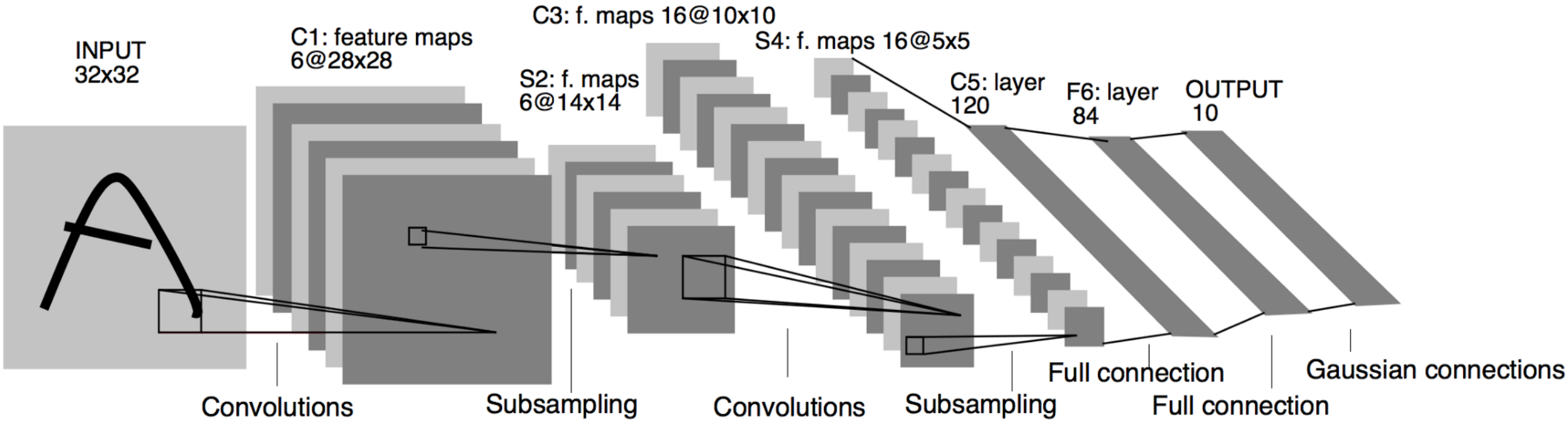
multi-k: [16, 3, 3, 3]

bias: [16]

out: [b, 16, 28, 28]

https://skymind.ai/wiki/convolutional-network

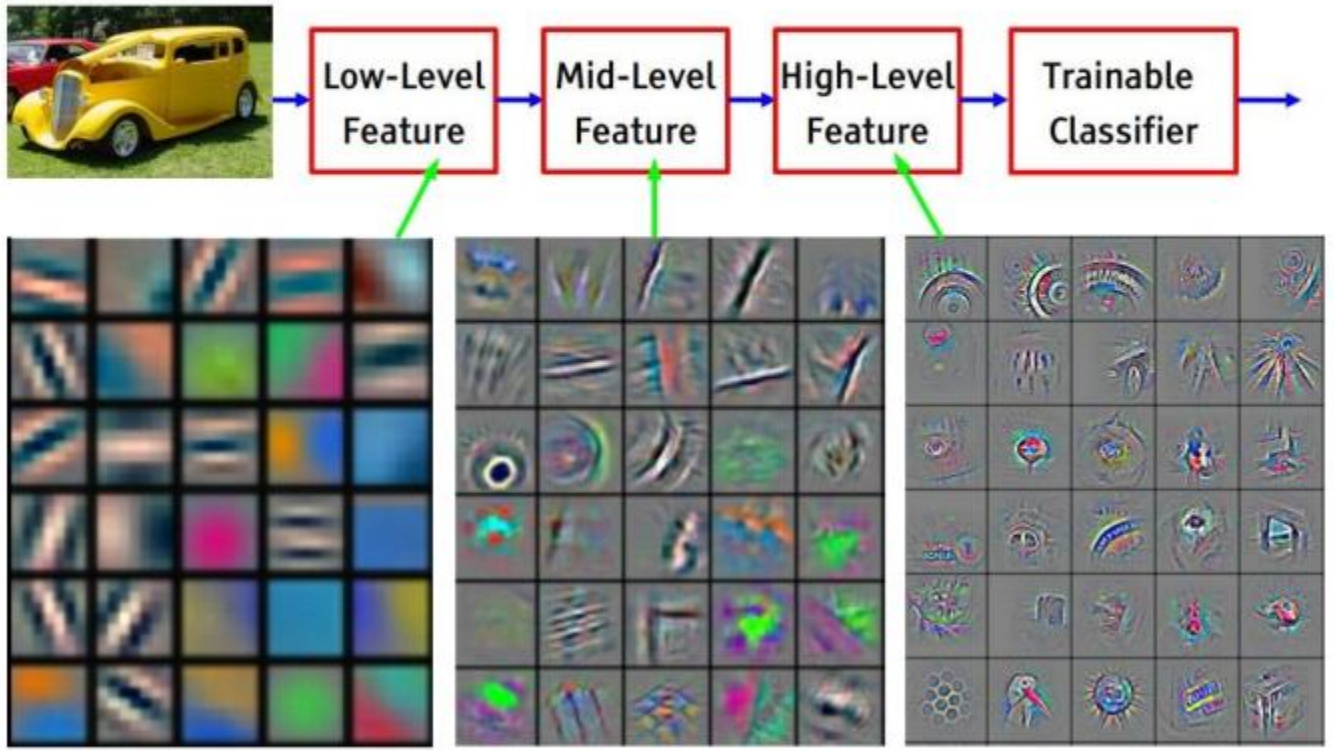
LeNet-5


INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

Pyramid Architecture

Preview

[From recent Yann LeCun slides]



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

http://cs231n.stanford.edu/slides/2016/winter1516_lecture7.pdf

nn.Conv2d

```
In [9]: layer=nn.Conv2d(1,3,kernel_size=3,stride=1,padding=0)
In [10]: x=torch.rand(1,1,28,28)

In [11]: out=layer.forward(x)
Out[12]: torch.Size([1, 3, 26, 26])

In [13]: layer=nn.Conv2d(1,3,kernel_size=3,stride=1,padding=1)
In [14]: out=layer.forward(x)
Out[15]: torch.Size([1, 3, 28, 28])

In [16]: layer=nn.Conv2d(1,3,kernel_size=3,stride=2,padding=1)
In [17]: out=layer.forward(x)
Out[18]: torch.Size([1, 3, 14, 14])

In [19]: out=layer(x) #__call__
Out[20]: torch.Size([1, 3, 14, 14])
```

Inner weight & bias

```
In [21]: layer.weight
Parameter containing:
tensor([[[[ 0.2727, -0.0923, -0.1530],
          [-0.0664,  0.2896,  0.0593],
          [-0.1967,  0.2786,  0.3163]]],


        [[[ 0.0825,  0.1090,  0.1183],
          [ 0.0857, -0.3036, -0.2539],
          [-0.3169,  0.0118, -0.2634]]],


        [[[ 0.1211,  0.1331, -0.2639],
          [ 0.3033,  0.1766, -0.0017],
          [-0.2050, -0.0187, -0.2170]]]], requires_grad=True)

In [22]: layer.weight.shape
Out[22]: torch.Size([3, 1, 3, 3])

In [23]: layer.bias.shape
Out[23]: torch.Size([3])
```

F.conv2d

```
In [24]: w=torch.rand(16,3,5,5)
In [25]: b=torch.rand(16)

In [26]: out=F.conv2d(x,w,b,stride=1,padding=1)
---------------------------------------------------------------------------
RuntimeError: Given groups=1, weight of size [16, 3, 5, 5],
expected input[1, 1, 28, 28] to have 3 channels, but got 1 channels instead

In [27]: x=torch.randn(1,3,28,28)

In [28]: out=F.conv2d(x,w,b,stride=1,padding=1)
Out[29]: torch.Size([1, 16, 26, 26])

In [30]: out=F.conv2d(x,w,b,stride=2,padding=2)
Out[31]: torch.Size([1, 16, 14, 14])
```

下一课时

池化层

Thank You.