复习

- 卷积运算：

输入

1	2	0	3
0	1	2	3
1	2	1	0
5	2	3	1

*

卷积核W

1	2	1
0	1	0
2	1	0

偏置b=1

= ?

同时，取P=1, S=1（进行宽度为1的零填充，步长为1）

复习

- 卷积的输出特征尺寸计算：

$$\left\lfloor \frac{M+2P-K}{S} \right\rfloor + 1$$

  - 输入（二维）特征的长、宽均为10；
  - 卷积核（二维）的长、宽均为5；
  - Padding=1，Stride=2；
  - 求输出特征的高度、宽度。

- 卷积层参数量、浮点运算量计算：
  - 某卷积层，其输入特征尺寸为32×32，卷积采用1个二维卷积核，其高度、宽度为3，且卷积层无偏置，Padding=1，Stride=2；
  - 求该卷积层可学习参数的数量；
  - 求该卷积层的浮点运算规模。

复习

- 代码填空：用PyTorch创建一个单隐藏层前馈神经网络：
  - 输入层神经元数量：10
  - 隐藏层神经元数量：128
  - 输出层神经元数量：5

```
from torch import nn

net = nn.Sequential(
    nn.Linear(10, 128)    ,
    nn.ReLU(),
    nn.Linear(128, 5)
)
```

复习

- 代码填空：用PyTorch创建一个单隐藏层前馈神经网络

```
from torch import nn
form torch.nn import functional as F

class MLP(nn.Module):
    def __init__(self):
        super().__init__()
        self.hidden = nn.Linear(10, 128)
        self.out = nn.Linear(128, 5)

    def forward(self, X):
        out = self.hidden(X)
        out = F.relu(out)
        out = self.out(out)
        return out
```

复习

- 代码填空：填充代码以完成模型训练。

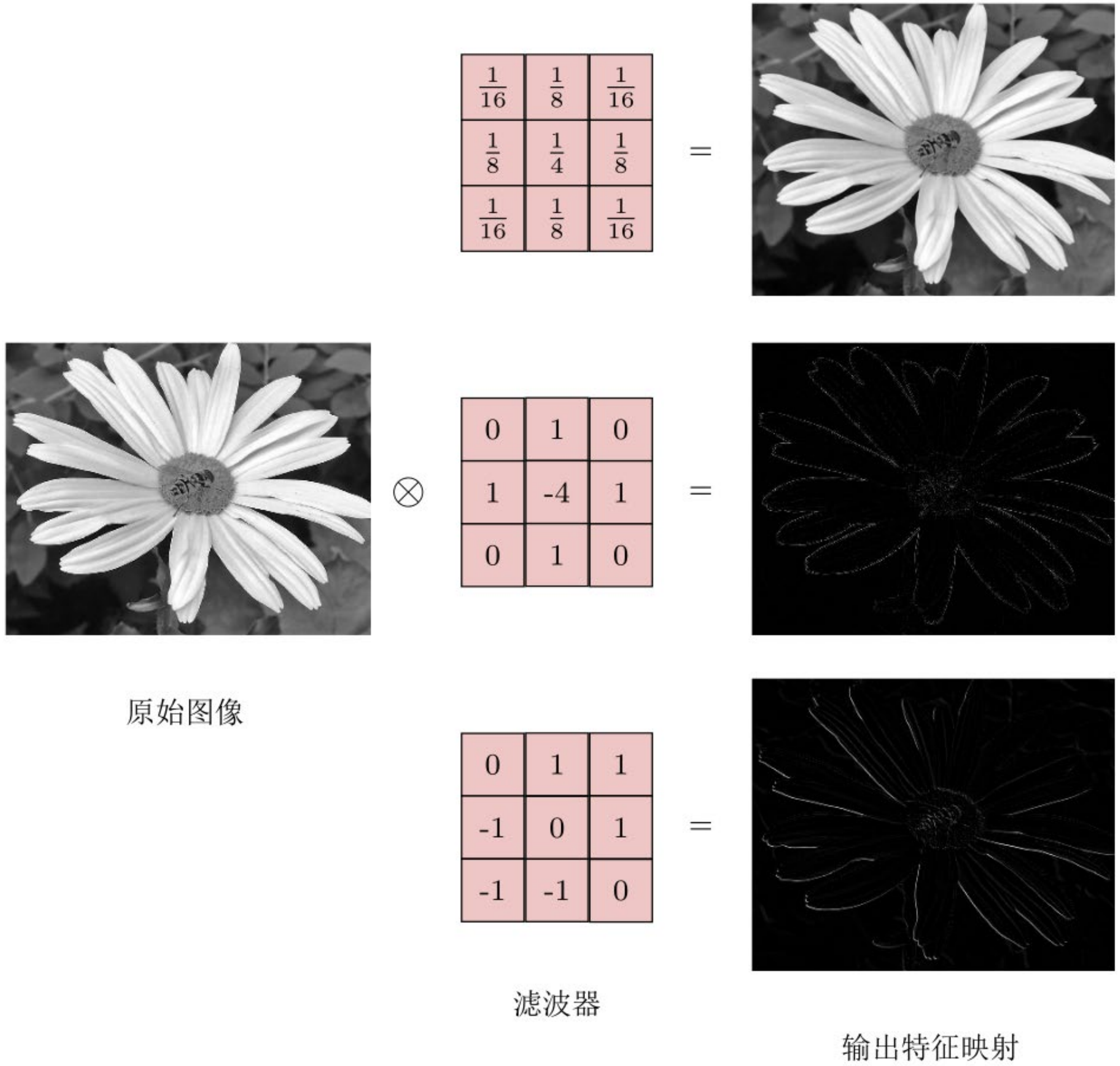
```
from torch import nn
import torch.optim as optim

net = nn.Sequential(nn.Linear(20, 256), nn.ReLU(), nn.Linear(256, 10))
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(net.parameters(), lr=5e-3)

# 假设当前已加载训练数据X, y

for epoch in range(20):
    outputs = net(X)
    loss = criterion(outputs, y)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

卷积作为特征提取器

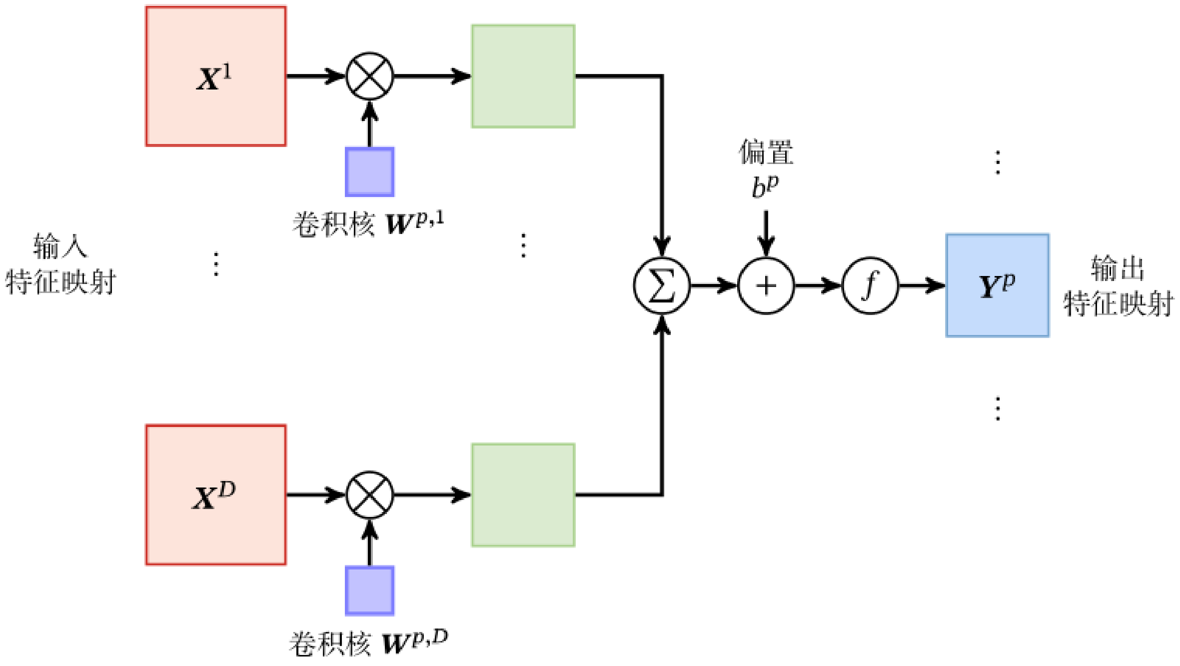


$$\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

卷积层的映射关系

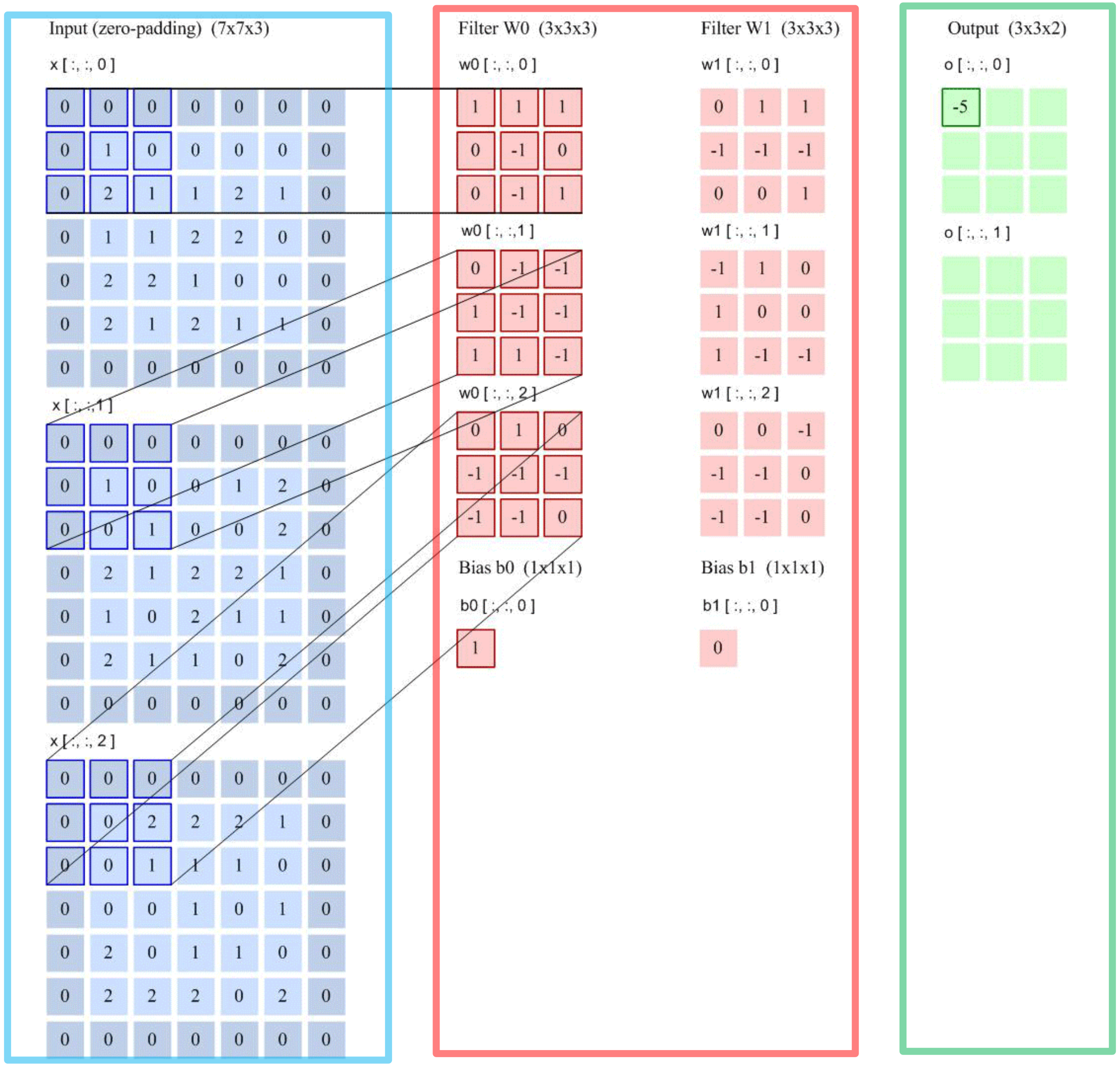
- 特征映射/特征图（Feature Map）：经过卷积后得到的特征。
  - 卷积核看成一个特征提取器
- 卷积层
  - 输入：D个特征映射 $M \times N \times D$
  - 输出：P个特征映射 $M' \times N' \times P$

卷积层的映射关系



$$Z^p = W^p \otimes X + b^p = \sum_{d=1}^{D} W^{p,d} \otimes X^d + b^p,$$

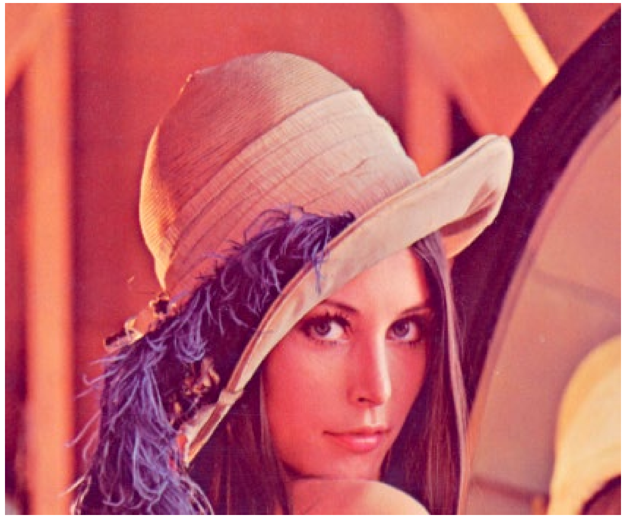
$$Y^p = f(Z^p).$$

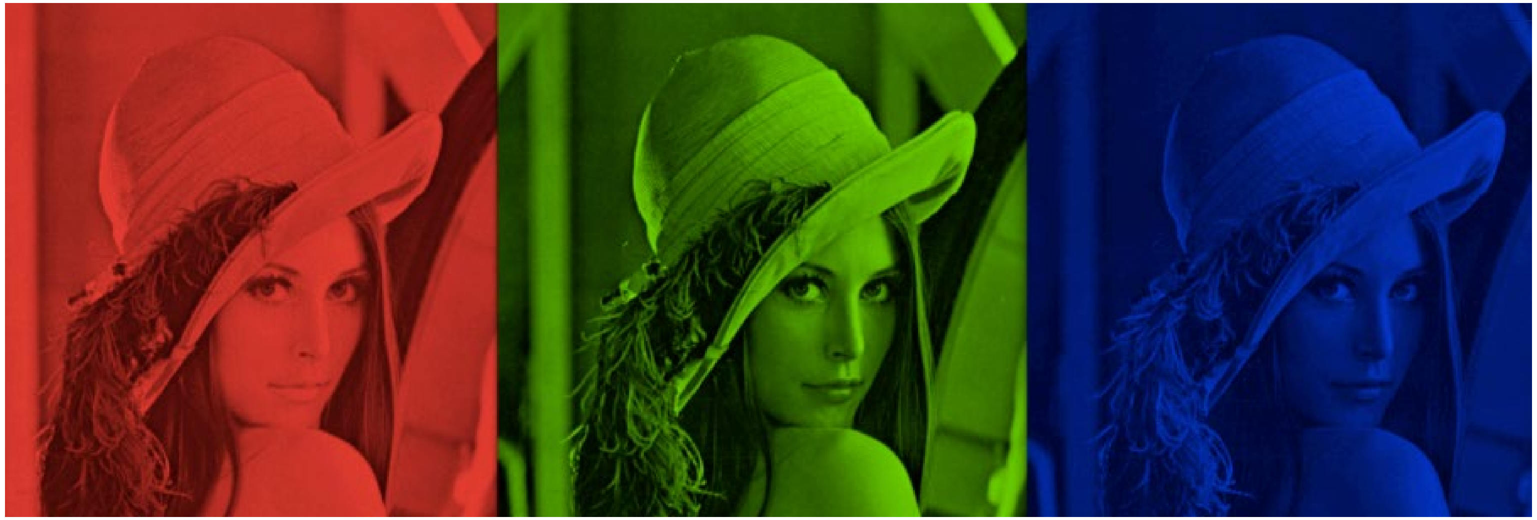


步长2
filter 3*3
filter个数6
零填充 1

多输入多输出通道

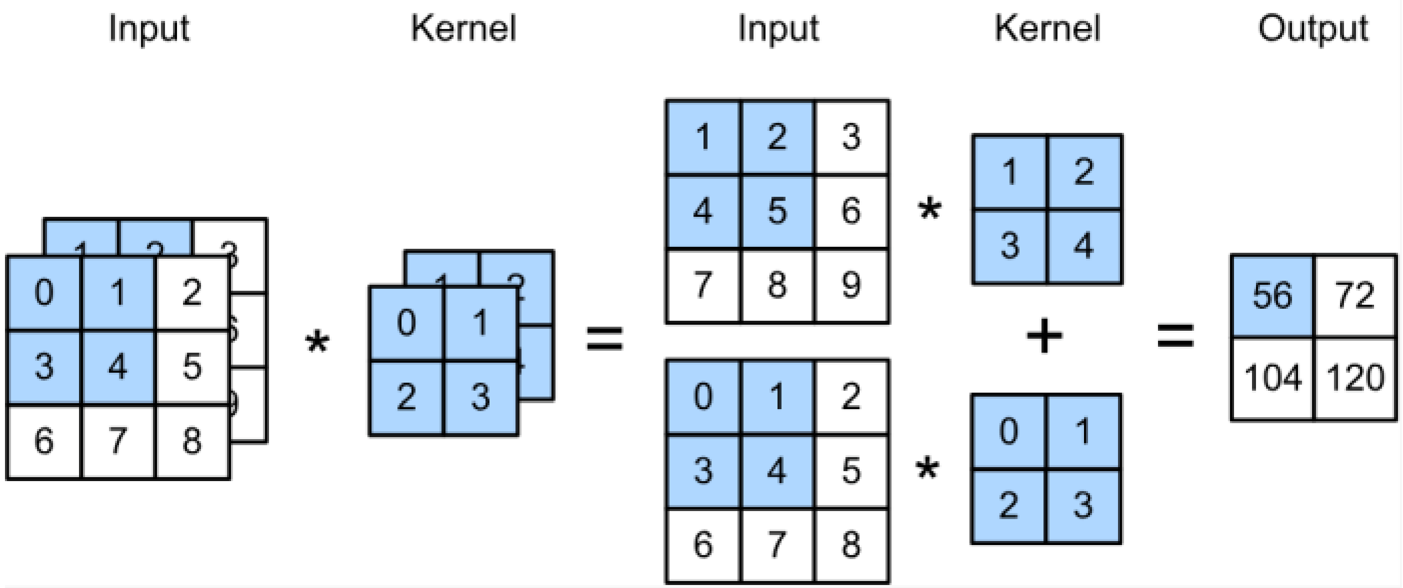
- 彩色图像有RGB三个通道





多输入多输出通道

- 每个通道都有一个卷积核，结果是所有通道卷积结果的和



$$
\begin{aligned}
&(1\times 1+2\times 2+4\times 3+5\times 4)\\
&+(0\times 0+1\times 1+3\times 2+4\times 3)=56
\end{aligned}
$$

多输入多输出通道

- 多个输入通道：

输入 $\mathbf{X}: c_i \times n_h \times n_w$

核 $\mathbf{W}: c_i \times k_h \times k_w$

输出 $\mathbf{Y}: m_h \times m_w$

$$\mathbf{Y} = \sum_{i=0}^{c_i} \mathbf{X}_{i,:,:} \star \mathbf{W}_{i,:,:}$$

- 如果仅有一个卷积核，那么无论输入包含多少个通道，输出都有且只有一个通道。

多输入多输出通道

- 多个输出通道：

输入 $\mathbf{X}: c_i \times n_h \times n_w$

核 $\mathbf{W}: c_o \times c_i \times k_h \times k_w$

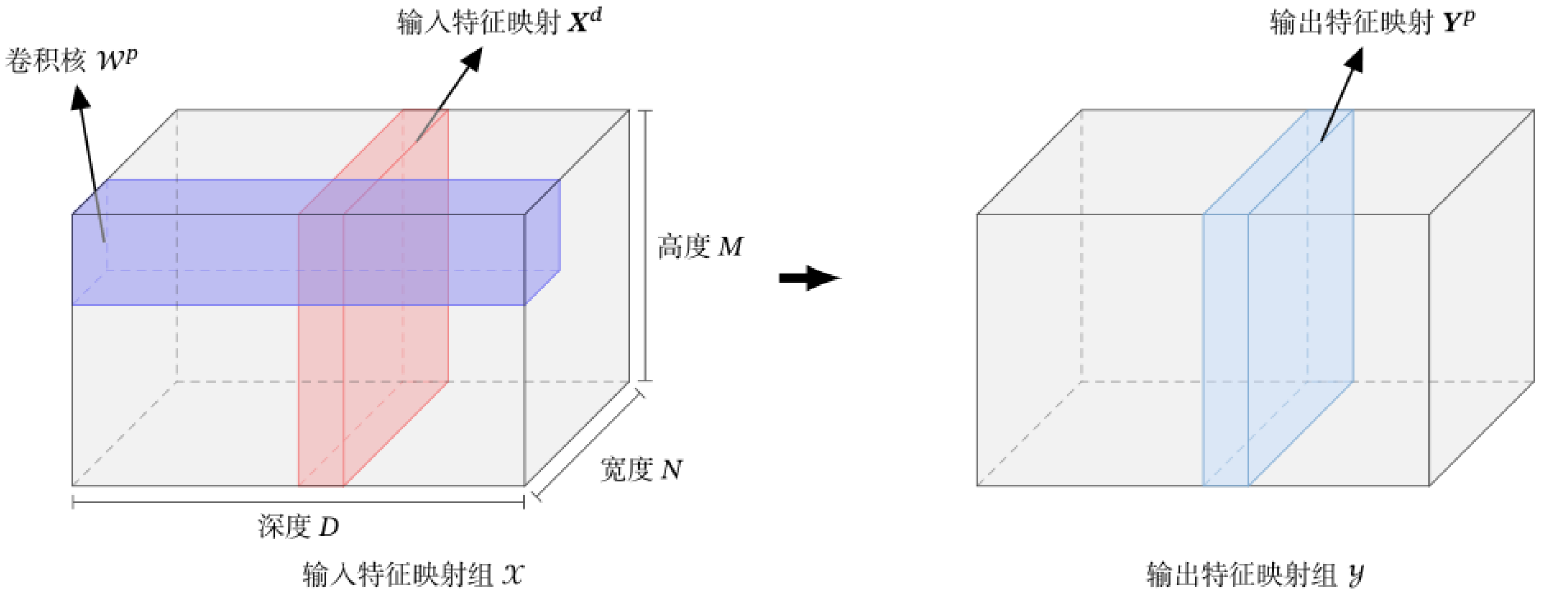
输出 $\mathbf{Y}: c_o \times m_h \times m_w$

$$\mathbf{Y}_{i,:,:} = \mathbf{X} \star \mathbf{W}_{i,:,:,:} \quad \text{for } i = 1, \dots, c_o$$

- 我们可以使用多个卷积核，每个核生成一个输出通道

多输入多输出通道

- 典型的卷积层为3维结构

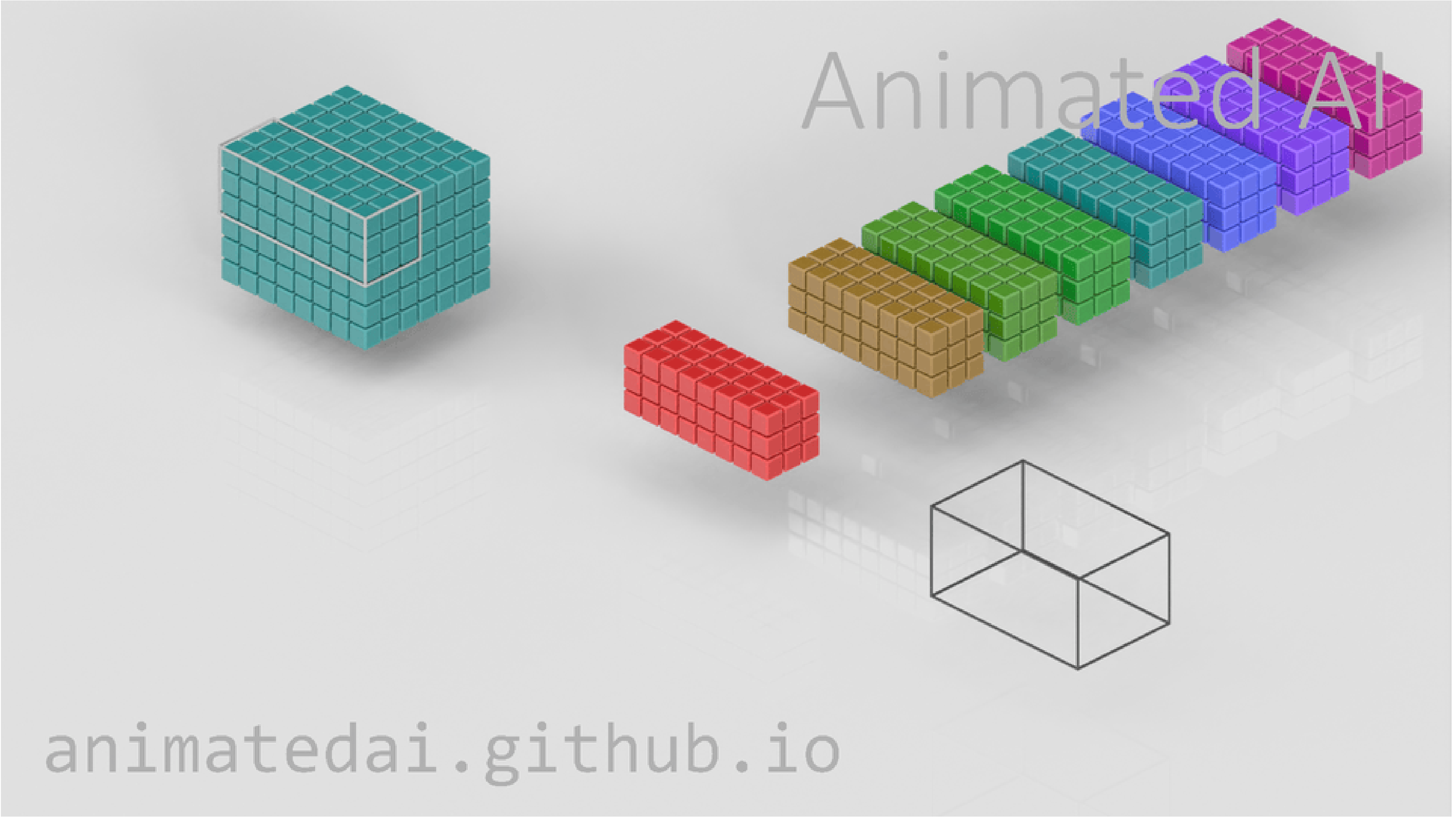


$$
\boldsymbol{Z}^p = \boldsymbol{W}^p \otimes \boldsymbol{X} + b^p = \sum_{d=1}^{D} \boldsymbol{W}^{p,d} \otimes \boldsymbol{X}^d + b^p,
$$

$$
\boldsymbol{Y}^p = f(\boldsymbol{Z}^p).
$$

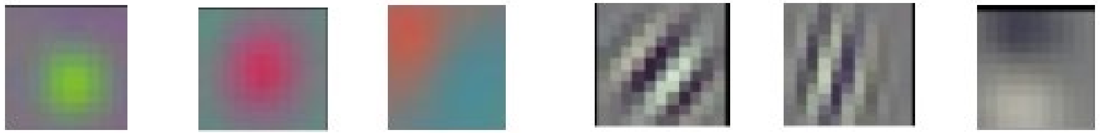
多输入多输出通道

- 典型的卷积层为3维结构



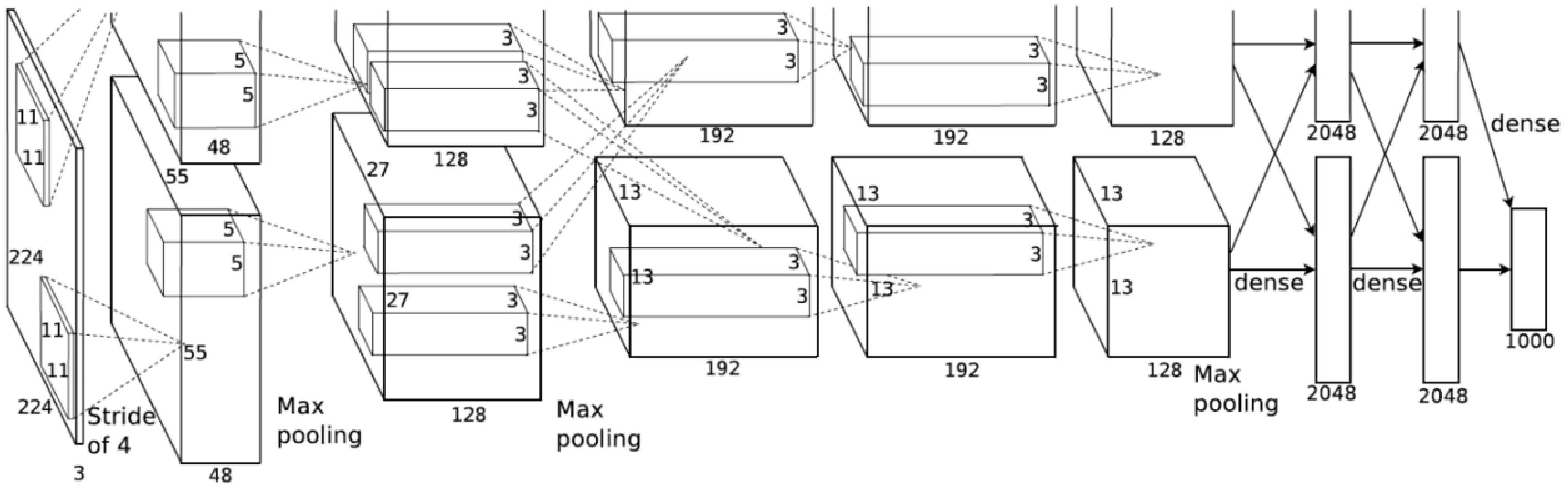
多输入多输出通道

- 每个输出通道可以识别特定的模式



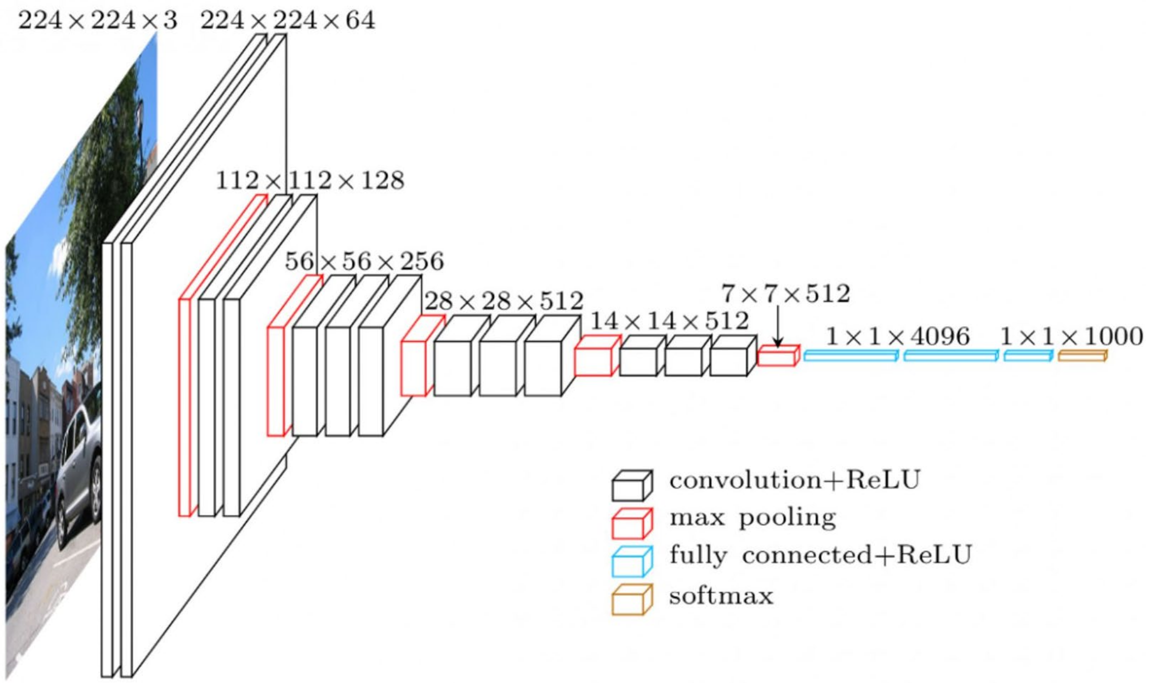
- 输入通道核识别并组合输入中的模式

多输入多输出通道


11
11
224
11
11
224
3
Stride
of 4
5
5
48
55
5
5
55
48
Max
pooling
3
3
3
128
27
3
3
27
3
3
128
Max
pooling
3
3
192
13
13
3
3
192
3
3
192
13
13
3
3
192
128
13
13
128
Max
pooling
dense
2048
2048
dense
dense
2048
2048
dense
1000

多输入多输出通道

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

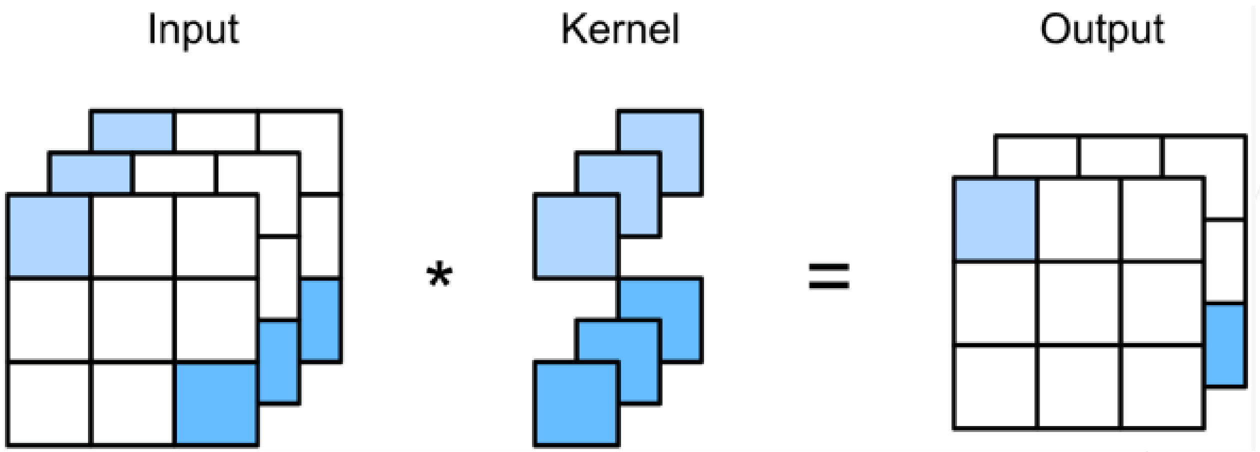


卷积参数量及浮点计算量

- 假设输入特征图包含C_i个通道，输出特征图包含C_o个通道，卷积核的长、宽均为K，输出特征图的长、宽分别为L_h、L_w，那么该卷积层的参数量是多少？浮点计算量是多少？

1×1卷积层

- 1×1卷积：卷积核长宽均为1——有什么作用?



- 1×1卷积不识别空间模式，只融合通道。

计算复杂度

- 卷积神经网络的计算复杂度：

输入 $\mathbf{X}: c_i \times n_h \times n_w$

核 $\mathbf{W}: c_o \times c_i \times k_h \times k_w$

偏差 $\mathbf{B}: c_o \times c_i$

输出 $\mathbf{Y}: c_o \times m_h \times m_w$

$$\mathbf{Y} = \mathbf{X} \star \mathbf{W} + \mathbf{B}$$

计算复杂度（浮点计算数 FLOP）$O(c_i c_o k_h k_w m_h m_w)$

$$c_i = c_o = 100$$
$$k_h = h_w = 5$$
$$m_h = m_w = 64$$

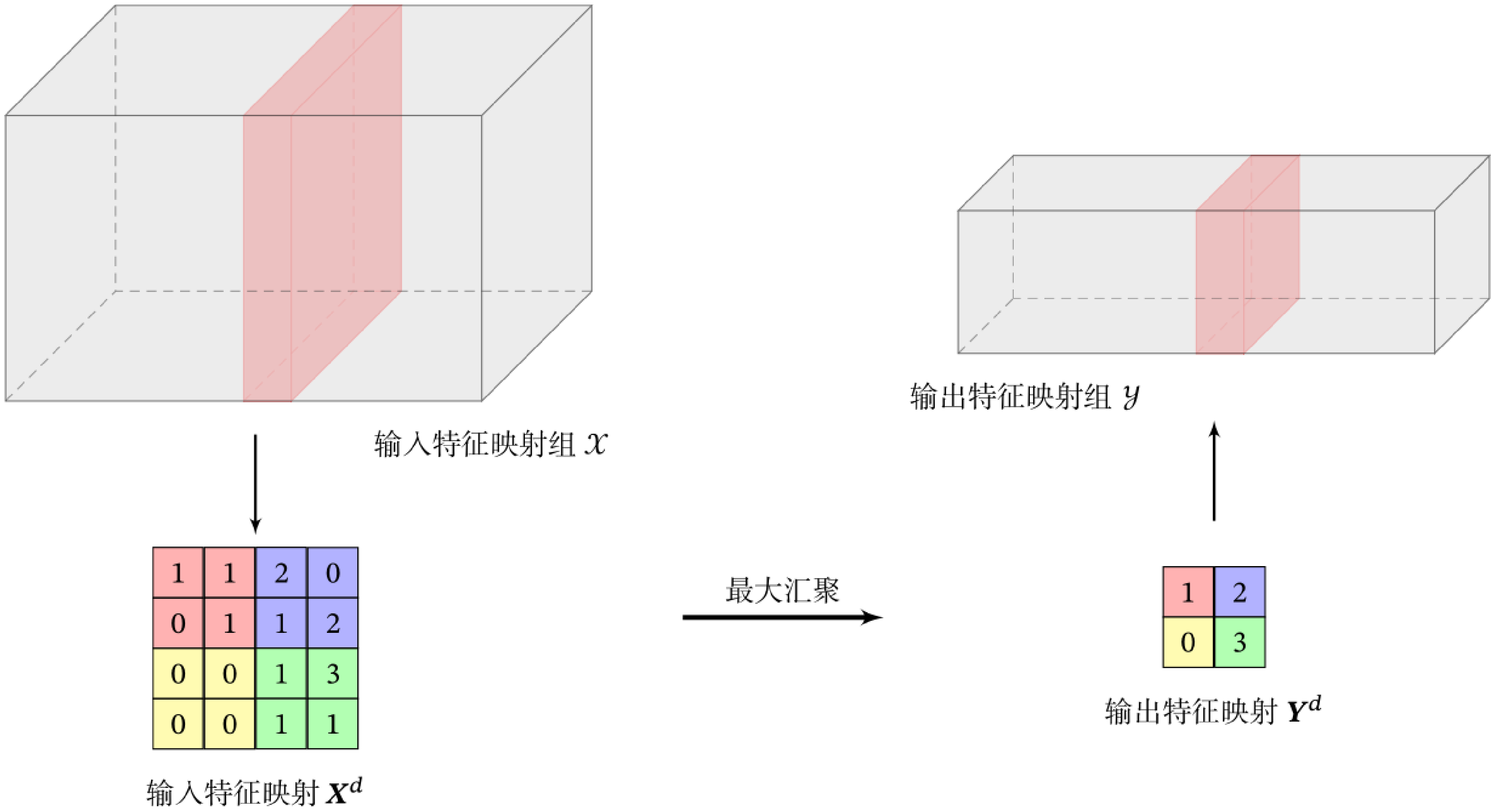
$\Rightarrow$ 1GFLOP

卷积总结

- 卷积层将输入与卷积核进行交叉相关运算，并加上偏置，得到输出；
- 卷积核、偏置都是可学习参数；
- 卷积核大小、步长、填充、输出通道数都是卷积层超参数；
- 卷积核个数决定输出通道数；
- 1×1卷积能够实现通道融合。

- 思考：上述超参数应该如何选取？有没有办法自动决定超参数的取值？

汇聚 / 池化（Pooling）



汇聚 / 池化（Pooling）

- 卷积对位置敏感
- 例：利用1×2卷积核[1, -1]检测垂直边缘

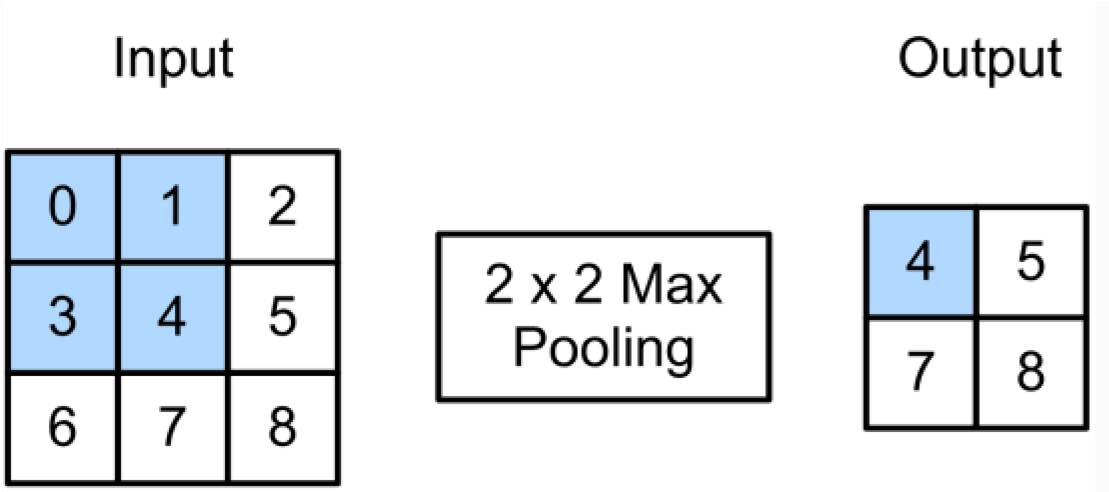
X
```
[[1. 1. 0. 0. 0.
 [1. 1. 0. 0. 0.
 [1. 1. 0. 0. 0.
 [1. 1. 0. 0. 0.
```

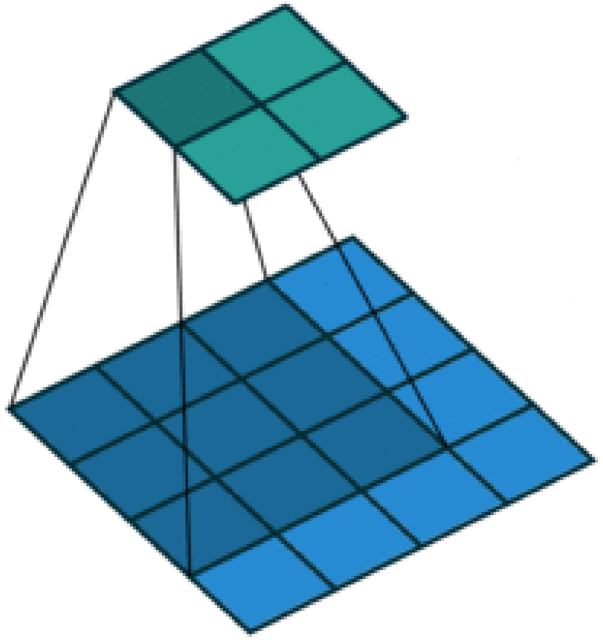
Y
```
[[ 0.  1.  0.  0.
 [ 0.  1.  0.  0.
 [ 0.  1.  0.  0.
 [ 0.  1.  0.  0.
```

  - 1像素的移位就会导致0输出
- 需要一定程度的平移不变性
  - 照明、物体位置、比例、外观等因图像而异

汇聚 / 池化（Pooling）

- 最大池化：返回滑动窗口中的最大值





$$\max(0,1,3,4) = 4$$

汇聚 / 池化（Pooling）

- 最大池化：返回滑动窗口中的最大值

垂直边缘检测

```
[[1. 1. 0. 0. 0.
 [1. 1. 0. 0. 0.
 [1. 1. 0. 0. 0.
 [1. 1. 0. 0. 0.
```

卷积输出

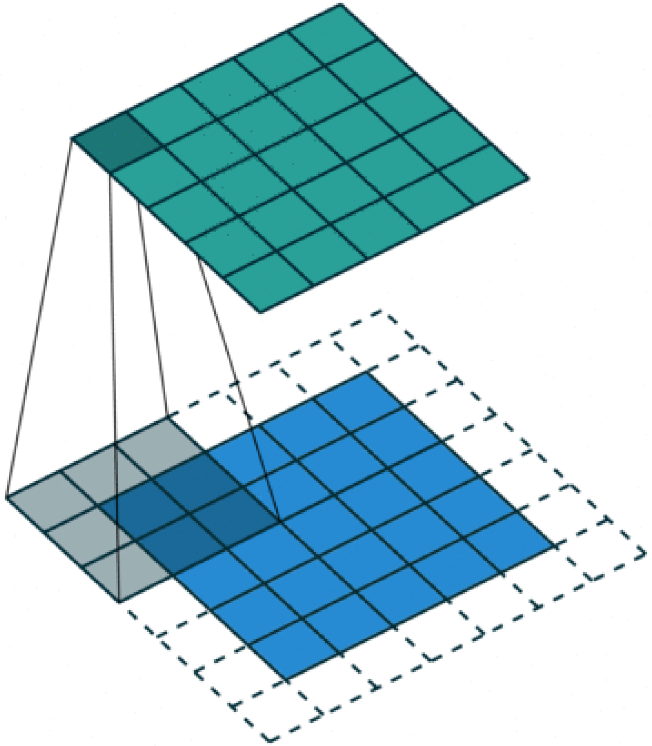
```
[[ 0.   1.   0.   0.
 [ 0.   1.   0.   0.
 [ 0.   1.   0.   0.
 [ 0.   1.   0.   0.
```

2 x 2 最大池化

```
[[  1.   1.   0.   0.
 [  1.   1.   0.   0.
 [  1.   1.   0.   0.
 [  1.   1.   0.   0.
```

汇聚 / 池化（Pooling）

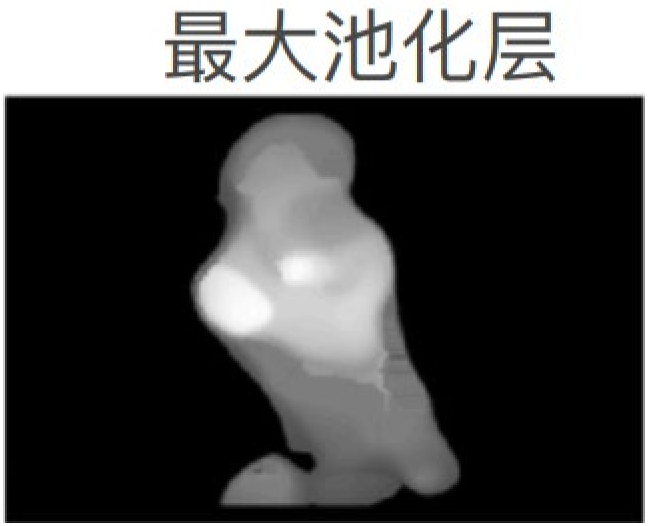
- 填充、步长与多个通道：
- 填充、步长的概念与卷积层类似；
- 没有可学习参数；
- 每个输入通道分别应用池化层，得到相应的输出通道，也就是说不改变通道数量，输出通道数=输入通道数。

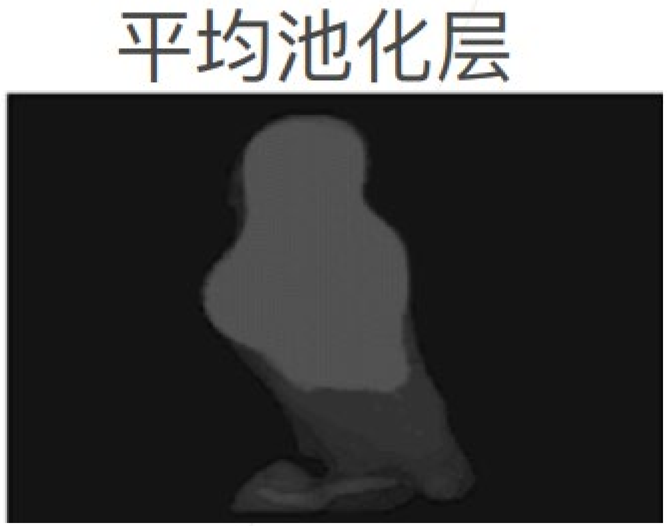


思考：比较卷积与汇聚（池化）运算的异同。

汇聚 / 池化（Pooling）

- 最大池化与平均池化



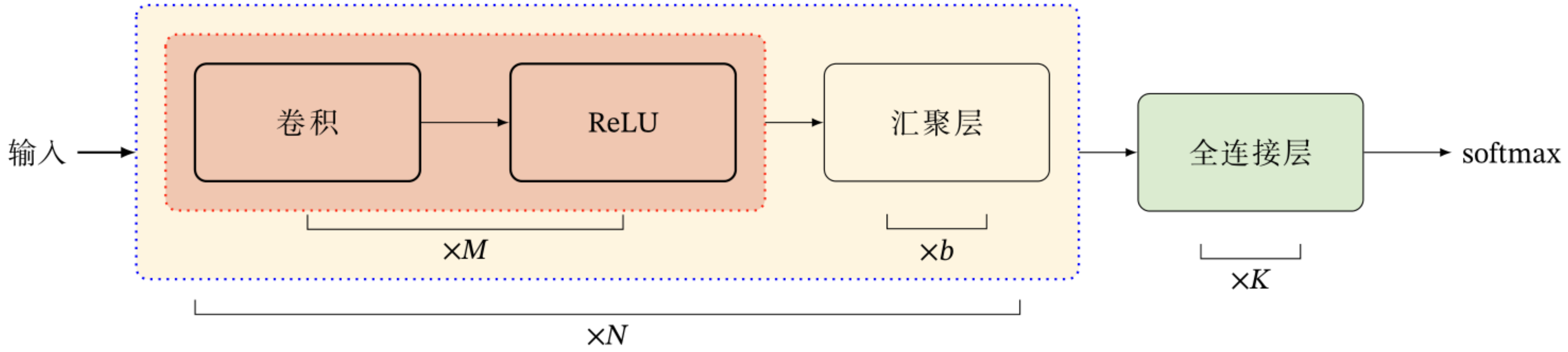


汇聚 / 池化总结

- 汇聚层（池化层）返回窗口中的最大值（或平均值）；
- 缓解卷积层对位置的敏感性；
- 同时，加快了特征映射尺寸缩减的速度；
- 有窗口大小、填充、步长等超参数；
- 没有可学习参数。

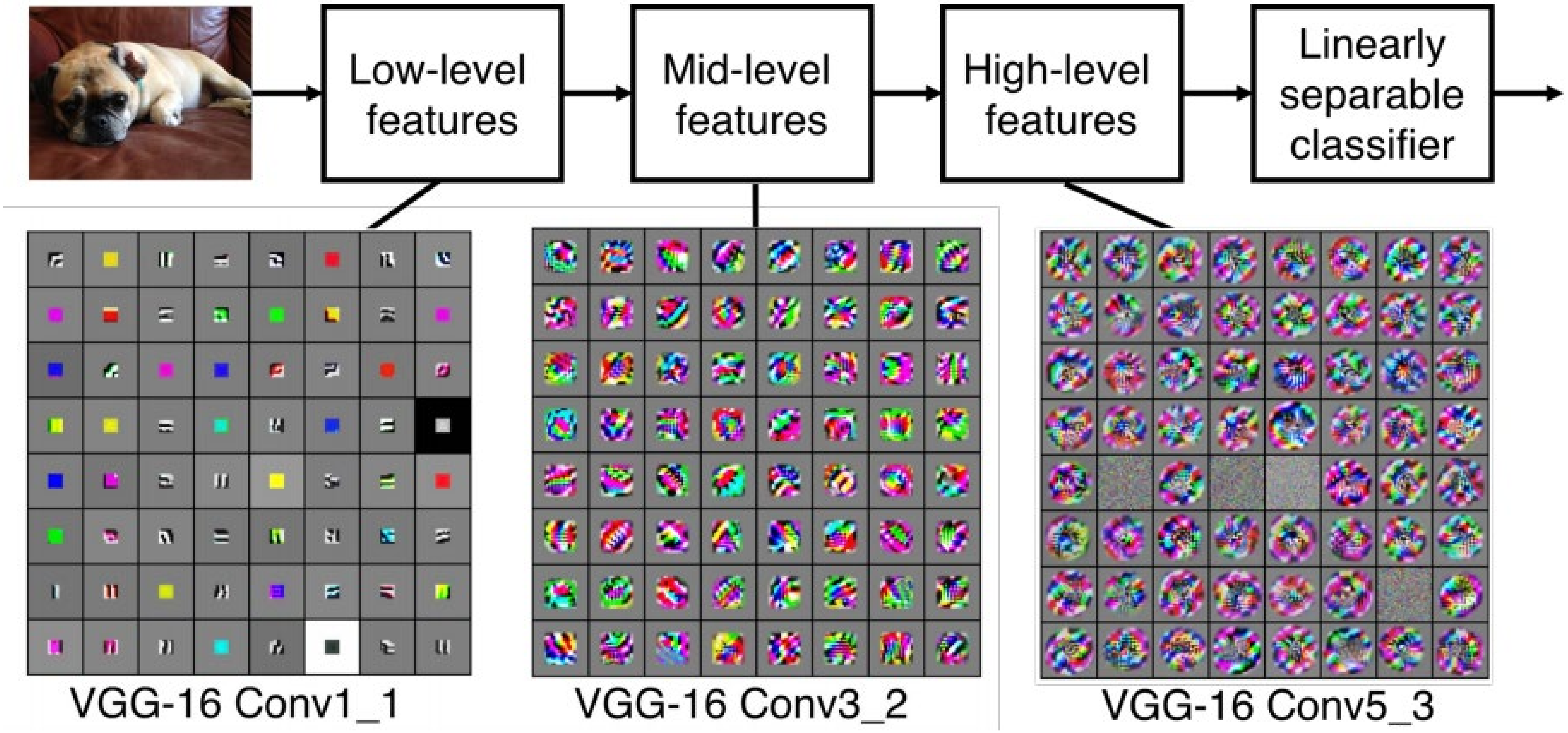
卷积网络结构

- 卷积网络是由卷积层、汇聚层、全连接层交叉堆叠而成。
  - 趋向于小卷积、大深度
  - 趋向于全卷积
- 典型结构



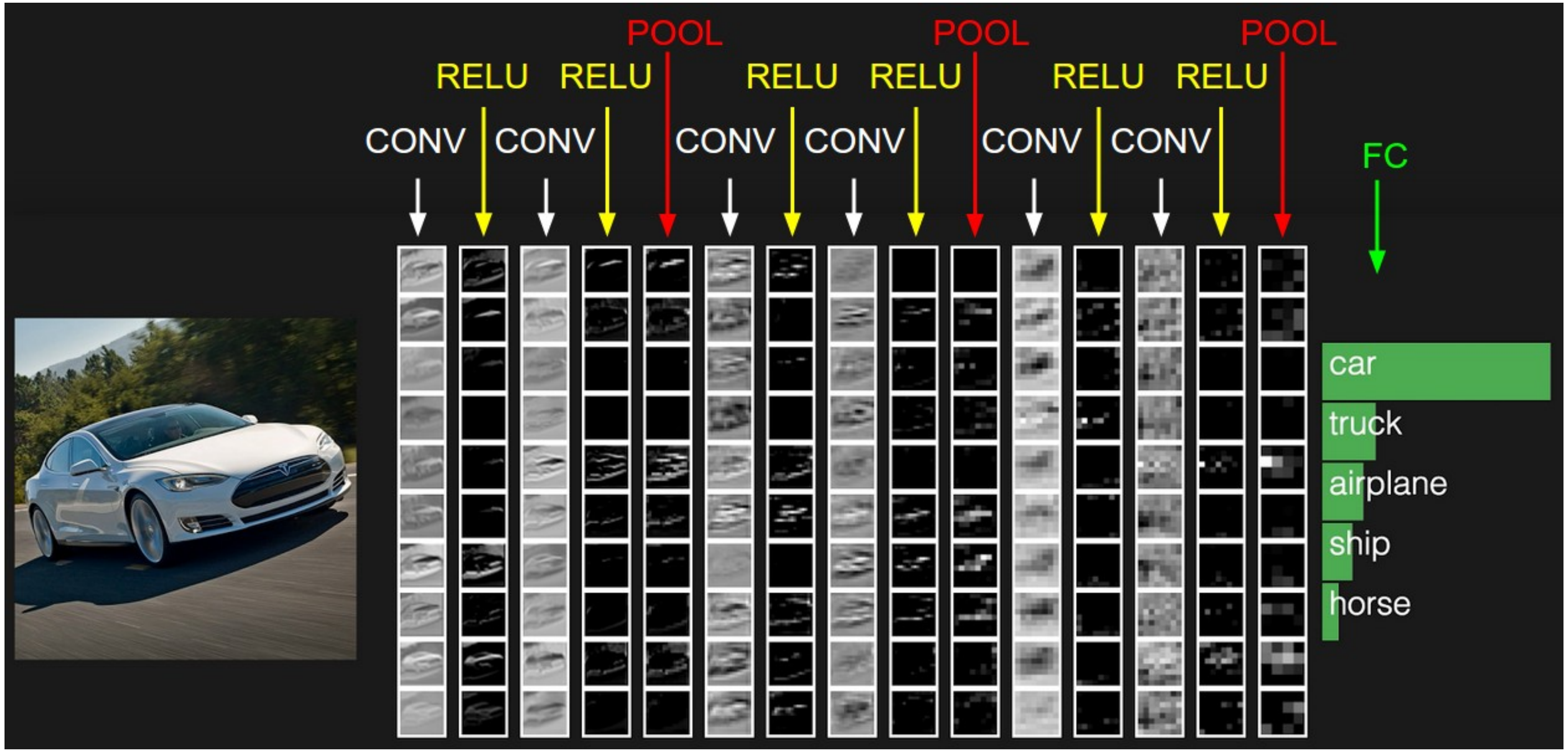
- 一个卷积块为连续M 个卷积层和b个汇聚层（M通常设置为2～5，b为0或1）。一个卷积网络中可以堆叠N 个连续的卷积块，然后在接着K 个全连接层（N 的取值区间比较大，比如1～100或者更大；K一般为0～2）。

表示学习



Low-level features
Mid-level features
High-level features
Linearly separable classifier
VGG-16 Conv1_1
VGG-16 Conv3_2
VGG-16 Conv5_3

表示学习


POOL
RELU
RELU
POOL
RELU
RELU
POOL
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

案例

[Convolution Visualizer (ezyang.github.io)]

[2DConv on RGB image (thomelane.github.io)]

[CNN Explainer (poloclub.github.io)]