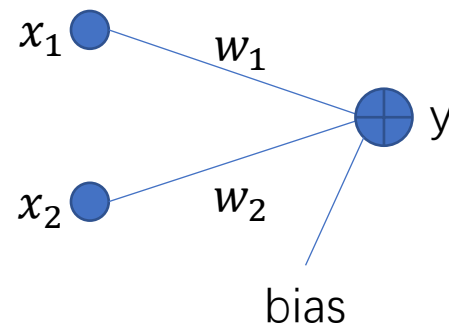


# 卷积神经网络简单计算

王知权

# Perceptron-感知机

- $y = x_1w_1 + x_2w_2 + bias$
- ●: 神经元

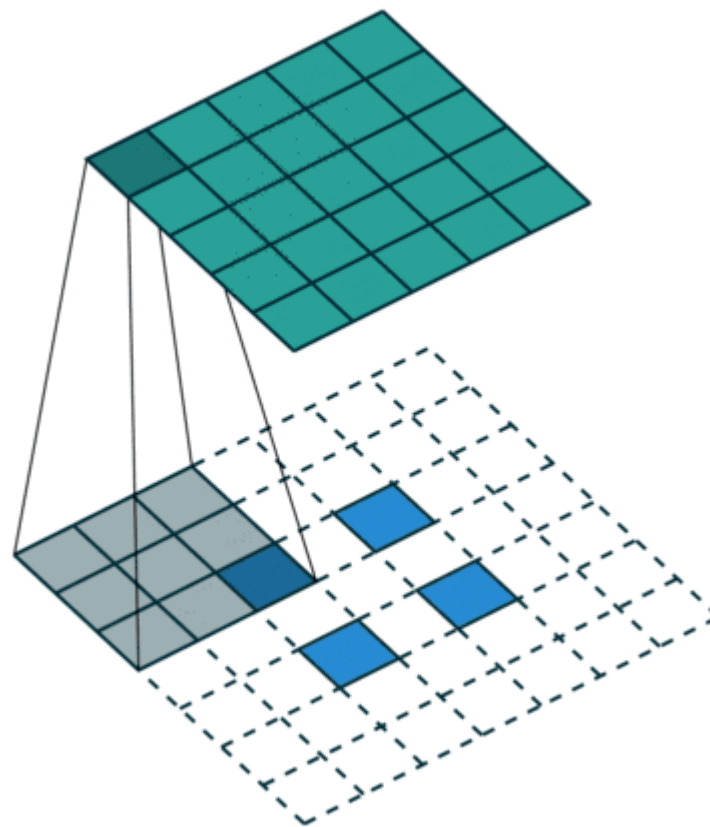


# Convolution-卷积

- Input layer & output layer
- 上一层的输出即为下一层的输入
- 参数: `Size(A, A), channel`
- 卷积操作的参数:
- 卷积核 (`number(output channels), size(f*f), depth`)) ,
- 步长 (`stride`) ,
- Padding等等

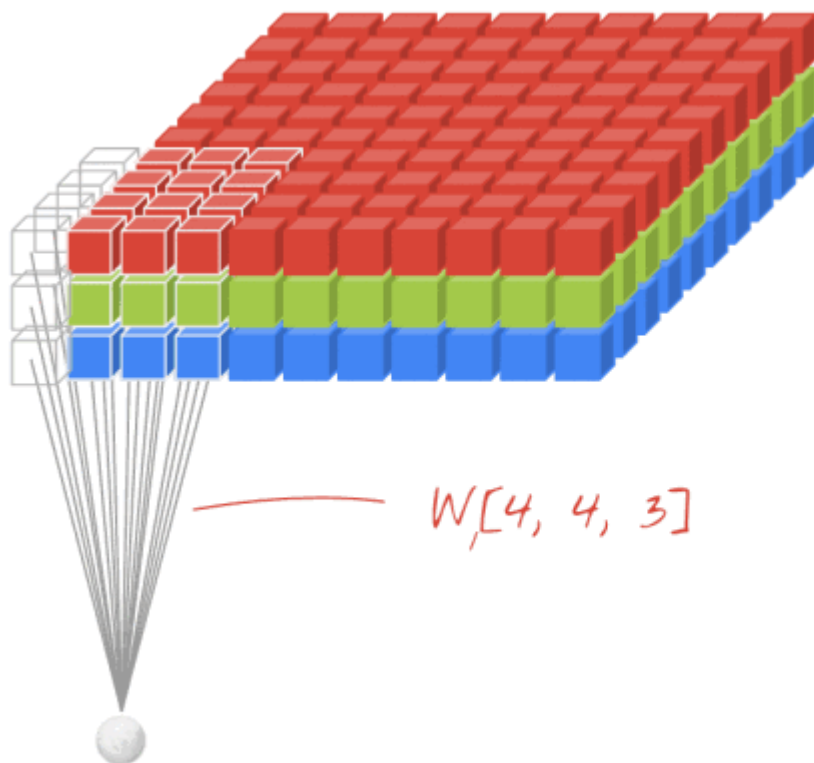
# Convolution-卷积

- 单通道(右侧举例)
- Input(size(7\*7),channel=1 )
- $F = 3$
- Depth = 1(由Input channel决定)
- Num = 1
- Stride = 1
- Padding = 0
- Output:
  - 边长= $\frac{a-f+2*p}{s} + 1 = \frac{7-3+0}{1} + 1 = 5$
  - 通道数=Num =1
  - 即 (size(5\*5),channel=1)
- Parameters:
  - 总参数= $F * F * D * N + \text{bias(忽略)} = 3 * 3 * 1 * 1 = 9$
  - 连接数= $F^2 * D * \text{边长(output)}^2 * N = 3 * 3 * 5 * 5 * 1 = 225$



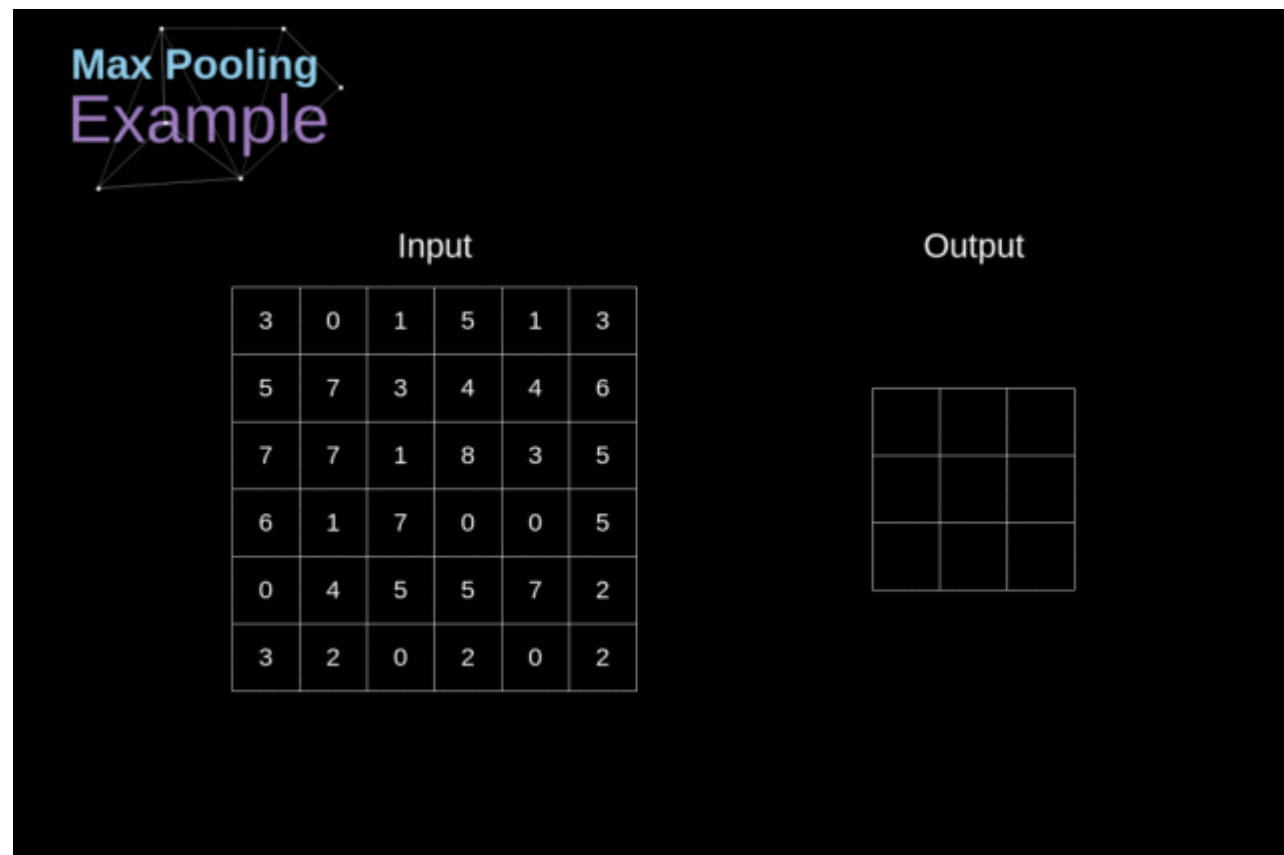
# Convolution-卷积

- 多通道 (以右侧RGB通道为例)
- Input(size(10\*10),channel=3 )
- $F = 4$
- Depth = 3(由Input channel决定)
- Num = 2
- Stride = 1
- Padding = (1+2)
- Output:
- 边长= $\frac{a-f+pl+pr}{s} + 1 = \frac{10-4+3}{1} + 1 = 10$
- 通道数=Num =1
- 即 (size(10\*10),channel=1)
- Parameters:
- 总参数= $F * F * D * N + \text{bias(忽略)} = 4 * 4 * 3 * 2 = 96$
- 连接数= $F^2 * D * \text{边长(output)}^2 * N = 4 * 4 * 3 * 10 * 10 * 2 = 9600$



# Subsampling- 下采样

- 均值池化(AvgPooling), 最大值池化(MaxPooling)等等
- 以右侧Max Pooling为例
- Input((size(6\*6),channel=1 )
- Size = (2\*2)
- Depth = 1(由Input channel决定)
- Stride = 2
- Padding = 0
- Output:
- 边长= $\frac{6-2+0}{2} + 1 = 3$
- 通道数=输入通道数=1
- 即 (size(3\*3),channel=1)

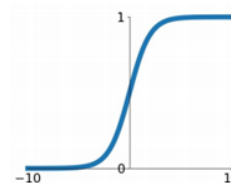


# Activation functions-激活函数

激活函数给神经网络引入了非线性因素，使得神经网络可以任意逼近任何非线性函数，这样神经网络就可以应用到众多的非线性模型中。

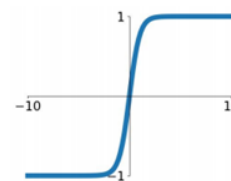
## Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



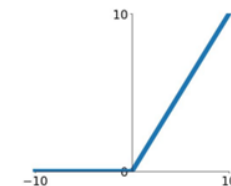
## tanh

$$\tanh(x)$$



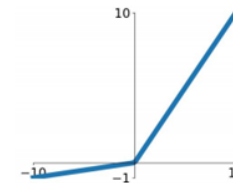
## ReLU

$$\max(0, x)$$



## Leaky ReLU

$$\max(0.1x, x)$$

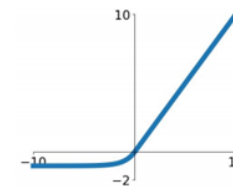


## Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

## ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



# Full connection

- Input(size(7\*7),channel=1 )
- Output(size(5\*5),channel=1)

Parameters:

总数=7\*7\*5\*5=1225

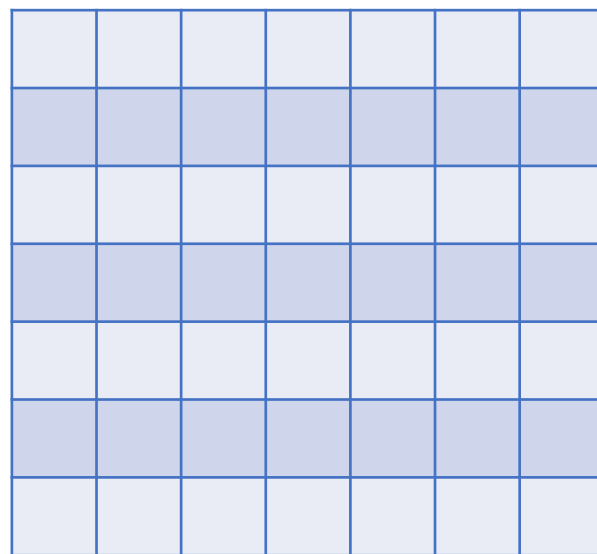
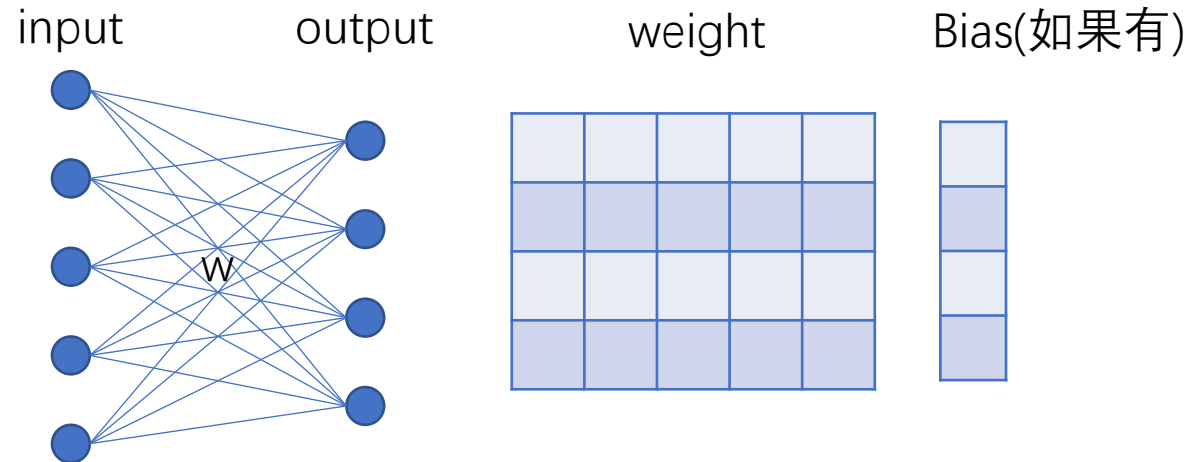
连接数=1225

对比卷积

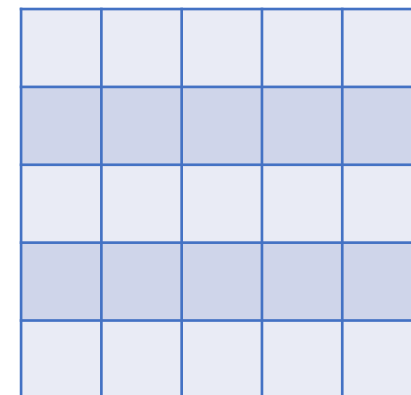
Parameters:

总参数=3\*3\*1\*1=9

连接数=3\*3\*5\*5\*1=225



input



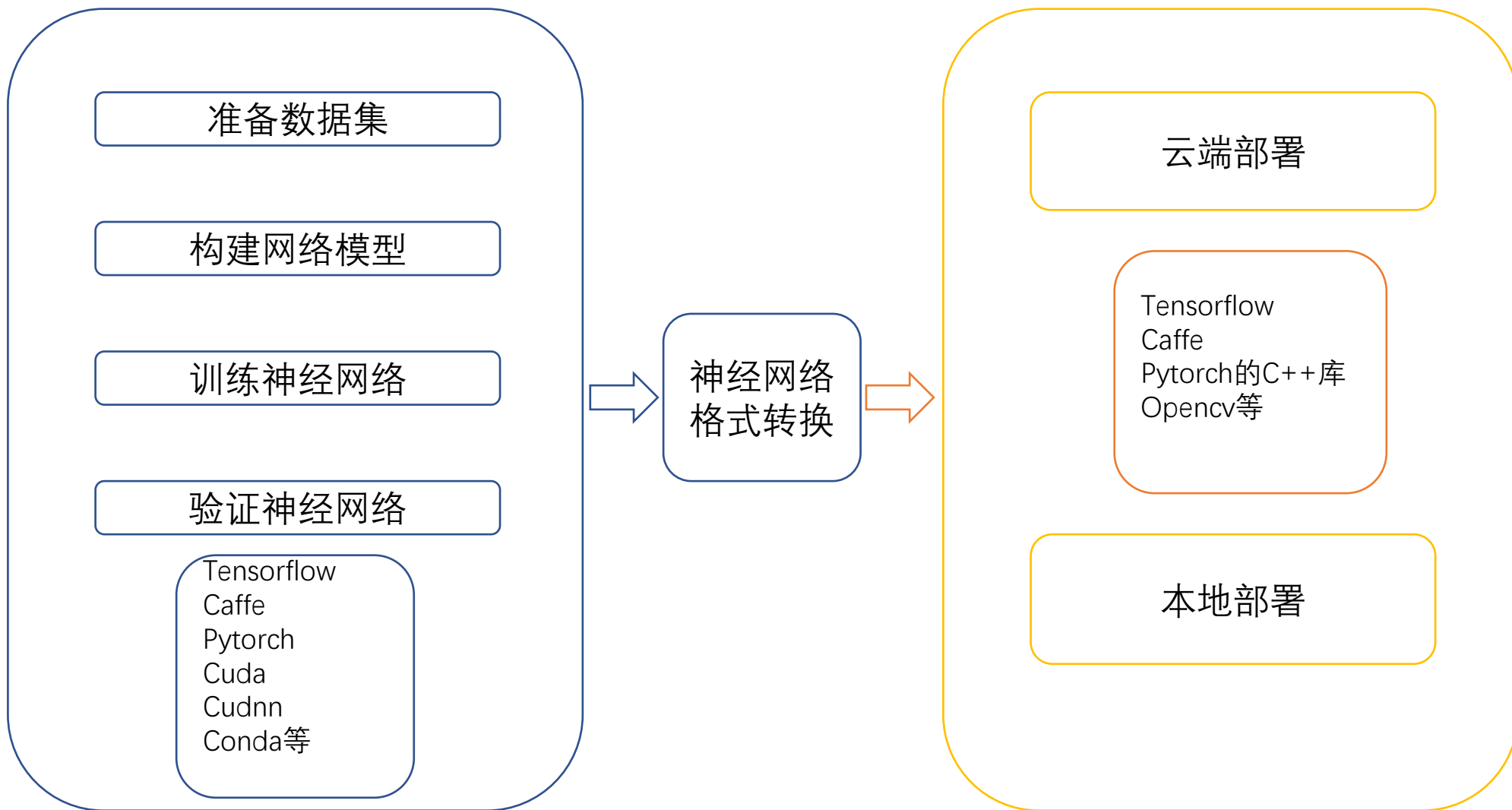
output



# Back propagation-BP网络

- 监督式的学习方法
- 核心就是损失函数(或者代价函数)
- 损失函数 (Loss Function)：是定义在单个样本上的，是指一个样本的误差。
- 0-1损失函数、平方损失函数、绝对值损失函数、对数损失函数
- 代价函数 (Cost Function)：是定义在整个训练集上的，是所有样本误差的平均，也就是所有损失函数值的平均。
- 均方误差、均方根误差、平均绝对误差、交叉熵代价函数
- 通过梯度下降法减小损失函数(代价函数) 值

# 神经网络训练与部署



# 经典的神经网络

- LeNet

