CHAPTER 6

卷积神经网络与深度学习

主讲人：吴春彪

2025年7月5日

01

神经元与神经网络

神经元与神经网络

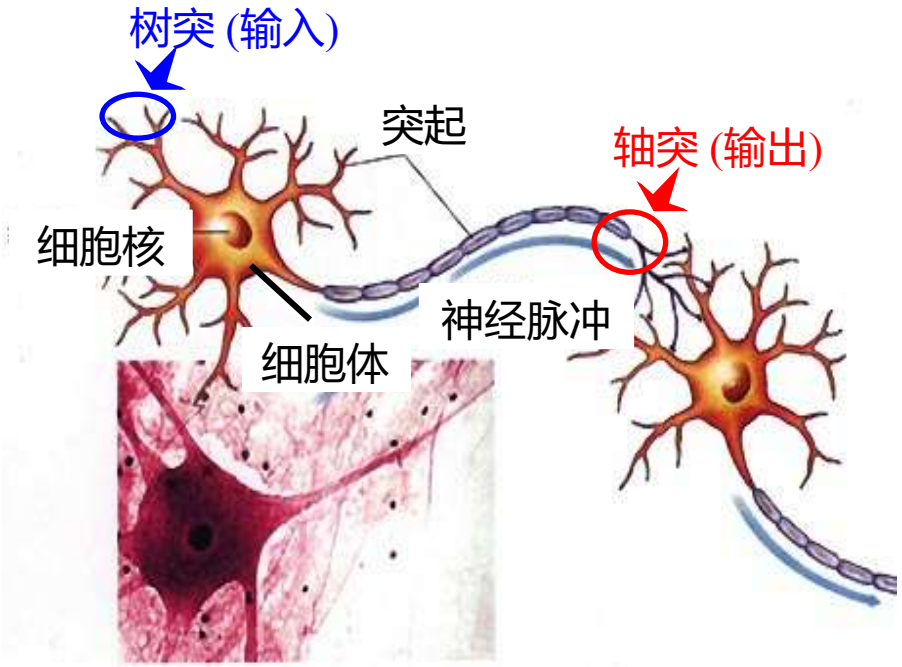
■ **神经元(Neuron)，是携带和传输信息的细胞，是人脑神经系统中最基本的单元。**

新陈代谢中心，接受并处理外部输入信息

- **神经元主要由细胞体 (Soma) 和突起(Neurite) 组成**

① 树突(Dendrite)：接受外部信号传入细胞体。

② 轴突(Axon): 将胞体产生信号传送另一个神经元。



- **工作原理：外部信号从树突出发，经过细胞体处理，然后由轴突传导，输送至下一个神经元的树突。**

神经元与神经网络

- **人工神经元 (Artificial neuron)**，1943年，美国神经生物学家**麦克洛奇**(W.S. McCulloch)和数学家**皮兹**(W. Pitts)提出最早的神经元模型 --- **M-P模型**，开创了神经科学理论研究的时代。

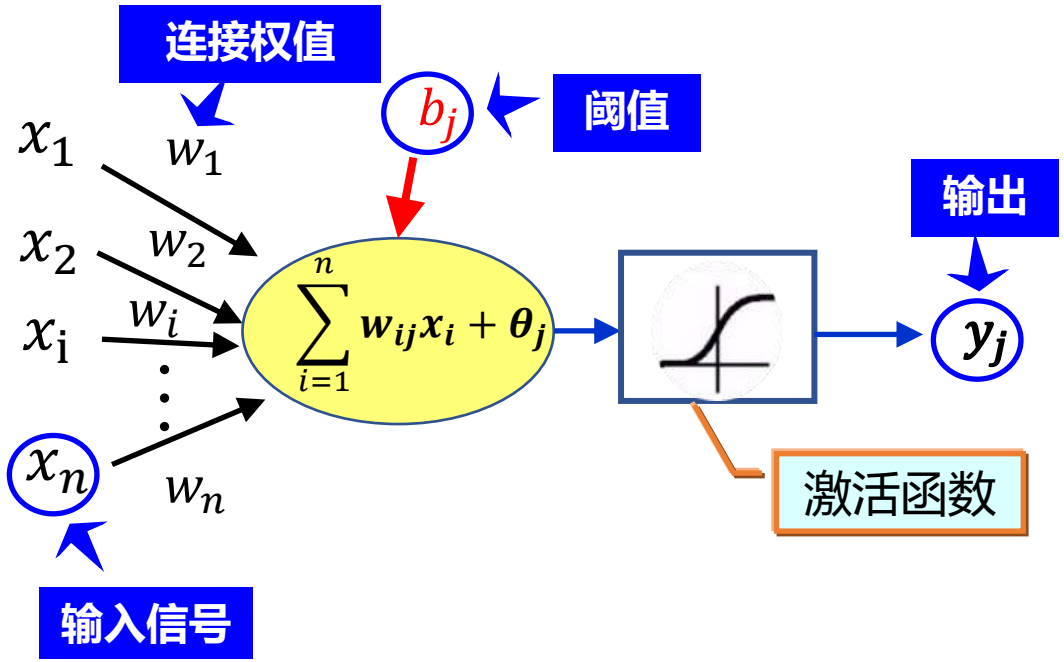
- **人工神经元数学计算过程：**

(1) 加权求和

$$net_j = \sum_{i=1}^{n} w_{ij} x_i + b_j$$

(2) 非线性函数映射

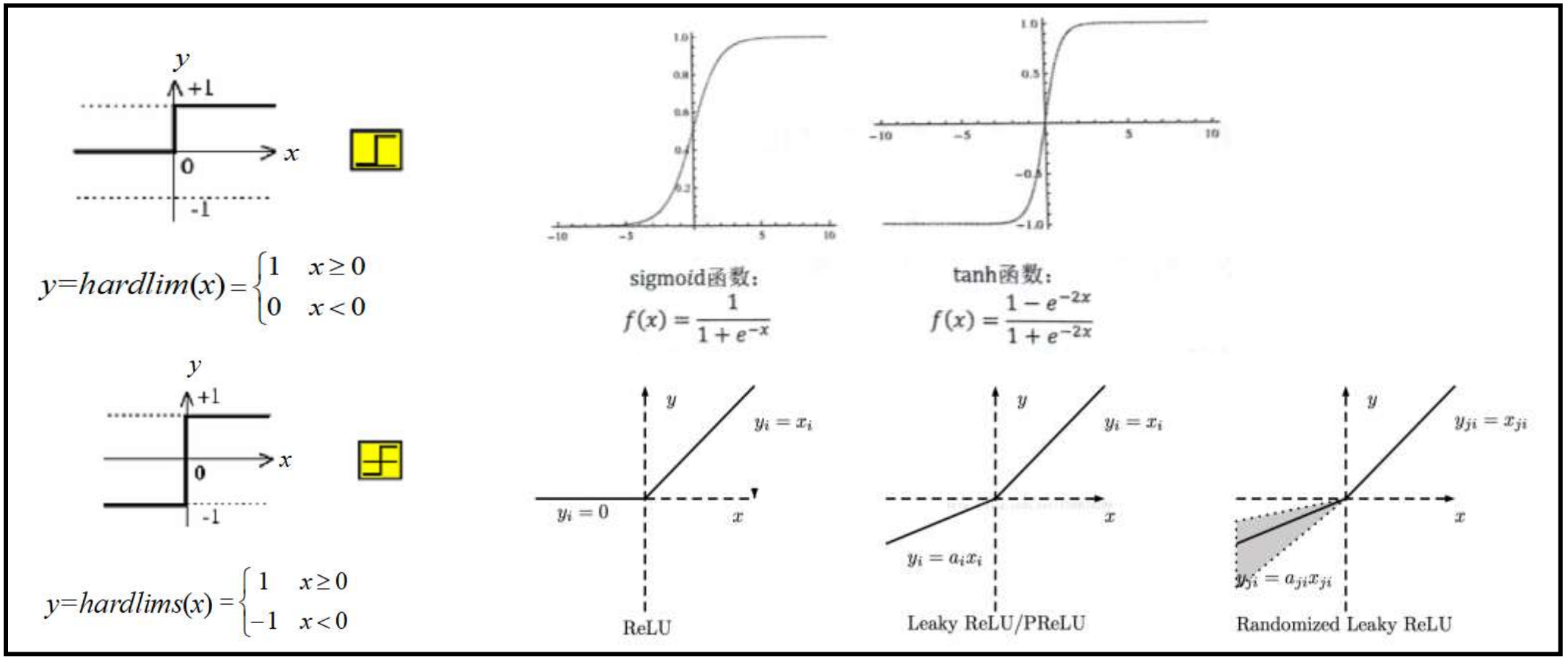
$$y_j = f(net_j)$$



神经元与神经网络

- **激活函数 (activation function)，加入非线性**因素，**增强模型表达能力**，使之可处理复杂的问题。

常用的激活函数主要有：阶跃函数、sigmoid函数、 tanh函数、 ReLU函数、 Leaky ReLU函数。



神经元与神经网络

■ 人工神经网络 (Artificial Neuron Network , ANN)

- 人工神经网络包含一系列基本的神经元、通过权重相互连接。神经元是人工神经网络最基本的单元。单元以层的方式组合，每一层的每个神经元和前一层、后一层的神经元连接，共分为输入层、输出层和隐藏层，三层连接形成一个神经网络。

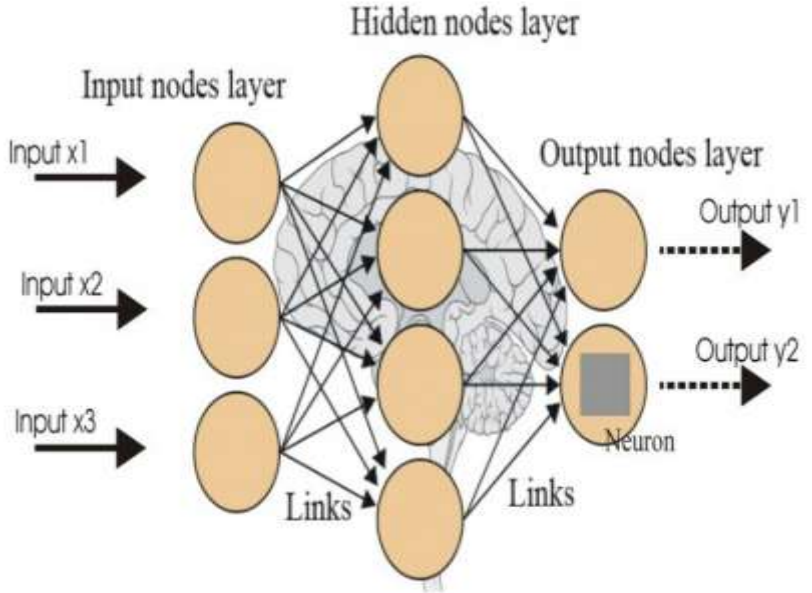
① 样本集：S=$\{(X_1, T_1),(X_2, T_2),\ldots,(X_s, T_s)\}$

② 输入信号的前向传播：

- 逐一地根据样本集中的样本(X_k, T_k)计算出实际输出O_k及其误差E_k，

$$y_j = f(net_j) = f(\sum_{i=1}^{n} w_{ij}x_i + b_j)$$

$$E_{\text{total}} = \frac{1}{2}\sum_{k=1}^{m}(t_k - o_k)^2$$ 均方误差 (MSE)



神经元与神经网络

③ 输出误差的反向传播：

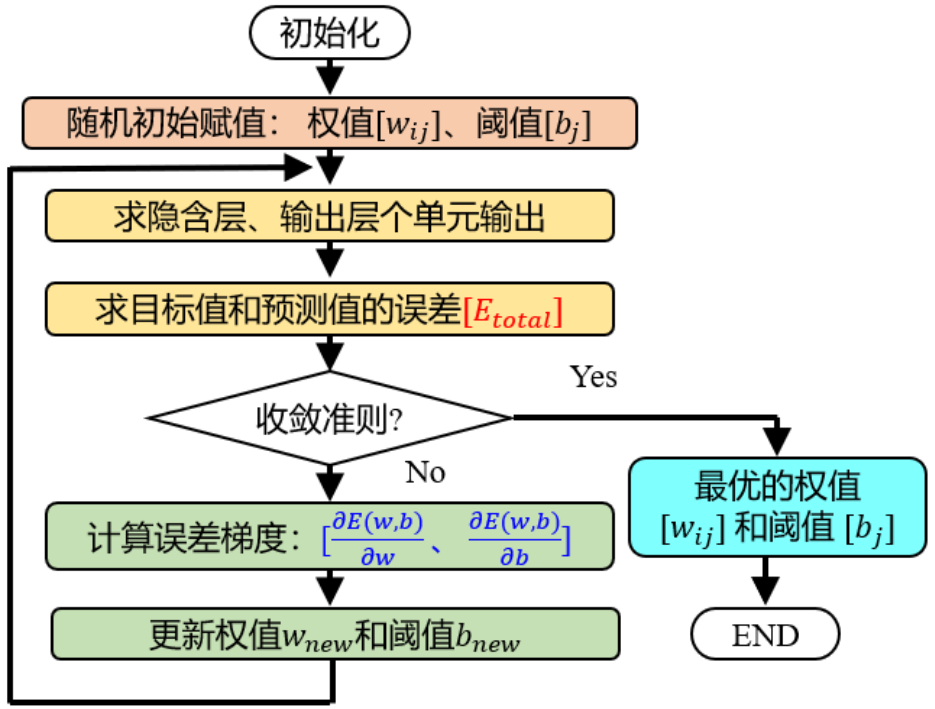
- 对各层神经元的权值$W^{(1)}$，$W^{(2)}$，…，$W^{(L)}$各做一次调整，一般采用 δ 学习规则来更新神经网络的权值，减少模型的预测误差。

$$\text{最小化误差} = \min_{w,b} E(w,b), \quad w,b \in R$$

- 求解目标函数$E(w,b)$最小值，采用梯度下降法 (沿着梯度的方向是函数变化做快方向)

$$\begin{cases} w_{ij}^{t+1} = w_{ij}^{t} - \eta * \dfrac{\partial E(w,b)}{\partial w_{ij}} \\ b_j^{t+1} = b_j^{t} - \eta * \dfrac{\partial E(w,b)}{\partial b_j} \end{cases}$$

η 是学习率$\in (0,1]$

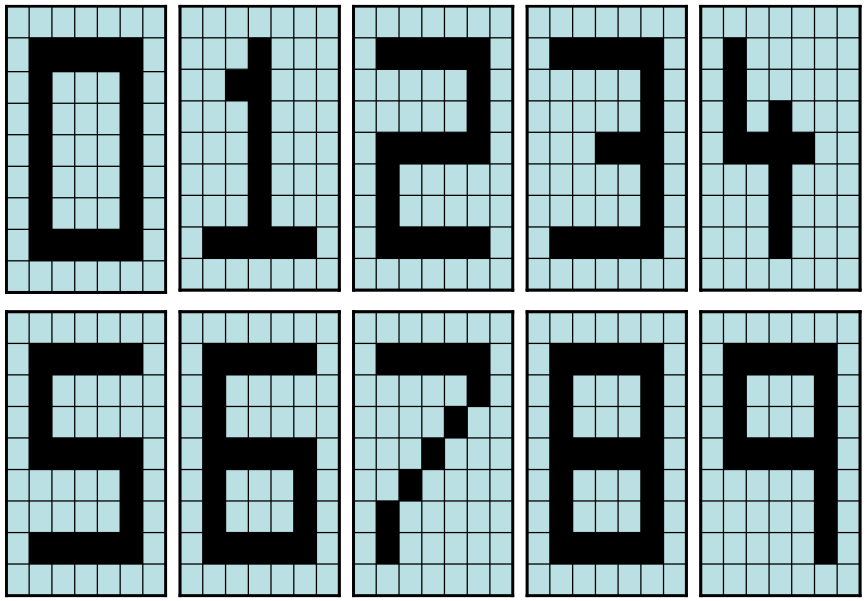


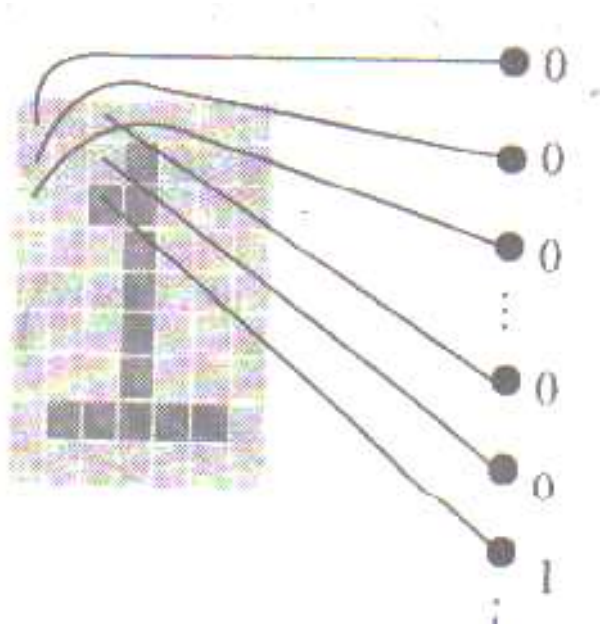
④ 重复这个循环，直到$\sum E_p < \varepsilon$（所有样本的误差之和）。

神经元与神经网络

■ BP神经网络应用案例 ---手写数字识别

案例：设计一个BP神经网络对数字0至9进行分类





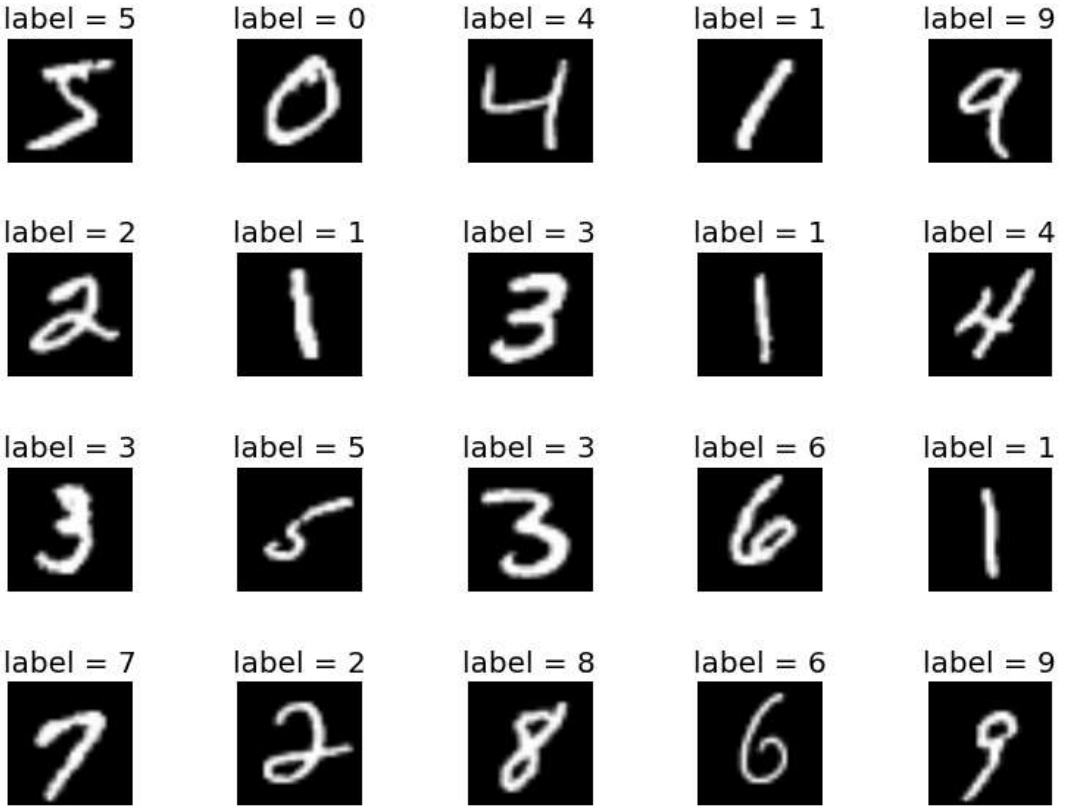
每个数字用n × m的网格表示，灰色像素代表0，黑色像素代表1。将每个网格表示为0，1的长位串。位映射由左上角开始向下直到网格的整个一列，然后重复其他列。

神经元与神经网络

■ BP神经网络应用案例 ---手写数字识别

- 数据集 mnist 手写数字 0-9

MNIST 数据集（手写数字数据集）是一个公开的公共数据集，数据集包含70000（60000+10000）个样本，其中有60000个训练样本和10000个测试样本，每个样本的像素大小为28*28，灰度图从 0（白色）到 255（黑色），每张图像对应一个 0 到 9 的数字标签。

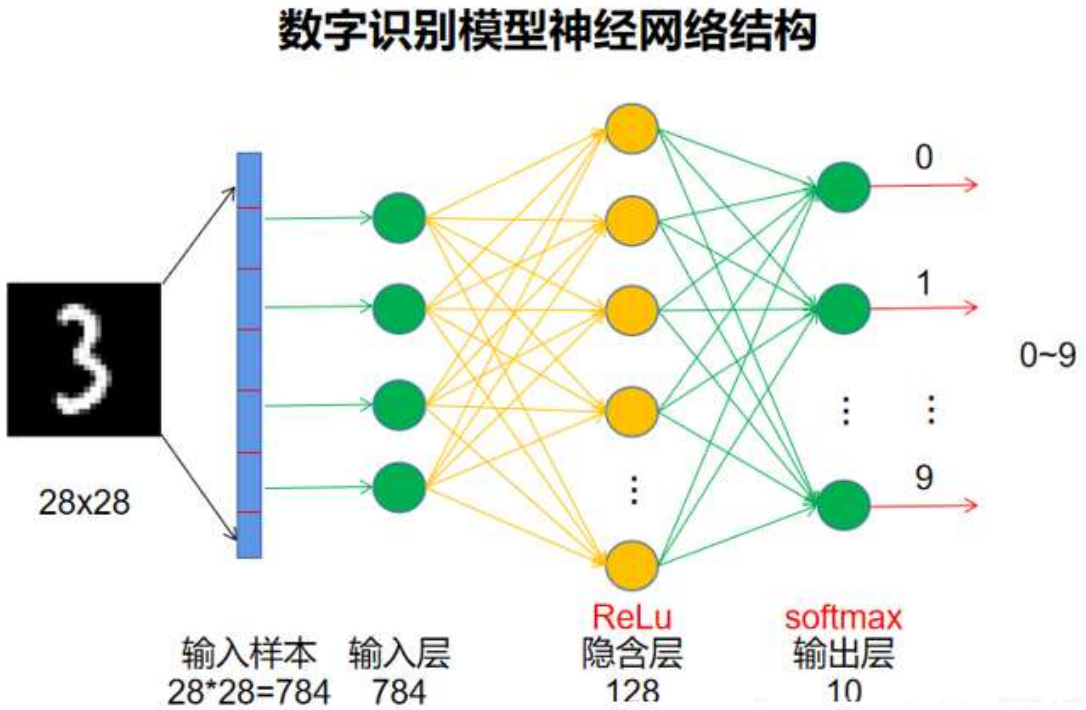


神经元与神经网络

■ BP神经网络应用案例 ---手写数字识别

- 神经网络模型训练

样本的像素大小为28*28，神经网络的输入变量(提取特征) 28 × 28=784 个，输出变量(0~9)10种分类；隐含层的节点数通过试错法多次尝试确定，当前网络隐藏层选择：隐含层尽量设置成2的n次幂个节点，故选择128个节点，使用ReLu激活函数。因此，BP神经网络的结构为：784-128-10。通过预测的分类结果概率，获得预测结果。



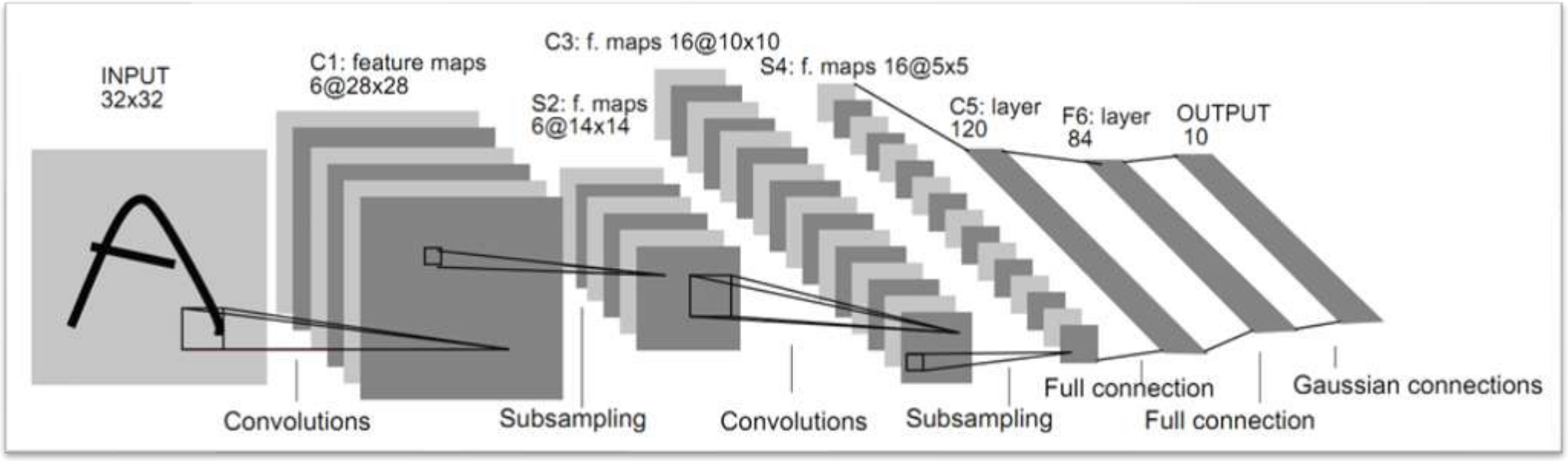
02

卷积神经网络CNN

卷积神经网络

(1) 卷积神经网络 (CNN)

1990年，“CNN之父”Yann LeCun (杨立昆)等人开创性地提出了LeNet系列网络，标志着卷积神经网络的诞生，也奠定了CNN网络的基本思想和结构。



AlexNet、GoogleNet 、VGG 、ResNet 等多种卷积神经网络被提出，CNN已被广泛应用于图像识别、计算机视觉、以及自然语言处理等领域。

卷积神经网络

（2）卷积神经网络基本架构

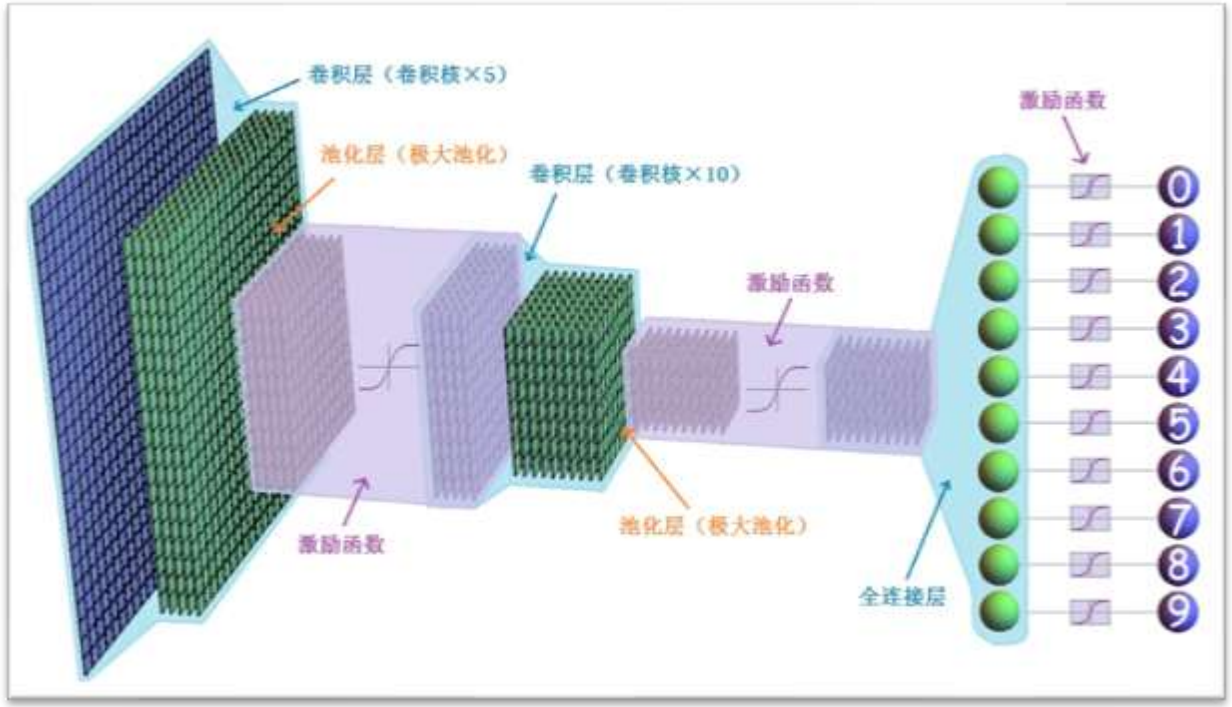
① 输入层（Input layer）

② 卷积层（Convolutional layer）

③ 池化层（Pooling layer）

④ 全连接层（Full connection layer）

⑤ 输出层（Output layer）



多层卷积层(含池化层)

全连接层
(多层感知器，BP网络)

卷积神经网络(CNN)一般由前段的多层卷积层和末段的全连接层构成。卷积层实现自动提取特征；全连接层完成模式分类。

卷积神经网络

① 输入层（Input Layer）

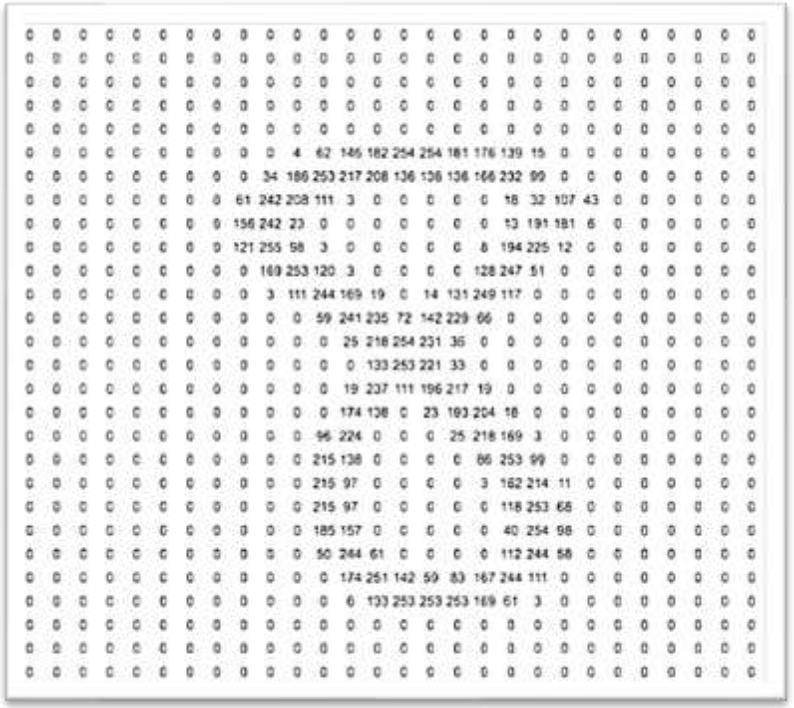
◆ 像素 (Pixel)：组成图像的最基本单位。一张图片由多个像素点以其明确的位置和分配的色彩数值构成。单位面积上像素点越多，图片清晰细腻。

◆ 灰度图像 — 像素矩阵

28 ×28
784 pixels

图片读取

取值范围[0，255]



28 ×28 矩阵

卷积神经网络

② 卷积层（Convolutional Layer）

◆ 卷积层：也被称为特征提取层。通过卷积层，CNN能够自动提取输入数据（图片）的不同特征。卷积层是CNN网络最重要的一个层级，层次越深，提取的特征越多。

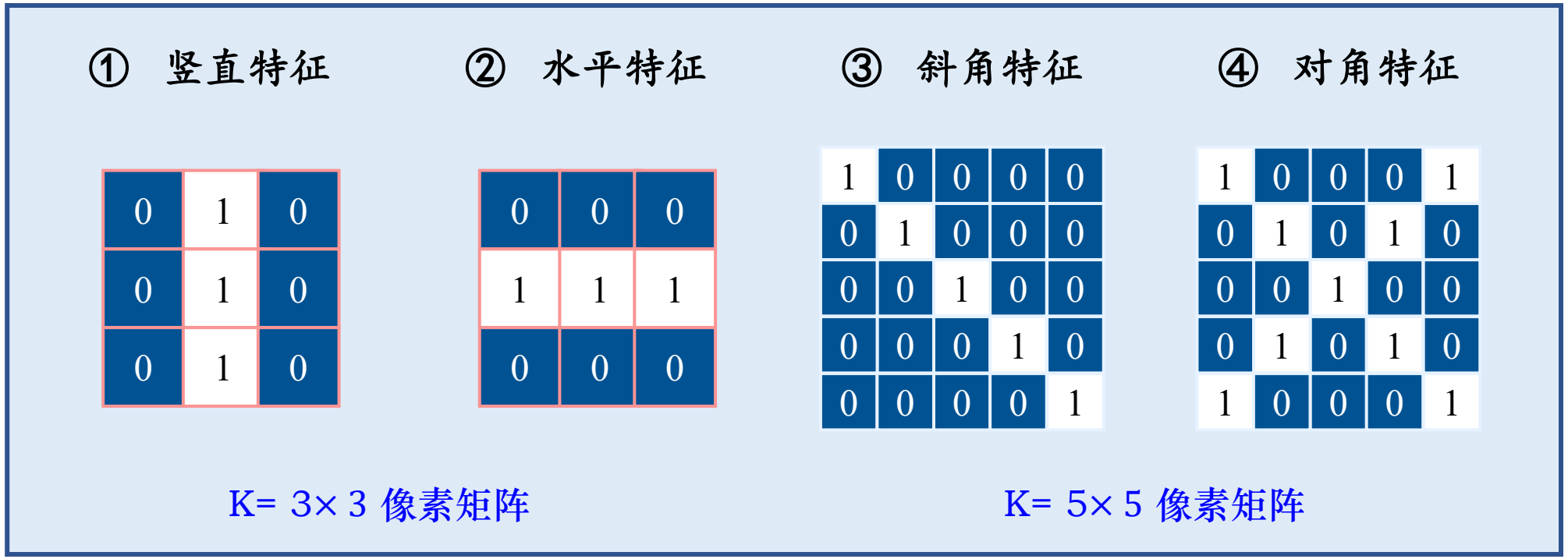
手写数字7

手写数字8

卷积神经网络

② 卷积层（Convolutional Layer）

◆ 卷积核 (Convolutional kernel)：也被称为特征过滤器，是一个 $N \times N$ (e.g., 3×3 or 5×5) 的像素矩阵，该矩阵也称为感受野。通过卷积核可提取图像的局部特征。



卷积神经网络

② 卷积层（Convolutional Layer）

◆ 卷积运算 → 提取特征

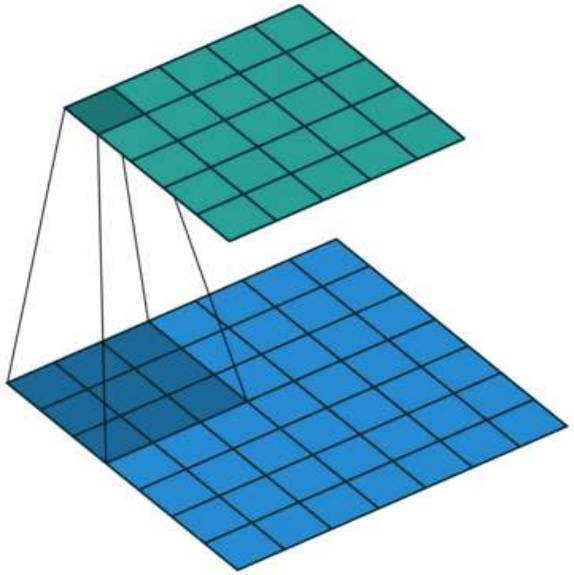
卷积运算：卷积核按顺序依次提取原始图片中卷积矩阵大小的区域，将其区域中每个像素单元依次和卷积核内相对应的像素值相乘，再求和。

$$S_1 = a_1 A_1 + b_1 B_1 + c_1 C_1 + a_2 A_2 + b_2 B_2 + c_2 C_2 + a_3 A_3 + b_3 B_3 + c_3 C_3$$

$$\begin{bmatrix} a_1 & b_1 & c_1 & d_1 & e_1 & f_1 \\ a_2 & b_2 & c_2 & d_2 & e_2 & f_2 \\ a_3 & b_3 & c_3 & d_3 & e_3 & f_3 \\ a_4 & b_4 & c_4 & d_4 & e_4 & f_4 \\ a_5 & b_5 & c_5 & d_5 & e_5 & f_5 \\ a_6 & b_6 & c_6 & d_6 & e_6 & f_6 \end{bmatrix} \times \begin{bmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{bmatrix} = \begin{bmatrix} S_1 & & & \\ & & & \\ & & S_{11} & \\ & & & \end{bmatrix}$$

$$S_{11} = c_1 A_1 + d_3 B_1 + e_3 C_1 + c_4 A_2 + d_4 B_2 + e_4 C_2 + c_5 A_3 + d_5 B_3 + e_5 C_3$$

■ 卷积过程：



卷积神经网络

② 卷积层（Convolutional Layer）

实例：手写数字识别

原始图片

6×6

0	1	1	1	1	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	1	0

6×6 像素矩阵

输入层（Input Layer）

0	1	0
0	1	0
0	1	0

K= 3×3，竖直特征

1	1	1	3
0	0	0	3
0	0	0	3
0	0	0	3

0	0	0
1	1	1
0	0	0

K= 3×3，水平特征

0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1

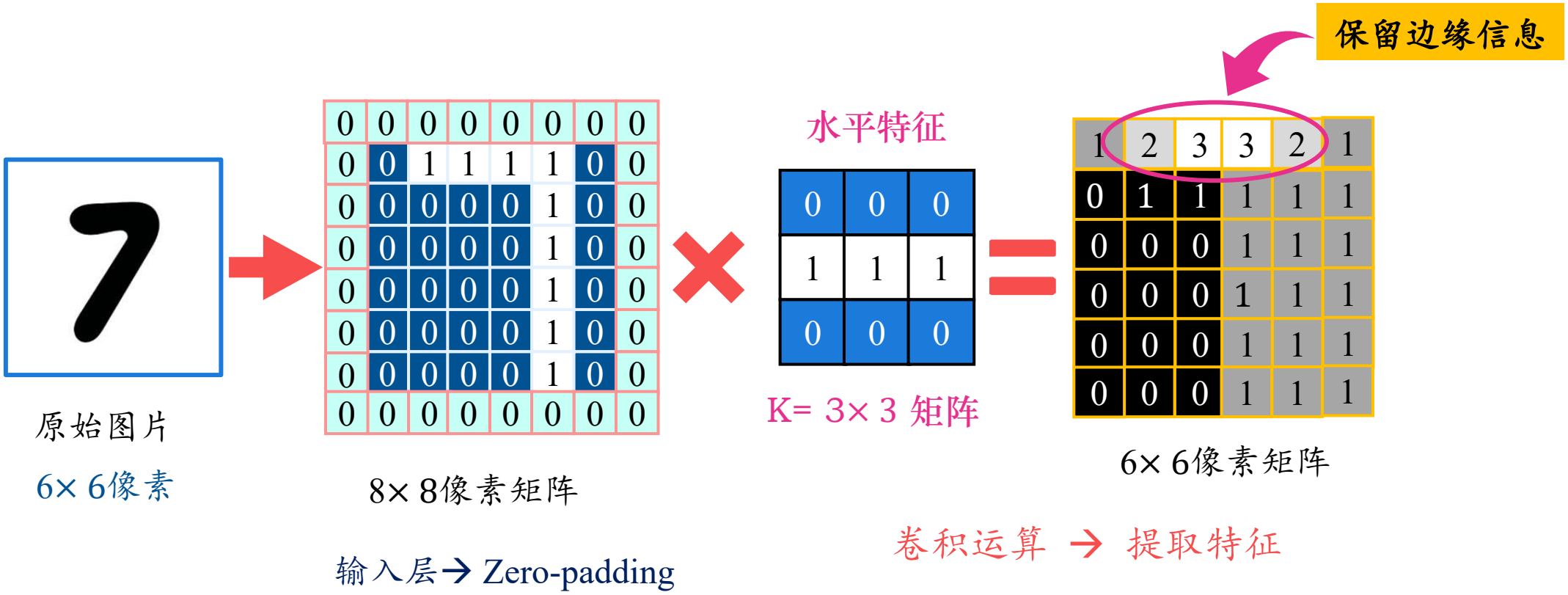
特征丢失

卷积运算 → 提取特征

卷积神经网络

② 卷积层（Convolutional Layer）

◆ 零填充（Zero-padding）：也被称为泛卷积，→为了保证卷积后特征图的大小与原图一致，即不丢弃原图信息，尤其是原始图片中的边缘信息。执行0-填充



卷积神经网络

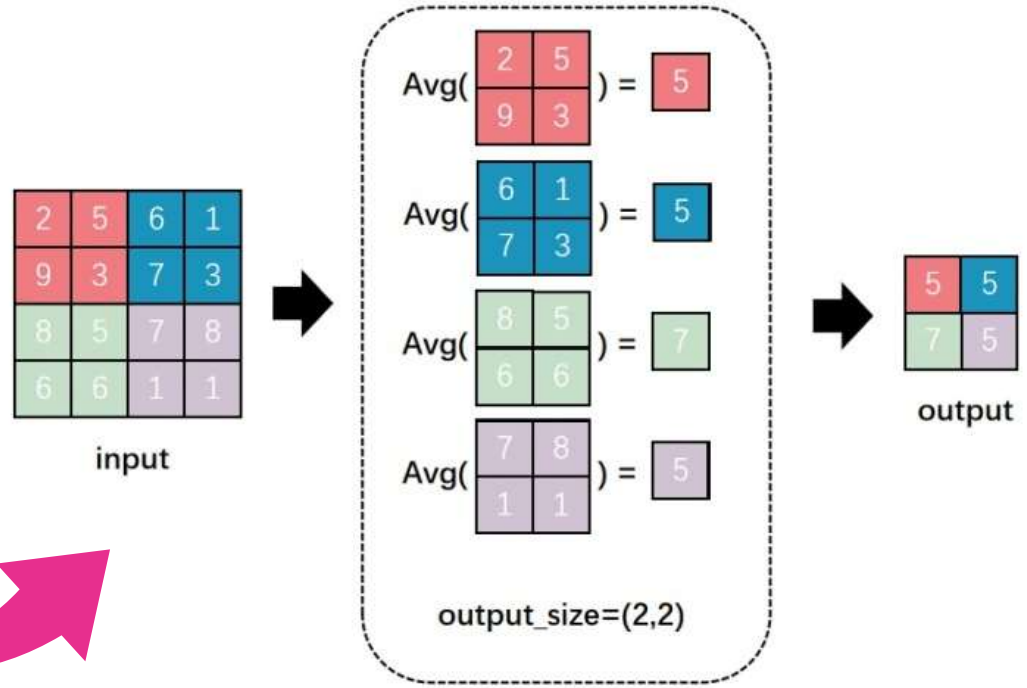
③ 池化层（Pooling layer）

◆ 池化层：也被称为下采样或降采样。通过池化层，减少下一步待处理的数据量，并尽量保留有用信息，加快CNN网络运算过程。池化层根据实际情况可省略。

■ 常用的池化操作: (1) 最大池化； (2) 均值池化。

> ➢ 最大池化： $S_i = \max\{v_i, v_{i+1}, v_{i+2}, \cdots v_{i+k}\}$
>
> ➢ 均值池化： $S_i = \text{mean}\{v_i, v_{i+1}, v_{i+2}, \cdots v_{i+k}\}$

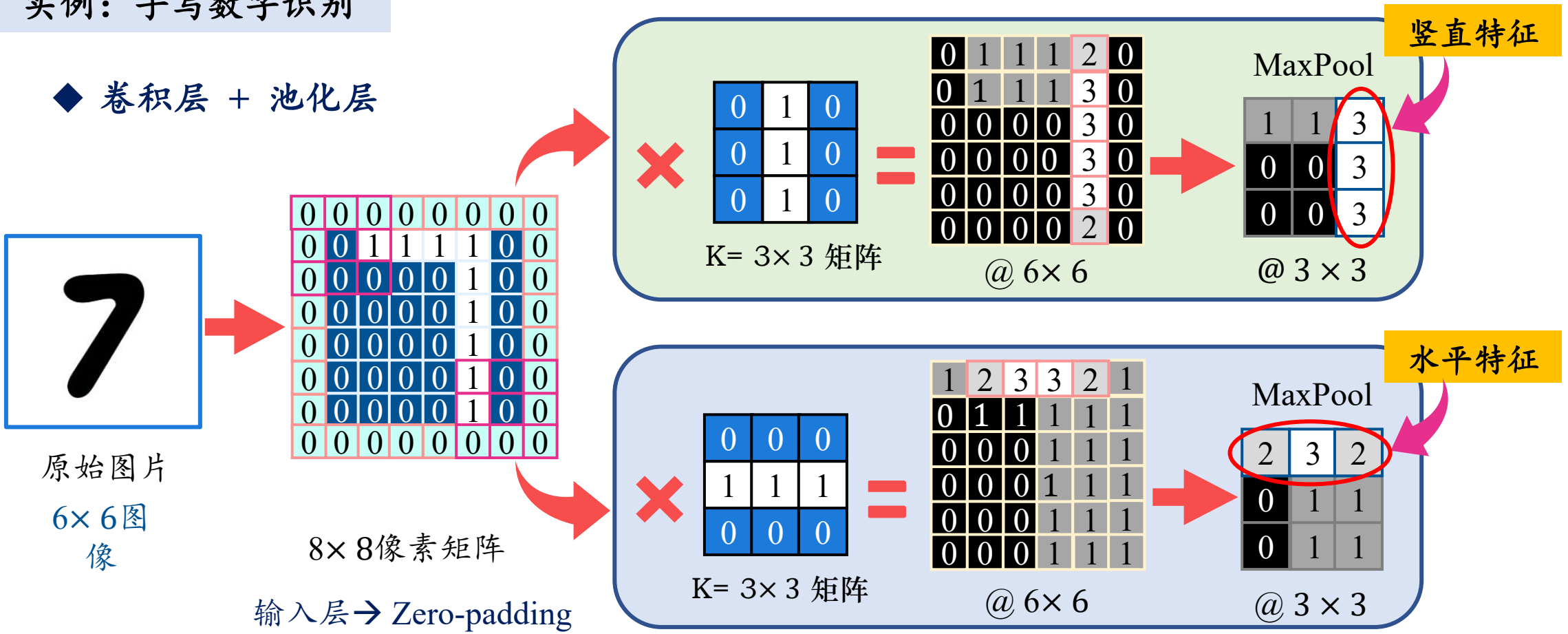
如图，采用均值池化，池化窗口为2×2，池化步长为2。池化后的数据为原有数据的1/4。



卷积神经网络

实例：手写数字识别

◆ 卷积层 + 池化层

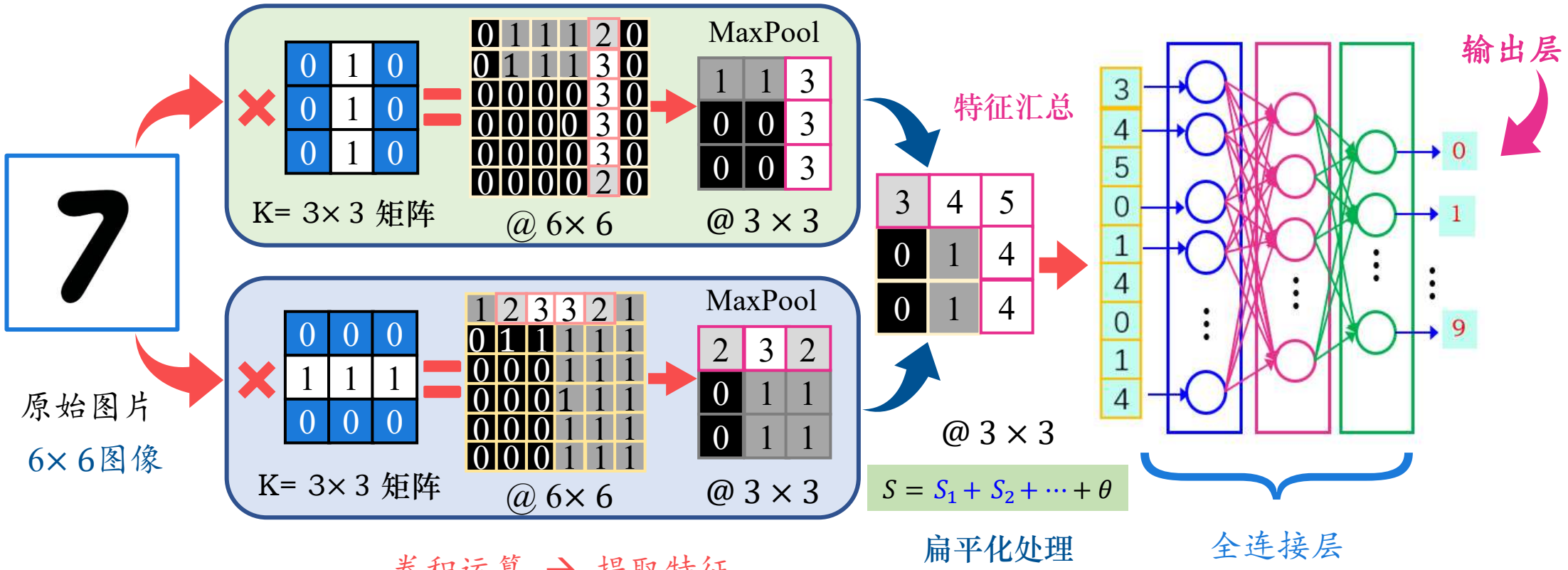


Step 1： 输入层 → Step 2： 卷积层 → Step 3： 池化层

卷积神经网络

④ 全连接层（Full connection layer）

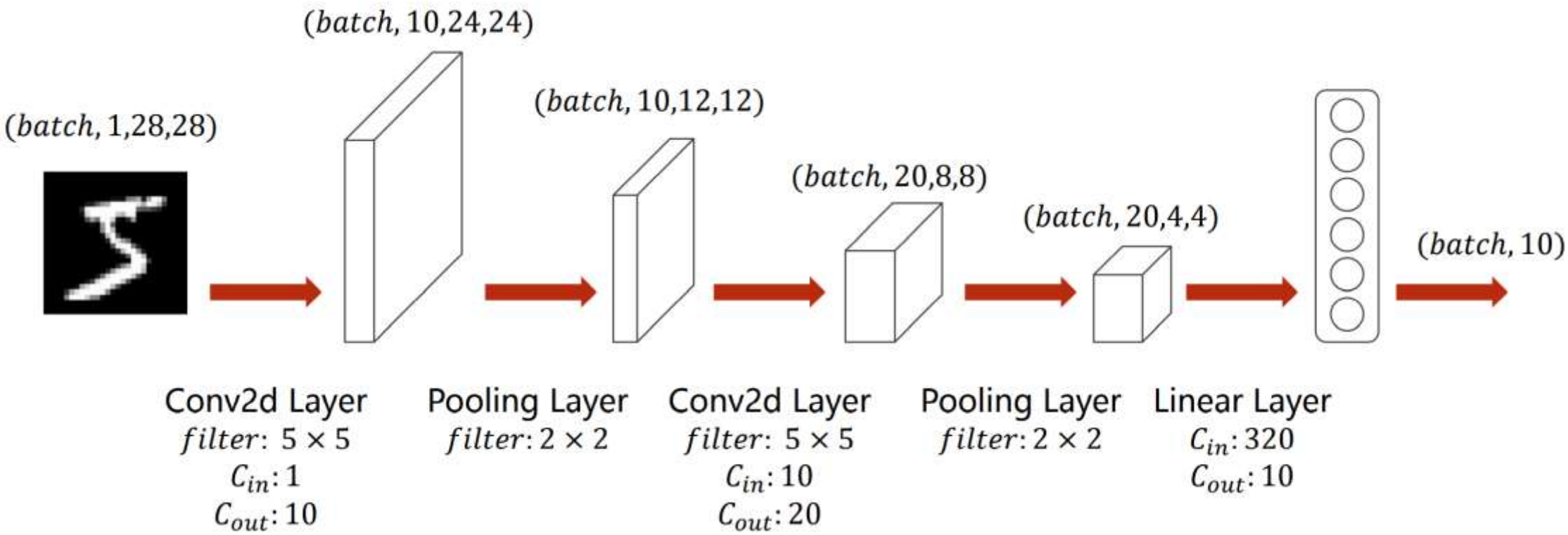
◆ 全连接层：实质是一个BP神经网络或多层感知器。卷积层提取的特征图（矩阵）经过扁平化处理后，特征输入全连接层，实现样本分类或回归预测。



卷积神经网络

■ **CNN神经网络应用案例 ---手写数字识别**

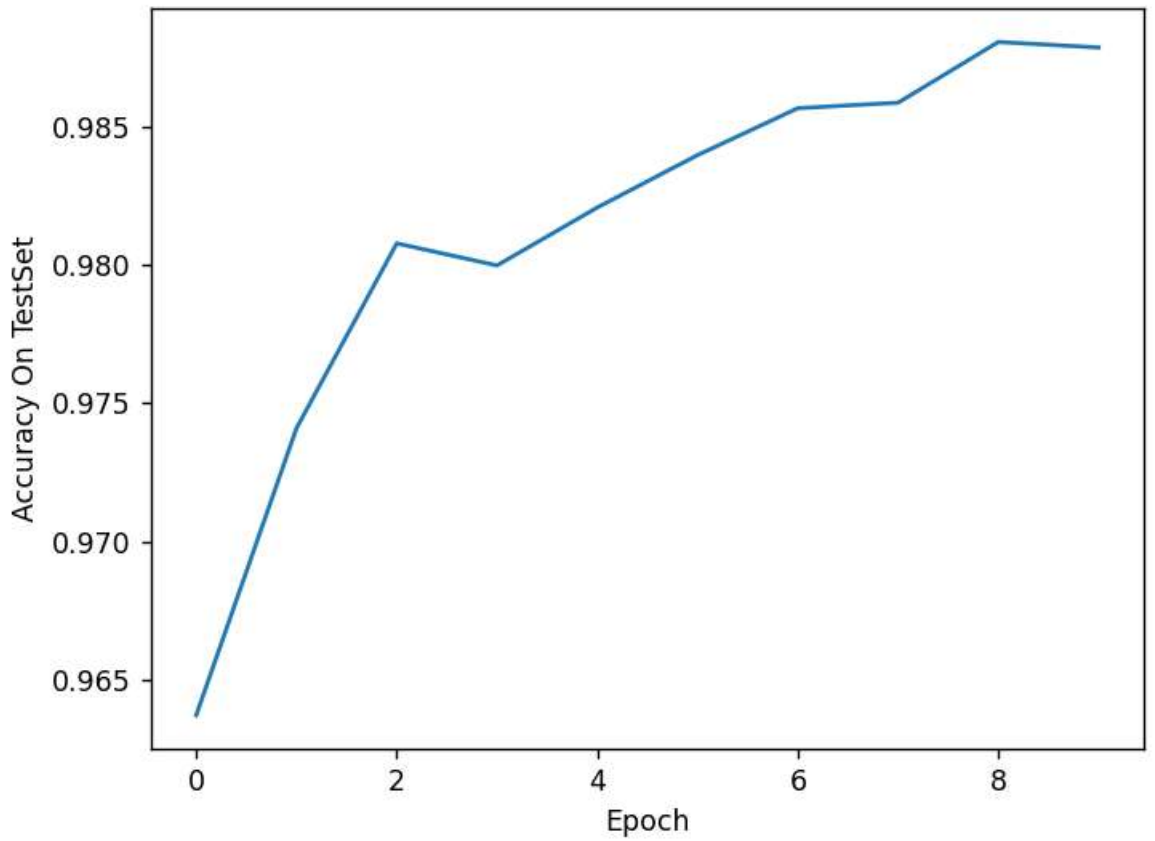
案例：设计一个CNN神经网络对数字0至9进行分类



卷积神经网络

■ **CNN神经网络应用案例 ---手写数字识别**

案例：设计一个CNN神经网络对数字0至9进行分类




```
[7 / 10]: Accuracy on test set: 98.6 %
[8,   300]: loss: 0.045 , acc: 98.66 %
[8,   600]: loss: 0.045 , acc: 98.58 %
[8,   900]: loss: 0.046 , acc: 98.48 %
[8 / 10]: Accuracy on test set: 98.6 %
[9,   300]: loss: 0.039 , acc: 98.85 %
[9,   600]: loss: 0.041 , acc: 98.71 %
[9,   900]: loss: 0.045 , acc: 98.64 %
[9 / 10]: Accuracy on test set: 98.8 %
[10,   300]: loss: 0.038 , acc: 98.81 %
[10,   600]: loss: 0.041 , acc: 98.60 %
[10,   900]: loss: 0.040 , acc: 98.83 %
[10 / 10]: Accuracy on test set: 98.8 %
```

03

YOLO与目标识别

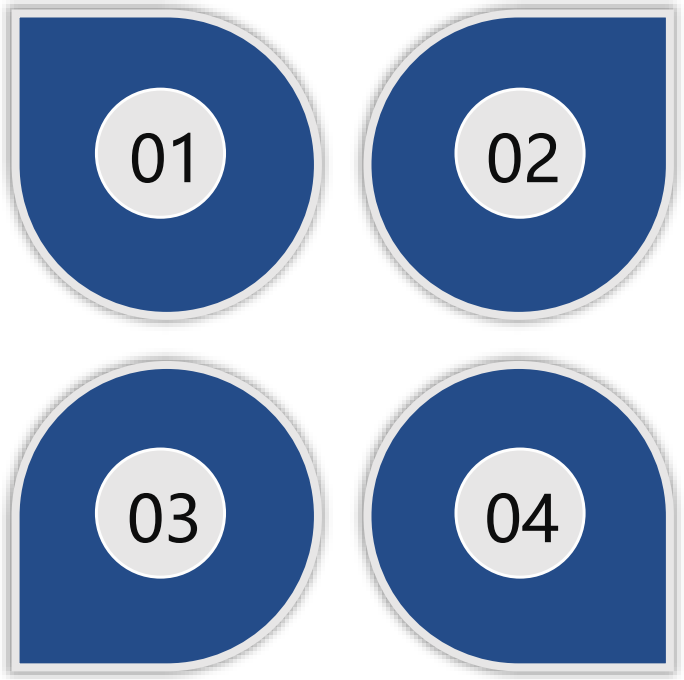
YOLO与目标识别

- **分类、定位、检测、分割的区别**

分类

• 目标：判断图像中是否包含某个类别

• 举例：一张图像被分类为“猫”或“狗”



定位

• 目标：不仅分类，还要指出目标在图像中的位置

• 举例：图像中是一只“猫”，其坐标框为(50, 50, 200, 200)

检测

• 目标：识别图像中所有目标对象的类别和位置

• 举例：图像中有“2只猫+1只狗”，每个目标都有框和类别

分割

语义分割（Semantic Segmentation）

• 目标：将图像中每个像素赋予一个类别标签

• 举例：整张图中所有“猫”的像素都标成“猫”，不管有几只

实例分割（Instance Segmentation）

• 目标：精确地分割每个目标实例，区分同类中的不同个体

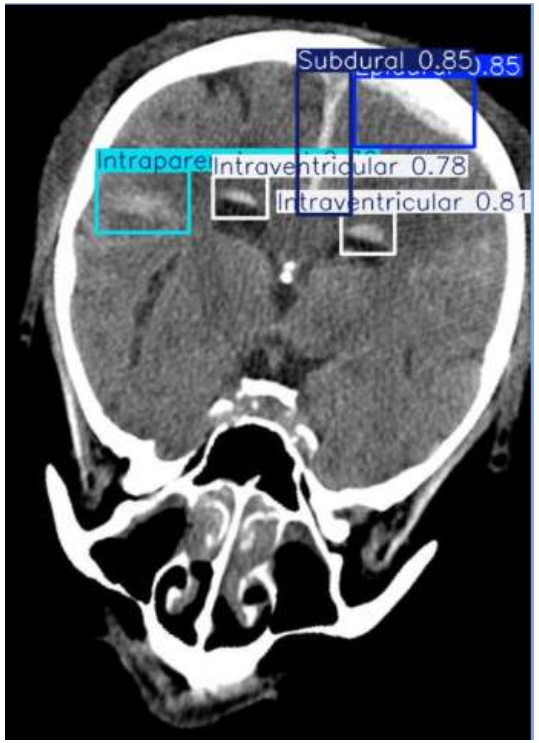
• 举例：图中两只猫分别标成“猫1”和“猫2”

YOLO与目标识别

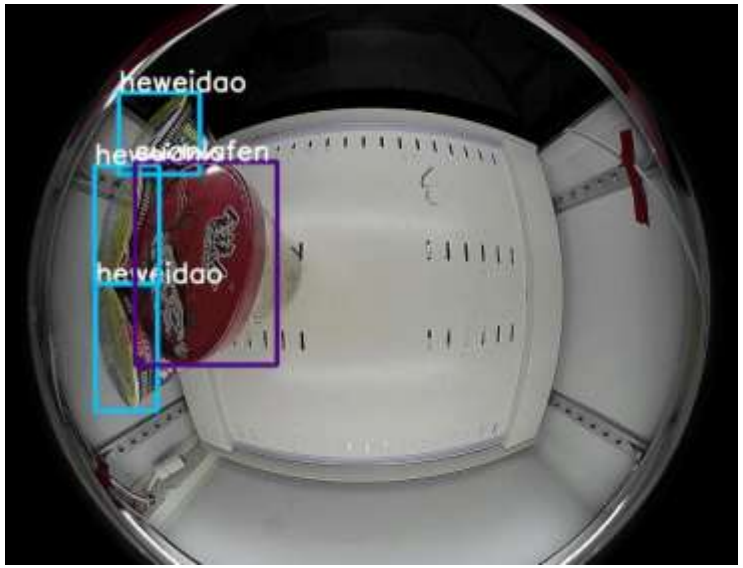
- **YOLO算法定义**

YOLO（You Only Look Once）是一种基于深度学习的目标检测算法。YOLO的核心思想是将目标检测任务看作一个单一的回归问题，直接从图像像素到边界框坐标和类别概率的映射。这意味着YOLO只需要对图像进行一次前向传播，就可以同时预测多个边界框和它们的类别，从而大大减少了计算量。

自动驾驶



医疗影像分析



零售与物流

YOLO与目标识别

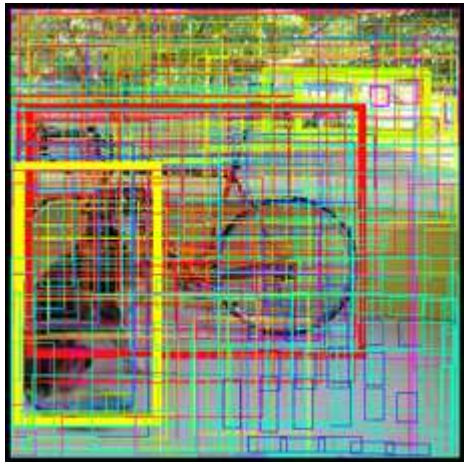
- YOLO的工作机制

划分网格

YOLO首先将输入图像划分为SxS的网格。

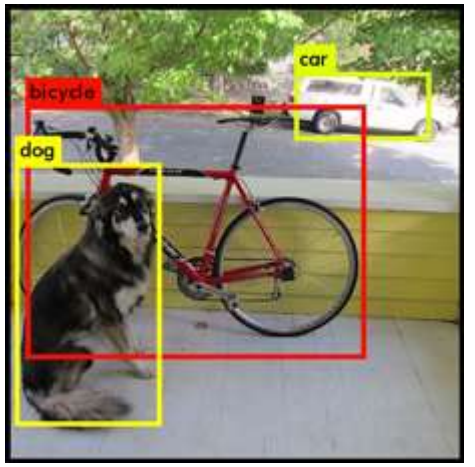
预测边界框和类别概率

每个网格会预测B个边界框，同时输出边界框的坐标、置信度以及类别信息。目标置信度表示该边界框内存在目标的可能性大小，而类别概率则用于判断目标属于哪一个类别。



过滤和选择

在预测结束后，YOLO会使用非极大值抑制（NMS）来过滤和选择最终的边界框。将所有置信度低于某个阈值和高度重叠的边界框去除，保留置信度最高的边界框，从而得到最终准确的检测结果。



YOLO与目标识别

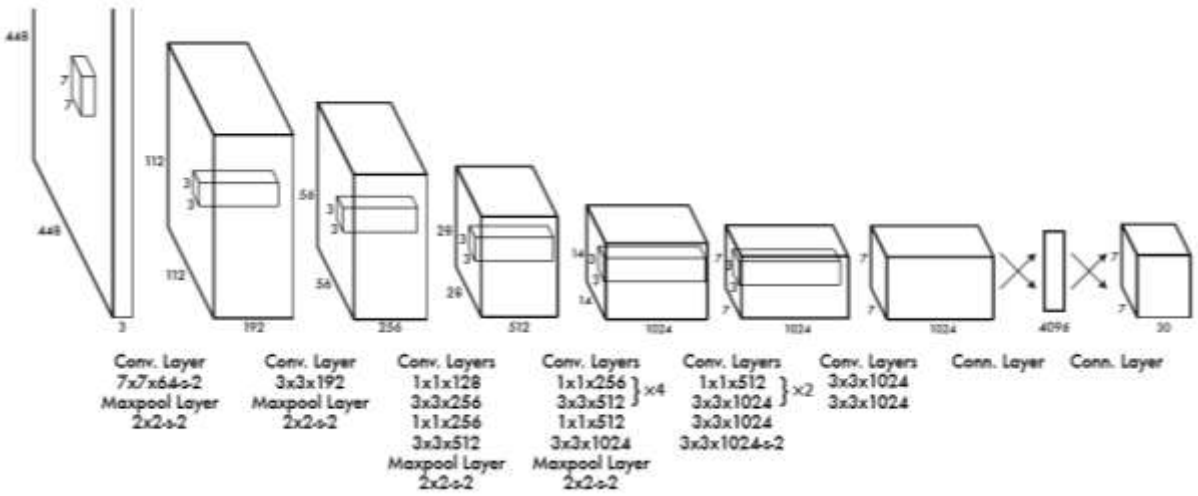
- YOLO的卷积神经网络结构：24个卷积层与2个全连接层

卷积层

YOLO网络包含24个卷积层，用于提取图像的特征，这些卷积层通过不同的卷积核和步长，逐步提取出图像的高层次特征。

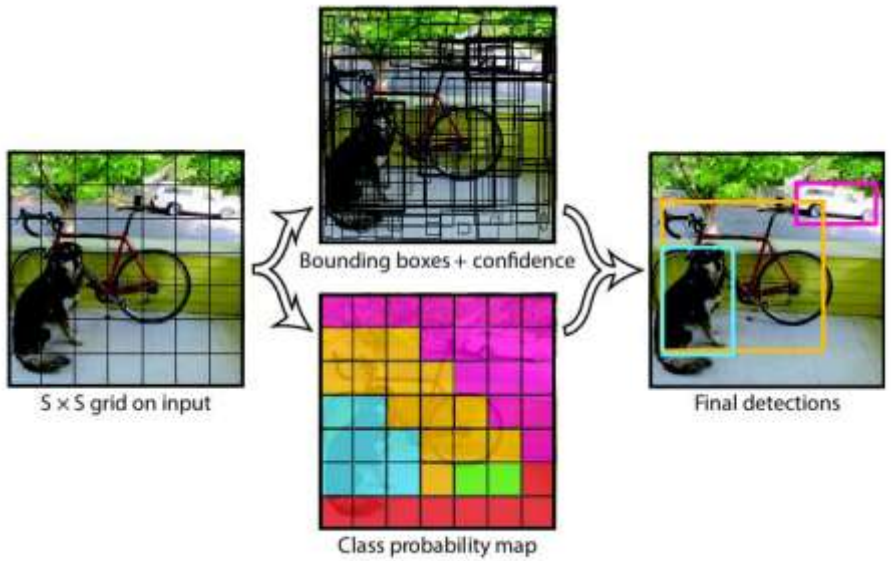
全连接层

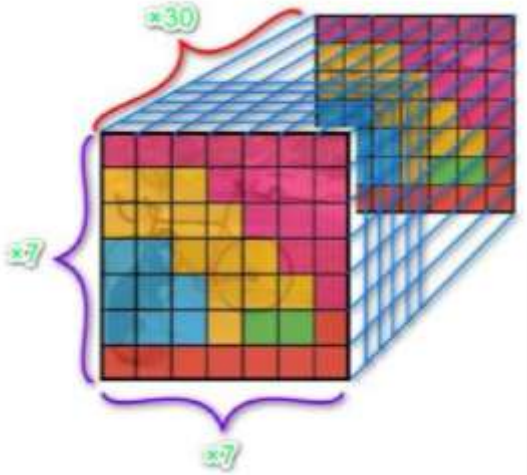
在卷积层之后，YOLO网络包含2个全连接层，用于将提取的特征映射到最终的输出，即边界框和类别概率。

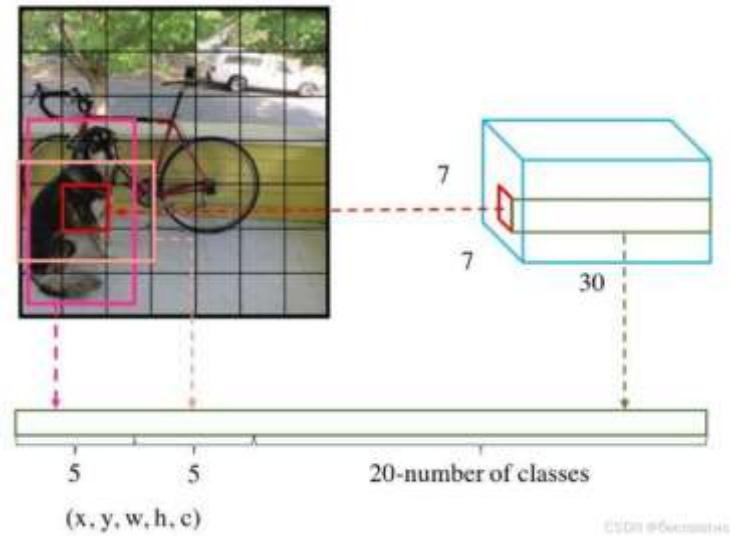


YOLO与目标识别

- YOLO的输出格式：7x7x30的特征图







网格划分

YOLO将输入图像划分为7x7的网格，每个网格负责预测一定数量的边界框和类别概率。

特征图

每个网格输出一个30维的特征向量，其中包含边界框的坐标、置信度和类别概率，这些信息共同构成了7x7x30的特征图。

04

数据标注

数据标注

一、YOLO数据集格式及分类

数据集格式

图像文件：数据集中的图像文件，通常是jpg或png格式。

YOLO数据格式规范

标注文件：一个文本文件，包含了每张图像中目标对象的类别和位置信息。

类别文件：一个文本文件，包含了数据集中所有目标对象的类别信息。

数据集分类

训练集：用于模型的训练和参数调整，占总数据的60-80%左右。train

验证集：用于超参数的调整和模型选择，占总数据的10-20%左右。val

测试集：用于最终模型性能的评估，占总数据的10-20%左右。

数据标注

- 数据集标签结构

VOC格式（XML）

VOC（Visual Object Classes）数据集的标注文件采用XML格式。XML文件详细记录了图片中每个目标对象的类别、位置（通过边界框表示）等信息。VOC数据集结构清晰，易于解析，广泛应用于目标检测模型的训练和评估。

COCO格式（JSON）

COCO（Common Objects in Context）数据集的标注文件采用JSON格式。COCO数据集更加复杂，不仅包含了目标检测任务，还涵盖了图像分割、关键点检测等多种任务。JSON格式的标注文件结构灵活，能够存储更多的信息，适用于复杂的视觉任务。

YOLO格式（TXT）

YOLO算法通常采用TXT格式的标注文件。TXT文件简单明了，每行代表一个目标对象的信息，包括类别编号和边界框的坐标（通常是中心点的x、y坐标以及边界框的宽度和高度）。这种格式便于YOLO算法快速读取和处理数据。

<object-class> <x> <y> <width> <height>

数据标注

- 图像与标签的对应规则

```
dataset/
├── images/
│   ├── train/
│   │   ├── image1.jpg
│   │   ├── image2.jpg
│   │   └── ...
│   └── val/
│       ├── image3.jpg
│       ├── image4.jpg
│       └── ...
└── labels/
    ├── train/
    │   ├── image1.txt
    │   ├── image2.txt
    │   └── ...
    └── val/
        ├── image3.txt
        └── image4.txt
```

01 文件命名要一致，图像和标签的文件名相同，只是后缀名和文件夹不同。

02 图像和标签一 一对应，每张图像只能有一个对应的标签文件，标签文件描述图像中所有目标的位置和类别。

03 标签文件使用归一化坐标系统，图像尺寸可能不同，但标签中的坐标都被归一化为 [0,1] 范围，避免尺寸不一致导致信息不匹配。

数据标注

二、LabelImg

- LabelImg:是一款开源图像标注工具，方便地对图像中的目标进行标注，标签可用于分类和目标检测，并支持多种格式的保存。

- <安装> 步骤：
- labelimg 标注工具使用版本不高于3.9，闪退；使用pyqt5界面，需要安装pyqt5。

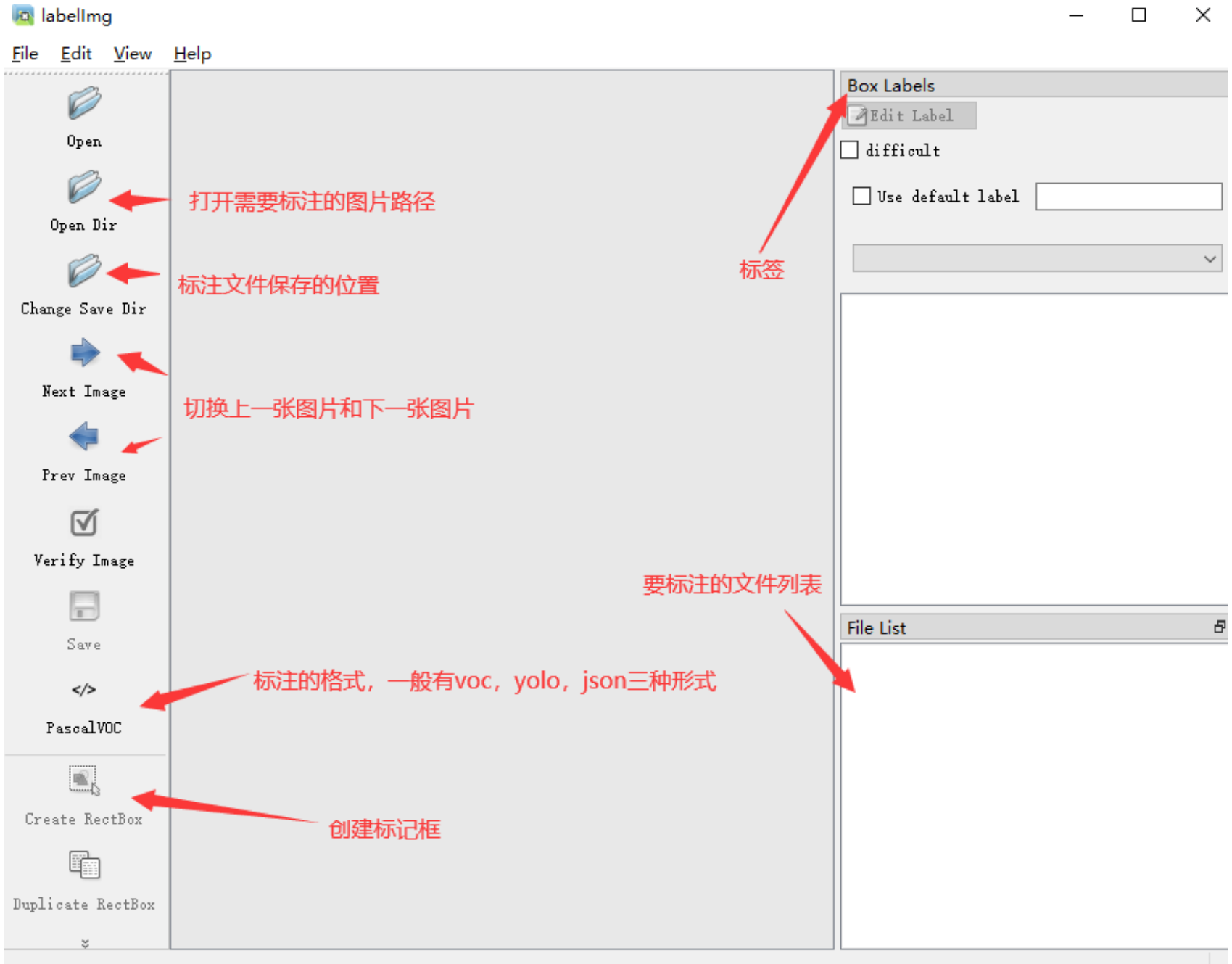
（1）创建虚拟环境（python = 3.8）： conda create –n labelimg python=3.8

（2） 安装pyqt5: pip3 install PyQt5 -i https://pypi.tuna.tsinghua.edu.cn/simple/

（3）安装LabelImg : pip3 install labelimg -i https://pypi.tuna.tsinghua.edu.cn/simple/

数据标注

二、LabelImg



数据标注

二、LabelImg

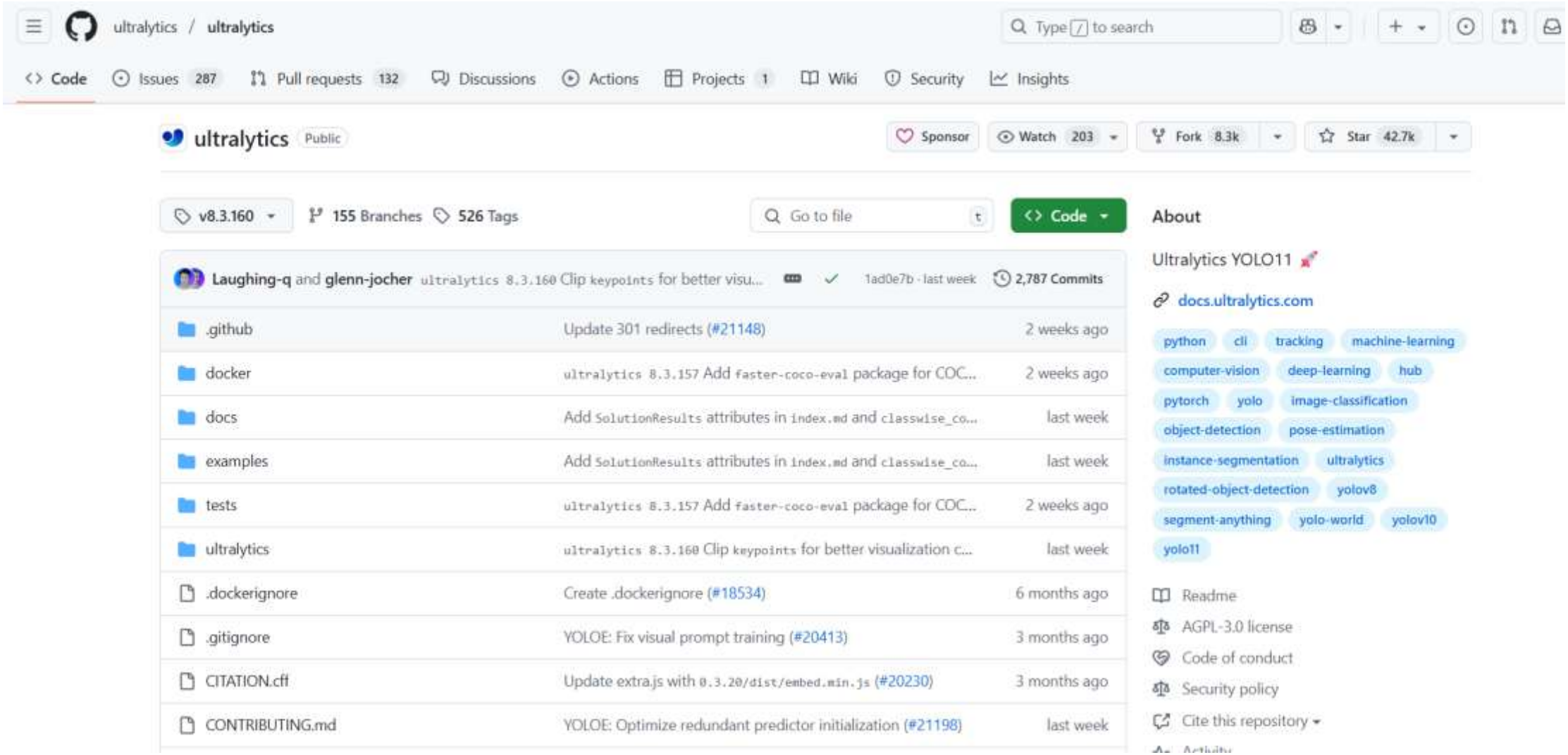
使用教程：
https://zhuanlan.zhihu.com/p/1892897074028188653

05

YOLO模型部署

YOLO模型部署

- 官网：[ultralytics/ultralytics: Ultralytics YOLO11 🚀]

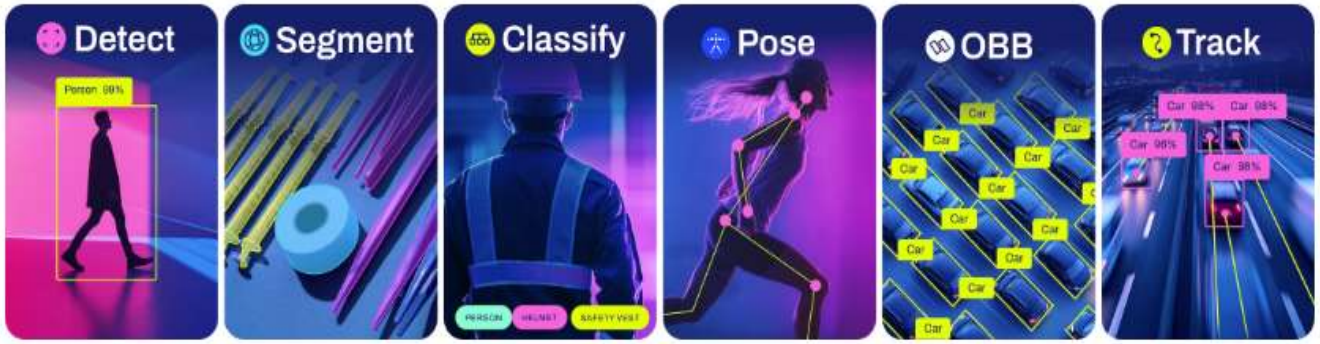


YOLO模型部署

- **Yolov11 预训练模型**

✨ Models

Ultralytics supports a wide range of YOLO models, from early versions like YOLOv3 to the latest YOLO11. The tables below showcase YOLO11 models pretrained on the COCO dataset for Detection, Segmentation, and Pose Estimation. Additionally, Classification models pretrained on the ImageNet dataset are available. Tracking mode is compatible with all Detection, Segmentation, and Pose models. All Models are automatically downloaded from the latest Ultralytics release upon first use.



- ▶ Detection (COCO)
- ▶ Segmentation (COCO)
- ▶ Classification (ImageNet)
- ▶ Pose (COCO)
- ▶ Oriented Bounding Boxes (DOTAv1)

- detect：目标检测
- segment：图像分割
- pose：姿态估计
- classify：图像分类
- obb：倾斜边界框检测

YOLOv11n 是 YOLOv11 系列中的轻量级版本，参数量较少，计算资源需求较低，非常适合在资源受限的环境（如嵌入式设备、移动设备）上进行推理。虽然模型较小，但依然可以在精度和速度之间取得良好的平衡。

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed T4 TensorRT10 (ms)	params (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

YOLO模型部署

- Yolov11 目标检测