



深度学习实战与计算视觉：数值图像 处理与卷积神经网络分类任务

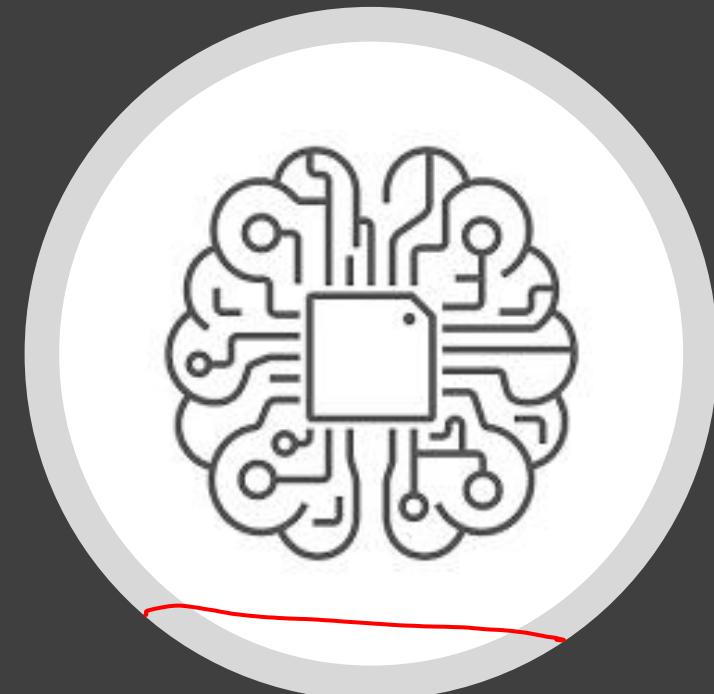
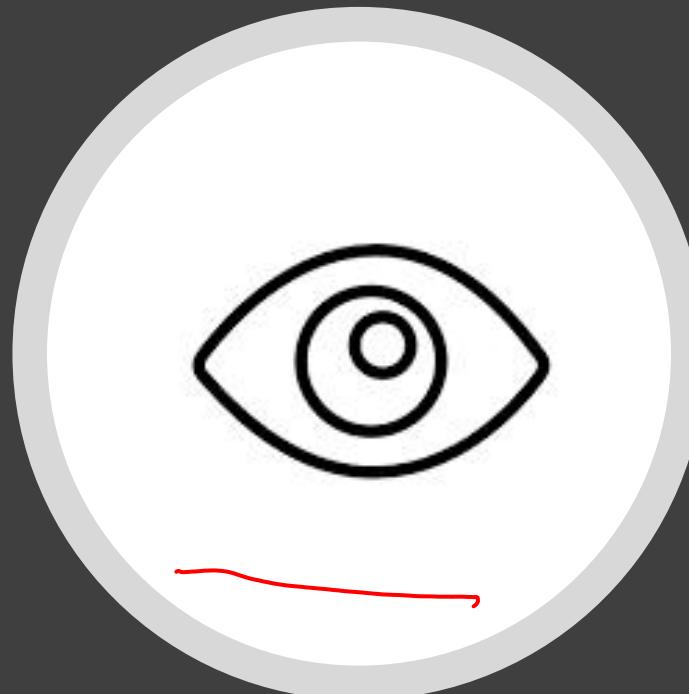
第一章：课程导学，深度学习计算视觉简介与课程介绍

Content

- 1 --课程导学
- 2 --讲师介绍
- 3 --计算视觉与人工智能深度学习
- 4 --深度学习在计算视觉的前沿应用
- 5 --课程设计和体系
- 6 --算法的基本实现框架与工具

Content

- 1 --课程导学
- 2 --讲师介绍
- 3 --计算视觉与人工智能深度学习
- 4 --深度学习在计算视觉的前沿应用
- 5 --课程设计和体系
- 6 --算法的基本实现框架与工具



人工智能和计算视觉为何如此重要

人工智能与计算视觉的行业前景

中华人民共和国中央人民政府

国务院 总理 新闻 政策 互动 服务 数据 国情 国家政务服务平台

www.gov.cn

索引号：00001439/2017-00142
发文机关：国务院
标题：国务院关于印发新一代人工智能发展规划的通知
发文字号：国发〔2017〕35号
主题词：

主题分类：科技、教育\科技
成文日期：2017年07月08日
发布日期：2017年07月20日

国务院关于印发
新一代人工智能发展规划的通知
国发〔2017〕35号

各省、自治区、直辖市人民政府，国务院各部委、各直属机构：
现将《新一代人工智能发展规划》印发给你们，请认真贯彻执行。

国务院
2017年7月8日

相关报道
国务院印发《新一代人工智能发展规划》
图解
国务院印发《新一代人工智能发展规划》

国务院印发《新一代人工智能发展规划》

国务院近日印发《新一代人工智能发展规划》，明确了我国新一代人工智能发展的战略目标：

到2020年

- ▶ 人工智能总体技术和应用与世界先进水平同步
- ▶ 人工智能产业成为新的重要经济增长点
- ▶ 人工智能技术应用成为改善民生的新途径

到2025年

- ▶ 人工智能基础理论实现重大突破
- ▶ 部分技术与应用达到世界领先水平
- ▶ 人工智能成为我国产业升级和经济转型的主要动力
- ▶ 智能社会建设取得积极进展

到2030年

- ▶ 人工智能理论、技术与应用总体达到世界领先水平，成为世界主要人工智能创新中心

新华社记者 崔迎 编制

From Mobile first to AI first – Google I/O 2017 conference.

May 19, 2017 - **Google make transition from mobile first to AI first**, which means that main users problems will be solved through **Artificial Intelligence**. Moreover ...

Google CEO Sundar Pichai hosted his second Google I/O this year. It was announced that Google is moving from "Mobile first" to "AI first" approach. And, they are doing this across their products. So today, if you're using Google Maps, street view automatically recognizes restaurant signs, street signs, using machine learning. They are re-imagining their entire suite of products. They want AI and machine learning to form the basis of all their products and services.

人工智能与计算视觉的行业前景

华为200万年薪聘请人工智能专业博士生,人工智能行业发展不容忽视

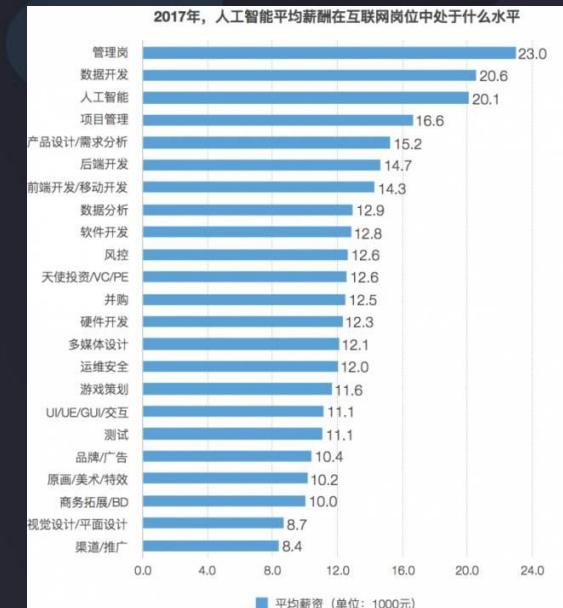
华为总裁任正非:人工智能时代,顶级的薪酬吸引顶尖人才值得!

人工智能正在取代华尔街的高薪工作

142858美元: 2019年薪酬最高的人工智能职位的平均值

到2022年, 人工智能创造了1.33亿个新工作岗位

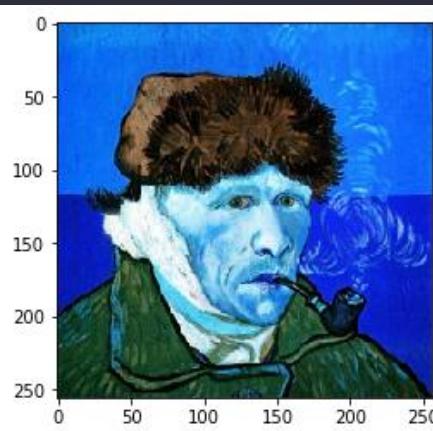
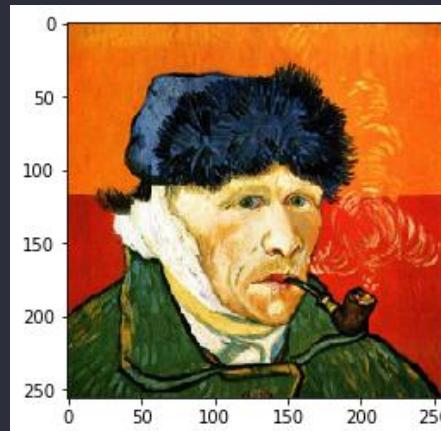
从2018年5月到2019年5月, 人工智能就业人数增长了29%——虽然仍在显著增长, 但一年前和两年前分别增长了58%和36%。与此同时, 人工智能求职者的数量下降了15%, 这表明提供这些人工智能职位的公司可能缺少人工智能专家。



企业	薪酬	岗位	候选人背景
腾讯	年薪60万+北京户口	机器学习基础研究	985博士, TOP硕博
腾讯	年薪60万+深圳户口	图像识别算法研究	港系博士+顶会Paper
百度	月薪30K-35K月薪	图像识别算法研究	985博士/海归博士
微软	年薪50-55万	机器学习基础研究	TOP2博士
谷歌	年薪50-55万	人工智能研究员	TOP2博士
美团	2K月薪+北京户口(北斗计划)	机器学习基础研究	985硕士, 专业rank前3
滴滴	年薪50万+	研究员	985博士
滴滴	月薪25K(新锐计划)	算法工程师	985硕士
今日头条	30K-35K+住房补助	AI Lab研究员	博士
网易	年薪45万	人工智能研究员	TOP2硕士
华为	50万(25K月薪+奖金+补助)	算法研究员	211博士
大疆	月薪35K	算法研究员	985博士
Face++	年薪35-40万	Researcher	985硕士
Face++	年薪50万+	Researcher	博士学历

|市面上常见课程的一些不足

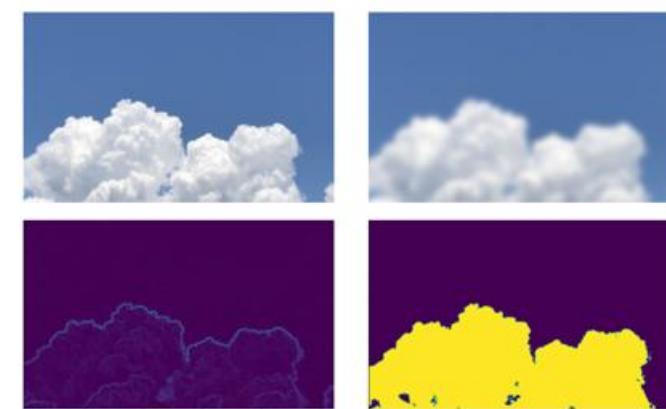
1. 缺少对于图像处理底层知识的学习
2. 经典算法和深度学习被割裂开来
3. 大而全但是深度和实战不足/针对性强但是门槛高，零基础的同学难以上手



图像底层的数值原理



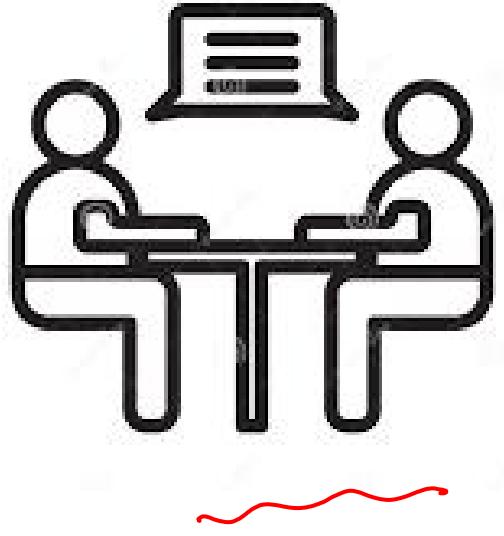
系统化的了解计算视觉



循序渐进的课程设计
极简的实战算法设计



系统的梳理计算视觉的前世今生和里程碑技术



课程的预期目标：

完成课程后，具有在学习，研究，面试或工作中独立开展/演示一个小型计算视觉项目的能力

Content

- 1 --课程导学
- 2 --讲师介绍
- 3 --计算视觉与人工智能深度学习
- 4 --深度学习在计算视觉的前沿应用
- 5 --课程设计和体系
- 6 --算法的基本实现框架与工具

|讲师介绍-Charlie老师



- 人工智能算法科学家
- 2019国家培训计划讲师-面向高职院校老师的人工智能培训课程
- 深圳市海外高层次人才认定(孔雀人才)
- 美国圣地亚哥国家超算中心博士后
- 加利福尼亚大学圣地亚哥全奖博士
- 参与美国自然科学基金(NSF)及加州能源局 (CEC)资助的392MW
IVANPAH等智慧电网项目
- 21篇国际期刊文章(sci收录17篇), 总引用接近1000
- 第一作者发明专利11份



欢迎大家扫码加入本次课程的学习小组讨论群，
老师们会在群里回答同学们关于课程和深度学习
计算视觉的问题

Content

- 1 --课程导学
- 2 --讲师介绍
- 3 --计算视觉与人工智能深度学习
- 4 --深度学习在计算视觉的前沿应用
- 5 --课程设计和体系
- 6 --算法的基本实现框架与工具

人工智能与深度学习

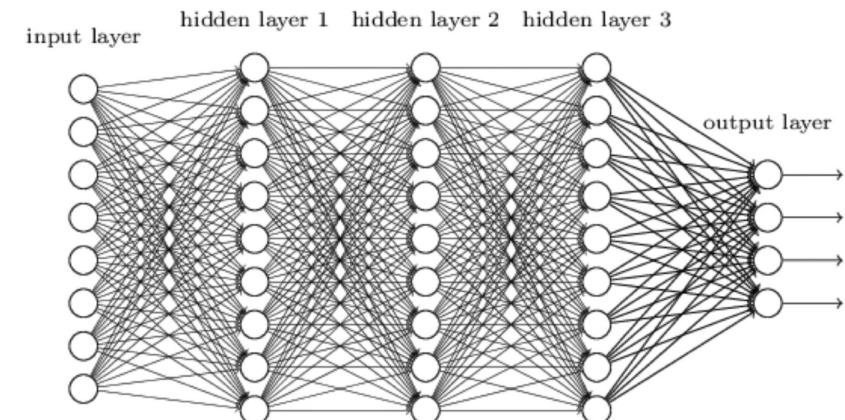
什么是人工智能？

人工智能就其本质而言是对人思维和解决问题过程的模拟



什么是深度学习？

是一种模拟人脑结构，以人工神经网络为架构，提取数据表象后面更高维度与更深逻辑关系从而达到更准确效果的算法



深度学习与计算视觉

人工智能：专家系统/物理
模型等

机器学习：
kNN/SVM等

深度学习：
1.全连接神经网络
2.卷积神经网络
3.循环神经网络

非监督学习

监督式学习

强化学习

大数据分析

计算视觉

自然语言处理

深度强化学习

元学习/类脑科学

计算视觉 (Computer Vision)



计算机视觉是一门研究如何使机器“看”的科学，更进一步的说，就是指用计算机和视觉系统代替人眼对目标进行识别、跟踪和测量等，并进一步做图像处理和分析

Ref:

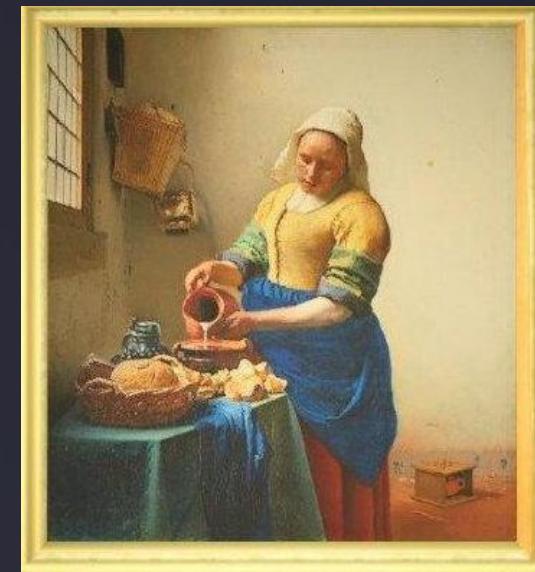
NASA/JPL/Cornell University, Maas Digital LLC - <http://photojournal.jpl.nasa.gov/catalog/PIA04413> (image link)

<https://images.app.goo.gl/KvGmQxq5ivDPvFBD9>

计算视觉 (Computer Vision)

视觉是人脑最主要的信息来源，也是进入人工智能这个殿堂的大门

0	255	255	0
0	0	255	0
0	0	255	0
0	0	255	255



1. 这是什么图片？
2. 这个人是谁？
3. 她在做什么？
4. 桌子上有哪些食物？放在哪里？
5. 背景中哪些是窗户，哪些是墙，哪些是地面？

计算视觉 (Computer Vision)

- (1) 控制过程：指引机器臂/工业机器人
- (2) 导航：自动驾驶或移动机器人
- (3) 检测：视频监控和 人脸识别
- (4) 组织信息：基于图像和图像序列的智能搜索
- (5) 造型对象或环境：医学图像分析系统或地形模型；
- (6) 智能交互：情感能识别，人机交互；



科学与艺术

健康

金融

游戏

机器人

教育

能源

交通

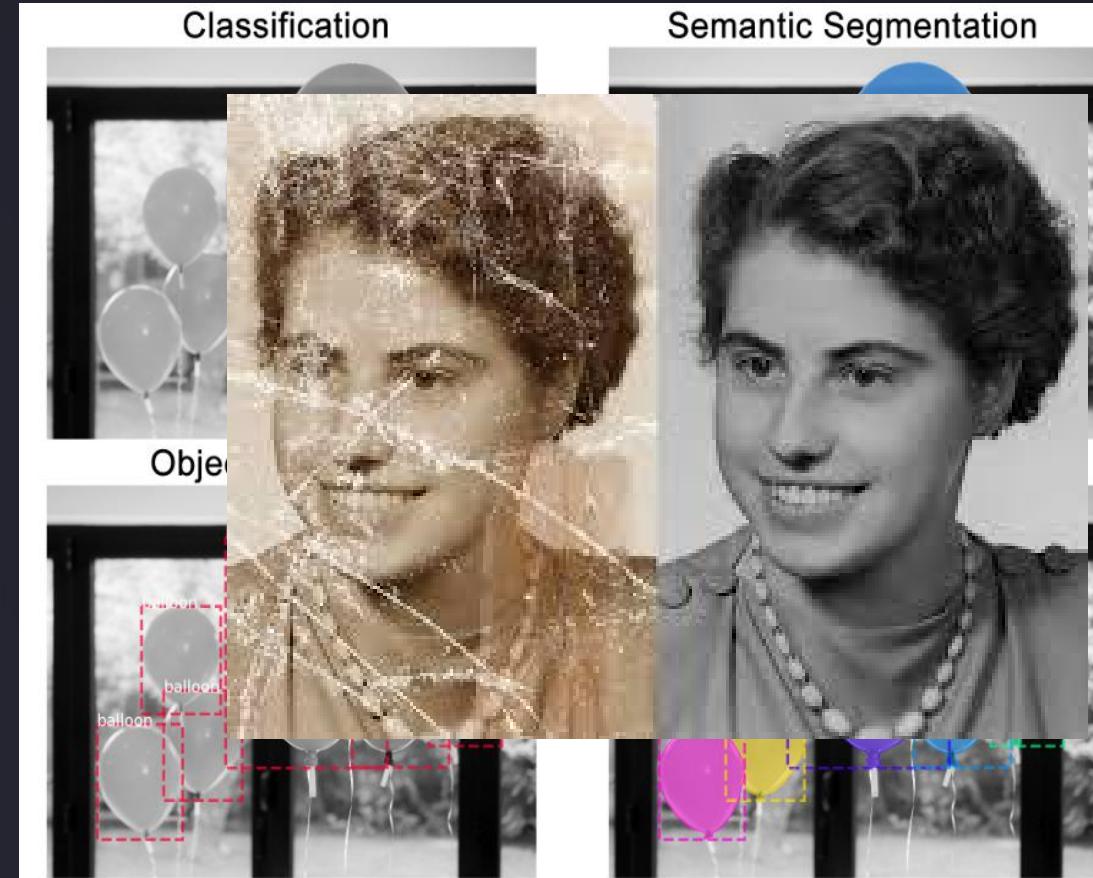
计算视觉 (Computer Vision)

计算视觉四大主流任务

1. 图像分类/识别
2. 语义分割
3. 目标检测
4. 实例分割

其他任务

1. 图片增强
2. 目标追踪
3. 视觉创意



Content

- 1 --课程导学
- 2 --讲师介绍
- 3 --计算视觉与人工智能深度学习
- 4 --深度学习在计算视觉的前沿应用
- 5 --课程设计和体系
- 6 --算法的基本实现框架与工具

|深度学习在计算视觉的前沿应用案例：人脸识别



AI跨年龄识别别人脸仅凭童年照找回走失15年的孩子！

新浪网 - 3 Sep 2019

去年在张学友演唱会拿下80名违法犯罪分子

江苏商报 2019年11月29日 06:00



|深度学习在计算视觉的前沿应用案例：OCR

车牌识别



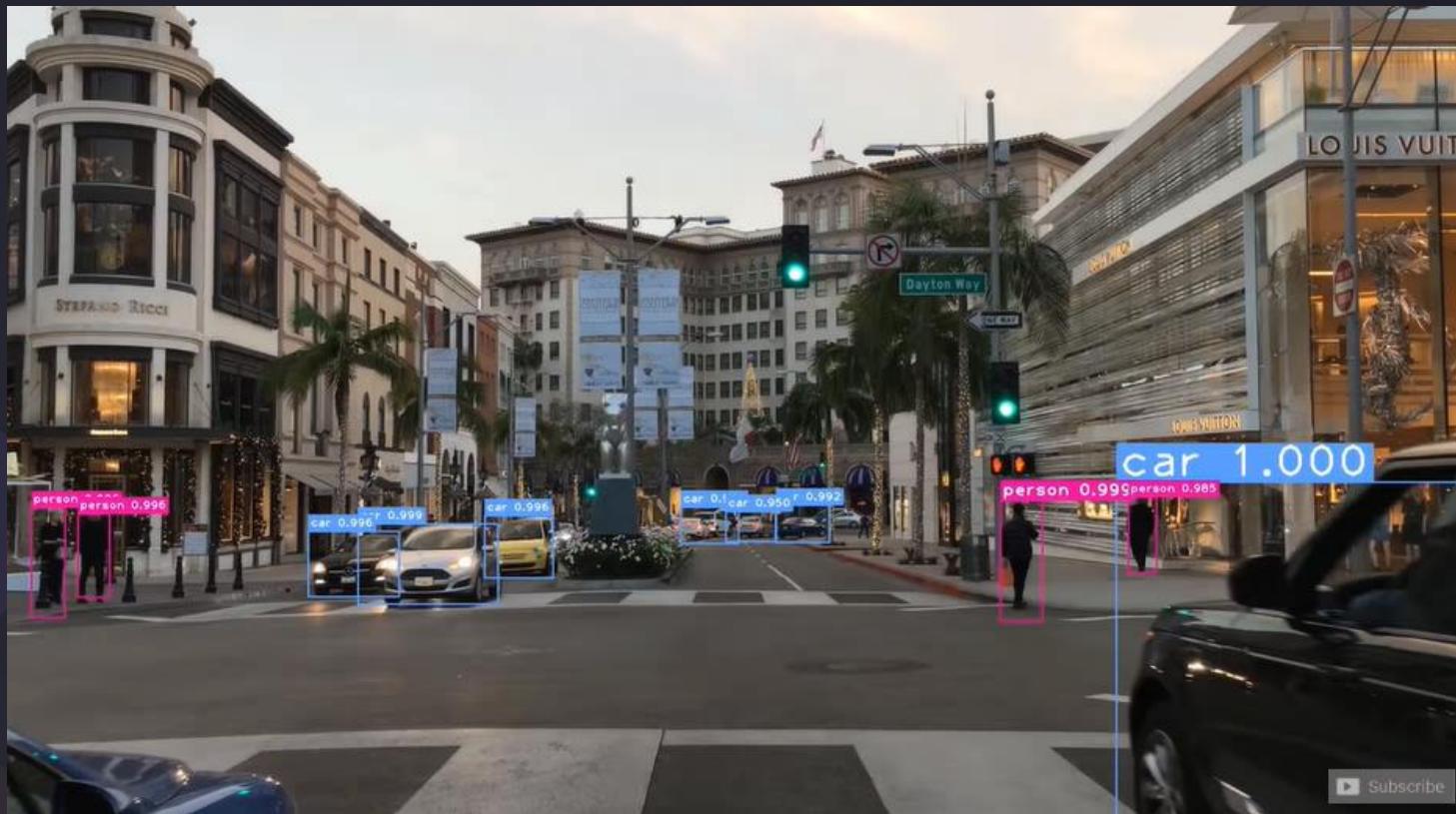
身份证与银行卡识别



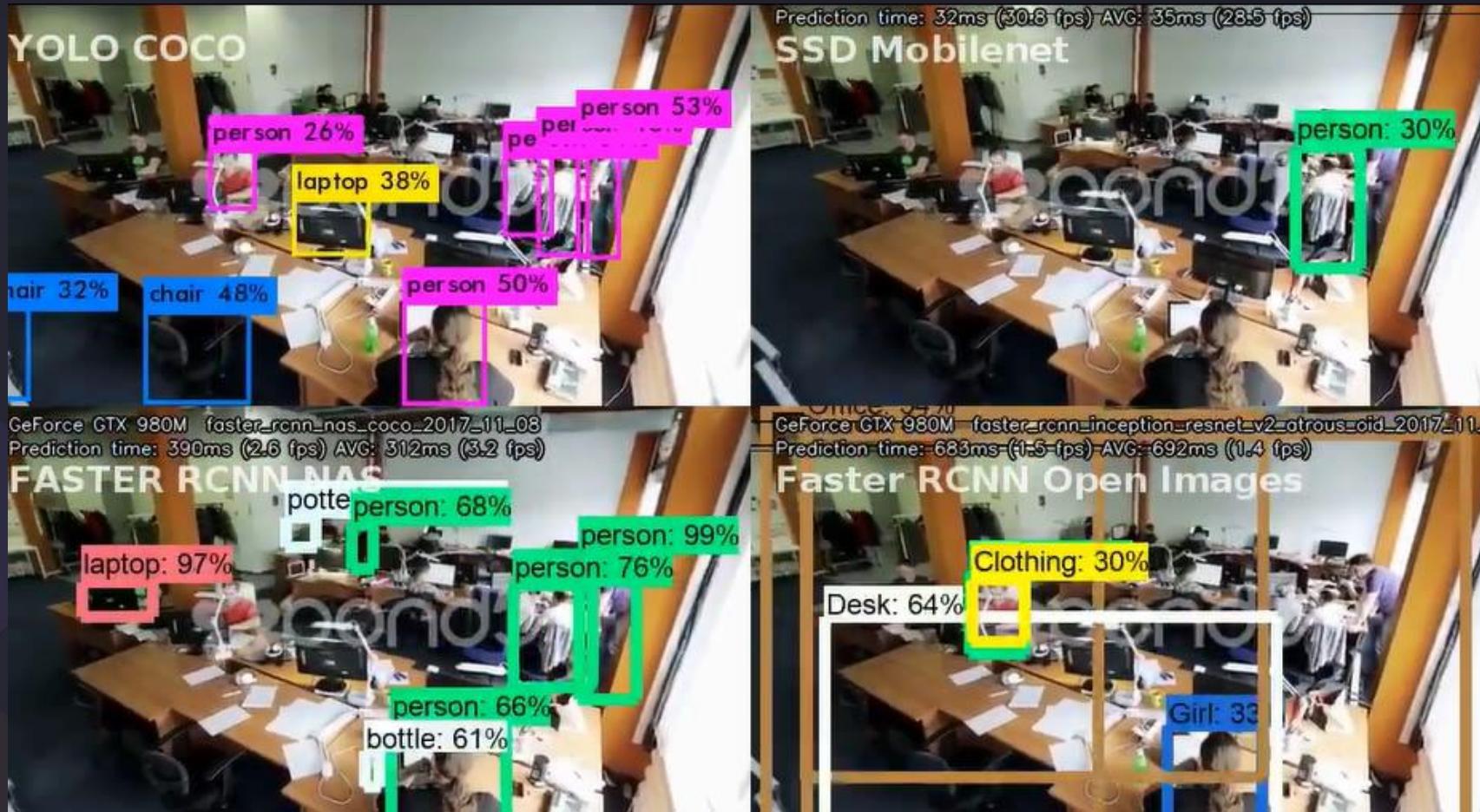
深度学习在计算视觉的前沿应用案例：图像搜索引擎



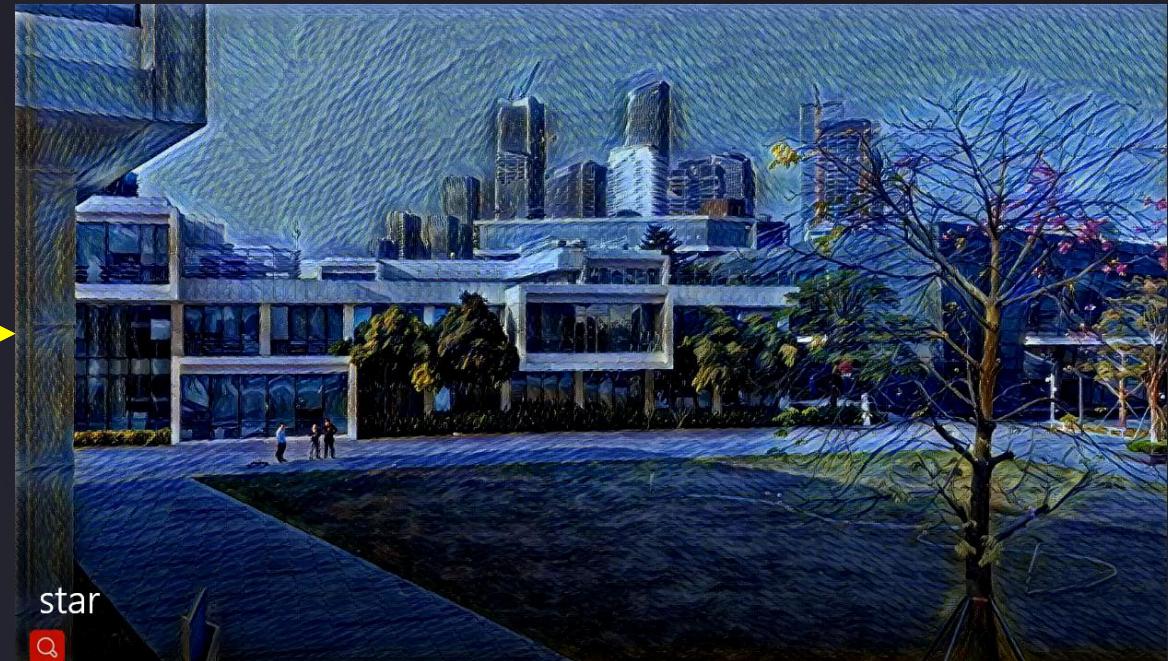
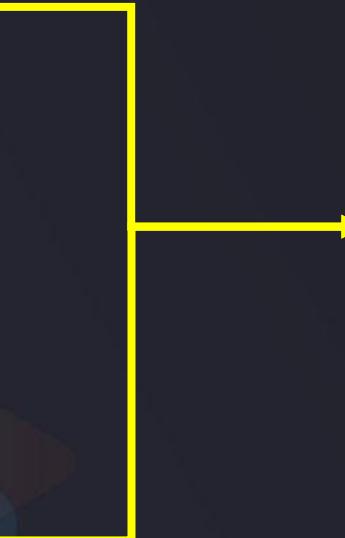
|深度学习在计算视觉的前沿应用案例：自动驾驶



|深度学习在计算视觉的前沿应用案例：智能监控



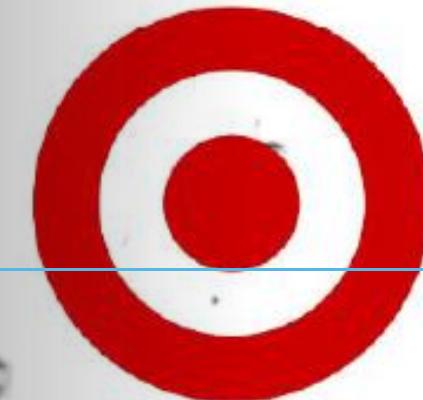
|深度学习在计算视觉的前沿应用案例：视觉创意



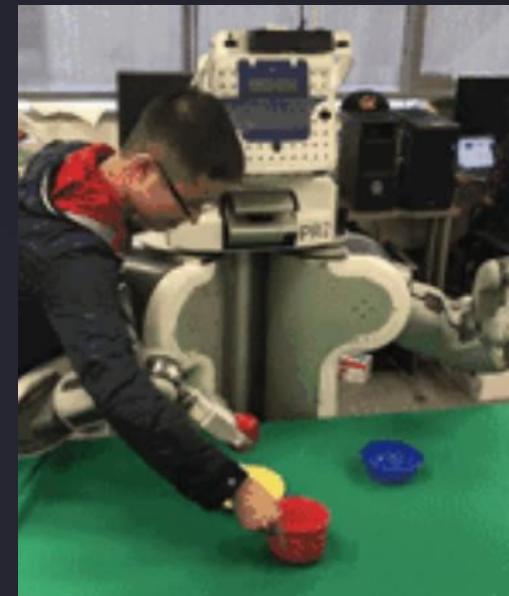
深度学习在计算视觉的前沿应用案例： 金融另类数据

另类数据：通过卫星图识别停车场信息分析预测大型零售企业的年报表现

Katona, Z., Painter, M., Patatoukas, P. N., & Zeng, J. (2018, July). On the capital market consequences of alternative data: Evidence from outer space. In *9th Miami Behavioral Finance Conference*.



深度学习在计算视觉的前沿应用案例：机器臂指引



伯克利ONE-SHOT-LEARNING机器人

谷歌机器人农场

REF:

<http://bair.berkeley.edu/blog/2017/07/18/learning-to-learn/>

谷歌QT-OPT: http://www.sohu.com/a/272905934_651893

Content

- 1 --课程导学
- 2 --讲师介绍
- 3 --计算视觉与人工智能深度学习
- 4 --深度学习在计算视觉的前沿应用
- 5 --课程设计和体系
- 6 --算法的基本实现框架与工具

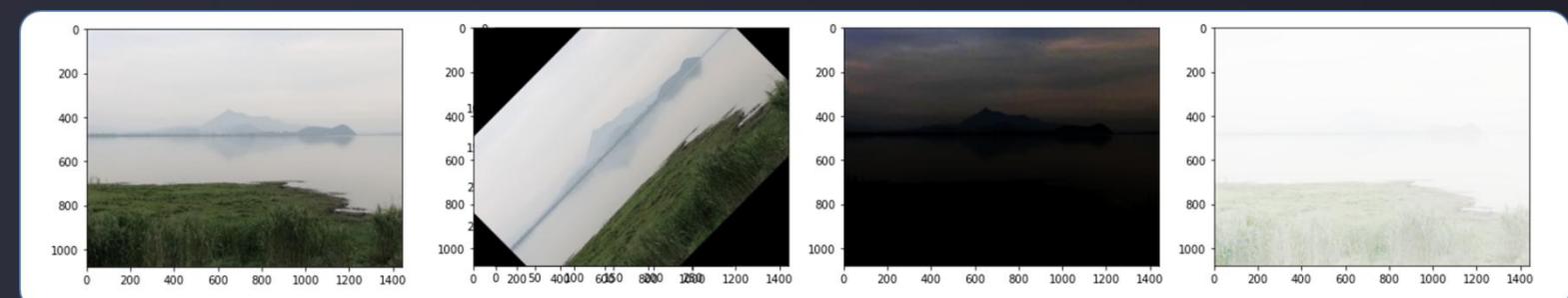
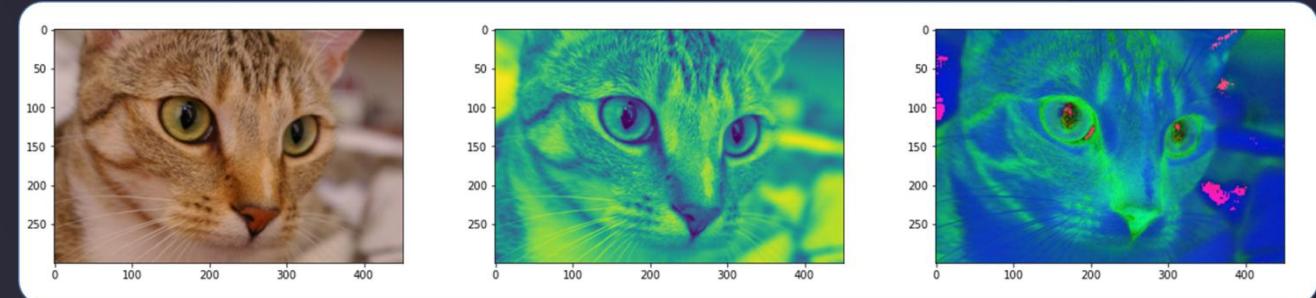
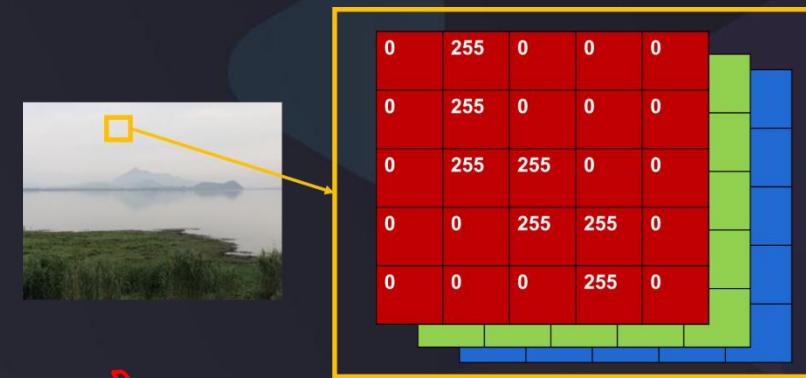
课程设计

- 基础的数值图像处理
- 进阶的经典图像算法
- 卷积神经网络基本原理及优化策略
- 实战搭建一个深度学习神经网络
- 迁移学习和多目标分类理论及实战

课程设计

第二章：基础的数值图像处理

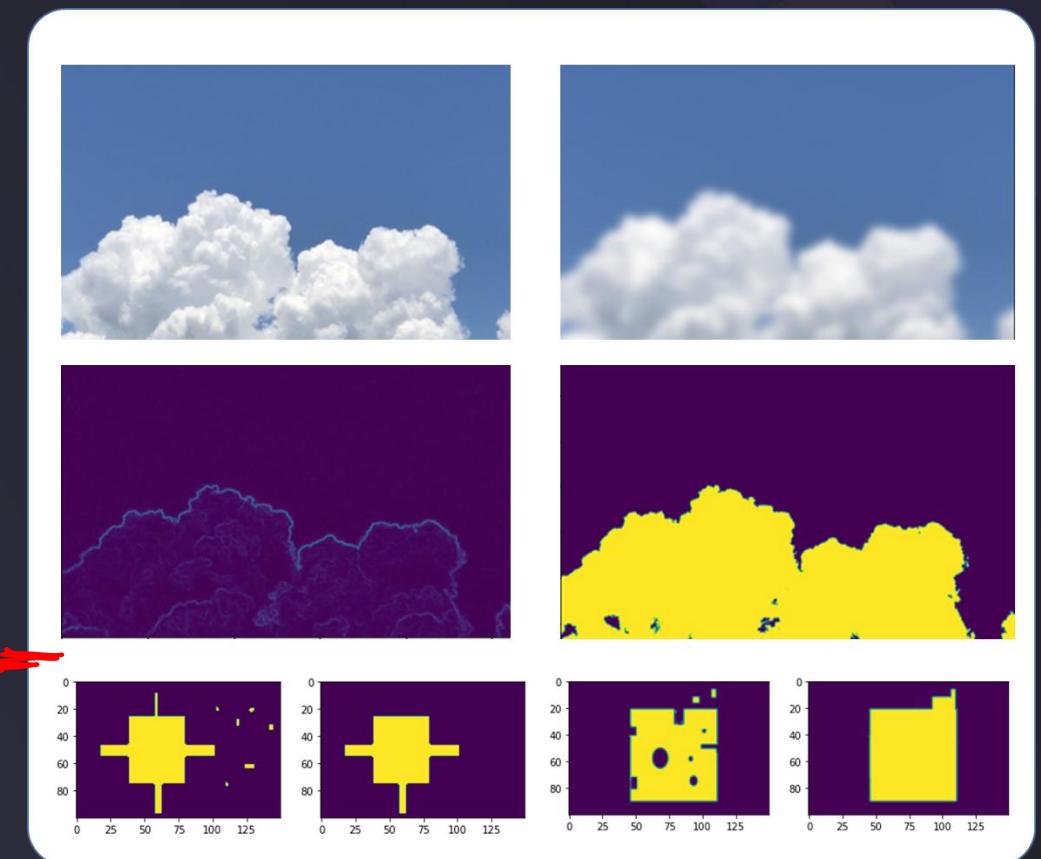
- 图像数据结构
- 图像颜色空间转化
- 基本图像处理



课程设计

第三章：进阶数值图像处理方法

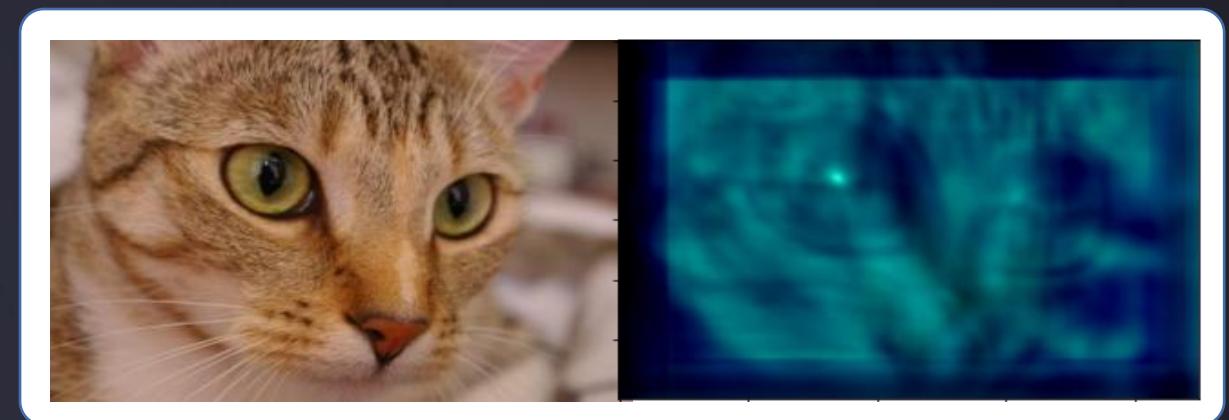
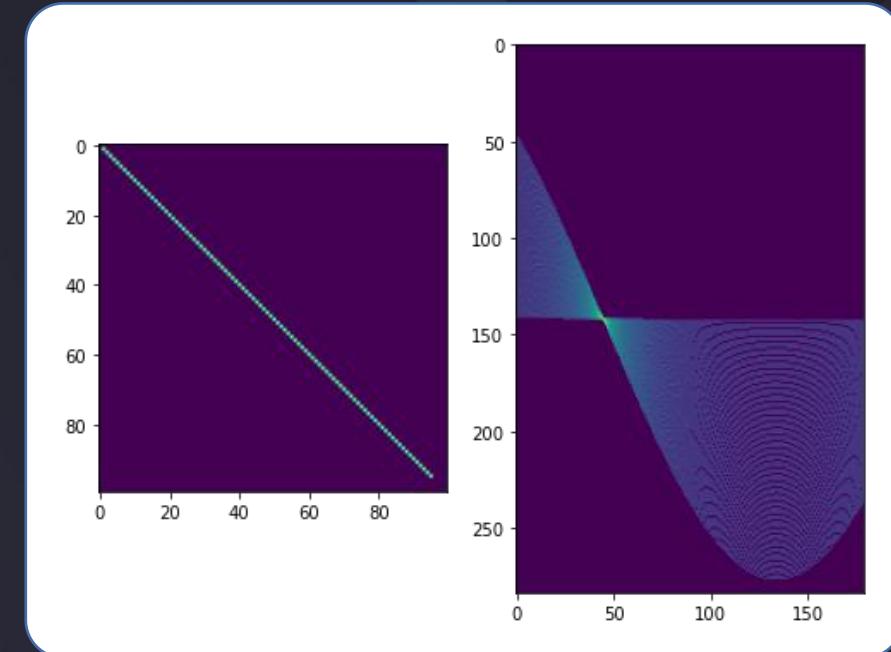
- 平滑滤波与边缘检测滤波
- 阈值图像分割
- 基本形态学处理



课程设计

第四章：经典图像处理算法

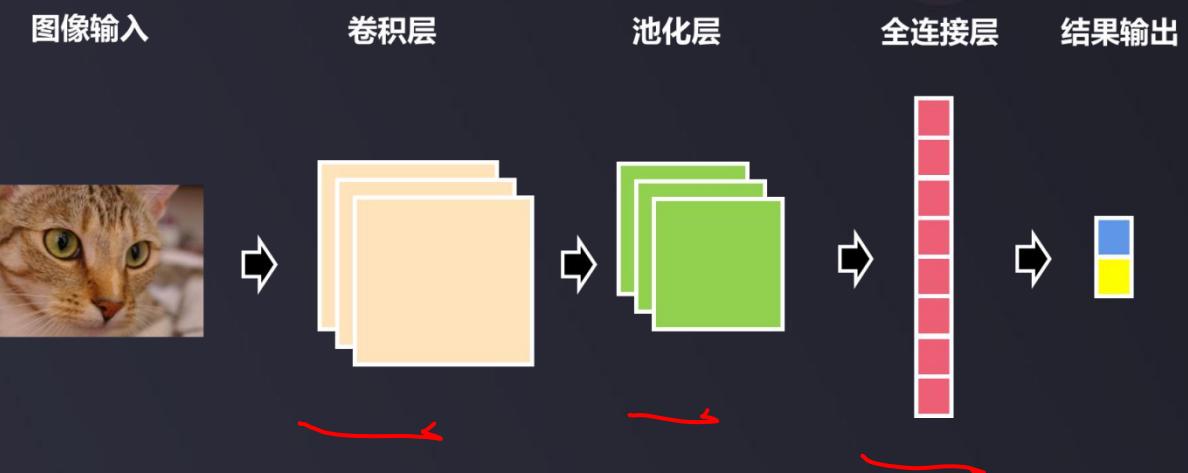
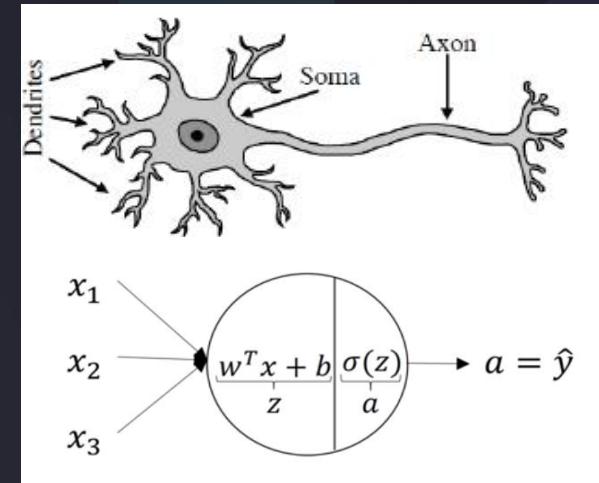
- 霍夫变换
- 模板匹配
- 经典算法的不足



课程设计

第五章：卷积神经网络的机理

- 神经元机理
- 反向传播算法
- 卷积神经网络机理
- 经典卷积神经网络架构



课程设计

第六章：卷积神经网络的优化

- 过拟合问题
- 梯度消失/爆炸问题
- 训练的优化
- 其他优化策略

		真实结果	
		正	负
预测结果	正	True Positive	False Positive
	负	False Negative	True Negative

验证	训练	训练	训练
训练	验证	训练	训练
训练	训练	验证	训练
训练	训练	训练	验证

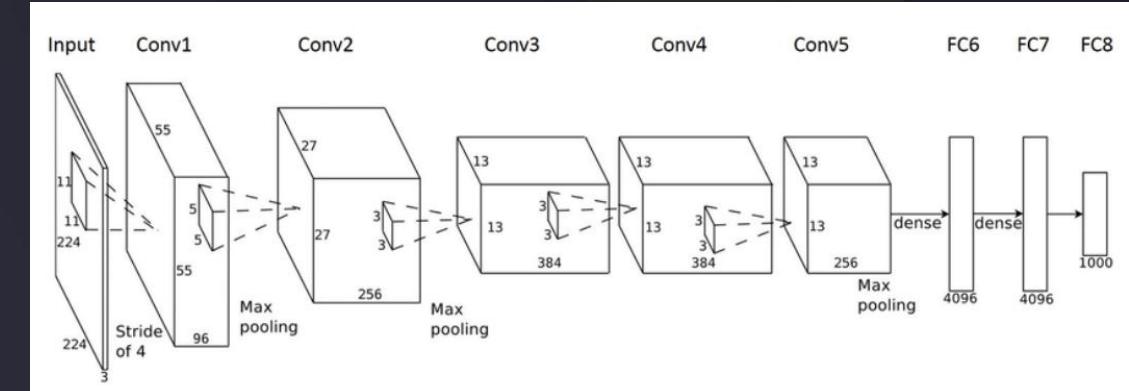
$$\begin{aligned}
 v_w &= \beta_1 v_w + (1 - \beta_1) dW \\
 v_b &= \beta_1 v_b + (1 - \beta_1) db \\
 S_w &= \beta_2 S_w + (1 - \beta_2) dW^2 \\
 S_b &= \beta_2 S_b + (1 - \beta_2) db^2 \\
 v_w &= v_w / (1 - \beta_1^t) \quad v_b = v_b / (1 - \beta_1^t) \\
 s_w &= s_w / (1 - \beta_2^t) \quad s_b = s_b / (1 - \beta_2^t) \\
 W &= W - \alpha \frac{V_w}{\sqrt{(S_w) + \varepsilon}} \quad b = b - \alpha \frac{V_b}{\sqrt{(S_b) + \varepsilon}}
 \end{aligned}$$

课程设计

第七章：卷积神经网络实战

- Tensorflow和Keras的使用
- 用于Minst分类的AlexNet

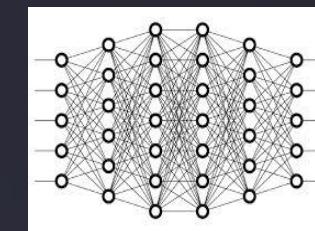
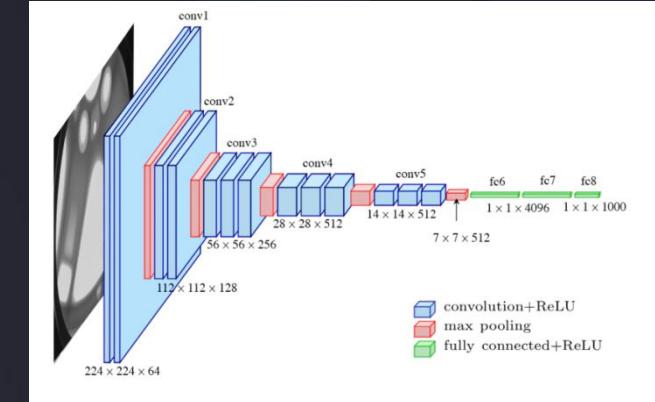
0 ~ 9



课程设计

第八章：迁移学习实战

- ImageNet
- VGG
- 迁移学习的三种方法



课程设计

第九章：多标签分类任务

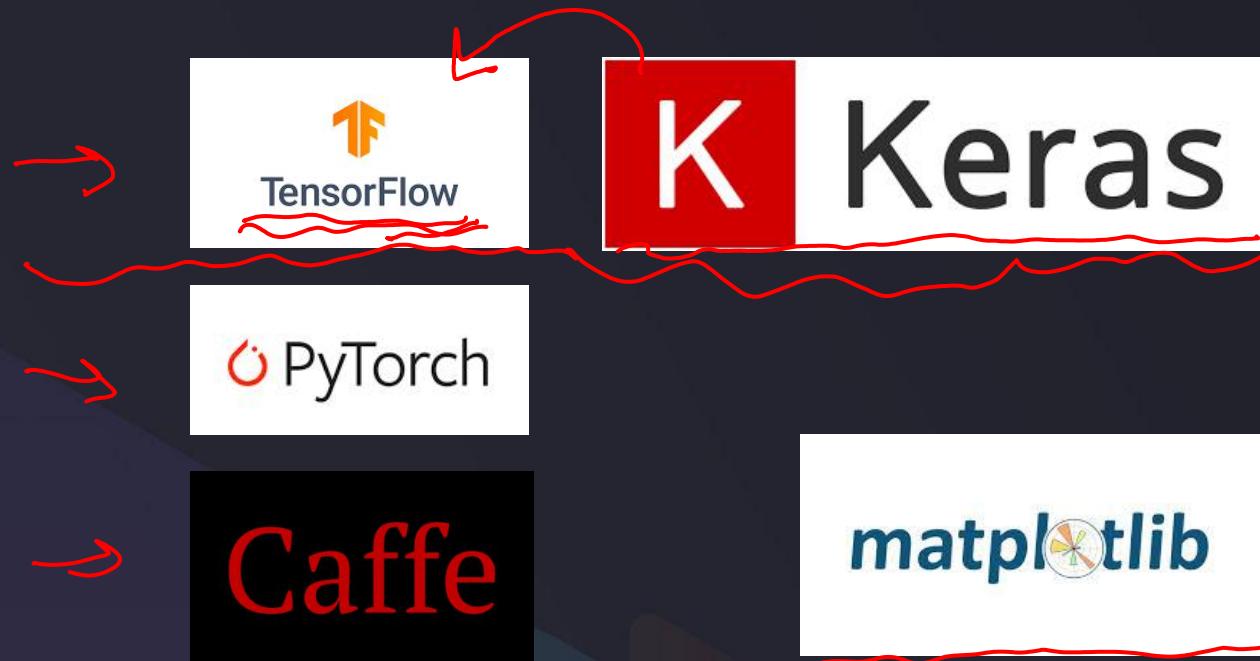
- 监督学习的难点
- 多目标分类任务
- 计算视觉算法工程应用



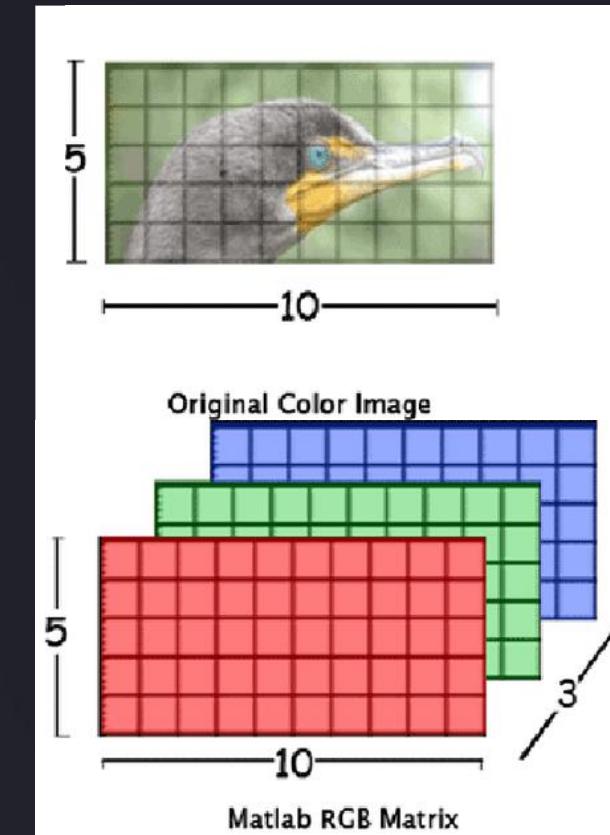
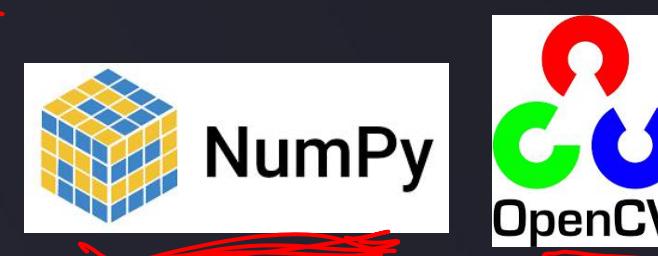
Content

- 1 --课程导学
- 2 --讲师介绍
- 3 --计算视觉与人工智能深度学习
- 4 --深度学习在计算视觉的前沿应用
- 5 --课程设计和体系
- 6 --算法的基本实现框架与工具

|深度学习计算视觉的常用工具



|下一章展望





深度学习实战与计算视觉1：数值图像 处理与卷积神经网络分类任务

第二章：基础工具，环境配置与图形单值化处理

Content

- 1 --开发工具简介
- 2 --基本环境配置
- 3 --数值图像处理的基本知识
- 4 -- (实战) 工具环境准备
- 5 -- (实战) 数值图像处理手把手实现



<https://www.anaconda.com/products/individual>

python

开发工具





NumPy

matplotlib



scikit-image
image processing in python

OpenCV

基本环境配置

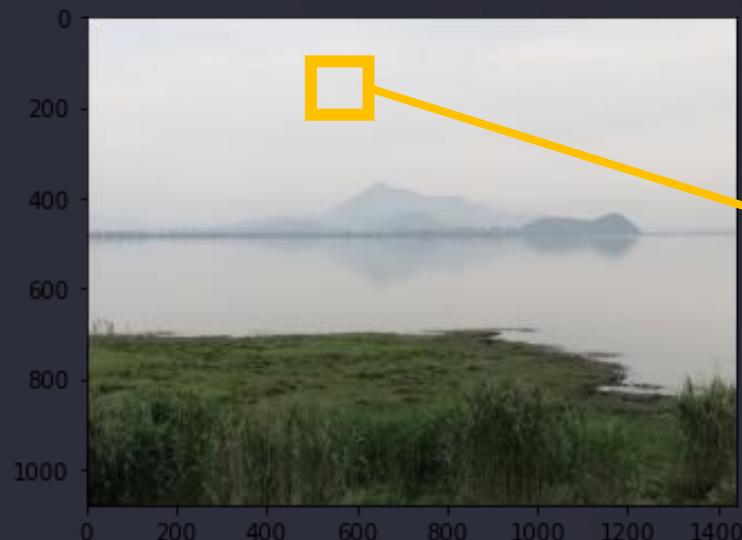
计算机视觉与数字图像处理

1. 计算机视觉 (Computer Vision) 是模仿人眼和大脑 “看图” 和 “理解” 的过程，关键词是 “真实” 和 “理解”；输入是图片，输出是模型/识别结果等从图象中提取的信息；例如背景分割，运动检测，物体识别，人脸识别
2. 数字图像处理 (Digital Image Processing) 是在看图前对图像进行的各种预处理工作，包括已有的图像进行变换、分析、重构、像素级的处理；输入是图像，输出也是图像；例如图片增强，去噪，滤镜等
3. 计算机图形学 (Computer Graphics) 类似于人 “画图”，是利用计算机进行图形的生成；输入是模型，输出是图像（像素），创造新的视觉感知；例如指纹生成、3D特效、游戏电影制作等

输入\输出	图像	知识
图像	数字图像处理	计算机视觉
知识	计算机图形学	智能过程

图像处理

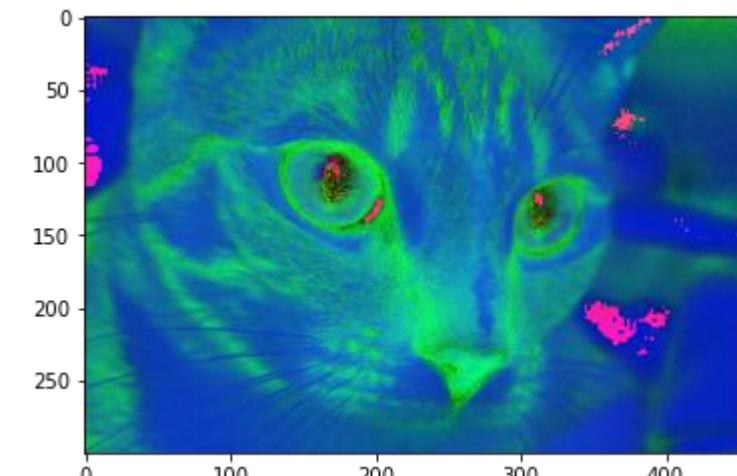
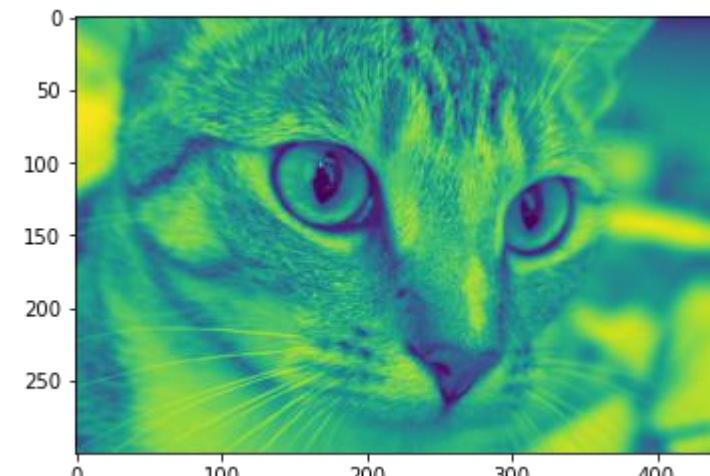
- 图像的读取，显示和储存
- 图像的数字矩阵格式
- RGB三通道



0	255	0	0	0			
0	255	0	0	0			
0	255	255	0	0			
0	0	255	255	0			
0	0	0	255	0			

| 图像处理

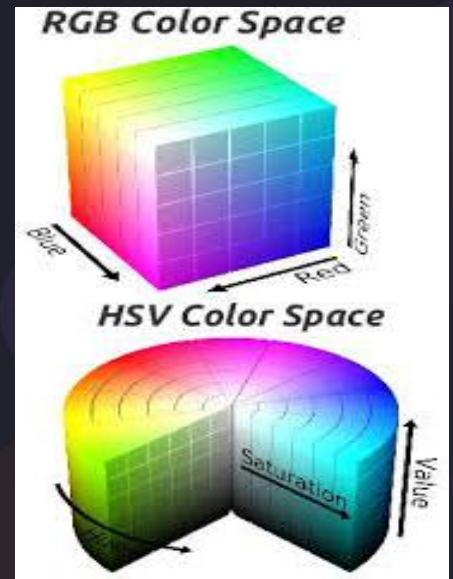
- 颜色空间的转化 (灰度, HSV)



HSV色彩空间

类似于人类感觉颜色的方式，具有较强的感知度

- 色相 (H) : 是色彩的基本属性，就是平常所说颜色名称，如红色、黄色等。
- 饱和度 (S) : 是指色彩的纯度，越高色彩越纯，低则逐渐变灰，取0-100%的数值。
- 明度 (V) : 又称为亮度 (L) , 取0-100%



RGB to HSV conversion formula

The R, G, B values are divided by 255 to change the range from 0..255 to 0..1:

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

$$C_{max} = \max(R', G', B')$$

$$C_{min} = \min(R', G', B')$$

$$\Delta = C_{max} - C_{min}$$

Hue calculation:

$$H = \begin{cases} 0^\circ & , \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta} \text{ mod } 6 \right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right) & , C_{max} = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

Value calculation:

$$V = C_{max}$$

HSV to RGB conversion formula

When $0 \leq H < 360$, $0 \leq S \leq 1$ and $0 \leq V \leq 1$:

$$C = V \times S$$

$$X = C \times (1 - |(H / 60^\circ) \text{ mod } 2 - 1|)$$

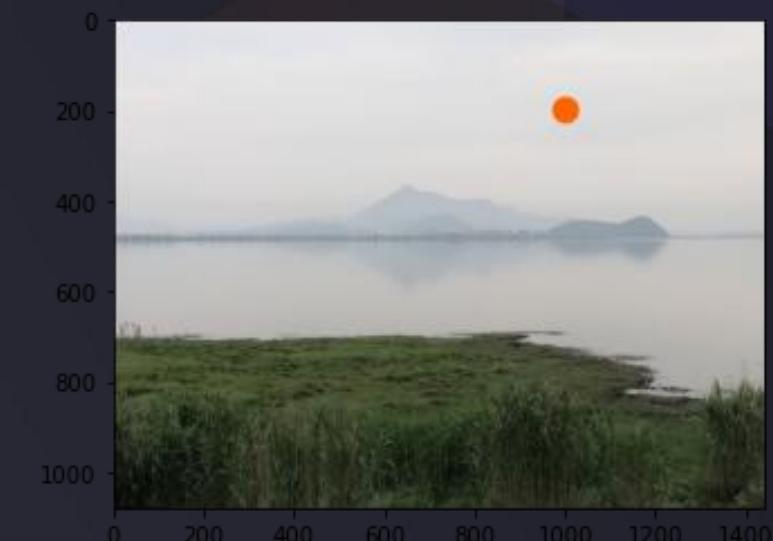
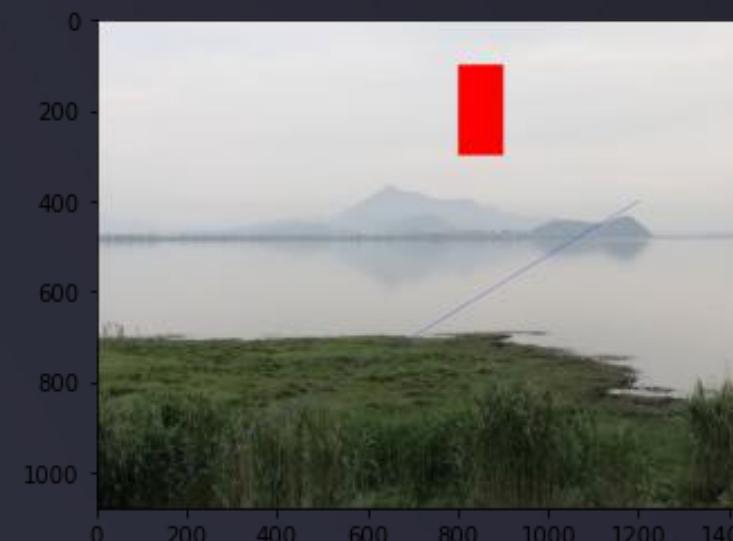
$$m = V - C$$

$$(R', G', B') = \begin{cases} (C, X, 0) & , 0^\circ \leq H < 60^\circ \\ (X, C, 0) & , 60^\circ \leq H < 120^\circ \\ (0, C, X) & , 120^\circ \leq H < 180^\circ \\ (0, X, C) & , 180^\circ \leq H < 240^\circ \\ (X, 0, C) & , 240^\circ \leq H < 300^\circ \\ (C, 0, X) & , 300^\circ \leq H < 360^\circ \end{cases}$$

$$(R, G, B) = ((R' + m) \times 255, (G' + m) \times 255, (B' + m) \times 255)$$

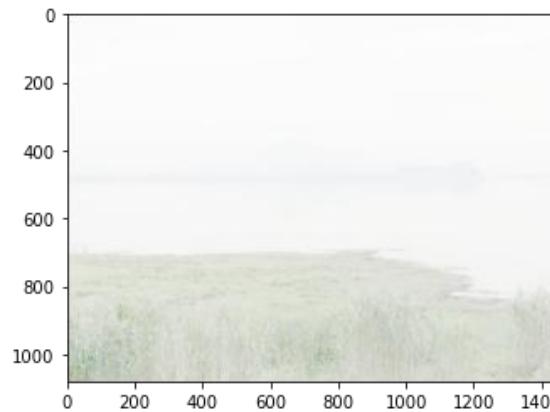
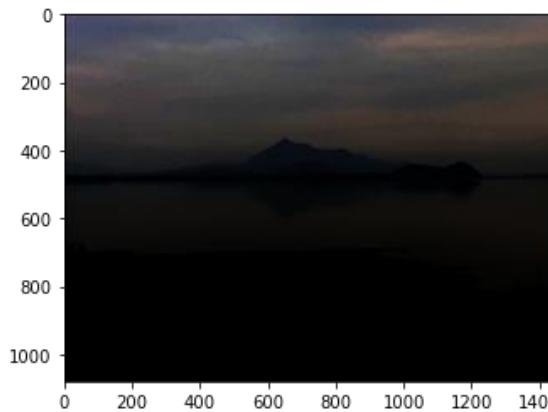
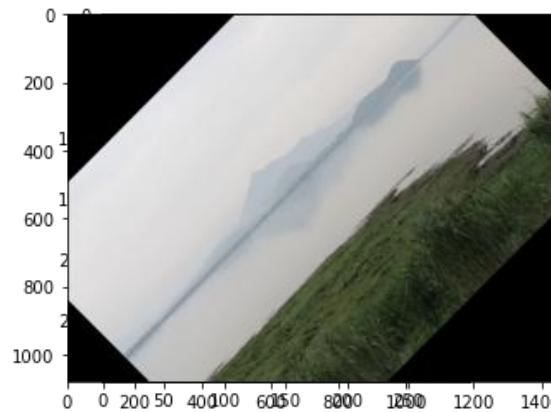
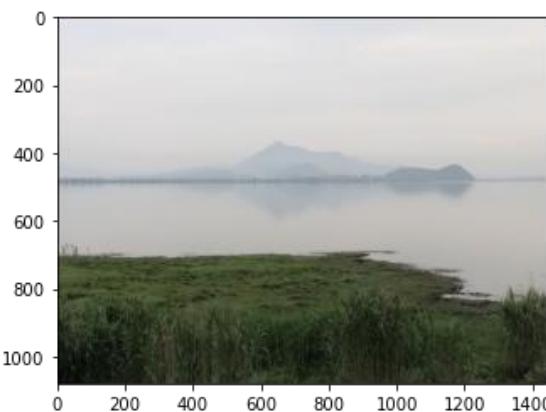
| 图像处理

- 基本图形的绘制



| 图像处理

- 图像形变，缩放，旋转和亮度变化





深度学习实战与计算视觉1：数值图像 处理与卷积神经网络分类任务

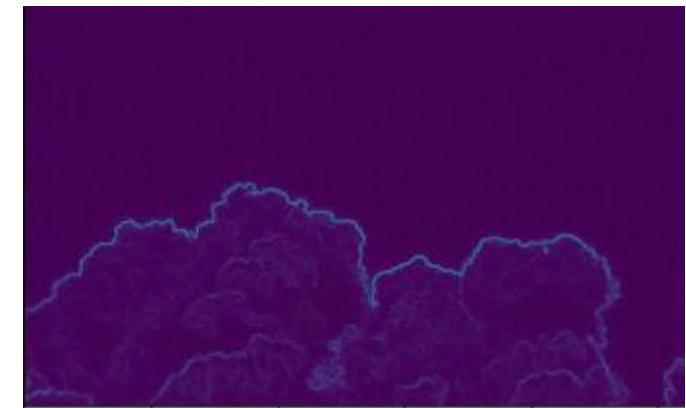
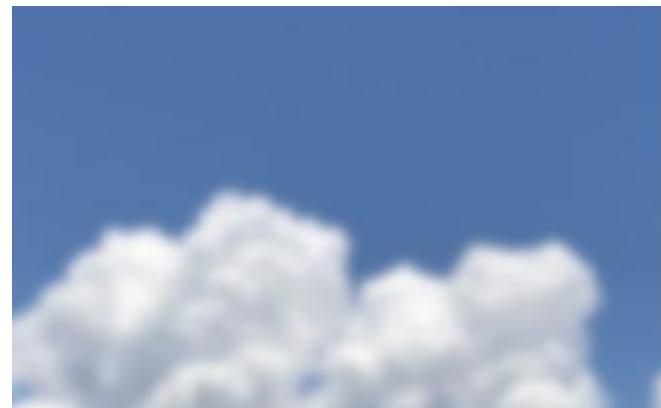
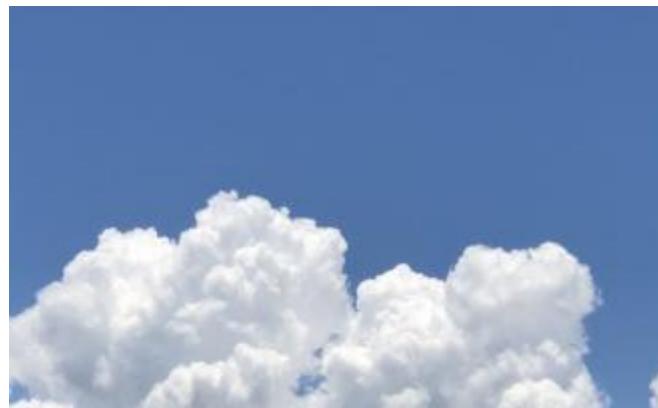
第三章：图像数值化处理进阶方法

Content

- 1 --图像滤波 (平滑滤波)
- 2 --图像滤波 (边缘检测)
- 3 --图像阈值分割
- 4 --基本形态学滤波：膨胀，腐蚀，开关运算
- 5 -- (实战) 进阶数值图像处理方法手把手实现

| 图像滤波（平滑滤波和边缘检测）

- 消除图像中混入的噪声；
- 为图像识别抽取出图像特征。



| 图像滤波：实现方式

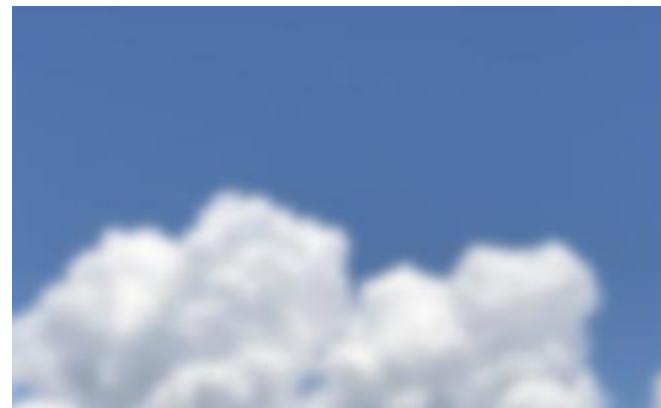
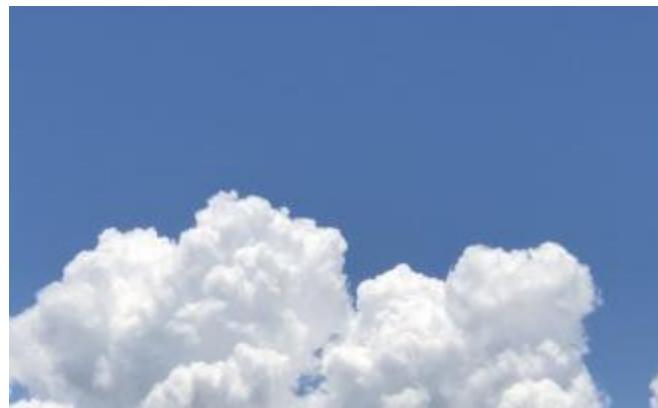


90	0	90	0	90	0
0	90	0	90	0	90
90	0	90	0	90	0
0	90	0	90	0	90
90	0	90	0	90	0
0	90	0	90	0	90
0	90	0	90	0	90

?	?	?
?	?	?
?	?	?

| 图像滤波：平滑滤波

- 简单平均法
- 高斯滤波



| 图像滤波：平滑滤波

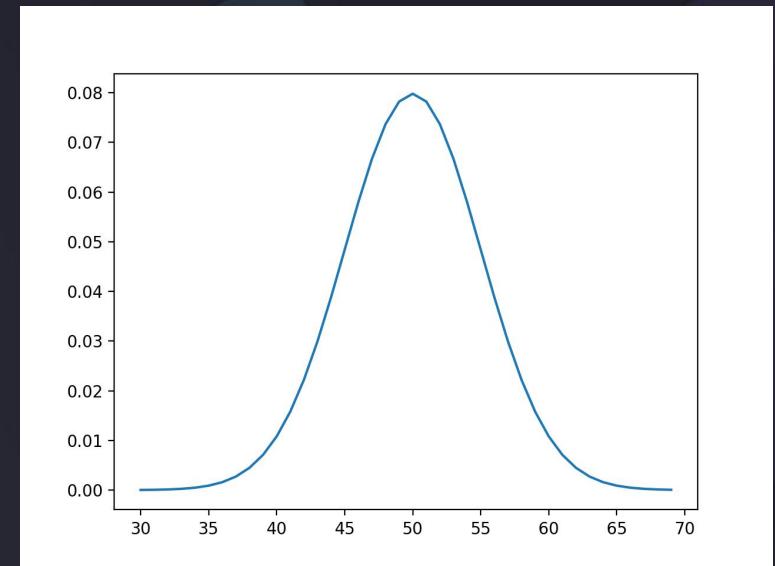
- 简单平均法

90	0	90	0
0	90	0	90
90	0	90	0
0	90	0	90

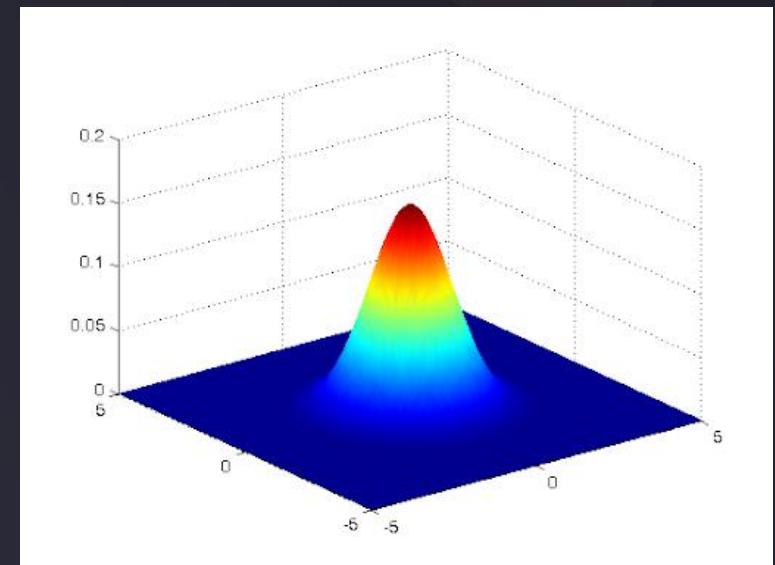
| 图像滤波：平滑滤波

- 高斯滤波

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$



$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



| 图像滤波：平滑滤波

- 高斯滤波

100	0	0	0
0	100	0	0
0	0	100	0
0	0	0	100

Sigma=1.5

0.0454	0.0566	0.0454
0.0566	0.0707	0.0566
0.0454	0.0566	0.0454

N/A	N/A	N/A	N/A
N/A	33.73	N/A	N/A
N/A	N/A	N/A	N/A
N/A	N/A	N/A	N/A

| 图像滤波：高斯平滑代码实现

```
Skimage.filters.gaussian(image, sigma=1,  
output=None, mode='nearest', cval=0,  
multichannel=None, preserve_range=False,  
truncate=4.0)
```

```
Import skimage.filters  
img_e = filters.gaussian(img,sigma=10)
```

Content

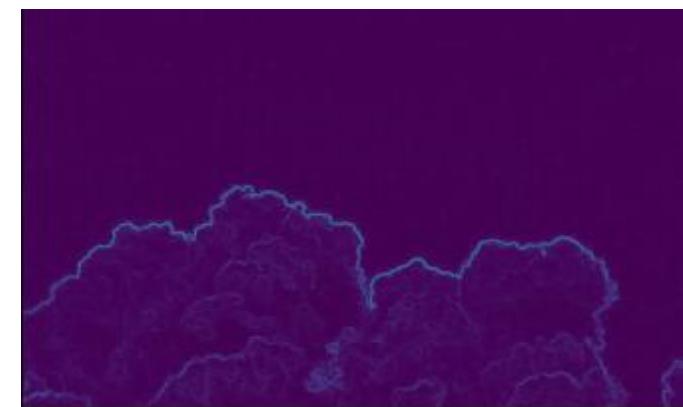
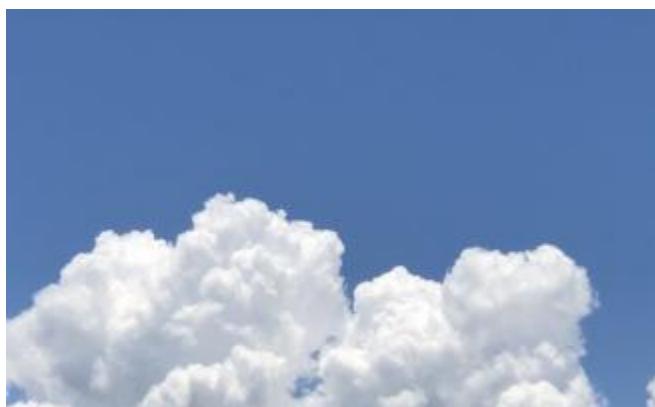
- 1 --图像滤波 (平滑滤波)
- 2 --图像滤波 (边缘检测)
- 3 --图像阈值分割
- 4 --基本形态学滤波：膨胀，腐蚀，开关运算
- 5 -- (实战) 进阶数值图像处理方法手把手实现

Content

- 1 --图像滤波 (平滑滤波)
- 2 --图像滤波 (边缘检测)
- 3 --图像阈值分割
- 4 --基本形态学滤波：膨胀，腐蚀，开关运算
- 5 -- (实战) 进阶数值图像处理方法手把手实现

| 图像滤波：边缘检测

- Roberts算子
- Prewitt算子
- Sobel算子



| 图像滤波：边缘检测

- Roberts算子

0	0	0
0	100	100
0	100	100

-1	0
0	1

0	-1
1	0

图像滤波：边缘检测

- Prewitt算子

-1	-1	-1
0	0	0
1	1	1

水平

-1	0	1
-1	0	1
-1	0	1

垂直

0	1	1
-1	0	1
-1	-1	0

-1	-1	0
-1	0	1
0	1	1

对角线

- Sobel算子

-1	-2	-1
0	0	0
1	2	1

水平

-1	0	1
-2	0	2
-1	0	1

垂直

0	1	2
-1	0	1
-2	-1	0

-2	-1	0
-1	0	1
0	1	2

对角线

| 图像滤波：边缘检测

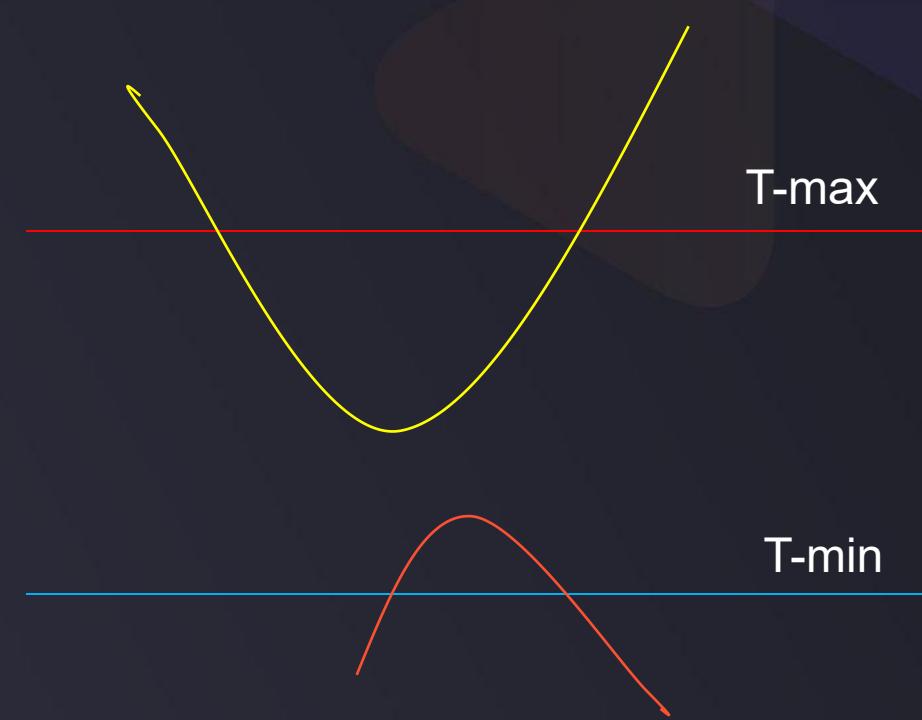
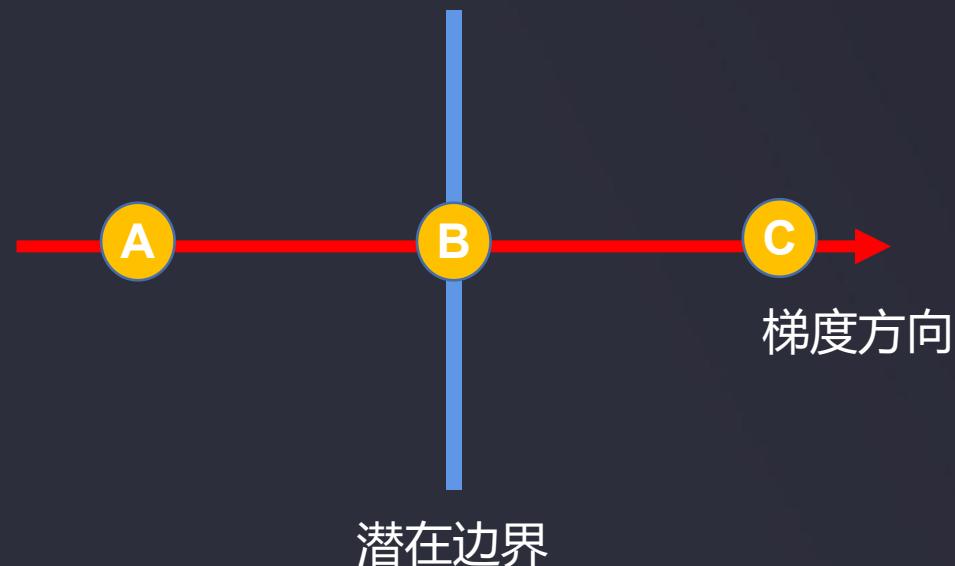
- Canny算子

1. 用高斯滤波平滑图像
2. 用sobel等梯度算子计算梯度幅值和方向
3. 对梯度幅值进行非极大值抑制
4. 用双阈值算法检测和连接边缘



| 图像滤波：边缘检测

- Canny算子
 - 3. 对梯度幅值进行非极大值抑制
 - 4. 用双阈值算法检测和连接边缘



| 图像滤波：边缘检测代码的实现

```
from skimage import filters  
from skimage import feature
```

```
filters.roberts(img)  
filters.prewitt(img)  
• filters.prewitt_h(img)  
• filters.prewitt_v(img)  
filters.sobel(img)  
• filters.sobel_h(img)  
• filters.sobel_v(img)
```

```
feature.canny(img,sigma=0.1)
```

| 不同算子效果对比

算子	优缺点
Robert	擅长低噪音和梯度较大的图像，但是边缘提取比较粗，定位不是很准
Prewitt	对于灰度渐变和噪音较多的图像处理较好
Sobel	对于灰度渐变和噪音较多的图像处理较好
Laplacian	对于阶跃性边缘点比较好，但是对噪音敏感，容易产生不连续的检测边缘
Log	善于判断边缘在图像明区还是暗区，对噪音比较敏感
Canny	抗噪音好，对于弱边缘检测较好

Content

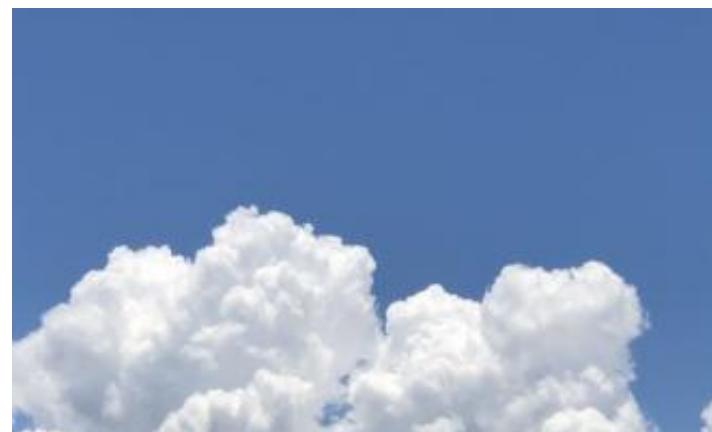
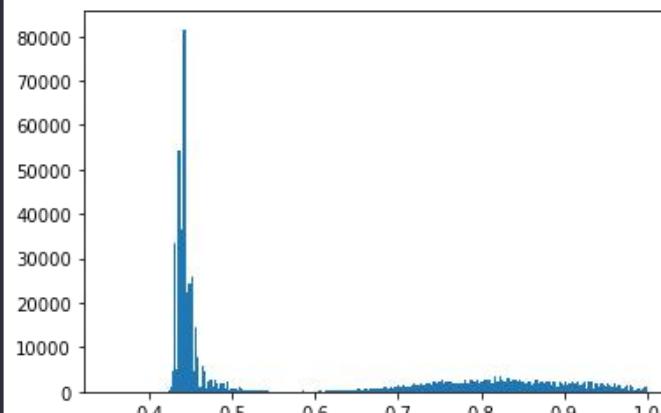
- 1 --图像滤波 (平滑滤波)
- 2 --图像滤波 (边缘检测)
- 3 --图像阈值分割
- 4 --基本形态学滤波：膨胀，腐蚀，开关运算
- 5 -- (实战) 进阶数值图像处理方法手把手实现

Content

- 1 --图像滤波 (平滑滤波)
- 2 --图像滤波 (边缘检测)
- 3 --图像阈值分割
- 4 --基本形态学滤波：膨胀，腐蚀，开关运算
- 5 -- (实战) 进阶数值图像处理方法手把手实现

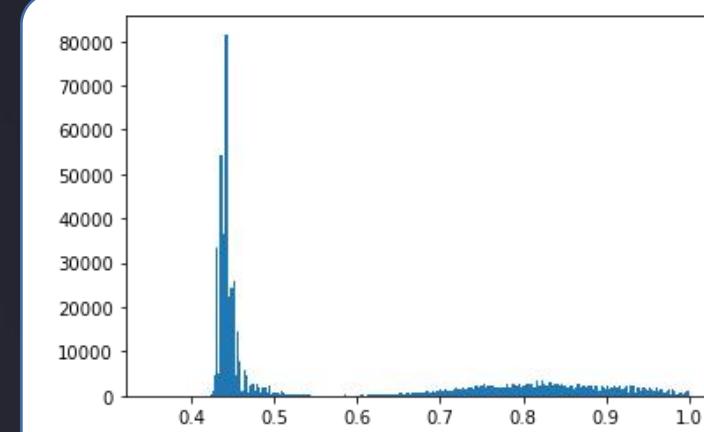
图片分割

- 基于Otsu算法（大津算法）的阈值分割方法



图片分割

- 把图像转成灰度图
- 计算全部平均灰度 w
- 选定一个阈值 T 把所有像素分成 N_0, N_1 两个部分
- 计算 N_0 的灰度 w_0 , 和 N_1 的灰度 w_1
- 计算类间方差 $g = N_0 * (w_0 - w)^2 + N_1 * (w_1 - w)^2$
- 采用遍历法找到 T_{max} 使得 g 最大



$$g = N_0 N_1 (w_0 - w_1)^2$$

| 图像分割-OTSU代码实现

```
from skimage import feature
```

```
theta = filters.threshold_otsu(img_gray,nbins=256)
```

```
img_seg=np.zeros(img_gray.shape)
```

```
img_seg[img_gray>theta]=1
```

Content

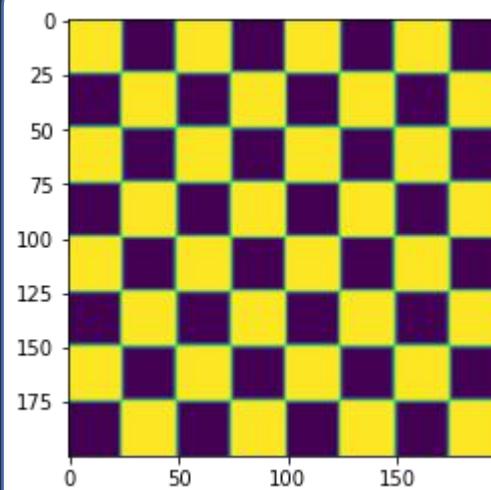
- 1 --图像滤波 (平滑滤波)
- 2 --图像滤波 (边缘检测)
- 3 --图像阈值分割
- 4 --基本形态学滤波：膨胀，腐蚀，开关运算
- 5 -- (实战) 进阶数值图像处理方法手把手实现

Content

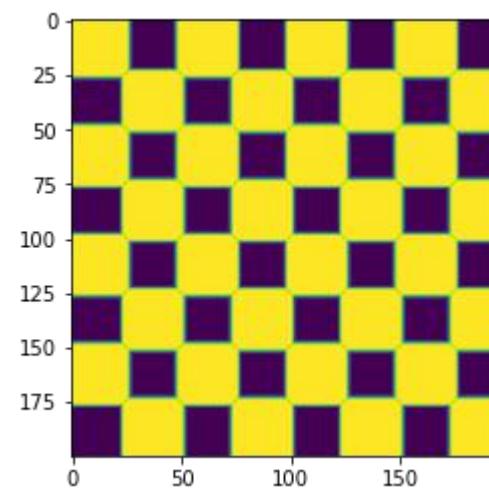
- 1 --图像滤波 (平滑滤波)
- 2 --图像滤波 (边缘检测)
- 3 --图像阈值分割
- 4 --基本形态学滤波：膨胀，腐蚀，开关运算
- 5 -- (实战) 进阶数值图像处理方法手把手实现

|基本形态学滤波-膨胀与腐蚀

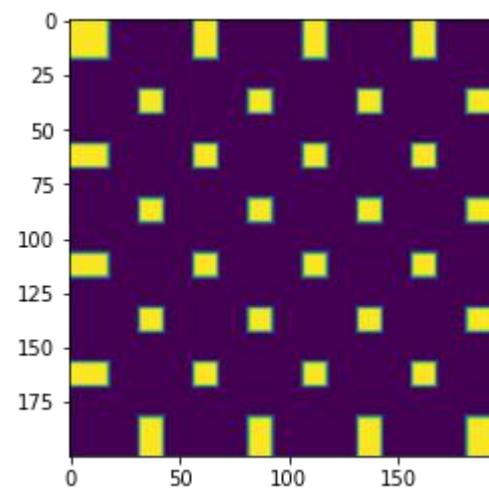
原图



膨胀

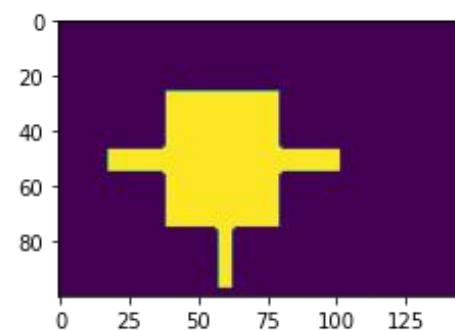
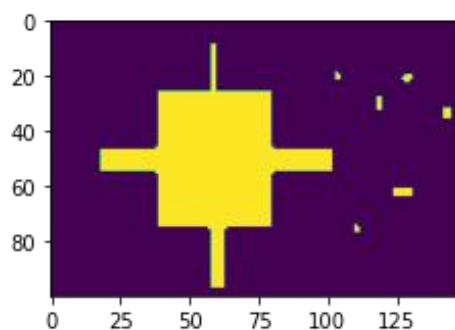


腐蚀

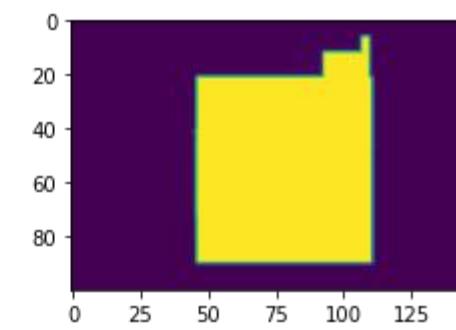
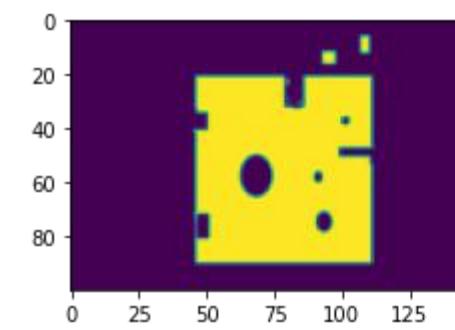


|基本形态学滤波-开运算与闭运算

开运算，先腐蚀再膨胀



闭运算，先膨胀再腐蚀



|基本形态学滤波-代码实现

```
from skimage import morphology  
  
img_d=morphology.dilation(img,morphology.square(2))  
  
img_d=morphology.erosion(img,morphology.square(2))  
  
img_d=morphology.opening(img_gray,morphology.square(5))  
  
img_d=morphology.closing(img_gray,morphology.square(15))
```

|本期内容与深度学习/计算视觉的关联

- 边缘检测—深度学习神经网络卷积层的数学原理
- 阈值图片分割—计算视觉分割任务的经典应用
- 平滑滤波与基本形态学滤波—图像的降噪与数据清洗

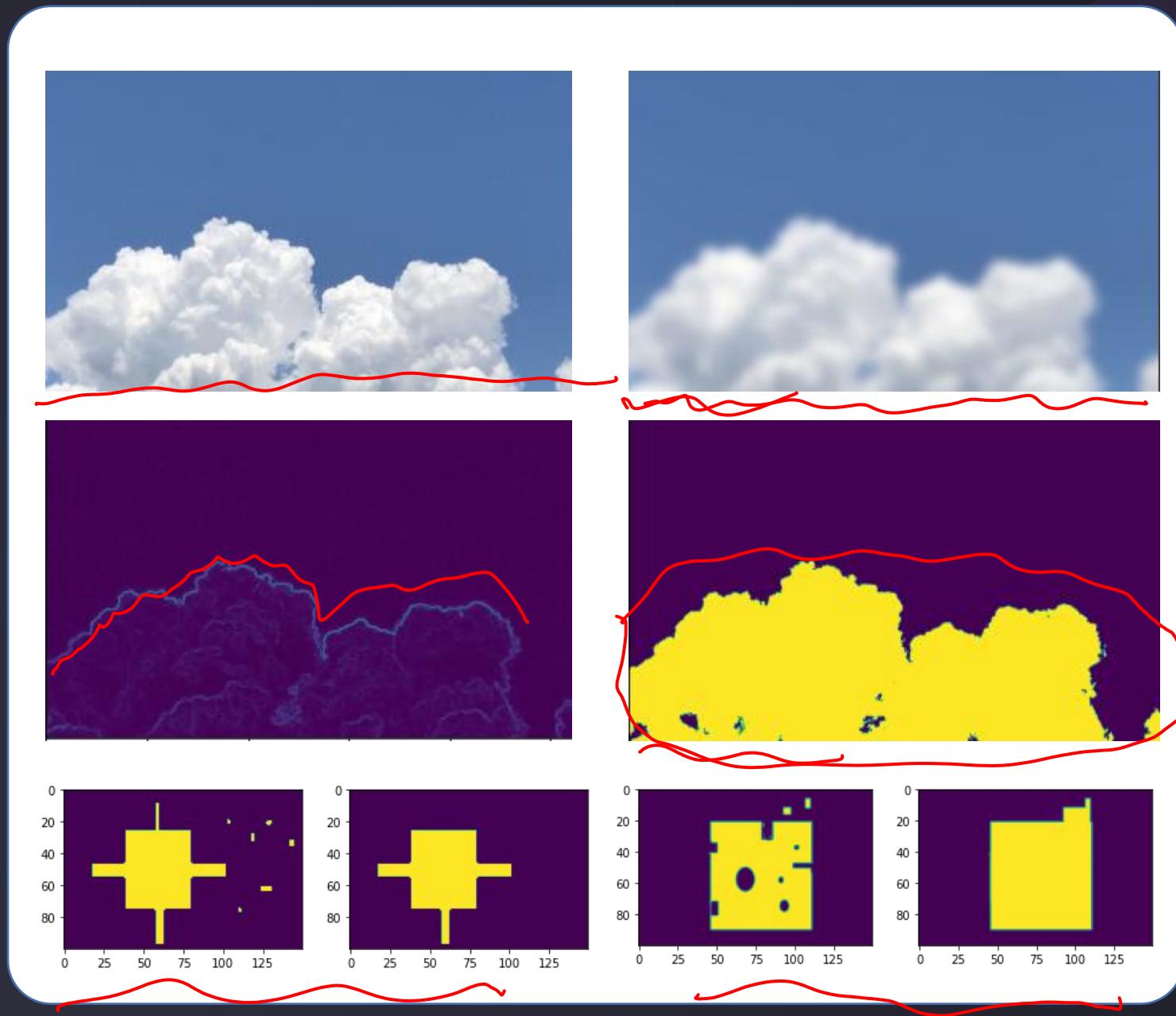


深度学习实战与计算视觉1：数值图像 处理与卷积神经网络分类任务

第四章：经典计算视觉方法

|上一章回顾

- 平滑滤波
- 边缘检测滤波
- 阈值图像分割
- 基本形态学滤波



Content

- 1 --霍夫变换
- 2 --模板匹配
- 3 --经典算法的不足

Content

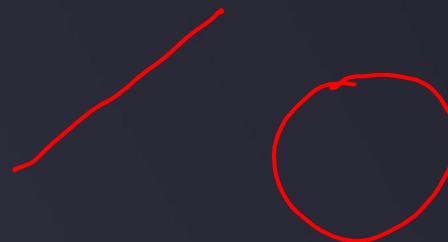
1 --霍夫变换

2 --模板匹配

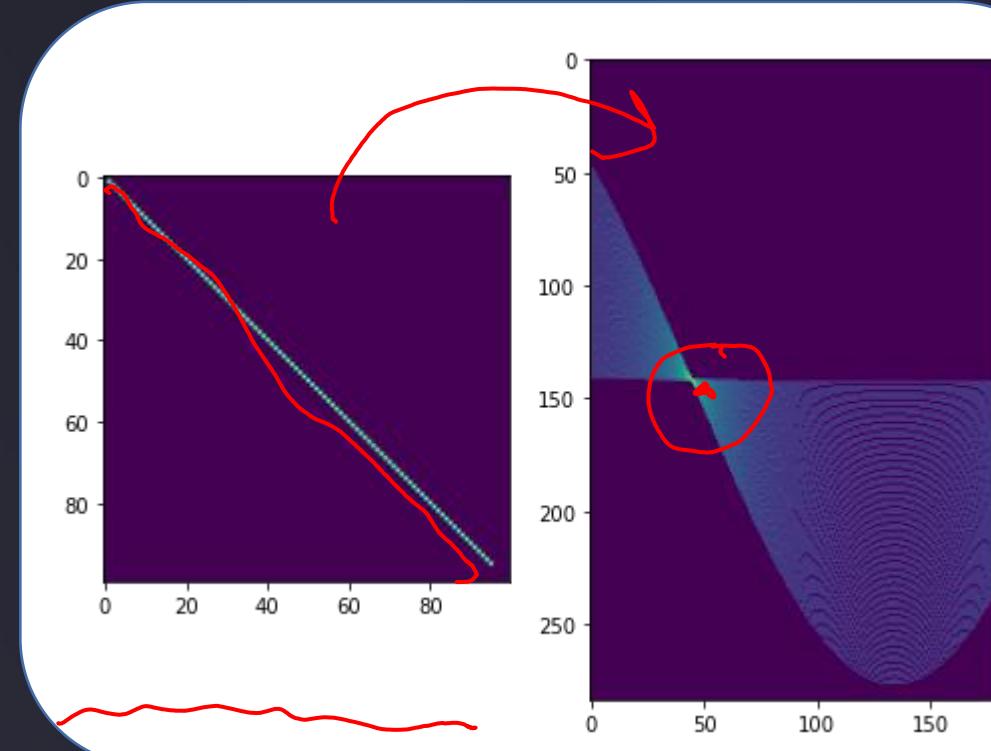
3 --经典算法的不足



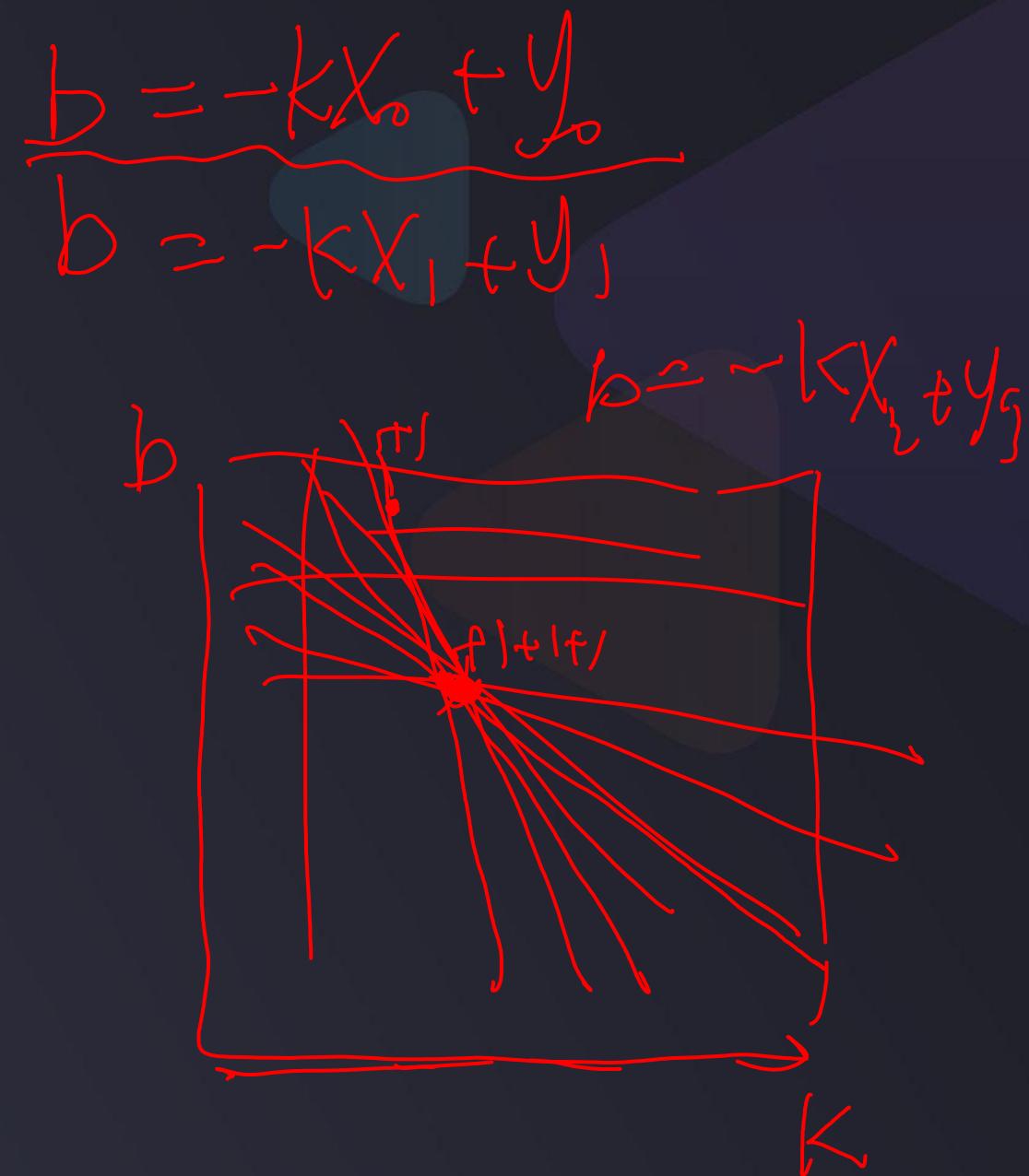
|霍夫变换



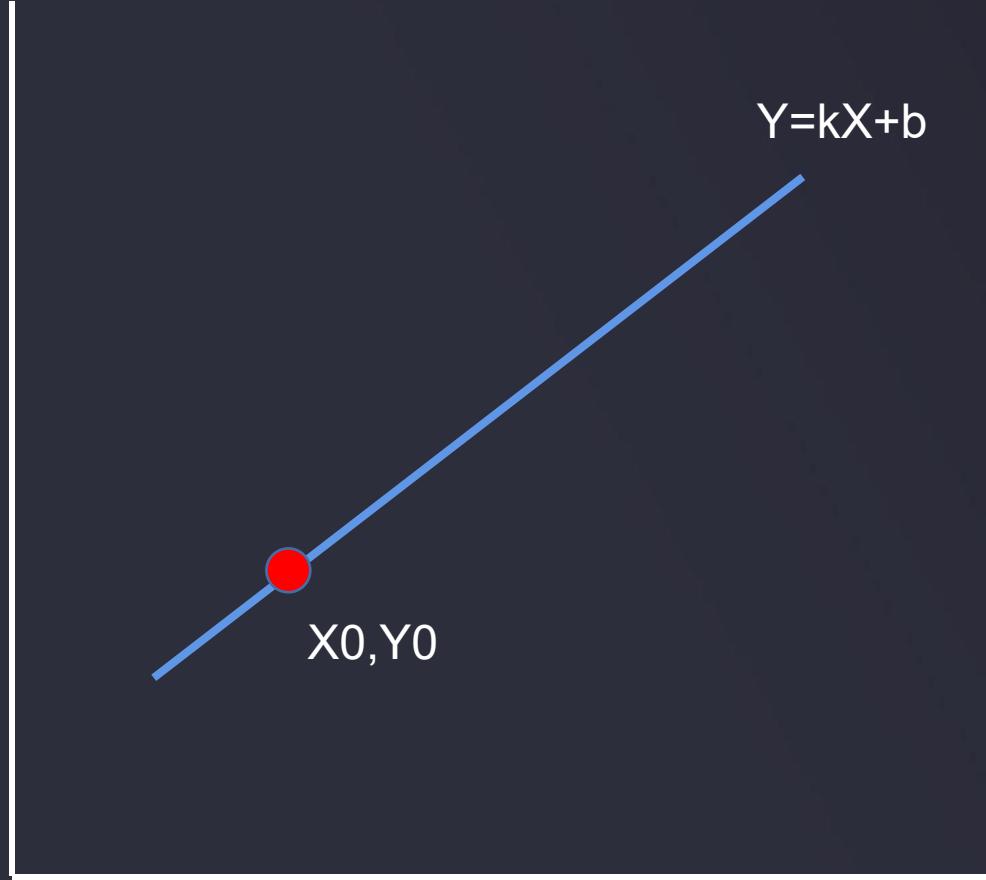
1. 选定要识别的形状的种类
2. 将直角坐标系的参数空间投影到特殊的参数空间
3. 寻找交点确定识别到的形状 (通过在参数空间中累加的局部最大值(local maximum)来决定。)



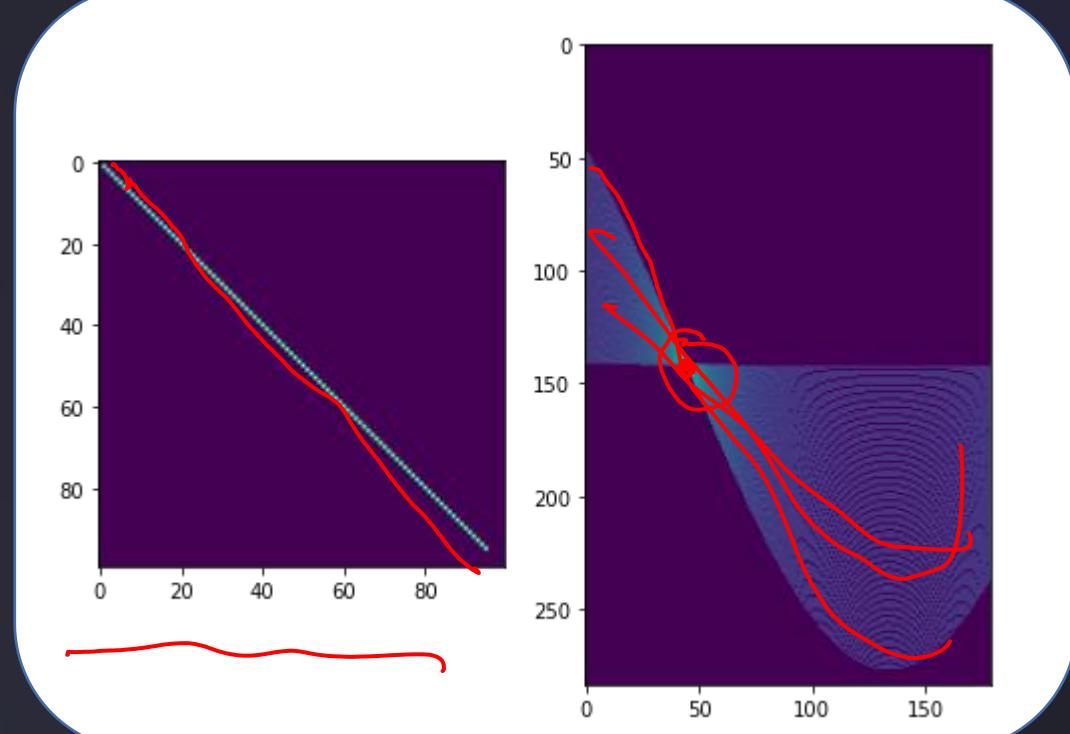
|霍夫变换-线检测



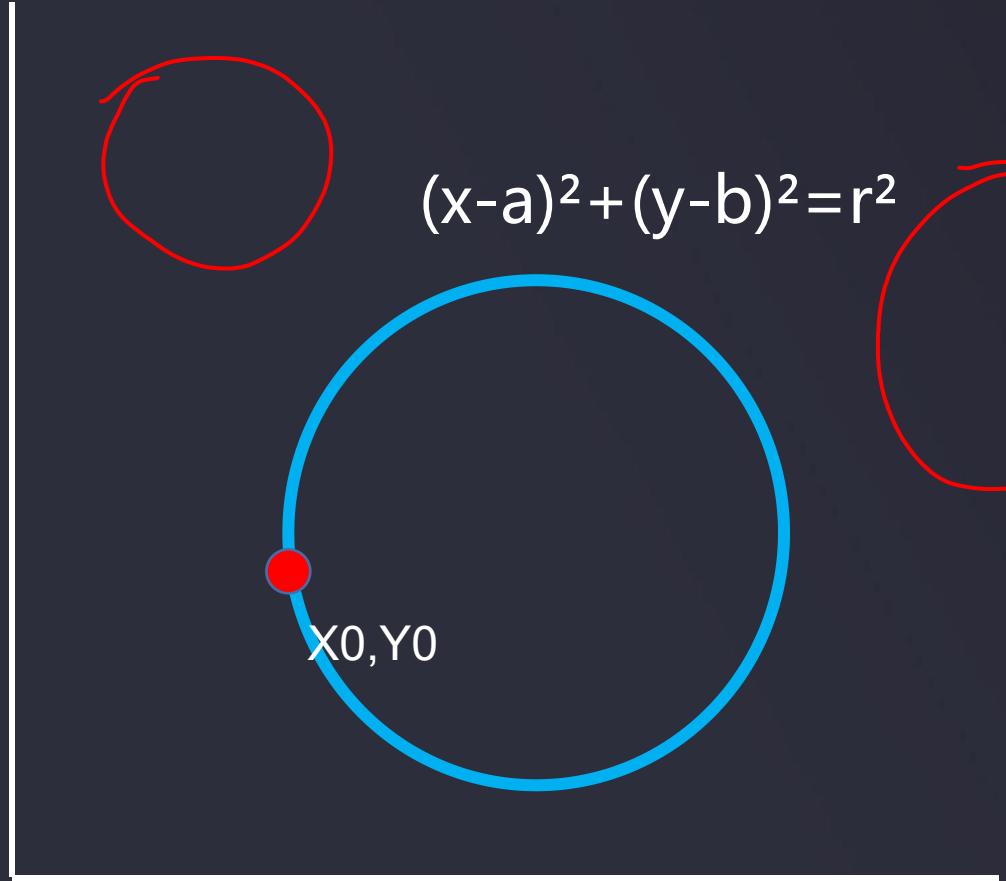
|霍夫变换-线检测：极坐标系 ρ - θ



$$y = \left(-\frac{\cos\theta}{\sin\theta} \right) x + \left(\frac{\rho}{\sin\theta} \right)$$
$$\rho = x_0 \cos\theta + y_0 \sin\theta$$



霍夫变换-圆检测



$$(a-x)^2 + (b-y)^2 = r^2$$

for r in range [5, 100]:
 | | |

5 7 9 ... 199

霍夫变换：代码实现

```
import skimage.transform as transform  
transform.hough_line(img)  
transform.hough_circle(img, radii)      5 100  
  
import skimage.measure as measure  
contours=measure.find_contours(img, 0.5)
```

Content

1 --霍夫变换

2 --模板匹配

3 --经典算法的不足

Content

1 --霍夫变换

2 --模板匹配

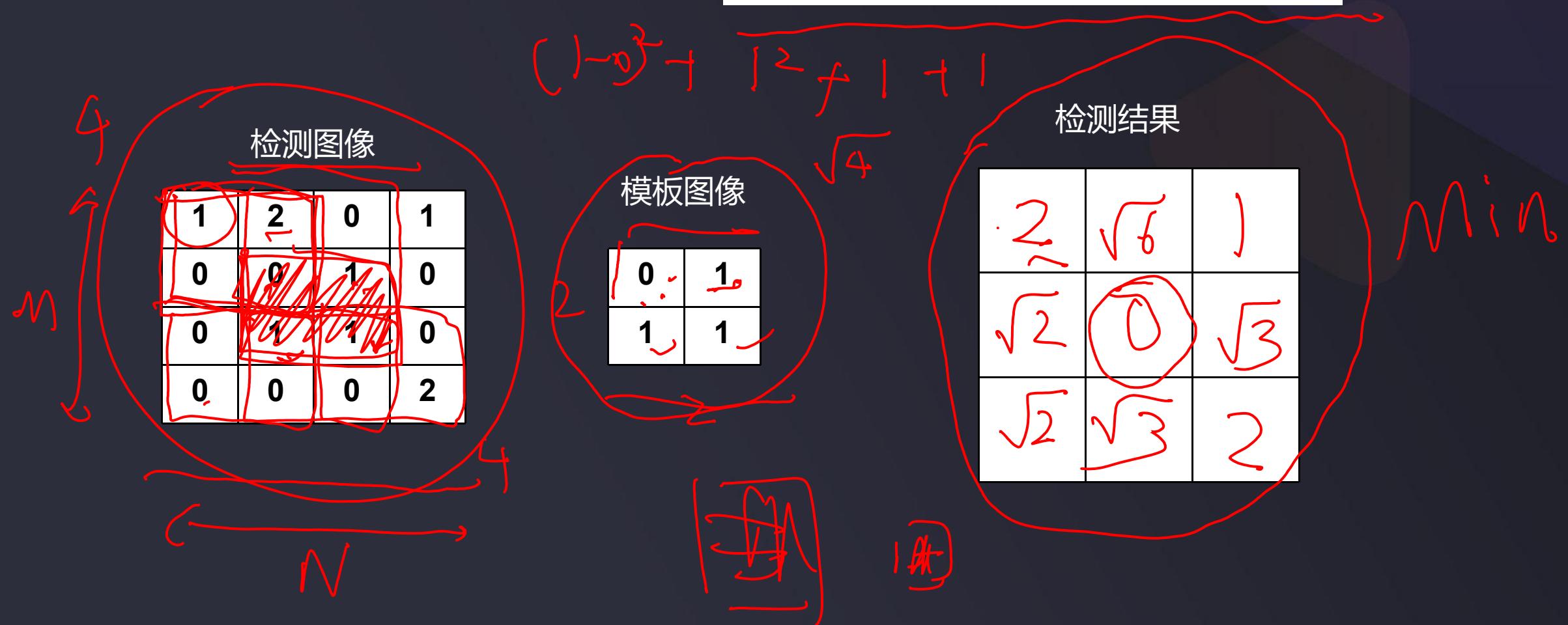
3 --经典算法的不足

模板匹配：原理

RMS_E

|MAE|

$$D(i, j) = \sum_{m=1}^M \sum_{n=1}^N [S_{ij}(m, n) - T(m, n)]^2$$



| 模板匹配：原理



| 模板匹配：代码实现

```
import skimage.feature as feature  
res=feature.match_template(img, template,  
pad_input=True, mode='constant',  
constant_values=0)  
index=np.where(res==np.max(res))
```

x, y
row col

res



Content

1 --霍夫变换

2 --模板匹配

3 --经典算法的不足

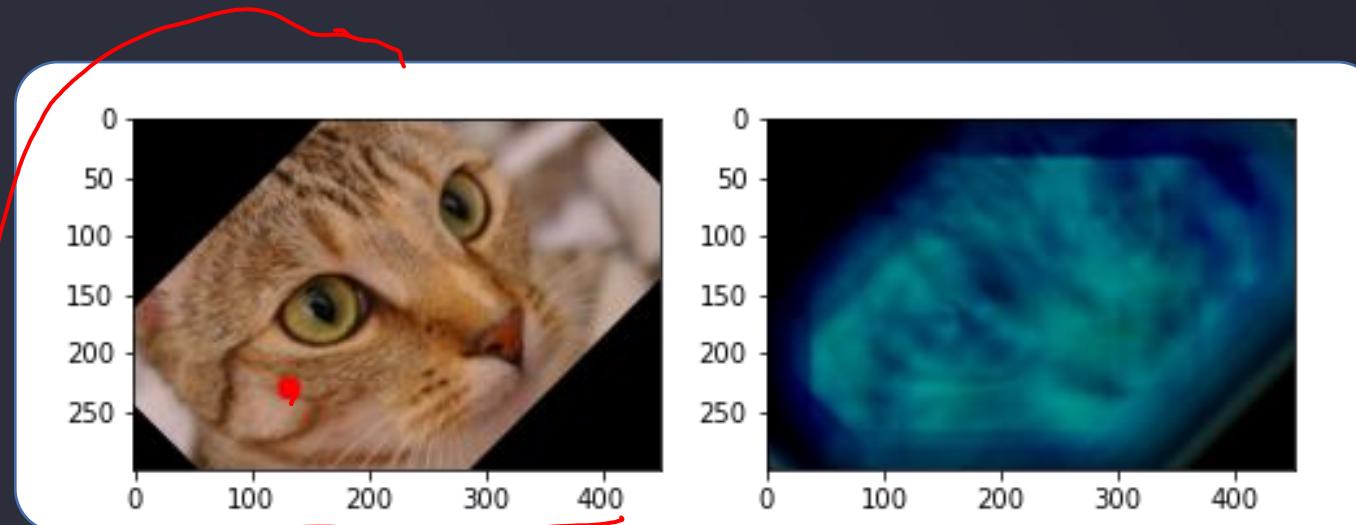
Content

1 --霍夫变换

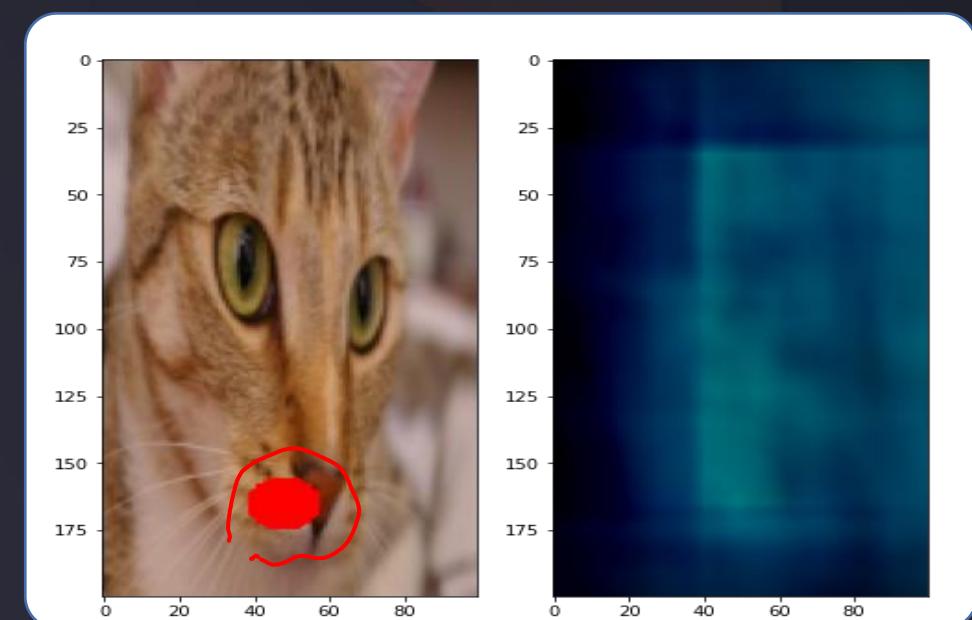
2 --模板匹配

3 --经典算法的不足

| 模板匹配：经典算法的一些不足



45°



| 模板匹配：经典算法的不足

- 旋转
- 大小
- 形变
- 遮挡
- 亮度
- 其他泛化问题

|本期内容回顾

- 高阶特征的识别和提取-霍夫变换
- 模板匹配—计算视觉识别任务的经典应用
- 经典计算视觉算法的不足—模板匹配抗旋转和形变等干扰的不足

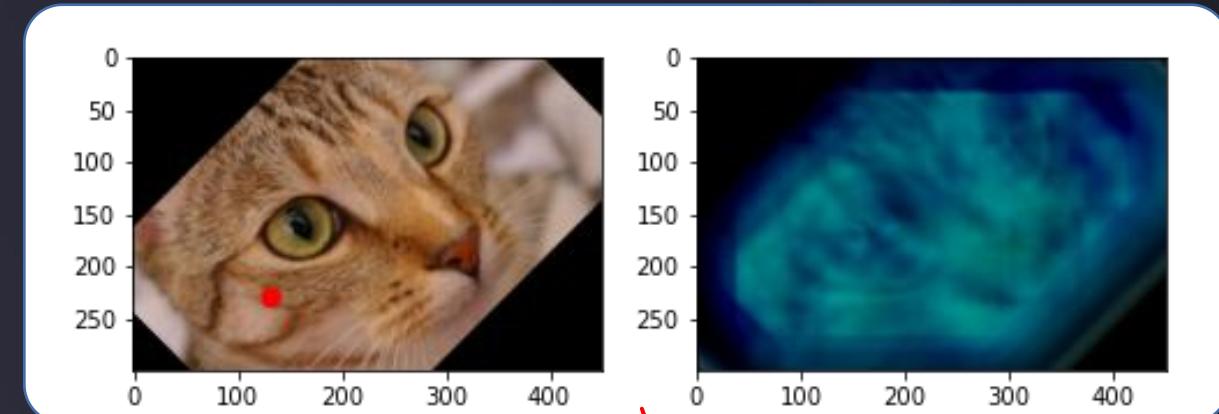
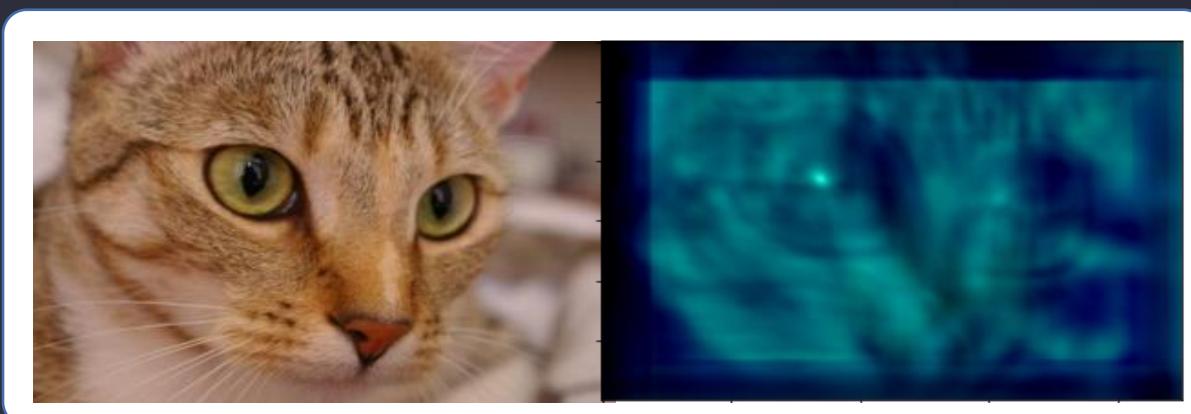
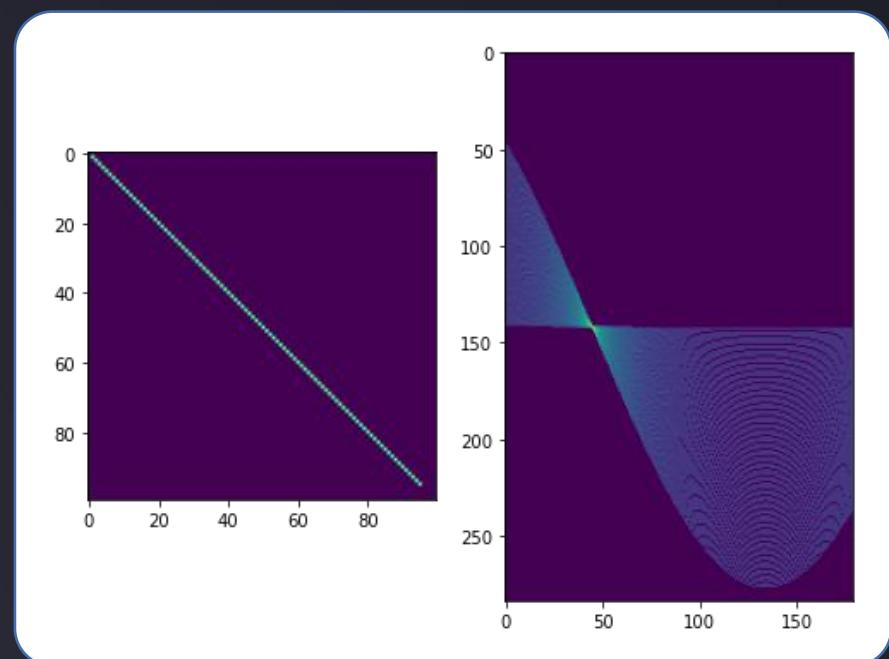


深度学习实战与计算视觉1：数值图像 处理与卷积神经网络分类任务

第五章：卷积神经网络

|上一章回顾

- 霍夫变换
- 模板匹配算法
- 经典算法的不足



Content

- 1 --神经网络的基础介绍
- 2 --神经网络的训练
- 3 --卷积神经网络的基础介绍
- 4 --经典的卷积神经网络结构

卷积神经网络与计算视觉 (Computer Vision)

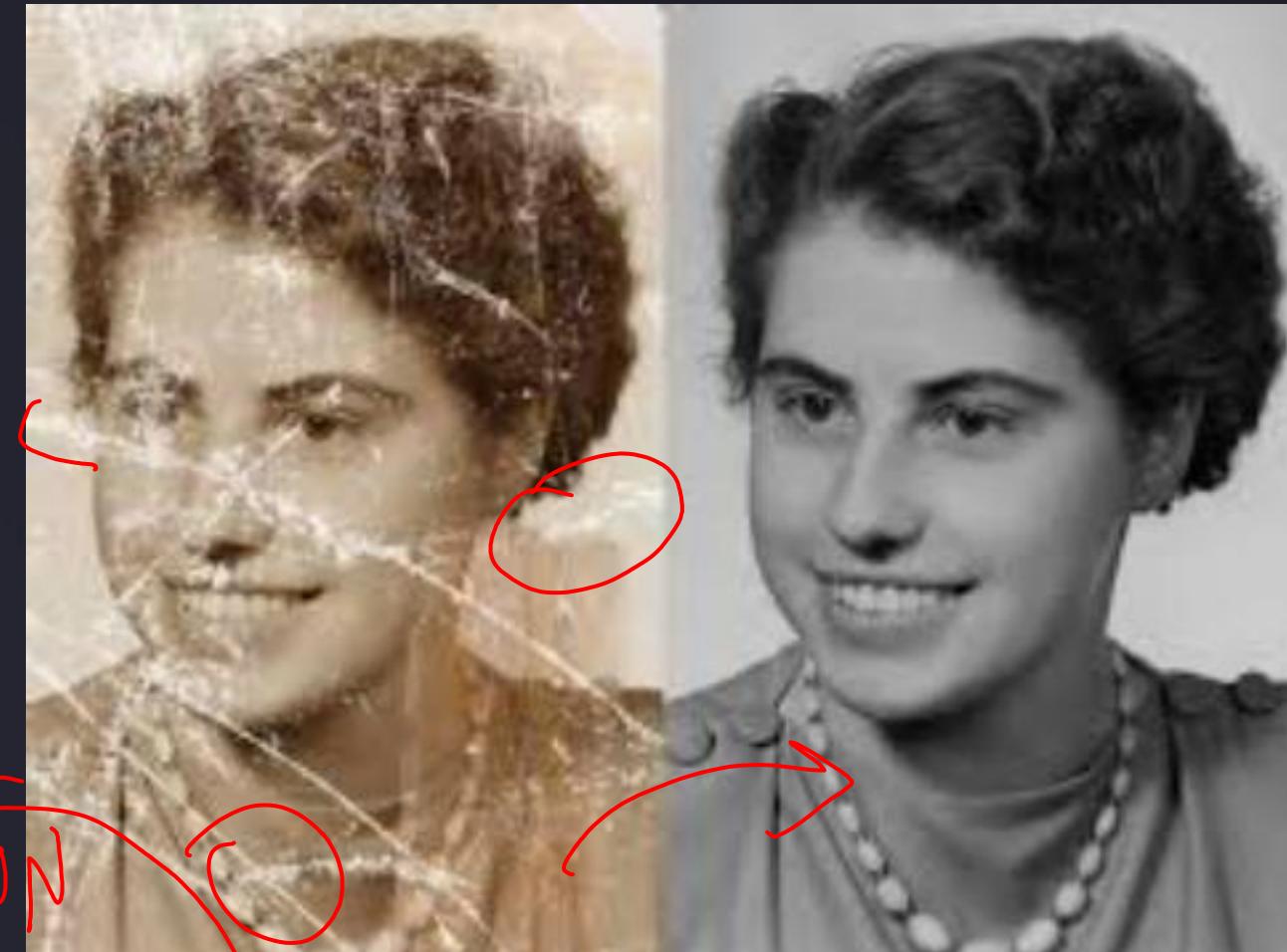
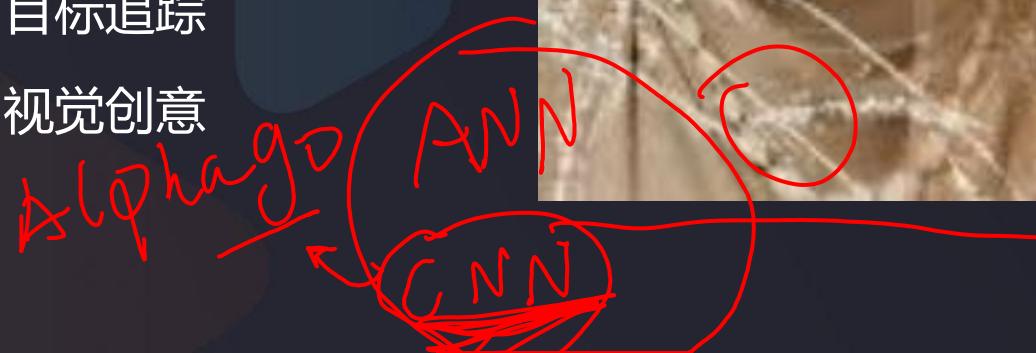
计算视觉四大主流任务

1. 图像分类/识别
2. 语义分割
3. 目标检测
4. 实例分割

其他任务

1. 图片增强
2. 目标追踪
3. 视觉创意

2016



Content

- 1 --神经网络的基础介绍**
- 2 --神经网络的训练**
- 3 --卷积神经网络的基础介绍**
- 4 --经典的卷积神经网络结构**

ANN

1943

1958

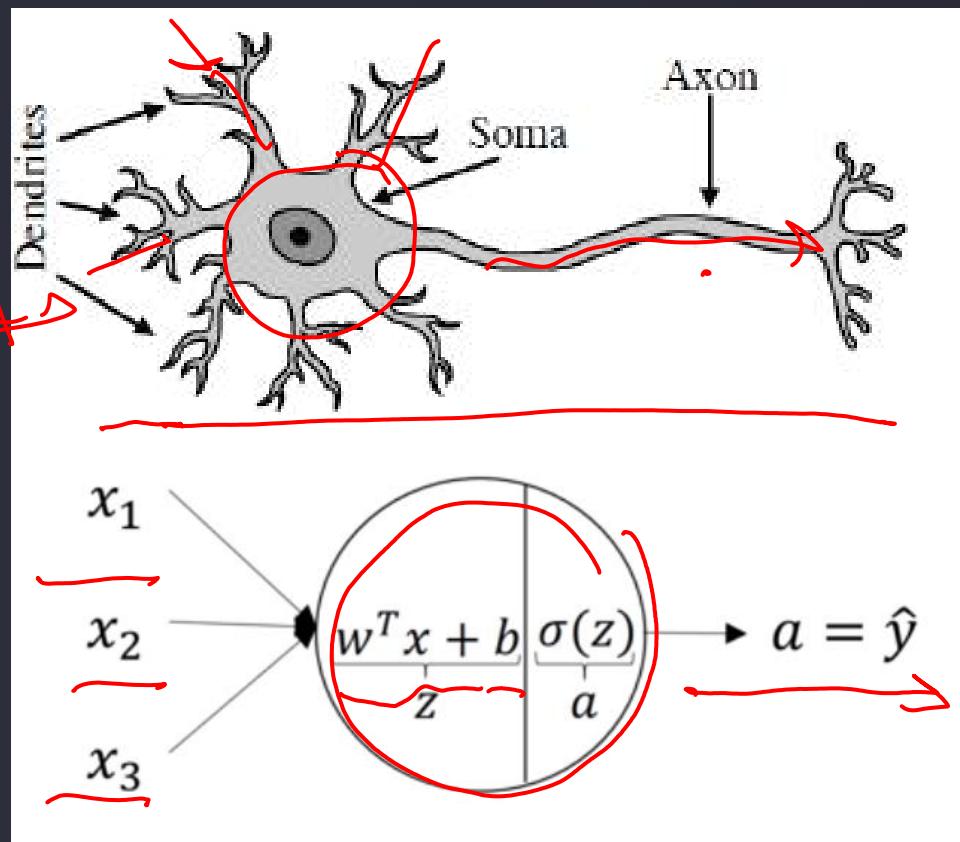
1985



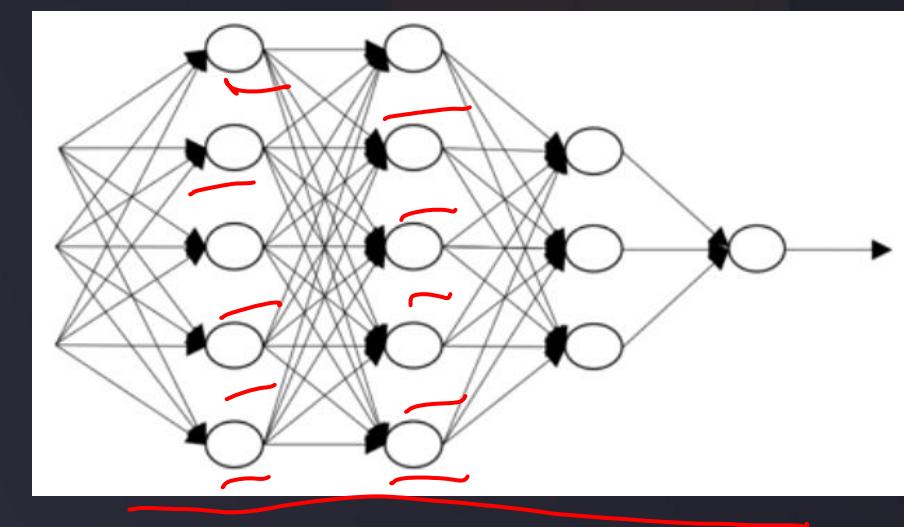
BP

0
0
6
0
—

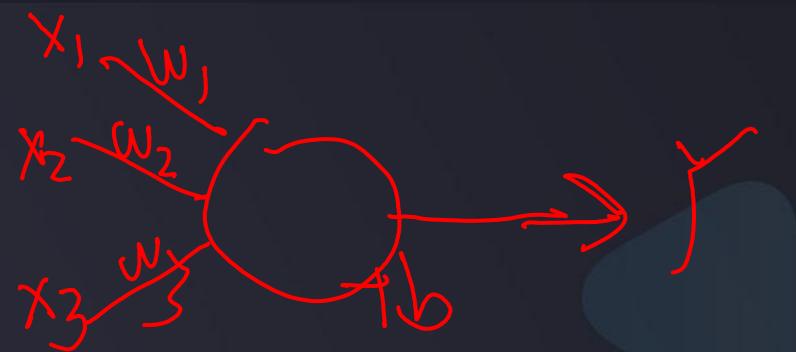
| 神经网络的基本构成



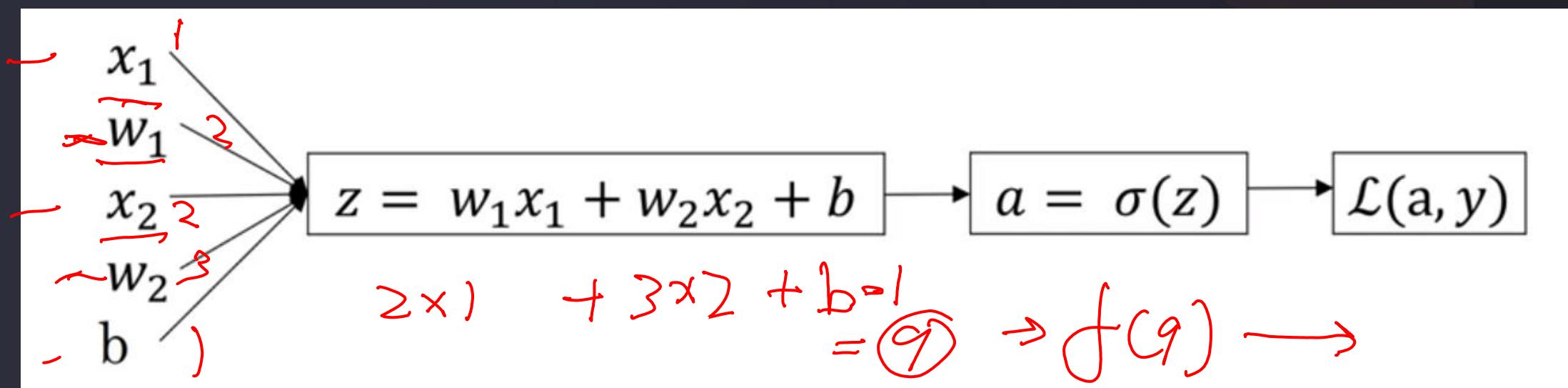
CNN
RNN



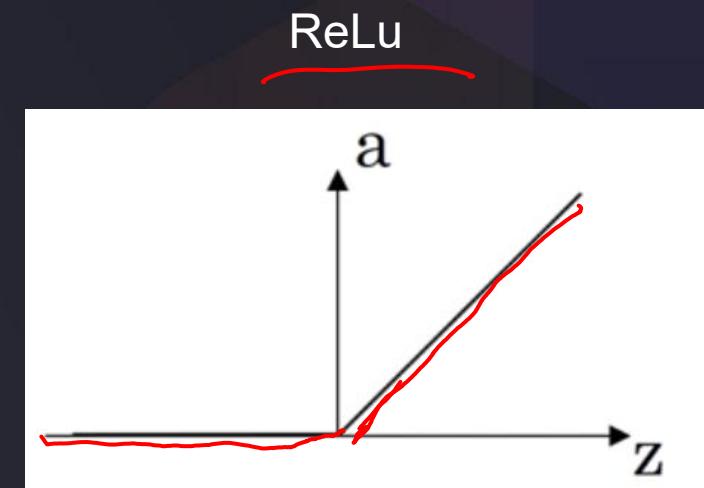
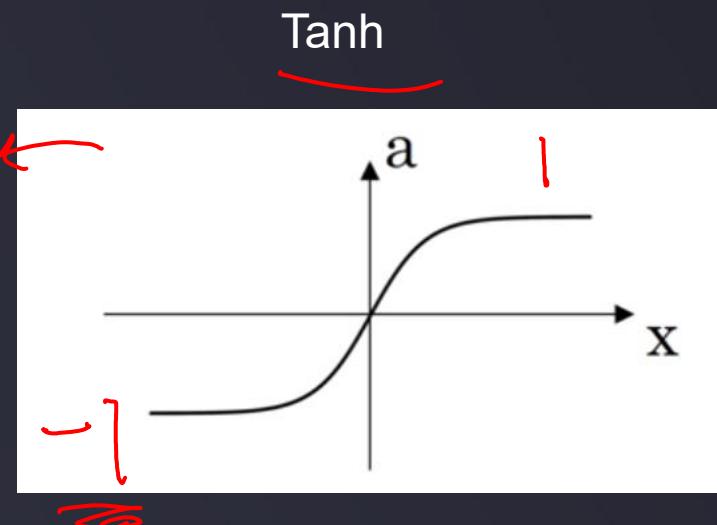
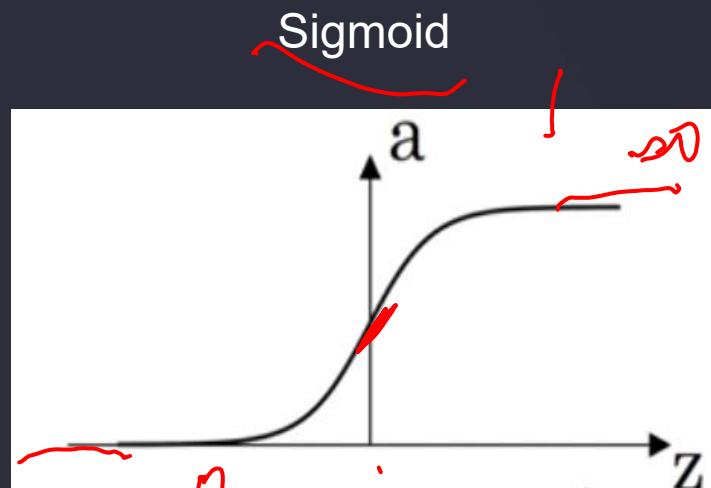
| 神经网络的基本原理



$$\begin{aligned} z &= w^T x + b \\ a &= \sigma(z) \end{aligned}$$



| 神经网络的基本原理：激活函数



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

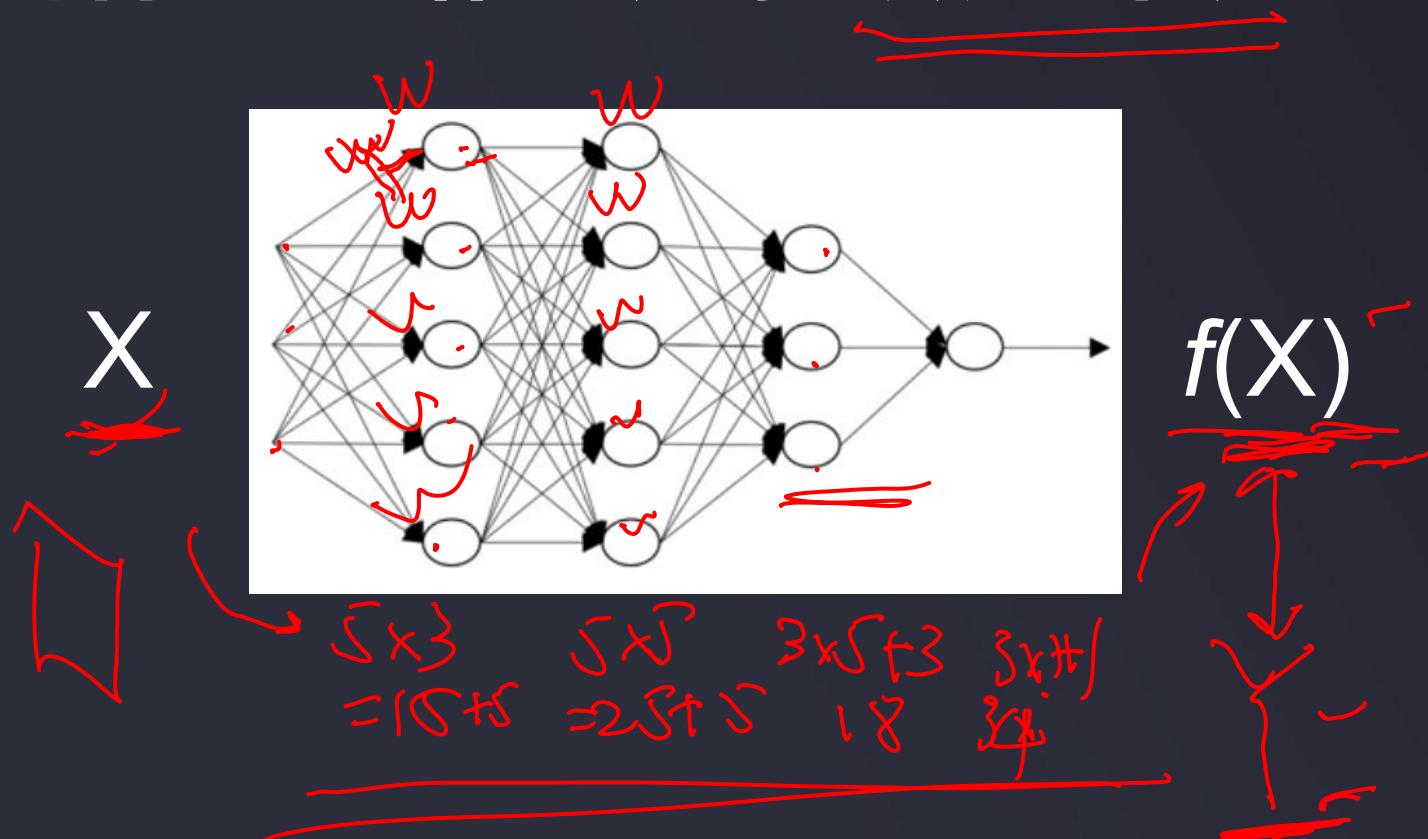
$$f(x) = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Content

- 1** --神经网络的基础介绍
- 2** --神经网络的训练
- 3** --卷积神经网络的基础介绍
- 4** --经典的卷积神经网络结构

神经网络的训练-损失函数



L1

$$L(Y, f(x)) = |Y - f(x)|$$

L2

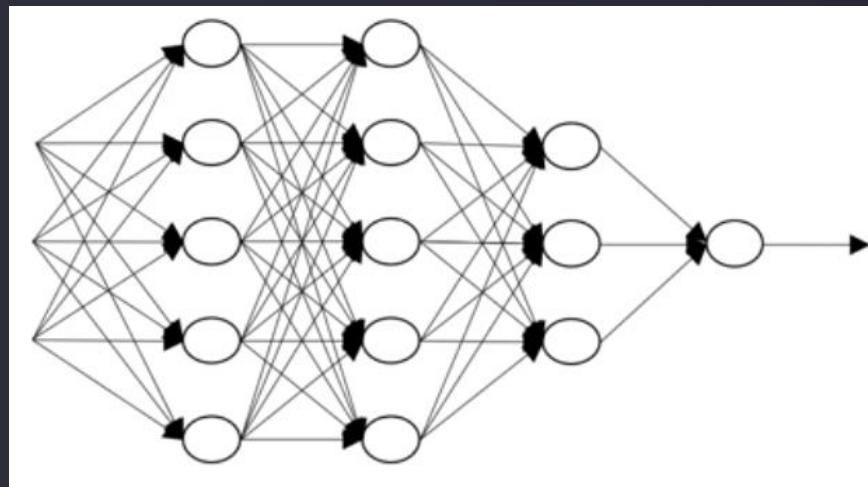
$$L(Y, f(X)) = (Y - f(X))^2$$

交叉熵

$$L(p, q) = - \sum p(x) \log(q(x))$$

| 神经网络的训练-损失函数

X



$f(X)$

L1 ✓

$$L(Y, f(x)) = |Y - f(x)|$$

L2 ✓

$$L(Y, f(X)) = (Y - f(X))^2$$

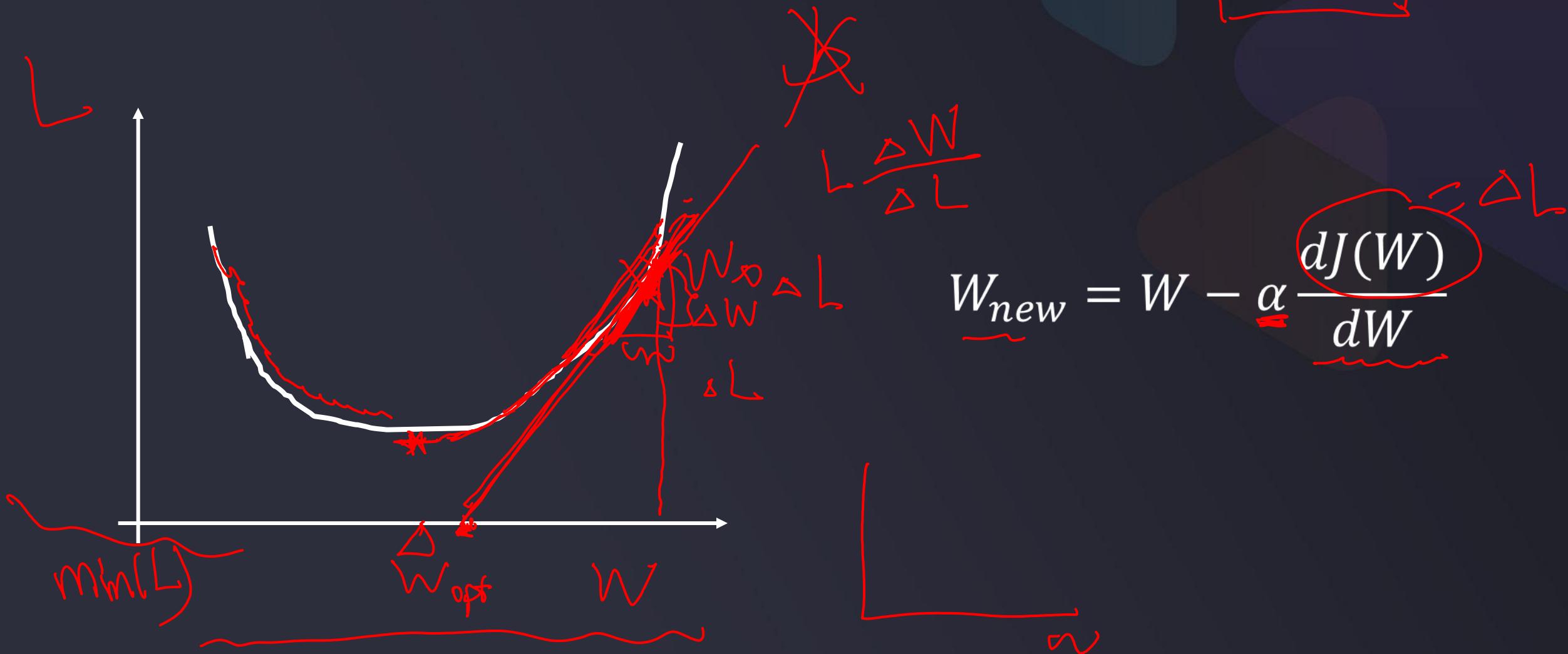
交叉熵 ✓

$$L(Y, f(X, w, b))$$

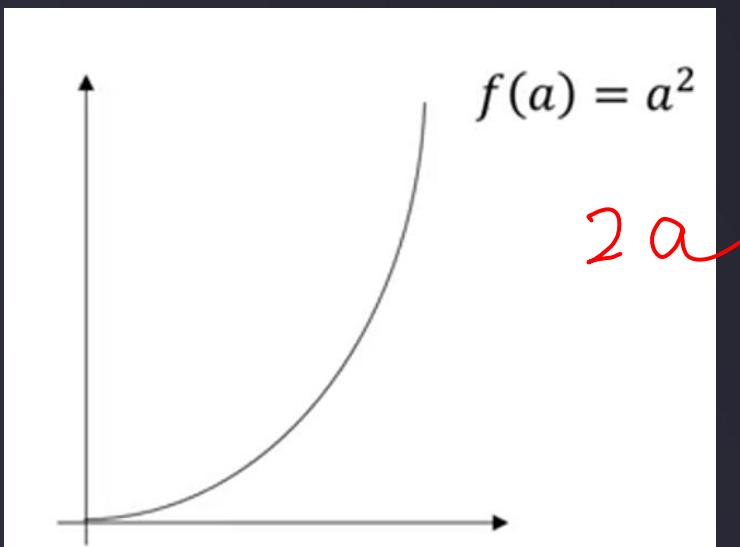
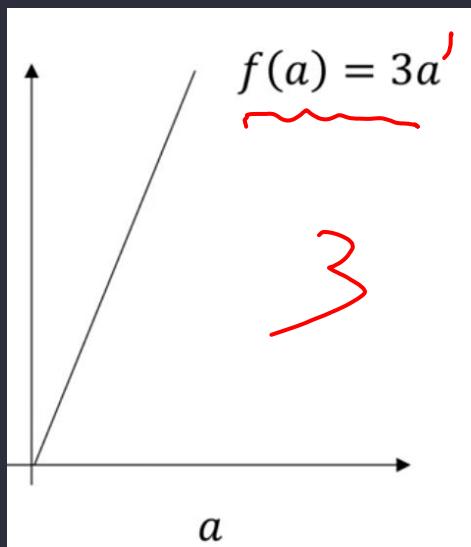
找到神经网络权重w,b最小化损失函数L

| 神经网络的训练-梯度下降

$$L(x, w)$$



神经网络的训练-计算梯度 (泰勒展开)



$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x - a)^n$$

A red wavy line highlights the general term $\frac{f^{(n)}(a)}{n!} (x - a)^n$ in the Taylor series expansion.

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \frac{e^{\theta x}}{(n+1)!} x^{n+1},$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + o(x^{2n+2}).$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots + (-1)^n \frac{x^{2n}}{(2n)!} + o(x^{2n}).$$

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots + (-1)^n \frac{x^{n+1}}{n+1} + o(x^{n+1}).$$

$$\frac{1}{1-x} = 1 + x + x^2 + \cdots + x^n + o(x^n)$$

$$(1+x)^m = 1 + mx + \frac{m(m-1)}{2!} x^2 + \cdots.$$

神经网络的训练-反向传导算法

以一个神经元为例



$$L = Y - (wx + b)$$

$$w' = w - \frac{\partial L}{\partial w}$$

$$1 - 0.5 \times 1 = 0.5$$

$$1 + 0.5 = 1.5$$

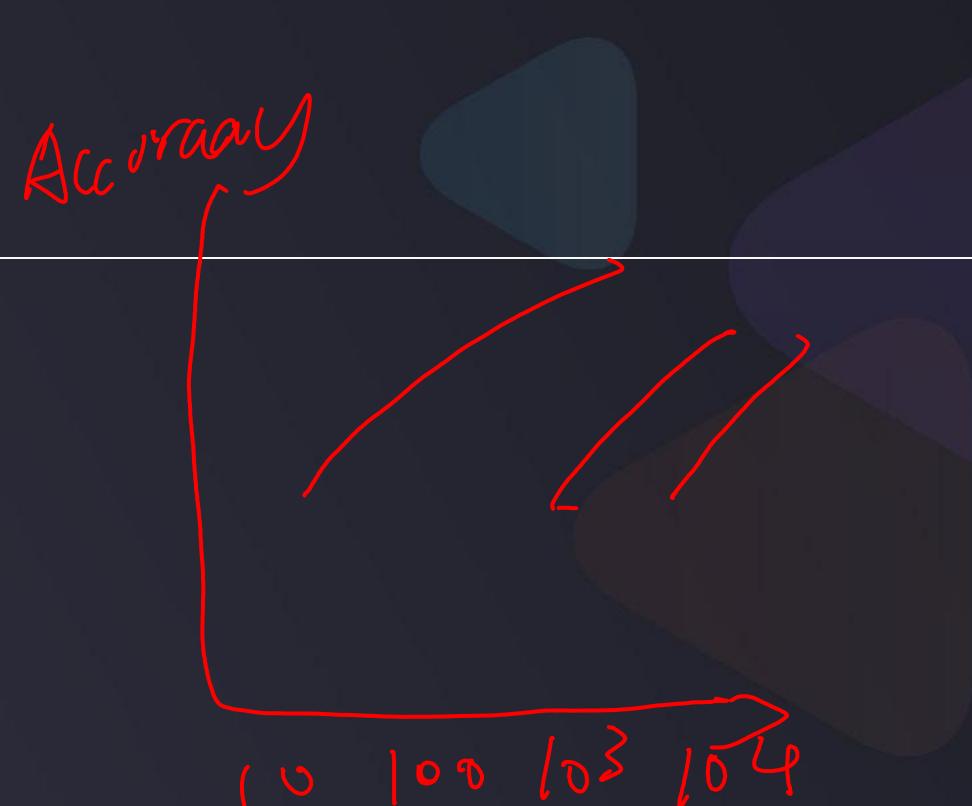
$$b' = b - \frac{\partial L}{\partial b}$$

$$1 - (-0.5) = 1.5$$

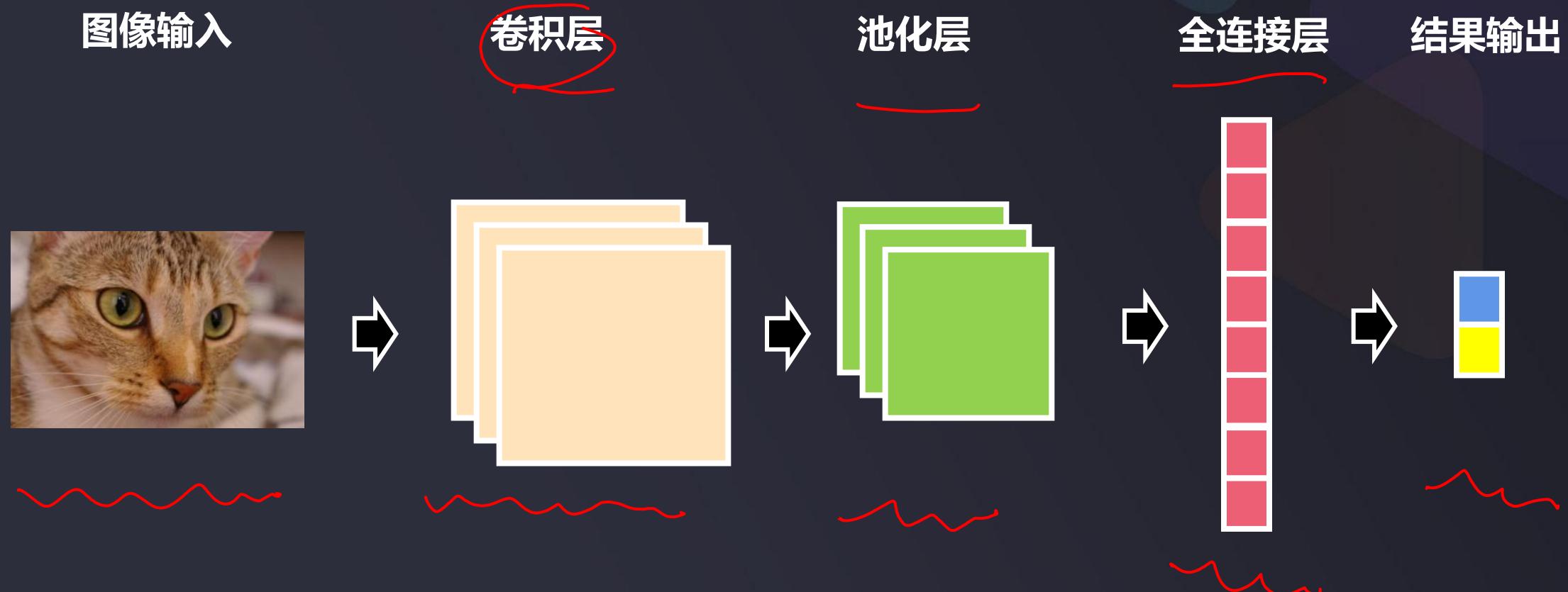
$$f(x) = wx + b = 1.5 \times 1 + 1.5 = 3$$

Content

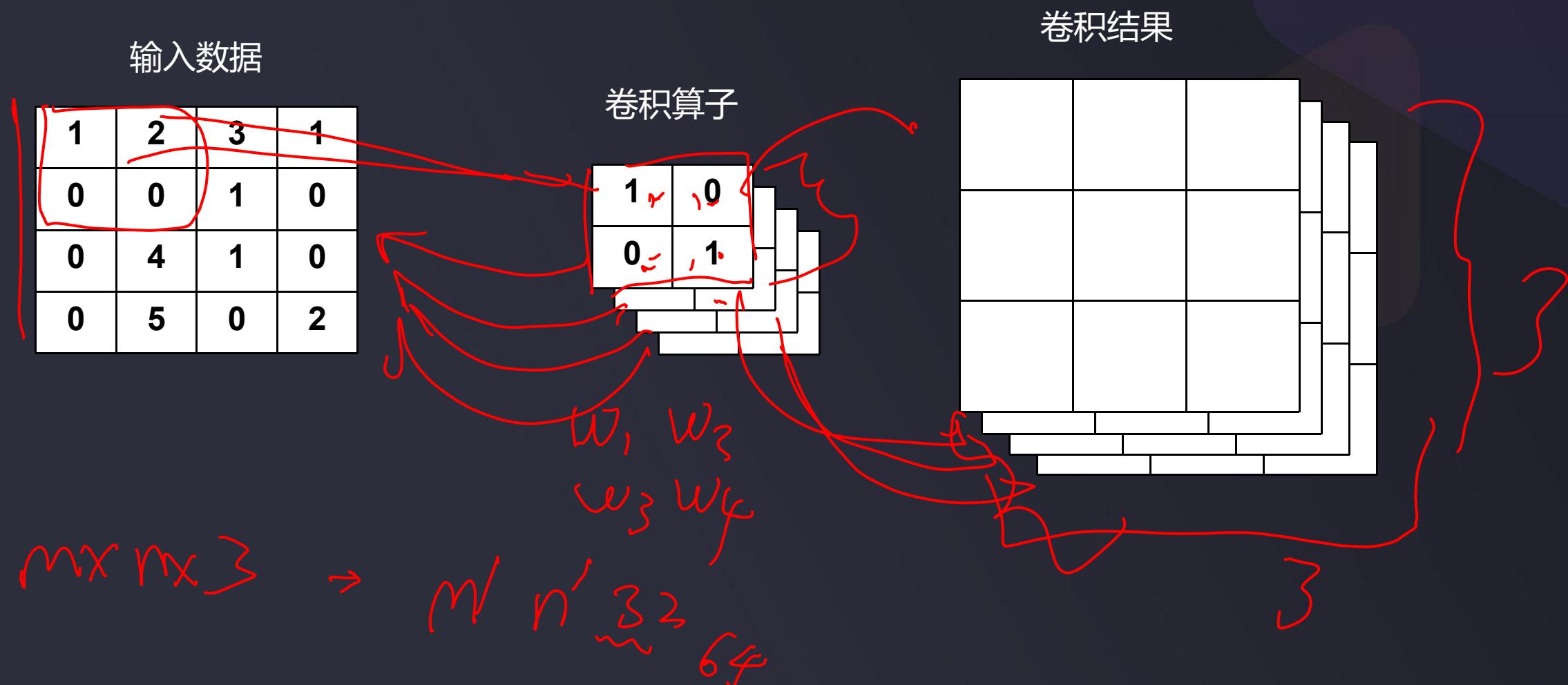
- 1 --神经网络的基础介绍
- 2 --神经网络的训练
- 3 --卷积神经网络的基础介绍
- 4 --经典的卷积神经网络结构



卷积神经网络



卷积层：边缘检测



卷积层：卷积步长stride和padding



卷积层：卷积后尺寸大小

输入数据

1	2	3	1
0	0	1	0
0	4	1	0
0	5	0	2

n

卷积算子

1	0
0	1

2

3~J

卷积结果



$$\frac{n+2p-f}{s} + 1$$

~~~~~

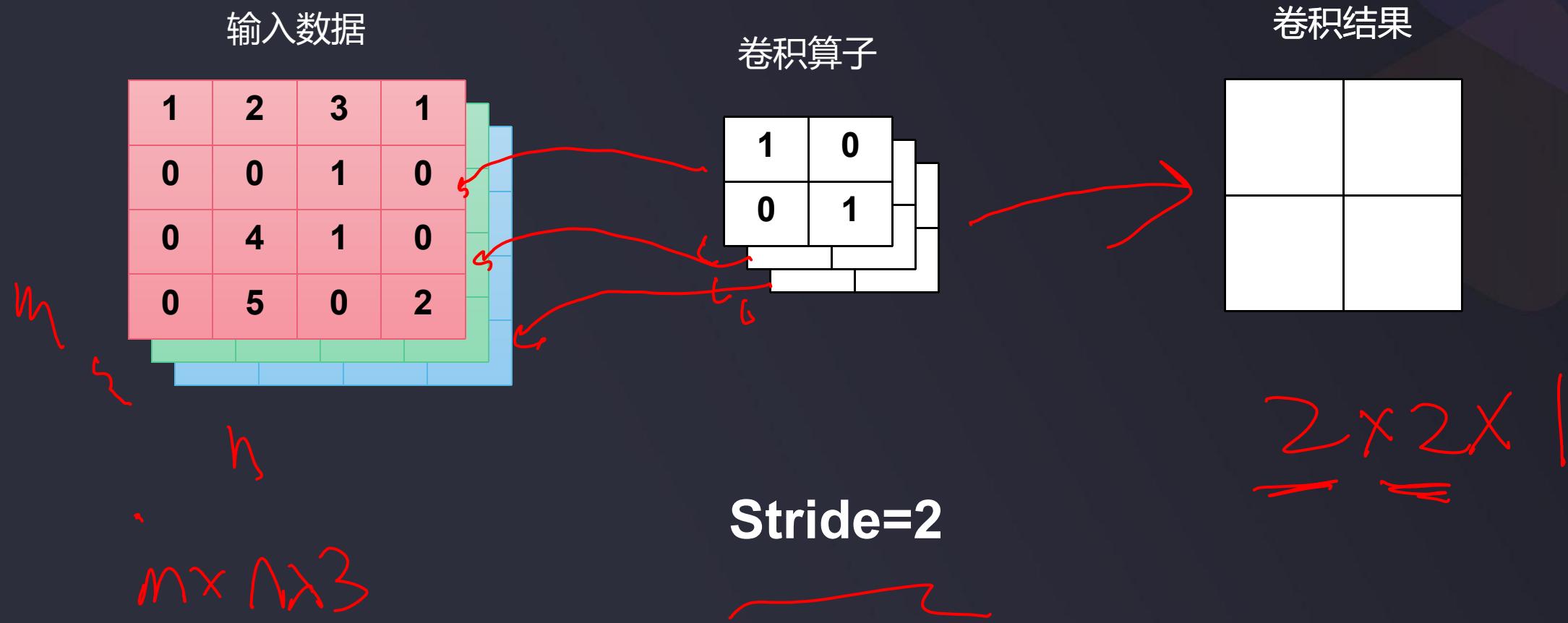
n: 图片大小

p: padding大小

f: 卷积算子大小

s: 步长Stride

# 卷积层：RGB3维图像卷积或更高维度矩阵卷积



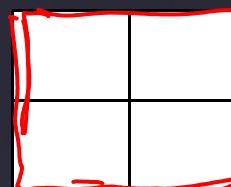
# 卷积神经网络：池化层

输入数据

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 4 | 1 | 0 |
| 0 | 5 | 0 | 2 |



池化算子



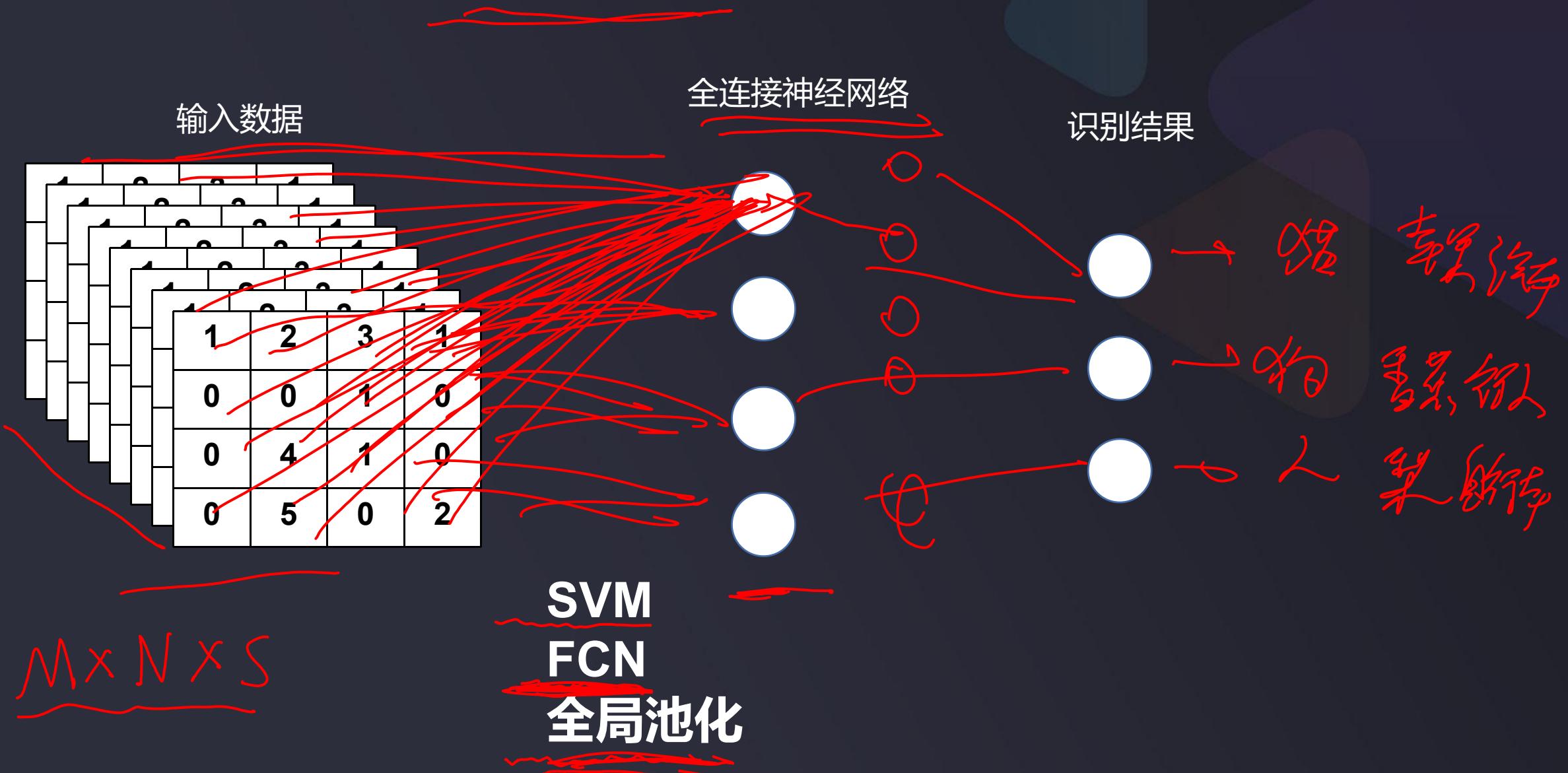
池化结果

|   |   |
|---|---|
| 2 | 3 |
| 5 | 2 |

2x2

步长 = 2  
算子大小 = 2  
MaxPooling  
AveragePooling

# 卷积神经网络：全连接层

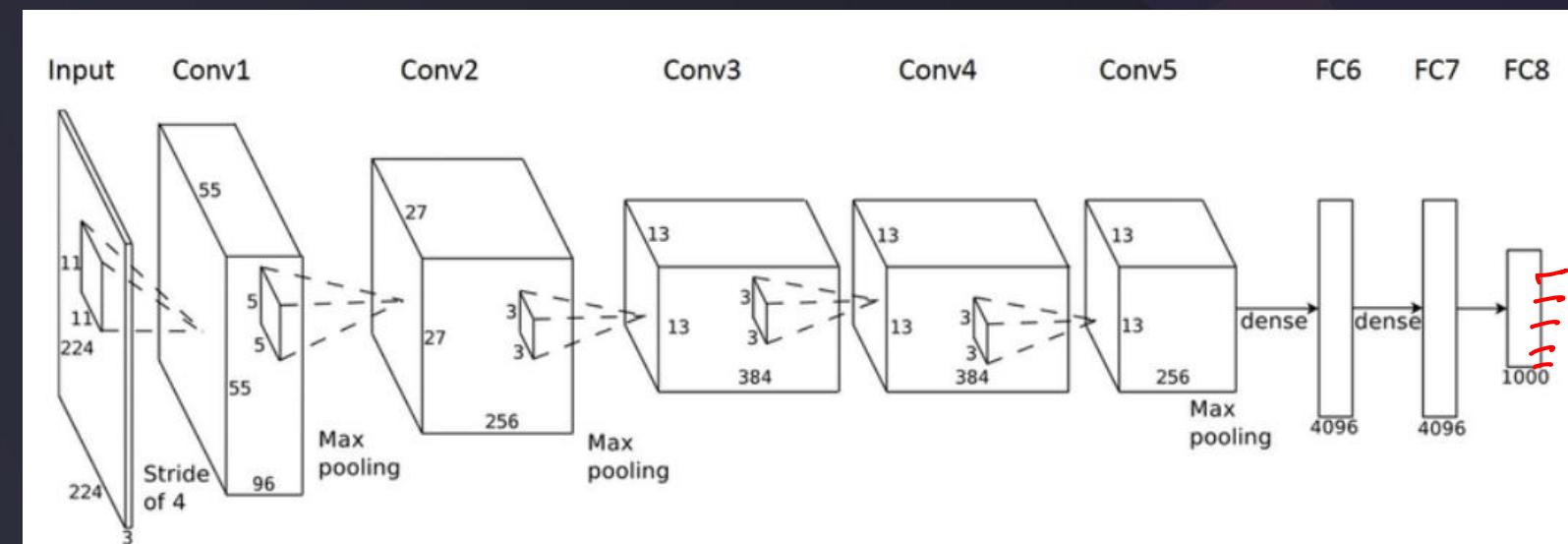
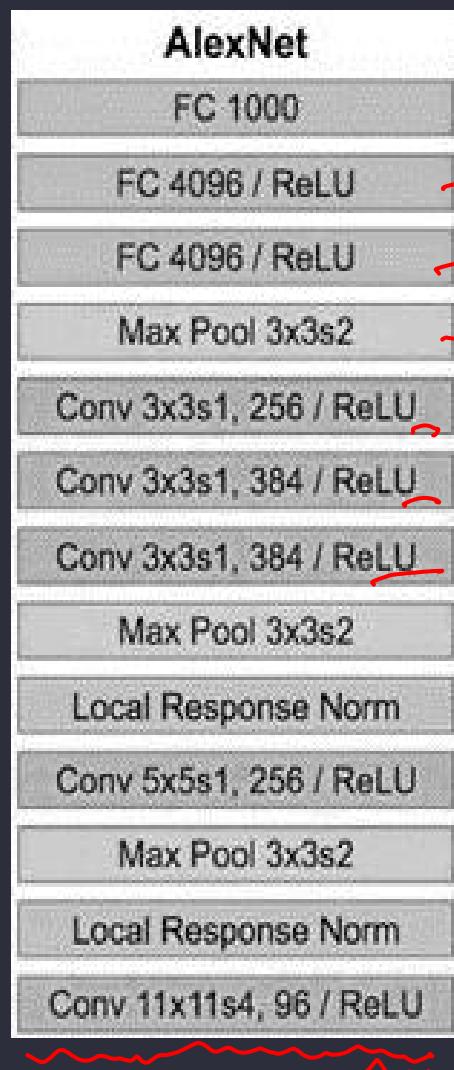


# Content

---

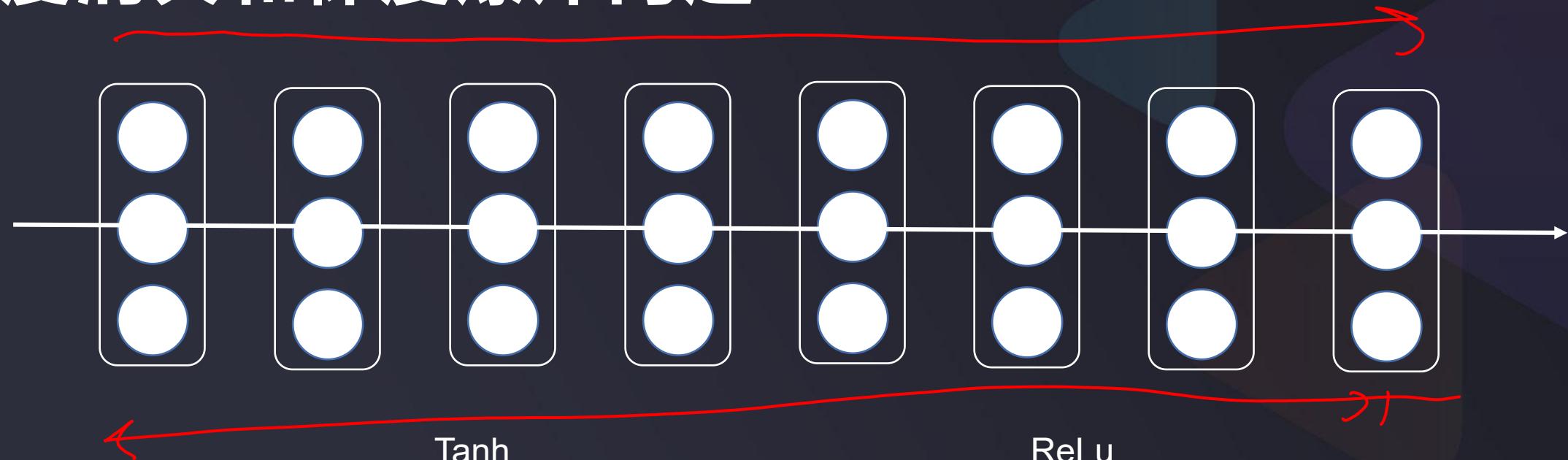
- 1 --神经网络的基础介绍
- 2 --神经网络的训练
- 3 --卷积神经网络的基础介绍
- 4 --经典的卷积神经网络结构

# 经典卷积神经网络结构：ALEXNET

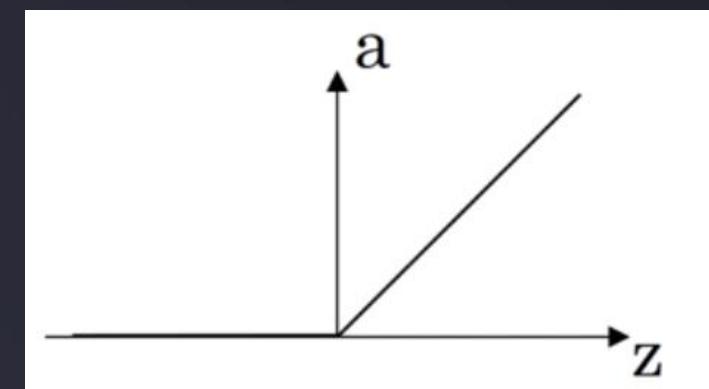
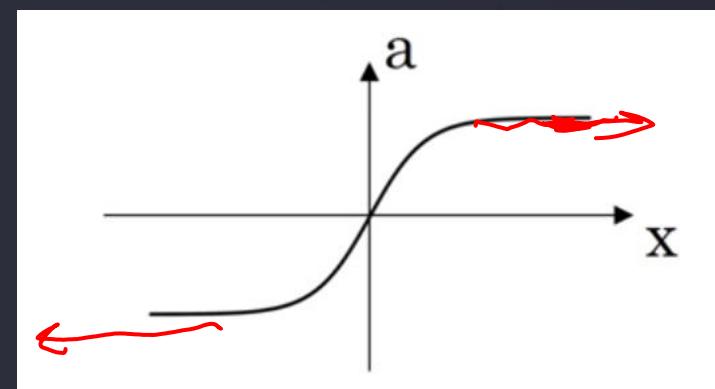


Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

# | 梯度消失和梯度爆炸问题



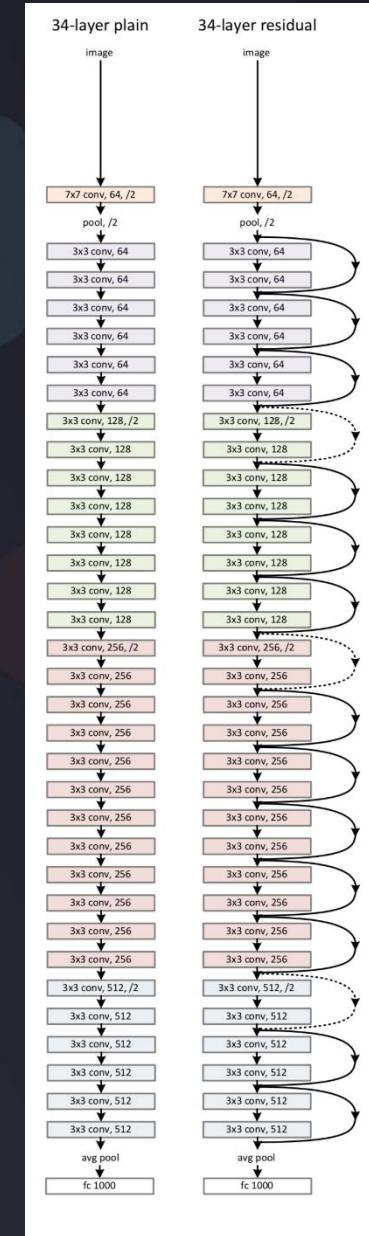
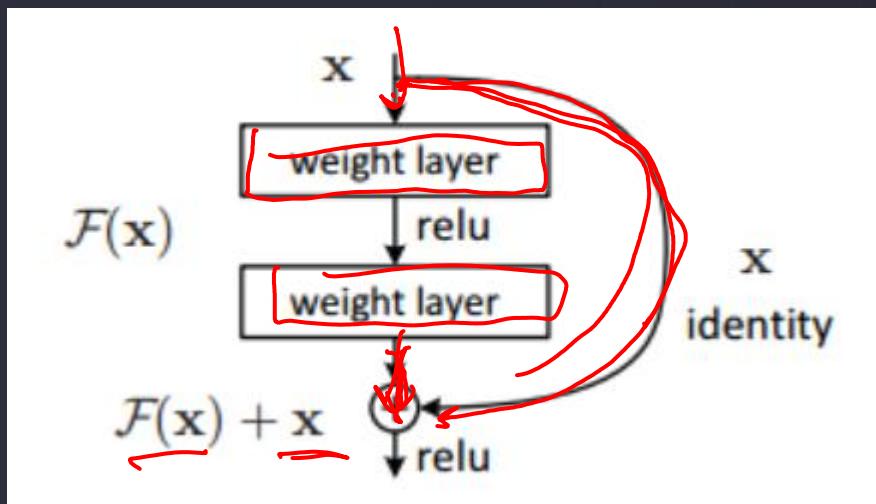
$2^{10 \sim 1024}$   
1 0.03<sup>10</sup> ~ 0.0009]



$w > 1$

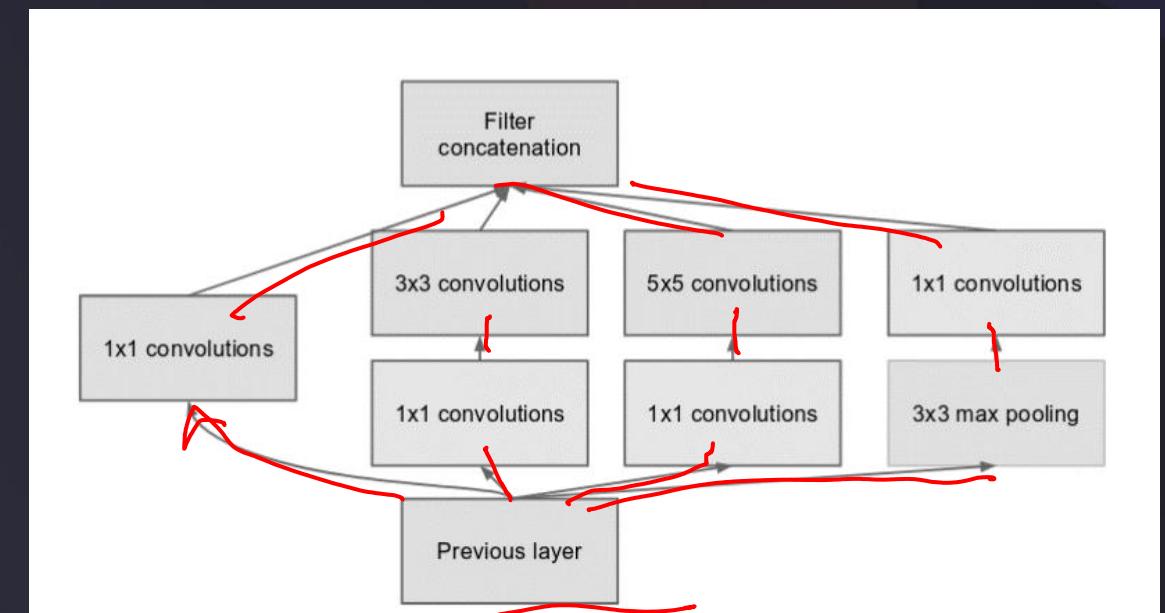
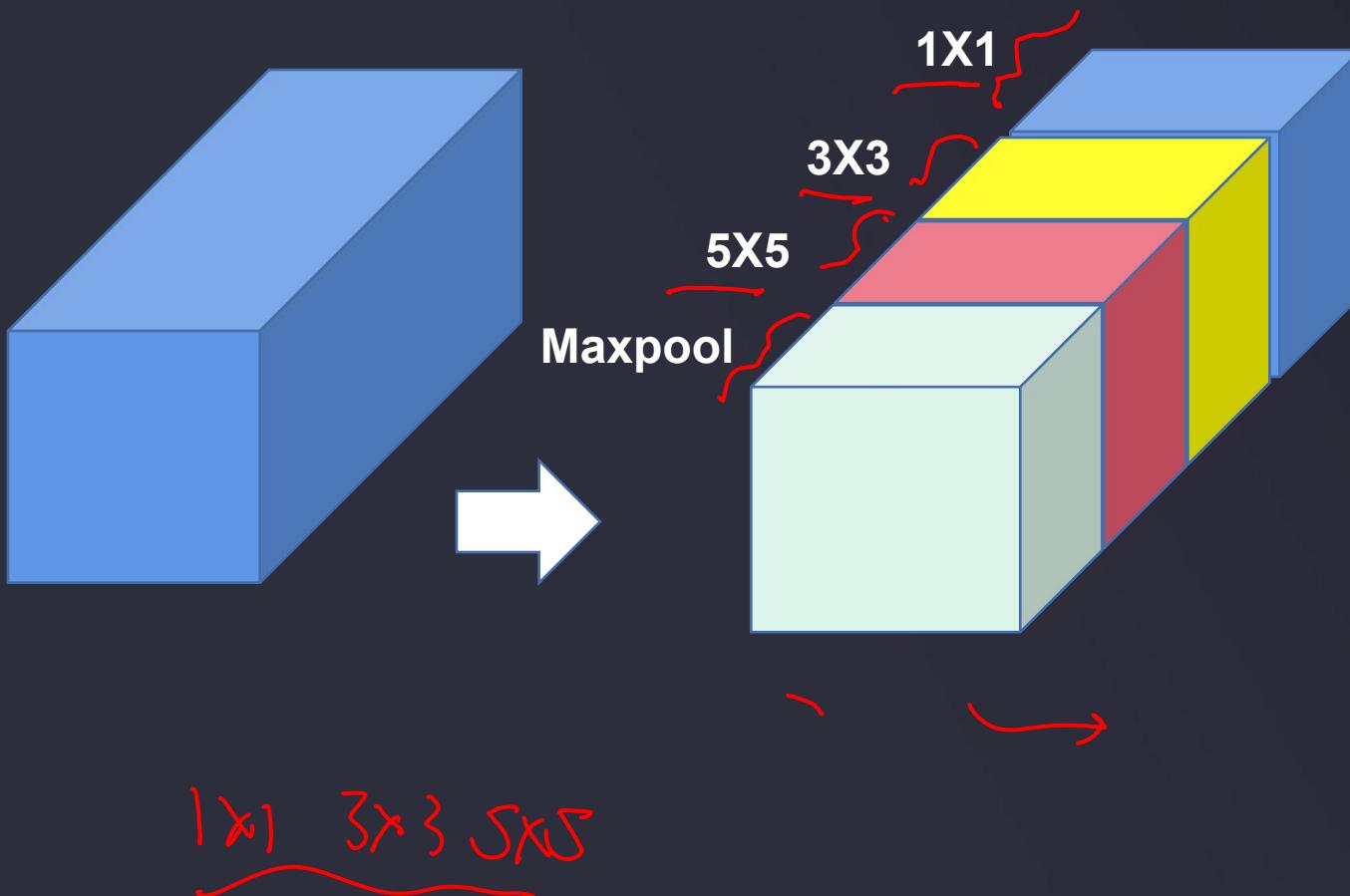
# 经典卷积神经网络结构：ResNet

ILSVRC



He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

# |经典卷积神经网络结构：Inception



Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

# | 小结

- 神经网络和卷积神经网络的发展历程，底层逻辑，训练方式，和经典的结构
- 下一章我们将讲解卷积神经网络实战应用和优化的底层知识



# 深度学习实战与计算视觉1：数值图像 处理与卷积神经网络分类任务

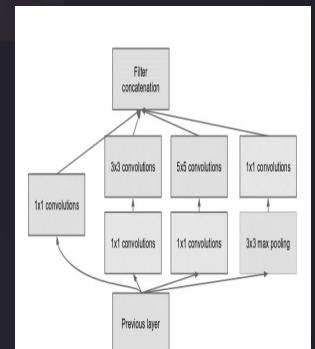
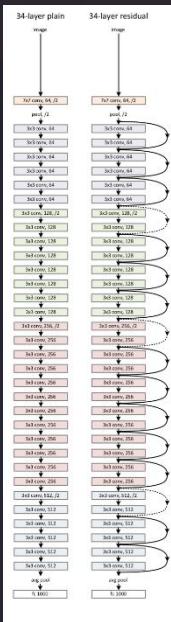
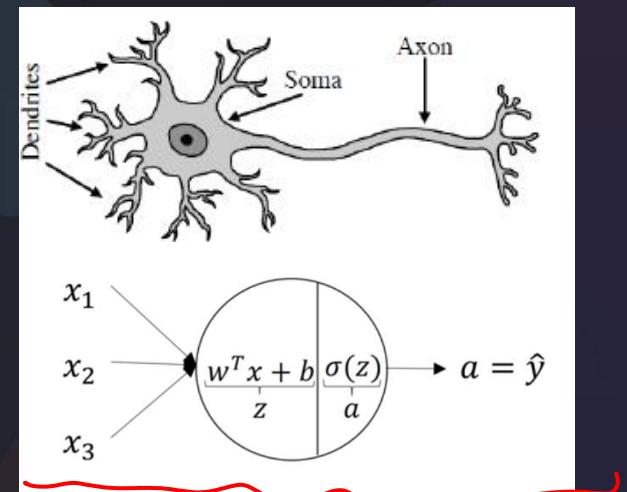
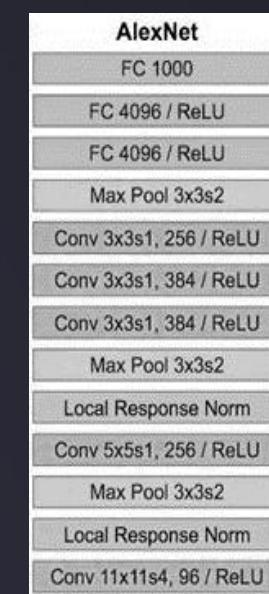
第六章：卷积神经网络的优化

# 上一章回顾

## 1. 神经网络的基础知识和训练

## 2. 卷积神经网络的基础知识

## 3. 经典的卷积神经网络结构



# Content

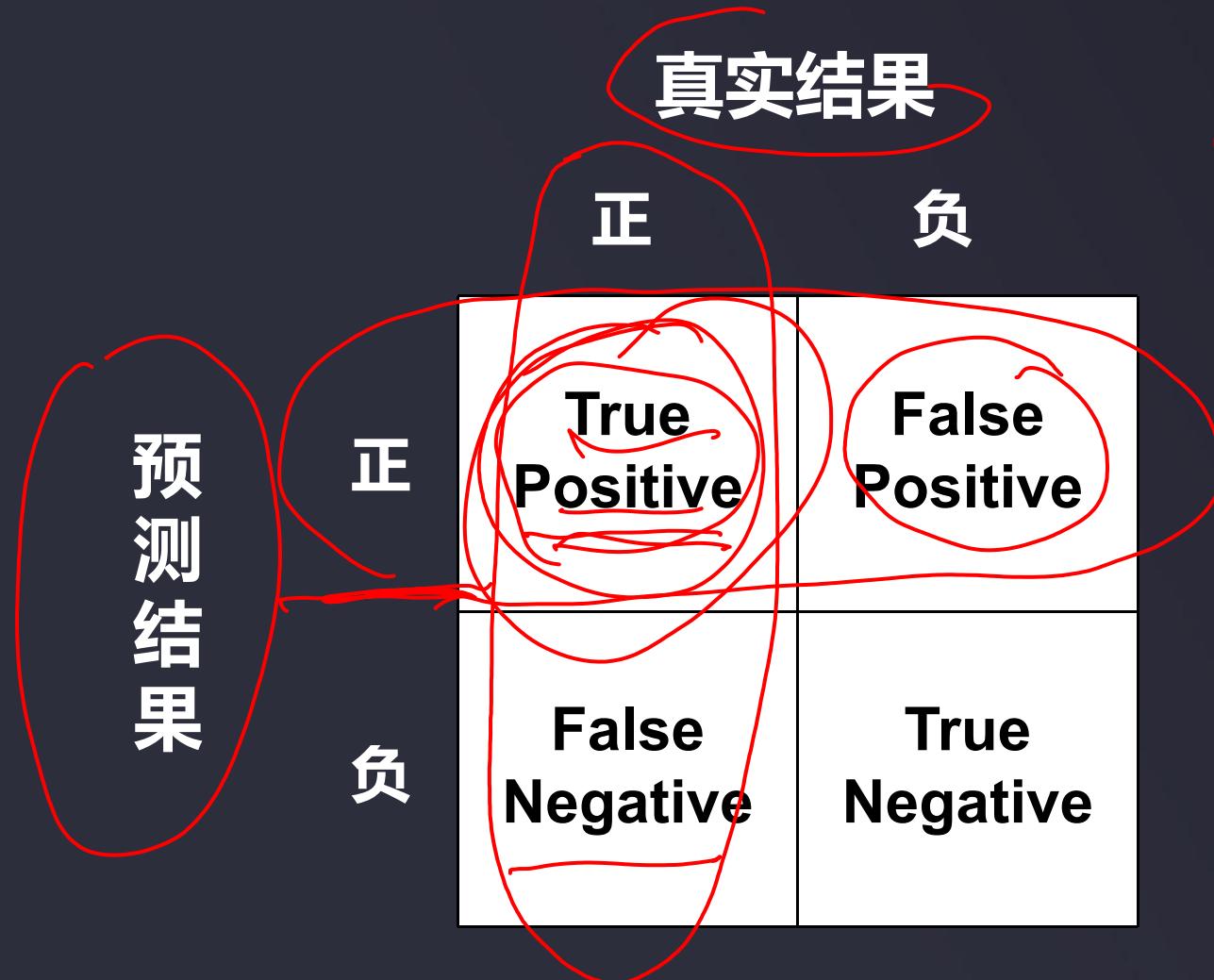
- 1 --模型优化：过拟合问题
- 2 --模型优化：梯度消失/爆炸问题
- 3 --模型优化：训练的优化
- 4 --模型优化：其他优化策略

# Content

---

- 1 --模型优化：过拟合问题
- 2 --模型优化：梯度消失/爆炸问题
- 3 --模型优化：训练的优化
- 4 --模型优化：其他优化策略

# 衡量模型的表现

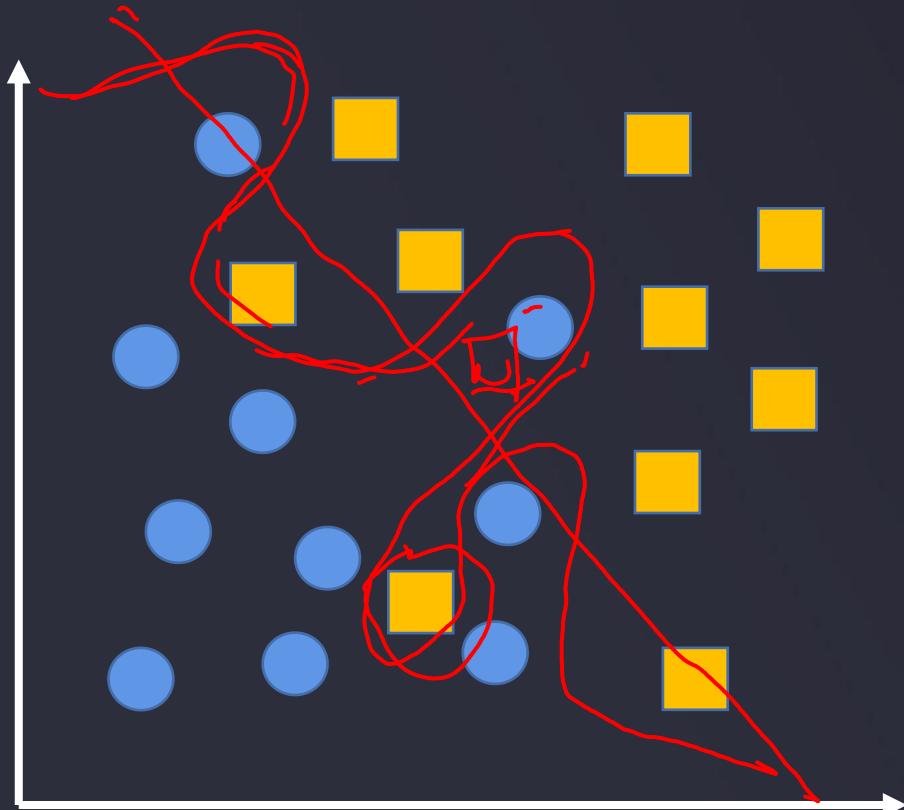


$w_1 \text{ Precision} + w_2 \text{ Recall}$

1. Accuracy =  $(TP+TN)/(TP+FP+FN+TN)$
2. Precision =  $TP/(TP+FP)$
3. Recall =  $TP/(TP+FN)$

100 99 1  
 ② →  $\frac{1}{3}$   $Precision = \frac{1}{3}$   
 $Recall = \frac{1}{1} = 100\%$

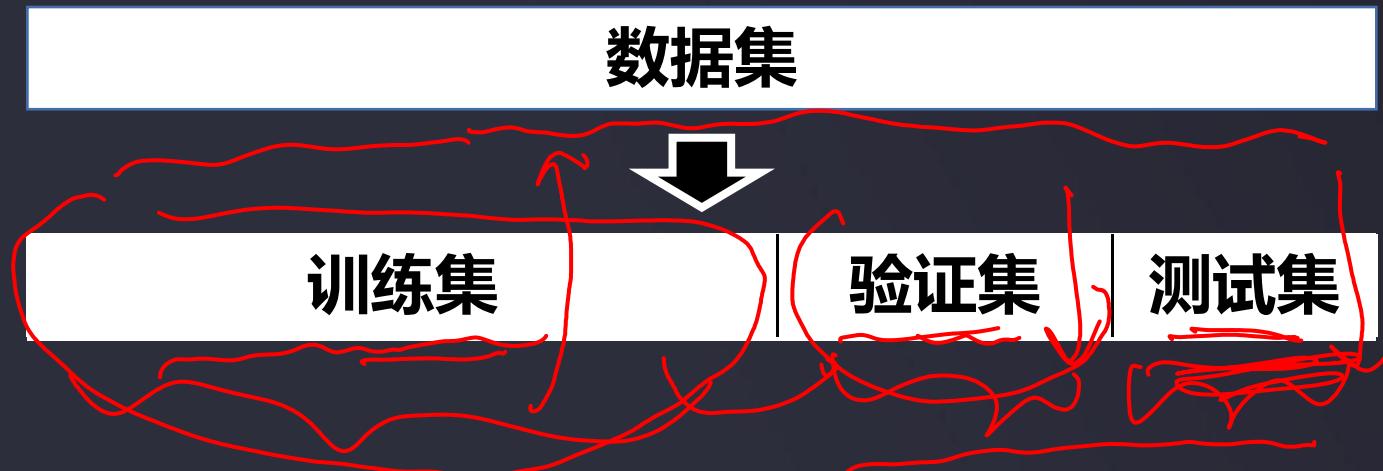
# 过拟合问题



## 优化方法

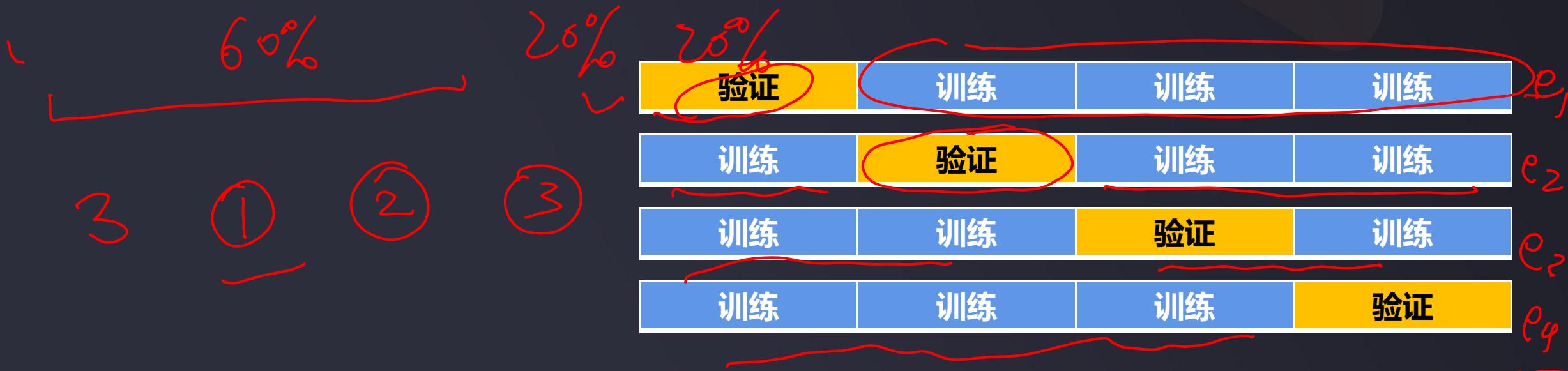
1. 训练验证集划分/交叉验证
2. 扩大数据集和代表性/数据增强Data augmentation
3. 正则化Regularization
4. 随机失活Dropout

# 训练验证集划分/交叉验证



10-fold

4-fold



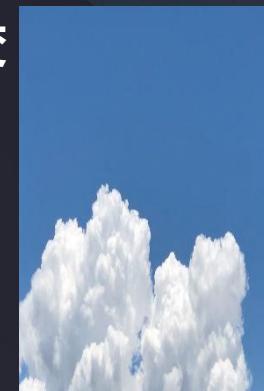
# | 数据增强 Data augmentation



旋转



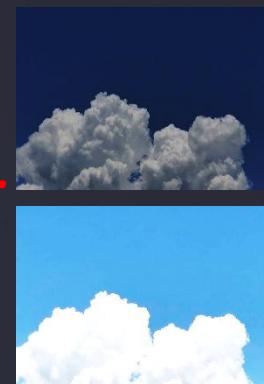
形变



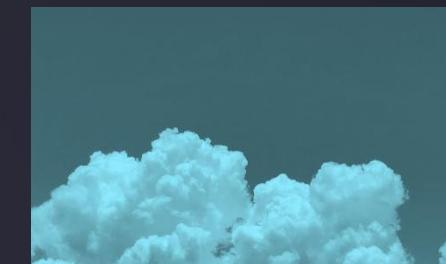
镜像



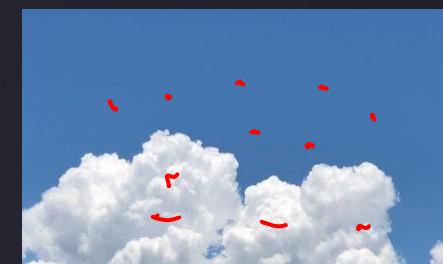
亮度



变色



噪音



# 正则化 Regularization

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

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

$$L(Y, f(x)) \rightarrow L(Y, f(x), w)$$

→

$$L(Y, f(x)) = \|Y - f(x)\|_2 + \alpha \|w\|_2$$

$$(1) (2) (1) = 4$$

$$\begin{pmatrix} 0 & 2 & -1 \\ 1 & 1 & -1 \end{pmatrix} = 6$$

(

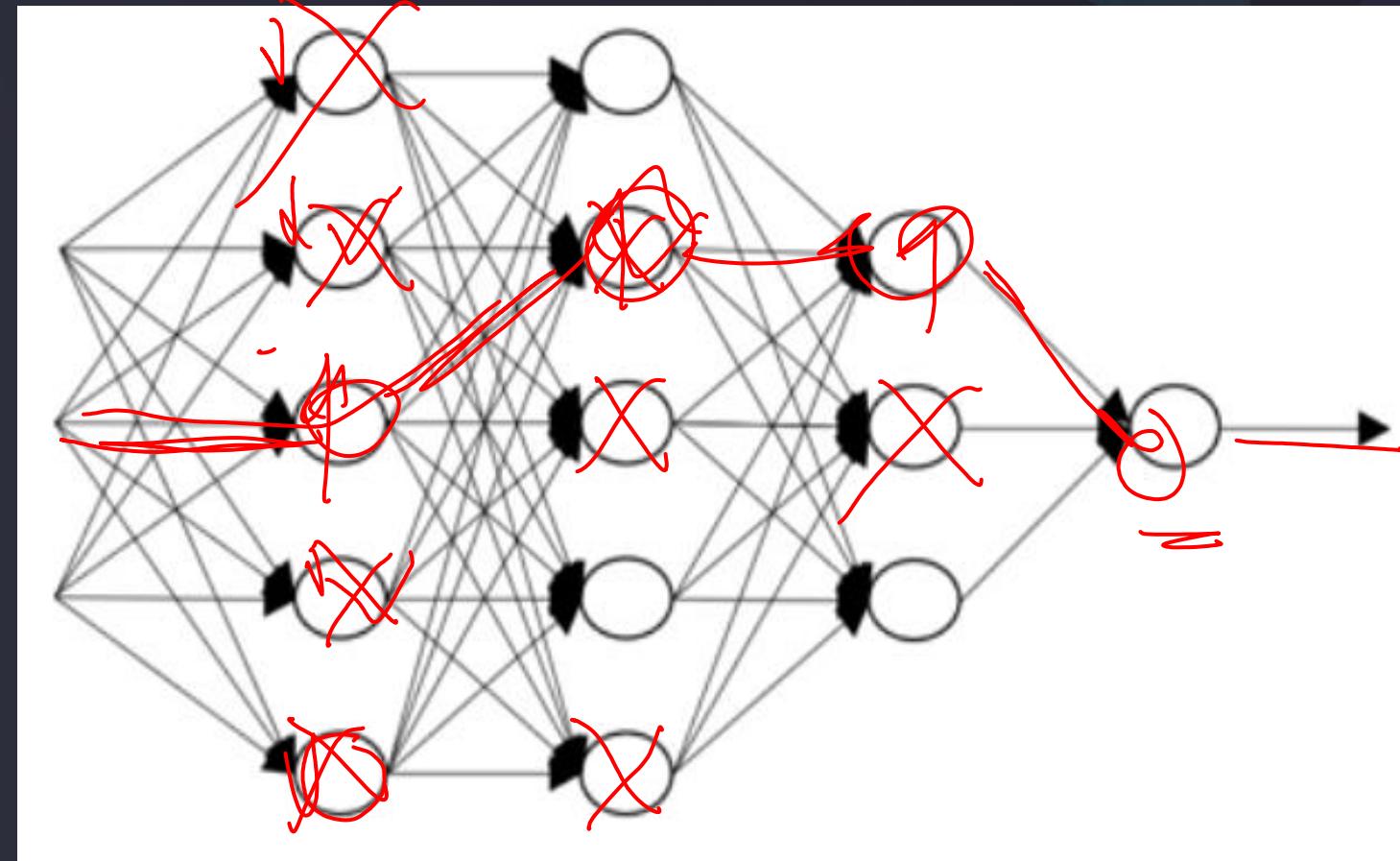
- L0范数：向量中非零元素的个数，记为 $\|w\|_0$
- L1范数：绝对值之和，记为 $\|w\|_1$
- L2范数：平方和或者说模，记为 $\|w\|_2$



$f(x)$

$w$

# | 随机失活Dropout

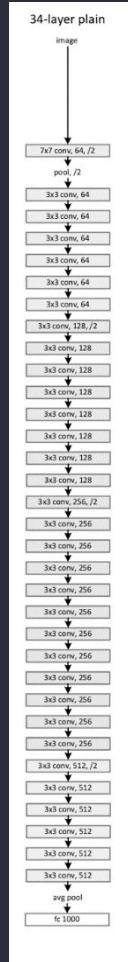


# Content

---

- 1 --模型优化：过拟合问题
- 2 --模型优化：梯度消失/爆炸问题
- 3 --模型优化：训练的优化
- 4 --模型优化：其他优化策略

# 深度卷积神经网络梯度消失和梯度爆炸问题



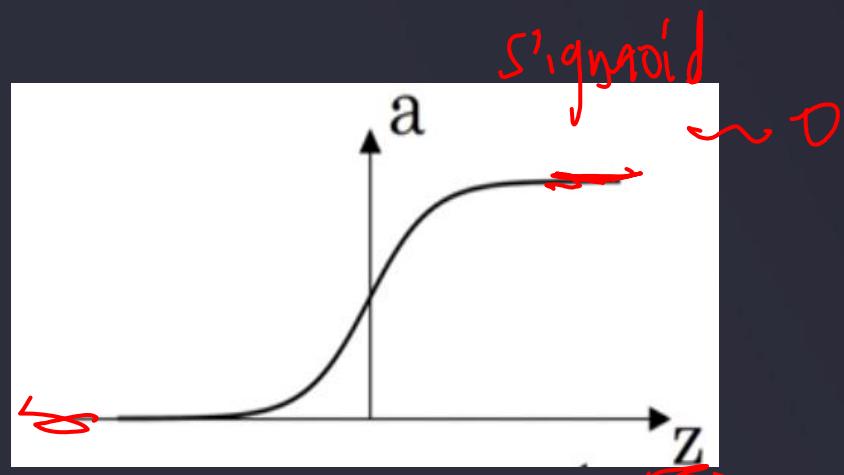
$$\frac{\partial C}{\partial b_1} = \frac{\partial C}{\partial y_4} \frac{\partial y_4}{\partial z_4} \frac{\partial z_4}{\partial x_4} \frac{\partial x_4}{\partial z_3} \frac{\partial z_3}{\partial x_3} \frac{\partial x_3}{\partial z_2} \frac{\partial z_2}{\partial x_2} \frac{\partial x_2}{\partial z_1} \frac{\partial z_1}{\partial b_1}$$

$$\begin{matrix} 2^0 &= 1 \\ 2^1 &= 2 \\ 2^2 &= 4 \\ 2^3 &= 8 \\ 2^4 &= 16 \\ 2^5 &= 32 \\ 2^6 &= 64 \\ 2^7 &= 128 \\ 2^8 &= 256 \\ 2^9 &= 512 \\ 2^{10} &= 1024 \end{matrix}$$

## 优化方法

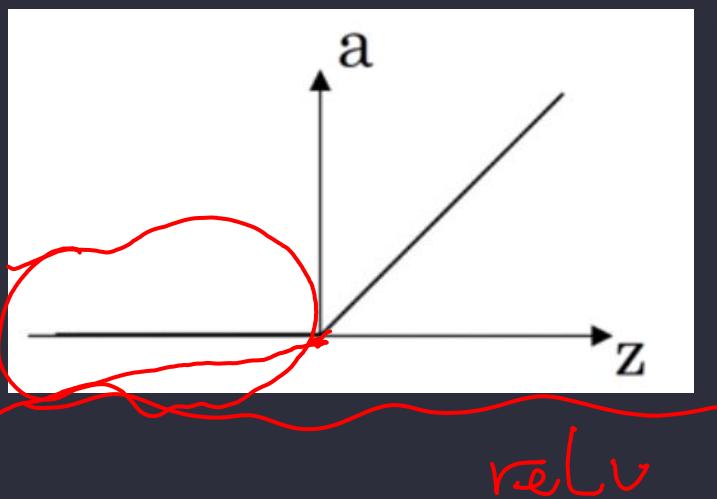
1. 残差神经网络连接方式
2. 非饱和的激活函数 (如 ReLU)
3. 梯度截断 (Gradient Clipping)
4. 好的参数初始化方式
5. 批量规范化 (Batch Normalization)
6. 更快的优化器
7. LSTM

# 非饱和的激活函数 (ReLU) 与梯度截断 (Gradient Clipping)



Leaky ReLU

$> \theta$        $\approx \theta$   
 $< -\theta$        $\approx -\theta$



# 网络参数的初始化

初始化的方法：

1. 全零/全一
2. 随机初始化
3. Xavier初始化
4. He初始化
5. Pre-train初始化（迁移学习）

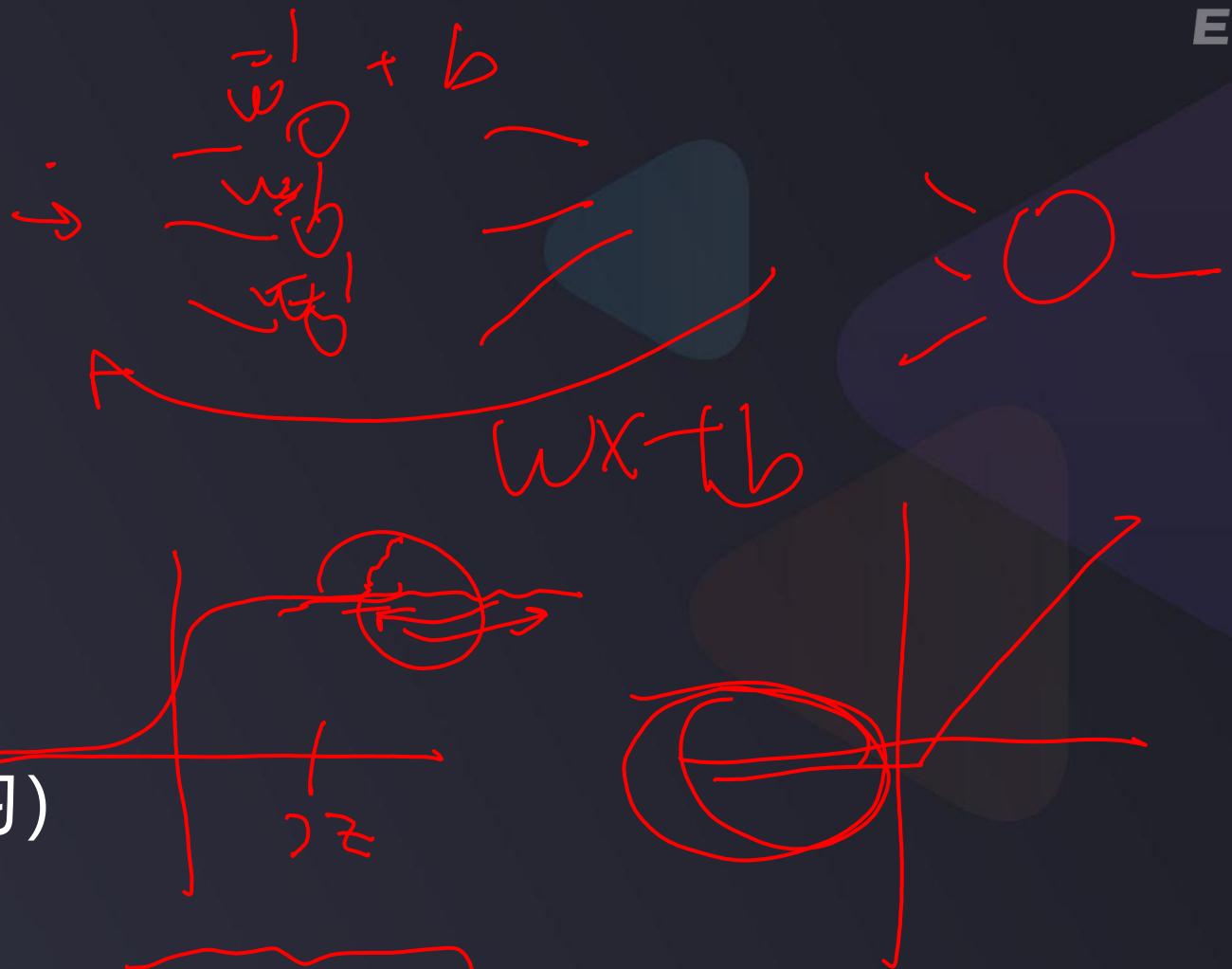


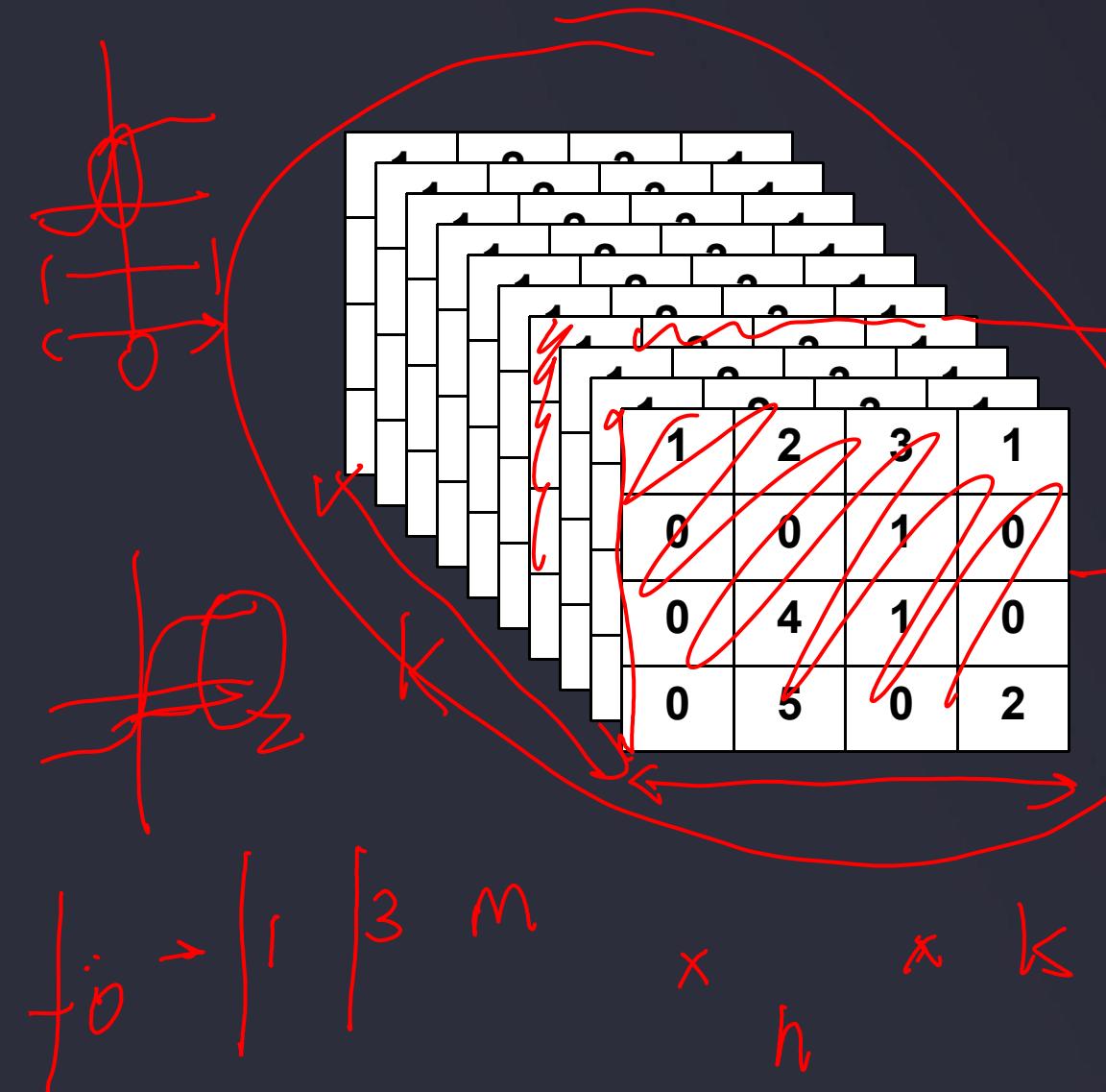
Diagram illustrating Xavier and He initialization:

The diagram shows a neural network structure with two layers. The first layer has  $n_j$  neurons, and the second layer has  $n_{j+1}$  neurons. The input to the first layer is  $x_1, x_2, x_3$ .

**Xavier Initialization:** The weight matrix  $W$  is initialized such that its elements follow a uniform distribution between  $- \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}$  and  $\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}$ .

**He Initialization:** The weight matrix  $W$  is initialized such that its elements follow a uniform distribution between  $- \frac{\sqrt{12}}{\sqrt{n_j + n_{j+1}}}$  and  $\frac{\sqrt{12}}{\sqrt{n_j + n_{j+1}}}$ .

# 批量规范化 (Batch Normalization)



- 对防止gradient vanish
- 解决了Internal Covariate Shift的问题,可以提高学习速率
- 减少了对于好的权重初始化的依赖
- 有助于解决overfitting

# Content

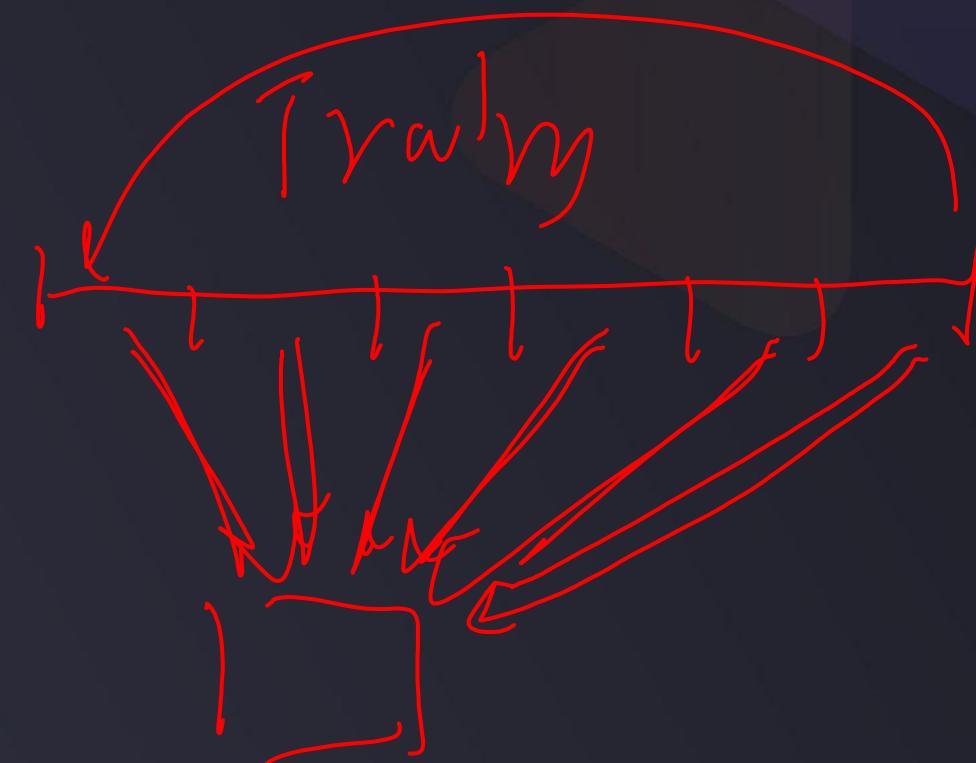
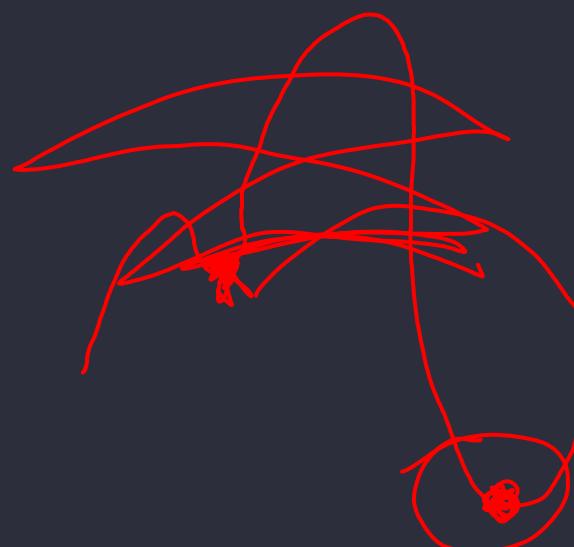
---

- 1 --模型优化：过拟合问题
- 2 --模型优化：梯度消失/爆炸问题
- 3 --模型优化：训练的优化
- 4 --模型优化：其他优化策略

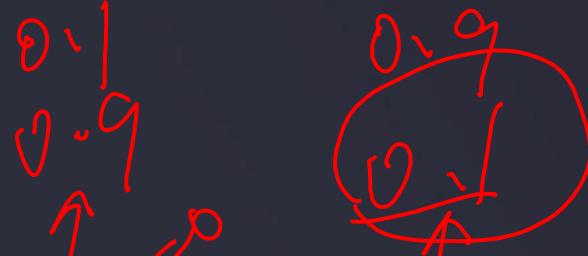
# |训练优化：批训练

1. 提高训练速度 ✓

2. 对训练过程引入随机性 ✓



# 训练优化：动量梯度下降法(gradient descent with momentum)优化器



$$\underline{v_w} = \beta v_w + (1 - \beta) dW$$

$$\underline{v_b} = \beta v_b + (1 - \beta) db$$

$$\underline{W = W - \alpha v_w} \quad \underline{b = b - \alpha v_b}$$

$$W - \alpha \underline{dW}$$



# |训练优化: RMSProp优化器

$$S_w = \beta S_w + (1 - \beta) dW^2$$

$$S_b = \beta S_b + (1 - \beta) db^2$$

$$W = W - \alpha \frac{dW}{\sqrt{(S_w)}}$$

$$b = b - \alpha \frac{db}{\sqrt{(S_b)}}$$



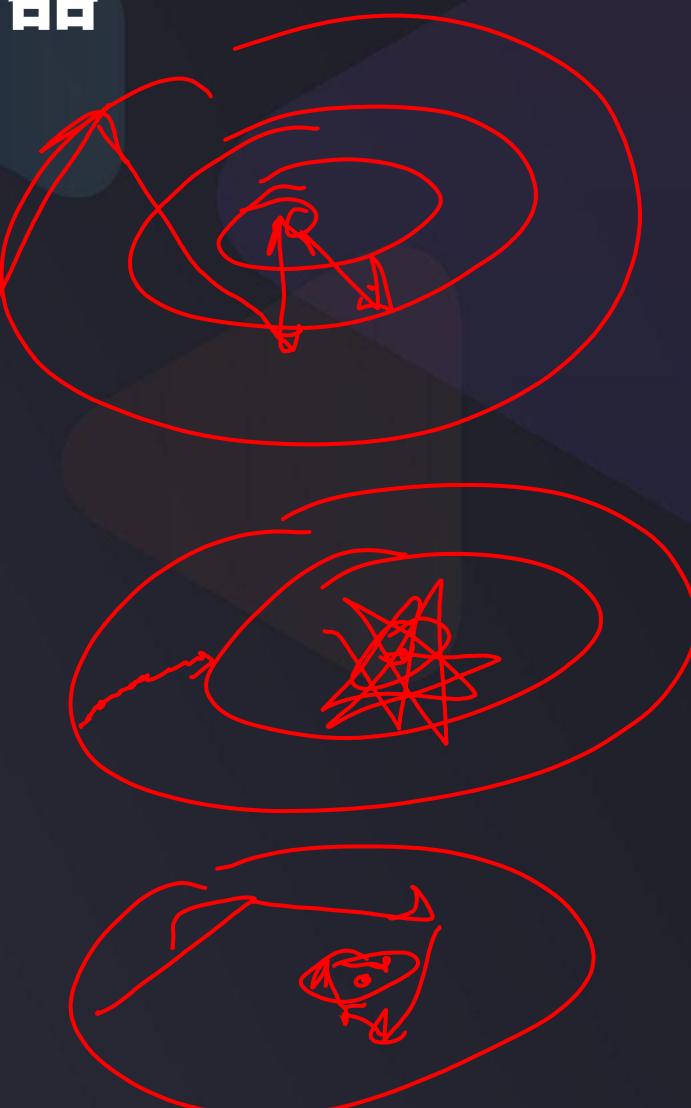
# 训练优化：自适应矩估计Adam优化器

$$\begin{cases}
 v_w = \beta_1 v_w + (1 - \beta_1) dW \\
 v_b = \beta_1 v_b + (1 - \beta_1) db \\
 S_w = \beta_2 S_w + (1 - \beta_2) dW^2 \\
 S_b = \beta_2 S_b + (1 - \beta_2) db^2
 \end{cases}$$

$v_w = v_w / (1 - \beta_1^t)$     $v_b = v_b / (1 - \beta_1^t)$   
 $s_w = s_w / (1 - \beta_2^t)$     $s_b = s_b / (1 - \beta_2^t)$

$$W = W - \alpha \frac{V_w}{\sqrt{(S_w) + \varepsilon}} \quad b = b - \alpha \frac{V_b}{\sqrt{(S_b) + \varepsilon}}$$

$10^8$     $10^9$

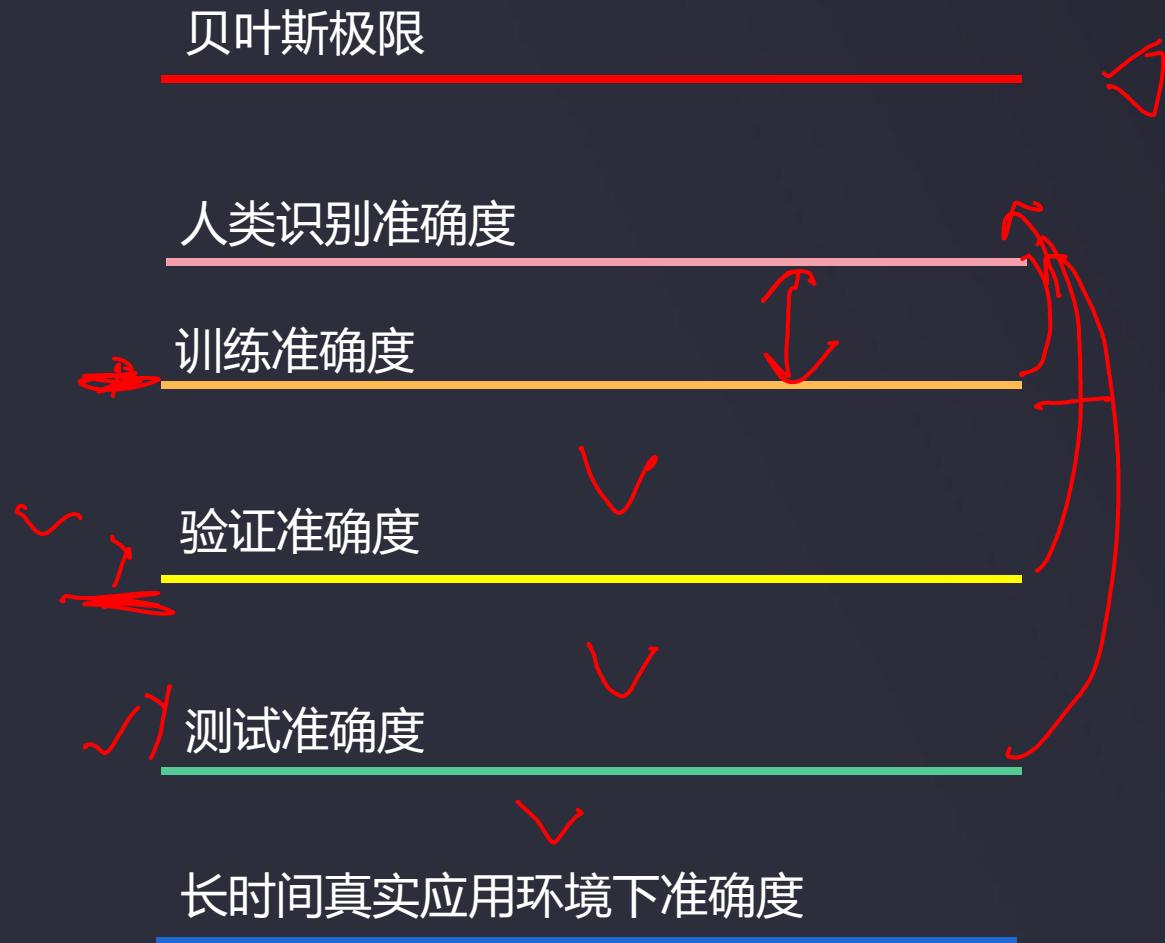


# Content

---

- 1 --模型优化：过拟合问题
- 2 --模型优化：梯度消失/爆炸问题
- 3 --模型优化：训练的优化
- 4 --模型优化：其他优化策略

# 贝叶斯极限



## 降低训练误差

1. 更复杂的模型
2. 更长时间的训练和优化
3. 更优的超参数

## 降低验证/测试误差

1. 更多更全面的数据
2. 解决过拟合的策略
3. 简化的模型结构/参数组合

# 满足指标和优化指标

>95% 达到要求  
准确率

准确率

| 模型   | 准确率 | 运算时间   |
|------|-----|--------|
| 分类器1 | 90% | 100ms  |
| 分类器2 | 95% | 110ms  |
| 分类器3 | 96% | 1000ms |

accuracy

X

efficiency

- 一个优化指标、

- 其它的满足指标、

≤25

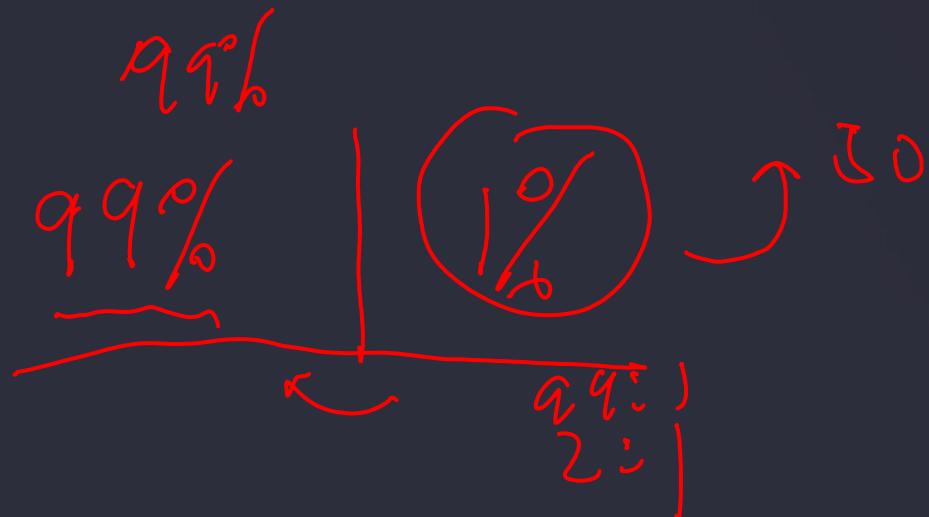
# 输出层的考量

- Linear
- Sigmoid
- Softmax
- One-hot-coding



# | 非对称数据训练和优化

- 数据增强扩大比例较低的样本的数量
- 修改损失函数赋予比例较低样本更高的权重



# 本章回顾

## ▷ 过拟合问题优化方法

1. 训练验证集划分/交叉验证
2. 扩大数据集和代表性/数据增强  
Data augmentation
3. 正则化Regularization
4. 随机失活Dropout

## 训练优化方法

1. 批训练
2. 优化器 (RMSprop, Adam)

## 梯度消失和爆炸优化方法

1. 残差神经网络连接方式
2. 非饱和的激活函数 (如 ReLU) ✓
3. 梯度截断 (Gradient Clipping) ✓
4. 好的参数初始化方式 ✓
5. 批量规范化 (Batch Normalization) ✓

## 其他优化策略

贝叶斯极限, 优化/满足指标, 输出层激活函数, 非对称数据



# 深度学习实战与计算视觉1：数值图像 处理与卷积神经网络分类任务

第七章：卷积神经网络的代码实现（文字识别OCR）

# |上一章回顾

卷积神经网络的优化

1. 过拟合问题

2. 梯度消失/爆炸问题

3. 训练的优化

4. 其他优化策略

# Content

---

OCR

- 1 --环境准备
- 2 --数据准备
- 3 --模型搭建和设定
- 4 --模型训练、测试、和存取

# Content

---

1 --环境准备

2 --数据准备

3 --模型搭建和设定

4 --模型训练、测试、和存取

# |环境准备



Google brain

安装使用

Keras.io/zh

# 环境准备-在命令行安装

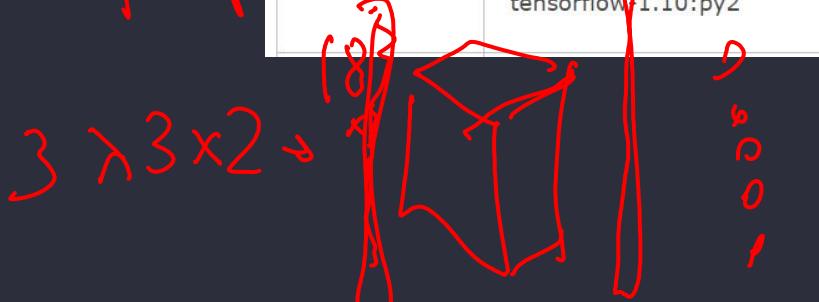
Tensor

pip ins

pip ins

Pip

| Framework       | Env name (--env parameter) | Description                                    | Docker Image        | Packages and Nvidia Settings               |
|-----------------|----------------------------|------------------------------------------------|---------------------|--------------------------------------------|
| TensorFlow 1.14 | tensorflow-1.14            | TensorFlow 1.14.0 + Keras 2.2.5 on Python 3.6. | floydhub/tensorflow | TensorFlow-1.14                            |
| TensorFlow 1.13 | tensorflow-1.13            | TensorFlow 1.13.0 + Keras 2.2.4 on Python 3.6. | floydhub/tensorflow | TensorFlow-1.13                            |
| TensorFlow 1.12 | tensorflow-1.12            | TensorFlow 1.12.0 + Keras 2.2.4 on Python 3.6. | floydhub/tensorflow | TensorFlow-1.12<br>import tensorflow as tf |
|                 | tensorflow-1.12:py2        | TensorFlow 1.12.0 + Keras 2.2.4 on Python 2.   | floydhub/tensorflow |                                            |
| TensorFlow 1.11 | tensorflow-1.11            | TensorFlow 1.11.0 + Keras 2.2.4 on Python 3.6. | floydhub/tensorflow | TensorFlow-1.11                            |
|                 | tensorflow-1.11:py2        | TensorFlow 1.11.0 + Keras 2.2.4 on Python 2.   | floydhub/tensorflow |                                            |
| TensorFlow 1.10 | tensorflow-1.10            | TensorFlow 1.10.0 + Keras 2.2.0 on Python 3.6. | floydhub/tensorflow | TensorFlow-1.10                            |
|                 | tensorflow-1.10:py2        | TensorFlow 1.10.0 + Keras 2.2.0 on Python 2.   | floydhub/tensorflow | Flatten,                                   |



from keras.layers import Conv2D, MaxPooling2D  
from keras import backend as K

# Content

---

- 1** --环境准备
- 2** --数据准备
- 3** --模型搭建和设定
- 4** --模型训练、测试、和存取

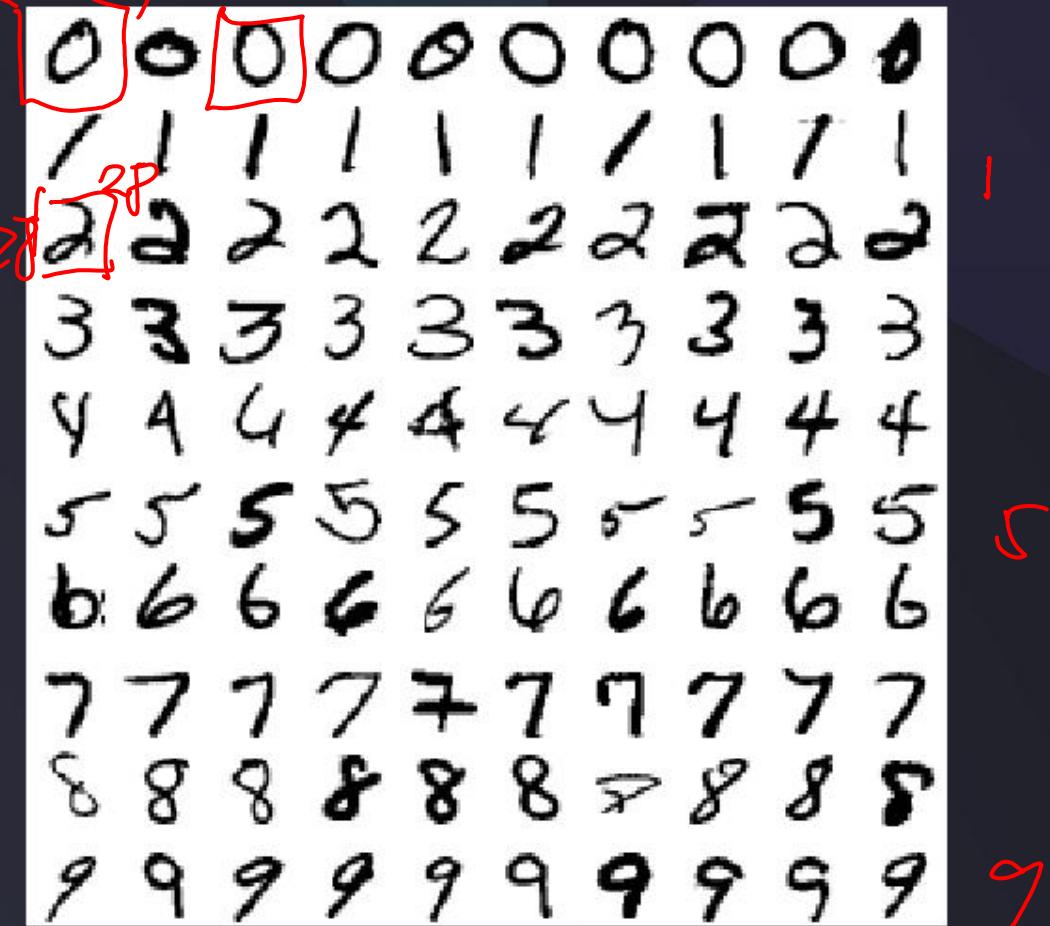
# MNIST数据集

0 ~ 9

训练数据60000张，尺寸 $28 \times 28 \times 1$

测试数据10000张，尺寸 $28 \times 28 \times 1$

Hello World



# MNIST数据预处理代码实现

#载入数据，如果没有会自动在网上下载

```
(X_train, Y_train), (X_test, Y_test) = keras.datasets.mnist.load_data()
```

#把输入整理成模型可以接受的格式

```
X_train = X_train.reshape(X_train.shape[0], 28, 28, 1)
```

```
X_test = X_test.reshape(X_test.shape[0], 28, 28, 1)
```

#对输入进行归一化处理

```
X_train = X_train / 255
```

```
X_test = X_test / 255
```

#对于输出进行one-hot-coding

```
Y_train = keras.utils.to_categorical(Y_train, 10)
```

```
Y_test = keras.utils.to_categorical(Y_test, 10)
```

$K \times 28 \times 28 \times 1$

$60000, 28, 28, 1$   
 $10000, 28, 28, 1$

$6000, 1$

$6000, 1$

$6000, 10$

$10000, 10$

$0 \sim 1$

$\Rightarrow$

{0  
1  
2  
3  
4  
5  
6  
7  
8  
9}

{1 0 0 0 0 0 0 0 0 0  
0 1 0 0 0 0 0 0 0 0  
0 0 1 0 0 0 0 0 0 0  
0 0 0 1 0 0 0 0 0 0  
0 0 0 0 1 0 0 0 0 0  
0 0 0 0 0 1 0 0 0 0  
0 0 0 0 0 0 1 0 0 0  
0 0 0 0 0 0 0 1 0 0  
0 0 0 0 0 0 0 0 1 0  
0 0 0 0 0 0 0 0 0 1}

$6000, 10$

# Content

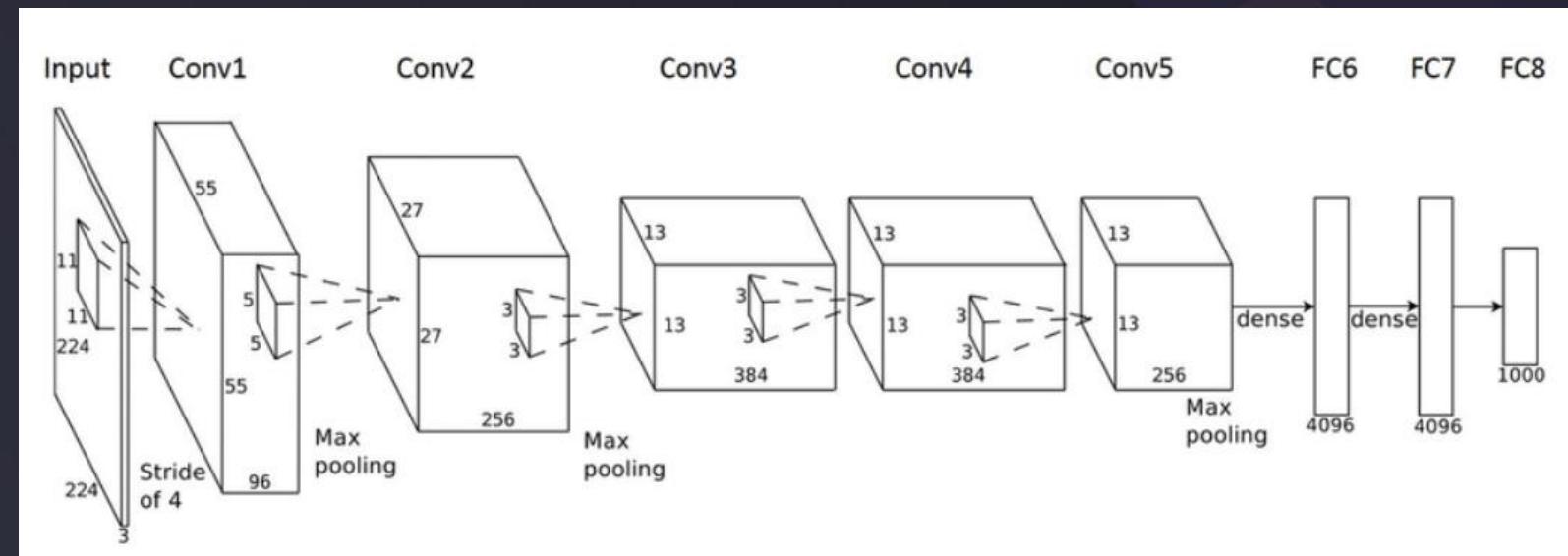
---

- 1** --环境准备
- 2** --数据准备
- 3** --模型搭建和设定
- 4** --模型训练、测试、和存取

# AlexNet

2012 Imagenet

20% → 15%



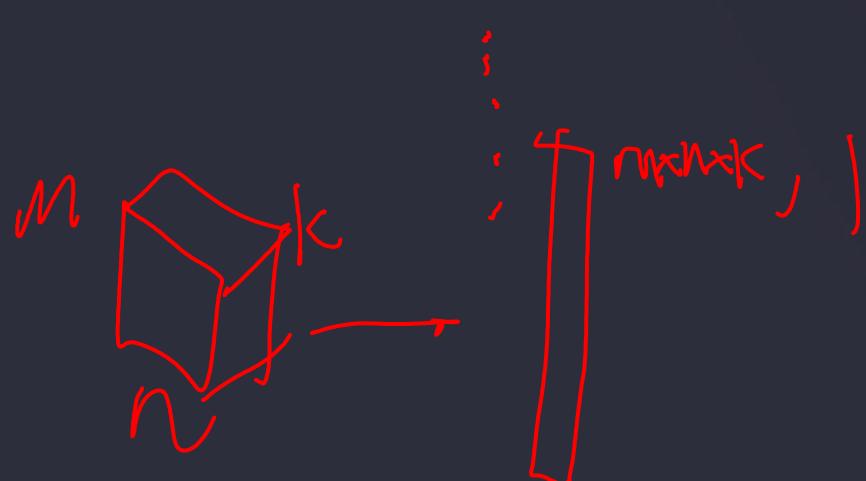
Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

# AlexNet代码实现

```

model = keras.models.Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(28, 28, 1)))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

```



```

model.add(Flatten())
model.add(Dense(200))
model.add(Activation('relu'))
model.add(Dense(200))
model.add(Activation('relu'))
model.add(Dense(10, activation='softmax'))

```

*PHC  
r Code/*

# | 模型损失函数和优化器的设定

# 定义模型的优化器

```
adam = keras.optimizers.Adam(lr=0.001, beta_1=0.9, beta_2=0.999,  
epsilon=1e-08)
```

# 编译模型，定义损失函数，优化器等

```
model.compile(loss='categorical_crossentropy', optimizer=adam,  
metrics=['accuracy'])
```

MSE  
MAE  
LO

Keras, io

# Content

---

- 1 --环境准备
- 2 --数据准备
- 3 --模型搭建和设定
- 4 --模型训练、测试、和存取

# | 模型训练测试存取等相关代码

```
#训练模型  
model.fit(X_train, Y_train, batch_size=100, epochs=1, verbose=1, validation_data=(X_test,  
Y_test))  
  
#展示训练好的模型在测试集图片上的表现效果  
test_index=2  
plt.imshow(X_test[test_index,:,:0])  
output=model.predict(X_test[test_index,:,:,:].reshape([1,28,28,1]))  
  
#模型的存储  
model.save("model/My_Alexnet_mnist.h5")  
  
#载入预先训练好的模型的  
from keras.models import load_model  
model=load_model("model/My_Alexnet_mnist.h5")  
model.summary()
```

# 本期回顾

- 1 --环境准备
- 2 --数据准备
- 3 --模型搭建和设定
- 4 --模型训练、测试、和存取

Mnist

Transfer Learning



# 深度学习实战与计算视觉1：数值图像 处理与卷积神经网络分类任务

第八章：迁移学习

A simple red wavy line graphic is located below the chapter title, centered horizontally.

# | 上一章回顾

搭建一个卷积神经网络并进行OCR文字识别

--环境准备

--数据准备

--模型搭建和设定

--模型训练、测试、和存取



Wi SJ

# Content

---

- 1 --迁移学习
- 2 --数据与模型
- 3 --相关实战代码

# Content

---

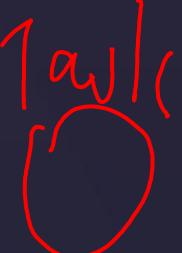
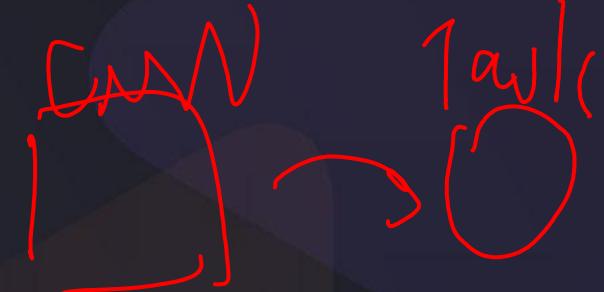
1 --迁移学习

2 --数据与模型

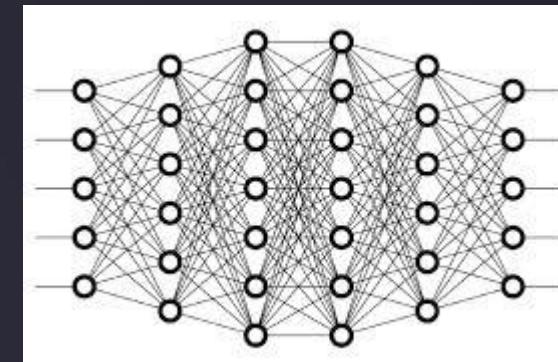
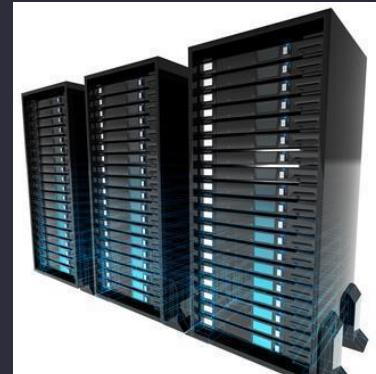
3 --相关实战代码

# 迁移学习

Random → CNN



ImageNet Coco



# |迁移学习的三种方法

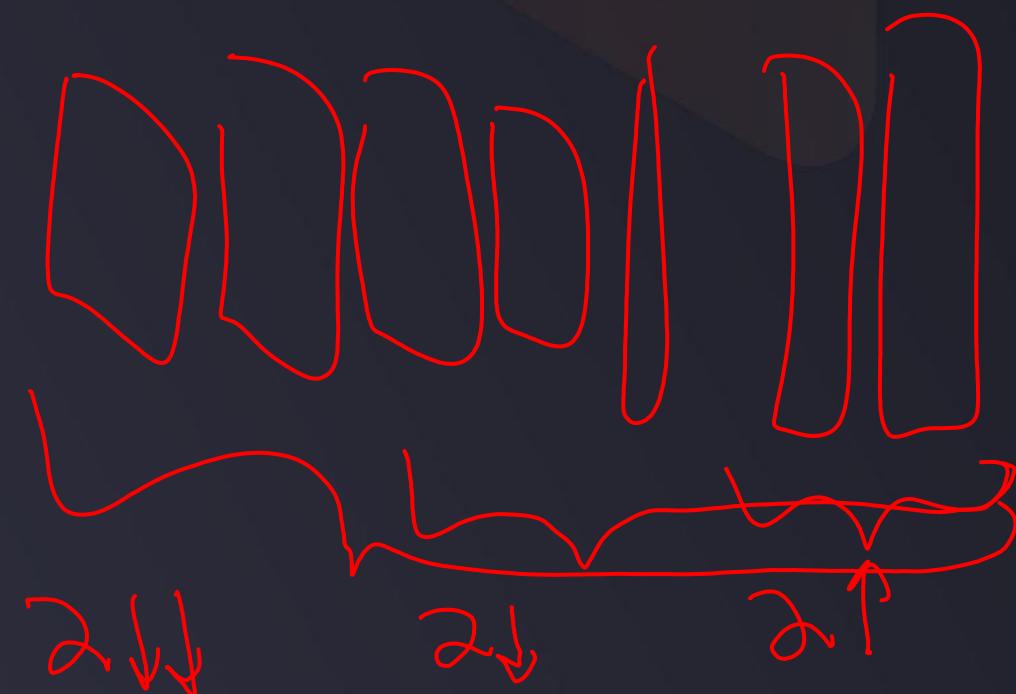
1. 只训练分类器



2. 全网络微调 FINE-TUNING



3. 两者结合



# Content

---

1 --迁移学习

2 --数据与模型



3 --相关实战代码

# | 数据

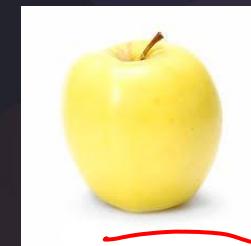
250

2010

IL SVRC



1000

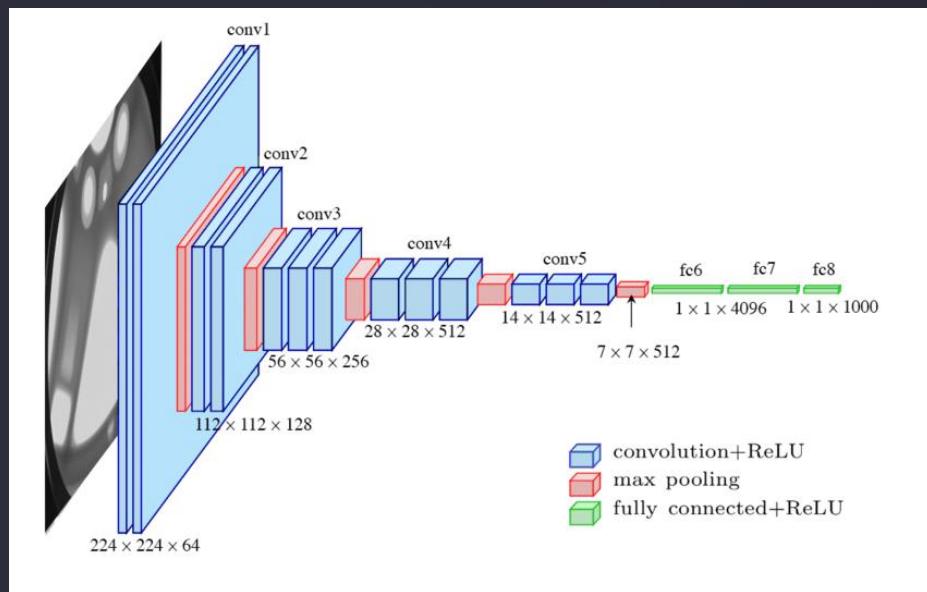


1400万

标注

92

# 模型架构



VGG-16  
-19  
ICLR2015

$3 \times 3$   
 $[x]$   
140M

Visual Geometry Group  
2014

<https://arxiv.org/abs/1409.1556>

# Content

---

1 --迁移学习

2 --数据与模型

3 --相关实战代码

# 数据相关实现代码

OCR MNIST 28x28x1

92

```
from keras.preprocessing.image import ImageDataGenerator
```

1000x1000

#做数据增强的指令

```
train_datagen = ImageDataGenerator(rescale=1./255,  
rotation_range=45, width_shift_range=0.2, height_shift_range=0.2,  
shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
```

0~255  
0~1

#其他进行数据预处理的参数

```
train_gen = train_datagen.flow_from_directory("data/train",  
target_size=(240,240), batch_size=8, class_mode='categorical')  
Next(train_gen)
```

# | 模型准备相关实战代码

#载入网上预先训练好的模型，注意设置好输入图片大小

```
VGG = keras.applications.vgg16.VGG16(weights='imagenet', include_top=False,  
input_shape=[ 240, 240, 3])  
VGG.summary()
```

ResNet Inception MobileNet

#如果网速太慢，用我们为大家提前下载好的模型，记得要放到定义好的路径下

```
from keras.models import load_model  
VGG=load_model("model/VGG_Pretrained.h5")  
VGG.summary()
```

# 设置进行迁移学习的分类器

#加载一些必要的库

```
from keras.layers import Dense, Flatten
```

#在模型后面加入分类器，注意输出层神经元个数要和标签一致，一定要加入

flatten，不然dense layer无法接受矩阵输入

```
x=VGG.layers[-1].output
```

```
x=Flatten()(x)
```

```
x=Dense(64,activation="sigmoid")(x)
```

```
predictions=Dense(5,activation="softmax")(x)
```

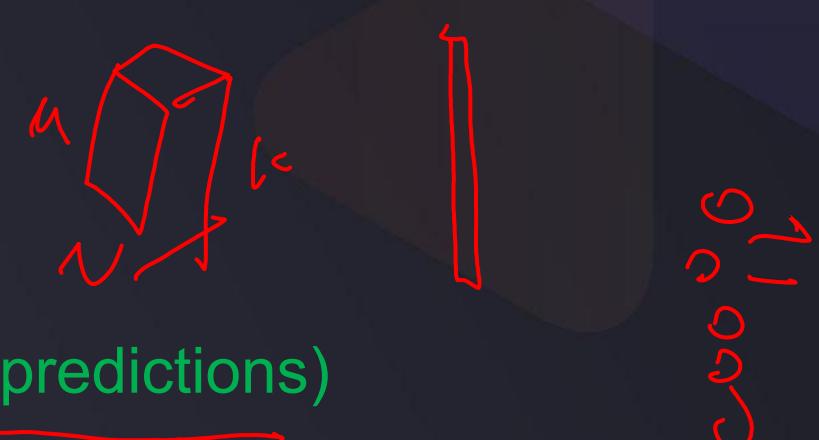
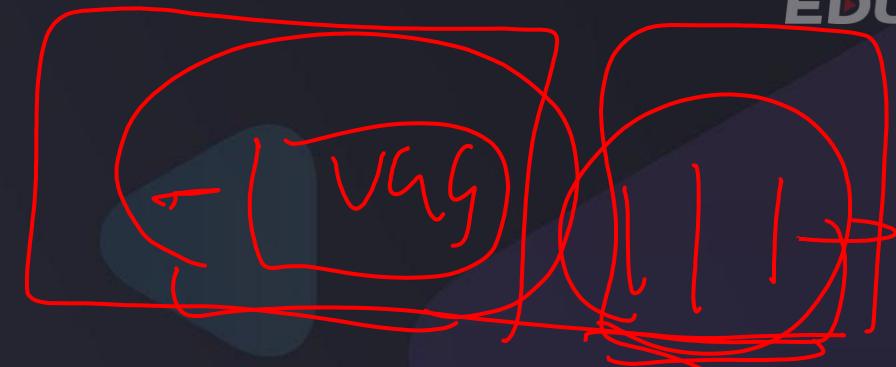
```
model=keras.Model(inputs=VGG.inputs,outputs=predictions)
```

```
model.summary()
```

#预先设置好的层将被固定不参与训练，因此可训练参数只包含最后的两层

```
for layer in model.layers[:-2]:
```

```
    layer.trainable = False
```



# |训练模型和测试模型

90+

#训练模型

model.fit\_generator(train\_gen, steps\_per\_epoch=12, epochs=10)

测试结果

output=model.predict(img.reshape([1,240,240,3]))

print(output)

output\_r=np.where(output==np.max(output))

print("predicted output is: ", np.max(output\_r[1]) )

# 本期回顾

1 --迁移学习

2 --数据与模型

3 --相关实战代码



# 深度学习实战与计算视觉1：数值图像 处理与卷积神经网络分类任务

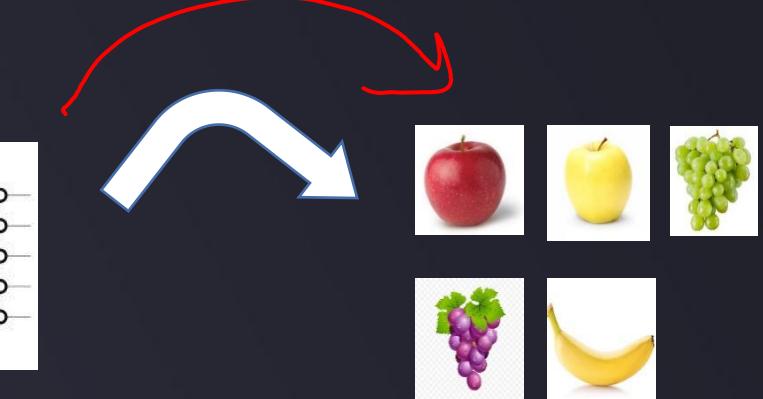
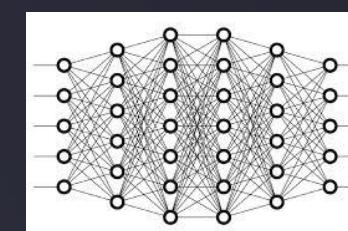
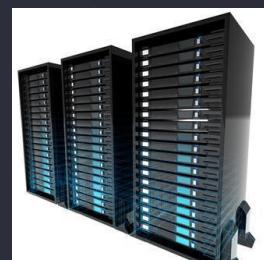
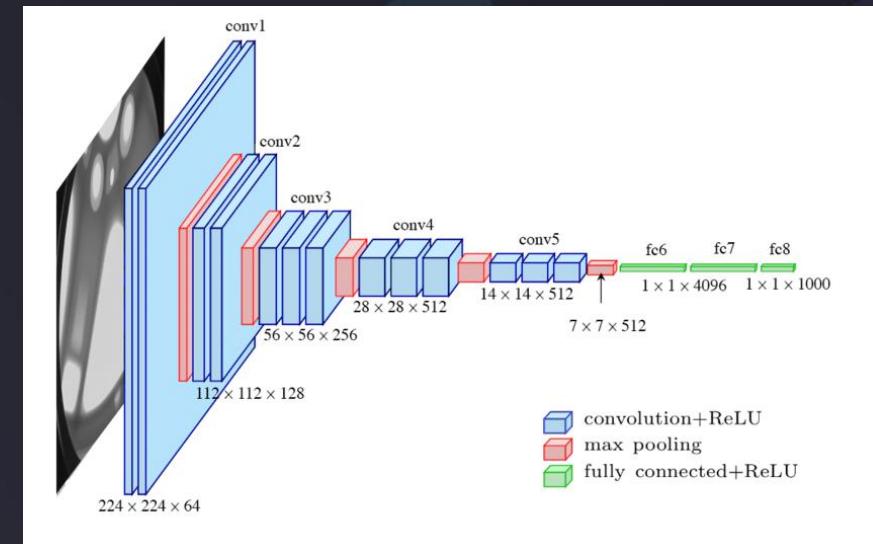
第九章：多标签深度学习分类任务工程

# | 上一章回顾

1 --迁移学习

2 --数据与模型

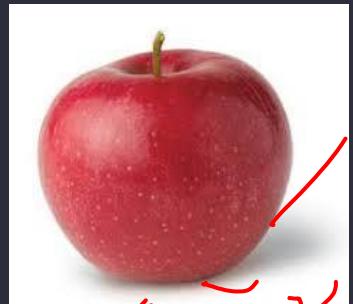
3 --相关实战代码



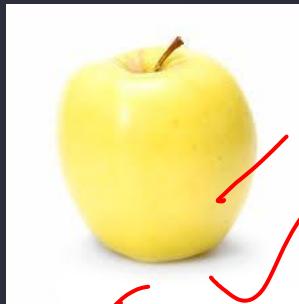
# 一个关于监督式学习的难题

监督学习问题

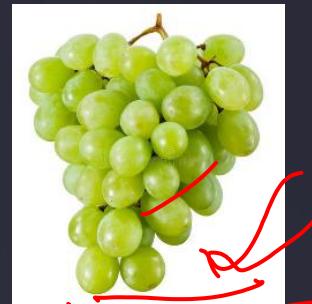
类别1



类别2



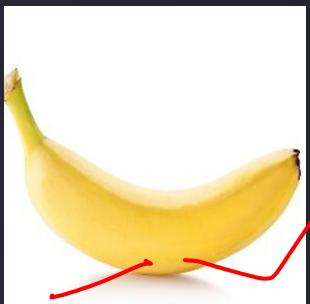
类别3



类别4



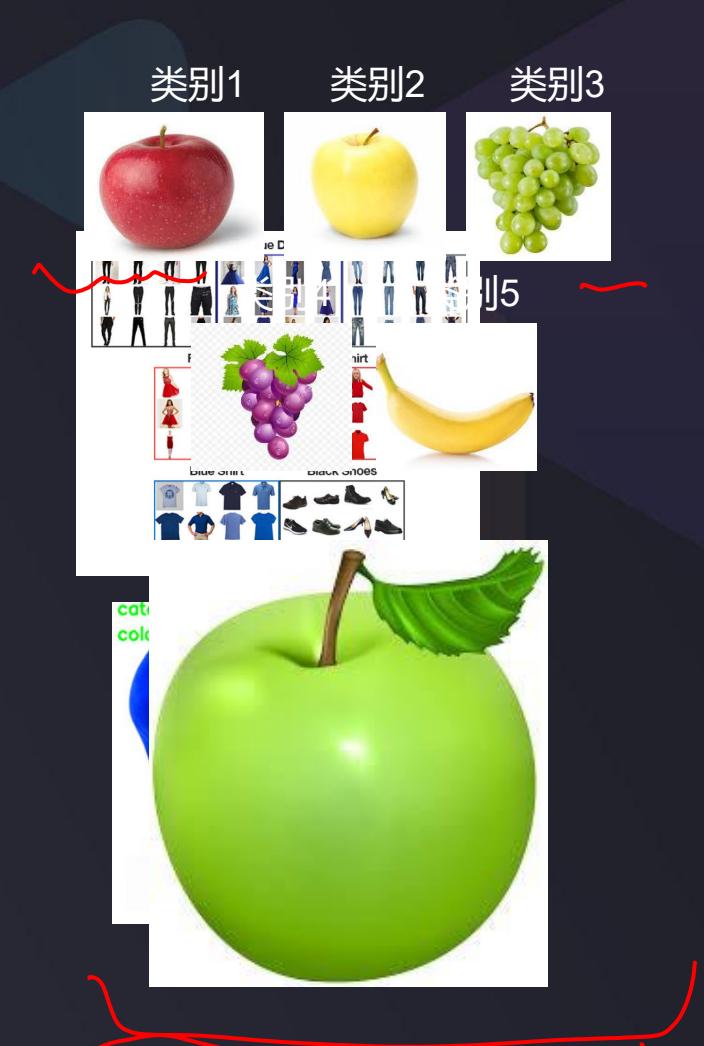
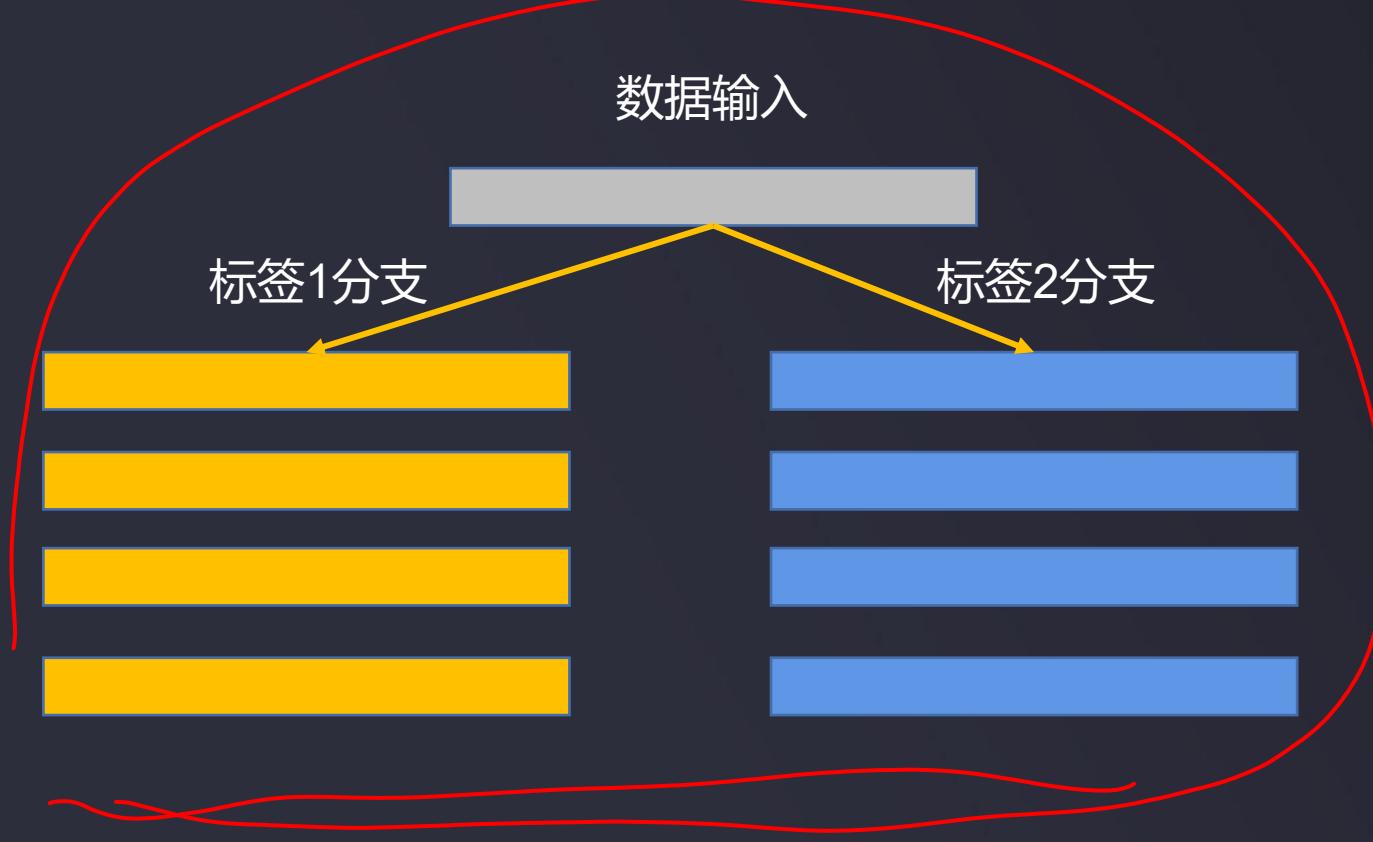
类别5



类别?

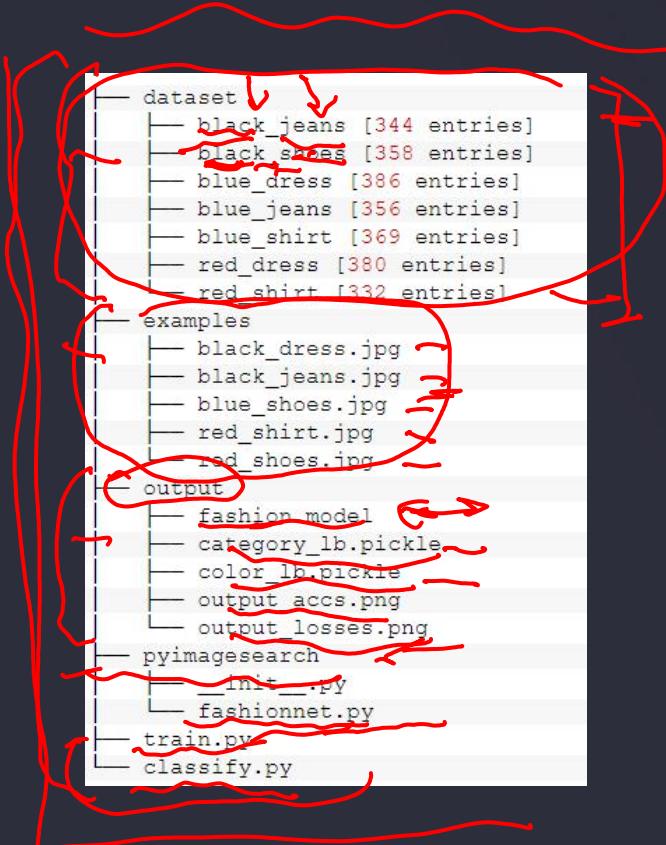


# 多标签分类任务-FashionNet



{ Adrian Rosebrock  
<https://www.pyimagesearch.com/2018/06/04/keras-multiple-outputs-and-multiple-losses/#>

# 多标签分类任务-工程文件



FashionNet工程需要用到的模组

pip install scikit-learn

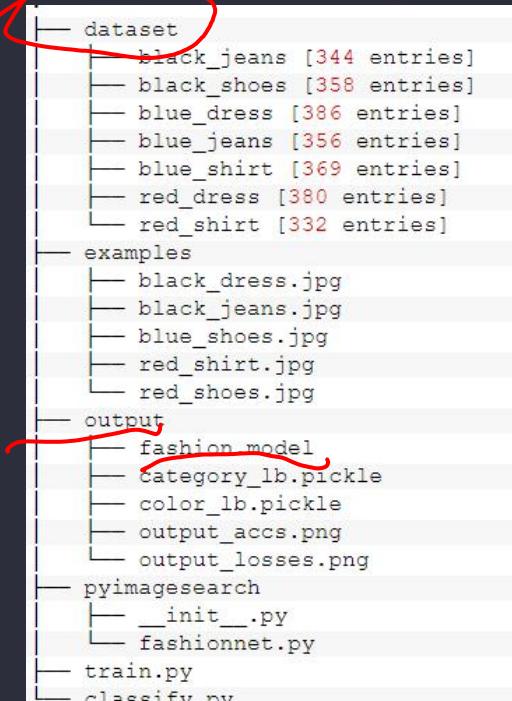
pip install imutils

在命令行或者Jupyter Notebook里面执行代码

`python train.py --dataset dataset --model  
output/fashion.model`

`%run train.py --dataset dataset --model  
output/fashion.model`

# |多标签分类任务-工程文件



FashionNet工程需要用到的模组

pip install scikit-learn

pip install imutils

*Scikit-Image*

在命令行或者Jupyter Notebook里面执行代码

→ python train.py --dataset dataset --model  
output/fashion.model

%run train.py --dataset dataset --model  
output/fashion.model

过了若干时间后。 . . .

# 本期回顾

---

- 1 --多标签分类任务
- 2 --FashionNet/FruitNet
- 3 --深度学习项目工程

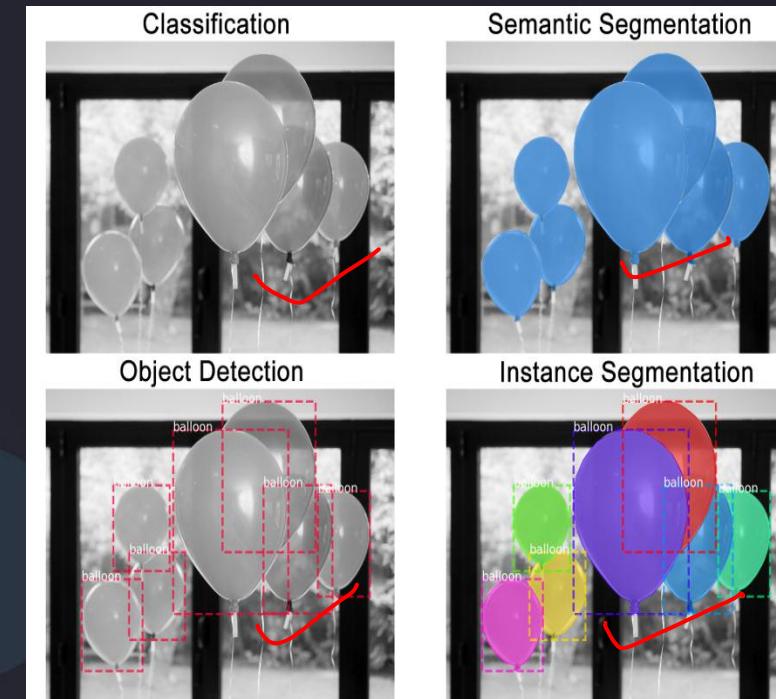
# 课程回顾：深度学习与计算视觉的历史，定义和应用

## 计算视觉四大主流任务

- 1. 图像分类/识别
- 2. 语义分割
- 3. 目标检测
- 4. 实例分割

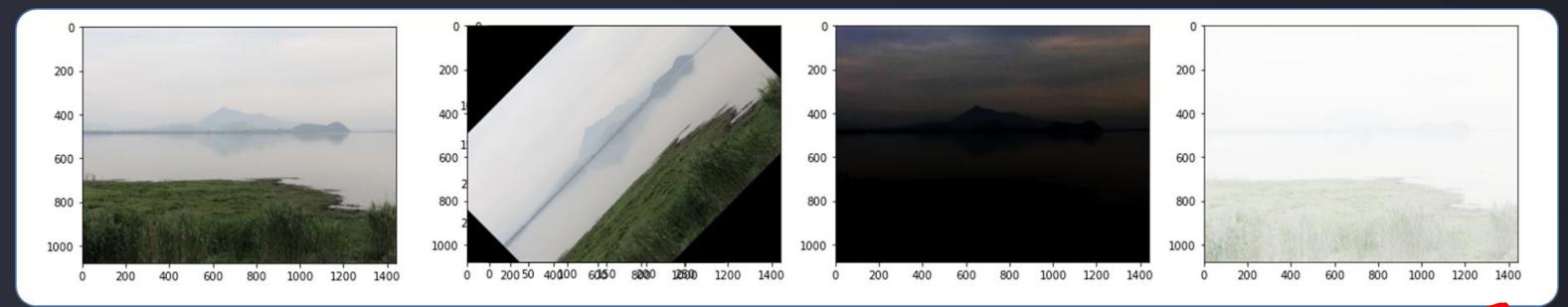
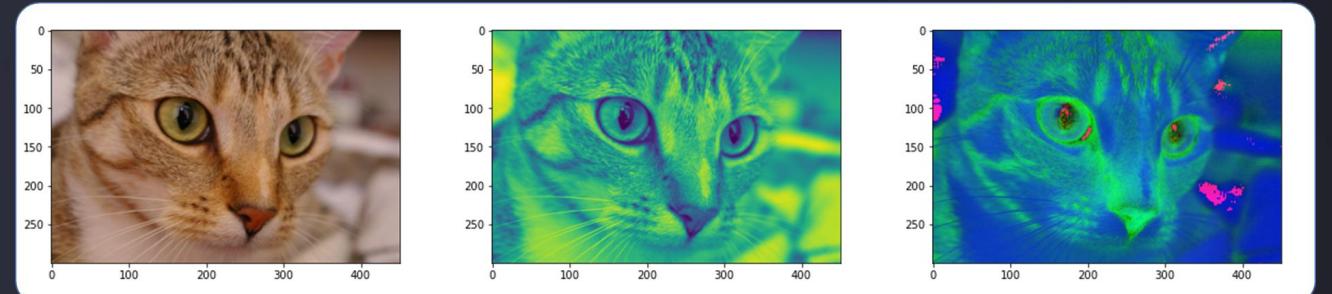
## 其他任务

- 1. 图片增强
- 2. 目标追踪
- 3. 视觉创意



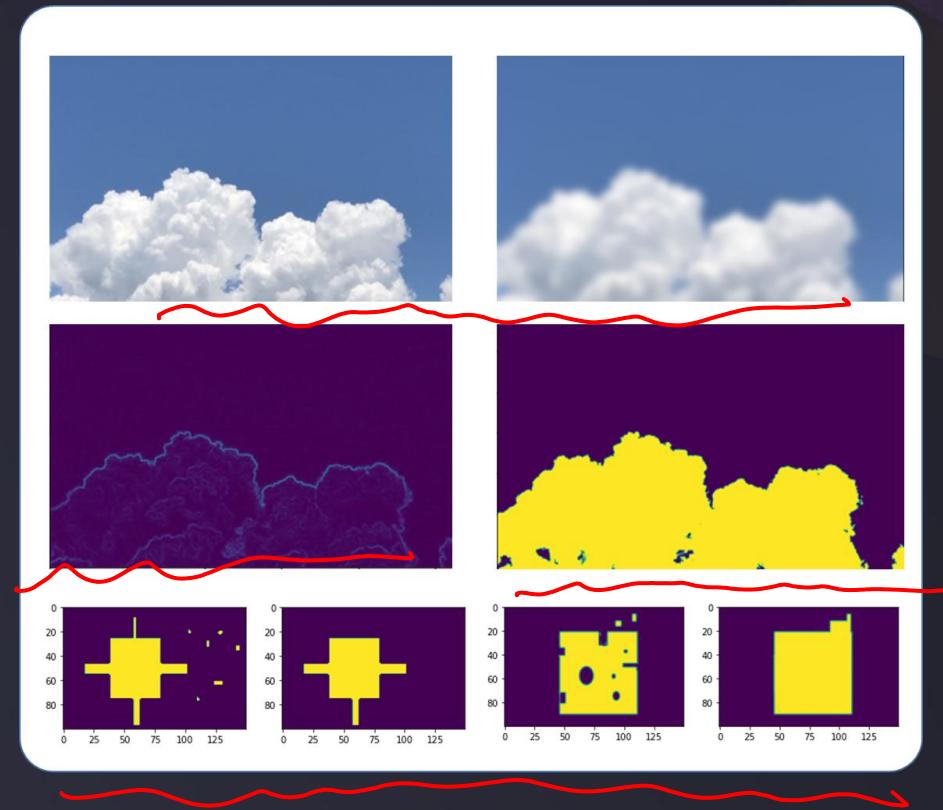
# 课程回顾

## 基础的数值图像处理



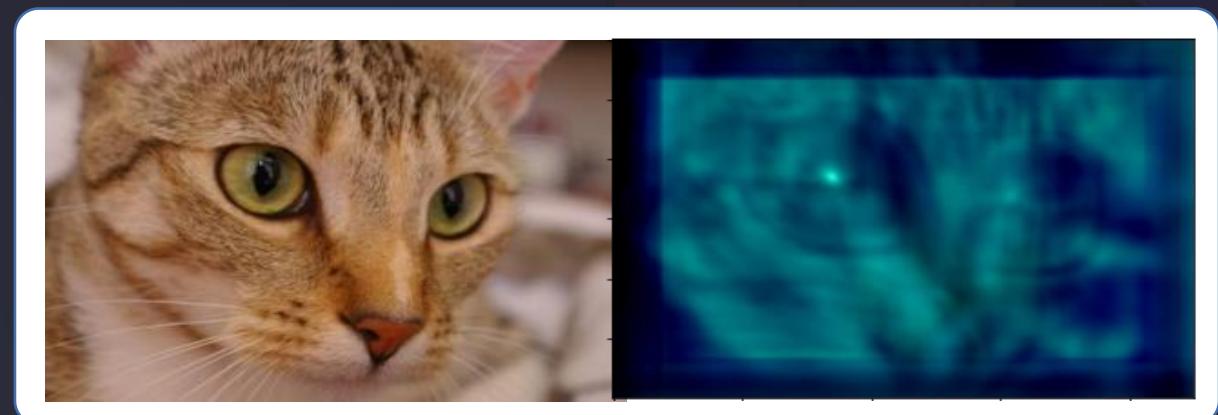
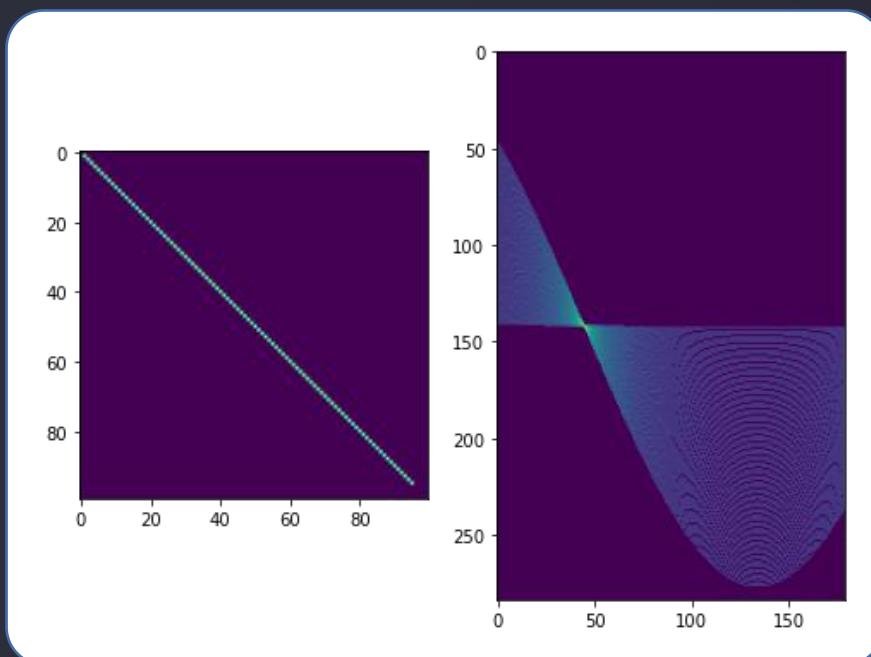
# 课程回顾

数值图像处理的滤波，分割，基本形态学

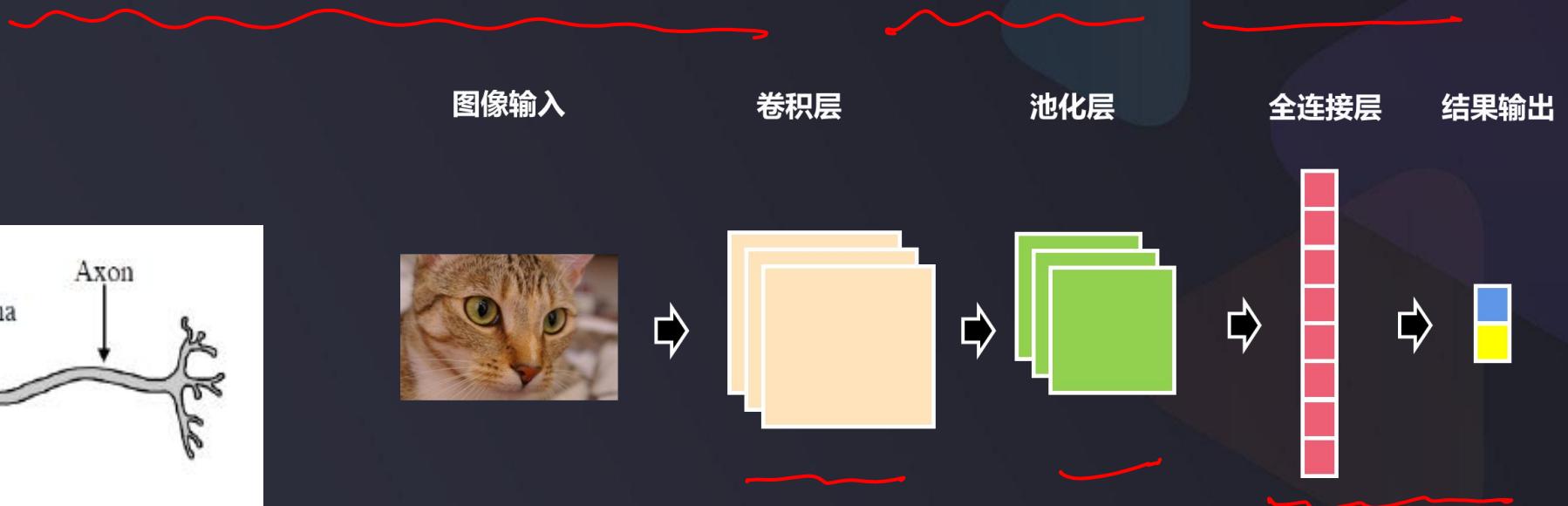
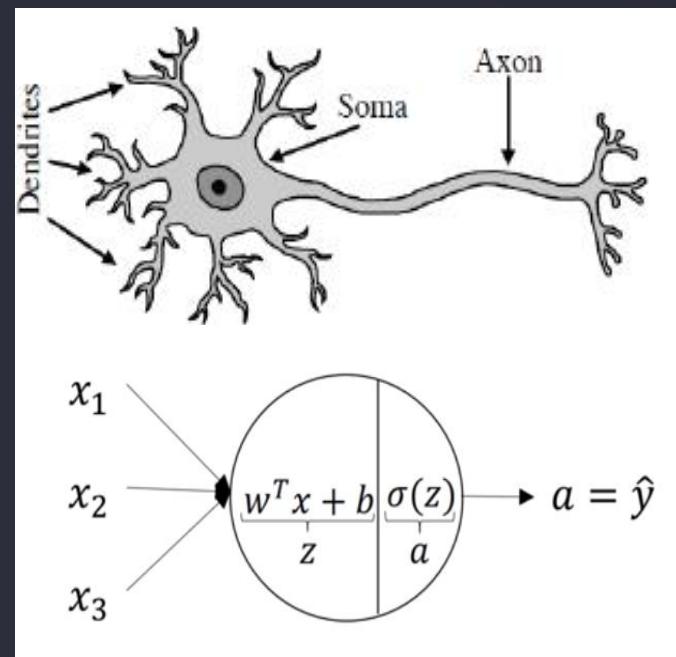


# 课程回顾

## 经典图像算法：霍夫变换和模板匹配

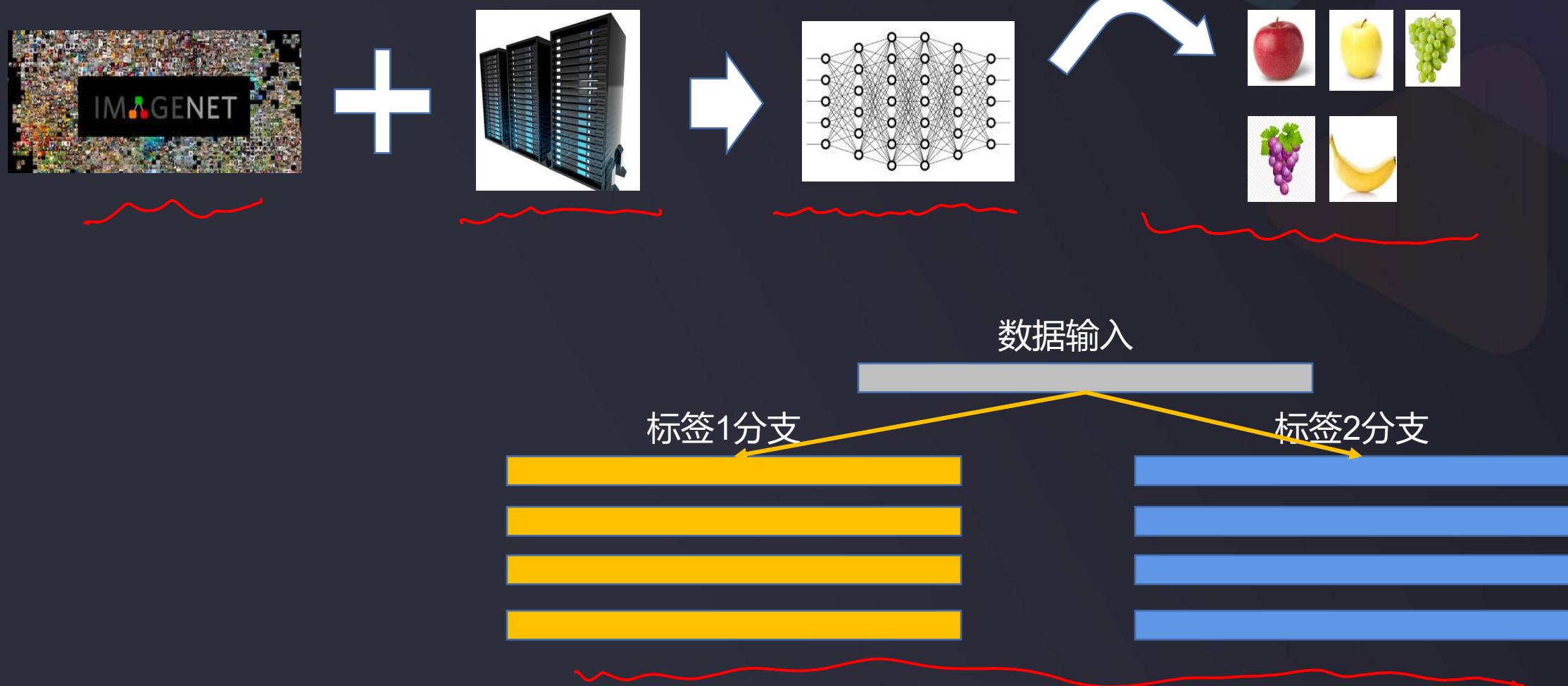


# 课程回顾：卷积神经网络的基本原理，经典框架和优化策略



- 模型优化：过拟合问题
- 模型优化：梯度消失/爆炸问题
- 模型优化：训练的优化
- 模型优化：其他优化策略

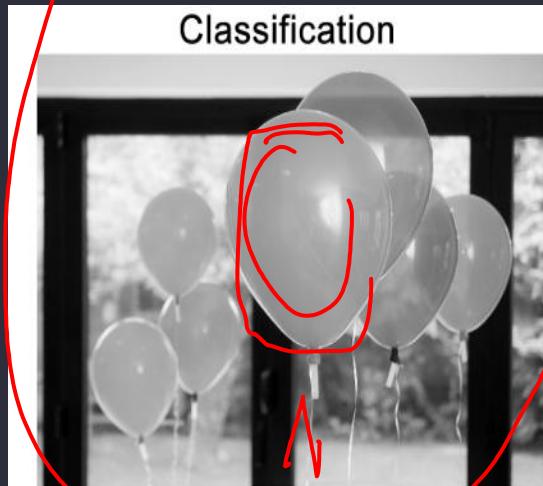
# 课程回顾：迁移学习和多标签分类任务



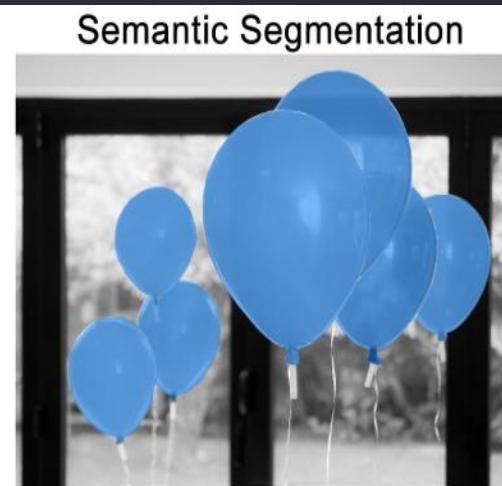
# 未来展望



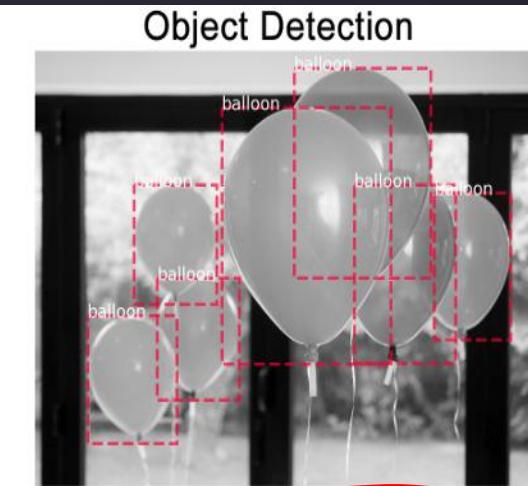
## 分类



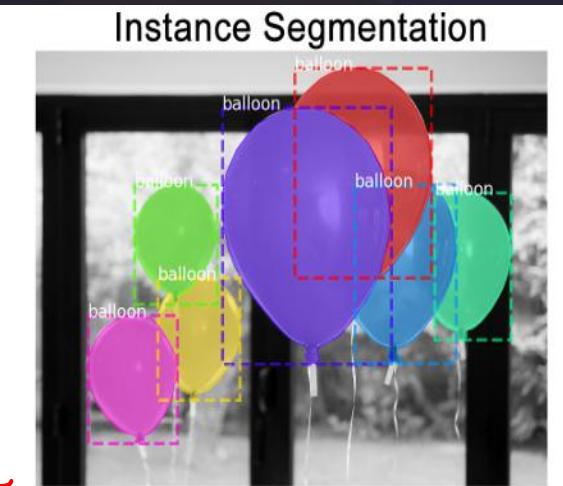
## 分割



## 目标检测



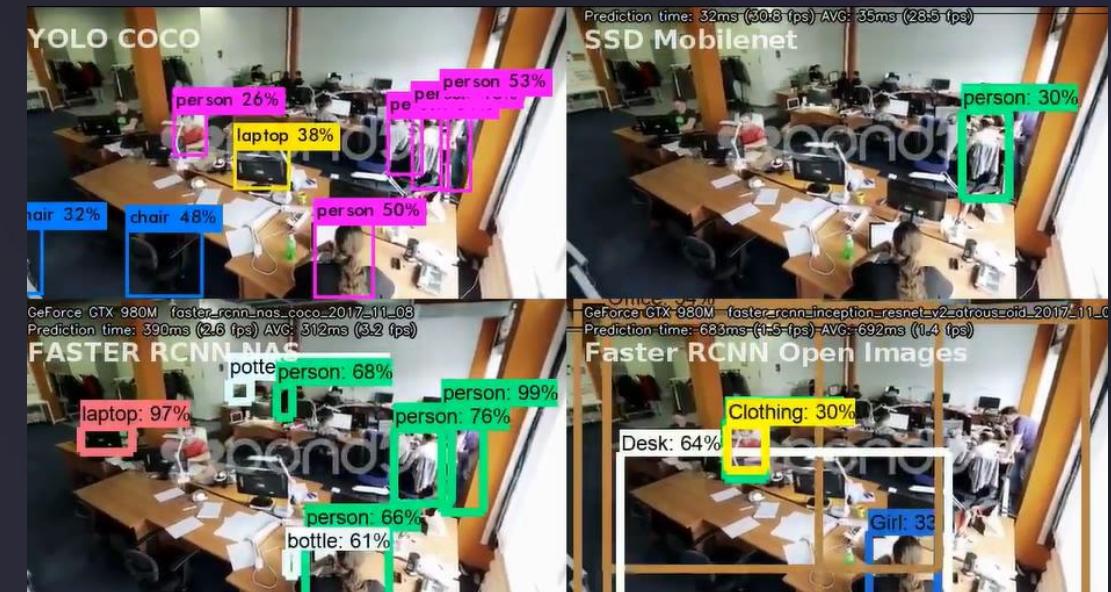
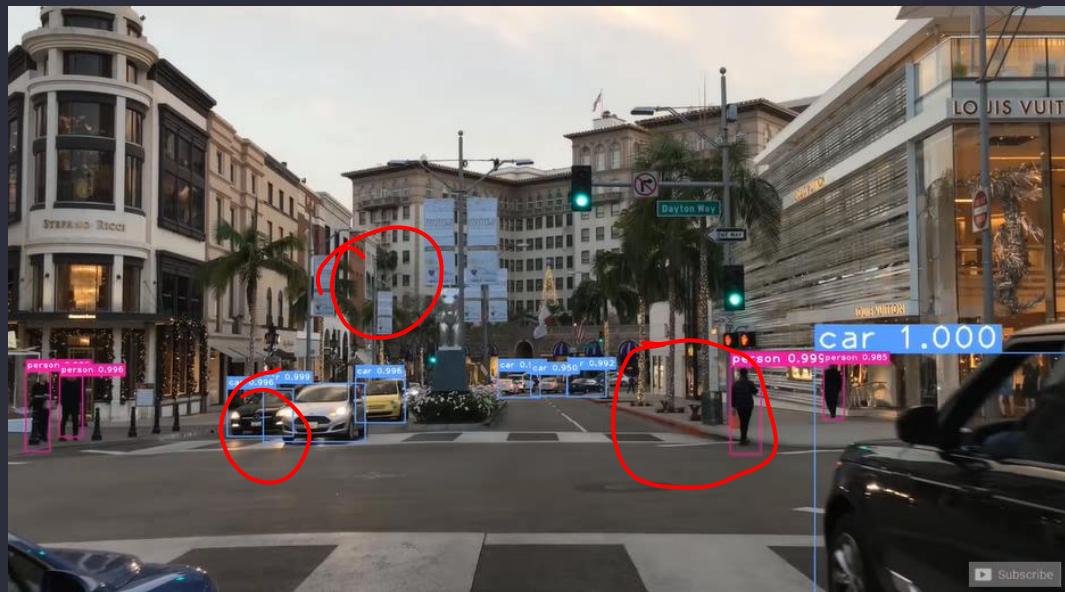
## 实例分割



从-50

→

# 未来展望



谢谢大家！