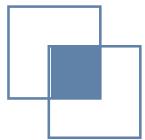


飞桨



人工智能概述与入门基础

中国科学院大学 AI小鸭团队



目录

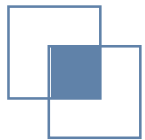


人工智能概述

深度学习入门基础

Python基础语法

作业：在线体验与Python练习



人工智能无处不在

飞桨



智能家居



智能汽车



智能手机



智能手环



智能手表



智能交通



智能教育



智慧金融



智慧城市



智能医疗



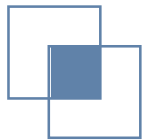
智能旅游



互联网

AI+生活

AI+行业



人工智能无处不在



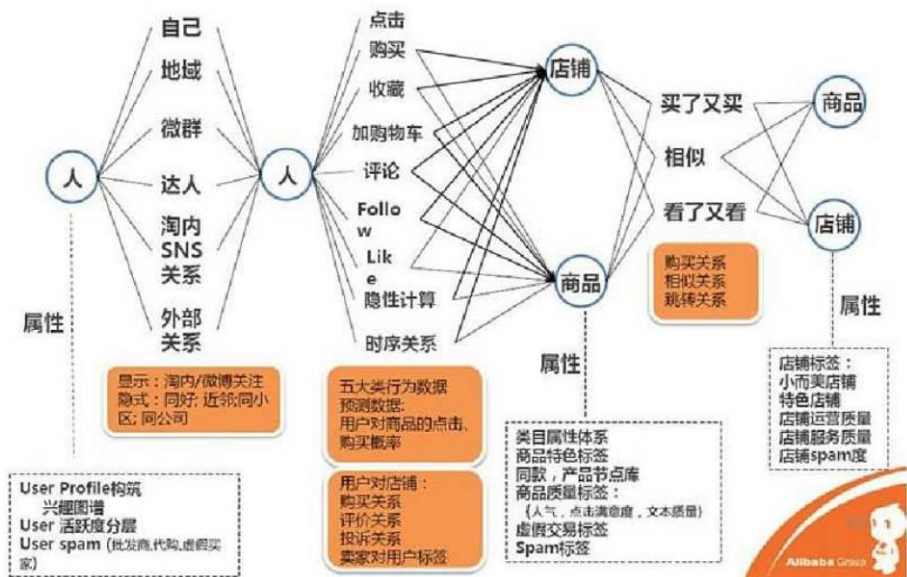
搜索引擎：网页、图片、视频、新闻、学术、地图

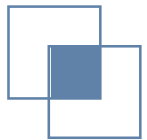
信息推荐：新闻、商品、游戏、书籍

图片识别：人像、用品、动物、交通工具

用户分析：社交网络、影评、商品评论

机器翻译、摘要生成.....





什么是人工智能



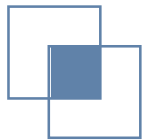
人工智能 (Artificial Intelligence)，英文缩写为**AI**。它是研究、开发用于**模拟、延伸和扩展人的智能**的理论、方法、技术及应用系统的一门新技术科学。

- ✓ 企图了解智能的实质，并生产出一种新的能以**人类智能相似的方式做出反应**的智能机器，相关研究包括机器人、语言识别、图像识别、自然语言处理和专家系统等。
- ✓ 人工智能是对人的意识、思维的信息过程的模拟。人工智能不是人的智能，但能像人那样思考、也可能超过人的智能。

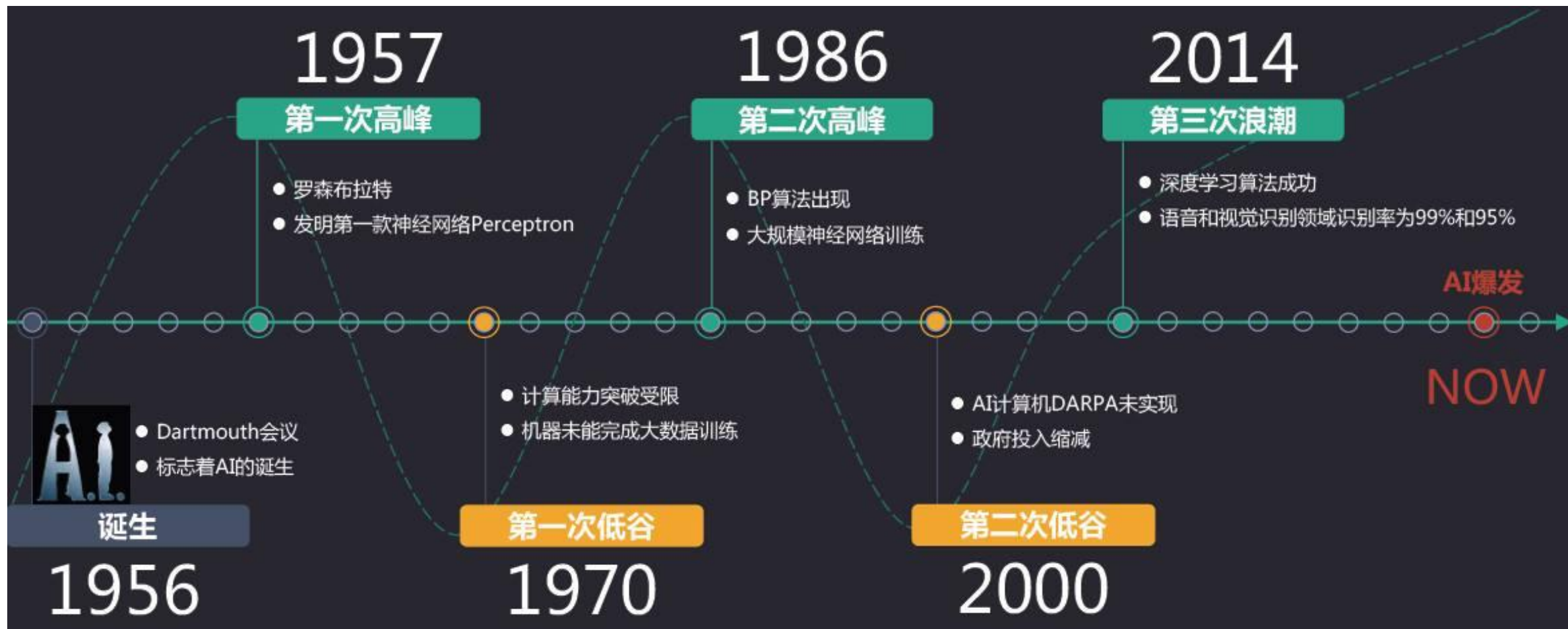


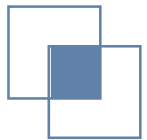
人工智能

使一部机器的反应方式像人一样进行**感知、认知、决策、执行**的人工程序或系统



人工智能发展历程





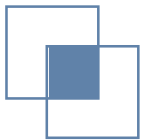
人工智能与机器学习、深度学习关系



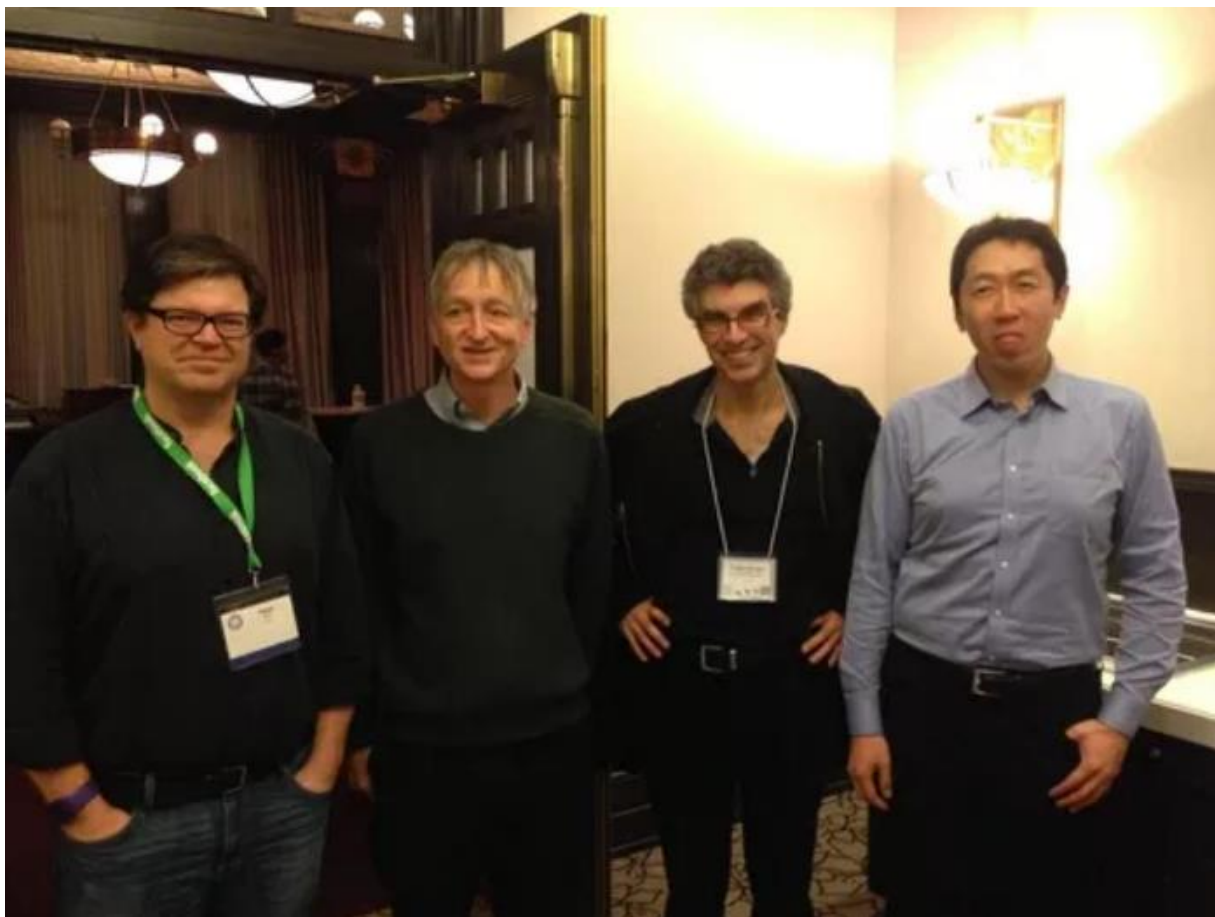
机器学习：一种实现人工智能的方法

深度学习：一种实现机器学习的技术，人工神经网络是机器学习中的一个重要的算法“深度”就是说神经网络中众多的层。





深度学习大牛



ACM（国际计算机学会）宣布，有“深度学习三巨头”之称的Yoshua Bengio、Yann LeCun、Geoffrey Hinton共同获得了2018年的图灵奖，这是图灵奖1966年建立以来少有的~~一年~~一年颁奖给三位获奖者。



Juergen Schmidhuber
Swiss AI Lab & NNAISENSE



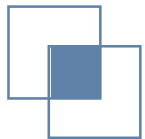
Richard Socher
Founder, CEO, CTO of MetaMind

Yann LeCun
NYU & FaceBook

Geoff Hinton
Tornoto U & Google

Yoshua Bengio
Montreal U

Andrew Ng
Stanford & Baidu



深度学习大玩家

飞桨

facebook



YAHOO!

Google

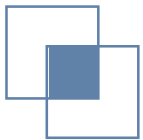


IBM



NVIDIA®

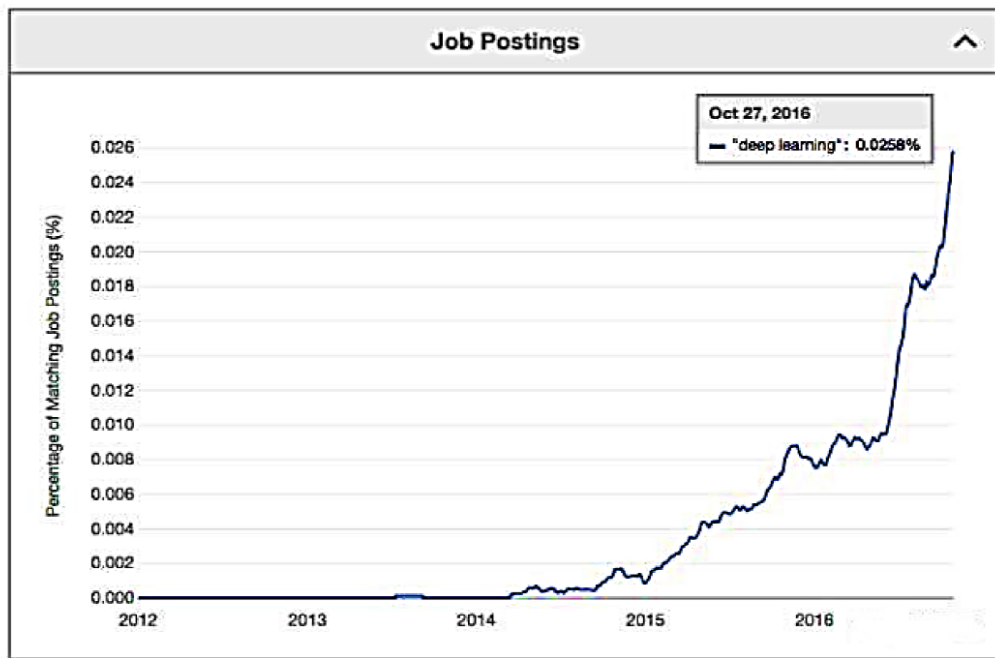
Baidu 百度



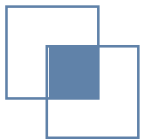
深度学习就业情况



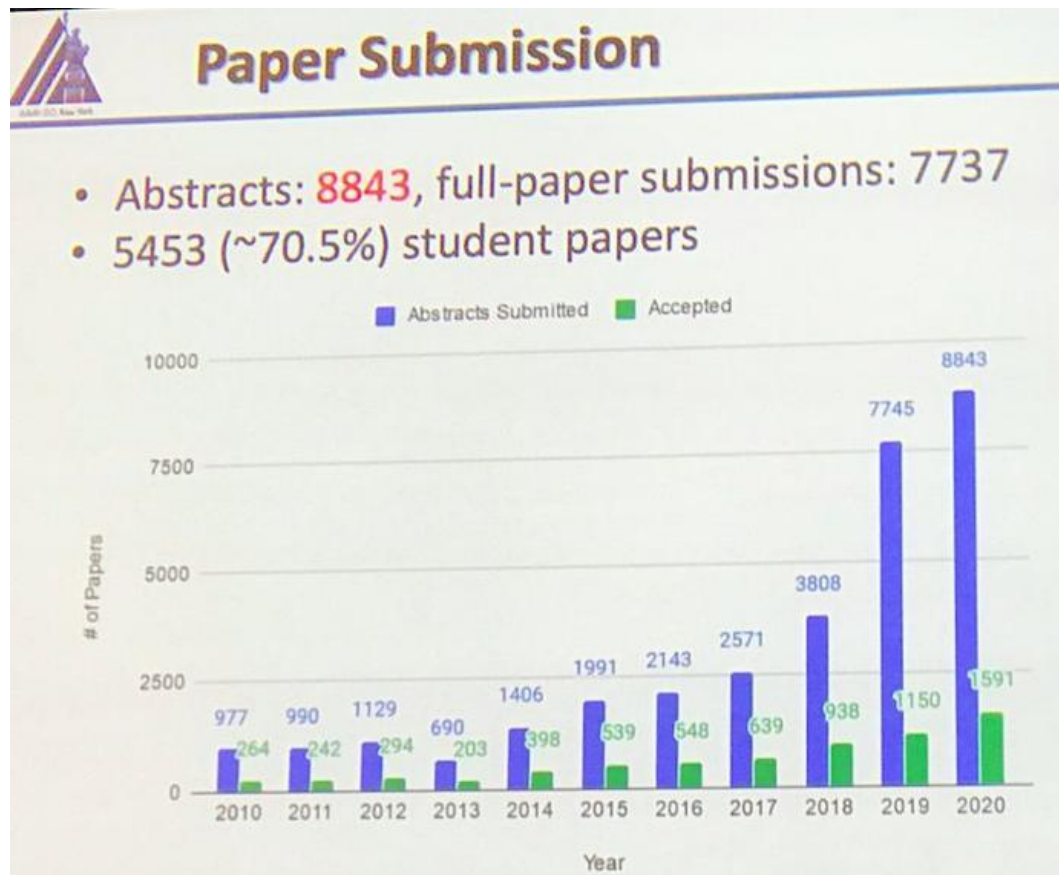
DL人才需求量的指数级增长



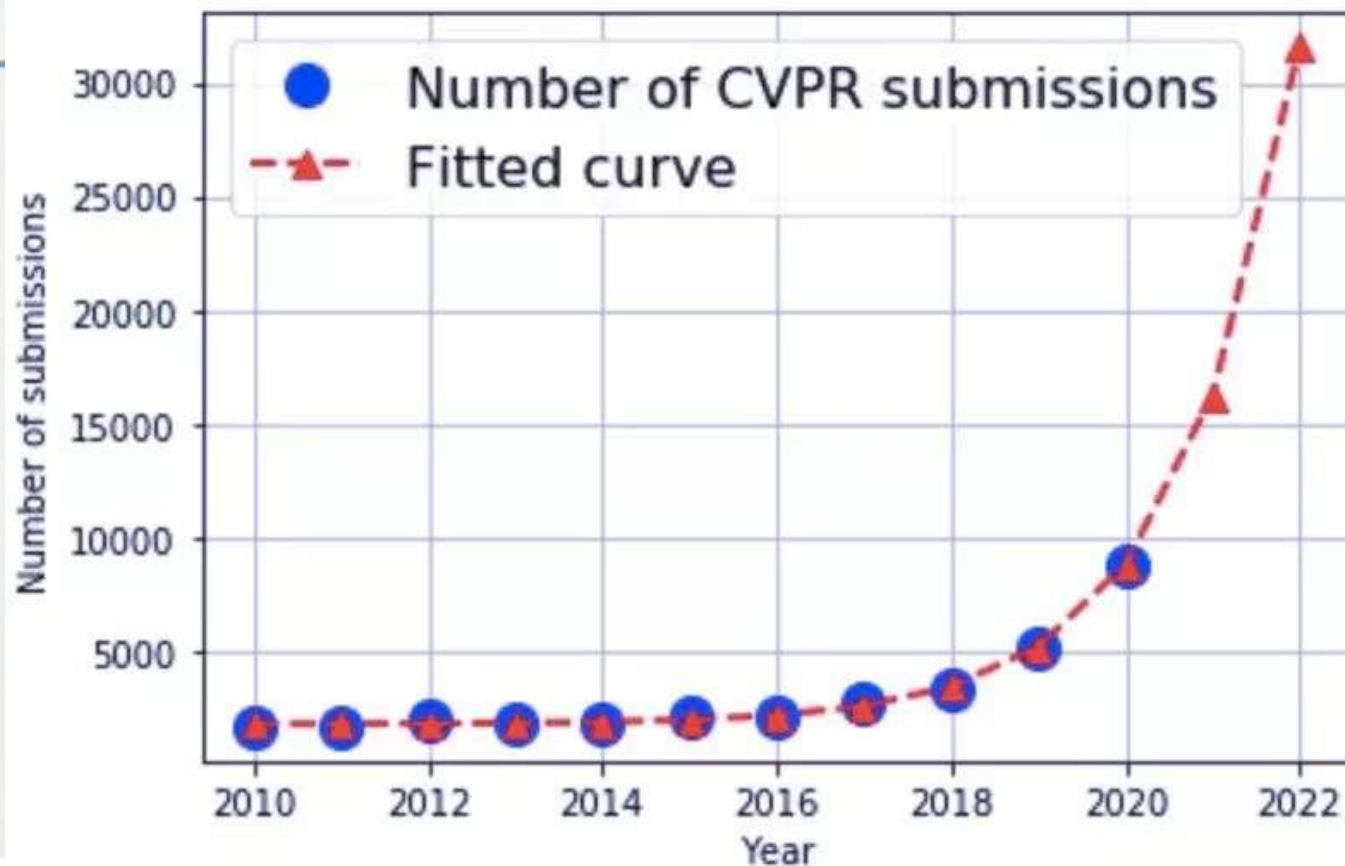
- 从2015年开始，市场对深度学习人才的需求直线上涨，2016年下半年开始更是以超大幅度上涨
- 在各大公司“算法工程师”的招聘条件中，“自然语言处理”、“深度学习”、“机器学习”成为热词



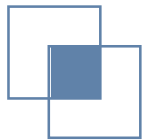
深度学习学术情况



AAAI 2020



CVPR 2020



目录

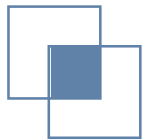


人工智能概述

深度学习入门基础

Python基础语法

作业：在线体验与Python练习



深度学习优势——从图像识别说起



实际目标：让计算机将语义概念相似的图像划分为同一类别

理想目标：让计算机像人一样理解图像



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

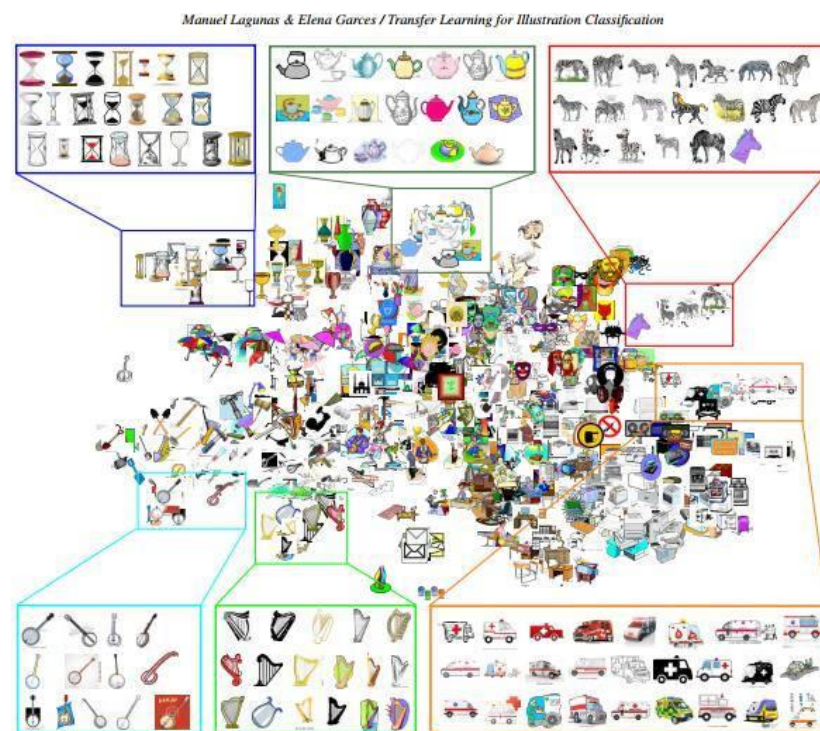
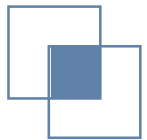


Figure 6: t-SNE algorithm on the image descriptors of the optimized network. The boxes show groups of images of the same class that have been grouped together after applying the algorithm. This shows us that the network is able to understand the low-level image characteristics and, consequently, that these image descriptors can be classified with a support vector machine obtaining great results.

我们看到的

计算机看到的

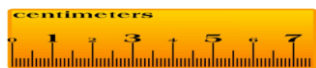


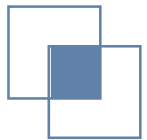
图像识别面临挑战

语义鸿沟(Semantic Gap)现象

◆ **Semantic Gap: the gap between low-level visual features and high-level concepts (图像的底层视觉特性和高层语义概念之间的鸿沟)**

➤ 例如：相似的视觉特性(color, texture, shape , ...) ,不同的语义概念





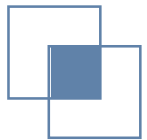
图像识别面临挑战

语义鸿沟(Semantic Gap)现象

◆ **Semantic Gap: the gap between low-level visual features and high-level concepts (图像的底层视觉特性和高层语义概念之间的鸿沟)**

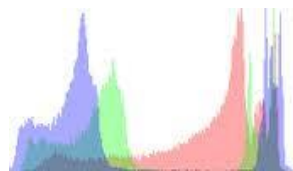
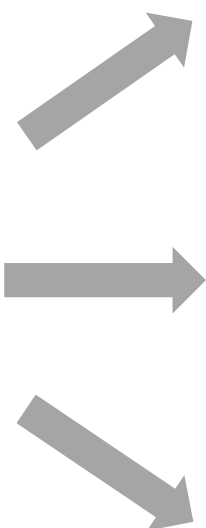
➤ 又例如：不相似的视觉特性,相同的语义概念



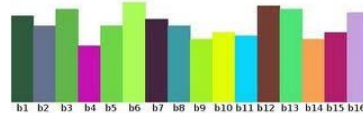


传统方法

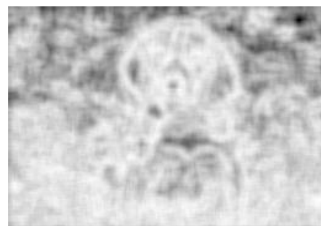
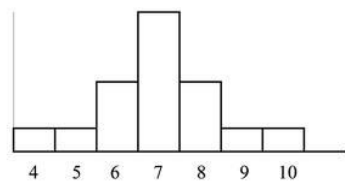
◆ 用全局的视觉底层特性统计量表示图像



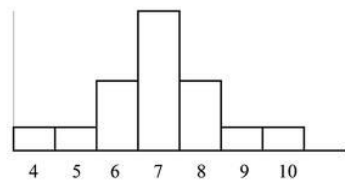
颜色



形状

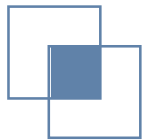


纹理



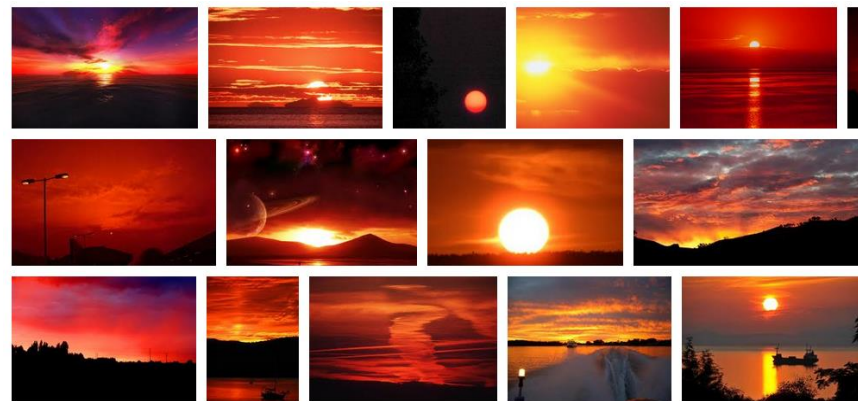
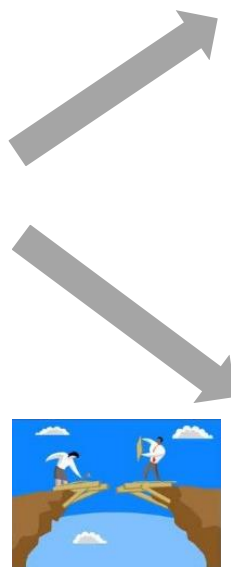
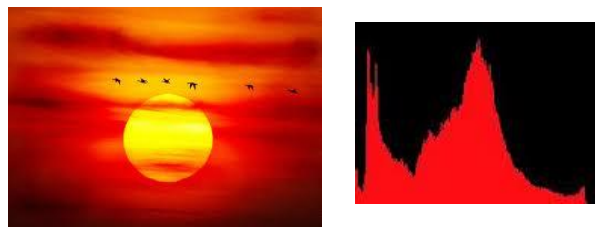
特征向量



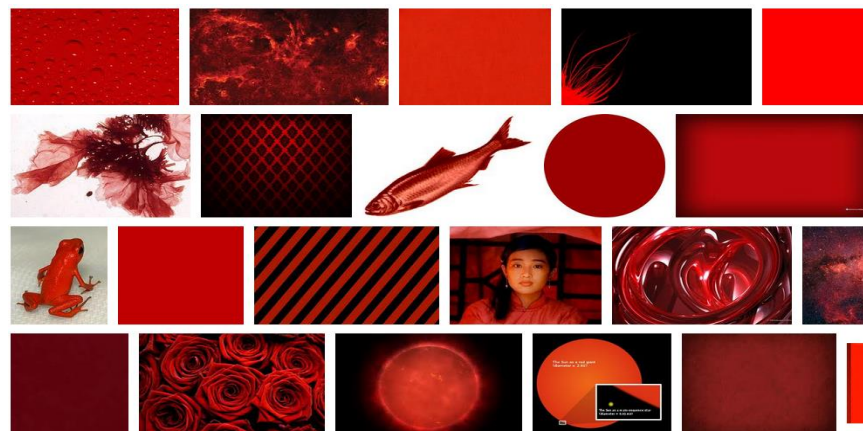


存在的问题

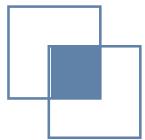
◆ 全局特征丢掉了图像细节



正确匹配



错误匹配



深度学习优势

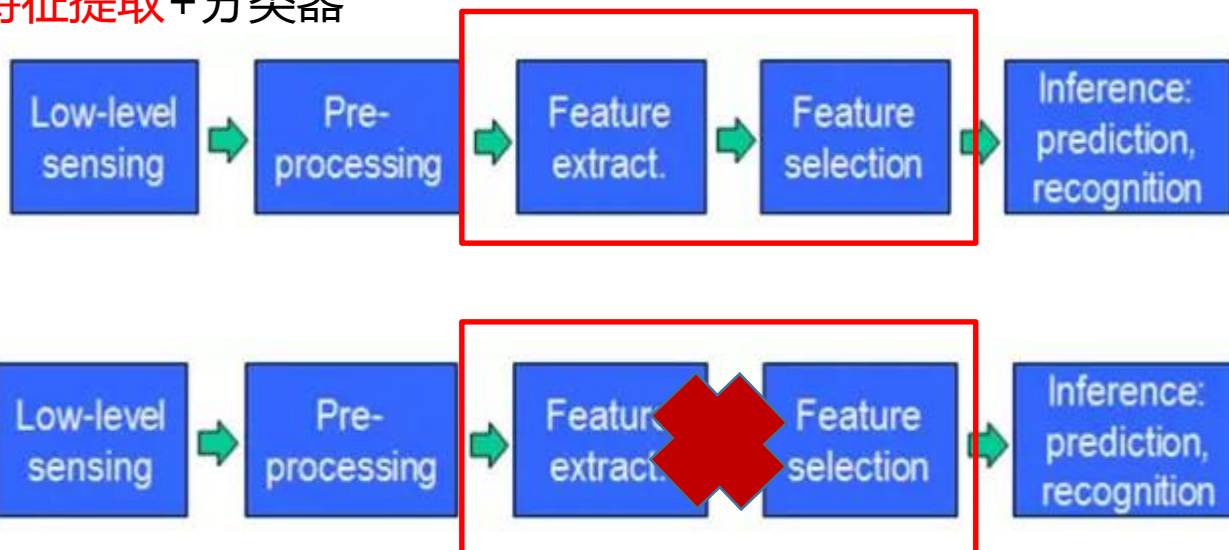


测量空间 \longrightarrow 特征空间 \longrightarrow 类别空间

核心任务

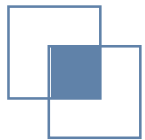
传统方法

➤ 人工特征提取+分类器



深度学习

➤ 手工地选取特征是一件非常费力、启发式（需要专业知识）的方法，能不能选取好很大程度上靠经验和运气，而且它的调节需要大量的时间，而深度学习是**学习特征**。



深度学习的一般过程



$$f(\text{狗}) = \text{狗}$$

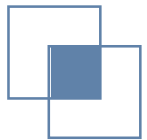
$$f(\text{狗}) = \text{狗}$$

$$f(\text{狗}) = \text{狗}$$

$$f(\text{狗}) = \text{狗}$$

$$f(\text{未知狗}) = ?$$

$$f(\text{未知狗}) = ?$$



深度学习的一般过程

深度学习的一般过程



模型

全连接神经网络

卷积神经网络

循环神经网络



策略

模型选择

损失函数选择

数学基础

编程语言

深度学习框架（算力、算法）

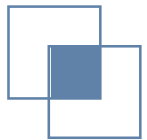


算法

学习参数

优化算法

反向传播算法



数学基础



◆ 数据表示——标量、向量、矩阵和张量

- 向量的范数、常见的向量
- 常见的矩阵、矩阵的操作

◆ 优化的基础——导数及其应用

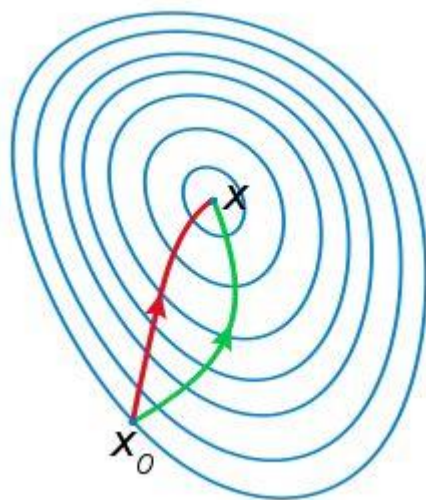
- 导数、泰勒公式
- 拉格朗日乘数法

◆ 概率模型的基础——概率论

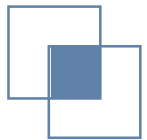
- 概率分布、边缘概率、条件概率
- 期望、方差和协方差

$$f(g(x))' = f'(g(x))g'(x)$$

$$\frac{dy}{dx} = \frac{dy}{dz} \bullet \frac{dz}{dx}$$



$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$



深度学习框架



深度学习框架的优势：

◆易用性

屏蔽底层，用户只需关注模型结构。同时，深度学习工具简化了计算，降低了深度学习入门门槛。

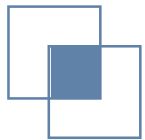
◆高效性

灵活的移植性，可将代码部署到CPU/GPU/移动端上。选择具有分布式性能的深度学习工具会使模型训练更高效。

AI Studio是针对AI学习者的在线一体化开发实训平台



选择一个合适的深度学习框架和编程语言可以事半功倍！



目录

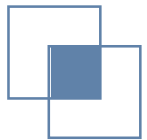


人工智能概述

深度学习入门基础

Python基础语法

作业：在线体验与Python练习



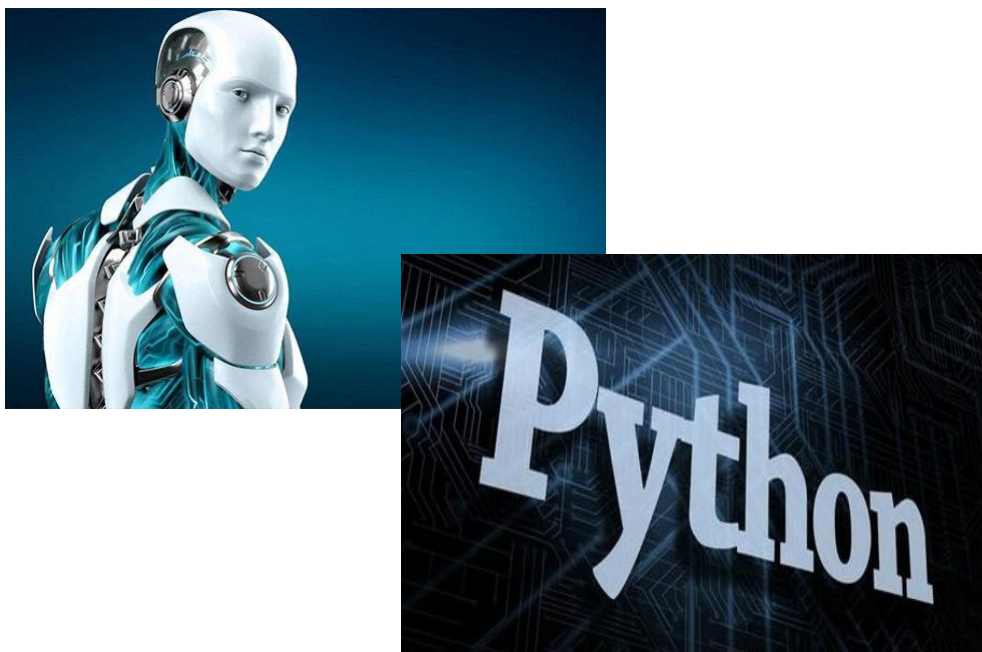
为什么选择Python



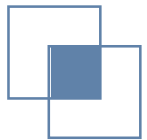
研究AI非常难，需要数学、编程、机器学习的基础，但是使用AI却很简单。Python最大的优势，就是它非常接近自然语言，易于阅读理解，编程更加简单直接，更加适合初学者。

人工智能和Python互相之间成就对方，人工智能算法促进Python的发展，而Python也让算法更加简单。

IEEE Spectrum 2019 编程语言 Top 10



Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥️ 🧠	100.0
2. C++	📱 🖥️ 🧠	98.4
3. C	📱 🖥️ 🧠	98.2
4. Java	🌐 📱 🖥️	97.5
5. C#	🌐 📱 🖥️	89.8
6. PHP	🌐	85.4
7. R	🖥️	83.3
8. JavaScript	🌐 📱	82.8
9. Go	🌐 🖥️	76.7
10. Assembly	🧠	74.5



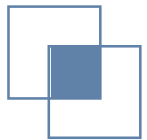
Python的优点



Python的设计混合了传统语言的软件工程的特点和脚本语言的易用性，
具有如下**特性**：

- 开源、易于维护、可移植
- 易于使用、简单优雅
- 广泛的标准库、功能强大
- 可扩展、可嵌入
- **所有的深度学习框架一般都有一个Python版的接口**
- ...





Python典型应用

人工智能



数据分析

对数据进行清洗、去重、规格化和针对性的分析是大数据行业的基石。Python是数据分析的主流语言之一

科学计算

随着NumPy, SciPy, Matplotlib等众多程序库的开发, Python越来越适合于做科学计算、绘制高质量的2D和3D图像

常规软件开发

支持函数式编程和OOP面向对象编程, 适用于常规的开发、脚本编写、网络编程

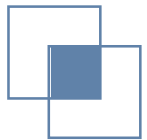
Python在人工智能大范畴领域内的机器学习、神经网络、深度学习等方面都是主流的编程语言, 得到广泛的支持和应用

网络爬虫

大数据行业获取数据的核心工具。Python是编写网络爬虫的主流编程语言, Scrapy爬虫框架应用非常广泛

WEB开发

基于Python的Web开发框架很多, 如Django, Flask



Python基础语法

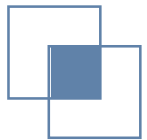


基本操作

```
age = 20           # 声明一个变量age 用来存储一个数字 20
print('Hello World!') # 打印Hello World!
1+1               # 基础数学加法
```

比较操作符	功能
<	小于
<=	小于或等于
>	大于
>=	大于或等于
==	等于
!=	不等于

逻辑操作符	描述
and	如果两个操作数均为True, 则condition变为True.
or	如果两个操作数中的任何一个为True, 则condition变为True.
not	用于反转逻辑 (不是False变为True, 而不是True变为False)



Python基础语法

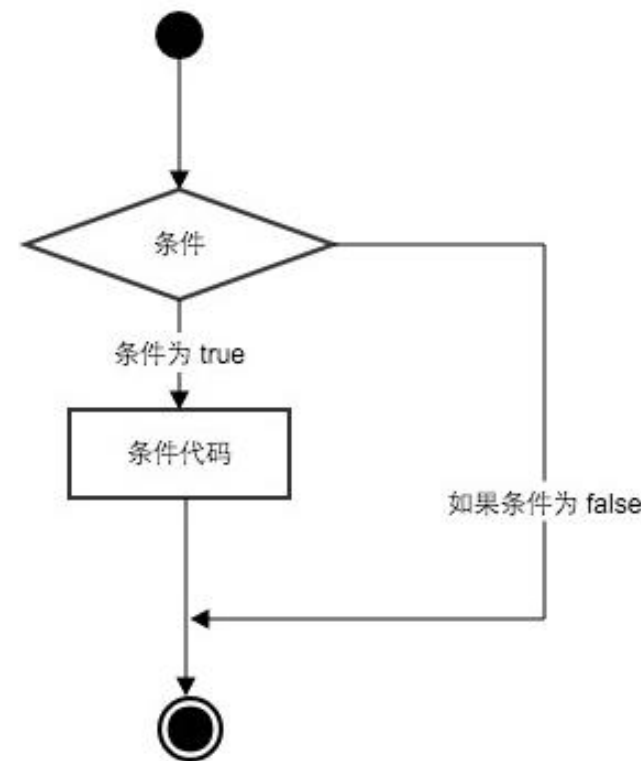


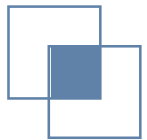
条件判断if

- 计算机之所以能做很多自动化的任务，因为它可以自己做条件判断。
- Python指定任何非0和非空值为True，0 或者 None为False
- Python 编程中 if 语句用于控制程序的执行，基本形式为：

```
if 判断条件1:  
    执行语句1  
elif 判断条件2:  
    执行语句2  
else:  
    执行语句3
```

- ✓ 其中"判断条件"成立时（非零），则执行后面的语句，而执行内容可以多行，以缩进来区分表示同一范围。
- ✓ else 为可选语句，当需要在条件不成立时执行内容则可以执行相关语句。





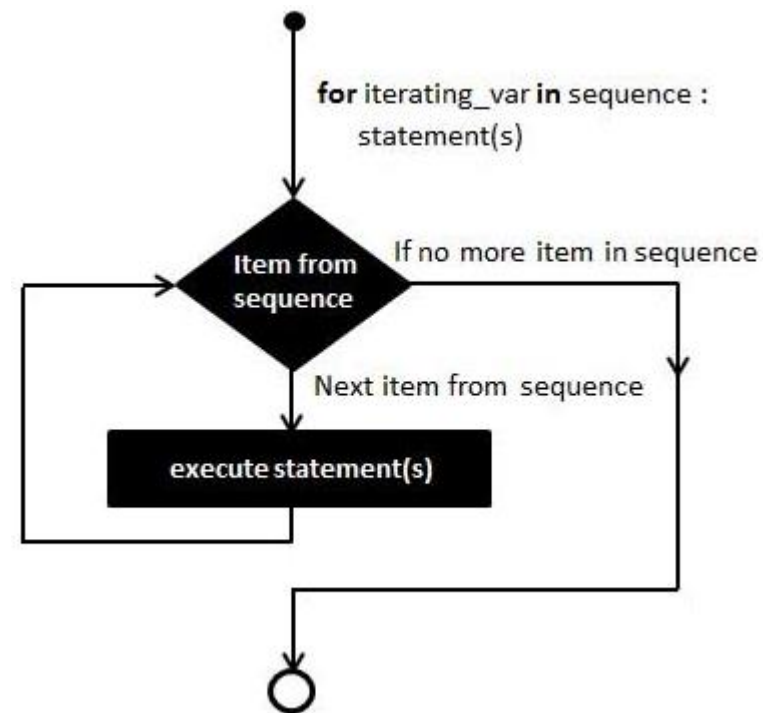
Python基础语法

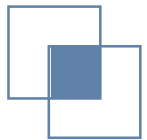


循环操作

Python的循环有两种，一种是for...in循环，可以依次把list或tuple中的元素迭代出来

```
>>> for i in range(5):  
...     print(i)  
...  
0  
1  
2  
3  
4
```





Python基础语法



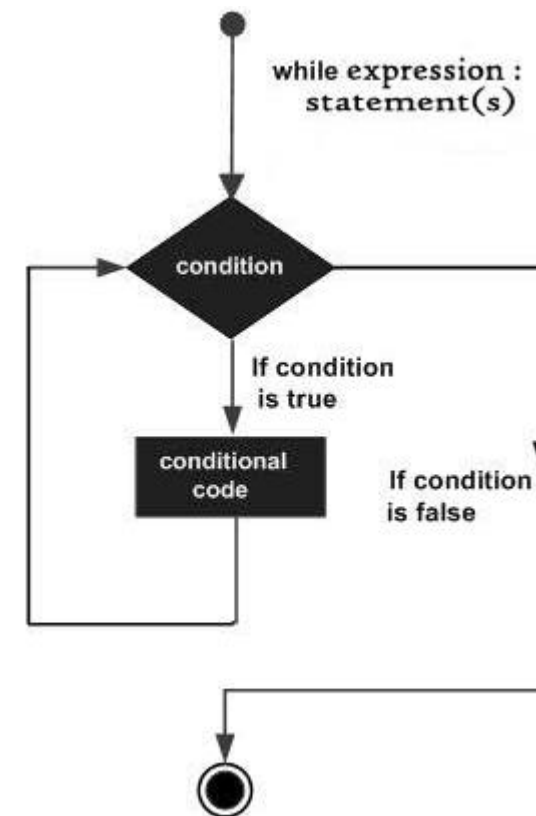
循环操作

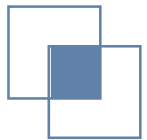
第二种是while，只要条件满足，就不断循环，条件不满足时退出循环

在 while ... else 在条件语句为 false 时执行 else 的语句块

```
>>> sum = 0
>>> n = 99
>>> while n > 0:
...     sum = sum + n
...     n = n - 2
...
>>> print(sum)
2500
```

```
>>> count = 0
>>> while count < 3:
...     print(count, "小于 3")
...     count = count + 1
... else:
...     print(count, "大于或等于 3")
...
0 小于 3
1 小于 3
2 小于 3
3 大于或等于 3
```





Python基础语法



break、continue、pass语句

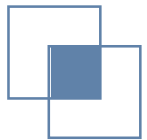
- break语句可以跳出 for 和 while 的循环体
- continue语句跳过当前循环，直接进行下一轮循环
- pass是空语句，一般用做占位语句，不做任何事情

```
>>> n = 1
>>> while n <= 100:
...     if n > 10:
...         break
...     print(n)
...     n += 1
```

```
>>> n = 1
>>> while n < 10:
...     n = n + 1
...     if n % 2 == 0:
...         continue
...     print(n)
```

```
>>> for letter in 'Room':
...     if letter == 'o':
...         pass
...     print('pass')
...     print(letter)
...
R
pass
o
pass
o
m
```

结果：打印1~10



Python基础语法

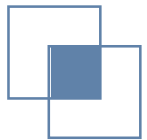


核心数据结构

◆ Number (数字)

- Python支持int, float, complex三种不同的数字类型

```
>>> a = 3
>>> b = 3.14
>>> c = 3 + 4j
>>> print(type(a), type(b), type(c))
<class 'int'> <class 'float'> <class 'complex'>
>>> isinstance(a, int)
True
```

Python基础语法

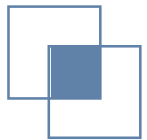


核心数据结构

◆ String (字符串)

- Python中的字符串可以使用单引号、双引号和三引号（三个单引号或三个双引号）括起来，使用反斜杠 \ 转义特殊字符
- Python3源码文件默认以UTF-8编码，所有字符串都是unicode字符串
- 支持字符串拼接、截取等多种运算

```
>>> a = "Hello"
>>> b = "Python"
>>> print("a + b 输出结果：", a + b)
a + b 输出结果： HelloPython
>>> print("a[1:4] 输出结果：", a[1:4])
a[1:4] 输出结果： ell
```

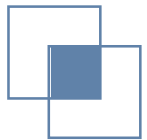


核心数据结构

◆ List (列表)

- 列表可以完成大多数集合类的数据结构实现。列表中元素的类型可以不相同，它支持数字，字符串甚至可以包含列表（所谓嵌套）。
- 列表是写在方括号 [] 之间、用逗号分隔开的元素列表。
- 列表索引值以 0 为开始值，-1 为从未尾的开始位置。
- 列表可以使用+操作符进行拼接，使用*表示重复。

```
>>> list = ['abcd', 786 , 2.23, 'runoob', 70.2]
>>> print(list[1:3])
[786, 2.23]
>>> tinylist = [123, 'runoob']
>>> print(list + tinylist)
['abcd', 786, 2.23, 'runoob', 70.2, 123, 'runoob']
```



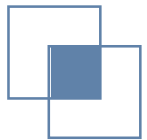
核心数据结构

◆ Tuple (元组)

- tuple与list类似，不同之处在于tuple的元素不能修改。tuple写在小括号里，元素之间用逗号隔开。
- 元组的元素不可变，但可以包含可变对象，如list。

! 注意：定义一个只有1个元素的tuple，必须加逗号。

```
>>> t = ('abcd', 786 , 2.23, 'runoob', 70.2)
>>> t1 = (1, )
>>> t2 = ('a', 'b', ['A', 'B'])
>>> t[2][0] = 'X'
>>> t
('a', 'b', ['X', 'B'])
```

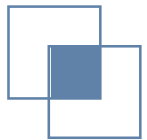


核心数据结构

◆ dict (字典)

- 字典是无序的对象集合，使用键-值 (key-value) 存储，具有极快的查找速度。
- 键(key)必须使用不可变类型。
- 同一个字典中，键(key)必须是唯一的。

```
>>> d = {'Michael': 95, 'Bob': 75, 'Tracy': 85}
>>> d['Michael']
95
```



核心数据结构

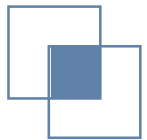
◆ set (集合)

- set和dict类似，也是一组key的集合，但不存储value。由于key不能重复，所以，在set中，没有重复的key。
- set是无序的，重复元素在set中自动被过滤。

```
>>> s = set([1, 2, 3])
>>> s
{1, 2, 3}
>>> s = set([1, 1, 2, 2, 3, 3])
>>> s
{1, 2, 3}
```



set可以看成数学意义上的无序和无重复元素的集合，因此，两个set可以做数学意义上的交集（&）、并集（|）、差集（-）等操作。



Python基础语法



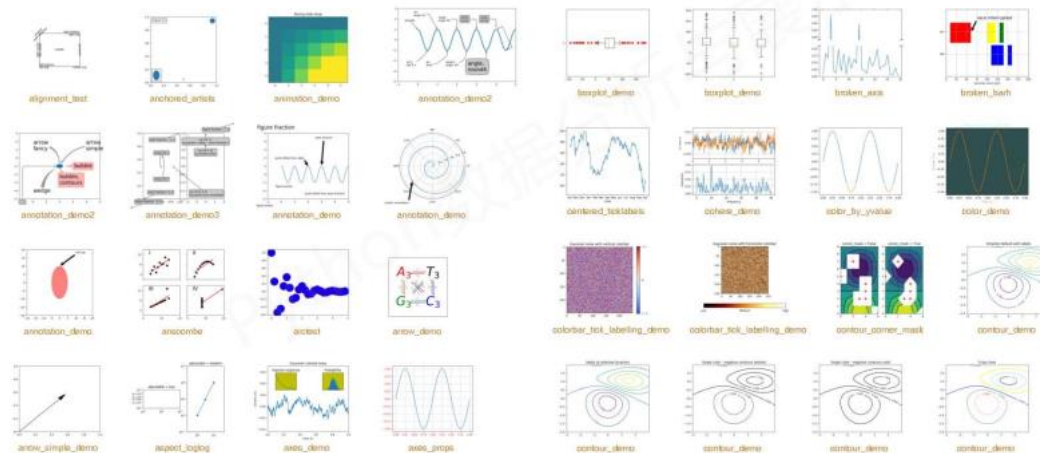
常用库

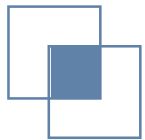
- **numpy** (Numerical Python的简称) 是高性能科学计算和数据分析的基础包。
- **matplotlib**由各种可视化类构成，内部结构复杂的可视化基础包。
-

方法	说明
sum	对数组中全部或某轴向的元素求和
mean	算术平均数。零长度的数组的mean为NaN
std, var	分别为标准差和方差，自由度可调（默认为n）
min, max	最大值和最小值
argmin, argmax	分别为最大和最小元素的索引
cumsum	所有元素的累加
cumprod	所有元素的累积

Matplotlib库的效果

<http://matplotlib.org/gallery.html>





目录

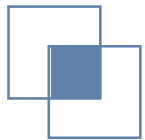


人工智能概述

深度学习入门基础

Python基础语法

作业：在线体验与Python练习

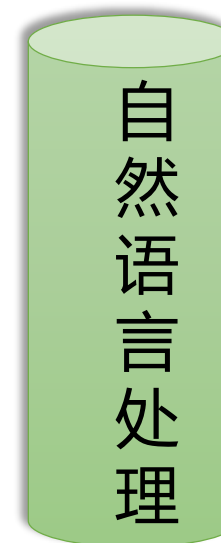


在线体验——百度AI开放平台



官方地址：<https://ai.baidu.com/productlist>

热门方向



技术能力

技术能力

语音技术

图像技术

文字识别

人脸与人体识别

视频技术

AR与VR

自然语言处理

数据智能

知识图谱

场景方案

部署方案

开发平台

行业应用方案

语音技术

语音识别 >

语音识别

识别率高，支持中文、英语、粤语、四川话等

语音识别极速版

极速识别60秒内语音，简单易用

长语音识别

支持将不限时长的语音实时转换为文字

远场语音识别

适用于智能家居、机器人等远场的语音识别

语音合成 >

在线合成-基础音库

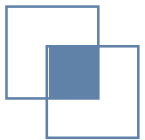
提供标准男声女声、情感男声女声四种发音人

在线合成-精品音库

提供包含童声在内的五种精选发音人

离线语音合成

在无网或弱网环境下，可在智能硬件设备终端进行...



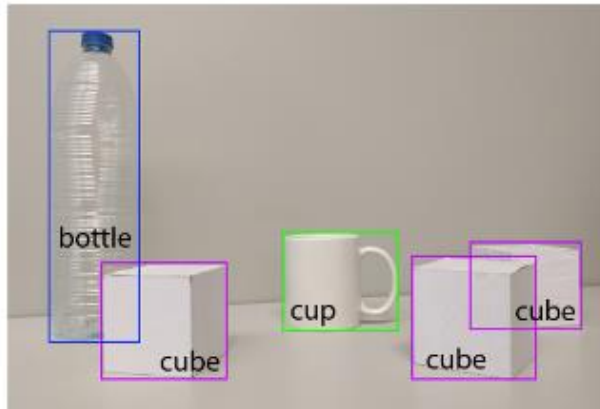
计算机视觉



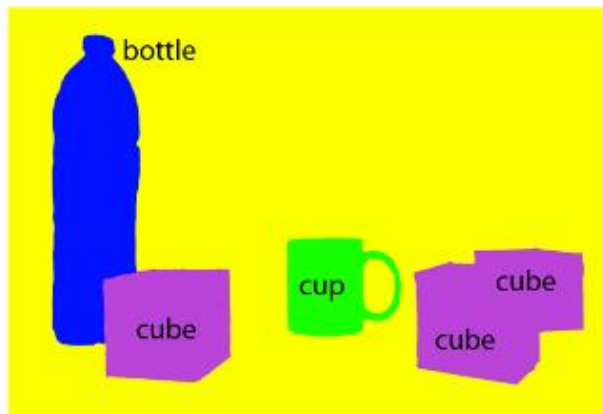
计算机视觉 (Computer Vision) , 顾名思义是一门“教”会计算机如何去“看”世界的学科。



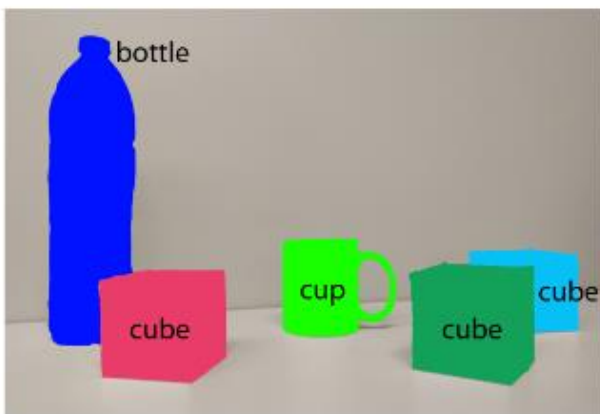
(a) Image classification



(b) Object localization



(c) Semantic segmentation



(d) Instance segmentation

左图展示了四种主要的计算机视觉任务（由粗粒度到精细粒度）

1、图像分类 (what)

为图像赋予一个或多个语义标签

2、目标检测 (what & where)

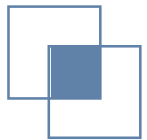
找到图像中物体的类别及所在位置

3、图像语义分割 (what & where)

找到图像中物体的类别并精确勾勒出其所所在位置

4、图像实例分割 (what & where)

多个同类物体存在时，将它们一一区分出来



计算机视觉在线体验



人像动漫化

运用对抗生成网络技术，结合人脸检测、头发分割、人像分割等技术，为用户量身定制千人千面的二次元动漫形象，并支持通过参数设置，生成戴口罩的二次元动漫人像。

优化前



优化后



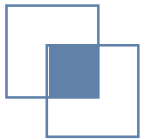
请输入网络图片URL

检测 或 本地上传

图片文件类型支持PNG、JPG、JPEG、BMP，图片大小不超过2M。

图片文件类型支持PNG、JPG、JPEG、BMP，图片大小不超过2M。

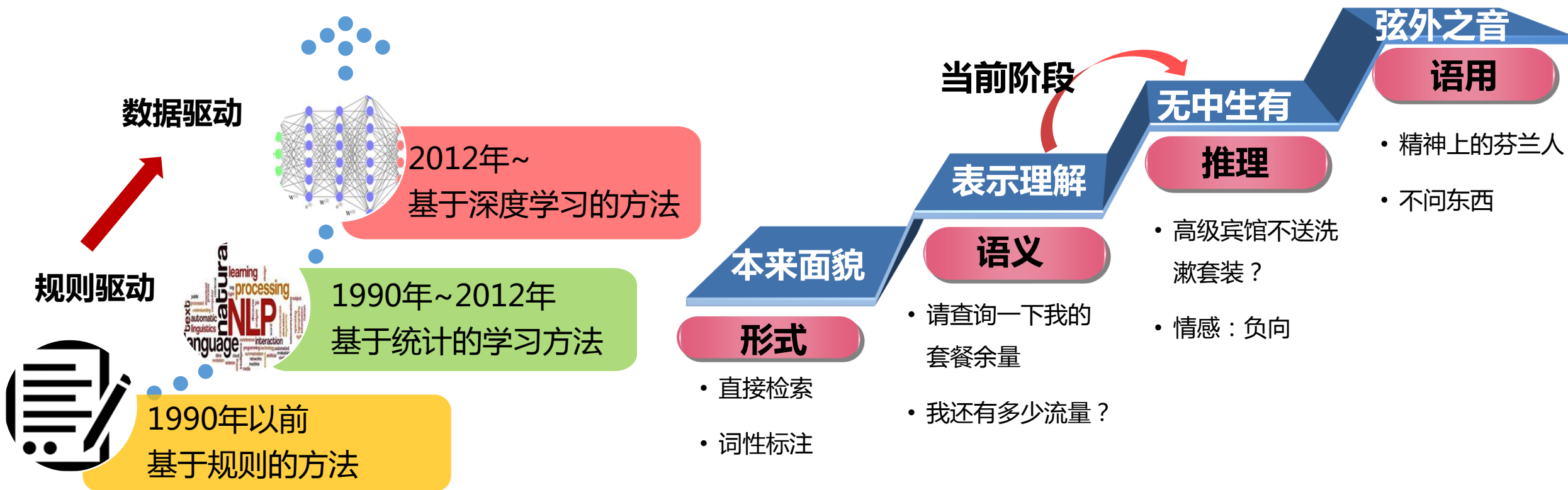
轻松拥有属于你自己的二次元头像！

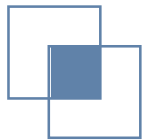


自然语言处理



自然语言处理研究实现人与计算机之间用自然语言进行有效通信的各种理论和方法。自然语言处理技术发展经历了**基于规则的方法**、**基于统计学习的方法**和**基于深度学习的方法**三个阶段。自然语言处理由浅入深的四个层面分别是**形式**、**语义**、**推理**和**语用**，当前正处于由语义向推理的发展阶段。





自然语言处理在线体验



对话情绪识别

基于大数据和深度学习，自动学习词义及词序关系，具备较强的泛化能力，情绪识别精度高，在多个垂类（金融客服、闲聊对话、导航交互），负向情绪的识别准确率可达90%以上。

请输入一段想分析的文本：[随机示例](#)

通用场景

客服场景

闲聊场景

任务类型对话

你继续玩吧，我没事，不用管我！|

直男，你可能要失去你女朋友了！

分析结果：



负向情绪



中性情绪



正向情绪

请输入一段想分析的文本：[随机示例](#)

通用场景

客服场景

闲聊场景

任务类型对话

好想一巴掌打死你

还可以输入 247 个字

分析结果：



负向情绪：愤怒

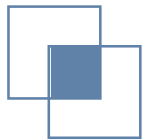


中性情绪



正向情绪

参考安抚话术：珍惜生命



课后作业



□ 作业一：Python应用

- ✓ 输出九九乘法表（注意格式）

□ 作业二：查找特定名称文件

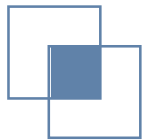
- ✓ 遍历“Day1-homework”目录下文件；
- ✓ 找到文件名包含“2020”的文件；
- ✓ 将文件名保存到数组result中；
- ✓ 按照序号、文件名分行打印输出。

□ 体验：百度AI开放技术

- ✓ 在<https://ai.baidu.com/productlist>任意选取你感兴趣的技术体验一下，后续在班主任指引下，在心得中说出你的感受。

```
1*1=1
1*2=2 2*2=4
1*3=3 2*3=6 3*3=9
1*4=4 2*4=8 3*4=12 4*4=16
1*5=5 2*5=10 3*5=15 4*5=20 5*5=25
1*6=6 2*6=12 3*6=18 4*6=24 5*6=30 6*6=36
1*7=7 2*7=14 3*7=21 4*7=28 5*7=35 6*7=42 7*7=49
1*8=8 2*8=16 3*8=24 4*8=32 5*8=40 6*8=48 7*8=56 8*8=64
1*9=9 2*9=18 3*9=27 4*9=36 5*9=45 6*9=54 7*9=63 8*9=72 9*9=81
```

```
[1, 'Day1-homework/26/26/new2020.txt']
[2, 'Day1-homework/18/182020.doc']
[3, 'Day1-homework/4/22/04:22:2020.txt']
```



感谢聆听



有什么问题吗？