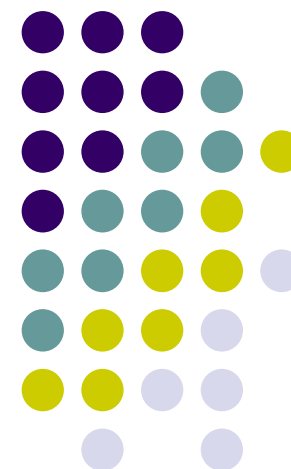


机器学习概述

哈尔滨工业大学计算学部 刘远超



机器学习概述



本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- 机器学习方法的分类
- 半监督学习
- 主动学习
- 排序学习

机器学习概述



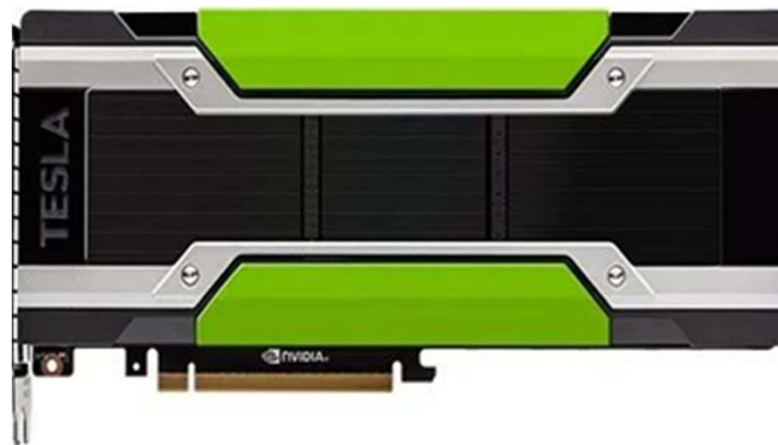
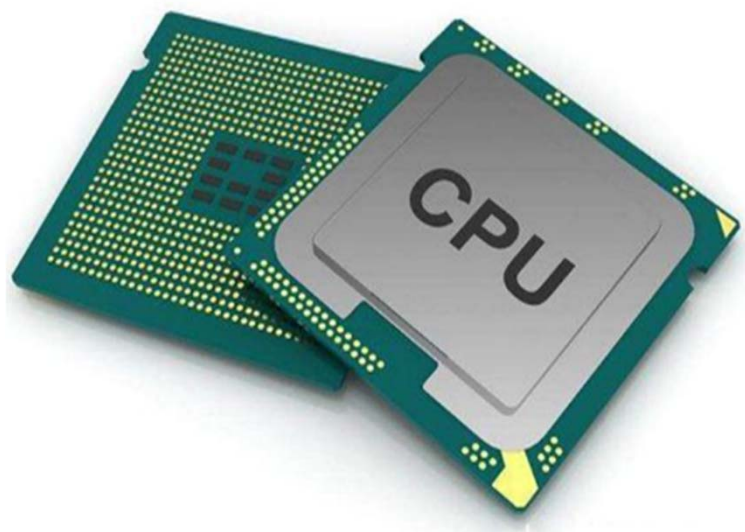
本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- 机器学习方法的分类
- 半监督学习
- 主动学习
- 排序学习

深度学习的硬件计算环境

4

深度学习中的重要芯片：GPU



GPU (Graphics Processing Unit) 即图形处理单元，又称显示核心、视觉处理器、显示芯片，是显卡上的一块芯片，与显卡集成在一起。

深度学习的硬件计算环境

5

GPU和CPU进行对比，来进一步理解什么是GPU

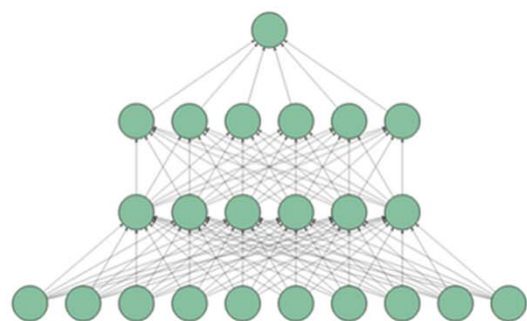
- 1) **从分工和擅长来看。** GPU则为专用处理器，其适合数据之间关联性不高、可分块处理、高度线程化的大规模并行计算。
- 2) **从CPU和GPU的协作关系来看。** CPU是主芯片而GPU是从芯片，GPU是在替CPU分担部分工作，而不是尝试完全取代CPU。
- 3) **从内部构成上看。** 与CPU相比，GPU有数量众多的逻辑运算单元（ALU）和超长的流水线，可以并行处理大量但较为简单的任务，特别适合处理类型统一的数据。
- 4) **从所在位置上看。** GPU位于独立显卡上，与显卡集成在一起。而CPU则通常是单独的芯片，插在PC机的主板插槽上。
- 5) **从制造厂商来看。** 比较知名的 GPU 制造商有AMD、NVIDIA等公司，其对应的产品分别为人们通常所说的A卡和N卡。

深度学习的硬件计算环境

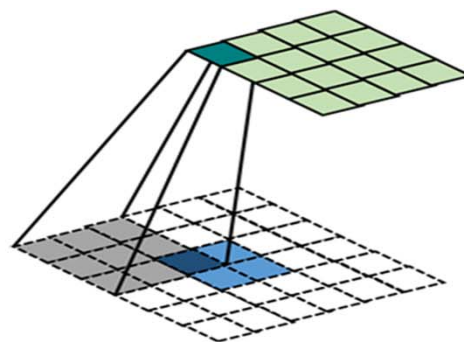
6

深度学习为什么需要GPU芯片

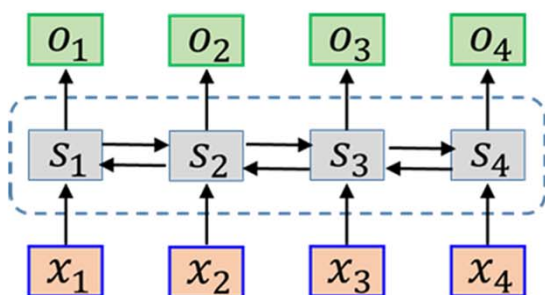
神经网络中有很多计算实际上是可以并行进行的。理论上，对于不依赖于其他计算结果的计算，就可以采用并行的方式。



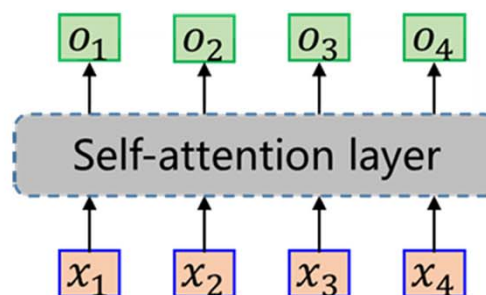
(a) MLP



(b) CNN



(c) RNN



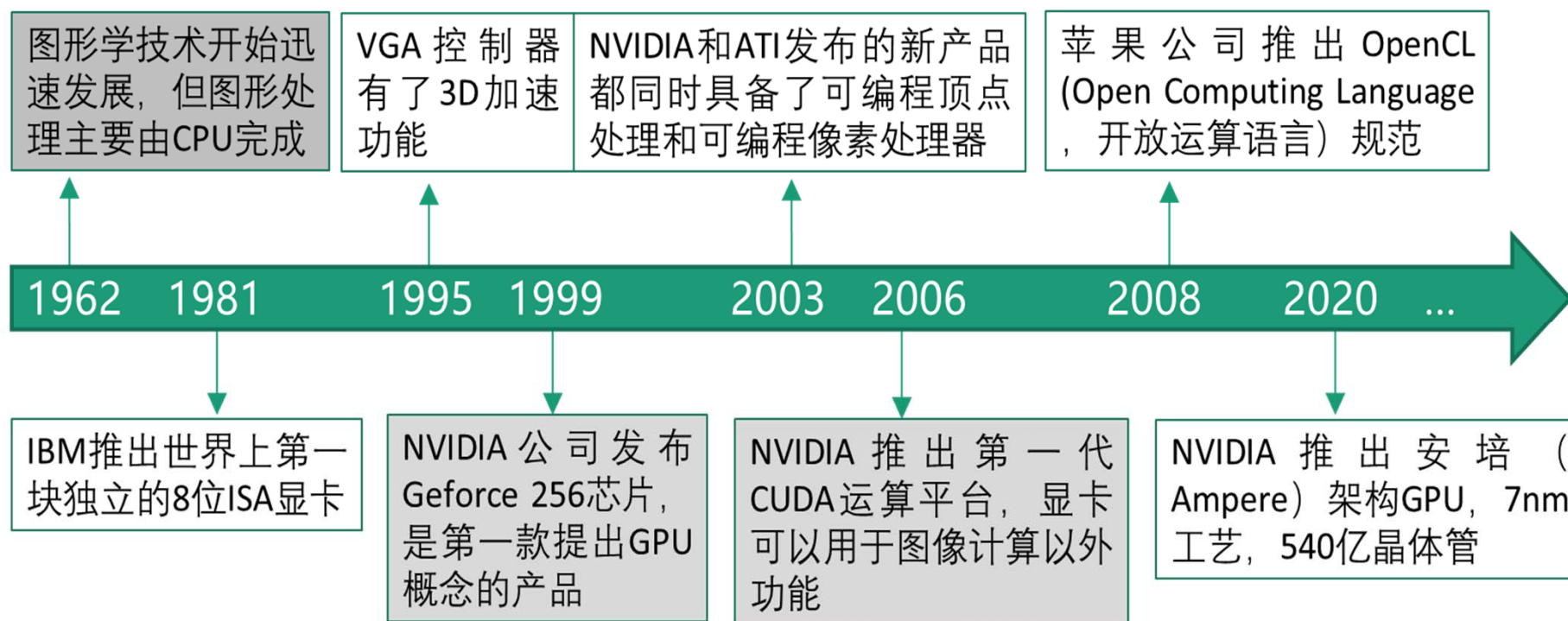
(d) Transformer

几种典型的神经网络

深度学习的硬件计算环境

7

GPU芯片发展历程



深度学习的硬件计算环境

8

深度学习GPU环境的搭建

如果使用GPU，以**NVIDIA**产品为例，则通常需要进行如下预先步骤：

- 1) 根据需要，购置相应的**GPU显卡**（例如NVIDIA RTX 3080或者3090），并安装相应的驱动程序。
- 2) **安装CUDA**。CUDA是NVIDIA公司推出的一种通用并行计算平台和编程模型，从而使其可以高效地解决大规模计算问题。
- 3) **安装cuDNN**（The NVIDIA CUDA Deep Neural Network library）。cuDNN是专用于深度神经网络的GPU加速库，可加速很多流行的深度学习框架，如Caffe2、Keras、PyTorch 和 TensorFlow等。cuDNN 可大幅优化标准神经网络标准功能（例如用于前向传播和反向传播的卷积层、池化层、归一化层和激活层）的实施，使研究人员和开发者专注于训练神经网络及开发软件应用，不必花时间进行低层级的 GPU 性能调整。

机器学习概述

9

本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- 机器学习方法的分类
- 半监督学习
- 主动学习
- 排序学习

机器学习概述

10

本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- 机器学习方法的分类
- 半监督学习
- 主动学习
- 排序学习

深度学习的软件计算环境

11

基于Python的重要软件计算平台：Anaconda

Anaconda提供和集成了若干科学计算的常用编程语言和工具，如python、安装包的管理、科学计算常用函数库和典型的机器学习算法库等。安装完毕Anaconda，则可以一次性的设置好上述环境。其包含如下常用模块：

- **Conda**：具有安装、更新、删除、解决包依赖关系的包管理功能。
- **Numpy** (Numerical Python)：重点在于数组和矩阵运算；
- **Scipy**：包括用于最优化、线性代数、积分、插值、拟合、特殊函数等常用科学计算软件包。其依赖于Numpy；
- **Matplotlib**：是 Python 中类似 MATLAB 的重要的绘图工具和数据可视化工具（用图表的形式对数据进行展示）；
- **Scikit-learn**：是一款重要的开源机器学习工具包，涵盖分类，回归和聚类等的常用算法。例如包括**支持向量机、逻辑回归、朴素贝叶斯、随机森林，k-means等算法**等；
-

深度学习的软件计算环境

12

深度学习框架及其安装

- **什么是深度学习框架。** 深度学习框架是一种便于深度学习和研究人员而开发的开源平台和工具，其中包含了深度学习模型常用的开发包、预训练模型等资源。
- **为什么需要深度学习的框架。**
 - 深度学习框架为模型的实现提供了清晰而简洁的方法，使复杂神秘的深度学习模型开发被大大简化，是AI开发的利器。
 - 借助深度学习框架，深度学习和研究人员可以利用预先构建和优化好的组件集合定义模型，避免了写大量重复代码的工作（即**避免“重复造轮子”**）。

深度学习的软件计算环境

13

深度学习框架及其安装

■ 典型的深度学习框架。

- TensorFlow
- PyTorch
- MindSpore等。

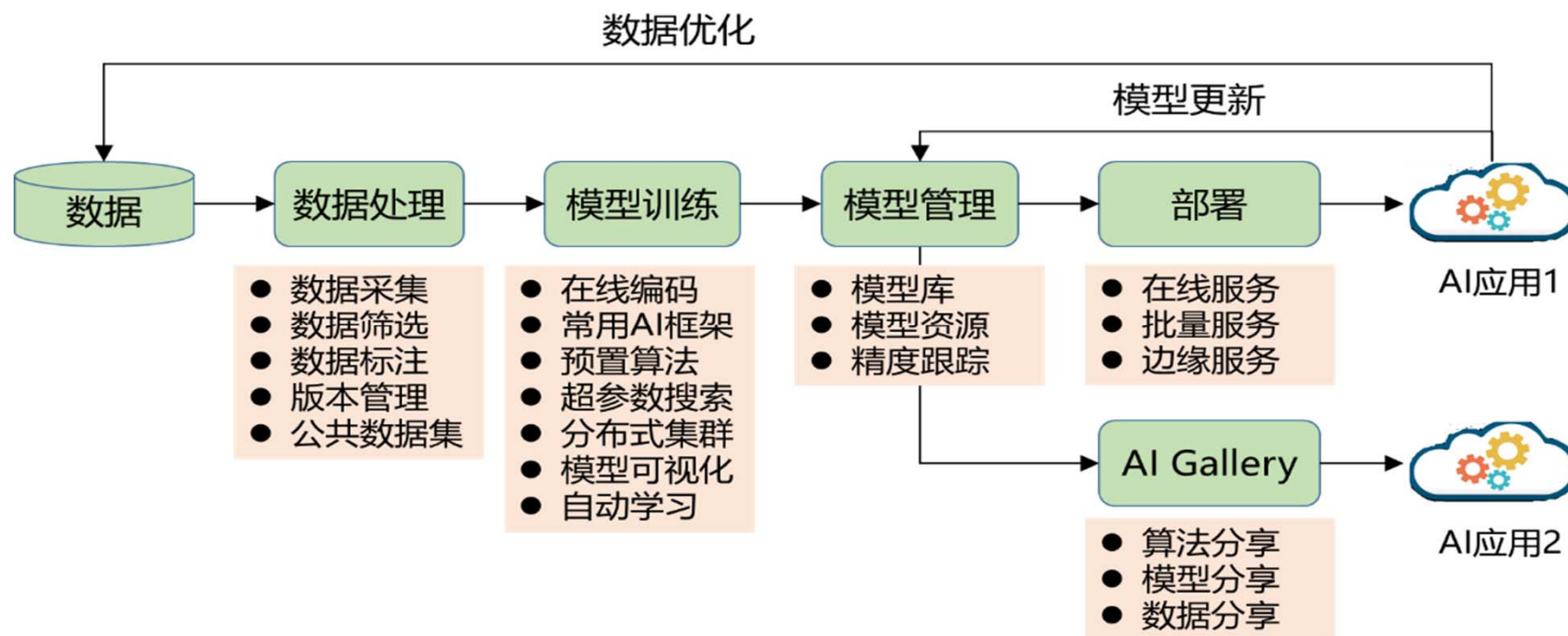
■ 使用虚拟环境进行深度学习框架安装的必要性。

- 其优点在于，不同虚拟环境下的安装结果互不影响，可以很容易的切换不同环境。
- 反之，如果不创建虚拟环境，而直接在同一个物理环境下安装，则只能使用同一种配置或者版本，无法自由切换（只能采取卸载旧版本、重新安装新版本的办法更换计算环境）。

深度学习的软件计算环境

14

深度学习的云服务计算环境



华为昇腾人工智能平台ModelArts

机器学习概述

15

本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- 机器学习方法的分类
- 半监督学习
- 主动学习
- 排序学习

机器学习概述

16

本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- **数据集**
- 机器学习方法的分类
- 半监督学习
- 主动学习
- 排序学习

数据集

17

数据集的数学表示：以鸢尾花(Iris)数据集为例



山鸢尾(*Iris setosa*)



变色鸢尾(*Iris versicolor*)

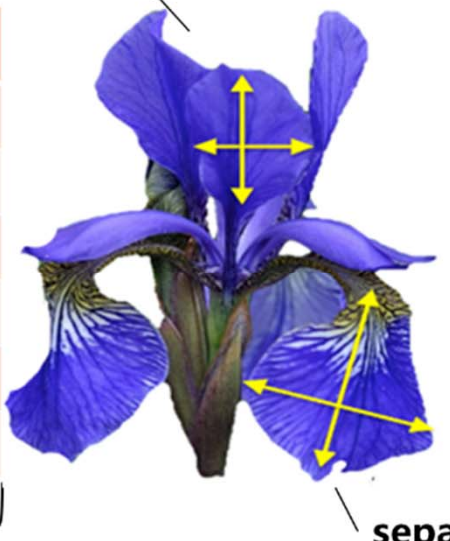


维吉尼亚鸢尾(*iris virginica*)

数据集

18

鸢尾花数据集的数据样例

| Samples(instances, observations) | | | | | | |
|--|------|------|------|------|------------------------|--|
| | 花萼长度 | 花萼宽度 | 花瓣长度 | 花瓣宽度 | 分类标签 | |
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | Setosa |  |
| 2 | 4.9 | 3.0 | 1.4 | 0.2 | Setosa | |
| ... | ... | ... | ... | ... | ... | |
| 50 | 6.4 | 3.5 | 4.5 | 1.2 | Versicolor | |
| ... | ... | ... | ... | ... | ... | |
| 150 | 5.9 | 3.0 | 5.0 | 1.8 | Virginia | |
| Features(attributes, measurements, dimensions) | | | | | Class labels (targets) | |

数据集

19

数据集的数学表示

- 数据集由若干样本（或称实例）组成，其中每个样本 i 通常由特征（Features） x_i （或称属性）和类别标签 y_i （或称目标值）组成。因此数据集在数学上通常表示为 $\{(x_1, y_1), \dots, (x_i, y_i), \dots, (x_m, y_m)\}$ 的形式。
- 由于样本一般有多个特征，例如鸢尾花数据集有4个特征，因而 x_i 通常进一步表示为向量，即 $x_i = \{x_i^1, x_i^2, \dots, x_i^n\}^T$ 。而 y_i 则为样本 i 的类别标签，或称目标值（target value）。
- 数据集中样本的类别标签一般被称之为真实值（Ground Truth Value），而通过机器学习算法预测的标签则称之为预测值（Prediction Value）。

类别标签的ground truth与gold standard



- **ground truth**: 可翻译为地面实况等。在机器学习领域一般用于表示真实值、**标准答案**等，表示通过直接观察收集到的真实结果。
- **gold standard**: 可翻译为金标准。医学上一般指诊断疾病公认的最可靠的方法。
- 在机器学习领域，更倾向于使用“**ground truth**”。而如果用 **gold standard**这个词，则表示其可以很好地代表**ground truth**。

数据集

21

人工标注的一致性分析

■ 皮尔森相关系数 (Pearson correlation coefficient)

- **问题举例：**如何评价两个评委的一致性？

rater1 = [0.5, 1.6, 2.5, 2.5, 2.4]

rater2 = [1.5, 2.6, 3.5, 3.5, 3.4]

- 皮尔森相关系数(Pearson coefficient)的应用背景：
 - 用来衡量两个用户之间兴趣的一致性
 - 用来衡量预测值与真实值之间的相关性
 - 既适用于离散的、也适用于连续变量的相关分析
- X和Y之间的皮尔森相关系数计算公式：

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

其中, $\text{cov}(X,Y)$ 表示X 和Y 之间的协方差 (Covariance), σ_X 是X的均方差, μ_X 是X的均值, E 表示数学期望

- 取值区间为[-1, 1]。 -1: 完全的负相关, +1: 表示完全的正相关, 0: 没有线性相关

数据集

22

人工标注的一致性分析

科恩·卡帕相关系数 (Cohen's kappa correlation coefficient)

- Cohen's kappa相关系数也可用于衡量两个评价者之间的一致性。其特点在于：
 - 与pearson相关系数的区别：Cohen's kappa 相关系数通常用于离散的分类的一致性评价。
 - 其通常被认为比两人之间的简单一致百分比更强壮，因为Cohen's kappa考虑到了二人之间的随机一致的可能性。
- 如果评价者多于2人时，可以考虑使用[Fleiss' kappa](#).

数据集

23

人工标注的一致性分析

Cohen's kappa计算方法

● 假设有50个人申请奖学金。有两个评委A 和 B。每个评委对每个申请者说“Yes” or “No”。假设AB一致性情况如下矩阵所示。

| | | B | |
|---|-----|-----|----|
| | | Yes | No |
| A | Yes | a | b |
| | No | c | d |

| | | B | |
|---|-----|-----|----|
| | | Yes | No |
| A | Yes | 20 | 5 |
| | No | 10 | 15 |

- 先计算AB一致性比例为 $p_o = \frac{(a+d)}{(a+b+c+d)} = \frac{20+15}{20+5+10+15} = 0.7$
- 再计算AB之间的随机一致性概率 p_e (the probability of random agreement), 注意到:
 1. A对25个申请者说yes, 因此 比例为25/50=50%
 2. B对30个申请者说yes, 因此比例为30/50=60%
 3. 因此, AB两人随机都说YES的概率 $p_{yes} = \frac{(a+b)}{(a+b+c+d)} \cdot \frac{(a+c)}{(a+b+c+d)} = 0.5 * 0.6 = 0.3$
 4. 同样, $p_{no} = \frac{(c+d)}{(a+b+c+d)} \cdot \frac{(b+d)}{(a+b+c+d)} = 0.5 * 0.4 = 0.2$
 5. AB的整体随机一致性(Overall random agreement) 概率 $p_e = p_{yes} + p_{no} = 0.3+0.2=0.5$
- 最后应用Cohen's kappa公式, 得到 $k = \frac{p_o - p_e}{1 - p_e} = \frac{0.7 - 0.5}{1 - 0.5} = 0.4$

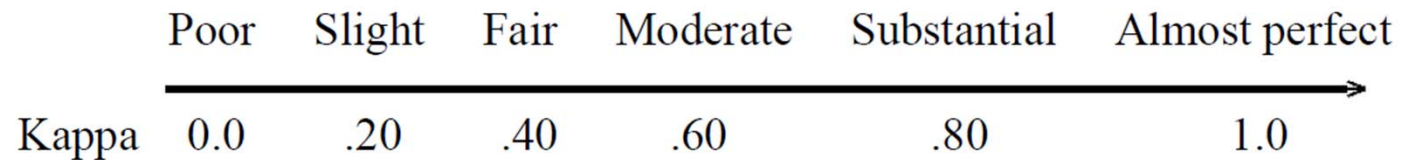
数据集

24

人工标注的一致性分析

- kappa score是一个介于-1到+1之间的数.

Interpretation of Kappa



| <u>Kappa</u> | <u>Agreement</u> |
|--------------|----------------------------|
| < 0 | Less than chance agreement |
| 0.01–0.20 | Slight agreement |
| 0.21– 0.40 | Fair agreement |
| 0.41–0.60 | Moderate agreement |
| 0.61–0.80 | Substantial agreement |
| 0.81–0.99 | Almost perfect agreement |

Fleiss' kappa

| n_{ij} | 1 | 2 | 3 | 4 | 5 |
|----------|-------|-------|-------|-------|-------|
| 1 | 0 | 0 | 0 | 0 | 14 |
| 2 | 0 | 2 | 6 | 4 | 2 |
| 3 | 0 | 0 | 3 | 5 | 6 |
| 4 | 0 | 3 | 9 | 2 | 0 |
| 5 | 2 | 2 | 8 | 1 | 1 |
| 6 | 7 | 7 | 0 | 0 | 0 |
| 7 | 3 | 2 | 6 | 3 | 0 |
| 8 | 2 | 5 | 3 | 2 | 2 |
| 9 | 6 | 5 | 2 | 1 | 0 |
| 10 | 0 | 2 | 2 | 3 | 7 |
| Total | 20 | 28 | 39 | 21 | 32 |
| p_j | 0.143 | 0.200 | 0.279 | 0.150 | 0.229 |

以上是14个评价者对10个item进行5级评价的结果（ $N = 10$ ， $n = 14$ ， $k = 5$ ）。

则计算Fleiss Kappa相关系数的过程为：

step 1. 对每一列计算 p_j ，即同列数据相加除以任务总数（ $14*10=140$ ）。 p_j 可以理解为每个分类的随机一致概率。

$$\text{以第一列为例，则 } p_1 = \frac{0+0+0+0+2+7+3+2+6+0}{14*10} = 0.143$$

Fleiss' kappa(续)

| n_{ij} | 1 | 2 | 3 | 4 | 5 | P_i |
|----------|-------|-------|-------|-------|-------|-------|
| 1 | 0 | 0 | 0 | 0 | 14 | 1.000 |
| 2 | 0 | 2 | 6 | 4 | 2 | 0.253 |
| 3 | 0 | 0 | 3 | 5 | 6 | 0.308 |
| 4 | 0 | 3 | 9 | 2 | 0 | 0.440 |
| 5 | 2 | 2 | 8 | 1 | 1 | 0.330 |
| 6 | 7 | 7 | 0 | 0 | 0 | 0.462 |
| 7 | 3 | 2 | 6 | 3 | 0 | 0.242 |
| 8 | 2 | 5 | 3 | 2 | 2 | 0.176 |
| 9 | 6 | 5 | 2 | 1 | 0 | 0.286 |
| 10 | 0 | 2 | 2 | 3 | 7 | 0.286 |
| Total | 20 | 28 | 39 | 21 | 32 | |
| p_j | 0.143 | 0.200 | 0.279 | 0.150 | 0.229 | |

Step 2. 计算 $P_i = \frac{1}{n(n-1)} (\sum_{j=1}^k n_{ij}^2 - n)$, 即对每一个标注任务进行实际一致性的计算,

以第2个item为例: $P_2 = \frac{1}{14(14-1)} (0^2 + 2^2 + 6^2 + 4^2 + 2^2 - 14) = 0.253$

Fleiss' kappa (续)

| n_{ij} | 1 | 2 | 3 | 4 | 5 | P_i |
|----------|-------|-------|-------|-------|-------|-------|
| 1 | 0 | 0 | 0 | 0 | 14 | 1.000 |
| 2 | 0 | 2 | 6 | 4 | 2 | 0.253 |
| 3 | 0 | 0 | 3 | 5 | 6 | 0.308 |
| 4 | 0 | 3 | 9 | 2 | 0 | 0.440 |
| 5 | 2 | 2 | 8 | 1 | 1 | 0.330 |
| 6 | 7 | 7 | 0 | 0 | 0 | 0.462 |
| 7 | 3 | 2 | 6 | 3 | 0 | 0.242 |
| 8 | 2 | 5 | 3 | 2 | 2 | 0.176 |
| 9 | 6 | 5 | 2 | 1 | 0 | 0.286 |
| 10 | 0 | 2 | 2 | 3 | 7 | 0.286 |
| Total | 20 | 28 | 39 | 21 | 32 | |
| p_j | 0.143 | 0.200 | 0.279 | 0.150 | 0.229 | |

Step 3. 计算 p_o 和 p_e :

$$p_o = \frac{1}{N} \sum_{i=1}^N P_i = \frac{1}{10} (1.000 + 0.253 + \dots + 0.286 + 0.286) = \frac{1}{10} \cdot 3.780 = 0.378$$

$$p_e = \sum_{j=1}^k p_j^2 = 0.143^2 + 0.200^2 + 0.279^2 + 0.150^2 + 0.229^2 = 0.213$$

Step 4. 最后计算Fleiss Kappa系数

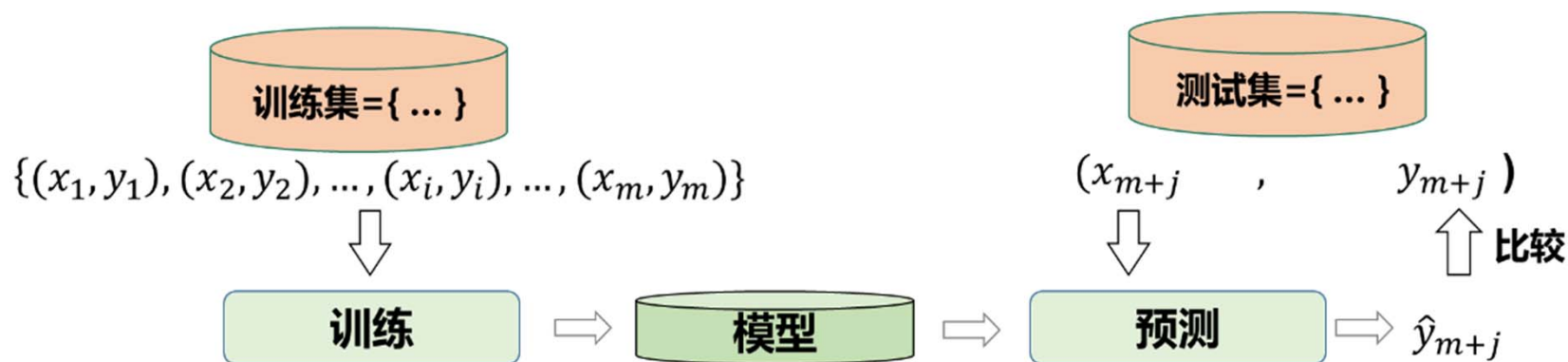
$$k = \frac{p_o - p_e}{1 - p_e} = \frac{0.378 - 0.213}{1 - 0.213} = 0.210$$

哈尔滨工业大学计算机学院 刘远超

数据集

28

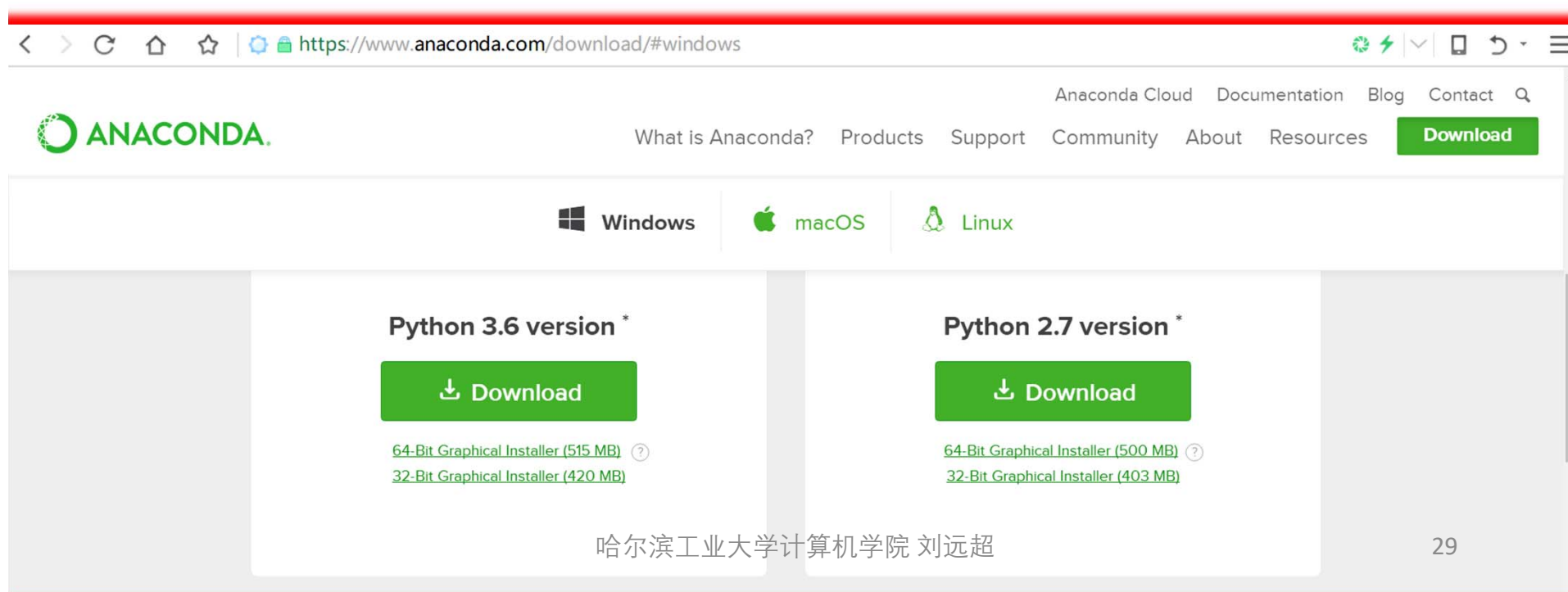
数据集的拆分问题



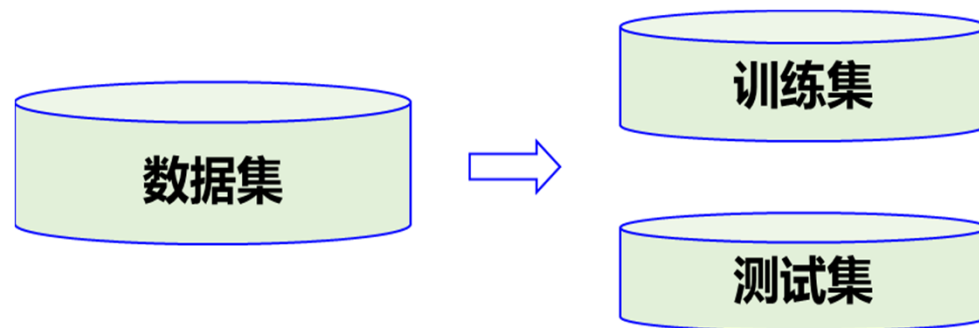
- 有监督学习中数据通常分成训练集、测试集两部分。
 - 训练集(training set)用来训练模型，即被用来学习得到系统的参数取值。
 - 测试集(testing set)用于最终报告模型的评价结果，因此在训练阶段测试集中的样本应该是unseen的。
- 有时对训练集做进一步划分为训练集和验证集(validation set)。验证集与测试集类似，也是用于评估模型的性能。区别是验证集主要用于模型选择和调整超参数，因而一般不用于报告最终结果。

训练集、测试集的拆分

- 可以使用sklearn (即scikit-learn) 进行训练集、测试集的拆分。
- 如何安装sklearn: anaconda是一个开源的Python发行版本，其包含了很多科学包及其依赖项，也包含sklearn。
 - 网址: <https://www.anaconda.com/download/#windows>



训练集测试集拆分—留出法



- 留出法（**Hold-Out Method**）数据拆分步骤：
 1. 将数据随机分为两组，一组做为训练集，一组做为测试集
 2. 利用训练集训练分类器，然后利用测试集评估模型，记录最后的分类准确率为此分类器的性能指标
- 留出法的优点是处理简单。而**不足之处**是在测试集上的预测性能的高低与数据集拆分情况有很大的关系，所以基于这种数据集拆分基础上的性能评价结果不够稳定。

K折交叉验证



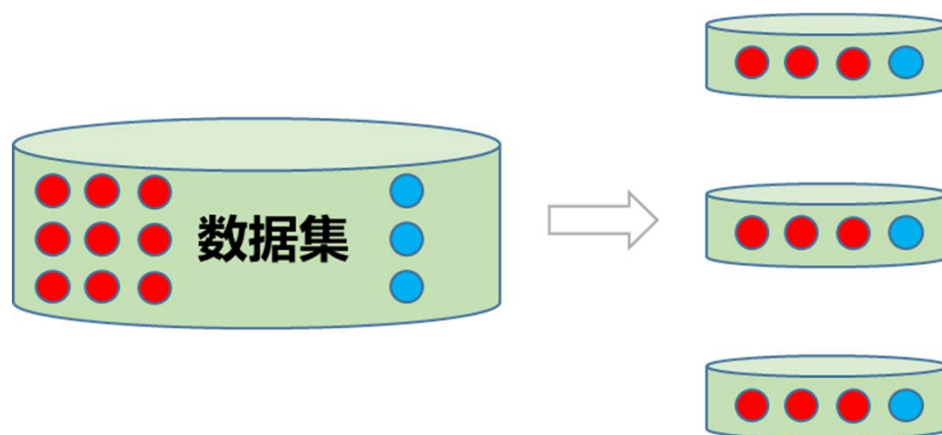
●过程:

1. 数据集被分成K份 (K通常取5或者10)
2. 不重复地每次取其中一份做测试集, 用其他K-1份做训练集训练, 这样会得到K个评价模型
3. 将上述步骤2中的K次评价的性能均值作为最后评价结果

● K折交叉验证的上述做法有助于提高评估结果的稳定性

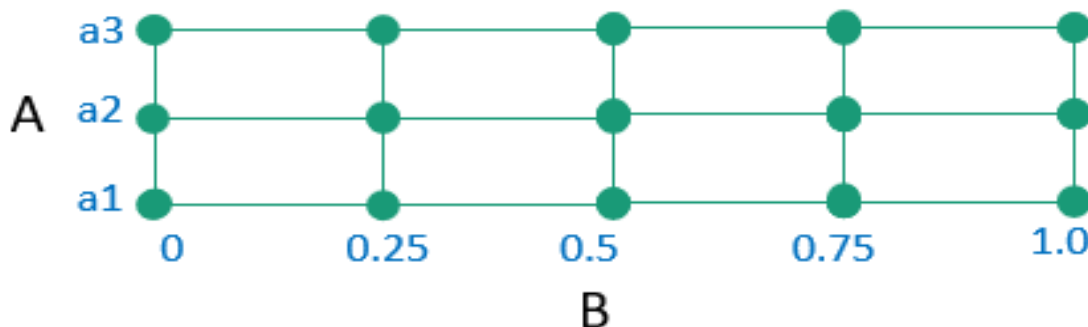
分层抽样策略 (Stratified k-fold)

- 将数据集划分成k份，特点在于，划分的k份中，每一份内各个类别数据的比例和原始数据集中各个类别的比例相同。

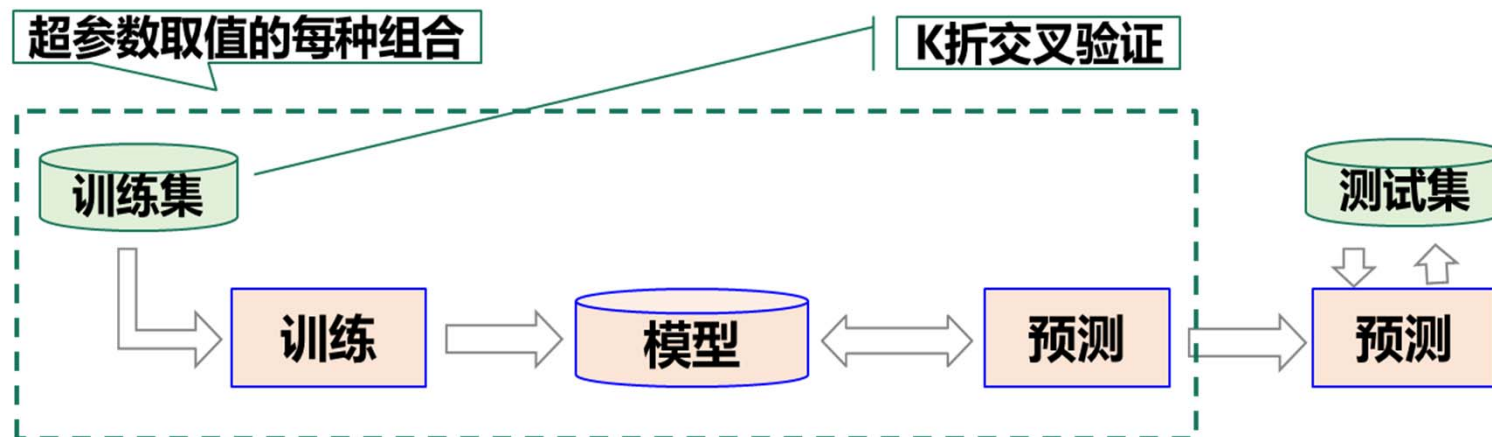


用网格搜索来调超参数 (一)

- **什么是超参数?** 指在学习过程之前需要设置其值的一些变量, 而不是通过训练得到的参数数据。如深度学习中的学习速率等就是超参数。
- **什么是网格搜索?**
 - 假设模型中有2个超参数: A和B。A的可能取值为{a1, a2, a3}, B的可能取值为连续的, 如在区间[0-1]。由于B值为连续, 通常进行离散化, 如变为{0, 0.25, 0.5, 0.75, 1.0}
 - 如果使用网格搜索, 就是尝试各种可能的(A, B)对值, 找到能使的模型取得最高性能的(A, B)值对。



用网格搜索来调超参数 (二)



网格搜索与K折交叉验证结合调整超参数的具体步骤：

1. 确定评价指标；
2. 对于超参数取值的每种组合，在训练集上使用交叉验证的方法求得其K次评价的性能均值；
3. 最后，比较哪种超参数取值组合的性能最好，从而得到最优超参数的取值组合。

机器学习概述

本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- 机器学习方法的分类
- 半监督学习
- 主动学习
- 排序学习

机器学习概述

本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- **机器学习方法的分类**
- 半监督学习
- 主动学习
- 排序学习

机器学习方法的分类

37

按样本标签的情况分类

| 分类名称 | 样本标签的特点 | 学习任务特点 |
|-----------------|----------------------|--|
| 有监督学习 | 训练集中的样本有标签 | 学习输入到输出的映射 |
| 无监督学习 | 训练集中的样本无标签 | 学习、揭示数据的内在性质及规律 |
| 纯的半监督学习 | 训练集中部分样本有标签，其余无标签 | 不依赖人工干预,将无标签样本集中分类（或回归）置信度高的样本自动加入到训练集中，以提升模型的泛化性能 |
| 特殊的半监督学习，即直推式学习 | 训练集样本有标签，测试集样本特征信息可见 | 训练时利用了测试集除了标签以外的有用信息（例如基于特征的样本相似度或者连接关系等） |
| 主动学习 | 训练集中部分样本有标签，其余无标签 | 依赖人工干预。将无标签样本集中分类（或回归）置信度低的样本交由人工专家标注并加入到训练集中，以提升模型的泛化性能 |
| 强化学习 | 数据标签未知，但知道与输出目标相关的反馈 | 适用决策类问题 |
| 排序学习 | 不关心样本标签 | 学习给定查询情况下样本的排序 |

机器学习概述

38

本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- 机器学习方法的分类
- 半监督学习
- 主动学习
- 元学习
- 排序学习

机器学习概述

39

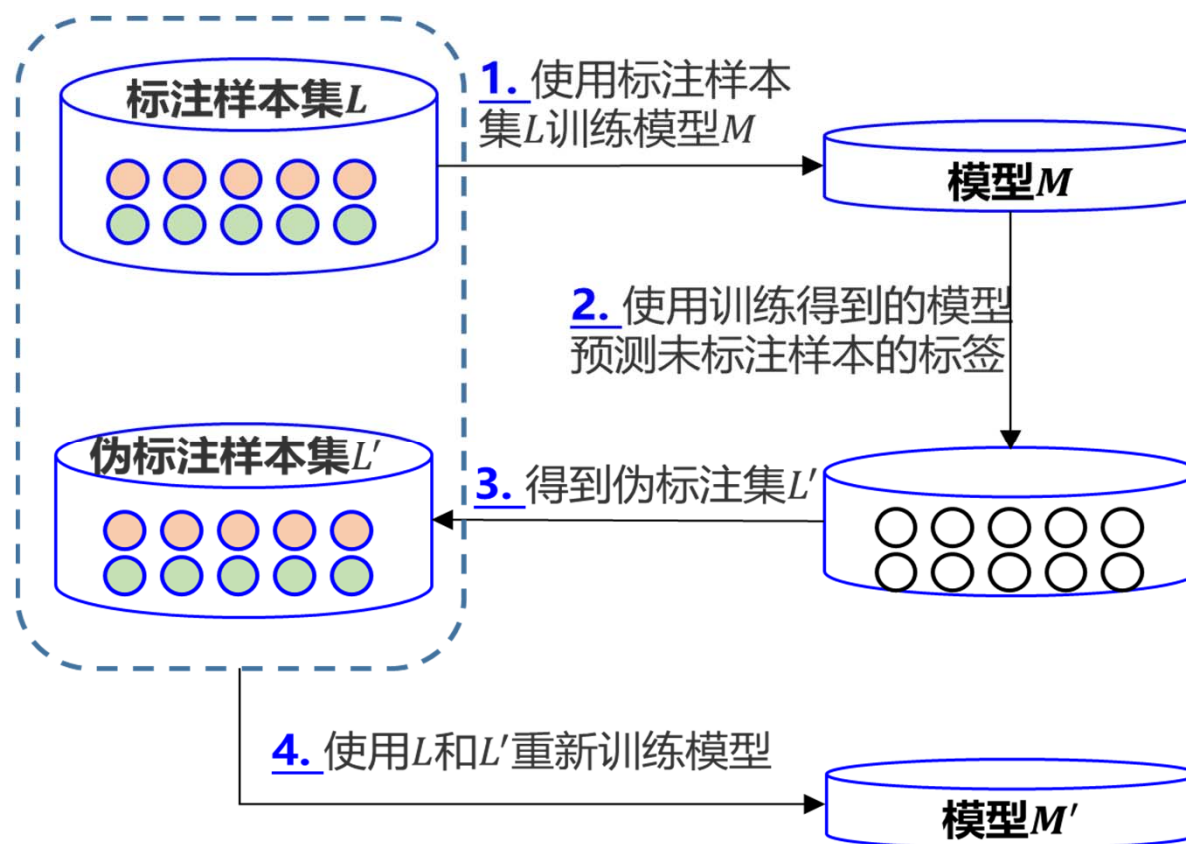
本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- 机器学习方法的分类
- **半监督学习**
- 主动学习
- 排序学习

半监督学习举例

40

简单自训练学习



半监督学习

41

简单自训练学习

算法 2. 1: 半监督学习中的简单自训练算法 (Self-training)

Input: 有标签的样本集 (X_l, y_l) , 没有标签的样本集 X_u

Output: 扩充后的训练集 (X_{train}, y_{train})

- 1 给定 $(X_{train}, y_{train}) = (X_l, y_l)$
 - 2 **while** 没有满足停止准则 **do**
 - 3 从 (X_{train}, y_{train}) 训练分类器 C_{int}
 - 4 使用 C_{int} 预测 X_u 的类别标签 y_u
 - 5 从 (X_u, y_u) 中选择置信度较高的样本 (X_{conf}, y_{conf})
 - 6 从 (X_u, y_u) 中移除未能加上标签的样本: $X_u \leftarrow X_u - X_{conf}$
 - 7 //将新打上标签的样本加入到训练集中:
 - 8 $(X_{train}, y_{train}) \leftarrow (X_l, y_l) \cup (X_{conf}, y_{conf})$
 - 9 **end**
-

机器学习概述

42

本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- 机器学习方法的分类
- 半监督学习
- 主动学习
- 排序学习

机器学习概述

43

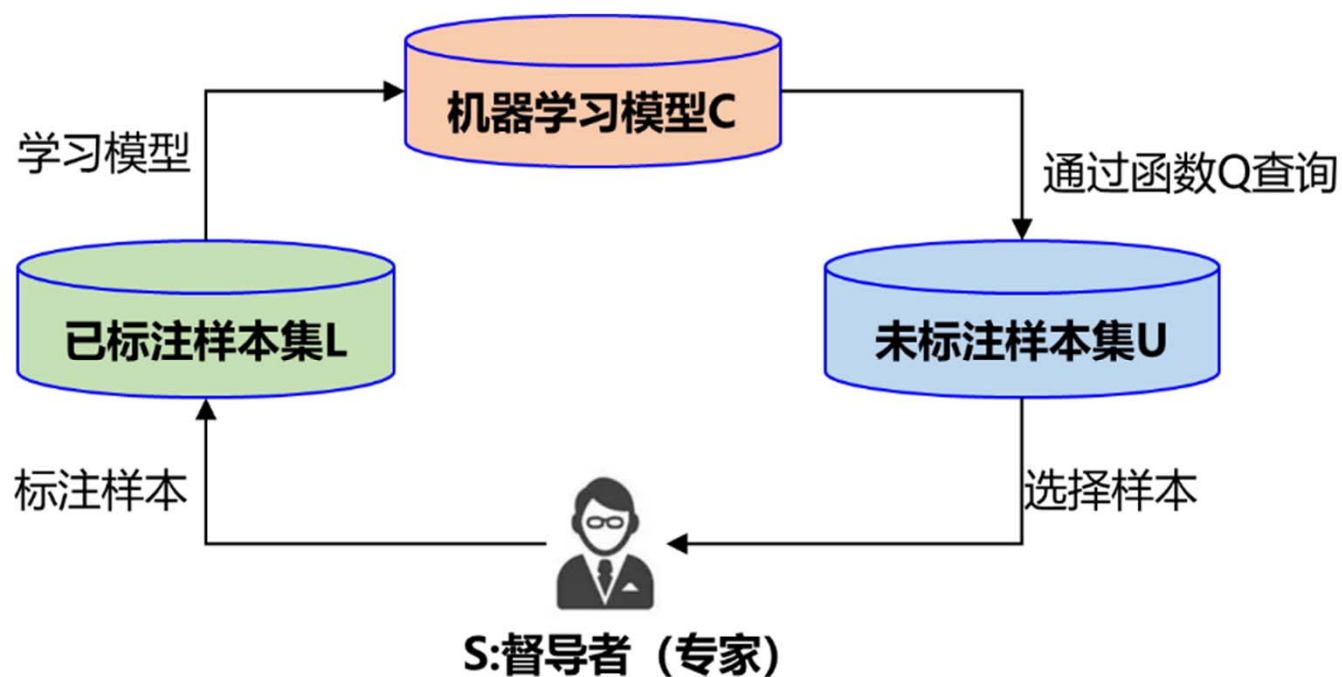
本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- 机器学习方法的分类
- 半监督学习
- 主动学习
- 排序学习

主动学习

44

主动学习的原理示意图



主动学习

45

主动学习的查询策略Q(1)--基于不确定性采样的查询策略

主动学习的关键在于如何选择出合适的样本交由人类专家标注，选择的方法即查询策略Q是主动学习的关键之处。典型的策略包括基于不确定性采样的查询策略（Uncertainty Sampling）、基于委员会的查询策略（Query-By-Committee）等。

其中不确定性采样策略是指将模型中难以区分的样本数据提取出来，由人类专家进行标注。可以描述样本不确定性的方法有：

1) **最低置信度（Least Confident）方法**。其认为那些在不同类别的**概率分布中最大概率最小的样本**更难以区分。例如，三分类中假设两个样本的类别概率分布分别为（0.8、0.1、0.1）和（0.4, 0.4,0.2）。则第二个样本更难以被区分，因为其最大概率为0.4，小于第一个样本的最大概率值0.8，因此更有必要交由人类专家继续标注；

2) **边缘采样（Margin Sampling）方法**。选择模型预测的类别概率分布中**最大概率和次大概率的差值最小的样本**。对于上面的例子，则第二个样本同样也将被选中，因为其最大概率和次大概率的差值为 $0.4 - 0.4 = 0$ ；

3) **信息熵（Information Entropy）方法**。信息论中熵被用来衡量一个系统的不确定性，熵越大则表示不确定性越大，熵越小表示不确定性越小。因此可以对某个样本的概率分布的熵进行计算，并选择熵比较大的样本交由人类专家标注。

主动学习

46

主动学习的查询策略Q(2)--基于委员会的查询策略

另外一种基于委员会的查询（Query-By-Committee）策略是指采用类似集成学习的方法，通过多个模型投票的模式，来选出那些较难区分的样本。**基于委员会的查询策略**通常也包括两种方法：

1) **投票熵（Vote Entropy）**。对于某一个样本，可以考察其使用多个不同分类器的分类结果。**如果这些结果都一致，则认为该样本容易被区分**；但如果这些结果差距较大，则表示该样本难以区分。不同分类器的分类结果也可以用熵来衡量，称之为投票熵；

2) **平均 KL 散度**。即用 KL 散度找到那些不同分类器下的**分类概率分布 KL 散度较大的样本**。

机器学习概述

47

本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- 机器学习方法的分类
- 半监督学习
- 主动学习
- 排序学习

机器学习概述

48

本章内容简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- 机器学习方法的分类
- 半监督学习
- 主动学习
- 排序学习

排序学习

49

什么是排序学习

■ 什么是排序学习？

- 排序学习中，训练集和测试集由带有顺序关系的样本列表组成。这种顺序通常根据每个样本的分值来给出。
- 排序学习的任务是，利用机器学习的方法得到模型，对新的之前未见的样本列表（测试集）给出预测的排序，排序的方式尽可能接近测试集中的实际排序。

- **排序学习的应用领域**：在搜索引擎检索结果排序、推荐系统候选产品排序、社交网络中用户排序等问题中具有广泛应用。

排序学习

50

排序学习有三类方法

(一) 单点法 (Pointwise)

- 以信息检索为例，单点法假设训练集中的每个查询-文档对都有一个相关性分值。从而构造出类似下表的数据集。
- 表中，其中每行为一个查询-文档对，可以看作一条样本，中间各列为查询-文档对的各项特征，最后一列为该查询-文档对之间的相关性分值（目标值 y ）。

表2.6 排序学习的数据集示例

| 查询-文档对 | 余弦相似度 | BM25相似度 | 网页Pagerank值 | | 相关性分值 |
|-------------|-------|---------|-------------|-------|-------|
| $Q_1 - D_1$ | 0.15 | 0.15 | 6 | | 0 |
| $Q_1 - D_2$ | 0.30 | 0.35 | 4 | | 1 |
| $Q_1 - D_3$ | 0.65 | 0.68 | 3 | | 1 |
| $Q_2 - D_4$ | 0.55 | 0.57 | 3 | | 2 |
| $Q_2 - D_5$ | 0.64 | 0.45 | 5 | | 3 |

排序学习

51

排序学习有三类方法

(二) 配对法 (Pairwise)

- 相比之下，配对法将排序学习问题近似为一个分类问题，**即学习一个二类分类器，判断相对于同一查询Q，一对给定的文档中哪个更好。**
- 同样以信息检索为例，对于与同一查询Q相关的文档集合中的任何两个具有不同相关性分值的文档 D' 和 D'' ，可以构造一个训练样本实例：该样本实例的特征可以参照表2.6给出；如果 D' 与Q的相关性分值大于 D'' ，则该样本实例的标签赋值为1；反之为0。从而可以构造出二类分类器训练所需的训练样本和测试样本。
- 在测试阶段，通过对测试集中所有的文档对进行分类，可以得到与同一查询Q相关的所有文档的偏序关系，可以进一步处理以实现排序。

表2.6 排序学习的数据集示例

| 查询-文档对 | 余弦相似度 | BM25相似度 | 网页Pagerank值 | | 相关性分值 |
|-------------|-------|---------|-------------|-------|-------|
| $Q_1 - D_1$ | 0.15 | 0.15 | 6 | | 0 |
| $Q_1 - D_2$ | 0.30 | 0.35 | 4 | | 1 |
| $Q_1 - D_3$ | 0.65 | 0.68 | 3 | | 1 |
| $Q_2 - D_4$ | 0.55 | 0.57 | 3 | | 2 |
| $Q_2 - D_5$ | 0.64 | 0.45 | 5 | | 3 |

排序学习

52

排序学习有三类方法

(三) 列表法 (Listwise)

- 与单点法和配对法不同，列表法直接考虑给定查询下的文档集合的整体序列，使得其尽可能接近真实文档序列。
- 对于查询 Q ，假设与之对应的文档集合为 $D = \{D_1, D_2, \dots, D_m\}$ ，且这些文档的真实标签值集合为 $Y = \{y_1, y_2, \dots, y_m\}$ ，则可以通过如下方式预测每个文档的排序分值：

$$score_i = h_{\theta}(Q, D_i) \quad (2.7)$$

$$S = softmax([score_1, score_2, \dots, score_m]) \quad (2.8)$$

其中 h_{θ} 表示参数为 θ 的模型。因此可以通过利用最小化 S 与 Y 之间的交叉熵损失来优化模型中的参数 θ 。

排序学习

53

排序结果的评价方法

(一) 平均排序倒数 (Mean Reciprocal Rank, 简称MRR)

- MRR是一种简单的排序评价方法。以信息检索为例，MRR的基本思想是认为排序结果的评价与最相关的文档（正确答案）的位置有关，即如果排序结果中最相关文档的位置越靠前，则说明排序算法的预测排序结果越好。具体地：对于某查询，若最相关文档排在第 n 个位置，则对该查询而言，MRR得分即为 $1/n$ 。（如果预测的排序结果中没有出现最相关文档，则该查询对应的MRR得分为0）。因此对于给定的查询集合 Q ，则MRR的计算公式为

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \quad (2.9)$$

其中， Q 为查询集合， $|Q|$ 表示 Q 中的查询个数， $rank_i$ 表示排序算法对第 i 个查询给出的排序文档集中最相关文档的排序位置。

- 例如，假设测试集中共有4个查询，在排序算法返回的文档排序结果中，前三个查询的最相关文档在各自文档列表中的排序位置分别为3、1、5，并且排序算法没能在返回列表中给出第4个查询的最相关文档。则该排序算法的MRR得分计算为：
 $MRR = (1/3 + 1/1 + 1/5 + 0) \div 4 = 0.383$ 。

排序学习

54

排序结果的评价方法

(二) 归一化折扣累积增益 (Normalized Discounted cumulative gain, 简称NDCG)

举个例子：表2.7为对于某个查询，排序算法返回的文档排序结果（第一行），以及人类专家给出的每个文档对该查询的相关性分值（参考真值）。其中0表示不相关，1和2 表示中间值，3表示完全相关。针对这个例子，NDCG是如何计算的呢？

表2.7 对于某个查询，算法给出的文档排序及专家打分的对比示例

| 排序算法给出的排序 | D1 | D2 | D3 | D4 | D5 |
|-------------------|----|----|----|----|----|
| 位置索引 i | 1 | 2 | 3 | 4 | 5 |
| 相关性分数参考真值 rel_i | 2 | 3 | 1 | 0 | 3 |

排序学习

55

排序结果的评价方法

在介绍NDCG之前，先了解一个铺垫性的排序指标：

1) 第一步，计算累积增益 (Cumulative Gain, 简称CG)。CG衡量的是返回结果的整体相关性，即排序结果中每个条目的相关性分值的和。具体地，前K个排序位置的CG计算为

$$CG@K = \sum_{i=1}^K rel_i \quad (2.10)$$

其中 rel_i 是位置*i*的相关性分数参考值。

因此，对于表2.7所示例子中的数据，则有

$$CG@5 = \sum_{i=1}^5 rel_i = 2 + 3 + 1 + 0 + 3 = 9 \quad (2.11)$$

可见，CG没有考虑排序结果中各个条目的位置信息。

排序学习

56

排序结果的评价方法

2) 第二步, 计算折扣累积增益 (Discounted Cumulative Gain, DCG)。DCG考虑到了每条结果的位置信息。DCG认为与查询高度相关但却被算法排在靠后位置的条目应该被惩罚。Top K排序的DCG 计算如下:

$$DCG@K = \sum_{i=1}^K \frac{rel_i}{\log_2(i+1)} = rel_1 + \sum_{i=2}^K \frac{rel_i}{\log_2(i+1)} \quad (2.11)$$

则对于表2.7给出的示例, $DCG@K$ 计算中间过程如表2.8中逐行所示。

| 排序算法给出的排序 | D1 | D2 | D3 | D4 | D5 |
|-----------------------------|----|-------|-----|-------|-------|
| 位置索引 i | 1 | 2 | 3 | 4 | 5 |
| $\log_2(i+1)$ | 1 | 1.585 | 2 | 2.322 | 2.585 |
| 相关性分数参考真值 rel_i | 2 | 3 | 1 | 0 | 3 |
| $\frac{rel_i}{\log_2(i+1)}$ | 2 | 1.893 | 0.5 | 0 | 1.161 |

因此, 最终得到 $DCG@5 = \sum_{i=1}^5 \frac{rel_i}{\log_2(i+1)} = 2 + 1.893 + 0.5 + 0 + 1.161 = 5.553$ 。

排序学习

57

排序结果的评价方法

3) 第三步, 计算IDCG (Ideal DCG), 即理想排序结果。理想的排序结果是人类专家给出的条目相关性分值的单调递减排序。本例中为: 3、3、2、2、1、0。表2.9逐行给出了IDCG计算的中间过程。

| 理想的排序结果 | D1 | D3 | D2 | D5 | D4 |
|-------------------------------|----|-------|----|-------|-------|
| i | 1 | 2 | 3 | 4 | 5 |
| $\log_2(i + 1)$ | 1 | 1.585 | 2 | 2.322 | 2.585 |
| 相关性分数 rel_i | 3 | 3 | 2 | 1 | 0 |
| $\frac{rel_i}{\log_2(i + 1)}$ | 3 | 1.893 | 1 | 0.431 | 0 |

因此 $IDCG@5 = \sum_{i=1}^5 \frac{rel_i}{\log_2(i+1)} = 3 + 1.893 + 1 + 0.431 + 0 = 6.323$ 。

4) 第四步, 则最终归一化的折扣累积增益计算结果为 $nDCG@5 = \frac{DCG_5}{IDCG_5} = \frac{5.553}{6.323} = 0.878$ 。

机器学习概述

58

本章小结简介

- 深度学习的硬件计算环境
- 深度学习的软件计算环境
- 数据集
- 机器学习方法的分类
- 半监督学习
- 主动学习
- 排序学习