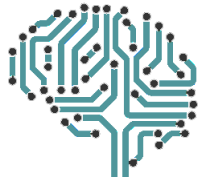
暨南大學
JINAN UNIVERSITY



机 器 学 习

第三章：模型评估

黄斐然

2022/3/7

机器学习的数据

- 数据集：数据整体叫数据集（dataset）。
- 样本：每一行数据可以称为一个样本。
- 特征：每一列表示数据的一个特征，数据的特征的向量的集合可用X表示。
- 标签：最后一列称为标签（label），数据标签集可以用Y表示。
- 特征向量：每一条样本的特征组成的向量称为特征向量。

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	label
126	6.2	2.8	4.8	1.8	2
60	5.0	2.0	3.5	1.0	1
31	5.4	3.4	1.5	0.4	0
87	6.3	2.3	4.4	1.3	1
125	7.2	3.2	6.0	1.8	2
42	4.4	3.2	1.3	0.2	0
131	7.9	3.8	6.4	2.0	2
22	4.6	3.6	1.0	0.2	0

X　　y

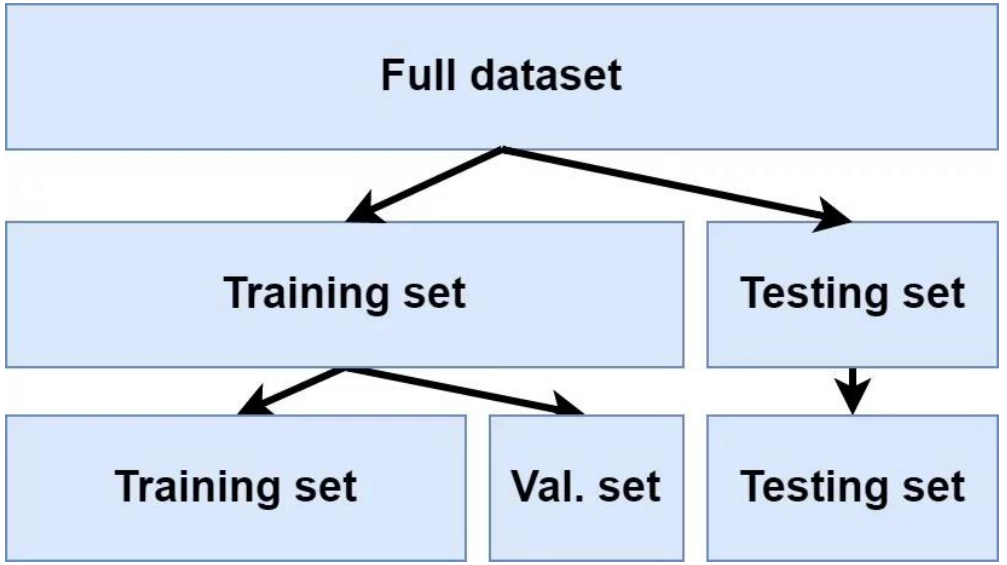
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	label
126	6.2	2.8	4.8	1.8	2
60	5.0	2.0	3.5	1.0	1
31	5.4	3.4	1.5	0.4	0
87	6.3	2.3	4.4	1.3	1
125	7.2	3.2	6.0	1.8	2
42	4.4	3.2	1.3	0.2	0
131	7.9	3.8	6.4	2.0	2
22	4.6	3.6	1.0	0.2	0

x_i　　y_i

机器学习的数据

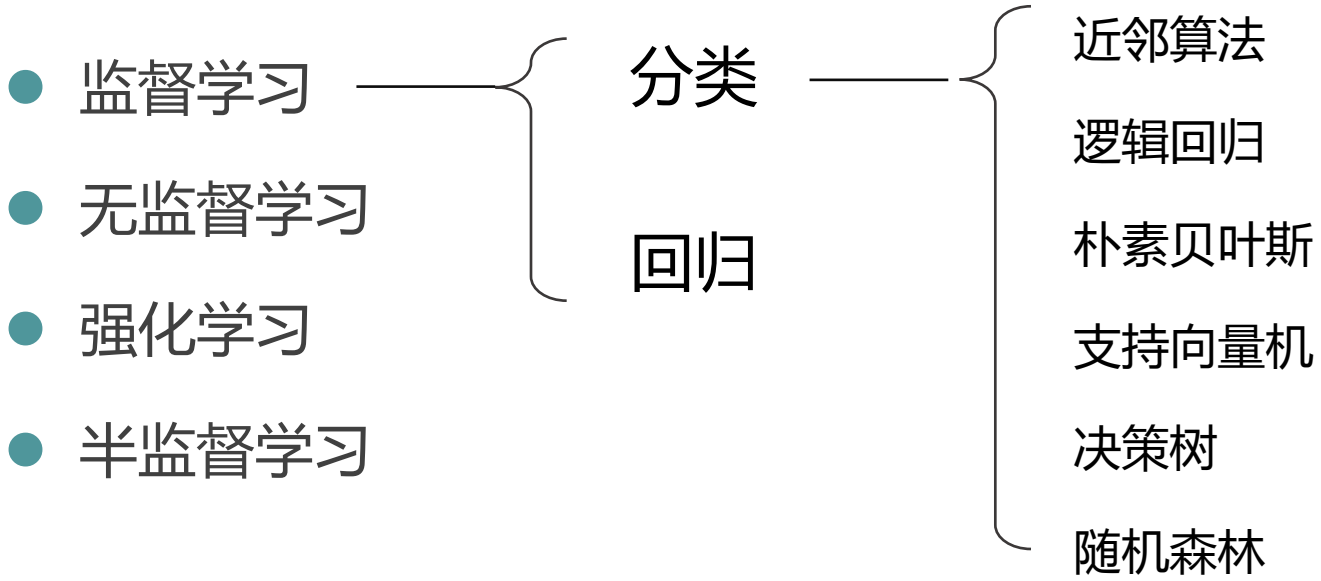
数据集分类

- 训练数据（training data）：在训练过程中使用的数据称为训练数据，每一个样例称为训练样本，全体训练样本集合称为训练集。
- 测试数据（testing data）：用于测试学习得到模型的数据称为测试数据，每一个样例称为测试样本，全体测试样本集合称为测试集。
- 验证数据（Validation data）是训练模型时所保留的数据样本，我们在调整模型超参数时，需要根据它来对模型的能力进行评估。



机器学习的学习方式

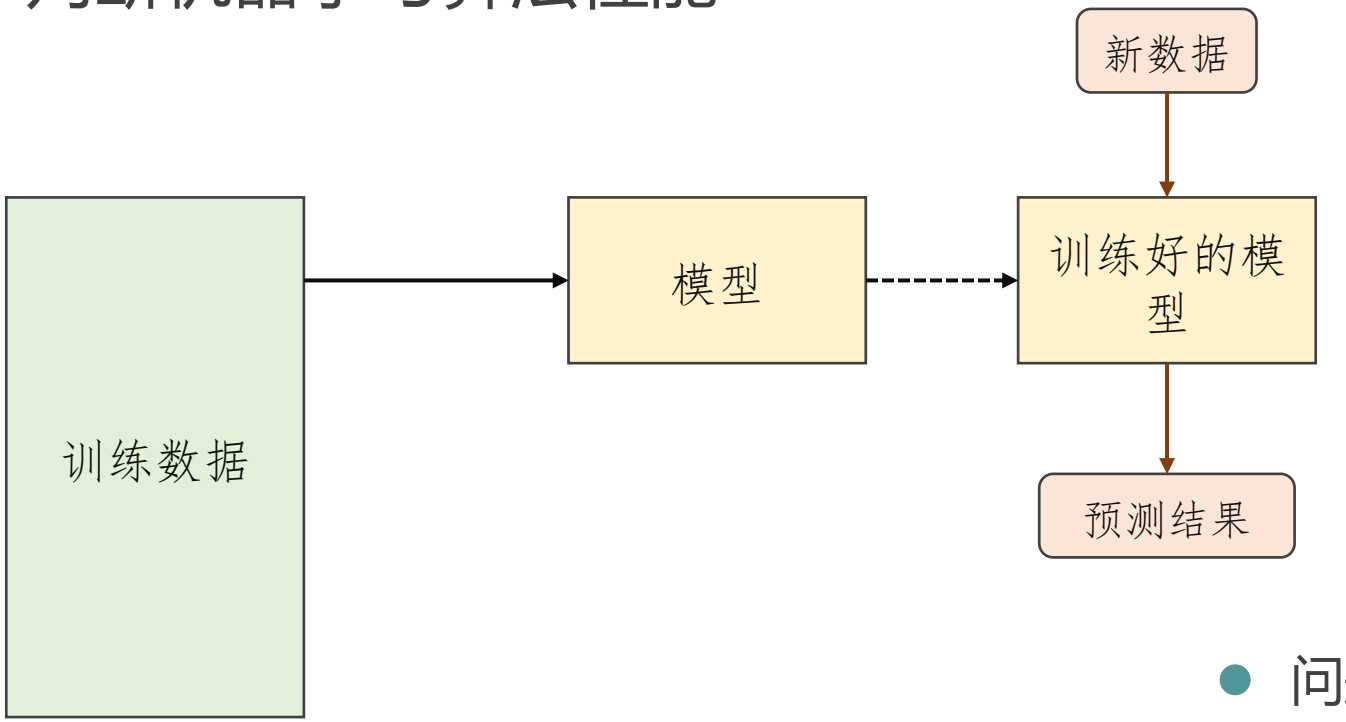
- 按照学习方式的不同，机器学习可分为很多的类型：



目录

1 数据分割

判断机器学习算法性能



全体数据

- 问题：如何判断算法的真实性能？
  - 在训练集上的表现不能代表真实性能
  - 真实环境很难拿到标签

1 数据分割

在训练集上的表现不能代表真实性能

- 举例

西瓜数据集

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

- 构建以下分类模型

```
def predict(color, root, knock):
    if color == “青绿” and root== “蜷缩” and knock = “浊响” : return 1
    elif color == “乌黑” and root== “蜷缩” and knock = “浊响” : return 1
    elif color == “青绿” and root== “硬挺” and knock = “清脆” : return 0
    elif color == “乌黑” and root== “稍蜷” and knock = “沉闷” : return 0
    else: return 0
```

很多复杂的模型能让其训练数据集上的表现非常好

1 数据分割

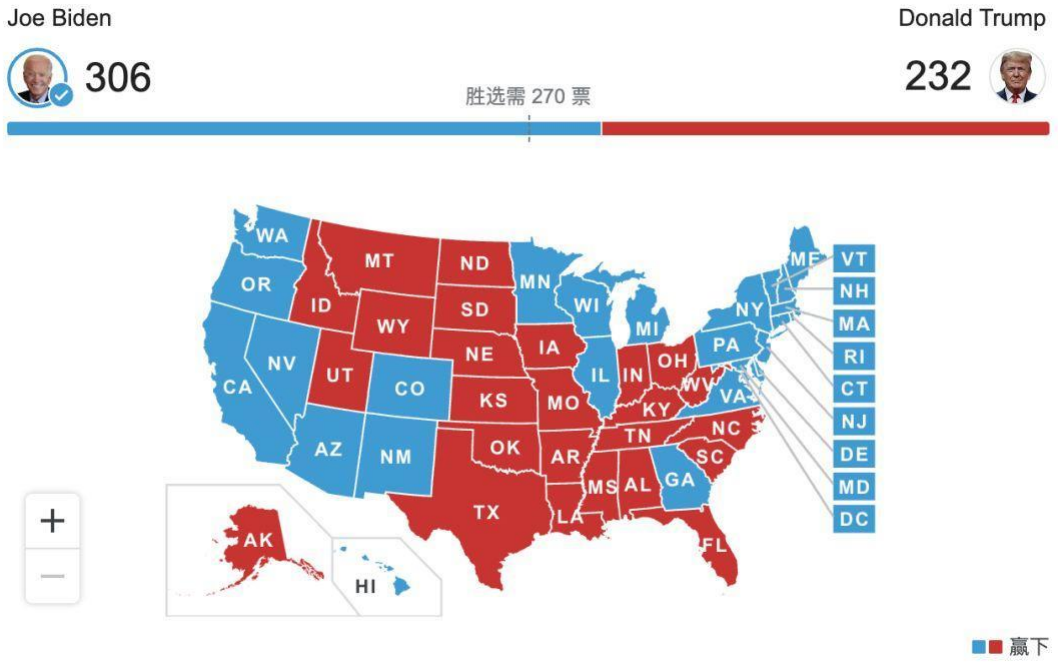
真实环境很难拿到标签

- 西瓜分类

西瓜数据集

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

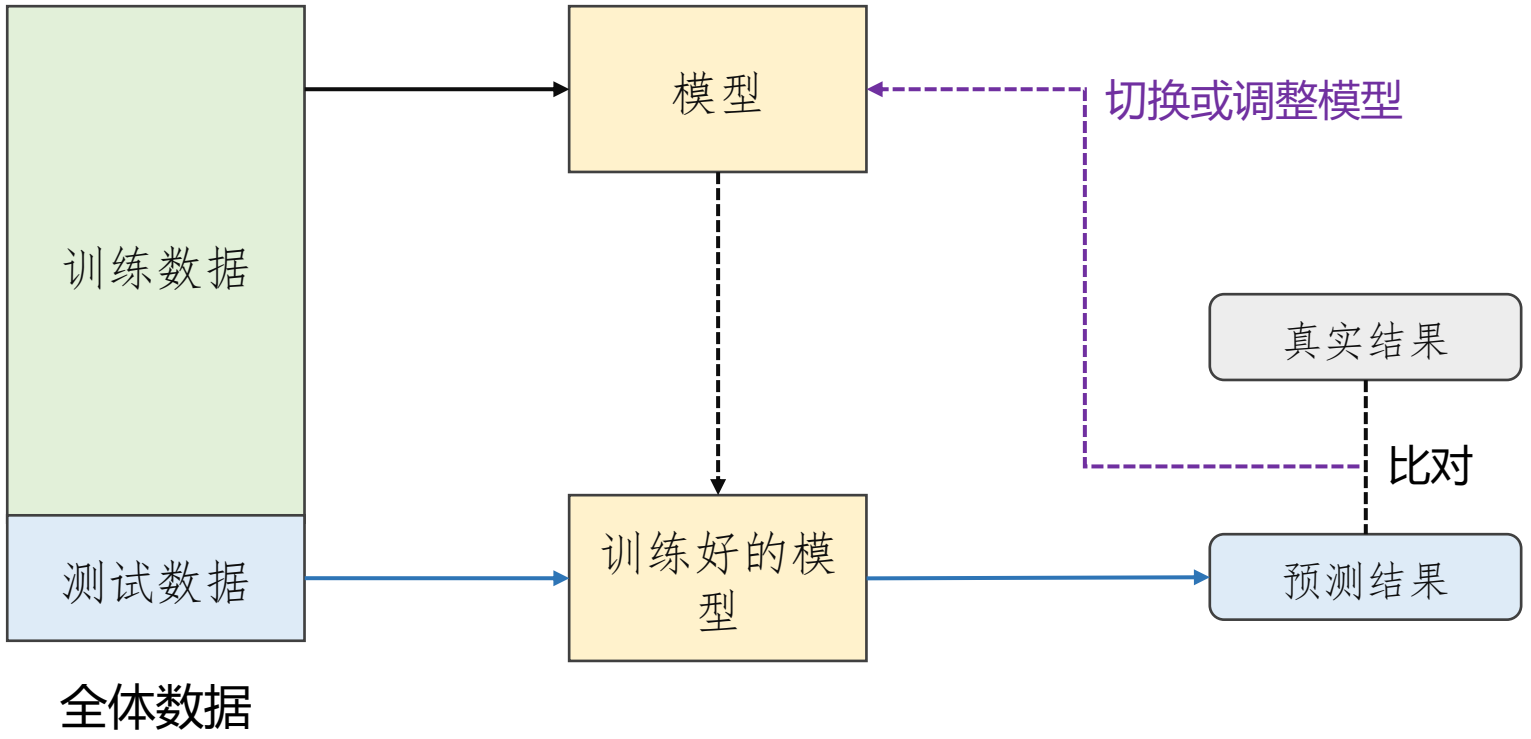
- 选举预测



无法及时有效地调整模型，造成真实的损失

1 数据分割

■ 判断机器学习算法性能



1 数据分割

留出法

- 留出法直接将数据集划分为两个互斥的部分，其中一部分作为训练集，另一部分用作测试集。
- 通常训练集的比例为70%-90%。
- 训练集测试集的划分有个注意事项：
  - 尽可能保持数据分布的一致性。避免因数据划分过程引入的额外偏差而对最终结果产生影响。在分类任务中，保留类别比例的采样方法称为“分层采样”（stratified sampling）。

1 数据分割

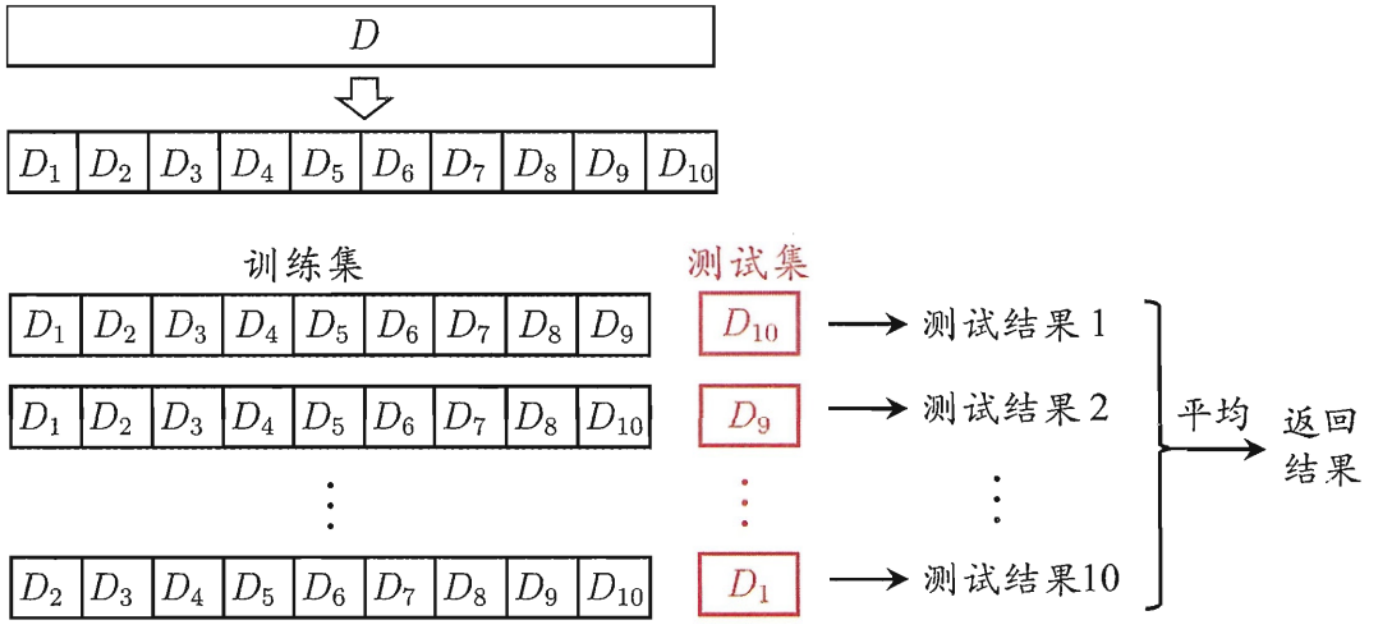
留出法

- 优点：
  - 逻辑简单
  - 运算量低，适合大型数据集
- 缺点：
  - 损失一定的样本信息
  - 对于中小型数据集，结果不稳定

1 数据分割

交叉验证法

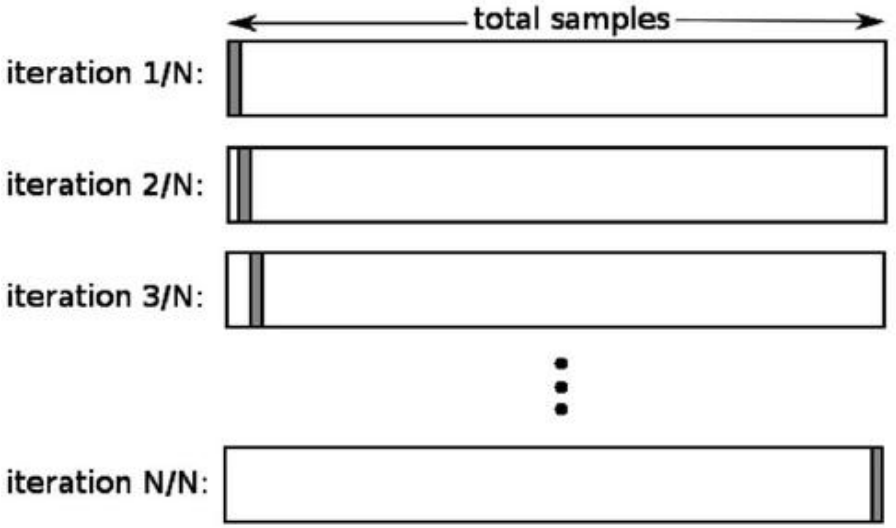
- 交叉验证法先将数据集划分为k个大小相似的互斥子集，每次采用k－1个子集的并集作为训练集，剩下的那个子集作为测试集。进行k次训练和测试，最终返回k个测试结果的均值。又称为“k折交叉验证”（k-fold cross validation）。



1 数据分割

留一法（leave-one-out）

- 留一法是k折交叉验证k = m（m为样本数）时候的特殊情况。即每次只用一个样本作测试集。该方法计算开销较大。



目录

2 模型的评价指标

- 性能量度：
  - 性能度量是衡量模型泛化能力的评判标准，性能度量反映了任务需求。
  - 使用不同的性能度量往往会导致不同的评判结果。

什么样的模型是“好”的，不仅取决于算法和数据，还取决于任务需求

泛化能力是指机器学习算法对新鲜样本的适应能力。

2 模型的评价指标

1、错误率和准确率：

- 正确率(Accuracy)是最常见也是最基本的评估标准，它表示被分对的样本数在所有样本数中的占比。
- 错误率(Error Rate)与准确率相反，它描述被分类器错分的比例。

2 模型的评价指标

- 准确率的问题：
  - 举例1：

    一个癌症预测系统，输入体检信息，可以判断是否有癌症。准确率99.5%。

  - 举例2：

    一个网络入侵流量预测系统，输入流量信息，可以判断是否是恶意流量。准确率99.9%。

举例一：全部预测为不是癌症
举例二：全部预测为正常流量

2 模型的评价指标

- 准确率的问题：
  - 对于极度偏斜的数据或类别重要性不一样的数据，只使用分类准确率是远远不够的。
  - 需要使用混淆矩阵做进一步分析。

真实值 \ 预测值	正例	负例
正例	True Positive	False Negative
负例	False Positive	True Negative

2 模型的评价指标

2、混淆矩阵：

- 混淆矩阵是机器学习中总结分类模型预测结果的情形分析表，以矩阵形式将数据集中的记录按照真实的类别与分类模型预测的类别判断两个标准进行汇总。其中矩阵的行表示真实值，矩阵的列表示预测值。

真实值 \ 预测值	正例	负例
正例	TP	FN
负例	FP	TN

TP(True Positive)：实际为正例且被分类器分为正例的个数

FP(False Positive)：实际为负例且被分类器分为正例的个数

FN(False Negative)：实际为正例且被分类器分为负例的个数

TN(True Negative)：实际为负例且被分类器分为负例的个数

2 模型的评价指标

2、混淆矩阵：

- 混淆矩阵是机器学习中总结分类模型预测结果的情形分析表，以矩阵形式将数据集中的记录按照真实的类别与分类模型预测的类别判断两个标准进行汇总。其中矩阵的行表示真实值，矩阵的列表示预测值。

真实值 \ 预测值	正例	负例
正例	TP	FN
负例	FP	TN

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

2 模型的评价指标

- 准确率的问题：
  - 肿瘤是否为恶性

真实值 \ 预测值	恶性	良性
恶性	100	100
良性	0	800

恶性肿瘤为少数

- 计算准确率：

Accuracy= (100+800)/(100+100+800)

= 90%

2 模型的评价指标

2、混淆矩阵：

- 精确率(Precision)表示被分为正例的样本中实际为正例的比例。
- 召回率(Recall)是覆盖面的度量，表示有多少个正例被分为正例

真实值 \ 预测值	正例	负例
正例	TP	FN
负例	FP	TN

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

2 模型的评价指标

2、混淆矩阵：

- 精确率(Precision)表示被分为正例的样本中实际为正例的比例。
- 召回率(Recall)是覆盖面的度量，表示有多少个正例被分为正例

真实值 \ 预测值	恶性	良性
恶性	100	100
良性	0	800

$$Precision = \frac{TP}{TP+FP} = 100/(100+0) = 100\%$$

$$Recall = \frac{TP}{TP+FN} = 100/(100+100) = 50\%$$

2 模型的评价指标

2、混淆矩阵：

- F1值(F1-score)如果综合考虑精确率(Precision)和召回率(Recall)，可以得到新的评价指标F1-score，也称为综合分类率。

真实值 \ 预测值	恶性	良性
恶性	100	100
良性	0	800

$$F1\ score = \frac{2Precision * Recall}{Precision + Recall}$$

2 模型的评价指标

2、混淆矩阵：

- F1值(F1-score)如果综合考虑精确率(Precision)和召回率(Recall)，可以得到新的评价指标F1-score，也称为综合分类率。

真实值 \ 预测值	恶性	良性
恶性	100	100
良性	0	800

$$F1\ score = \frac{2Precision * Recall}{Precision + Recall}$$

$$= 2*50\%*100\%/(50\%+100\%)$$

$$= 66.7\%$$

2 模型的评价指标

3、多分类混淆矩阵：

- 如果只有一个二分类混淆矩阵，那么用以上的指标就可以进行评价，没有什么争议，但是当我们在n个二分类混淆矩阵上要综合考察评价指标的时候就会用到宏平均和微平均。

真实值 \ 预测值	分类1	分类2	分类3
分类1	20	2	3
分类2	0	30	4
分类3	6	3	25

2 模型的评价指标

3、多分类混淆矩阵：

- 宏平均（Macro-averaging），是先对每一个类统计指标值，然后在对所有类求算术平均值。
- 微平均（Micro-averaging），是对数据集中的每一个实例不分类别进行统计建立全局混淆矩阵，然后计算相应指标。

真实值 \ 预测值	分类1	分类2	分类3
分类1	20	2	3
分类2	0	30	4
分类3	6	3	25

$$P_{macro} = \frac{1}{n}\sum_{i=1}^{n} P_i$$

$$P_{micro} = \frac{\bar{TP}}{\bar{TP} + \bar{FP}} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i}$$

2 模型的评价指标

4、均方误差：

- 均方误差是回归任务最常用的性能度量就是均方误差（mean squared error），即：
- $MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$

2 模型的评价指标

5、RMSE (均方根误差):

- MSE 公式有一个问题是会改变量纲。因为公式平方了，比如说 y 值的单位是万元，MSE 计算出来的是万元的平方，对于这个值难以解释它的含义。为了消除量纲的影响，可以使用均方根误差 RMSE，即:

- $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$

目录

3 欠拟合与过拟合

- 训练误差与泛化误差：
  - 对于深度学习或机器学习模型而言，我们不仅要求它对训练数据集有很好的拟合（训练误差），同时也希望它可以对未知数据集（测试集）有很好的拟合结果（泛化能力），所产生的测试误差被称为泛化误差。

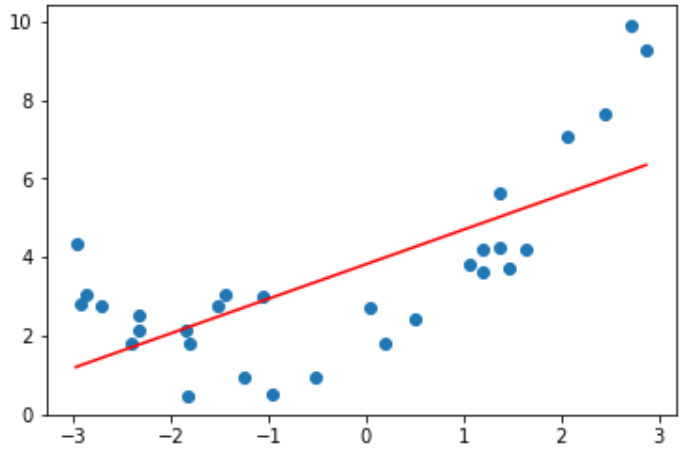
3 欠拟合与过拟合

什么是欠拟合?

- 欠拟合是指模型不能在训练集上获得足够低的误差。换句换说，就是模型复杂度低，模型在训练集上就表现很差，没法学习到数据背后的规律。

如何解决欠拟合?

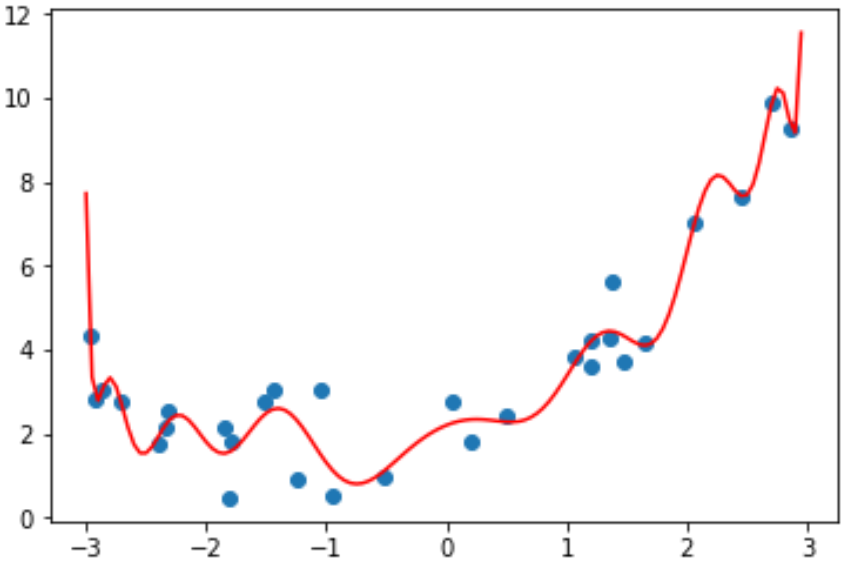
- 欠拟合基本上都会发生在训练刚开始的时候，经过不断训练之后欠拟合应该不怎么考虑了。但是如果真的还是存在的话，可以通过增加网络复杂度或者在模型中增加特征，这些都是很好解决欠拟合的方法。



3 欠拟合与过拟合

什么是过拟合?

- 过拟合是指训练误差和测试误差之间的差距太大。换句换说，就是模型复杂度高于实际问题，模型在训练集上表现很好，但在测试集上却表现很差。模型对训练集"死记硬背"（记住了不适用于测试集的训练集性质或特点），没有理解数据背后的规律，泛化能力差。。



3 欠拟合与过拟合

为什么会出现过拟合现象？

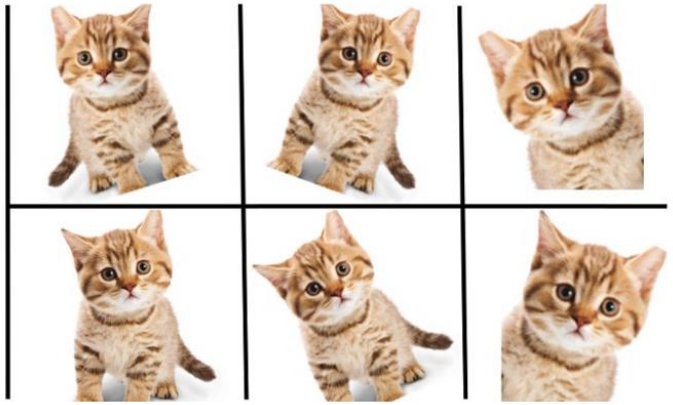
- 1、训练数据集样本单一，样本不足。如果训练样本只有负样本，然后那生成的模型去预测正样本，这肯定预测不准。所以训练样本要尽可能的全面，覆盖所有的数据类型。
- 2、训练数据中噪声干扰过大。噪声指训练数据中的干扰数据。过多的干扰会导致记录了很多噪声特征，忽略了真实输入和输出之间的关系。
- 3、模型过于复杂。模型太复杂，已经能够“死记硬背”记下了训练数据的信息，但是遇到没有见过的数据的时候不能够变通，泛化能力太差。我们希望模型对不同的模型都有稳定的输出。模型太复杂是过拟合的重要因素。

3 欠拟合与过拟合

如何防止过拟合?

1、 获取和使用更多的数据

- 这是解决过拟合的根本性方法。让机器学习或深度学习模型泛化能力更好的办法就是使用更多的数据进行训练。但是，在实践中，我们拥有的数据量是有限的。
- 解决这个问题的一种方法就是创建“假数据”并添加到训练集中——数据集增强。通过增加训练集的额外副本来增加训练集的大小，进而改进模型的泛化能力。比如对于图像，我们可以通过图像平移、翻转、缩放、切割等手段将数据库成倍扩充。

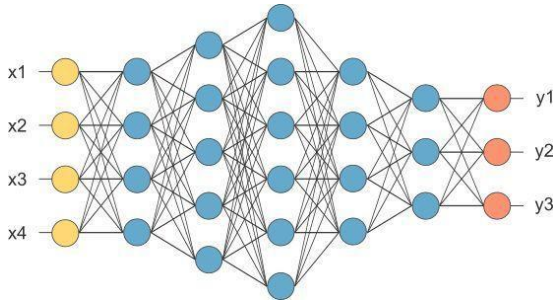


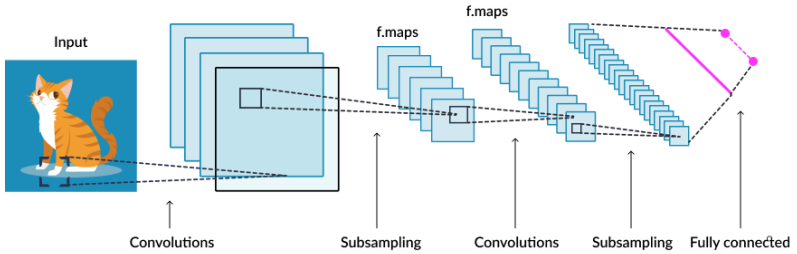
3 欠拟合与过拟合

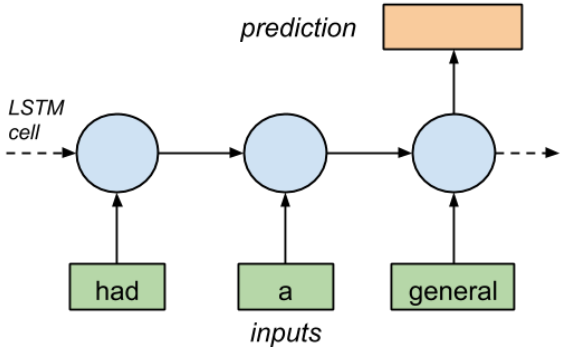
如何防止过拟合?

2、采用合适的模型（控制模型的复杂度）

- 过于复杂的模型会带来过拟合问题。对于模型的设计，目前公认的一个深度学习规律"deeper is better"。国内外各种大牛通过实验和竞赛发现，对于CNN来说，层数越多效果越好，但是也更容易产生过拟合，并且计算所耗费的时间也越长。
- 根据奥卡姆剃刀法则：在同样能够解释已知观测现象的假设中，我们应该挑选“最简单”的那一个。对于模型的设计而言，我们应该选择简单、合适的模型解决复杂的问题。







3 欠拟合与过拟合

如何防止过拟合?

3、降低特征的数量

- 对于一些特征工程而言，可以降低特征的数量——删除冗余特征，人工选择保留哪些特征。这种方法也可以解决过拟合问题。

姓名	性别	收入	学历	是否贷款
张三	男	5000	高中	是
李四	女	5500	高中	否
王五	男	2800	初中	是
陈刘	女	3300	小学	是
刘七	男	10000	初中	是

3 欠拟合与过拟合

如何防止过拟合?

4、正则化

- 正则化中我们将保留所有的特征变量，但是会通过减小参数来控制输入变量的数量级（参数数值的大小），从而减小模型的复杂度。
- 这些参数的值越小，通常对应于越光滑的函数，也就是更加简单的函数。因此 就不易发生过拟合的问题。

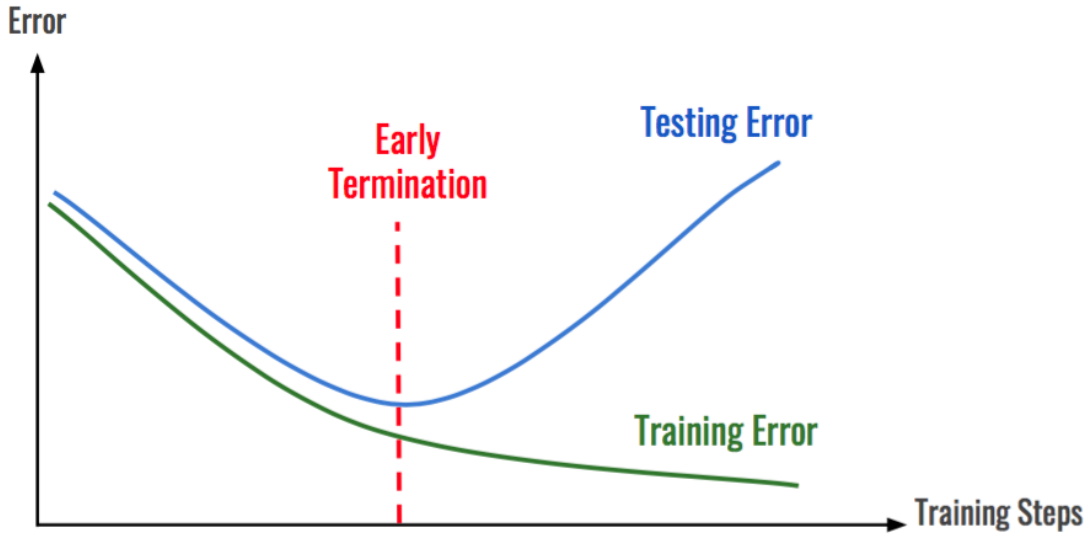
$$\text{Cost} = \underbrace{\sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2}_{\text{Loss function}} + \underbrace{\lambda \sum_{j=0}^{M} W_j^2}_{\text{Regularization Term}}$$

3 欠拟合与过拟合

如何防止过拟合?

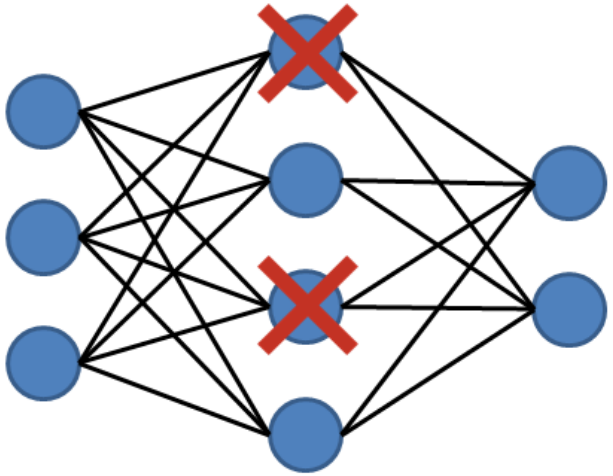
5、Early stopping（提前终止）

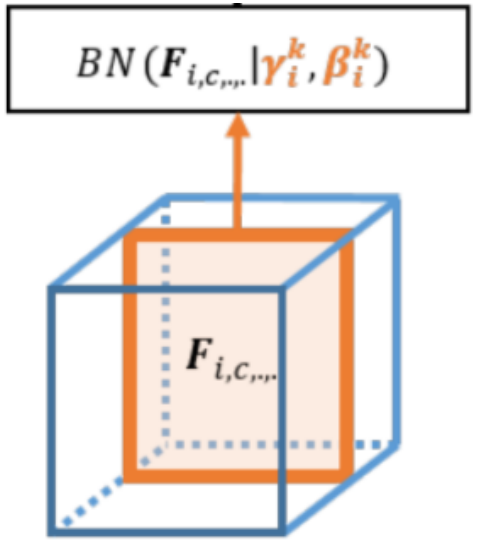
- 对模型进行训练的过程即是对模型的参数进行学习更新的过程，这个参数学习的过程往往会用到一些迭代方法，如梯度下降（Gradient descent）。Early stopping是一种迭代次数截断的方法来防止过拟合的方法，即在模型对训练数据集迭代收敛之前停止迭代来防止过拟合。



3 欠拟合与过拟合

- 如何防止过拟合?
  - 5、Dropout
  - 6、Batch Normalization





问题？

暨南大學
JINAN UNIVERSITY