第一章 机器学习简介

1. **有监督学习**
    - 有标注的数据(特征和结果)
        - 每个数据样本由多个特征和结果组成，特征和结果的值既可以是连续的，也可以是离散的
        - 我们的目标是学习特征和结果之间的函数关系，即通过特征得到结果
        - 在输入时对离散的特征值，可以使用one-hot表示法
        - 当结果是离散值时称为分类问题
        - 当结果是连续值时称为回归问题
        - *问题：当特征和结果具有顺序呢？*
    - 基本框架
        - step1: 对原始问题进行数学建模，变成一个分类问题或者回归问题。在问题输入上提取特征，得到一组特征
        - step2: 获得问题的标注数据，划分成训练集，校验集，测试集。有时候数据要经过清洗，结构化。
        - step3: 选择模型，即假设空间，我们要学习函数的基本形式。
            - 关键问题: 如何选择模型呢？
                - 学界已证明全连接神经网络能够表达任一函数，即假设空间的表达能力是可以充分满足的。但是我们为什么不直接全部用全连接神经网络来解决问题呢？
                - 首先原因是训练难度问题，全连接经常会导致参数过多，无法训练。
                - 另一个原因是合适性的问题，即对一个相对简单的问题来说假设空间太大了，没必要，甚至会导致过拟合的问题。当然过拟合问题，可以通过正则化来处理
                - 其他一些小原因，比如可解释性，通常来说决策树模型具有更好的可解释性
                - No free lunch theorem: 没有万能的模型，它对任何问题都容易训练，而且假设空间完全充分。选择模型是一个折中问题，也是个实际问题，我们需要了解问题，了解模型，从而选择适合的模型。
        - step4: 确定参数
            - 分为两类，分别是超参和可学习参数
            - 超参(这里讨论模型的超参，另外还有训练的超参)
                - 基本特征: 人为确定，人为可调整
                - 在选定模型(假设空间)之后，还是没有确定函数的形式的，还有一些参数要人为确定，这些参数称为超参，比如当我们选择全连接神经网络作为模型之后，还需要确定网络的层数，每层神经元的个数等。当然超参和基本模型之间是没有严格划分的，比如当我们选择三层全连接神经网络作为基本模型，那么网络的层数就不是超参，因为模型中已经确定了。超参是确定基本模型之后，还没确定的参数，不能通过训练学习，训练前需要人为确定，训练后可以人为调整(调参)
            - 可学习参数
                - 基本特征: 随机初始化，在训练过程中学习调整
                - 我们所说的学习训练模型，就是指学习那些可学习参数，比如全连接神经网络中神经元之间的权值。
                - 可学习参数与超参之间没有严格的划分，通过一些技巧，可以使得一些超参变得可以学习，比如正则化。
                - 需要初始化，比如初始化为零，但这容易导致参数对称，所以更多情况下是在高斯分布中随机初始化
            - 超参是一个麻烦的东西，在实际工作中，调参是一个重要但麻烦的步骤。如何调参呢？
                - 把超参变得可学习
                - 有技巧地调参?????
        - step5: 进行训练
            - 我们希望学习一个函数映射特征与结果之间的关系，那么就等价希望函数的结果接近实际结果。在此我们需要定义一个损失函数来衡量函数结果和实际结果的差异度，每个样本都会产生一个差异度，加起来就是总的损失程度，我们希望总的损失程度小，即最小化(极小化)，这就等价与最优化问题了，常用的方法是梯度下降。
            - 训练方法有多种，有训练上的超参，比如dropout的比例，具体实现上有很多trick。根据no free lunch theorem, 同样不会存在万能的训练方法，也要根据具体问题，具体模型，选择不同的训练方法。
        - step6: 训练完之后得到一个模型结果，应用到校验集上看效果，然后调整超参，重新训练
        - step7:如上步骤重复多次，选择在校验集上效果最好的模型结果作为最后结果，应用到测试集上看效果
    - 经典问题：校验集和测试集的区别
        - 因为要求模型具有泛化能力，所以不能看模型在训练集上的效果。因此我们需要在其他数据集上进行应用，看效果。校验集和测试集都能满足这个要求。这是相同点。
        - 校验集是用了调参的。每次训练得到一个模型之后，我们应用到校验集上看模型的效果，根据效果，我们调整超参，重新训练。如是多次，选择在校验集上效果最好的模型。注意，在这里我们是期望模型在校验集上效果最好，但这能说明模型效果就好吗?这有可能导致模型倾向于校验集数据，使得模型只在校验集上表现好，但在实际数据中效果不好。因此需要测试集
        - 测试集是用来最后评价模型的好坏。利用校验集我们多次调参，多次训练，最后得到一个模型。这么的模型真正的好坏，不能用训练集或校验集来说明，因此需要测试集。
    - *数据和特征决定了机器学习的上限，而模型和算法只是逼近这个上限而已*
2. **无监督学习**
    - 相比于有监督学习目标是学习一个函数来表示特征与结果之间的关系，无监督学习更倾向于学习数据之间的关系
    - 无标注数据
        - 相对于有标注数据，即是数据。
        - 可以是原始数据，也可以是数学建模后的数据，比如向量化之后的数据
        - 在实际中大多数据都是无标注数据，有标注数据是很少有的。
    - 应用
        - 聚类
            - 比较简单，常用k-mean算法
        - 向量化
            - 通过向量之间的关系来表达数据之间的关系，相比于one-hot表示法，向量表示更make sense
        - 维度压缩，提取隐含信息
            - 基于svd
                - pca，lsa等

        - 局部嵌入(local embeddings)
            - 学习一个对高维数据的低维表示。原始高维数据满足一个分布，我们学习的低维表示也满足一个分布。我们学习的方向是使得低维数据的分布接近高维数据的分布，当两分布接近时，低维数据就是高维数据的一个很好低维表示
    - 发现图的结构(discovering graph structure)
    - 信息补全(Matrix completion)
3. 其他机器学习内容补充
    - 非参数模型(non-parametric model) k-近邻
        - 有监督学习
        - 没有训练过程，预测一个新样本的结果(类别或者实数值)时，在训练集上选择k个和新样本最接近的样本作为近邻。如果是判断新样本的类别，则选择近邻中最多数的类别作为新样本的类别预测值。如果是判断新样本的实数值，则取近邻的结果平均数作为新样本的结果预测值
        - 维度诅咒
            - 当样本的特征维度多时，k-近邻的会变得很差
            - 原因：
                - 我们的数据集是有限的，不会随着特征数增多而指数地增多
                - 我们的数据集在特征空间里均匀分布
                - 而特征空间随着特征维度的增加指数增大，因此当特征维度过多时，数据集在特征空间中变得很稀疏，此时选择k个最接近的样本也是不接近的，效果肯定不好
        - *(思考)先通过维度压缩，再用k-近邻是否能解决问题呢?*