数据科学概论

1. 有监督学习和无监督学习的区别。

有监督学习和无监督学习是机器学习中的两种基本学习方式。有监督学习需要有带标签的训练数据，通过训练模型来预测新的未知数据。无监督学习则不需要标签，通过对数据的自我组织、聚类等方式来发现数据的内在结构和规律。因此，有监督学习更适用于分类、回归和预测等任务；而无监督学习则更适用于探索数据的特征、分类和聚类等任务。

有监督：线性回归、分类方法、集成方法、LDA

无监督：聚类、降维方法（除LDA）

2. 线性回归方法模型、基本假设前提，优缺点，是否会出现过拟合，对于数据 的敏感程度等，线性回归模型存在的问题以及如何解决相关问题。复习参考课
堂 PPT 内容。

模型建立

$$L(w) = (\boldsymbol{y} - \boldsymbol{X}\mathbf{w})^T(\boldsymbol{y} - \boldsymbol{X}\mathbf{w})$$

求解（梯度下降）

$$x^{(t+1)} = x^{(t)} - \lambda_t \nabla f(x^{(t)}), \quad \lambda_t \text{为步长}$$

一元情况下：

$$w_1 \leftarrow w_1 - \frac{\eta}{|B|}\sum_{i\in B}\frac{\partial L(w_1,w_0)}{\partial w_1} = w_1 - \frac{\eta}{|B|}\sum_{i\in B} x_i(x_i w_1 + w_0 - y_i)$$
$$w_0 \leftarrow w_0 - \frac{\eta}{|B|}\sum_{i\in B}\frac{\partial L(w_1,w_0)}{\partial w_0} = w_0 - \frac{\eta}{|B|}\sum_{i\in B} (x_i w_1 + w_0 - y_i)$$

损失函数

$$\sum_{i=1}^{n}(\boldsymbol{w}\boldsymbol{x_i} + b - y_i)^2$$

模型评估方法（见后）

基本假设前提（见后）

适用范围：数据线性分布
常见存在的问题：过拟合、变量多重共线性、数据非线性
Solution：正则化（岭回归、LASSO回归）、非线性回归

3. 本课程涉及到的分类方法有哪些？ 这些方法的复习参考课堂 PPT 内容和课后 相关作业。主要了解这些分类方法的核心思想，基本假设前提，优缺点，是否会出现过拟合，对于数据的敏感程度等。
逻辑回归，朴素贝叶斯，KNN，决策树，SVM

	逻辑回归（LR）	K近邻(KNN)	朴素贝叶斯	决策树（DR）	支撑向量机（SVM）
基本思想	假设数据服从Logistic分布，在线性回归的基础上，将输出值通过Sigmoid函数映射到[0,1]区间,使用极大似然估计做参数估计	当对测试样本进行分类时，通过扫描训练样本集，找到与该样本集最相似的k 个样本，根据k 个样本的类别确定测试样本的类别	基于特征条件独立性假设，通过学习联合概率分布，利用贝叶斯公式，计算后验概率分布	根据样本点训练得到一棵树，内部节点表示一个特征或属性，叶节点表示一个类。关键是如何选择节点属性和属性分割点	通过最大化分类间隔构建一个或多个高维超平面分割样本点，超平面即为分类边界
模型	$z = w^T x + b$ $\emptyset(z) = \frac{1}{1+e^{-z}}$ $y = \begin{cases} 1, if\ \emptyset(z) \geq 0.5 \\ -1, otherswise \end{cases}$	无	$argmax_k\ P(Y=k\|X)$ $= P(X\|Y=k)P(Y=k)$	无	$\text{sign}(w^T x + b)$
损失函数	$J(w) = -\sum_{i=1}^{n} \ln(p(y_i\|x_i))$ $p(y_i\|x_i;w) = \frac{1}{1+e^{-y_i w^T x_i}}$ $= \emptyset(-y_i w^T x_i)$	无	$argmax_k\ P(Y=k\|X)$ $= P(X\|Y=k)P(Y=k)$	无	$\text{argmax}_{w,b}$ $\{\frac{1}{\|w\|} min_i[y_i(w^T x_i + b)]\}$
参数求解	极大似然估计+梯度下降法	无	主要是先验参数和条件参数的估计，通常使用极大似然估计	无	拉格朗日乘子法 + SMO算法
假设前提	无	无	基于特征条件独立性假设	无	线性可分SVM基于训练集线性可分假设
优缺点	应用最为广泛的分类算法之一，可解释性强；对自变量的多重共线性比较敏感	对异常数据不敏感，简单易实现，易并行，训练集大时效果好；需要取k 值，占用大量存储空间，计算效率不高	较为稳定，无需训练；如果特征之间不满足独立性假设，可能会降低贝叶斯分类器的后验概率	原理简单，易于理解，具有较高的精确度，可解释性高，对于缺失特征也有很好的处理方式；容易陷入局部最优解，无法处理复杂边界	应用范围广，且建模效果好；选择不同的核函数可以处理不同类型的数据；较间隔SVM可以处理带噪声的数据
过拟合	1.增加数据量（万能办法） 2. 减少特征：手动剔除；特征选择算法 3.正则化：L1、L2正则化	训练集较小时，K近邻算法易导致过拟合	独立性假设可有效对抗过拟合	易发生过拟合，通过剪枝避免过拟合	损失函数中本身有正则化项，一定程度上能够对抗过拟合

4. 聚类方法有哪些？ 不同聚类方法适用于哪类数据的聚类？ 重点关注 K-means 和 DBSCAN(基于密度的聚类方法)

K-means（簇状、凸形、球形），层次聚类（凸形），谱聚类（半月形，圆环形），DBSCAN（任意形状）。

K-means和 DBSCAN ：

- DBSCAN：
  - 聚类结果不是完全分割
  - 能处理不同形状和大小的簇
  - 能处理噪声和异常点
  - 要求密度的定义有意义
  - 处理高维数据时效率较低
  - 对数据分布没有任何假定

- K-means：
  - 聚类结果完全分割
  - 偏向于球形聚类
  - 出现异常点时聚类效果差
  - 要求聚类中心的定义有意义
  - 处理高维数据时效率高
  - 隐含数据服从高斯球形分布

5. 降维方法有哪些？ 不同降维方法的区别是什么， 降维后保持的信息有何不同？

线性降维方法：PCA（保持方差信息），LDA（保持分类信息），ICA

非线性降维方法：

基于核函数：KPCA, KICA, KDA

基于流形学习：ISOMAP（保持数据点之间的测地距离），LLE，MDS，Diffusion Maps

基本思想

PCA模型：

优化后数据$\mathbf{Y}=\boldsymbol{XW}$，其优化模型为：（$\Sigma=\frac{1}{n-1}\boldsymbol{X}^T\boldsymbol{X}$，$\boldsymbol{X}=(x_1,x_2,\ldots,x_n)^T$，$\boldsymbol{X}_{n\times d}$矩阵）

$$\max_W tr(W^T\Sigma W),\ s.t.\ \ w_i^T w_i=1, i=1,2,\ldots l$$

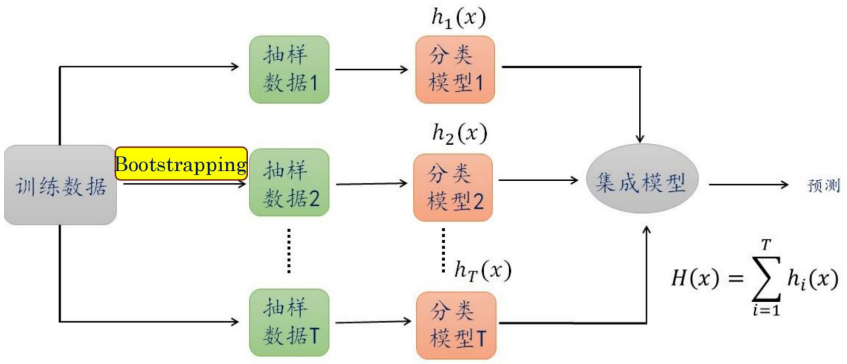
PCA和LDA区别：

（1）思想不同：PCA保持方差信息，LDA保持类别信息

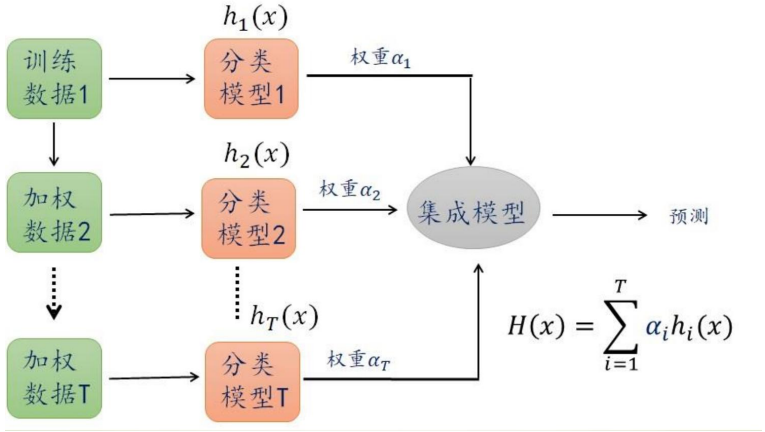
（2）学习模式不同：PCA无监督，LDA有监督

6.集成方法有哪些？ 三种集成方法的算法示意图是什么，注意其中的区别。

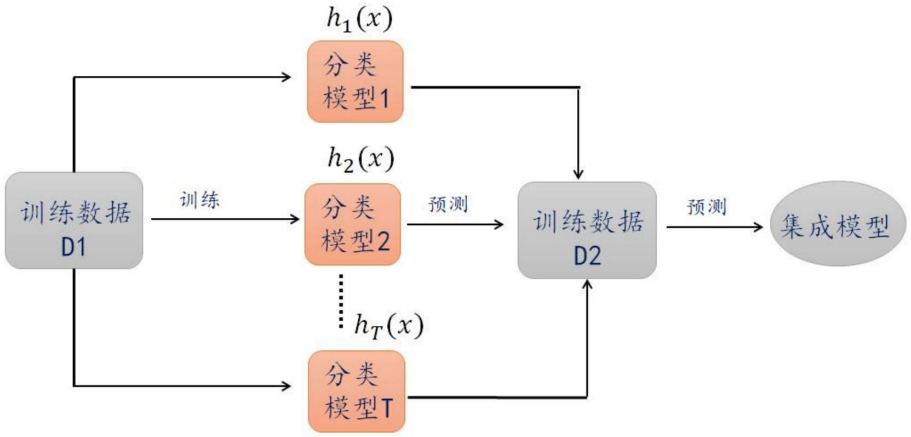
Bagging：



Boosting：



Stacking：



7.随机森林的核心思想是什么？如何保证基模型的独立性（也即随机森林如何保

证随机性）

随机森林的核心思想是通过集成多个决策树来提高分类准确率和避免过拟合。每个决策树都是基于随机的样本（bootstrapping）和随机的特征构建的，从而保证基模型的独立性和多样性，避免过拟合。具体实现时，每个决策树的建立过程中，可以随机采样一部分样本和一部分特征，用于构建该模型，从而保证了基模型的独立性和随机性。最终的分类结果由多个决策树的投票决定。

8. 模型评估需要划分数据集，常用的划分数据集的方法是什么？

Hold out、Cross validation（简单CV、K-fold、留一法）、Bootstrapping

9. 线性回归、PCA、朴素贝叶斯、随机森林等模型是基于什么假设前提的？ 总结课程中提到的模型的假设前提。

线性回归：数据满足同方差性，随机误差项独立同分布于N(0, $sigma^2$)

PCA：输入数据高斯分布、均值化处理

朴素贝叶斯：特征条件独立性假设

随机森林：每棵树各不相同且预测结果相互独立。

10. 图像分类的深度学习方法有哪些？（主要是卷积神经网络方法）

LeNet、AlexNet、VGG、GoogLeNet、ResNet、DenseNet

11. 一些名词解释： KNN 、SVM 、PCA 、MDS 、K-means 、Adaboost 、BP（反向传播）、DBSCAN 、CNN 、LDA 、LLE 等。

KNN（K-近邻法）：一种有监督学习方法，KNN，即K近邻算法，K近邻就是K个最近的邻居。当需要预测一个未知样本的时候，就由与该样本最接近的K个邻居来决定，常用于分类和回归问题。

SVM（支持向量机）：一种有监督分类算法，构建了一个或多个高维的超平面来分割样本点，超平面即为分类边界。

PCA（主成分分析）：一种常用于数据降维的方法，无监督，通过线性变换将高维数据转换为低维数据。构造原变量的一系列线性组合，以去除数据的相关性，并使得降维后数据最大程度保持原始高维数据的方差信息。

MDS（多维尺度变换）：多维尺度变换，非线性降维方法，其基本思想是降维后尽可能保持数据的相似度信息。

K-means：一种无监督聚类算法，将一组数据分成K个簇，每个簇都包含最接近其质心的数据点。该算法通过计算数据点与各个簇中心点之间的距离来将数据点分配到离其最近的簇中心点所在的簇中。算法迭代进行，依次更新每个簇的中心点位置，并将分类结果重新计算。这个过程一直进行直到分类结果稳定或者达到预定的迭代次数，最终将所有数据点划分为K个不同的簇。

Adaboost：一种集成学习方法，通过集合多个分类器来提高分类性能。利用同一批训练样本的不同加权版本，训练一组弱分类器。然后把这些弱分类器以加权的形式集成起来，形成一个最终的强分类器。

BP（反向传播）：一种常用于神经网络中的优化算法，该方法会计算神经网络中损失函数对各参数的梯度，配合优化方法更新参数，降低损失函数。

DBSCAN（基于密度聚类）：是一种基于密度的聚类算法。它可以对具有复杂形状的数据集进行聚类，可以有效地处理噪声数据和非球形簇。DBSCAN算法通过确定核心点和直接密度可达的点来确定簇。

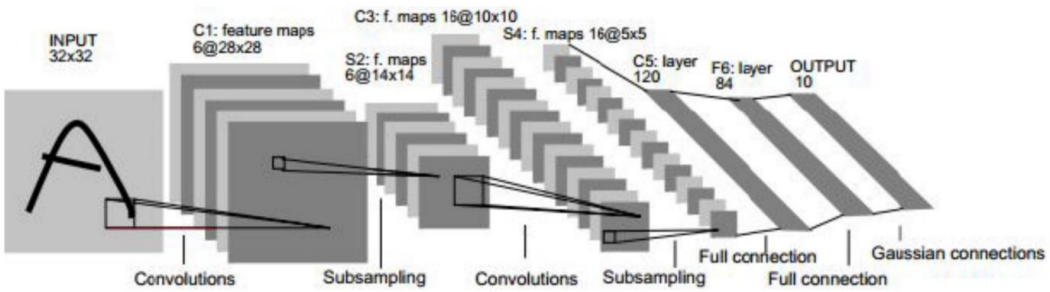
CNN（卷积神经网络）：是一种针对图像或序列信号等高维数据的深度神经网络。它具有卷积层、池化层和全连接层等组成。

LDA（线性判别分析）：有监督线性降维，，通过将高维数据投影到一个低维空间中，使得数据的类别区分度最大，来实现分类。

LLE（局部线性嵌入）：局部线性嵌入，非线性降维方法，其基本思想是降维后尽可能保持数据的局部几何信息（局部线性关系）。

11. 卷积神经网络 CNN 的特点和基本网络架构。

CNN示意图（LeNet：



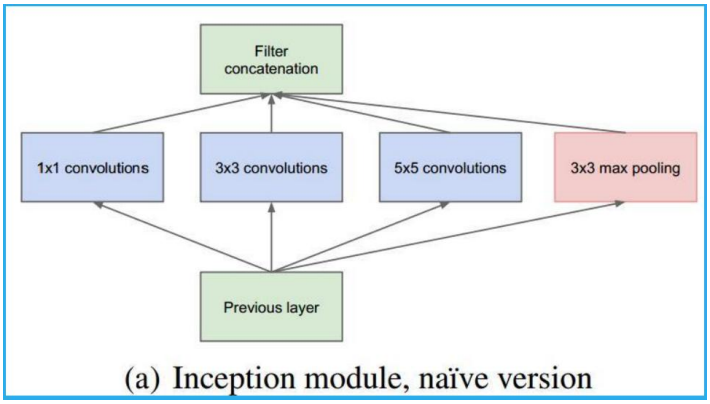
特点：局部连接， 权值共享， 池化操作， 多层次结构。

CNN的基本结构由输入层、卷积层（convolutional layer）、池化层（pooling layer）、全连接层及输出层构成。

常见的卷积神经网络及特点

LeNet：保留图像的输入形状，采用滑动窗口的计算方式，是最早的CNN；
AlexNet：相较LeNet有着更深的网络层数，成功应用了ReLU，Dropout，数据增强；
VGG：堆叠小卷积核，参数更少的情况下能达到与大卷积核相同的感受野，因此在计算代价更小的情况下使得网络更深；
NIN：采用1*1 卷积核，控制通道数； 卷积网络的最终输出层不采用全连接层；
GoogleLeNet：采用Inception block；通过不同尺度的卷积核提取不同尺度的特征；最后拼接使得不同尺度特征融合，让网络自行学习需要何种尺度特征；1*1卷积核减少通道数；
ResNet：采用跳连结构，逼近恒等映射，神经网络本质上在学习残差；
DenseNet：跳连结构中之前的信息都会累加 。

Inception Block



(a) Inception module, naïve version

CNN Code（PyTorch Version）

假设输入数据维数为3，作简单线性回归

```
model = nn.Linear(3,1);
criterion = nn.MSELoss() # Loss and optimizer
optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate)
num_epochs = 50
# Train the model
for epoch in range(num_epochs):
    # Convert numpy arrays to torch tensors
    inputs = torch.from_numpy(x_train)
    targets = torch.from_numpy(y_train)
    # Forward pass
    outputs = model(inputs)
    loss = criterion(outputs, targets) #MSE loss
    # Backward and optimize
    optimizer.zero_grad()  #清零
    loss.backward()  #反向传播 loss 关于model里面的参数的梯度
    optimizer.step()  #优化参数， model 里面的参数被更新
```

Two-layer FNN

```
class NeuralNet2(nn.Module):
    def __init__(self, input_size, hidden_size1,hidden_size2,num_classes):
        super(NeuralNet2, self).__init__()
        self.fc1 = nn.Linear(input_size, hidden_size1)
        self.relu = nn.ReLU()
        self.fc2 = nn.Linear(hidden_size1,hidden_size2)
        self.fc3 = nn.Linear(hidden_size2, num_classes)

    def forward(self, x):
        out = self.fc1(x)  #f1 = w1*x+ b1
        out = self.relu(out)
        out = self.fc2(out)
        out = self.relu(out)

        out = self.fc3(out)
        return out

model = NeuralNet2(input_size, hidden_size,hidden_size, num_classes)
```

```
for epoch in range(num_epochs):
    for i, (images, labels) in enumerate(train_loader):
        images = images.reshape(-1, 28*28).to(device)
        labels = labels.to(device)

        # Forward pass
        outputs = model(images) # 通过模型model预测得到输出
        loss = criterion(outputs, labels) # 计算损失函数


        # Backward and optimize
        optimizer.zero_grad() # 梯度清零
        loss.backward() # 反向传播，计算损失函数的梯度
        optimizer.step() # 对参数进行优化
```

13. 深度学习中常用的几种优化方法。

SGD，mini-batch，AdaGrad，动量梯度下降算法，RMSprop算法，Adma算法

14.深度学习中常用的几种激活函数是什么？以及这些激活函数各自的优缺点。

(1)sigmoid

优点：可以将任意实值输入压缩到（0，1）之间

缺点：容易产生“饱和现象”，造成梯度消失；

不以0为中心，使得参数更新效率变低；

函数带有指数函数，计算复杂。

(2) tanh

优点：可以将任意实值输入压缩到（-1，1）之间；

零均值，参数更新效率比sigmoid高。

缺点：会产生梯度消失；

(3) ReLU

优点：没有饱和现象，没有梯度消失；

没有指数运算，计算效率高；

收敛速率比sigmoid, tanh函数快。

缺点：

不以0为中心，使得参数更新效率变低；

神经元死亡问题。

(4) Leaky ReLU

优点：同ReLU；

解决死神经元问题。

(5) ELU

优点：继承了ReLU的优点；

有较高鲁棒性，因为负轴区域有饱和区域；

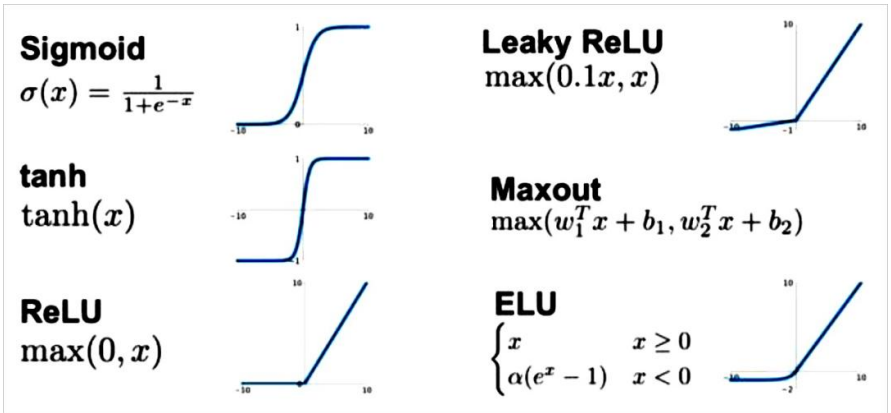
输出均值接近0，所以BP过程中效率提高。

(6) MaxOut

优点：没有饱和区域，不会出现失效神经元；

拟合能力强，可以拟合任意凸函数。

缺点：需要多组参数w_k，b_k。



15. 什么是过拟合现象？本课程中你学到了哪些可以避免过拟合现象的方法？

正则化、 Dropout、Batch Normalization、剪枝（DecisionTree）、数据增强、早停止、集成方法。