简答/名词解释

聚类

将物理或抽象对象的集合分成由类似的对象组成的多个类的过程被称为聚类。

最大似然估计：

待估计参数θ是客观存在的（本身常量），将样本的联合概率函数看成θ的函数，求 其最大值

已知某个参数的值在这个样本中出现的概率最大，把这个参数作为估计的真实值。

最大似然估计是已经知道了结果，然后寻找使该结果出现可能性最大的条件，以此作为估计值。

1.写出似然函数，2.对似然函数取对数，并整理3.求导，令导数为零，得到似然方程4，解似然方程，得到的参数即为所求。

测试集

在机器学习中，一般将样本分成独立的三部分训练集(train set)，验证集(validation set)和测试集(test set)。其中，测试集用来检验最终选择最优的模型的性能如何。

机器学习（过程）：

机器学习是用数据或以往的经验，以此优化计算机程序的性能标准；对于某类任务T和性能度量P，如果一个计算机程序在T上其性能P随着经验E而自我完善，那么我们称这个计算机程序从经验E中学习。

过程或者思路：获取数据，数据预处理，特征提取，特征选择，推理预测识别。

其中数据预处理，特征提取，特征选择部分称为特征表达，是关键性步骤

剪枝

剪枝指的是在深度优先搜索中去掉一些不符合题目要求的或是浪费时间而没有作用的答案，从而使得深度优先搜索能够更快得到正确答案。因为在搜索树中去掉答案形似剪掉树的枝叶，所以这一方法被称为剪枝。

KNN

所谓K最近邻，指每个样本都可以用它最接近的k个邻居来代表。kNN算法的核心思想是如果一 个样本在特征空间中的k个最相邻的样本中的大多数属于某一个类别，则该样本也属于这个类 别，并具有这个类别上样本的特性 由于kNN方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的，因 此对于类域的交叉或重叠较多的待分样本集来说，kNN方法较其他方法更为适合

1. 计算测试数据与各个训练数据之间的距离 2. 按照升序（从小到大）对距离（欧氏距离）进行排序 3. 选取距离最小的前k个点 4. 确定前k个点所在类别出现的频率 5. 返回前k个点中出现频率最高的类别作为测试数据的分类

6. **KNN算法k值的选取**

    1. 当k的取值过小时，一旦有噪声得成分存在们将会对预测产生比较大影响，整体模型变得复杂，容易发生过拟合
    2. 如果k的值取的过大时，学习的近似误差会增大，整体的模型变得简单。与输入目标点较远实例也会对预测起作用，使预测发生错误
    3. K的取值尽量要取奇数，以保证在计算结果最后会产生一个较多的类别，如果取偶数可能会产生相等的情况，不利于预测
    4. 常用的方法是从k=1开始，估计分类器的误差率。重复该过程，每次K增值1，允许增加一个近邻，直到产生最小误差率的k
    一般k的取值不超过20，上限是n的开方，随着数据集的增大，K的值也 要增大

kmeans

k均值聚类算法（k-means clustering algorithm）是一种迭代求解的聚类分析算法，其步骤是，预将数据分为K组，则随机选取K个对象作为初始的聚类中心，然后计算每个对象与各个种子聚类中心之间的距离，把每个对象分配给距离它最近的聚类中心。聚类中心以及分配给它们的对象就代表一个聚类。每分配一个样本，聚类的聚类中心会根据聚类中现有的对象被重新计算。这个过程将不断重复直到满足某个终止条件。终止条件可以是没有（或最小数目）对象被重新分配给不同的聚类，没有（或最小数目）聚类中心再发生变化，误差平方和局部最小。

Parzen窗估计

（小舱体积在全局处处保持不变） 核密度（Parzen 窗）估计通过离散样本点来的线性加和来构建一个连续的概率密度函数，从而 得到一个平滑的样本分布，可以看作是对直方图的一个自然拓展

h过大，由于假定小舱内p(x)为常数，则导致过于平均的估计结果 h过小，落入小舱的样本将会很少，或者没有样本落入，从而导致对p(x)的估计不连续（刺 头）

强化学习:

强调如何基于环境而行动，以取得最大化的预期利益；其关注点在于寻找探索 （对未知领域的）和利用（对已有知识的）的平衡

线性可分:

线性可分指的是可以用一个线性函数将两类样本分开（无误差），比如在二维空间中的直线、 三位空间中的平面以及高维空间中的线性函数

总结其最简化形式定义：一个训练样本集$\{(X_i, y_i)\}$在i=1~N线性可分，是指存在(w, b)，使得对于i=1~N有：

$$y_i(w^T X_i + b) > 0$$

多层感知器

（MLP）：是一种前向结构的人工神经网络，映射一组输入向量到一组输出向 量。MLP可以被看作是一个有向图，由多个的节点层所组成，每一层都全连接到下一层。 除了输入节点，每个节点都是一个带有非线性激活函数的神经元（或称处理单元）

奥卡姆剃刀原理

意为“简约法则”，如果关于同一个问题有许多种理论，每一种都能作出同样准确的预言，那么应 该挑选其中使用假定最少的

主动学习:

指通过自动的机器学习算法，从数据集中自动挑选出部分数据请求标签；有效 的主动学习数据选择策略可以有效地降低训练的代价，并同时提高模型的识别能力

ID3

使用**信息增益最小化**来选择分类任务节点（离散特征）

优点：1.假设空间包含所有的决策树，搜索空间完整。2.健壮性好，不受噪声影响。3.可以训练缺少属性值的实例。

总的来说，就是理论清晰、方法简单、学习能力较强

缺点：ID3算法的缺点也是很明显，正如我们上面所说，ID3算法会去选择子类别多的特征，因为这样分裂出来的结果会更纯，熵会更小，这有偏于我们的初衷，我们要的纯不是想通过让它分类分的更细得来的纯啊！如果这样那不如分100个类好了里面数据的纯度都很高。

神经网络

神经网络由许多相互关联的概念化的人造神经元组成， 它们之间传递相互数据，并且具有根据网络"经验"调整的相关权重。神经元具有激活阈值，如果通过其相关权重的组合和传递给他们的数据满足这个阈值的话，其将被解雇；发 射神经元的组合导致"学习"。 神经网络是一种人工智能方法，用于教计算机以受人脑启发的方式处理数据。是深度学习算法的核心，使用类似于人脑的分层结构中的互连节点或神经元。它可以创建自适应系 统，计算机使用该系统来从错误中进行学习并不断改进。因此，人工神经网络可以尝试解 决复杂的问题，例如更准确地总结文档或人脸识别。 由节点层组成，包含一个输入层、一个或多个隐藏层和一个输出层。 每个节点也称为一个 人工神经元，它们连接到另一个节点，具有相关的权重和阈值

监督学习和非监督学习：

监督学习，是其训练集的数据是提前分好类，带有标签的数据，进行学习到模型以及参数，当用测试集进行测试时，给出D测={Xi}=>{yi}。最常见的一种机器学习，可以由训练资料中学到或建立一个模式，它的训练数 据是有标签的，训练目标是能够给新数据（测试数据）以正确的标签。

非监督学习：需要将一系列没有标签的训练数据，输入到算法中，需要根据样本之间的相似性对样本集进行分类或者分析。用于在大量无标签数据中挖掘信息，它的训练数据是无标签的，训练目标是 能对观察值进行分类或者区分等。

半监督学习：用于在大量无标签数据中发现些什么。它的训练数据是无标签的，训练目标 是能对观察值进行分类或者区分等

独立同分布:

全体样本服从一个未知分布，每个样本都是独立地从这个分布上采样获得

独立：每次抽样之间没有关系，不会相互影响

同分布：训练集和全体分布一致、相同

信息增益:

在决策树算法中，信息增益是特征选择的一个重要指标，信息增益代表了在一个条件下，信息复杂度（不确定性）减少的程度

激活函数

一种添加到人工神经网络中的函数，旨在帮助网络 学习数据中的复杂模式，将线性函数转变为非线性的

深度学习：

深度学习是用于建立、模拟人脑进行分析学习的神经网络，并模仿人脑的机制来解释数据的一种机器学习技术

本质上多隐层人工神经网络

属于深层模型，对于具体地任务，利用给定的一批数据，先训练一个多隐层人工神经网络，然后使用它，这就是深度学习

自动学习特征、超强的非线性建模能力

难以人工定义特征、大量标记样本、高性能计算资源

鲁棒性不足：假阳性；不可解释性；超参选择

集成学习（ BOOSTING和BAGGING 思想)

集成学习（Ensemble Learning）是解决有监督机器学习任务的一类方法，通过有策略地生成 和组合多个弱学习器，合成强学习器来完成学习任务，提升预测结果。分为同质集成和异质集 成、并行集成和串行集成。结合策略主要有平均法、投票法和学习法等

4. **Bagging和Boosting的区别**

	Bagging	Boosting
样本选择	有放回，各轮训练集之间是独立	训练集不变，权重改变
样例权重	均匀取样，权重相等	错误率越大则权重越大
预测函数	所有预测函数的权重相等	分类误差小的分类器会有更大的权重
计算方式	并行	串行
侧重方向	降低方差，有利于不稳定分类器	减小偏差，强化弱分类器

5. 从期望损失的角度分析 adaboost 的合理性，可从分布和分类器权重更新方面阐述

交叉验证法

顾名思义，就是重复的使用数据，把得到的样本数据进行切分，组合为不同的训练集和测试集，用训练集来训练模型，用测试集来评估模型预测的好坏。在此基础上可以得到多组不同的训练集和测试集，某次训练集中的某样本在下次可能成为测试集中的样本，即所谓"交叉"。

学习率

学习率设置太大，参数更新的幅度就非常大，在最优值附近徘徊，或者loss开始增加；学习率设 置太小，网络收敛非常缓慢，会增大找到最优值的时间，且可能会进入局部极值点就收敛 在训练过程中，一般根据训练轮数设置动态变化的学习率。刚开始训练时，学习率以 0.01 ~ 0.001 为宜；一定轮数过后，逐渐减缓；接近训练结束，学习速率的衰减应该在100倍以上

	学习率 大	学习率 小
学习速度	快	慢
使用时间点	刚开始训练时	一定轮数过后
副作用	1.易损失值爆炸；2.易振荡。	1.易过拟合；2.收敛速度慢。

朴素贝叶斯:

若条件独立性假设成立，则朴素贝叶斯分类器是最佳分类器

朴素贝叶斯模型发源于古典数学理论，有稳定的分类效率

对缺失数据不敏感，算法简单，常用于文本分类，分类准确度高，速度快

但需要先知道先验概率，因此在某些时候由于假设的先验模型的原因导致预测的效果不佳

贝叶斯估计:

- 确定参数 θ 的先验概率密度函数 $p(\theta)$
- 由样本集 $X=\{x_1,x_2\cdots x_N\}$ 求出样本联合概率密度函数 $p(X|\theta)$, 它是 θ 的函数

$$p(X|\theta)=\prod_{n=1}^{N}p(x_n|\theta)$$

- 利用贝叶斯定理, 求 θ 的后验概率密度函数

$$p(\theta|X)=\frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)d\theta}$$

- 求出贝叶斯估计值

$$\bar{\theta}=\int \theta p(\theta|X)d\theta$$

parzen窗简述。为什么可以选用高斯密度函数作为窗函数?

我们可以认为每个样本对于空间上的每个点的概率密度都有一定的贡献，但是随着空间点距离样本点的距离增大，这个贡献率会减小。一个样本点对其所在位置的概率密度贡献最大，随着距离样本点的距离会依次减小，这个贡献率分布函数就是由窗函数定量给出的。将每个样本点对所有空间点的概率密度贡献累加，就是空间上的概率密度分布函数，这便是Parzen窗的算法原理。

在这个区域 R_n 上，频数 k_n 可以由窗口函数 $\psi(x)$ 来定义。常见的方形窗口函数如下：

$$\varphi(\mathbf{u}) = \begin{cases} 1 & |u_j| \leq 1/2 \\ 0 & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, d$$

方形窗口函数，如果采样点在边长为1的超方体内部，取值为1， 否则为0。

常见的高斯窗口函数如下：

$$\varphi(u) = \frac{1}{\sqrt{2\pi}} \, e^{-u^2/2}.$$

高斯窗口函数，随着采样点与中心点距离的增大而减小。

而 k_n 的估计为 $k_n = \sum_{i=1}^{n} \psi((x - x_i)/h_n)$ ，其中 x 是空间中的任意点，而 x_i 是所有观测到的采样点。 我们可以看到，当采用方形窗口函数时，它就是出现在该区域 R_n 内部的

1. 平滑性：高斯密度函数具有平滑性，其在中心点附近变化较为缓慢。这使得高斯窗函数能够在频域中产生较小的副瓣，从而减少频谱泄漏的问题。与其他窗函数相比，高斯窗函数能够提供更好的频域局部化，因此在信号处理和频谱分析中被广泛使用。
2. 快速衰减：高斯窗函数的幅度在离中心越远的地方衰减得越快。这意味着高斯窗函数能够有效地抑制远离中心的频域成分，从而提供更好的频域分辨率。
3. 数学性质：高斯函数在数学上具有一些重要的性质，例如它是连续可导、无界延伸的。这些性质使得高斯窗函数易于处理和分析，并且能够在许多领域中进行数学推导和计算。

什么是过拟合？模型为什么会出现过拟合？如何避免过拟合?

过拟合是指机器学习模型在训练数据上表现良好，但在未被训练的新数据上表现不佳的现象，导致泛化能力下降，即过多学习了错误数据（不具泛化能力的含有自身特点的训练数据特征）或噪音数据。过拟合的原因是模型参数数量太多，远远大于样本数量。

->解决过拟合的常用措施有以下几种：

->正则化：通过在模型的损失函数中引入正则化项，可以惩罚模型中较大的权重，从而降低模型的复杂度。常见的正则化方法包括L1正则化和L2正则化。

->提前停止训练：在训练过程中，当模型在验证集上的性能不再提高时，可以停止训练，以避免过拟合。

简要说明梯度下降法和牛顿法的基本思想和区别。解释为什么梯度下降法能够保证一定是下降的?

梯度下降法原理 （一阶优化算法）通过搜索方向和步长来对参数进行更新。其中搜索方向是目标函数在当前位 置的负梯度方向。因为这个方向是最快的下降方向。步长确定了沿着这个搜索方向下降的大 小。

牛顿法原理 （二阶优化算法）牛顿法是求解函数值为0时的自变量取值的方法，利用牛顿法求解目标函数的 最小值其实是转化成求使目标函数的一阶导为0的参数值（这一转换的理论依据是，函数的极值 点处的一阶导数为0）。其迭代过程是在当前位置x0求该函数的切线，该切线和x轴的交点x1， 作为新的x0,重复这个过程，直到交点和函数的零点重合。此时的参数值就是使得目标函数取得 极值的参数值。

梯度下降法	牛顿法	备注
最优值附近震荡		接近最优值时不断减少步长
	远离局部极小点可能不会收敛	先得到离最优点较近的点
	计算量、内存代价大	使用拟牛顿法
	计算速度快	二阶函数

4. **证明为什么梯度下降算法可以确保是下降的/为什么梯度下降选择负梯度优化目标函数/（为何负梯度是函数值减小的最快方向）**

   假设我们的object function为：$arg\min_w\ f(w)$，为了求得最优解，可以这样变形：
   $min = f(w + step_w) - f(w)$达到最小，利用泰勒展开公式有：
   $f(w + step_w) = f(w) + f'(w) * step_w$
   $f(w + step_w) - f(w) = f'(w) * step_w$
   为了得到min，我们只要使w的更新步长$step_w = -f'(w)$即可（这保证了函数一定不增，且梯度是函数增长最快的方向，取负为下降最快）
   这样就得到了梯度下降法的公式：
   $w = w - \alpha * f'(w)$

机器学习中怎么划分数据集的，评估方法有哪几种，分别解释

数据集的划分一般有三种方法：

1.留出法（Hold-out）：为了保证数据分布的一致性，通常我们采用分层采样的方式来对数据进行采样。

2.交叉验证法（Cross Validation）：K折交叉验证的基本思想是：把原始训练数据集分割成K个不重合的子数据集，然后做K次模型训练和验证。每一次，使用一个子数据集验证模型，并使用其它K−1个子数据集来训练模型。最后，对这K次训练误差和验证误差分别求平均。

3.自助法（BootStrapping）：留出法与交叉验证法都是使用分层采样的方式进行数据采样与划分，而自助法则是使用有放回重复采样的方式进行数据采样。

从期望损失角度解释adaboost，如分布和分类器权重更新的依据。

adaboost算法就是通过一堆弱分类器，经过训练，组成成一个强分类器。这一堆弱分类器中的每个分类器在分类的时候效果不理想，但是组成的强分类器能够几何这些弱分类器的优势和特点，使得最终的分类效果变得有效。

总结一下，得到AdaBoost的算法流程：

- 输入：训练数据集$T = \{(x_1, y_1), (x_2, y_2), (x_N, y_N)\}$，其中，$x_i \in X \subseteq R^n$，$y_i \in Y = -1, 1$，迭代次数$M$
- 1. 初始化训练样本的权值分布：$D_1 = (w_{1,1}, w_{1,2}, \ldots, w_{1,i}), w_{1,i} = \frac{1}{N}, i = 1, 2, \ldots, N$。
- 2. 对于$m = 1, 2, \ldots, M$
- (a) 使用具有权值分布D_m的训练数据集进行学习，得到弱分类器$G_m(x)$
- (b) 计算$G_m(x)$在训练数据集上的分类误差率：

$$e_m = \sum_{i=1}^{N} w_{m,i} I(G_m(x_i) \neq y_i)$$

- (c) 计算$G_m(x)$在强分类器中所占的权重：

$$\alpha_m = \frac{1}{2} log \frac{1 - e_m}{e_m}$$

- (d) 更新训练数据集的权值分布（这里，z_m是归一化因子，为了使样本的概率分布和为1）：

$$w_{m+1,i} = \frac{w_{m,i}}{z_m} exp(-\alpha_m y_i G_m(x_i)) \ , \ i = 1, 2, \ldots, 10$$

$$z_m = \sum_{i=1}^{N} w_{m,i} exp(-\alpha_m y_i G_m(x_i))$$

- 3. 得到最终分类器：

$$F(x) = sign(\sum_{i=1}^{N} \alpha_m G_m(x))$$

adaboost对权重a的推导

基分类器h_i的权重 α_t 应使得 $\alpha_t h_t$ 最小化指数损失函数：(D：权重w，h：分类器f)

$$\begin{aligned}\ell_{\exp}\left(\alpha_t h_t \mid \mathcal{D}_t\right) &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}\left[e^{-f(\boldsymbol{x})\alpha_t h_t(\boldsymbol{x})}\right] \\ &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}\left[e^{-\alpha_t}\mathbb{I}(f(\boldsymbol{x})=h_t(\boldsymbol{x}))+e^{\alpha_t}\left[f(\boldsymbol{x})\neq h_t(\boldsymbol{x})\right)\right] \\ &= e^{-\alpha_t}P_{x\sim D_t}\left(f(\boldsymbol{x})=h_t(\boldsymbol{x})\right)+e^{\alpha_t}P_{x\sim D_t}\left(f(\boldsymbol{x})\neq h_t(\boldsymbol{x})\right) \\ &= e^{-\alpha_t}\left(1-\epsilon_t\right)+e^{\alpha_t}\epsilon_t\end{aligned} \tag{1.8}$$

其中，$\epsilon_t = P_{\boldsymbol{x}\sim\mathcal{D}_t}\left(h_t(\boldsymbol{x})\neq f(\boldsymbol{x})\right)$，也即是第 t 轮迭代时，真实函数与该学习器学习到的数据不同的概率。为了使得损失函数最小化，同样的，我们对其求偏导：

$$\frac{\partial\ell_{\exp}\left(\alpha_t h_t \mid \mathcal{D}_t\right)}{\partial\alpha_t} = -e^{-\alpha_t}\left(1-\epsilon_t\right)+e^{\alpha_t}\epsilon_t = 0 \tag{1.9}$$

由此，我们不难得到：

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{1.10}$$

这既是图 2.1 中的分类器的权重更新公式。

决策树+SVM

24. 决策树的决策过程

用决策树进行分类，从根结点开始，对实例的某一个特征进行测试，根据测试结果分配往对应的子结点中去，每个子结点对应一个特征的取值，递归的进行分类测试和分配，最终到达对应的叶结点，完成本次的分类

25. 决策树的优缺点

优点：决策树可以完美的对训练集进行分类，每个实例都有一个完美的叶节点

缺点：• 但是不能很好地推广到新的例子

• 泛化能力差，假如对现有的决策树进行推广，往往会得到一个更紧凑的决策树，拟合程度更大

26. 如何简单化决策树

（1）预修剪
- 固定深度
- 固定叶子数

（2）后剪枝（独立性的卡方检验）
- 卡方检验
- 消除与标签无关的规则中的变量值
- 消除不必要的规则简化规则集
- 信息标准：MDL

简述线性回归与逻辑回归之间的联系与差别

线性回归

最原始的回归，损失函数用均方差

缺点：系数有可能极大,带来极为不合理的模型

岭回归

牺牲回归精确度来避免系数过大

缺点：几乎不会有0系数，这样会导致一些与y无关的变量，系数仍不为0

Lasso

牺牲回归精确度，并牺牲求解精度，来避免系数过大，且允许0系数的存在

缺点：求解更为复杂，未必取得全局最优解

综合题

贝叶斯决策过程

1. 设定先验概率 2. 通过给定的信息来设定条件概率 3. 将先验概率转化为后验概率 4. 根据后验概率大小进行决策分类

设在某个局部地区细胞识别中正常 ω_1 和异常 ω_2 两类的先验概率分别为：
正常状态： $P(\omega_1)=0.9$
异常状态： $P(\omega_2)=0.1$
现有一待识别的细胞，其观察值为 x，从类条件概率密度分布曲线上查得

$$p(x|\omega_1)=0.2,\qquad p(x|\omega_2)=0.4$$

试使用贝叶斯决策对该细胞 x 进行分类（要求给出具体计算过程及计算结果）
解：

利用贝叶斯公式，分别计算出 ω_1 及 ω_2 的后验概率

$$P(\omega_1|x)=\frac{p(x|\omega_1)p(\omega_1)}{\sum_{j=1}^{2}p(x|\omega_j)p(\omega_j)}=\frac{0.2\times0.9}{0.2\times0.9+0.4\times0.1}=0.818$$

$$P(\omega_2|x)=1-P(\omega_1|x)=0.182$$

根据贝叶斯决策规则，有

$$P(\omega_1|x)=0.818>P(\omega_2|x)=0.182$$

所以合理的决策规则是把 x 归类于正常状态。

2. 增加条件λ11=0, λ12=6, λ21=1, λ22=0, 请判断该细胞是否正常

$$R(\alpha_1|\mathbf{x})=\sum_{i=1}^{2}\lambda_{1i}P(\omega_i|\mathbf{x})=1.092$$

$$R(\alpha_2|\mathbf{x})=\sum_{i=1}^{2}\lambda_{2i}P(\omega_i|\mathbf{x})=0.818$$

$\because R(\alpha_1|x)>R(\alpha_2|x)\therefore x\in$异常细胞(第2类)，因此决策$\omega_1$类风险大。
因λ_{12}=6较大，决策损失起决定作用。

SVM

（1）从VC维和结构风险角度分析为什么margin要最大化。

有着直观地鲁棒性从而具有较强的泛化能力

（2）推导优化函数的对偶形式。

8. **线性可分SVM计算步骤**
    1. **间隔最大化**

    求解最大间隔超平面（两直线距离公式），即求：

    $$\max_{w,b}\frac{2}{||w||}=min\frac{1}{2}||w||$$

    $$s.t.\qquad y_i(w^Tx+b)\geq 1,\forall i$$

    2. **推导目标函数的对偶形式**
        1. 建拉格朗日函数，引进拉格朗日乘子：

        $$L(W,b,\alpha)=\frac{1}{2}||w||^2-\sum_{i=1}^{n}\alpha_i[y_i(w^Tx_i+b)-1]$$

优化问题为：

$$\min_{w,b}\max_{\alpha}L(w,b,\alpha)$$

对偶问题为：

$$\max_{\alpha}\min_{w,b}L(w,b,\alpha)$$

$$s.t.\quad \nabla_{w,b}L(w,b,\alpha)=0$$

$$\alpha\geq 0$$

2. 求$\min_{w,b}L(w,b,\alpha)$，令导数等于0：

$$\nabla_w L(w,b,\alpha)=w-\sum_{i=1}^{n}\alpha_i y_i x_i=0\Rightarrow w^*=\sum_{i=1}^{n}\alpha_i y_i x_i$$

$$\nabla_b L(w,b,\alpha)=\sum_{i=1}^{n}\alpha_i y_i=0$$

3. 代入拉格朗日函数：

$$
\begin{aligned}
L(w,b,\alpha) &= \frac{1}{2}|w|^2 - \sum_{i=1}^{n}\alpha_i[y_i(w^T x_i + b) - 1] \\
&= \frac{1}{2}\sum_{i=1}^{n}\alpha_i y_i x_i^T \sum_{j=1}^{n}\alpha_j y_j x_j - \sum_{i=1}^{n}\alpha_i y_i x_i^T \sum_{j=1}^{n}\alpha_j, y_j x_j - b\sum_{i=1}^{n}\alpha_i y_i + \sum_{i=1}^{n}\alpha_i \\
&= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\alpha_i y_i x_i^T \sum_{j=1}^{n}\alpha_j, y_j x_j \\
&= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_i
\end{aligned}
$$

即：$L(w,b,\alpha) = -\frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_i + \sum_{i=1}^{n}\alpha_i$

4. 求$\min_{w,b} L(w,b,\alpha)$对α的极大，等价于求：

$$
\begin{aligned}
&\min_{\alpha} \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j - \sum_{i=1}^{n}\alpha_i \\
s.t. \quad &\sum_{i=1}^{n}\alpha_i y_i = 0 \\
&\alpha_i \geq 0, \forall i
\end{aligned}
$$

至此，我们得到了原始最优化问题和对偶最优化问题

3. **决策函数的解**

假设得到了对偶问题的最优解α^*，则：$w^* = \sum_{i=1}^{n}\alpha_i^* y_i x_i$

由题设，至少存在一个$\alpha_j^* > 0$，根据KKT条件，至少存在一个j，使得$y_j(w^{*T}x_j + b^*) - 1 = 0$，即可求得最优$b^*$：

$b^* = y_j - w^{*T}x_j$

代回原式可得决策函数为：

$$f(X) = sgn(\sum_{i=1}^{n}\alpha_i^* y_i x^T x_i + b^*)$$

（3）简述SVM线性不可分的情况下如何求解。

1. 加入松弛变量和惩罚因子，找到相对“最好”超平面，尽可能地将数据正确分类 2. 使用核函数，将低维的数据映射到更高维的空间，使得在高维空间中数据是线性可分的， 在高维空间使用线性分类模型

（4）支持向量？margin

支持向量机：算法

1.选择内核函数 2.为C选择一个值 3.解决二次规划问题 4.根据支持向量构造判别函数

（5）SVM基本型

支持向量机在高维或无限维空间中构造超平面或超平面集合，其可以用于分类、回归或其他任 务。SVM是一种二类分类模型，他的基本模型是的定义在特征空间上的间隔最大的线性分类 器，SVM的学习策略就是间隔最大化

线性可分支持向量机：训练数据线性可分，通过硬间隔最大化，学习一个线性的分类器 线性支持向量机：训练数据近似线性可分，通过软间隔最大化，学习一个线性的分类器 非线性支持向量机：训练数据线性不可分，通过核技巧和软间隔最大化，学习一个非线性的分 类器

（6）核函数的条件、作用

SVM 通过选择一个核函数 K ，将低维非线性数据映射到高维空间中。原始空间中的非线性数据经过核函数映射转换后，在高维空间中变成线性可分的数据，从而可以构造出最优分类超平面

（7）软间隔、硬间隔区别（给定软间隔，写出对应硬间隔）

允许训练的模型中，部分样本（离群点或者噪音点）不必满足该约束，同时在最大化间隔时，不满足约束的样本应该尽可能的少。

（8）SVM概念,其目的,什么是最优化分类面

SVM本质模型是特征空间中最大化间隔的线性分类器，是一种二分类模型。

（9）验证核函数，基本如下，只是x,y换成了x1,x2，映射函数是（x^2 x 1/2）T

我们现在考虑核函数 $K(v_1, v_2) =< v_1, v_2 >^2$ ，即“内积平方”。
这里面 $v_1 = (x_1, y_1), v_2 = (x_2, y_2)$ 是二维空间中的两个点。

这个核函数对应着一个二维空间到三维空间的映射，它的表达式是：

$\Phi(v) = v = \begin{bmatrix} x \\ y \end{bmatrix}$ $\Phi(v) = \begin{bmatrix} x2 \\ \sqrt{2}xy \\ y2 \end{bmatrix}$

可以验证，

$$
\begin{aligned}
<\Phi(v_1), \Phi(v_2)> &= < (x_1^2, \sqrt{2}x_1y_1, y_1^2), (x_2^2, \sqrt{2}x_2y_2, y_2^2) > \\
&= x_1^2x_2^2 + 2x_1x_2y_1y_2 + y_1^2y_2^2 \\
&= (x_1x_2 + y_1y_2)^2 \\
&= < v_1, v_2 >^2 \\
&= K(v_1, v_2)
\end{aligned}
$$

五、**神经网络**

1、描述bp算法

反向传播算法主要由两个环节（激励传播、权重更新）反复循环迭代，直到网络的对输入的响 应达到预定的目标范围为止。 BP算法的学习过程由正向传播过程和反向传播过程组成。在正向传播过程中，输入信息通过输 入层经隐含层，逐层处理并传向输出层。如果在输出层得不到期望的输出值，则取输出与期望 的误差的平方和作为目标函数，转入反向传播，逐层求出目标函数对各神经元权值的偏导数， 构成目标函数对权值向量的梯量，作为修改权值的依据，网络的学习在权值修改过程中完成。 误差达到所期望值时，网络学习结束

2、前向传播表达式a1

1. 前向传播

$\alpha_h = \sum_{i=1}^{d} v_{ih}x_i$

$b_h = sigmoid(\alpha_h)$

$\beta_j = \sum_{h=1}^{q} w_{hj}b_h$

$\hat{y}_i = sigmoid(\beta_i)$

$E = \frac{1}{2}\sum_{i=1}^{l}(\hat{y}_j - y_j)^2$

2. 方向传播

$\triangle w_{hj} = -\eta\frac{\partial E}{\partial w_{hj}}$其中$\frac{\partial E}{\partial w_{hj}} = \frac{\partial E}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial \beta_j}\frac{\partial \beta_j}{\partial w_{hj}}$

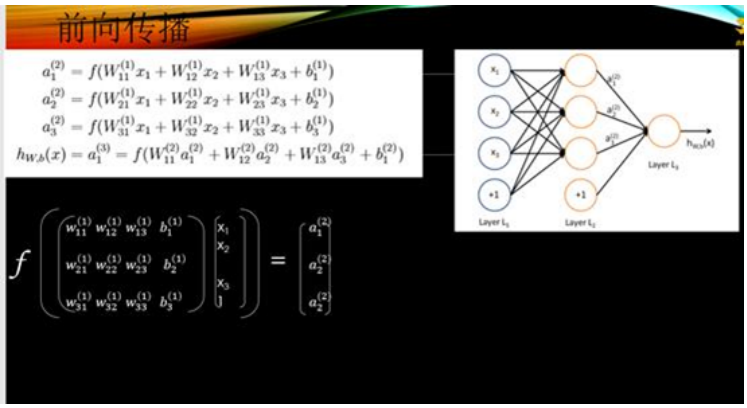
1. $\frac{\partial E}{\partial \hat{y}_j} = -(\hat{y}_j - y_j)$
2. $\frac{\partial \hat{y}_j}{\partial \beta_j} = \hat{y}_j(1 - \hat{y}_j)$
3. $\frac{\partial \beta_j}{\partial w_{hj}} = b_h$

$w_{hj} = w_{hj} + \Delta w_{hj} = w_{hj} - \eta(-(\hat{y}_j - y_j)\hat{y}_j(1 - \hat{y}_j)b_h)$

同理：

$$
\begin{aligned}
\frac{\partial E}{\partial v_{ih}} &= \frac{\partial E}{\partial b_h}\frac{\partial b_h}{\partial \alpha_h}\frac{\partial \alpha_h}{\partial v_{ih}} = \sum_{j=1}^{l}(\frac{\partial E}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial \beta_j}\frac{\partial \beta_j}{\partial b_h})\frac{\partial b_h}{\partial \alpha_h}\frac{\partial \alpha_h}{\partial v_{ih}} \\
&= \sum_{j=1}^{l}[-(-(\hat{y}_j - y_j)\hat{y}_j(1 - \hat{y}_j)w_{hj}][b_h(1 - b_h)][x_i])
\end{aligned}
$$

3、输出层输出形式



六、**深度学习**

1、卷积层作用

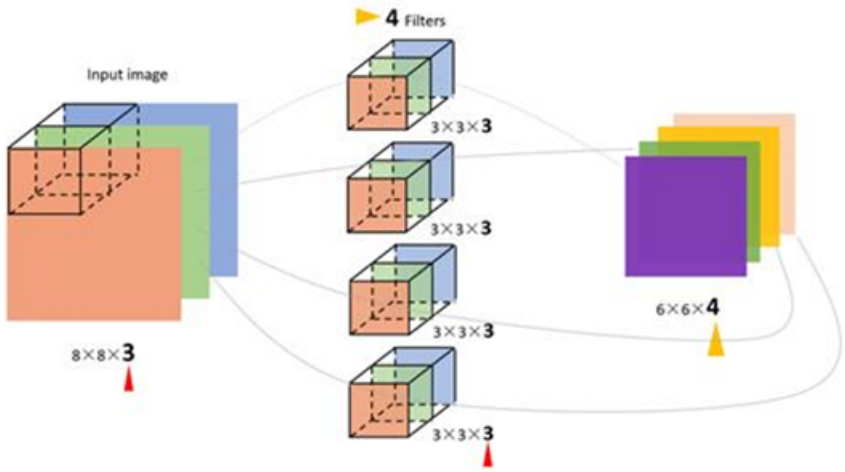
2、激活函数以及形式

a. 不使用激活函数，每一层输出都是上层输入的线性函数，无论神经网络有多少层，输出都是输入的线性组合。

b. 使用激活函数，能够给神经元引入非线性因素，使得神经网络可以任意逼近任何非线性函数，这样神经网络就可以利用到更多的非线性模型中。

3、输出

给出下图左1，左二，问你卷积后输出几个，大小为多少，（输出即右一图，这页ppt我根本就没瞅啊，现在就是后悔！！！）



七、**决策树**

1、决策树算法思想是什么，两个分类

决策树是通过 一系列规则 对数据进行分类的过程。它提供一种在什么条件下会得到什么值的类 似规则的方法。决策树分为分类树和回归树两种，分类树对离散变量做决策树，回归树对连续 变量做决策树。一棵决策树的生成过程主要分为以下3个部分：特征选择、决策树生成、剪枝

2、给出一个表格

1. 写出预处理数据集
2. 决策树缺失项 以及依据
3. 决策树分类规则
4. 给数据判断结果

NO.	属性				类别
	天气	气温	湿度	风	
1	晴	热	高	无风	N
2	晴	热	高	有风	N
3	多云	热	高	无风	P
4	雨	适中	高	无风	P
5	雨	冷	正常	无风	P
6	雨	冷	正常	有风	N
7	多云	冷	正常	有风	P
8	晴	适中	高	无风	N
9	晴	冷	正常	无风	P
10	雨	适中	正常	无风	P
11	晴	适中	正常	有风	P
12	多云	适中	高	有风	P
13	多云	热	正常	无风	P
14	雨	适中	高	有风	N

实体（样本）、概念的正例和反例、训练集

- 决策树是用实体的属性作为结点，用属性的取值作为分支的树结构。决策树叶子为类别名，即P 或者N
- 若要对一实体分类，从树根开始进行测试，按属性的取值分枝向下进入下层结点，对该结点进行测试，过程一直进行到叶结点，实体被判为属于该叶结点所标记的类别