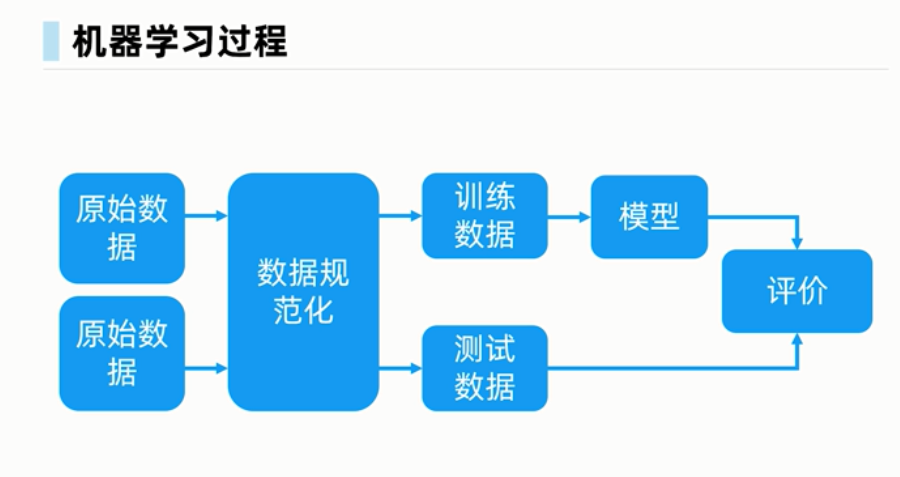
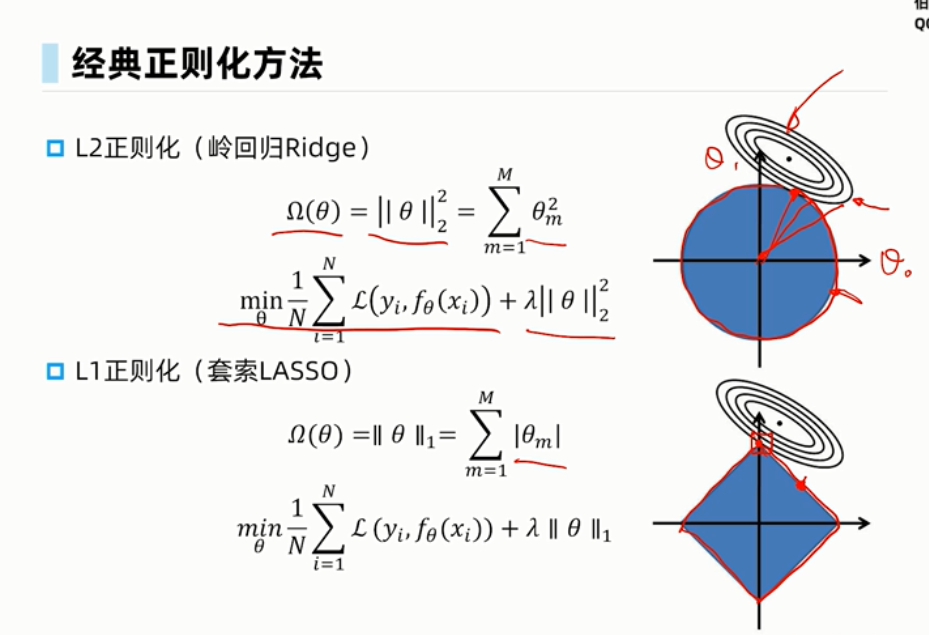
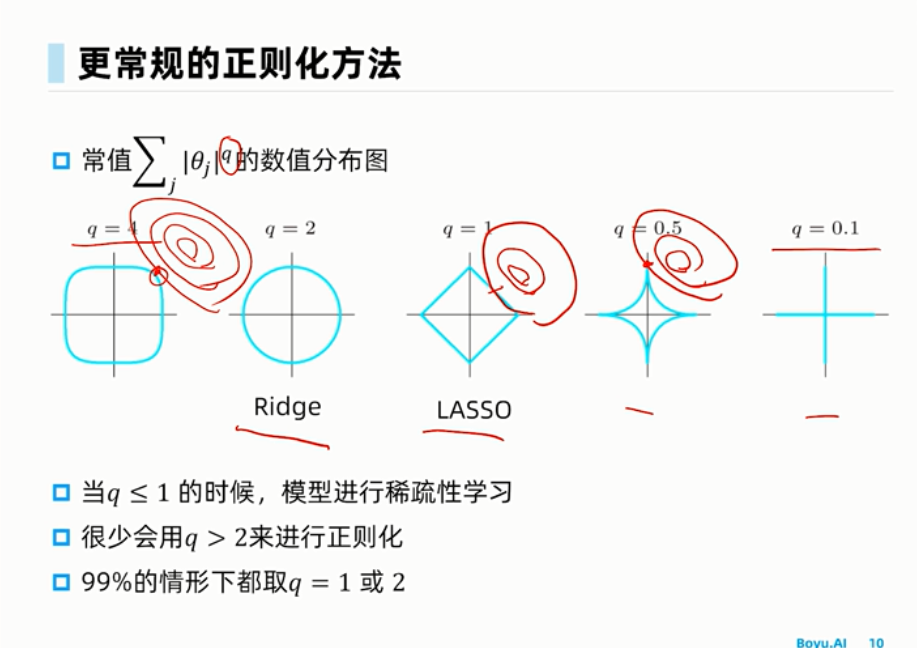
机器学习

监督（预测输出，带标签） 无监督 强化（决策）



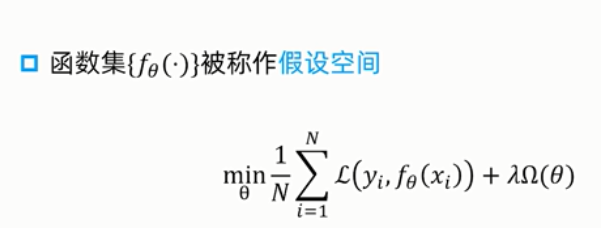
正则化：添加惩罚项



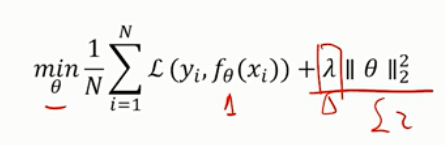


## 奥卡姆剃刀原则

有多个假设模型时，我们应该选择假设条件最少的建模方法。（少用特征或者特征交互少）



## 模型选择



一个机器学习的解决方案的模型包含参数和优化超参口超参数

**超参数**（模型不会自己学习，人为设置）

定义模型的更高层次的概念，如复杂性或学习能力。

在标准模型训练过程中**无法直接从数据中学习**，需要预先定义。

可以通过不同的参数设置、训练不同的模型以及选择最好的测试结果来进行超参数选择

**模型选择（或超参数优化）关注如何选择最佳超参数**

## 交叉验证



1. 折交叉验证
2. 1.设置超参数

2.将原始训练数据随机拆分为K份

3.重复K次：

若当前为第次重复（i=1?，K），选择第i份数据作为验证数据集，其余K-1份作为训练数据集

对训练数据进行建模，并在验证数据上对其进行评估，从而获得评估分数

4.对K个评估分数取平均作为模型性能

## 模型范化性

模型泛化性知识逻辑

* 指模型对潜在的/未知的数据的预测能力
* 理论上可以用泛化误差进行评估，但是p(x,y)是未知的；转向使用模型在有限的数据集下、关于****测试集****的误差来进行经验估计，其合理性可以由大数定律解释
* 最后给出泛化误差的估计与其真实值之间的关系

泛化能力指的是模型对未观测数据的预测能力 期望？？？

## 判别模型

**判别模型**是**对可观测变量和未知变量的关联性建模**，包括线性回归，逻辑回归，k近邻，支持向量机，（多层）感知机，决策树，随机森林等。

**生成模型是建模数据的联合概率分布**，包括朴素贝叶斯，隐马尔可夫模型，混合高斯，马尔可夫随机场，隐狄利克雷分布 (LDA)等。

1.判别模型：条件概率

x= x1 条件下y = y1发生 的概率

应用：线性回归，逻辑回归

2.生成模型：联合概率分布

特点：全面

缺点：需要知道机理

应用：朴素贝叶斯，隐马尔科夫模型，混合高斯模型，马尔科夫随机场，隐狄利克雷分布

## 数学期望

(mean)（或[均值](https://baike.baidu.com/item/%E5%9D%87%E5%80%BC/5922988" \t "https://baike.baidu.com/item/%E6%95%B0%E5%AD%A6%E6%9C%9F%E6%9C%9B/_blank)，亦简称期望）是试验中每次可能结果的[概率](https://baike.baidu.com/item/%E6%A6%82%E7%8E%87/828845" \t "https://baike.baidu.com/item/%E6%95%B0%E5%AD%A6%E6%9C%9F%E6%9C%9B/_blank)乘以其结果的总和，是最基本的数学特征之一。它反映随机变量平均取值的大小。

[大数定律](https://baike.baidu.com/item/%E5%A4%A7%E6%95%B0%E5%AE%9A%E5%BE%8B/410082" \t "https://baike.baidu.com/item/%E6%95%B0%E5%AD%A6%E6%9C%9F%E6%9C%9B/_blank)规定，随着重复次数接近无穷大，数值的算术平均值几乎肯定地收敛于期望值。

泊松分布

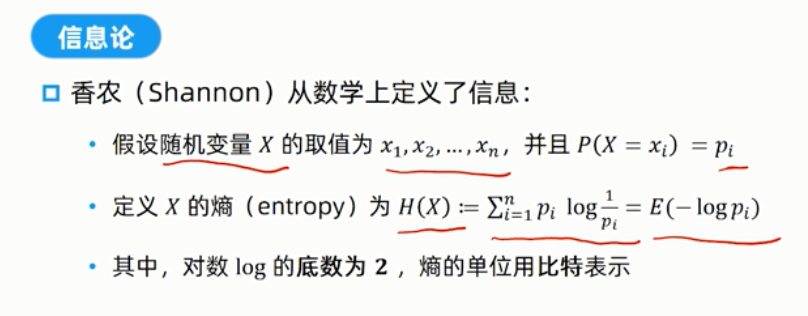
## 最大似然估计

这个概率最大可能估计

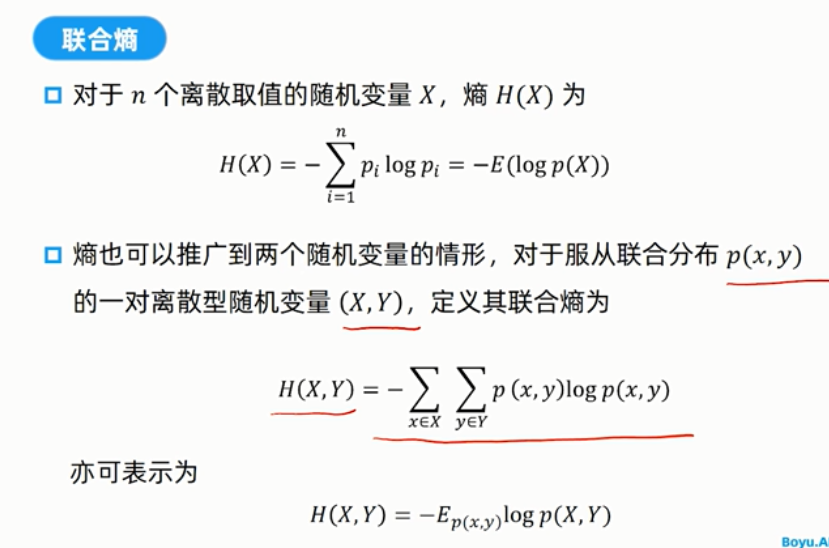
## 信息论

对不确定性的发生概率衡量 对不确定性的消除

香农



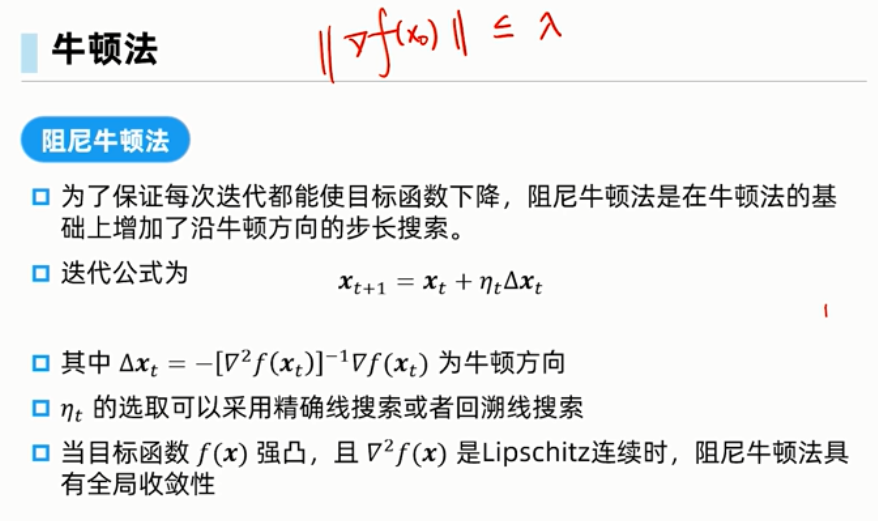
x的信息量



上面E有问题，为 -p log2 p

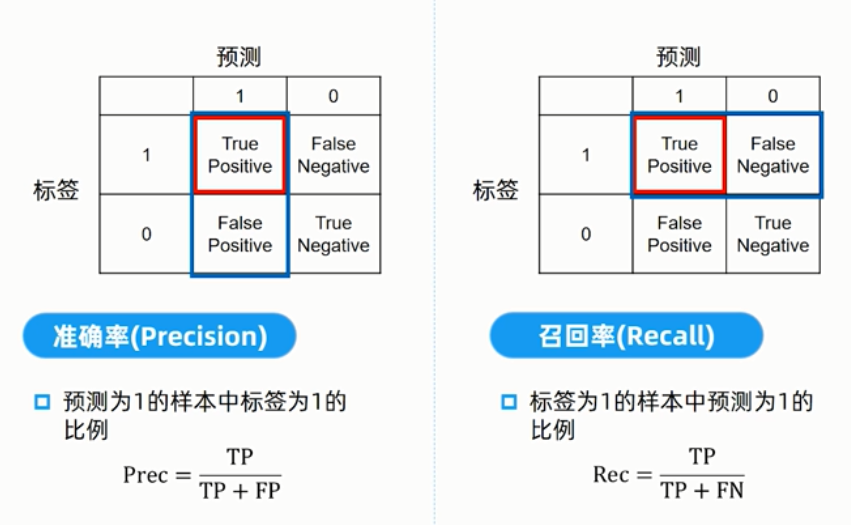
## 牛顿法

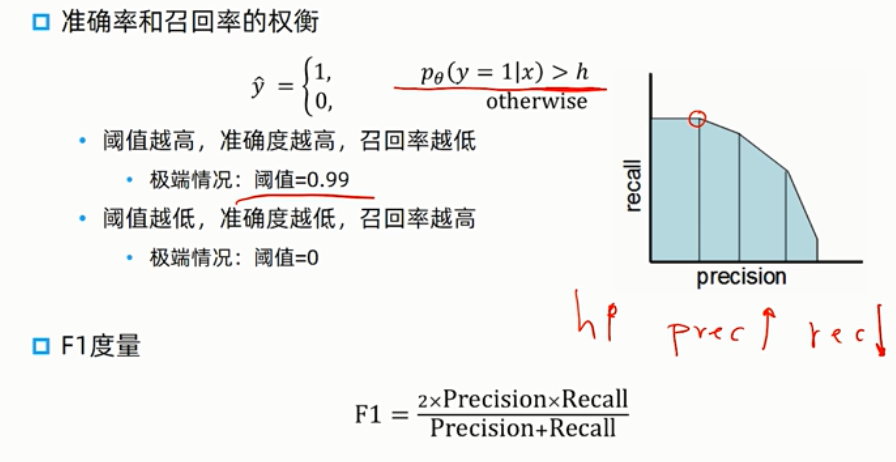
梯度下降 下降的地方为二次求导

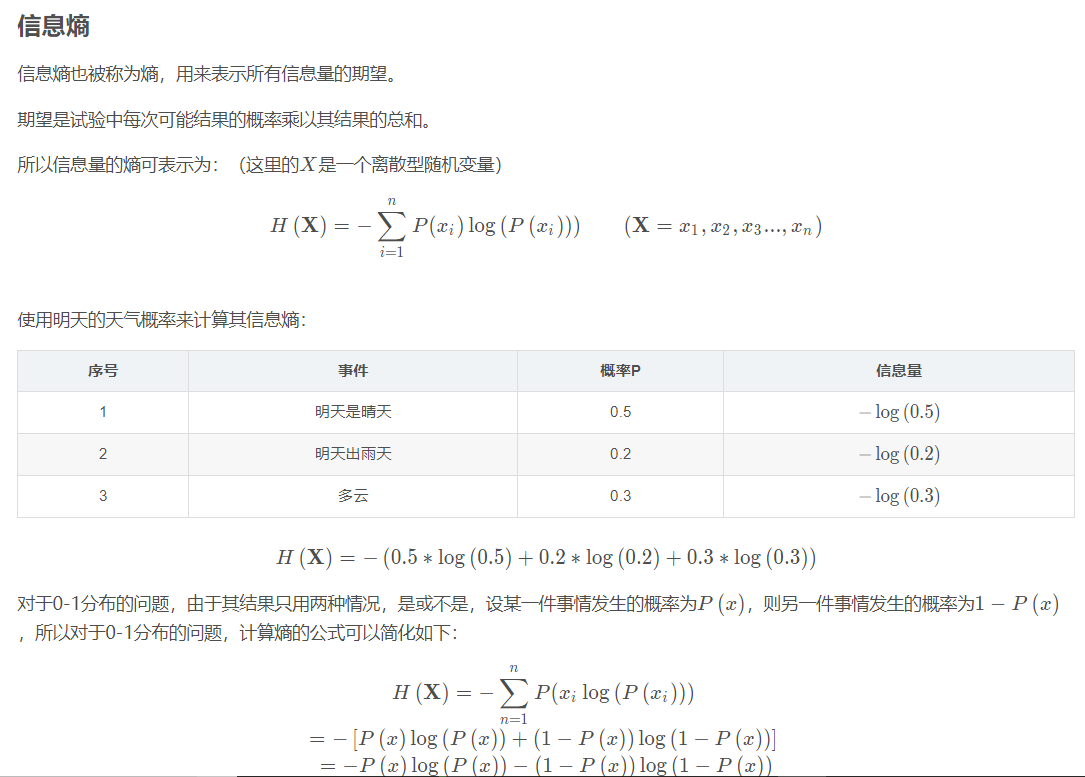


## 评估指标

准确率 召回率 F1



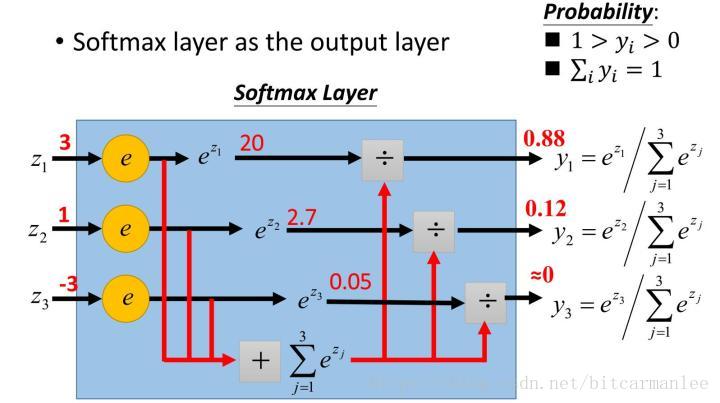


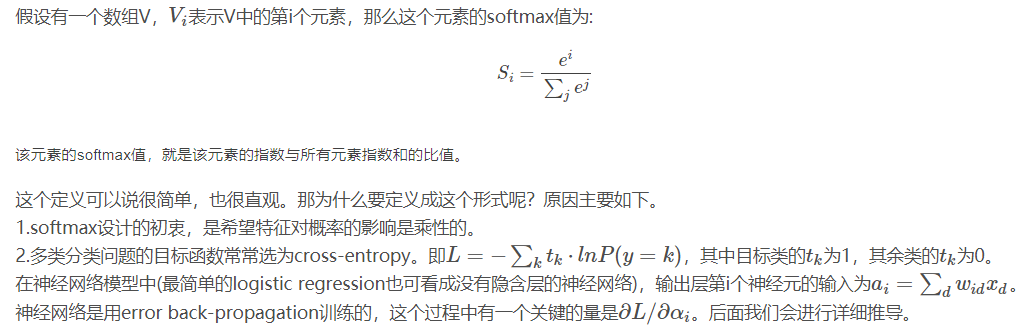


# Softmax

和线性的主要区别 损失函数不一样

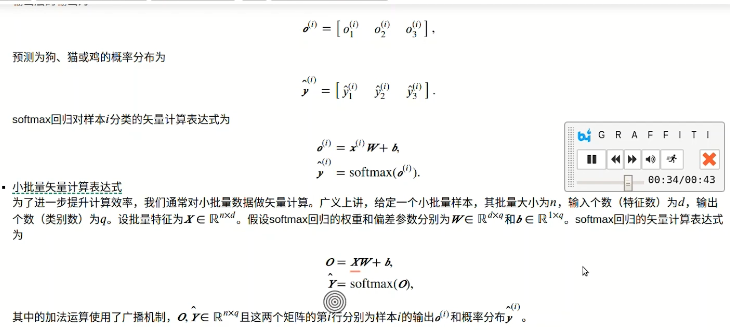
## Softmax 计算过程





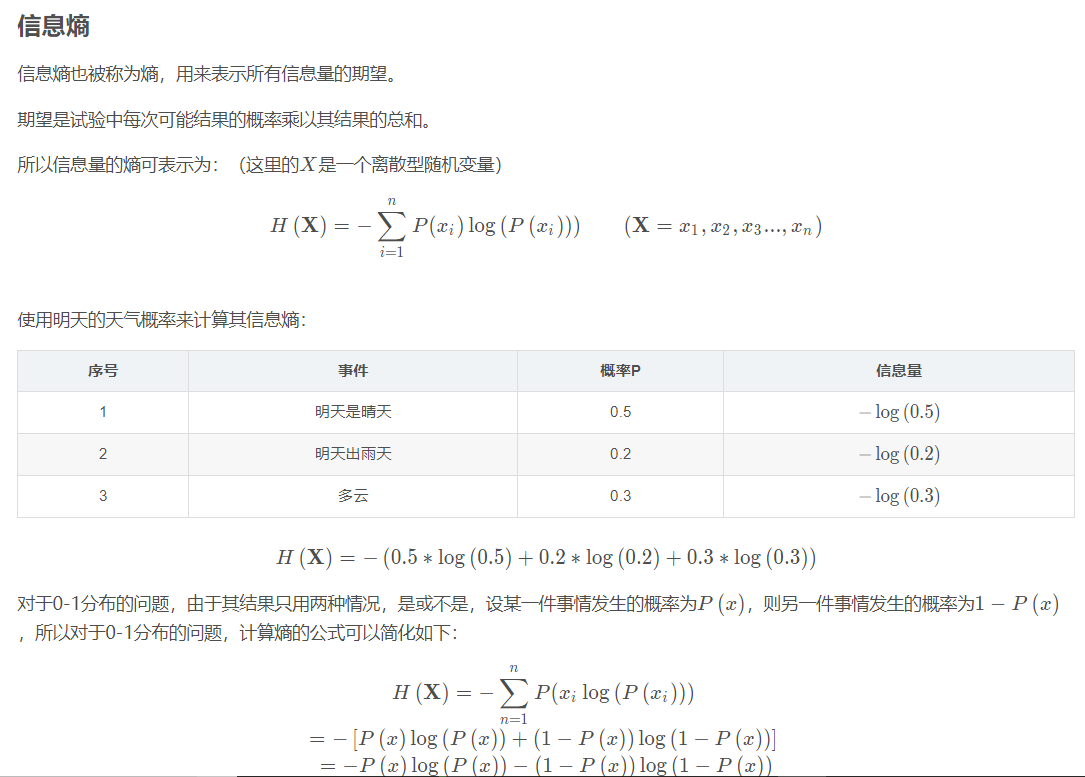
优点

1. 一方面，由于输出层的输出值的范围不确定，我们难以直观上判断这些值的意义。
2. 另一方面，由于真实标签是离散值，这些离散值与不确定范围的输出值之间的误差难以衡量。



在计算softmax概率的时候，为了保证数值稳定性（numerical stability），我们可以选择给输入项减去一个常数，比如x的每个元素都要减去一个x中的最大元素。当输入项很大的时候，如果不减这样一个常数，取指数之后结果会变得非常大，发生溢出的现象，导致结果出现inf。

## 交叉熵损失函数



平方和损失函数在0.8平方的时候更小了，为了克服这个问题，采用交叉熵损失函数

# 多层感知机

### **关于激活函数的选择[¶](https://www.kesci.com/api/notebooks/5e450d4e5f2816002cecfbf2/RenderedContent?embed_doc=true" \l "%E5%85%B3%E4%BA%8E%E6%BF%80%E6%B4%BB%E5%87%BD%E6%95%B0%E7%9A%84%E9%80%89%E6%8B%A9)**

ReLu函数是一个通用的激活函数，目前在大多数情况下使用。但是，ReLU函数只能在隐藏层中使用。

用于分类器时，sigmoid函数及其组合通常效果更好。由于梯度消失问题，有时要避免使用sigmoid和tanh函数。

在神经网络层数较多的时候，最好使用ReLu函数，ReLu函数比较简单计算量少，而sigmoid和tanh函数计算量大很多。

在选择激活函数的时候可以先选用ReLu函数如果效果不理想可以尝试其他激活函数

### **多层感知机**

多层感知机就是含有至少一个隐藏层的由全连接层组成的神经网络，且每个隐藏层的输出通过激活函数进行变换。多层感知机的层数和各隐藏层中隐藏单元个数都是超参数。以单隐藏层为例并沿用本节之前定义的符号，多层感知机按以下方式计算输出

### **反向传播(backpropagation)**

当建立神经网络的时候，会初始化一个权重和偏差，在一次迭代的过后，计算出网络的错误，然后将该错误与成本函数的梯度一同反馈给网络用来更新网络中的权重，以减少后续迭代的错误。使用成本函数梯度的权重的更新成为反向传播。

### 前向传播

求导

# 文本预处理

1. 读入文本
2. 分词
3. 建立字典，将每个词映射到一个唯一的索引（index）
4. 将文本从词的序列转换为索引的序列，方便输入模型

自己定义分词规则

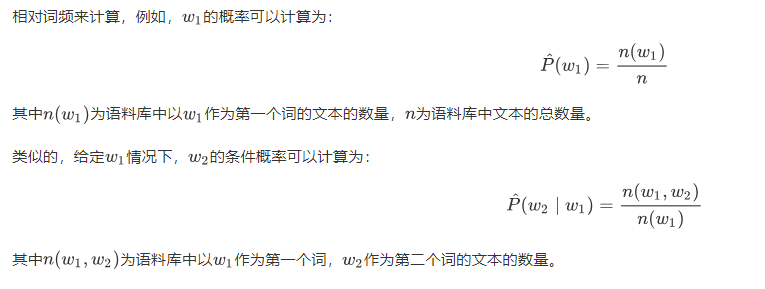
用工具分词

# 语音模型

给定一段序列，判断序列是否合理，对给定序列求概率，概率大合理。

第一个词存在比第二个词的概率

介绍字符随机采样与相邻采样 详细代码（回去重新看下）

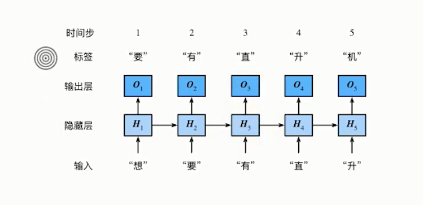


马尔科夫假设是指一个词的出现只与前面n个词相关

# 循环神经网络

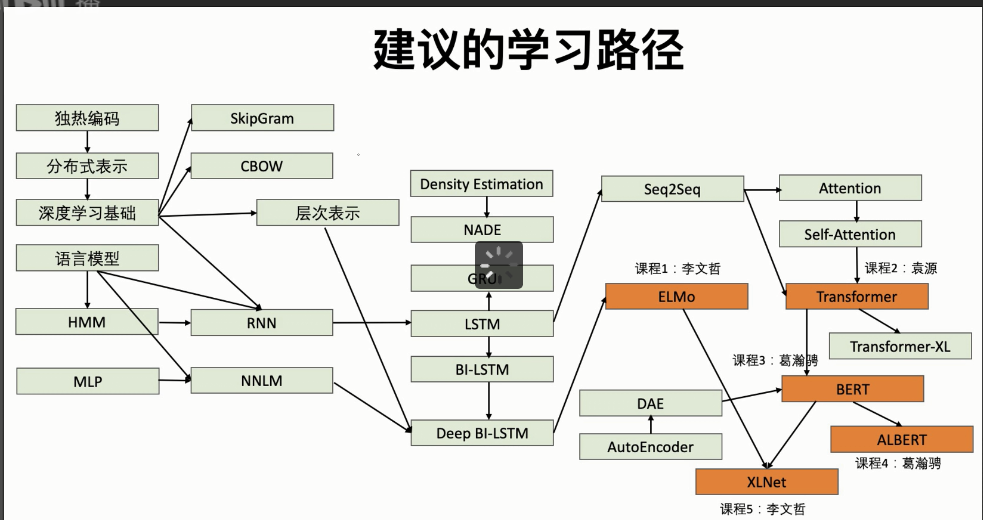
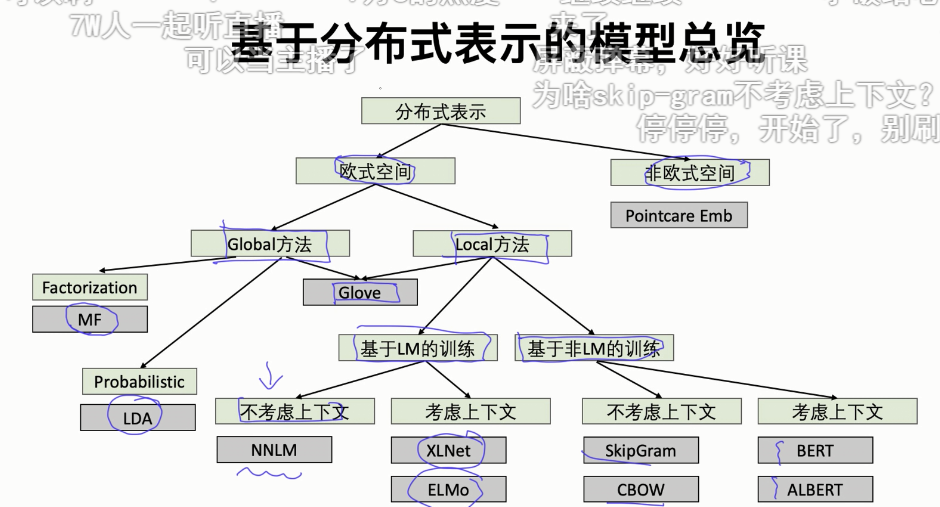
无论是卷积神经网络，还是人工神经网络，他们的前提假设都是：元素之间是相互独立的，****输入与输出也是独立的****，比如猫和狗。

在循环神经网络（RNN）中，每输入一步，每一层各自都共享参数U,V,W

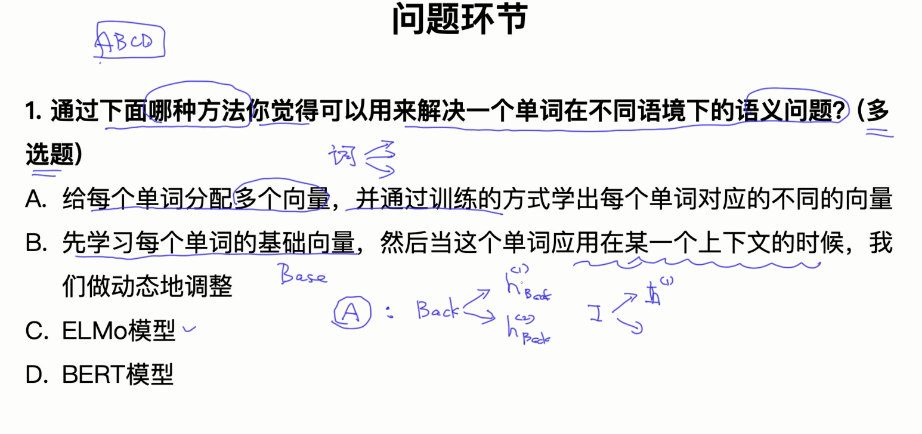


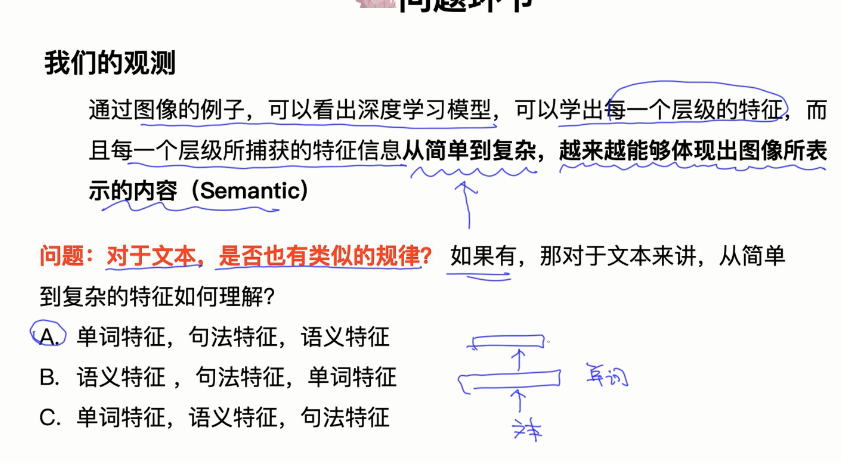
H2输出与H1的输入与要有关

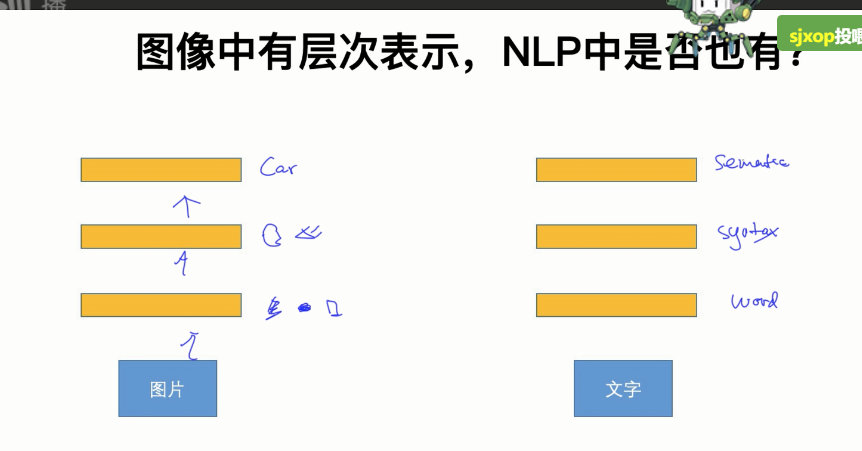
# 基于分布表示的模型

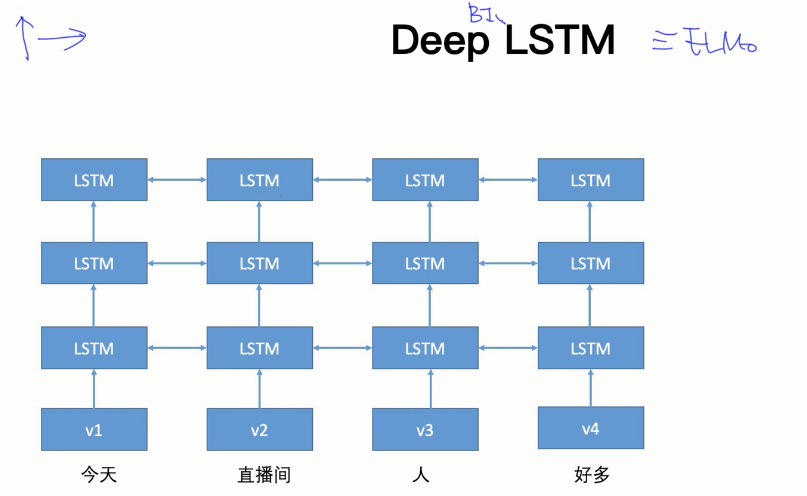


研究重点 语义问题



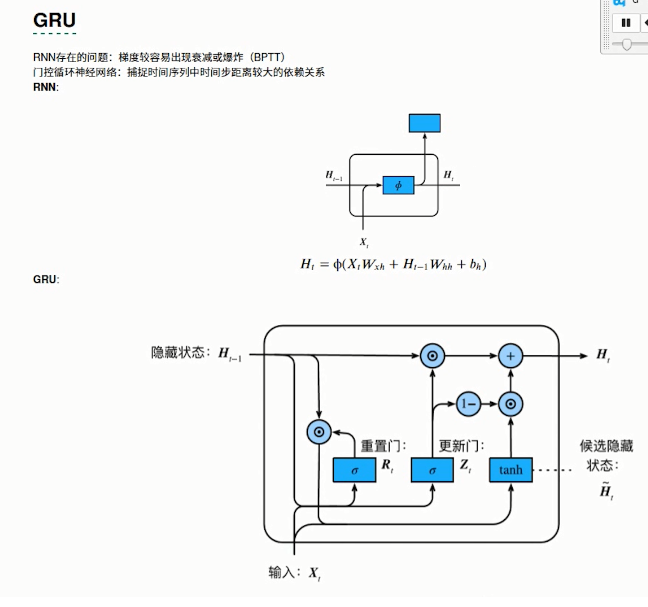


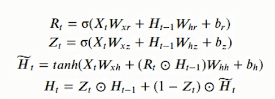




# 循环神经网络进阶

进阶对比





大体不变，增加了门控制前面信息与当前信息状态

从输出看Ht由Ht-1与 Ht~控制

公式

# 卷积神经网络基础

核： 小矩阵

互相关：乘上核

特征图：

感受野：

填充和步幅：外面填充0 核移动的间隔

卷积层和池化层，并解释填充、步幅、输入通道和输出通道的含义。

最常见的二维卷积层，常用于处理图像数据

# LeNet

1. lenet 模型介绍
2. lenet 网络搭建
3. 运用lenet进行图像识别-fashion-mnist数据集

卷积层块里的基本单位是卷积层后接平均池化层：卷积层用来识别图像里的空间模式，如线条和物体局部，之后的平均池化层则用来降低卷积层对位置的敏感性。

使用全连接层的局限性：

* 图像在同一列邻近的像素在这个向量中可能相距较远。它们构成的模式可能难以被模型识别。
* 对于大尺寸的输入图像，使用全连接层容易导致模型过大。

使用卷积层的优势：

* 卷积层保留输入形状。
* 卷积层通过滑动窗口将同一卷积核与不同位置的输入重复计算，从而避免参数尺寸过大。

# 批量归一化（BatchNormalization）

**对输入的标准化（浅层模型）**

处理后的任意一个特征在数据集中所有样本上的均值为0、标准差为1。  
标准化处理输入数据使各个特征的分布相近

**批量归一化（深度模型）**

利用小批量上的均值和标准差，不断调整神经网络中间输出，从而使整个神经网络在各层的中间输出的数值更稳定。

**2.对卷积层做批量归⼀化[¶](https://staticcdn.boyuai.com/course/jupyter/360/3-vp7VQrHu9WZ3icdyPbd.html" \l "2.%E5%AF%B9%E5%8D%B7%E7%A7%AF%E5%B1%82%E5%81%9A%E6%89%B9%E9%87%8F%E5%BD%92%E2%BC%80%E5%8C%96)**

位置：卷积计算之后、应⽤激活函数之前。  
如果卷积计算输出多个通道，我们需要对这些通道的输出分别做批量归一化，且每个通道都拥有独立的拉伸和偏移参数。 计算：对单通道，batchsize=m,卷积计算输出=pxq 对该通道中m×p×q个元素同时做批量归一化,使用相同的均值和方差。

**3.预测时的批量归⼀化**

训练：以batch为单位,对每个batch计算均值和方差。  
预测：用移动平均估算整个训练数据集的样本均值和方差。