# **多层感知机**

**1.输入层 读取给予神经网络的信息**

**2.隐藏层 提取输入信息的特征，通过激活函数和加权输入计算输出值**

**3.输出层 与隐藏层一样处理信息，并显示神经网络的计算结果**

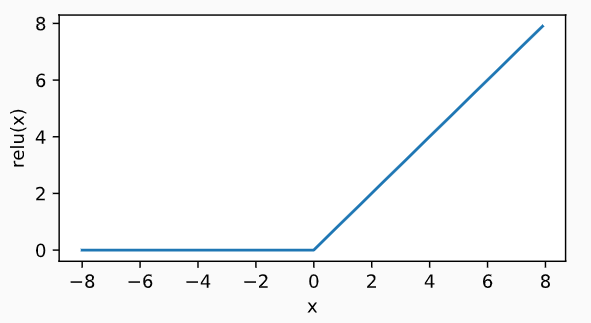
**多层感知器**

**1. 定义：多层感知机即在单层神经网络的基础上引入一到多个隐藏层**

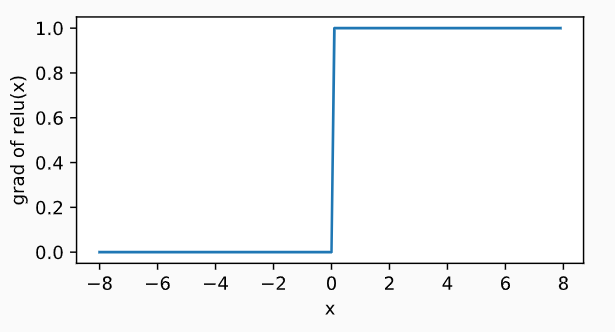
**2.激活函数： 若隐藏层只含有权重参数和偏差参数，则输出层与输入层总是呈线性关系，为了构建更复杂的模型，更好的拟合数据，需在隐藏层中引入激活函数，将输入至隐藏层的值进行激活后输出，常用的激活函数有如下几种**

**2.1 ReLU函数**

**ReLU(x)=max(x,0) ReLU函数只保留正数元素，并将负数元素清零**

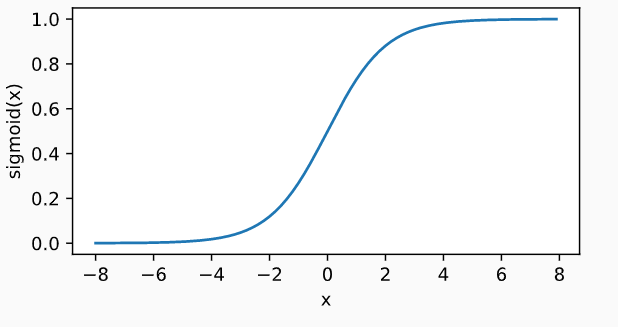
****

**导数**

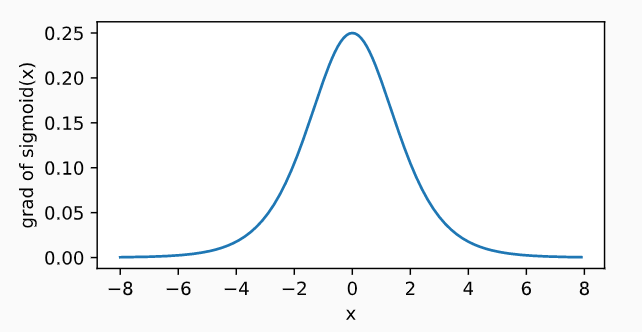
****

**2.2 sigmoid函数**

**sigmoid(x)=1/(1+exp(−x)) sigmoid函数可以将元素的值变换到0和1之间**

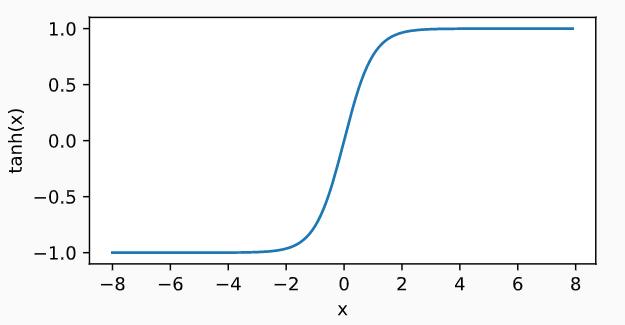
****

**导数**

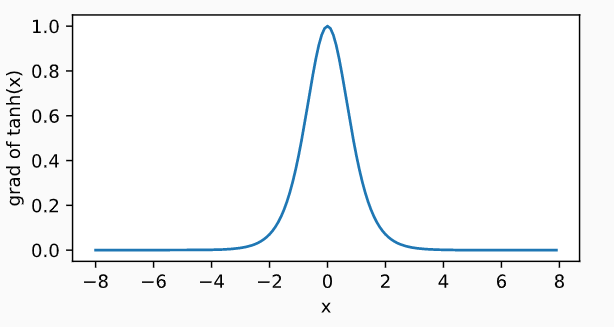
****

**2.3 tanh(双曲正切)函数**

**tanh(x)=(1−exp(−2x))/(1+exp(−2x)) tanh函数可以将元素的值变换到-1和1之间**

****

**导数**

****

**5.多层感知器与神经网络**

**感知器使用非线性的激活函数来引导神经元的输出，之所以使用非线性函数是因为更切合真实世界(多数数据都是非线性的)。信号进入输入层（Input layer),通过该层的激活函数后进入下一层隐藏层（hidden layer),经过隐藏层的激活函数再继续往下一个隐藏层传递，知道进入输出层（output layer).这些非线性的激活函数使得感知器有很强的获取信息的能力。**

**（王树仁总结）**

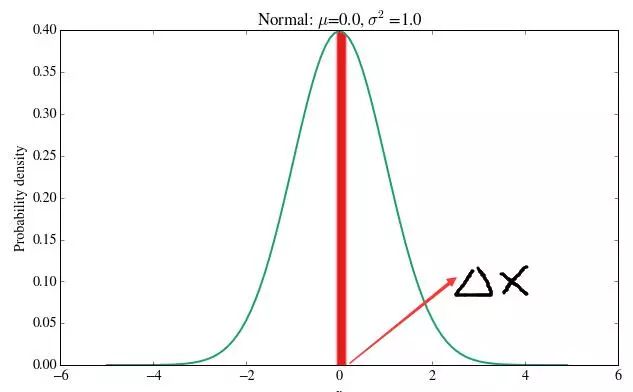
**Maximum likelihood（最大似然估计）**

**在统计学中，最大似然估计(Maximum likelihood estimation, MLE)，也称为最大概似估计，是用来估计一个概率模型(概率模型是用来描述不同随机变量之间关系的数学模型，通常情况下刻画了一个或多个随机变量之间的相互非确定性的概率关系)的参数的一种方法。**

**最大似然估计的原理：给定一个概率分布D，已知其概率密度函数(连续分布)或概率质量函数(离散分布)为fD,以及一个分布参数𝞱，我们可以从这个分布中抽出一个具有n个值的采样X1,X2,…,Xn，利用fD计算出概率：P(x1,x2,…,xn)=fD(x1,…,xn|𝞱)。**

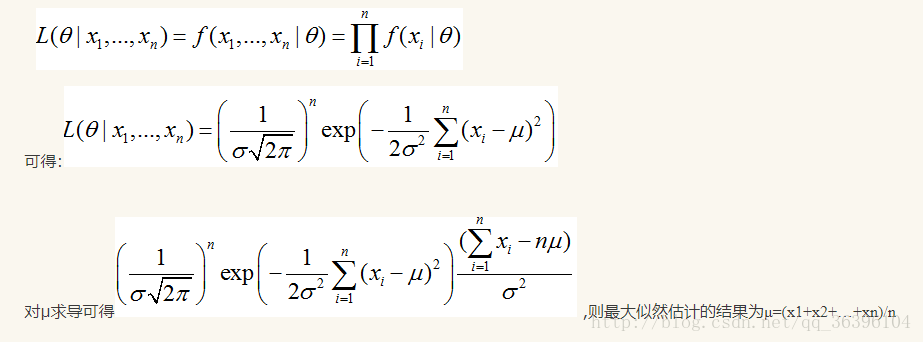
**但是，我们可能不知道𝞱的值，尽管我们知道这些采样数据来自于分布D。那么我们如何才能估计出𝞱呢？一个自然的想法是从这个分布中抽出一个具有n个值的采样X1,X2,…,Xn，然后用这些采样数据来估计𝞱。**

**假设某正态分布的概率密度分布图如下**

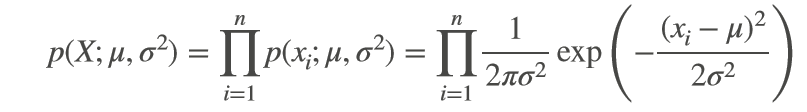
****

**那么变量等于0的概率密度是0.4 ，但这个点的概率理论上应该是0，可以求 (0,0 + △x) 的概率，那么 (0，0 +△x) 的概率可以近似于红色矩形的面积，所以 (0.,0. + △x) 的概率就是 0.4△x。假如△x很小，那么就可以近似的把 0 处的概率当做 0.4△x**

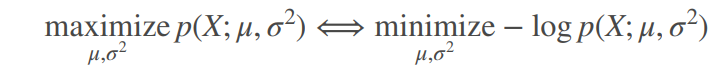
**二维的高斯分布：**

****

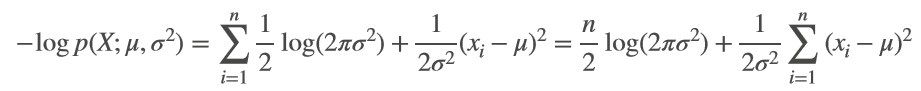
**从高斯函数模型中取数据得出公式：**

****

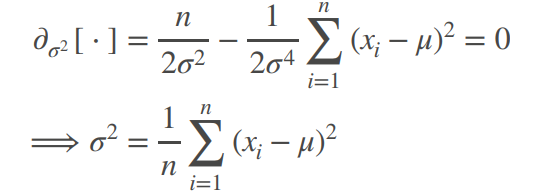
**最大似然化简为：**

****

**分解后者：**

****

**对结果求导：**

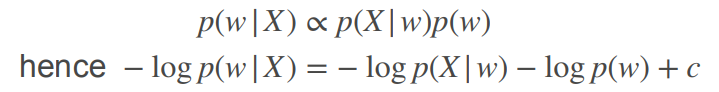
****

**最大后验概率估计（MAP）是求参数𝞱, 使似然函数P(x0|𝞱)P(x0|𝞱)最大。最大后验概率估计则是想求𝞱使P(x0|𝞱)P(𝞱)最大。求得的𝞱不单单让似然函数大，𝞱自己出现的先验概率也得大。**

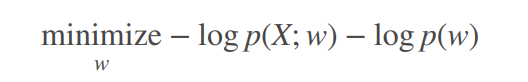
**MAP其实是在最大化**

**MAP就是多个作为因子的先验概率P(𝞱)。或者，也可以反过来，认为MLE是把先验概率P(𝞱)认为等于1，即认为𝞱是均匀分布。**

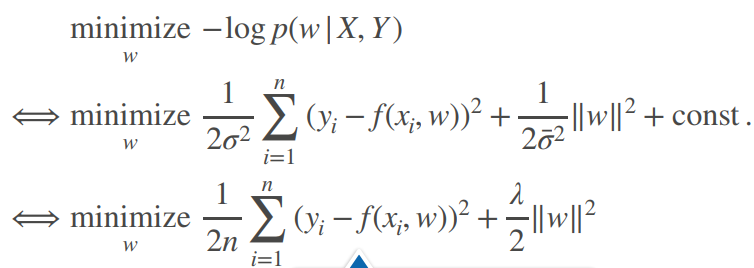
**后验概率：**

****

**最大后验概率估计：**

****

**与之前的估计回归问题结合得出公式：**

**‘**

**loss函数**

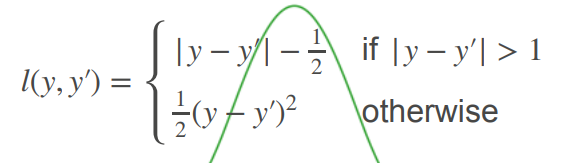
**1. l1\_loss&l2\_loss**

**衡量预测值与真实值的偏差程度的最常见的loss：误差的L1范数和L2范数**

**因为L1范数在误差接近0的时候不平滑，所以比较少用到这个范数**

**L2范数的缺点是当存在离群点（outliers)的时候，这些点会占loss的主要组成部分。比如说真实值为1，预测10次，有一次预测值为1000，其余次的预测值为1左右，显然loss值主要由1000主宰。**

**Huber Loss经常用于回归问题，相比与l2 loss,其对离群点（outliers)没有那么敏感（因为如果残差太大的话，由于是分段函数，loss为残差的线性函数）**

**函数定义：**

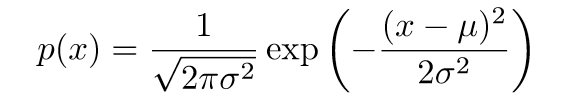
**Y表示真实值，y’表示预测值**

**（刘一鸣总结）**

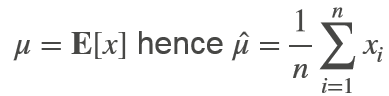
**Likelihood&Loss Functions**

**最大似然：**

**正态分布函数 ：**

****

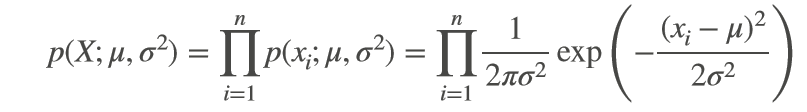
**平均值：**

****

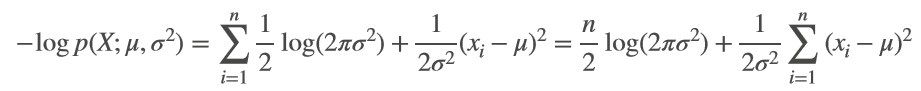
**方差：**

****

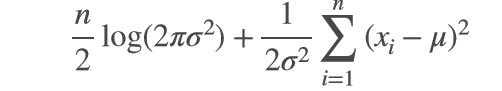
**似然：**

****

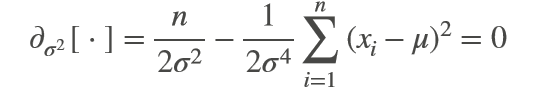
**伴随似然：**

****

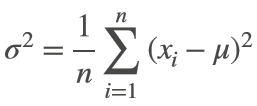
**估计方差可以得到：**

****

**求导可得：**

****

**得出方差：**

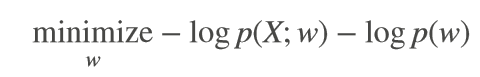
****

**最大似然估计：**

**后验概率：p(w|X)正比于p(X|w)p(w)**

**两年同求对数得 −logp(w|X)=−logp(X|w)−logp(w)+c**

**最大后验估计：**

****

**回归：**

**n**

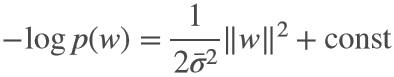
**Minimize ∑ (yi−f(xi,w))2+−logp(w)(w为增加的高斯噪声)**

**w i=1**

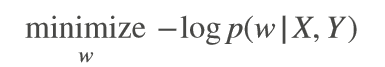
**数据生成模型：**

**yi =f(xi,w)+ϵi where ϵi ∼𝒩(0,σ2)**

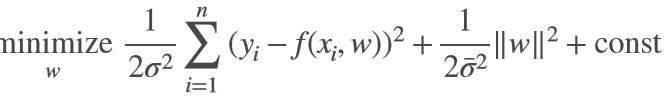
**因此：**

****

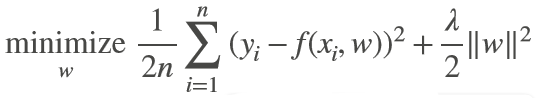
**最大化后验：**

****

**即：**

****

**即：**

****

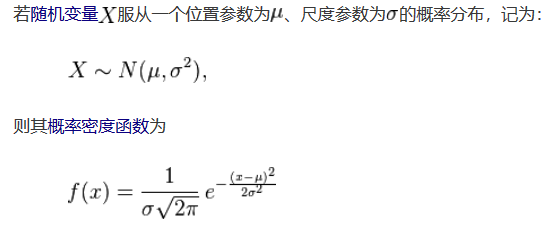
**（叶钰莹总结）**

**概率密度：变量在一个区间（事件的取值范围）的总的概率除以该段区间的长度。**

**概率密度函数：描述随机变量在某个确定的取值点附近的可能性的函数。**

**一．极大似然**

**1.高斯分布：**

****

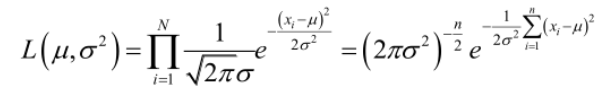
**图像关于μ对称**

**2.极大似然估计（利用已知的样本结果，反推最大可能导致这样结果的参数值）**

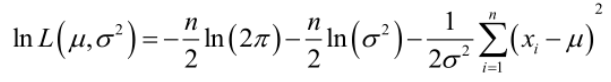
**求解最大似然估计值：**

**Eg:当样本服从1中所说的正态分布**

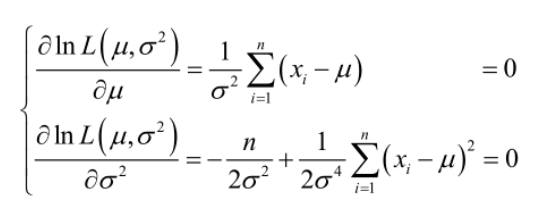
**1).写出似然函数**

****

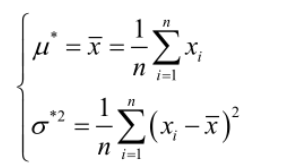
**2).对似然函数取对数**

****

1. **.求导数**

****

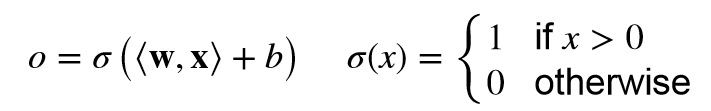
**4).解似然方程**

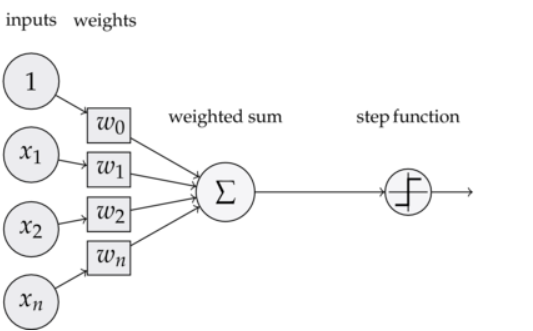
****

**二．感知器**

**感知器应该是一种具有输入和输出的算法，给定输入值后，输出既定的值。**

**1.输入x,权重w和偏差b。感知器输出：**

****

**如图：**

**2.感知器的训练**

**初始化w=0,b=0**

**执行：**

****

**直到Yi[<w,Xi>+b]>0**

1. **Softmax（柔性最大值传递函数）回归（徐明雪）**

**附：1.输出为离散值**

**2.输出为多个**

**1.问题起源：分类问题**

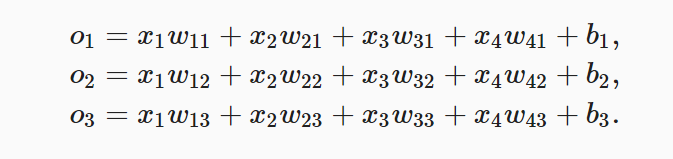
1. **假设训练数据集中图像的真实标签为狗，猫，鸡（假设用4像素x1,x2,x3,x4来表示）**
2. **通常离散的数值来表示类别**

**2．softmax回归模型**

1. **与“线归”不同，输出个数等于标签类别数**
2. **权重包含12个标量（w）**

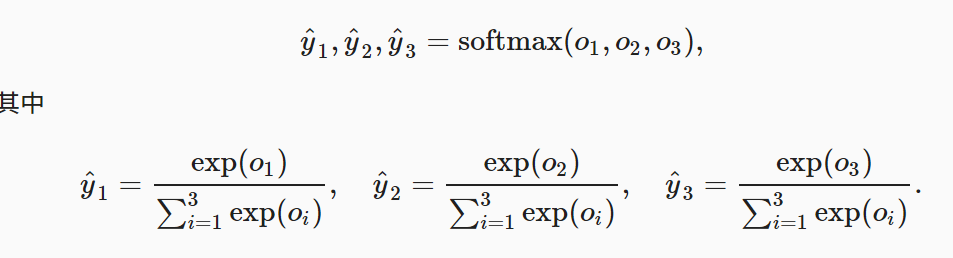
**偏重包含3个标量（b）**

**且对每个输入，计算O1,O2,O3,这三个输出**

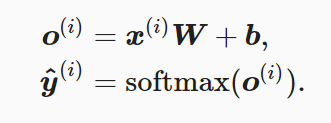
****

**3.softmax运算**

1. **值最大的输出对应类别作为预测输出**

****

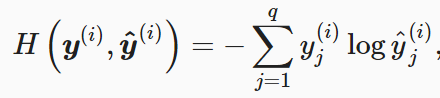
**4.单样本分类的矢量表达式**

****

**5.小批量样本分类的矢量计算表达式**

**6.交叉熵损失函数**

1. **平方损失可以，但是太严格**
2. **所以用交叉熵**

****

**上个式子中，y(i)非0，即1，则可简化为**

**（自然对数）**

1. **图像分类数据集**

**注：type（）返回参数的数据类型**

**dtype（）返回数组中元素的数据类型**

**MNIST:**

1. **MNIST:**

**数据集是计算机视觉和机器学习文献中最受研究的数据集之一。这个数集的目标是正确的分类手写数字0-9。在许多例子中，这个数据集是作为基准（benchmark），用于机器学习排名的基准。实际上，MNIST数据集用在深度学习的训练神经网络模型和其它语言中的“Hello World”示例是一样的。**

**MNIST本身包含60000张训练图片和10000张测试图片。每张图片是784维的特征向量，即对应于图像的28\*28的灰阶像素强度。这个像素值是[0, 255]之间的无符号整数。所有数字放置在前景白色背景黑色的区域。**

**我们主要在Starter Bundle的早期章节中使用这个数据集用于进入神经网络的学习。**

1. **最大似然估计**
2. **似然与概率在统计学中是不一样的，结果和参数相互对应的时候，似然和概率在数值上是相等的。**

**这里写图片描述2、概率可表示为：P(x|θ)，是条件概率的表示方法，θ是前置条件，理解为在θ 的前提下，事件 x 发生的概率，相对应的似然可以表示为：**

**这里写图片描述https://img-blog.csdn.net/20171007205955013?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvcXFfMzYzOTYxMDQ=/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast3、伯努利分布，两点分布。**

**表示形式： 或**

**注：1.与概率分布图不同的是，似然函数是一个(0, 1)内连续的函数，所以得到的图也是连续的，我们很容易看出似然函数的极值（也是最大值）在 p=0.5p=0.5 处得到，通常不需要做图来观察极值，令似然函数的偏导数为零即可求得极值条件。**

**4、似然函数的最大值**

**概率描述的是在一定条件下某个事件发生的可能性，概率越大说明这件事情越可能会发生；而似然描述的是结果已知的情况下，该事件在不同条件下发生的可能性，似然函数的值越大说明该事件在对应的条件下发生的可能性越大。**

**5、最大似然估计的一般求解过程：**

**（1） 写出似然函数；**

**（2） 对似然函数取对数，并整理；**

**（3） 求导数 ；**

**（4） 解似然方程**

**逻辑回归：（梁明总结）**

1. **用途：用于解决离散值预测问题。**
2. **特点：运用softmax运算，输出单元由一个变成多个，解决多分类问题。**
3. **softmax回归模型：将输入特征与权重做线性叠加（输出值各数等于标签里的类别数）**
4. **softmax运算：通过y^1,y^2,y^3=softmax(o1,o2,o3)将输出值变为值为正且和为1的概率分布。**

**回归和分类的区别：（张德龙总结）**

**1、回归估计一个连续的值，且只有一个输出，并且通过计算真实值和预测值之间的误差（第二范数）实现最小化损失函数（如线性回归，逻辑回归，估计值为连续的）**

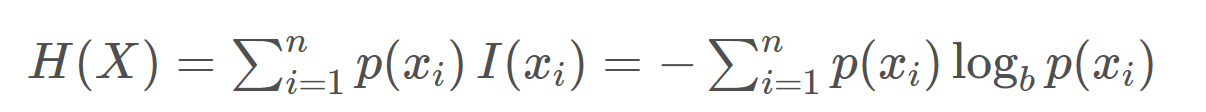
**2、而分类预测一个离散值，输出有多个值，通过计算预测值的对应概率实现分类任务，通过计算交叉熵计算概率导致的误差（如识别手写数字或者衣服种类，预测值为离散值）**

**信息论：**

**信息熵：随机变量自信息量I(xi)的数学期望（平均自信息量），用H（X）表示，即为熵的定义：**

**即一个值域为{x1, …, xn}的随机变量 X 的熵值 H 定义为：**

****

**其中，E 代表了期望函数，而 I(X) 是 X 的信息量（又称为信息本体）。I(X) 本身是个随机变量。如果 p 代表了 X 的机率质量函（probability mass function），则熵的公式可以表示为**

**在这里 b 是对数所使用的底，通常是 2，当b = 2，熵的单位是bit；**

**（胡涛总结）**

**信息论研究一个事件包含信息的量化，一个发生概率小的事发生，不确定性大，比大概率事发生提供更多信息。**

**信息熵表示对依据概率分布生成符号进行编码所需的比特数在平均意义上的下界。**

**直观的信息熵是在最优编码中对于一定判断或信息进行判断的比特数下界。**

**(邓楠补充)**

**相对熵**

1. **分布之间的距离（如真实值和估计值之间）**

**使用错误代码时的额外位数**

**D[p||q]== ∫ dp(x)log p(x)/q(x)= ∫ dp(x)[−log q(x)] （低效位）− [−log p(x)]（最佳位）;**

**kl散度的非负**

**D[p||q]== ∫ dp(x)log p(x)/q(x)=;**

**D[p||q]== ∫ dp(x)log p(x)/q(x)>=-∫ dp(x)log p(x)/q(x)=0;**

1. **（回到）交叉熵损失**

**交叉熵损失**

**l(y, o) = log∑iexp(oi) − y⊤o**

**kullback-leber分歧**

**D(softmax(o)∥q) = ∑iqi log qi − qi log softmax(o)i= − H[q] + log∑iexp(oi) − ∑qioi；**