一、熵（Entropy）

直观理解：事件的混乱程度—>事件的不确定性—>事件所含的信息量。

基本假设：发生概率(p)越小的事件(x)所包含的信息量就越大（bits），即

$$I(x) = -log_2 p(x)$$

其中，$I(x)$表示事件x所含的信息量。

信息熵（Shannon Entropy）：指整个系统中的不确定性的程度、信息量。

$$H(x) = E_{x\sim p}[I(x)] = -E_{x\sim p}[log_2 p(x)]$$

其中，$H(x)$表示信息熵，E表示期望，$x\sim p$表示事件x的离散分布，$I(x)$表示事件x所含的信息量。对上式进行进一步化简，可得：

$$H(x) = -E_{x\sim p}[log_2 p(x)] = -\sum_i p_i log_2 p_i$$

举例：

系统	事件x的分布	信息熵$H(x)$
Case1	$p(A) = 1$	$-log_2 1 = 0$
Case2	$p(A) = p(B) = 0.5$	$\sum_{i=1}^{2} \frac{1}{2} log_2 \frac{1}{2} = 1$
Case3	$p(A) = p(B) = p(C) = p(D) = 0.25$	$\sum_{i=1}^{4} \frac{1}{4} log_2 \frac{1}{4} = 2$

可见，当一个系统中的事件越复杂，其所包含的信息熵越大，即信息量越大。

必要知识补充 1：

香农（Shannon）第一定理指出信息熵(H)是无失真信源编码的极限值，若编码的平均码长小于信息熵值，则必然发生差错（即有损）。例如在上述举例中，当我们采用二进制编码时，$Case1$所需的最短无损编码为$0bits$，$Case2$所需的最短无损编码为$1bits$，$Case3$所需的最短无损编码为$2bits$。

而应用在机器学习的损失函数计算中，我们通常采用自然常数e作为对数的底，来进行信息熵H的计算，即：

$$H(x) = -E_{x\sim p}[log_2 p(x)] \longrightarrow H(x) = -E_{x\sim p}[lnp(x)]$$

$$bits \longrightarrow nats$$

二、交叉熵（Cross Entropy）

直观理解：用于度量两个概率分布之间的差异性信息。

交叉熵公式：

若假设$p(x)$为目标分布（通常未知），$q(x)$为模型（model）输出的预测分布，则 Cross Entropy 为：

$$H(p,q) = -E_{x\sim p}[lnq(x)] = -\sum_i p_i lnq_i$$

其中，$H(p,q)$表示用$q(x)$分布替换$p(x)$分布所需的编码量。而根据**香农第一定理**，我们知道$H(p)$是传达某一信息最少需要的编码量，因此：

$$H(p,q) \geq H(p)$$

当且仅当$q(x) = p(x)$时，$H(p,q) = H(p)$。

结论：

上述公式解释了——为什么在机器学习中，当我们采用交叉熵损失函数时，我们的训练目标是降低 Cross Entropy 的大小，因为只有这样才能使我们模型的预测分布$q(x)$不断接近于目标分布（真实分布）$p(x)$。

三、KL 散度（Kullback-Leiber Divergence）

一般理解：KL Divergence 衡量了两个分布之间的差异程度。

公式：

$$D_{KL}(p||q) = -E_{x\sim p}\left[ln\frac{q(x)}{p(x)}\right] = -E_{x\sim p}[lnq(x) - lnp(x)]$$

而实际上，$KL\ Divergence = \text{Cross Entropy} - \text{Shannon Entropy}$，即：

$$D_{KL}(p||q) = -E_{x\sim p}[lnq(x) - lnp(x)] = H(p,q) - H(p)$$

因此，KL 散度表示了以当前的编码方式q最多还可以减少多少的编码量。此外，还需要特别注意，KL 散度不是距离，它不具有对称性，即：

$$D_{KL}(p||q) \neq D_{KL}(q||p)$$

思考：

为什么在机器学习中，我们不使用 KL 散度来作为损失函数？

实际上，最小化（Minimize）交叉熵损失函数和最小化 KL 散度是等价的，因为我们已知$D_{KL}(p||q) = H(p,q) - H(p)$，而在分类问题中，$H(p)$对于我们的学习模型（$model$）来说是不变量，因此它无需梯度下降。所以，仅计算$H(p,q)$的效果和计算$D_{KL}(p||q)$的效果是一样的。因此，采用交叉熵损失函数的效率更高。

总结：

- Shannon Entropy：$H(p) = E_{x\sim p}[-lnp(x)]$
  - 代表传达一个系统所需要的最小信息量。
- Cross Entropy：$H(p,q) = E_{x\sim p}[-lnq(x)]$
  - 代表用$q(x)$来编码所需要的信息量。
- KL Divergence：$D_{KL}(p||q) = H(p,q) - H(p)$
  - 代表以目前的编码方式$q(x)$用的信息量还有多少下降空间。

四、最大似然估计（MLE）

问题描述：

给定一组 Dataset D以及训练模型 model 的超参数m，现需要找到一组 model 的权重参数θ，使得 model 能以最大概率生成D的分布。

解决思路：

我们可以先随机设定一组权重参数θ，然后根据模型目前的m和θ来计算产生D分布的可能性，即$likelihood$，而$likelihood$越大则代表模型的输出越接近目标分布。然后，通过梯度下降，来不断更新优化θ的值，直到最优。

***MLE*公式化描述：**

$$\theta_{MLE} = argmax_{\theta} p(D|m,\theta)$$

其中，θ_{MLE}表示用$likelihood$方式求出的一组最优权重参数θ；$argmax_{\theta}(\cdot)$表示当θ取到某一值时，此时括号内函数的值取到最大；$p(D|m,\theta)$表示用m和θ生成目标分布D的概率。

现在，若假设D中有很多不同的数据集$D_1, D_2, \dots, D_N$，且从每个数据集中进行一次抽样，则上述公式可改写为：

$$\theta_{MLE} = argmax_{\theta} p_{D_1,D_2,\dots,D_N}(d_1, d_2, \dots, d_N|m,\theta)$$

若每次抽样都是独立的，则：

$$\theta_{MLE} = argmax_\theta \prod_i p_{D_i}(d_i|m,\theta)$$

若每次都是从同个分布进行抽样的，则：

$$\theta_{MLE} = argmax_\theta \prod_i p(d_i|m,\theta)$$

（以上就是基于独立同分布的假设所进行的推导。）

进一步，对上述公式取对数ln，可得：

$$\theta_{MLE} = argmax_\theta \sum_i lnp(d_i|m,\theta)$$

若所进行的训练是监督学习，而 data 的形式为(x,y)，则：

$$\theta_{MLE} = argmax_\theta \sum_i lnp(x_i,y_i|m,\theta)$$

若y的产生是依赖于x的，则：

$$\theta_{MLE} = argmax_\theta \sum_i \ln[p(y_i|x_i,m,\theta)p(x_i|m,\theta)]$$

又因为x和模型 model 是独立的，所以：

$$\theta_{MLE} = argmax_\theta \sum_i \ln[p(y_i|x_i,m,\theta)p(x_i)]$$

根据对数ln的性质——相乘变相加，可得：

$$\theta_{MLE} = argmax_\theta \sum_i \ln p(y_i|x_i,m,\theta) + lnp(x_i)$$

因为$p(x_i)$与优化θ无关，所以可以去掉多余项，得到：

$$\theta_{MLE} = argmax_\theta \sum_i \ln p(y_i|x_i,m,\theta)$$

将上述公式进一步改写：

$$\theta_{MLE} = argmin_\theta \frac{1}{N}\sum_i -\ln p(y_i|x_i,m,\theta)$$

接着，可进一步等价为：

$$\theta_{MLE} = argmin_\theta E_{x\sim p_{data}}[-\ln p_{model}(y_i|x_i,m,\theta)]$$

$$\theta_{MLE} = argmin_\theta H(p_{data},p_{model})$$

因此，求一组权重参数θ的问题实际上就转变为了求最小交叉熵$H(p_{data}, p_{model})$的问题。

思考：

MLE 有什么不足？缺陷？

在 MLE 中，有一个基本假设，即所有出现的θ概率都是均等的，而这会产生一些无法避免的问题，举例如下：

假设有以下两组θ都可以得到相同的$likelihood$，那么你觉得哪一组更好？

$$(1)\ \theta_1 = 0.5, \quad \theta_2 = 0.1, \quad \theta_3 = -0.1$$

$$(2)\ \theta_1 = 1000.0, \quad \theta_2 = 12.5, \quad \theta_3 = -500.0$$

而根据经验，通常我们认为参数组(1)所构成的模型更稳定，较不易$overfit$。但是 MLE 本身无法区别此类情况。所以这就是 MLE 本身所存在的缺陷。

五、最大后验估计（MAP）

必要知识补充 2：

贝叶斯定理：

$$\because p(D, m, \theta) = p(D, m|\theta)p(\theta) = p(\theta|D, m)p(D, m)$$

$$\therefore p(\theta|D, m) = \frac{p(D, m|\theta)p(\theta)}{p(D, m)} = \frac{p(D|m, \theta)p(m|\theta)p(\theta)}{p(D|m)p(m)}$$

$$= \frac{p(D|m, \theta)p(m, \theta)}{p(D|m)p(m)} = \frac{p(D|m, \theta)}{p(D|m)}\frac{p(m, \theta)}{p(m)} = \frac{p(D|m, \theta)p(\theta|m)}{p(D|m)}$$

所以可得：

$$p(\theta|D, m) = \frac{p(D|m, \theta)p(\theta|m)}{p(D|m)}$$

其中，$p(D, m|\theta)$同 MLE 中的$likelihood$；$p(\theta|m)$代表先验概率（Prior Probability），由人为给定，比如正态分布（normal distribution）；$p(D|m)$代表资料概率，通常与 m 无关；$p(\theta|D, m)$代表后验概率（Posterior Probability），即给定D, m后出现θ的概率。

问题描述：同 MLE 中的问题描述。

解决思路：

采用贝叶斯定理来处理上述问题。再简单来说，相比 MLE，多了一个先验概率的条件。

***MAP*公式化描述：**

$$\theta_{MAP} = argmax_\theta \frac{p(D|m,\theta)p(\theta|m)}{p(D|m)}$$

若采样基于独立同分布假设，则：

$$\theta_{MAP} = argmax_\theta \left[\prod_i \frac{p(d_i|m,\theta)}{p(d_i|m)}\right] p(\theta|m)$$

进一步，对上述公式取对数ln，可得：

$$\theta_{MAP} = argmax_\theta \left[\sum_i ln\frac{p(d_i|m,\theta)}{p(d_i|m)}\right] + lnp(\theta|m)$$

$$\theta_{MAP} = argmax_\theta \left\{\sum_i [lnp(d_i|m,\theta) - lnp(d_i|m)]\right\} + lnp(\theta|m)$$

又因为d和模型 model 是独立的，所以：

$$\theta_{MAP} = argmax_\theta \left\{\sum_i lnp(d_i|m,\theta)\right\} + lnp(\theta|m)$$

然后，将d替换为(x,y)，并把x移到后面，则：

$$\theta_{MAP} = argmax_\theta \left\{\sum_i lnp(y_i|x_i,m,\theta)p(x_i|m,\theta)\right\} + lnp(\theta|m)$$

$$\theta_{MAP} = argmax_\theta \left\{\sum_i [lnp(y_i|x_i,m,\theta) + lnp(x_i|m,\theta)]\right\} + lnp(\theta|m)$$

进一步，因为x和模型 model 是独立的，所以：

$$\theta_{MAP} = argmax_\theta \left\{\sum_i lnp(y_i|x_i,m,\theta)\right\} + lnp(\theta|m)$$

因此，MAP 相当于在优化 MLE，而添加的优化项为$lnp(\theta|m)$。

思考 1：

若$p(\theta|m)$为均匀分布（uniform distribution），则$lnp(\theta|m)$为一常数。此时，$\theta_{MAP} = \theta_{MLE}$。因此，在贝叶斯定理的观点下，MLE 仅为当没有先验假设时的 MAP 特例。

思考 2：

由于 MLE 本身存在缺陷，而为了解决缺陷，我们希望θ的大小总是接近于 0，因此，我们希望$p(\theta|m)$的分布平均值接近于 0 且方差有限。所以，当我们假设

$p(\theta|m)$为正态分布（normal distribution）时，即：

$$p(\theta|m) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{\theta^2}{2\sigma^2}\right\} \rightarrow lnp(\theta|m) = \ln\left(\frac{1}{\sqrt{2\pi}\sigma}\right) - \frac{\theta^2}{2\sigma^2}$$

于是，可得：

$$\theta_{MAP} = argmin_\theta \sum_i -lnp(y_i|x_i, m, \theta) + \frac{1}{2\sigma^2}\theta^2$$

最终，我们就导出了 **L2 正则化**！因此，在机器学习中，我们在损失函数中使用的 **L2 正则化**其实就隐含着希望参数呈现正态分布的假设！

思考 3：

若我们假设$p(\theta|m)$为均值为 0 的拉普拉斯分布（Laplace Distribution）时，即：

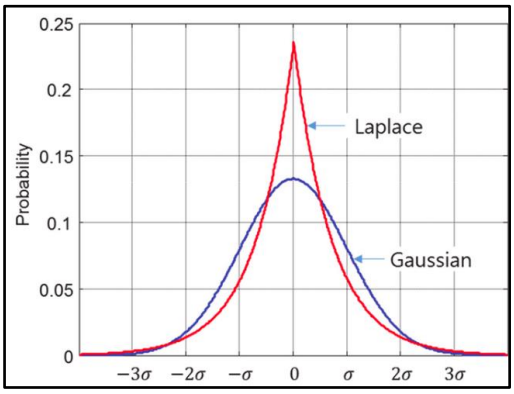
$$p(\theta|m) = \frac{1}{2b}\exp\left\{-\frac{|\theta|}{b}\right\} \rightarrow lnp(\theta|m) = \ln\left(\frac{1}{2b}\right) - \frac{|\theta|}{b}$$

于是，可得：

$$\theta_{MAP} = argmin_\theta \sum_i -lnp(y_i|x_i, m, \theta) + \frac{1}{b}|\theta|$$

最终，我们就导出了 **L1 正则化**！因此，在机器学习中，我们在损失函数中使用的 **L1 正则化**其实就隐含着希望参数呈现拉普拉斯分布的假设！

补充说明：



上图展示的是正态分布和拉普拉斯分布。

通常，在 L2 正则化下，因为是对权重参数的平方做惩罚，所以权重参数大小往往会比较均匀，而在 L1 正则化下，参数分布会比较稀疏。而这背后的实际原因来自于 MAP 中对于权重参数的 Normal & Laplace Distribution 假設。

相关参考资料：

1. https://www.bilibili.com/video/BV1hr4y1X7q5/?spm_id_from=333.999.0.0&vd_source=36e948dc2acdb36d7055f879d377b529
2. https://www.bilibili.com/video/BV1zj41127uG/?spm_id_from=333.999.0.0&vd_source=36e948dc2acdb36d7055f879d377b529