信息量、熵、交叉熵、KL散度等

在机器学习领域，我们总是能听到到“信息熵”、“交叉熵”等概念。这里我们就对这些概念进行一个总结介绍，并知道它们是如此简单。确保当再次遇到这些它们时，能够不被这些概念所吓倒。

信息量

信息量是对“一个事件是否发生”的不确定程度的度量。对于一个事件 x ，其发生的概率越小，则其信息量越大。如果我们使用 $I(x) = f(P(x))$ 来表示事件 x 的信息量(其中， $P(x)$ 表示事件 x 发生的概率)，那么我们希望它具有如下的性质：

- $I(x) = f(P(x))$ 是关于 $P(x)$ 的单调递减函数：事件发生的概率越高，产生的信息量越小；事件发生的概率越低，产生的信息量越大。
- $I(x) = f(P(x)) >= 0$ ： 信息量是非负的；
- $I(x) = f(P(x))$ 对与 $P(x)$ 是连续的；
- $I(x_1, x_2) = I(x_1) + I(x_2)$ ： 独立事件带来的总信息量，等于各事件信息量之和。

信息论中严格的推导证明： $I(x) = -log(P(x))$ 满足上述条件。因此，我们通常使用 $I(x) = -log(P(x))$ 来表示事件 x 的信息量。

信息熵

信息熵描述的是整个事件空间的平均信息量(期望)。

对于一个离散随机变量 x ，其信息熵定义为：

$$H(x) = -\sum_{x \in X} P(x)log(P(x))$$

其中， X 是随机变量 x 的取值空间， $P(x)$ 是随机变量 x 取值为 x 的概率。

对于一个连续随机变量 x ，其信息熵定义为：

$$H(x) = -\int P(x)log(P(x))dx$$

其中， $P(x)$ 是随机变量 x 的概率密度函数。

交叉熵 (Cross Entropy)

交叉熵是两个概率分布 P 和 Q 的差异的度量，其本质是按照概率分布 P 来计算的基于概率分布 Q 的信息量期望值。

对于离散随机变量，其交叉熵定义为：

$$H(P,Q) = -\sum_{x} P(x)log(Q(x))$$

对于连续随机变量，其交叉熵定义为：

$$H(P,Q) = -\int P(x)log(Q(x))dx$$

通常， P 表示真实分布， Q 表示模型预测的分布。因此，我们希望 Q 能够尽可能地接近 P ，从而使得交叉熵最小化。反之，如果针对某个事件 x 预测分布 $Q(x)$ 很小，则信息量很大，而真实分布 $P(x)$ 很大，则计算得到的交叉熵也会很大。这也是交叉熵**通常被用作损失函数**的原因。

KL 散度 (Kullback-Leibler Divergence)

从计算公式可以看出，即使 P 和 Q 完全相同，交叉熵也不会为 0。而且，交叉熵的计算中， $H(P)$ 是固定的，与模型无关。由于这些局限性，在聚焦 P 和 Q 之间的差异时，我们通常会使用 KL 散度。

对于一个离散随机变量，其 KL 散度定义为：

$$D_{KL}(P||Q) = \sum_{x} P(x)(log(P(x)) - log(Q(x))) = \sum_{x} P(x)log(\frac{P(x)}{Q(x)})$$

而对于连续随机变量，其 KL 散度定义为：

$$D_{KL}(P||Q) = \int P(x)(log(P(x)) - log(Q(x)))dx = \int P(x)log(\frac{P(x)}{Q(x)})dx$$

KL 散度不满足对称性($D_{KL}(P||Q) \neq D_{KL}(Q||P)$)和三角不等式。

KL 散度的应用领域有：

- 变分自编码器(VAE)：在VAE中，KL散度作为正则化器，确保潜在变量分布接近先验分布(通常是标准高斯分布)。
- 数据压缩：KL散度量化了使用一个概率分布压缩来自另一个分布的数据时的效率损失，这在设计和分析数据压缩算法时极为有用。

- 强化学习：在强化学习中，如近端策略优化(PPO)算法，KL散度用于控制新策略与旧策略之间的偏离程度。
- 数据漂移检测：在工业应用中，KL散度广泛用于检测数据分布随时间的变化。

JS散度 (Jensen-Shannon Divergence)

JS 散度是一种对称的散度度量，用于量化两个概率分布间的相似性。它基于 KL 散度构建，但克服了 KL 散度不对称的局限性。给定两个概率分布 P 和 Q ，JS 散度定义如下：

$$D_{JS}(P,Q) = \frac{1}{2}D_{KL}(P||(P+Q)/2) + \frac{1}{2}D_{KL}(Q||(P+Q)/2)$$

JS 散度解决了 KL 散度在分布比较中的不对称性问题。它不将 P 或 Q 视为"标准"分布，而是通过混合分布 $(P+Q)/2$ 来评估它们的综合行为。这使得 JS 散度在需要无偏比较分布的场景中特别有用。