

交叉熵 (CrossEntropy Loss) 损失函数

交叉熵是信息论中的一个重要概念，主要用于度量两个概率分布间的差异性。

信息量的大小与信息发生的概率成反比。

概率越大，信息量越小
概率越小，信息量越大

不太可能发生的事件具有较高的信息量

$$I(x) = -\log(P(x))$$

↑
信息量

一件事发生的概率

(信息)熵，用于表示对所有信息量的期望

试验中每次可能结果的概率乘以其结果的总和

离散型随机变量

$$H(X) = - \sum_{i=1}^n P(x_i) \log(P(x_i))$$

↑
一件事发生的概率

↑
信息量

相对熵 (KL 散度)

如果对于同一个随机变量 X 有两个单独的概率分布 $P(x)$ 和 $Q(x)$

可以使用 KL 散度来衡量两个概率分布之间的差异

$$D_{KL}(P||Q) = \sum_{i=1}^n P(x_i) \log\left(\frac{P(x_i)}{Q(x_i)}\right)$$

在 ML 中，常常使用 $P(x)$ 来表示样本的真实分布。 $Q(x)$ 来表示模型所预测的分布

e.g. 一个三分类任务中（猫狗与分类器）， x_1, x_2, x_3 分别代表猫，狗，鸟，一张照片的真实分布 $P(x) = [1, 0, 0]$
预测分布 $Q(x) = [0.7, 0.2, 0.1]$

KL 散度：

$$D_{KL}(P||Q) = \sum_{i=1}^n P(x_i) \log\left(\frac{P(x_i)}{Q(x_i)}\right)$$

$$= P(x_1) \log\left(\frac{P(x_1)}{Q(x_1)}\right) + P(x_2) \log\left(\frac{P(x_2)}{Q(x_2)}\right) + P(x_3) \log\left(\frac{P(x_3)}{Q(x_3)}\right)$$

$$= 1 \cdot \log\left(\frac{1}{0.7}\right) + 1 \cdot \log\left(\frac{0}{0.2}\right) + 1 \cdot \log\left(\frac{0}{0.1}\right)$$

$$= 1 \cdot \log\left(\frac{1}{0.7}\right) = 0.36$$

KL 散度越小，表示 $P(x)$ 与 $Q(x)$ 分布更加接近

可以通过反复训练 $Q(x)$ 来使 $Q(x)$ 的分布逼近 $P(x)$

交叉熵

首先将 KL 散度公式拆开：

$$D_{KL}(P||Q) = \sum_{i=1}^n P(x_i) \log\left(\frac{P(x_i)}{Q(x_i)}\right)$$

$$= \sum_{i=1}^n P(x_i) \log(P(x_i)) - \sum_{i=1}^n P(x_i) \log(Q(x_i))$$

$$= \underbrace{-H(x)}_{\text{信息熵}} + \underbrace{\left[-\sum_{i=1}^n P(x_i) \log(Q(x_i)) \right]}_{\text{交叉熵}}$$

KL 散度 = 交叉熵 - 信息熵

\Rightarrow 交叉熵 = KL 散度 + 信息熵

交叉熵公式：

$$H(p, q_e) = - \sum_{i=1}^n p(x_i) \log(q_e(x_i))$$

在机器学习训练网络时，输入数据与标签常常已经确定，那么真实概率分布 $P(x)$ 也确定下来了，所以信息熵是一个常量。由于 KL 散度的值表示真实概率分布 $P(x)$ 与预测概率分布 $Q(x)$ 之间存在差异，值越小表示预测的结果越好，所以需要最小化 KL 散度。

而交叉熵 = KL 散度 + 常量（信息熵），且公式相比于 KL 散度更加容易计算，所以在 ML 中常常使用 **交叉熵损失函数** 来计算 loss 就行。

交叉熵在单分类问题中的应用

在线性回归问题中，常常使用 MSE (Mean Squared Error)

作为 loss 函数，而在分类问题中常常使用 **交叉熵** 作为 loss 函数

e.g.

	猫	狗	马
label	0	1	0
Pred	0.2	0.7	0.1

$$\text{loss} = - \sum_{i=1}^n p(x_i) \log(q_e(x_i))$$

$$= - [p(x_1) \log(q_e(x_1)) + p(x_2) \log(q_e(x_2)) + p(x_3) \log(q_e(x_3))]$$

$$= - [0 \cdot \log(0.2) + 1 \cdot \log(0.7) + 0 \cdot \log(0.1)]$$

$$= - \log(0.7) \approx 0.36$$

一个 batch 的 loss 为

$$\text{loss} = -\frac{1}{m} \sum_{i=1}^m \sum_{j=1}^n p(x_{ij}) \log(g_e(x_{ij}))$$

其中 m 表示样本个数

交叉熵损失函数在多标签分类任务中的应用

多标签分类任务：即一个样本可以有多个标签，e.g. 一张图片中可能同时含有“猫”和“狗” \Rightarrow 这张照片便同时拥有属于“猫”和“狗”的两种标签。

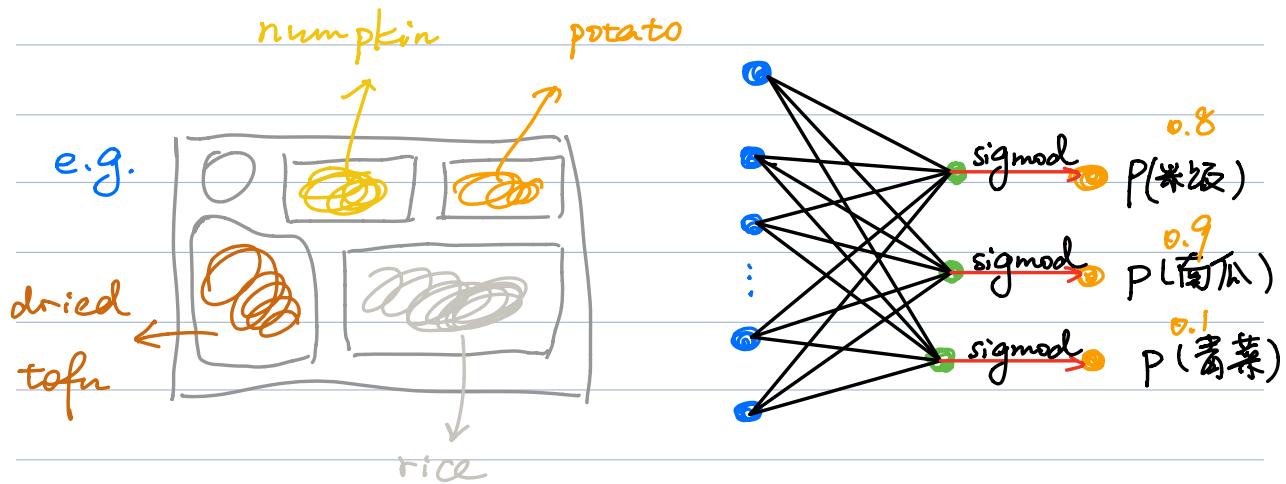
在这种情况下，将 sigmoid 函数作为网络最后一层输出，把网络最后一层的每个神经元看做任务中的一个类别。以图像识别为例，网络最后一层的输出应该理解为：网络认为图片中含有这一类别物体的概率。而每一类的真实标签只有两种可能值：“图片中含有这一类物体”和“图片中不含有这一类物体”

这是一个二项分布

综上所述，对多分类任务中的每一类单独分析的话，真实分布 P 是一个二项分布，可能的取值为 0 或 1。而网络预测的分布 Q 可以理解为标签为 1 的概率。此外，由于多标签分类任务中，每一类是相互独立的，所以网络最后一层神经元输出的概率值之和并不一定等于 1。交叉熵损失函数为：

$$\text{Loss} = -y \log \hat{y} - (1-y) \log(1-\hat{y})$$

总的交叉熵为多标签分类任务中每一类的交叉熵之和。



$$\text{loss}(\text{rice}) = -1 \times \log 0.8 - (1-1) \log (1-0.8) = -\log 0.8 \approx 0.2231$$

$$\text{Loss}(\text{pumpkin}) = -1 \times \log 0.9 - (1-1) \log (1-0.9) = -\log 0.9 \approx 0.1054$$

$$\text{loss}(\text{green}) = -0 \times \log 0.1 - (1-0) \log (1-0.1) = 0.1054$$

$$\begin{aligned}\text{Loss}(\text{all}) &= \text{Loss}(\text{rice}) + \text{Loss}(\text{pumpkin}) + \text{loss}(\text{green}) \\ &= 0.2231 + 0.1054 + 0.1054 = 0.4339\end{aligned}$$

总结：

- 交叉熵能够衡量同一个随机变量中的两个不同概率分布的差异程度。交叉熵的值越小，模型预测效果越好

- 交叉熵在分类问题中常与 softmax 是标配。

softmax 将输出的结果进行处理，使其多个分类的预测值和为 1，再通过交叉熵来计算损失