


北大旁听 - 深入Loss Function的来源


宇航员 发布于 2019-03-14

1. 想法

由于有朋友在北大，很高兴能蹭到深度学习的课程，李戈教授的课程十分精彩，比起只会念PPT的老师，他的教学就像在堆积知识的金字塔。

2. Loss Function

2.1 经典统计 vs 深度学习 vs 贝叶斯统计

概率论分为两大学派，贝叶斯学派认为先验知识很重要，而经典统计学派就是纯粹的看统计信息。

现在的深度学习最大的优点就是在**数据拟合**上表现非常好，但最大的缺点就是它的**不可解释性**。

在一篇论文：[Deep Learning: A Bayesian Perspective](#) 中提到，目前深度学习算法取得好效果的主要原因归功于**ReLU**、**learning_rate**、**Dropout**。

实际上先验知识只是以网络的模型结构的方式呈现的（包括Loss Function的设计等）。

2.2 最大似然估计

其实目前大部分使用的损失函数都是以**最大似然原理**为核心而设计的。

深度学习的核心问题就是**让网络产生的数据分布尽可能贴近样本分布**，所以极大似然原理就很自然的用在了深度学习上。

- Consider: a set of m examples $\mathbb{X} = \{[x^{(1)}, y^{(1)}], \dots, [x^{(m)}, y^{(m)}]\}$, drawn independently from the true but unknown data generating distribution $p_{data}(x, y)$.
- Let $p_{model}(x, \theta)$ be a parametric family of probability distributions over the same space indexed by θ .
- Then, the maximum likelihood estimator for θ is defined as:

样本也有数据分布

$$\theta_{ML} = \operatorname{argmax}_{\theta} p_{model}(\mathbb{X}; \theta) \tag{1}$$

$$= \operatorname{argmax}_{\theta} \prod_{i=1}^m p_{model}(x^{(i)}; \theta) \tag{2}$$

贴近样本分布

而要评判分布的“差别”，首先需要可以评判分布的指标，而这个指标就是香农的**信息熵**。

- 信息熵，表示一个信源发出的信号的不确定程度。在信源发出的信号中，某信号出现的概率越大，熵越小；反之越大。
- 因此，信息熵的估算需要满足两个条件：
 - ① 单调递减性，信息熵的值是信号 i 出现概率 p_i 的单调递减函数
 - ② 可加性，两个独立符号所对应的不确定程度应等于各自不确定程度之和
- Shannon 用 \log 函数定义信号 i （样本 i ）的信息熵：

$$f(p(i)) = \log \frac{1}{p(i)} = -\log p(i)$$

- 则，包含 n 个样本的样本集合的信息熵定义为：

$$\text{Entropy} \quad E(P) = \sum_i^n p(i) \log \frac{1}{p(i)}$$

注意：这里的 $p(i)$ 表示样本的真实分布；

有了评价指标后，我们还不急着对比，因为要计算信息熵，需要知道样本的真实分布和概率密度。在计算模型分布的信息熵时，此时就不叫信息熵了，而称为**交叉熵**，这也就是所谓的**cross-entropy**（而不是大家常见的 $1-\log(x)$ ）。

Many authors use the term "cross-entropy" to identify specifically the negative log-likelihood of a Bernoulli or softmax distribution, but that is a misnomer.

- 然而，在机器学习中，我们通常使用模型分布 $q(i)$ 来逼近真实分布，这时，样本集合的信息熵为（又称**交叉熵**）：

$$E(P, Q) = \sum_i^n p(i) \log \frac{1}{q(i)} = - \sum_i^n p(i) \log q(i)$$

这才叫交叉熵
↓
↓
真实分布 模型分布

根据Gibbs不等式，有： $E(P, Q) \geq E(P)$ ，其实很好解释，用模拟出来的概率密度去计算真实分布的信息熵，肯定是**比较混乱**的（相对于真实概率密度计算真实分布的信息熵）。

有了交叉熵和原分布的信息熵后，我们做差，就能得到**相对熵**（又称**KL散度**）。

- 我们将 $E(P, Q)$ 与 $E(P)$ 的差，定义为“使用模型分布 Q 来逼近真实分布 P 时的**相对熵**”（又称 **KL 散度**）：

$$\text{做差} \quad D(P \parallel Q) = E(P, Q) - E(P) = \sum P(i) \log \frac{P(i)}{Q(i)}$$

→ KL散度

- **KL 散度**，表示 2 个概率分布的差异程度；

终于，我们得到了相对熵，可以评判分布的“差别”后，我们就可以用一个**视角**来看Loss Function：

- Loss Function 用于计算模型输出数据与样本数据之间的“差别”。
- Loss Function 体现了人们对这种“差别”进行度量时所依赖的先验知识。
- 几种常见的“差别”度量方法
 - 个体模型结果与样本数据之间的“距离”。
 - 多个模型结果与样本数据之间存在数据分布上的差别。
 - 通过分析“**统计距离**”（**Statistical Distance**）来设计Loss。

好了，让我们回到最大似然原理上，为了让模型分布尽可能贴近样本分布，那么我们要解决的问题就是最小化KL散度

- One way to interpret ~~maximum~~ likelihood estimation is to view it as minimizing the dissimilarity between the empirical distribution p_{data} defined by the training set and the model distribution.
- The degree of dissimilarity between the two measured by the KL divergence:

$$D_{ML}(p_{data} || p_{model}) = \overbrace{\mathbb{E}_{x \sim p_{data}} [\log p_{data}(x) - \log p_{model}(x)]}^{KL \text{ 散度}} \quad (5)$$

训练数据分布
↪ p(x)
(已确定)
(只能更改5)

- The term on the left is a function only of the training data, not the model. (not include θ) This means when we try to minimize the KL divergence, we need only minimize.

$$-\mathbb{E}_{x \sim p_{data}} [\log p_{model}(x)] \quad (6)$$

由于真实样本分布是已经确定的值，而我们能改变的就是 $p_{model}(x)$ 中的权值（也就是神经网络的weights），所以我们将问题简化为最小化(6)，简写为：

$$\theta_{ML} = \operatorname{argmax}_{\theta} \mathbb{E}_{x \sim p_{data}} \log p_{model}(x; \theta) \quad (4)$$

2.3 基于上述原则推导MSE的合理性

MSE损失大家一定都知道，但是它是怎么来的呢？为什么要以这个形式出现呢？其实它是有一个非常严格的使用条件的：

由于模型的输出和真实数据的分布一定会有偏差，那么只有假设当这个偏差符合正态分布时，才可以推导出MSE的形式，具体如下：

假设目标值与输入变量之间存在如下关系：

$$y_i = h_{\theta}(x_i; \theta) + \epsilon_i \rightarrow \text{偏差}$$

可合理假设： $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ ，则其概率密度函数为：

偏差满足正态分布

$$p(\epsilon_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\epsilon_i^2}{2\sigma^2}\right)$$

代入上式，并由误差的定义，可以推知：

$$p(y_i | x_i; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - h_{\theta}(x_i))^2}{2\sigma^2}\right)$$

↪ 用最大似然 的 loss

得出了概率密度分布函数后，将其带入最大似然原理中，再取对数，就可以得到MSE的标准形式了：

对上式进行最大化求取，等价于：

$$\begin{aligned}\log(L(\theta)) &= \log \prod_{i=1}^m p(y_i|x_i;\theta) \\ &\stackrel{\text{对数}}{=} \sum_{i=1}^m \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - h_{\theta}(x_i))^2}{2\sigma^2}\right) \\ &= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^m (y_i - h_{\theta}(x_i))^2\end{aligned}$$

要通过调整 θ ，使上式最大化，只需要考虑使最后的二次项最小化即可，即最小化：

$$\operatorname{argmin}_{\theta} \frac{1}{2} \sum_{i=1}^m (y_i - h_{\theta}(x_i))^2 \quad \text{即：} \quad \operatorname{argmin}_{\theta} \frac{1}{2} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2$$

那么这里又出现了一个问题，为什么要让偏差符合正态分布呢？

其实这是由以下两条理论得出的：

- 同分布中心极限定理：n个独立同分布的随机变量之和，服从正态分布。
- 非同分布的李雅普诺夫定理：大量随机因素叠加的结果，近似服从正态分布。

有了这已经证明的两条理论，才可以基于正态分布，得出MSE的标准形式。

同理，我们可以很容易的得到交叉熵（这里不是真的交叉熵，只是大家都习惯这么叫它了）、softmax的一般形式的证明：

基于伯努利分布的 Loss Function

先验知识

若可假设网络输出满足如下分布：

$$\begin{aligned}p(y = 1|x; \theta) &= h_{\theta}(x) \\p(y = 0|x; \theta) &= 1 - h_{\theta}(x)\end{aligned}$$

上式可写为：

$$p(y|x; \theta) = (h_{\theta}(x))^y (1 - h_{\theta}(x))^{1-y}$$

则，似然函数为：

$$\begin{aligned}L(\theta) &= p(Y|X; \theta) \\&= \prod_{i=1}^m p(y^{(i)}|x^{(i)}; \theta) \\&= \prod_{i=1}^m \left((h_{\theta}(x))^{y^{(i)}} (1 - h_{\theta}(x))^{1-y^{(i)}} \right)\end{aligned}$$

进行 log 处理，得到需要最大化的 Loss Function 为：

交叉熵

$$\begin{aligned}l(\theta) &= \log L(\theta) \\&= \sum_{i=1}^m \left(y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log (1 - h(x^{(i)})) \right)\end{aligned}$$

Logistic Regression

< > < > < > < > < >

基于多项分布的 Loss Function

若可假设网络输出满足如下分布：

$$\begin{bmatrix} p(y^{(i)} = 1|x^{(i)}; \theta) \\ p(y^{(i)} = 2|x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k|x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^k f(x^{(i)}; \theta_j)} \begin{bmatrix} f(x^{(i)}; \theta_1) \\ f(x^{(i)}; \theta_2) \\ \vdots \\ f(x^{(i)}; \theta_k) \end{bmatrix}$$

定义函数：

$$1\{\text{表达式为真}\} = 1$$

则，似然函数为：

$$\begin{aligned} L(\theta) &= p(Y|X; \theta) \\ &= \prod_{i=1}^m p(y^{(i)}|x^{(i)}; \theta) \\ &= \prod_{i=1}^m \left(\sum_{j=1}^k 1\{y^{(i)} = j\} \frac{f(x^{(i)}; \theta_j)}{\sum_{j=1}^k f(x^{(i)}; \theta_j)} \right) \end{aligned}$$

进行 log 处理，得到需要最大化的 Loss Function 为：

$$\begin{aligned} l(\theta) &= \log L(\theta) \\ &= \sum_{i=1}^m \left(\sum_{j=1}^k (1\{y^{(i)} = j\}) \log \frac{f(x^{(i)}; \theta_j)}{\sum_{j=1}^k f(x^{(i)}; \theta_j)} \right) \end{aligned}$$

等同于最小化：
+m-1次方。
✓

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left(\sum_{j=1}^k (1\{y^{(i)} = j\}) \log \frac{f(x^{(i)}; \theta_j)}{\sum_{j=1}^k f(x^{(i)}; \theta_j)} \right)$$

为便于计算，通常取 $f(x^{(i)}; \theta_j) = \exp(\theta_j^T x^{(i)})$ ，得：
取exp函数计算会变得更简单

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left(\sum_{j=1}^k (1\{y^{(i)} = j\}) \log \frac{\exp(\theta_j^T x^{(i)})}{\sum_{j=1}^k \exp(\theta_j^T x^{(i)})} \right)$$

↓
分类的时候
↓
softmax

SoftMax

2.4 总结

终于，我们说完了现代常用的损失函数是怎么得到的了。来一个简单的总结：

一切的起源都是最大似然原理，为了衡量模型分布和真实分布的差异，我们从信息熵中得到了KL散度。于是我们将基于正态分布的偏差假设（MSE）、伯努利分布、多项分布代入最大似然原理，得到了我们现在常见的Loss Function（MSE、CrossEntropyLoss、Softmax）。

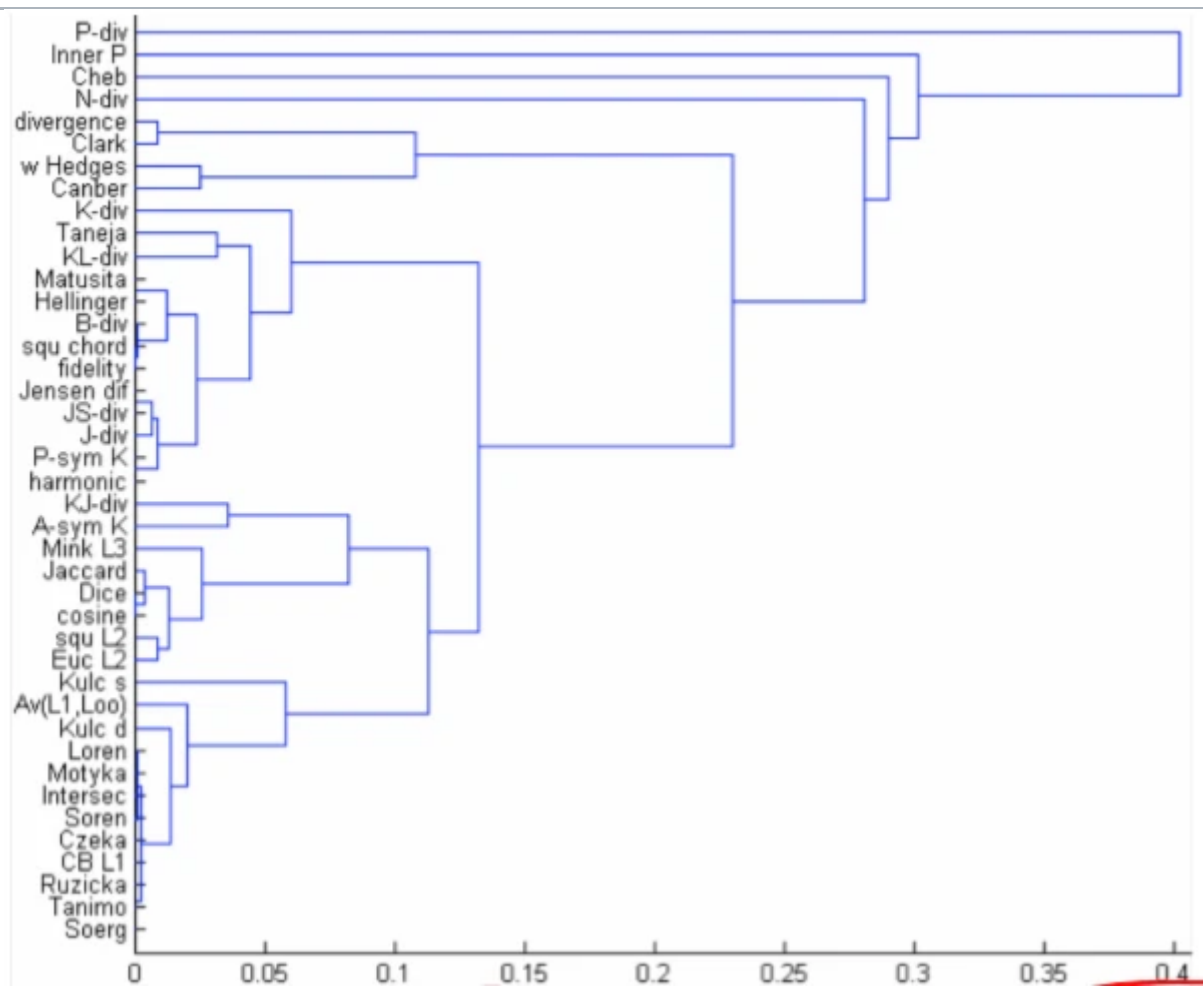
所以这里有很重要的一点，不要盲目地使用交叉熵！你要注意的是你的数据分布，如果它不符合正态分布假设，那么你很可能需要重新设计Loss Function了，那我们该如何做呢？

其实KL散度有它的缺点，比如它不符合距离的定义（不对称）

- KL 散度并不满足距离的概念，因为：

- ① KL 散度不是对称的 ($D(P \parallel Q) \neq D(Q \parallel P)$)
- ② KL 散度不满足三角不等式

于是乎，你需要自己寻找衡量数据分布的散度（Divergence）



[Comprehensive Survey on Distance/Similarity Measures Between Probability Density Functions, 2007](#)

然后根据散度重新你的设计Loss Function。

至此，关于Loss Function的内容就告一段落了，其实讲到最后，以我的数学水平实在达不到这个境界，所以到后面的概念如果讲的不太清楚，请各位读者谅解了。

[深度学习](#)

阅读 1.9k • 更新于 2019-03-14



宇航员

转了开发的计算机视觉研究生。

111 声望 **3** 粉丝

关注作者