分类问题

CiMorn

October 30, 2025

1 介绍

参考:https://zh-v2.d2l.ai/

分类问题：不是问“多少”，而是问“哪一个”：

- 某个电子邮件是否属于垃圾邮件文件夹?
- 某个用户可能注册或不注册订阅服务?
- 某个图像描绘的是驴、狗、猫、还是鸡?
- 某人接下来最有可能看哪部电影?

通常，机器学习实践者用分类这个词来描述两个有微妙差别的问题：

- 我们只对样本的“硬性”类别感兴趣，即属于哪个类别；
- 我们希望得到“软性”类别，即得到属于每个类别的概率。

这两者的界限往往很模糊。其中的一个原因是：即使我们只关心硬类别，我们仍然使用软类别的模型。(用概率描述离散类别)

我们从一个图像分类问题开始。假设每次输入是一个 2×2 的灰度图像。

我们可以用一个标量表示每个像素值，每个图像对应四个特征 x_1, x_2, x_3, x_4。

此外，假设每个图像属于类别“猫”“鸡”和“狗”中的一个。

接下来，我们要选择如何表示标签。我们有两个明显的选择：最直接的想法是选择 $y \in \{1, 2, 3\}$，其中整数分别代表 $\{狗, 猫, 鸡\}$。这是在计算机上存储此类信息的有效方法。

一种表示分类数据的简单方法：独热编码（one-hot encoding）。

独热编码是一个向量，它的分量和类别一样多。

类别对应的分量设置为 1，其他所有分量设置为 0。

在我们的例子中，标签 y 将是一个三维向量，其中 $(1, 0, 0)$ 对应于“猫”、$(0, 1, 0)$ 对应于“鸡”、$(0, 0, 1)$ 对应于“狗”：

$$y \in \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}.$$

现在我们将优化参数以最大化观测数据的概率

为了得到预测结果，我们将设置一个阈值，如选择具有最大概率的标签。

我们希望模型的输出 $\hat{y}_j$ 可以视为属于类 j 的概率，然后选择具有最大输出值的类别 $\operatorname{argmax}_j y_j$

作为我们的预测。

例如，如果 $\hat{y}_1$、$\hat{y}_2$ 和 $\hat{y}_3$ 分别为 0.1、0.1 和 0.8，那么我们预测的类别是 3，在我们的例子中代表“狗”。

下面我们为每个输入计算三个未规范化的预测（logit）：o_1、o_2 和 o_3。

$$
\begin{aligned}
o_1 &= x_1 w_{11} + x_2 w_{12} + x_3 w_{13} + x_4 w_{14} + b_1,\\
o_2 &= x_1 w_{21} + x_2 w_{22} + x_3 w_{23} + x_4 w_{24} + b_2,\\
o_3 &= x_1 w_{31} + x_2 w_{32} + x_3 w_{33} + x_4 w_{34} + b_3.
\end{aligned}
$$

然而我们能否将未规范化的预测 o 直接视作我们感兴趣的输出呢?

答案是否定的。

因为将线性层的输出直接视为概率时存在一些问题:

一方面，我们没有限制这些输出数字的总和为 1。另一方面，根据输入的不同，它们可以为负值。

这些违反了中所说的概率基本公理。

要将输出视为概率，我们必须保证在任何数据上的输出都是非负的且总和为 1。此外，我们需要一个训练的目标函数，来激励模型精准地估计概率。

例如，在分类器输出 0.5 的所有样本中，我们希望这些样本是刚好有一半实际上属于预测的类别。

这个属性叫做校准（calibration）。

2 softmax

softmax 函数能够将未规范化的预测变换为非负数并且总和为 1，同时让模型保持可导的性质。

为了完成这一目标，我们首先对每个未规范化的预测求幂，这样可以确保输出非负。

为了确保最终输出的概率值总和为 1，我们再让每个求幂后的结果除以它们的总和。如下式：

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{o}) \quad \text{其中} \quad \hat{y}_j = \frac{\exp(o_j)}{\sum_k \exp(o_k)}$$

这里，对于所有的 j 总有 $0 \leq \hat{y}_j \leq 1$。

因此，$\hat{\mathbf{y}}$ 可以视为一个正确的概率分布。

softmax 运算不会改变未规范化的预测 $\mathbf{o}$ 之间的大小次序，只会确定分配给每个类别的概率。

因此，在预测过程中，我们仍然可以用下式来选择最有可能的类别。

$$\operatorname{argmax}_j \hat{y}_j = \operatorname{argmax}_j o_j.$$

尽管 softmax 是一个非线性函数，但 softmax 回归的输出仍然由输入特征的仿射变换决定。

因此，softmax 回归是一个线性模型（linear model）。

3 损失函数

对于任何标签 $\mathbf{y}$ 和模型预测 $\hat{\mathbf{y}}$，损失函数为：

$$l(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{j=1}^{q} y_j \log \hat{y}_j.$$

损失函数被称为交叉熵损失（cross-entropy loss）。

由于 $\mathbf{y}$ 是一个长度为 q 的独热编码向量，所以除了一个项以外的所有项 j 都消失了。

由于所有 $\hat{y}_j$ 都是预测的概率，所以它们的对数永远不会大于 0。

因此，如果正确地预测实际标签，即如果实际标签 $P(\mathbf{y} \mid \mathbf{x}) = 1$，则损失函数不能进一步最小化。

注意，这往往是不可能的。

例如，数据集中可能存在标签噪声（比如某些样本可能被误标），或输入特征没有足够的信息来完美地对每一个样本分类。

4 softmax 及其导数

由于 softmax 和相关的损失函数很常见，因此我们需要更好地理解它的计算方式。

利用 softmax 的定义，我们得到：

$$
\begin{aligned}
l(\mathbf{y}, \hat{\mathbf{y}}) &= -\sum_{j=1}^{q} y_j \log \frac{\exp(o_j)}{\sum_{k=1}^{q} \exp(o_k)} \\
&= \sum_{j=1}^{q} y_j \log \sum_{k=1}^{q} \exp(o_k) - \sum_{j=1}^{q} y_j o_j \\
&= \log \sum_{k=1}^{q} \exp(o_k) - \sum_{j=1}^{q} y_j o_j.
\end{aligned}
$$

考虑相对于任何未规范化的预测 o_j 的导数，我们得到：

$$
\partial_{o_j} l(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\exp(o_j)}{\sum_{k=1}^{q} \exp(o_k)} - y_j = \mathrm{softmax}(\mathbf{o})_j - y_j.
$$

换句话说，导数是我们 softmax 模型分配的概率与实际发生的情况（由独热标签向量表示）之间的差异。

从这个意义上讲，这与我们在回归中看到的非常相似，其中梯度是观测值 y 和估计值 $\hat{y}$ 之间的差异。