交叉熵损失

概念定义

交叉熵(Cross-Entropy)是机器学习中常用的损失函数，特别是在分类问题中。它衡量的是两个概率分布之间的差异。

计算方法

对单个样本，假设我们有真实的标签分布 y 和预测的概率分布 $\hat{y}$ ，交叉熵的公式如下：

$$H(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

其中， C 是总类别数， y_i 是真实标签分布(通常是 one-hot 编码)，即对于正确类别 i，其值为 1；对于其他类别为 0。 $\hat{y}_i$ 是预测概率中第 i 个类别的概率。

对批量样本 (Batch size = N)，有计算公式：

$$H(Y, \hat{Y}) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{C} y_{n,i} \log(\hat{y}_{n,i})$$

其中， Y 和 $\hat{Y}$ 分别是真实标签分布和预测概率分布的矩阵。

实现代码

```python
import numpy as np

def cross_entropy_loss(y_true, y_pred, epsilon=1e-12):
    """
    计算交叉熵损失。
    参数:
        y_true: 真实标签（one-hot 编码），形状为 (N, C)。
        y_pred: 预测的概率分布，形状为 (N, C)。
        epsilon: 裁剪系数，默认为 1e-12。
    返回:
        平均交叉熵损失。
    """
    # 作用：对y_pred数组中的每个元素进行裁剪，确保它们的值位于epsilon和1 - epsilon之间
    # 目的：防止 log(0) 导致数值不稳定
    y_pred = np.clip(y_pred, epsilon, 1. - epsilon)
    # 计算交叉熵损失
    loss = -np.sum(y_true * np.log(y_pred)) / y_true.shape[0]
    return loss

# 示例数据
y_true = np.array([[1, 0, 0], [0, 1, 0], [0, 0, 1]])                    # 真实标签(one-h
y_pred = np.array([[0.7, 0.2, 0.1], [0.1, 0.8, 0.1], [0.2, 0.3, 0.5]])  # 模型预测的概率

# 计算交叉熵损失
loss = cross_entropy_loss(y_true, y_pred)
print(f"交叉熵损失: {loss}")
```

PyTorch中使用

在深度学习框架 PyTorch 中，可以直接调用内置的交叉熵损失函数 torch.nn.CrossEntropyLoss。需要注意的是，PyTorch 的 CrossEntropyLoss 内部已经包含了 softmax 操作，因此输入应该是未经 softmax 的 logits。

```python
import torch
import torch.nn as nn

# 实例化交叉熵损失函数
criterion = nn.CrossEntropyLoss()

# 示例数据
y_true = torch.tensor([0, 1, 2])                                  # 真实标签（非 one-hot 编码）
y_pred = torch.tensor([[2.0, 1.0, 0.1], [0.5, 2.0, 0.3], [0.2, 0.3, 3.0]])  # 模型输出的

# 计算交叉熵损失
loss = criterion(y_pred, y_true)
print(f"交叉熵损失: {loss.item()}")
```