

机器学习常用的损失函数

理解和实现常见的损失函数
研究深度学习中的关键模块

1. 均方误差 (Mean Squared Error, MSE)

基本理论

均方误差用于衡量预测值和真实值之间的平方差平均值。

公式为：

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- **优点：**对预测误差较大的情况给予更高惩罚
- **缺点：**对异常值较敏感

PyTorch 实现

```
# 之后的代码省略import
import torch
import torch.nn as nn

# 创建真实值和预测值
y_true = torch.tensor([1.0, 2.0, 3.0])
y_pred = torch.tensor([1.1, 2.0, 2.9])

# 使用nn.MSELoss
mse_loss = nn.MSELoss()
loss = mse_loss(y_pred, y_true)
print("MSE Loss:", loss.item())
```

```
MSE Loss: 0.0066666617058217525
```

2. Mean Absolute Error (MAE)

基本理论

平均绝对误差 (MAE) 是预测值与实际值之间差值的绝对值的平均值。公式如下：

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

PyTorch实现

```
criterion = nn.L1Loss()
y_pred = torch.tensor([2.5, 0.0, 2.1], requires_grad=True)
y_true = torch.tensor([3.0, -0.5, 2.0])
loss = criterion(y_pred, y_true)
print("MAE Loss:", loss.item())
```

MAE Loss: 0.36666664481163025

3. Huber Loss

基本理论

Huber Loss 结合了均方误差和平均绝对误差的优点，当误差较小时使用MSE，当误差较大时使用MAE：

$$L(a) = \begin{cases} \frac{1}{2}(y - \hat{y})^2, & \text{if } |y - \hat{y}| \leq \delta \\ \delta \cdot (|y - \hat{y}| - \frac{\delta}{2}), & \text{if } |y - \hat{y}| > \delta \end{cases}$$

PyTorch实现

```
criterion = nn.HuberLoss(delta=1.0)
y_pred = torch.tensor([2.5, 0.0, 2.1], requires_grad=True)
y_true = torch.tensor([3.0, -0.5, 2.0])
loss = criterion(y_pred, y_true)
print('HuberLoss:', loss.item())
```

```
HuberLoss: 0.08500000089406967
```