

# 法律声明

本课件包括演示文稿、示例、代码、题库、视频和声音等内容，深度之眼和讲师拥有完全知识产权；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或者机构不得盗版、复制、仿造其中的创意和内容，我们保留一切通过法律手段追究违反者的权利。

## 课程详情请咨询

- 微信公众号：深度之眼
- 客服微信号：deepshare0920



公众号



微信

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文





deepshare.net

深度之眼

# 损失函数

导师：余老师

---

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文



# 目录

1/ 损失函数概念

2/ 交叉熵损失函数

3/ NLL/BCE/BCEWithLogits Loss



# 损失函数是什么

What is Loss Function?

---



# 损失函数是什么

What is Loss Function

损失函数：衡量模型输出与真实标签的差异

损失函数(Loss Function):

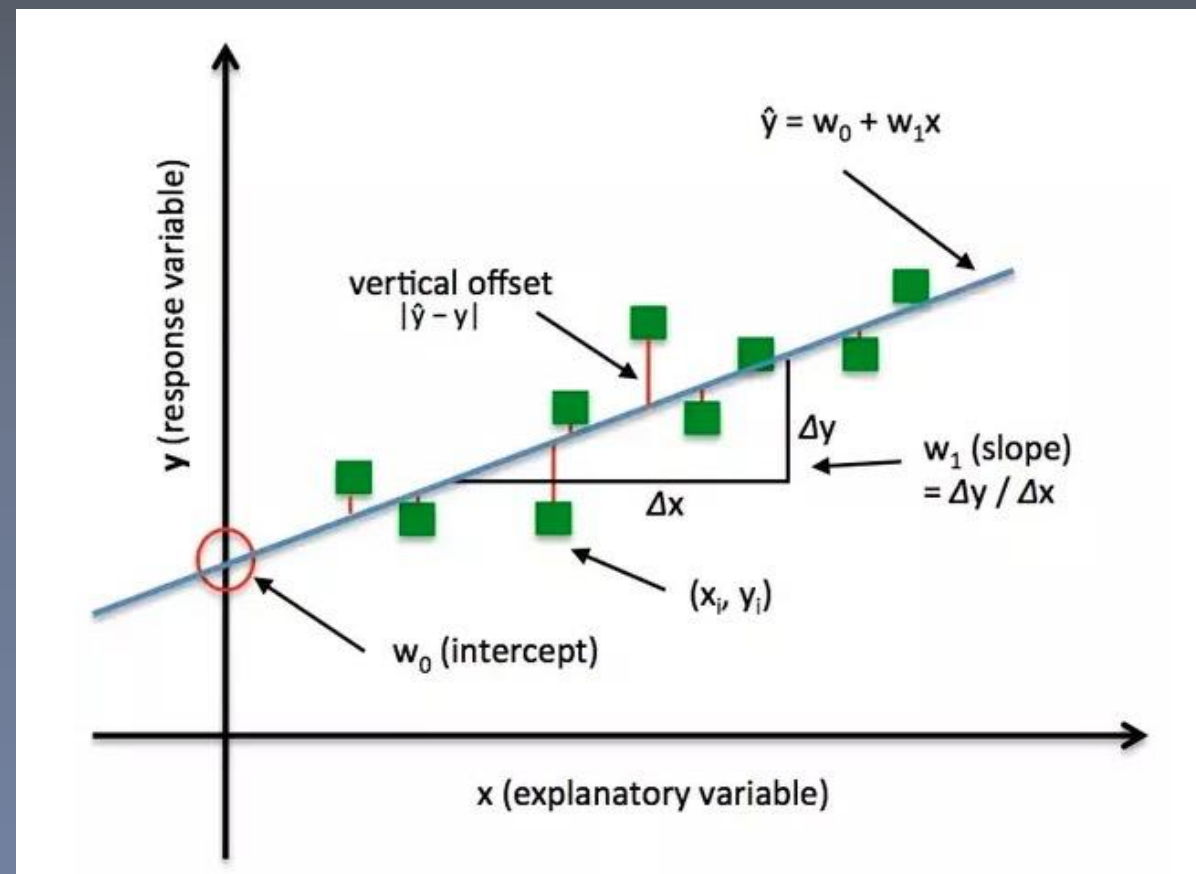
$$Loss = f(\hat{y}, y)$$

代价函数(Cost Function):

$$Cost = \frac{1}{N} \sum_i^N f(\hat{y}_i, y_i)$$

目标函数(Objective Function):

$$Obj = Cost + Regularization$$



# 损失函数是什么

What is Loss Function



deepshare.net

深度之眼

```
class _Loss(Module):
    def __init__(self, size_average=None, reduce=None,
                  reduction='mean'):
        super(_Loss, self).__init__()
        if size_average is not None or reduce is not None:
            self.reduction = _Reduction.legacy_get_string(
                size_average, reduce)
        else:
            self.reduction = reduction
```

# 损失函数

Loss Function



deepshare.net

深度之眼

## 1、nn.CrossEntropyLoss

功能： nn.LogSoftmax()与nn.NLLLoss()结合，进行交叉熵计算

主要参数：

- **weight**：各类别的loss设置权值
- **ignore\_index**：忽略某个类别
- **reduction**：计算模式，可为none/sum/mean

none- 逐个元素计算

sum- 所有元素求和，返回标量

mean- 加权平均，返回标量

```
nn.CrossEntropyLoss(weight=None,  
                      size_average=None,  
                      ignore_index=-100,  
                      reduce=None,  
                      reduction='mean')
```

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

$$\text{loss}(x, \text{class}) = - \log \left( \frac{\exp(x[\text{class}])}{\sum_j \exp(x[j])} \right) = -x[\text{class}] + \log \left( \sum_j \exp(x[j]) \right)$$

$$\text{loss}(x, \text{class}) = \text{weight}[\text{class}] \left( -x[\text{class}] + \log \left( \sum_j \exp(x[j]) \right) \right)$$

# 损失函数

Loss Function



deepshare.net

深度之眼

交叉熵 = 信息熵 + 相对熵

交叉熵:  $H(P, Q) = -\sum_{i=1}^N P(x_i) \log Q(x_i)$

自信息:  $I(x) = -\log[p(x)]$

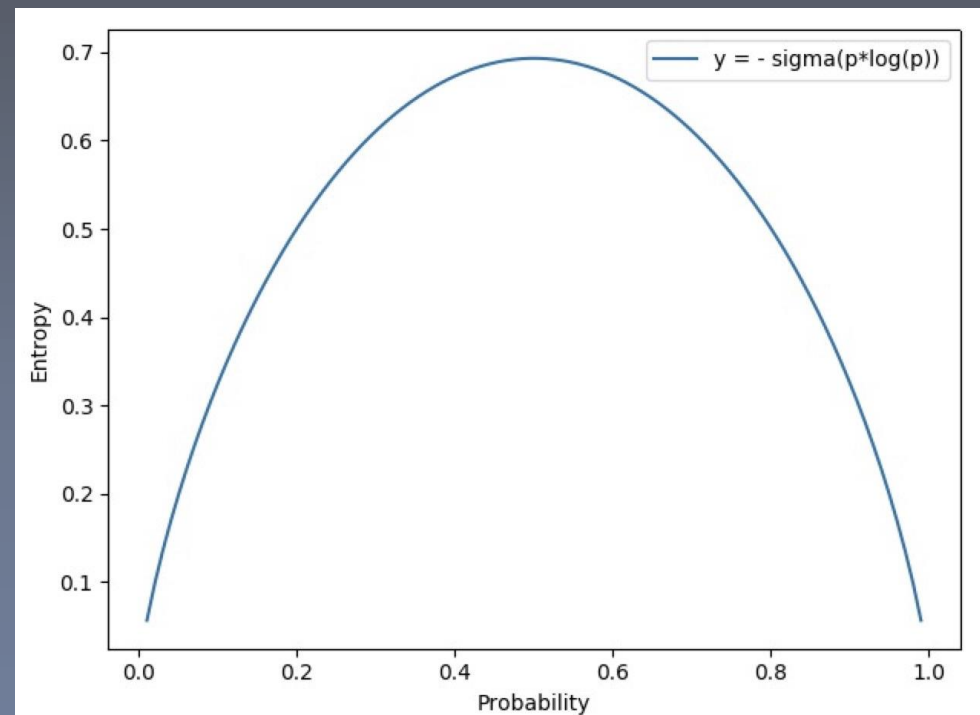
熵:  $H(P) = E_{x \sim p}[I(x)] = -\sum_{i=1}^N P(x_i) \log P(x_i)$

相对熵:  $D_{KL}(P, Q) = E_{x \sim p} \left[ \log \frac{P(x)}{Q(x)} \right]$   
 $= E_{x \sim p} [\log P(x) - \log Q(x)]$

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

$$= \sum_{i=1}^N P(x_i) \log P(x_i) - \sum_{i=1}^N P(x_i) \log Q(x_i) \quad \text{交叉熵: } H(P, Q) = D_{KL}(P, Q) + H(P)$$

$$= H(P, Q) - H(P)$$



关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文



# 损失函数

Loss Function



deepshare.net

深度之眼

## 2、nn.NLLLoss

功能：实现负对数似然函数中的**负号功能**

主要参数：

- **weight**：各类别的loss设置权值
- **ignore\_index**：忽略某个类别
- **reduction**：计算模式，可为none/sum/mean

none-逐个元素计算

sum-所有元素求和，返回标量

mean-加权平均，返回标量

```
nn.NLLLoss(weight=None,  
            size_average=None,  
            ignore_index=-100,  
            reduce=None,  
            reduction='mean')
```

$$\ell(x, y) = L = \{l_1, \dots, l_N\}^T, \quad l_n = -w_{y_n} x_{n, y_n}$$

# 损失函数

Loss Function



deepshare.net

深度之眼

## 3、nn.BCELoss

功能：二分类交叉熵

注意事项：输入值取值在[0,1]

主要参数：

- **weight**：各类别的loss设置权值
- **ignore\_index**：忽略某个类别
- **reduction**：计算模式，可为none/sum/mean

none-逐个元素计算

sum-所有元素求和，返回标量

mean-加权平均，返回标量

```
nn.BCELoss(weight=None,  
            size_average=None,  
            reduce=None,  
            reduction='mean')
```

$$l_n = -w_n [y_n \cdot \log x_n + (1 - y_n) \cdot \log(1 - x_n)]$$



# 损失函数

Loss Function



deepshare.net

深度之眼

## 4、nn.BCEWithLogitsLoss

功能：结合Sigmoid与二分类交叉熵

注意事项：网络最后不加sigmoid函数

主要参数：

- **pos\_weight**：正样本的权值
- **weight**：各类别的loss设置权值
- **ignore\_index**：忽略某个类别
- **reduction**：计算模式，可为none/sum/mean

none-逐个元素计算

sum-所有元素求和，返回标量

mean-加权平均，返回标量

```
nn.BCEWithLogitsLoss(weight=None,  
                      size_average=None,  
                      reduce=None, reduction='mean',  
                      pos_weight=None)
```

$$l_n = -w_n [y_n \cdot \log \sigma(x_n) + (1 - y_n) \cdot \log(1 - \sigma(x_n))]$$

# —— 结 语 ——

在这次课程中，学习了损失函数的概念，  
以及4种损失函数

在下次课程中，我们将会学习  
**pytorch中其余14种损失函数**



关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文





**deepshare.net**

深度之眼

联系我们:

电话: 18001992849

邮箱: [service@deepshare.net](mailto:service@deepshare.net)

QQ: 2677693114



公众号



客服微信

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文