

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/

5. nn.L1Loss

6. nn.MSELoss

7. nn.SmoothL1Loss

8. nn.PoissonNLLLoss

9. nn.KLDivLoss

10. nn.MarginRankingLoss

11. nn.MultiLabelMarginLoss

12. nn.SoftMarginLoss

13. nn.MultiLabelSoftMarginLoss

14. nn.MultiMarginLoss

15. nn.TripletMarginLoss

16. nn.HingeEmbeddingLoss

17. nn.CosineEmbeddingLoss

18. nn.CTCLoss

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



5、nn.L1Loss

功能： 计算inputs与target之差的绝对值

6、nn.MSELoss

功能： 计算inputs与target之差的平方

主要参数：

- **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.L1Loss(size_average=None, reduce=None,  
reduction='mean')
```

$$l_n = |x_n - y_n|$$

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

$$l_n = (x_n - y_n)^2$$

损失函数

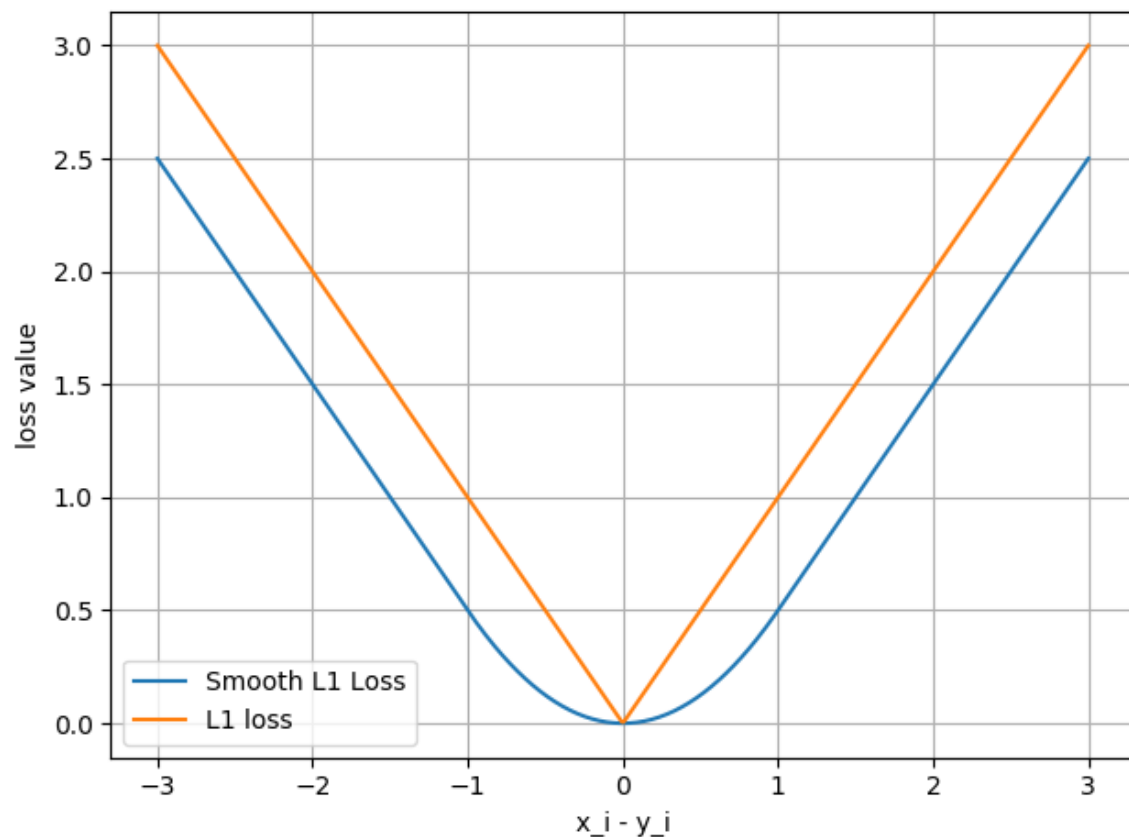
Loss Function



deepshare.net

深度之眼

7、SmoothL1Loss



```
nn.SmoothL1Loss(size_average=None, reduce=None, reduction='mean')
```

$$\text{loss}(x, y) = \frac{1}{n} \sum_i z_i$$

$$z_i = \begin{cases} 0.5(x_i - y_i)^2, & \text{if } |x_i - y_i| < 1 \\ |x_i - y_i| - 0.5, & \text{otherwise} \end{cases}$$

损失函数

Loss Function



deepshare.net

深度之眼

8、PoissonNLLLoss

功能：泊松分布的负对数似然损失函数

主要参数：

- **log_input**：输入是否为对数形式，决定计算公式
- **full**：计算所有loss，默认为False
- **eps**：修正项，避免log (input) 为nan

```
nn.PoissonNLLLoss(log_input=True, full=False,  
size_average=None, eps=1e-08, reduce=None,  
reduction='mean')
```

log_input = **True**
 $\text{loss}(\text{input}, \text{target}) = \exp(\text{input}) - \text{target} * \text{input}$

log_input = **False**
 $\text{loss}(\text{input}, \text{target}) = \text{input} - \text{target} * \log(\text{input} + \text{eps})$

损失函数

Loss Function



deepshare.net

深度之眼

9、nn.KLDivLoss

功能：计算KLD (divergence) , KL散度, 相对熵

注意事项：需提前将输入计算log-probabilities, 如通过nn.logsoftmax()

主要参数：

- **reduction** : none/sum/mean/batchmean

batchmean- batchsize维度求平均值

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.KLDivLoss(size_average=None, reduce=None, reduction='mean')

$$D_{KL}(P \parallel Q) = E_{x \sim p} \left[\log \frac{P(x)}{Q(x)} \right] = E_{x \sim p} [\log P(x) - \log Q(x)]$$
$$= \sum_{i=1}^N \underbrace{P(x_i)} \cdot (\underbrace{\log P(x_i)} - \underbrace{\log Q(x_i)})$$

$$l_n = \underbrace{y_n} \cdot (\underbrace{\log y_n} - \underbrace{x_n})$$

损失函数

Loss Function



deepshare.net

深度之眼

10、nn.MarginRankingLoss

功能：计算两个向量之间的相似度，用于排序任务

特别说明：该方法计算两组数据之间的差异，返回一个n*n的 loss 矩阵

主要参数：

- **margin**：边界值，x1与x2之间的差异值
- **reduction**：计算模式，可为none/sum/mean

y = 1时，希望x1比x2大，当x1 > x2时，不产生loss

y = -1时，希望x2比x1大，当x2 > x1时，不产生loss

```
nn.MarginRankingLoss(margin=0.0,  
size_average=None, reduce=None,  
reduction='mean')
```

$$\text{loss}(x, y) = \max(0, -y * \underline{x_2 - x_1}) + \text{margin})$$

损失函数

Loss Function



deepshare.net

深度之眼

11、nn.MultiLabelMarginLoss



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

12、 nn.SoftMarginLoss

功能：计算二分类的logistic损失

主要参数：

- **reduction**：计算模式，可为none/sum/mean

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

$$\text{loss}(x, y) = \sum_i \frac{\log(1 + \exp(-y[i] * x[i]))}{x.\text{nelement}()}$$

损失函数

Loss Function



13、nn.MultiLabelSoftMarginLoss

功能：SoftMarginLoss多标签版本

主要参数：

- **weight**: 各类别的loss设置权值
- **reduction**: 计算模式，可为none/sum/mean

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

$$loss(x, y) = -\frac{1}{C} * \sum_i y[i] * \log((1 + \exp(-x[i]))^{-1}) + (1 - y[i]) * \log\left(\frac{\exp(-x[i])}{(1 + \exp(-x[i]))}\right)$$

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



14、nn.MultiMarginLoss

功能：计算多分类的折页损失

主要参数：

- **p**：可选1或2
- **weight**：各类别的loss设置权值
- **margin**：边界值
- **reduction**：计算模式，可为none/sum/mean



```
nn.MultiMarginLoss(p=1, margin=1.0, weight=None,
size_average=None, reduce=None, reduction='mean')
```

$$\text{loss}(x, y) = \frac{\sum_i \max(0, \text{margin} - x[y] + x[i])^p}{x.\text{size}(0)}$$

where $x \in \{0, \dots, x.\text{size}(0) - 1\}$, $y \in \{0, \dots, y.\text{size}(0) - 1\}$, $0 \leq y[j] \leq x.\text{size}(0) - 1$, and $i \neq y[j]$ for all i and j .

损失函数

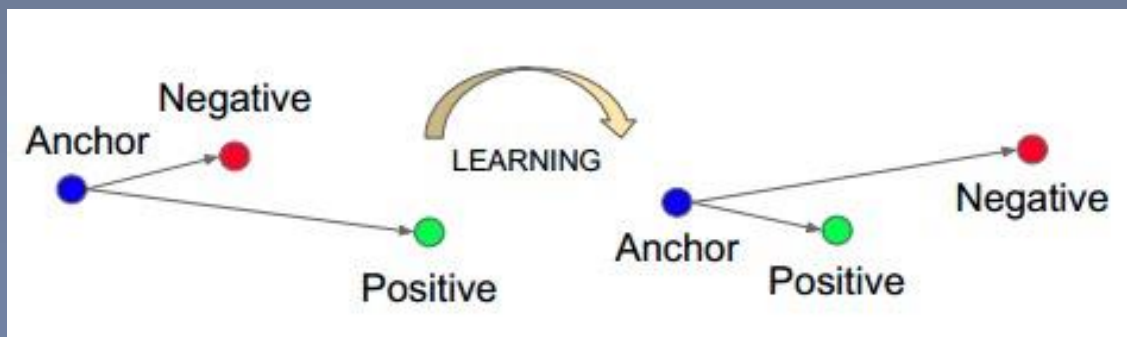
Loss Function

15、nn.TripletMarginLoss

功能：计算三元组损失，人脸验证中常用

主要参数：

- **p**：范数的阶，默认为2
- **margin**：边界值
- **reduction**：计算模式，可为none/sum/mean



```
nn.TripletMarginLoss(margin=1.0, p=2.0, eps=1e-06,  
swap=False, size_average=None, reduce=None,  
reduction='mean')
```

$$L(a, p, n) = \max\{d(a_i, p_i) - d(a_i, n_i) + \text{margin}, 0\}$$

$$d(x_i, y_i) = \|\mathbf{x}_i - \mathbf{y}_i\|_p$$

损失函数

Loss Function



deepshare.net

深度之眼

16、nn.HingeEmbeddingLoss

功能：计算两个输入的相似性，常用于非线性embedding和半监督学习

特别注意：输入x应为两个输入之差的绝对值

主要参数：

- **margin**：边界值
- **reduction**：计算模式，可为none/sum/mean

```
nn.HingeEmbeddingLoss(margin=1.0, size_average=None,
                        reduce=None, reduction='mean')
```

$$l_n = \begin{cases} x_n, & \text{if } y_n = 1, \\ \max\{0, \Delta - x_n\}, & \text{if } y_n = -1, \end{cases}$$

损失函数

Loss Function



deepshare.net

深度之眼

17、nn.CosineEmbeddingLoss

功能：采用余弦相似度计算两个输入的相似性

主要参数：

- **margin**：可取值 $[-1, 1]$ ，推荐为 $[0, 0.5]$
- **reduction**：计算模式，可为none/sum/mean

`nn.CosineEmbeddingLoss(margin=0.0, size_average=None, reduce=None, reduction='mean')`

$$\text{loss}(x, y) = \begin{cases} 1 - \cos(x_1, x_2), & \text{if } y = 1 \\ \max(0, \cos(x_1, x_2) - \text{margin}), & \text{if } y = -1 \end{cases}$$

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

损失函数

Loss Function



deepshare.net

深度之眼

18、nn.CTCLoss

功能： 计算CTC损失，解决时序类数据的分类

Connectionist Temporal Classification

主要参数：

- **blank** : blank label
- **zero_infinity** : 无穷大的值或梯度置0
- **reduction** : 计算模式，可为none/sum/mean

```
torch.nn.CTCLoss(blank=0, reduction='mean', zero_infinity=False)
```

参考文献：

A. Graves et al.: Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks

损失函数

Loss Function



deepshare.net

深度之眼

1. nn.CrossEntropyLoss
2. nn.NLLLoss
3. nn.BCELoss
4. nn.BCEWithLogitsLoss
5. nn.L1Loss
6. nn.MSELoss
7. nn.SmoothL1Loss
8. nn.PoissonNLLLoss
9. nn.KLDivLoss
10. nn.MarginRankingLoss
11. nn.MultiLabelMarginLoss
12. nn.SoftMarginLoss
13. nn.MultiLabelSoftMarginLoss
14. nn.MultiMarginLoss
15. nn.TripletMarginLoss
16. nn.HingeEmbeddingLoss
17. nn.CosineEmbeddingLoss
18. nn.CTCLoss

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

—— 结 语 ——

在这次课程中，学习了14种损失函数

在下次课程中，我们将会学习

pytorch的优化器



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

Q Q：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