

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

优化器 Optimizer (二)

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/ learning rate 学习率

2/ momentum 动量

3/ torch.optim.SGD

4/ Pytorch的十种优化器

学习率

Learning Rate



deepshare.net

深度之眼

梯度下降: $w_{i+1} = w_i - g(w_i)$

$$y = f(x) = 4 * x^2$$

$$y' = f'(x) = 8 * x$$

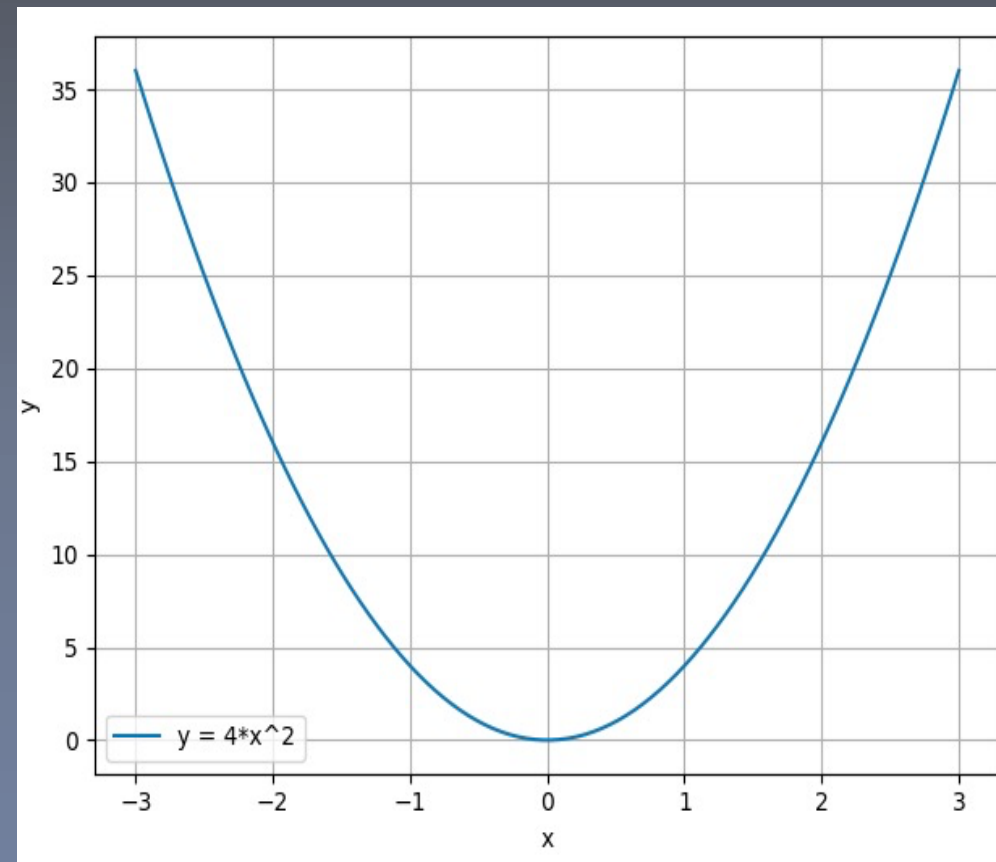
$$x_0 = 2, \quad y_0 = 16, \quad f'(x_0) = 16$$

$$x_1 = x_0 - f'(x_0) = 2 - 16 = -14$$

$$x_1 = -14, \quad y_1 = 784, \quad f'(x_1) = -112$$

$$x_2 = x_1 - f'(x_1) = -14 + 112 = 98, \quad y_2 = 38416$$

.....



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

学习率

Learning Rate



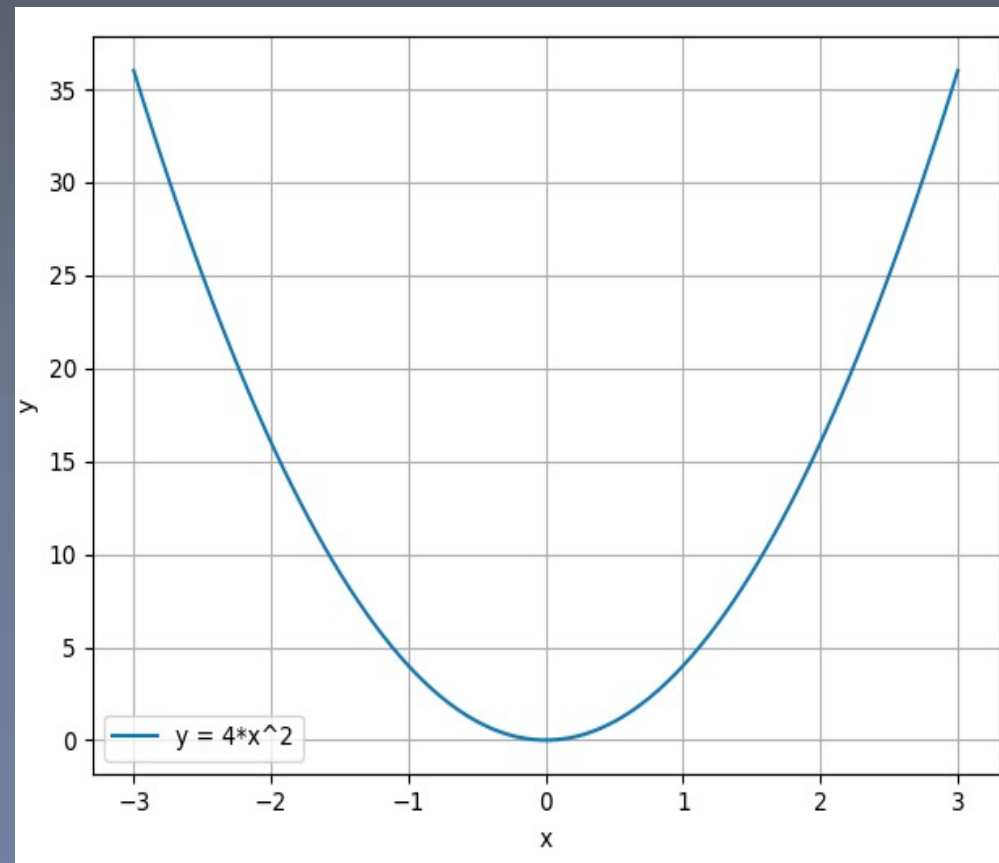
deepshare.net

深度之眼

梯度下降: $w_{i+1} = w_i - g(w_i)$

$w_{i+1} = w_i - \text{LR} * g(w_i)$

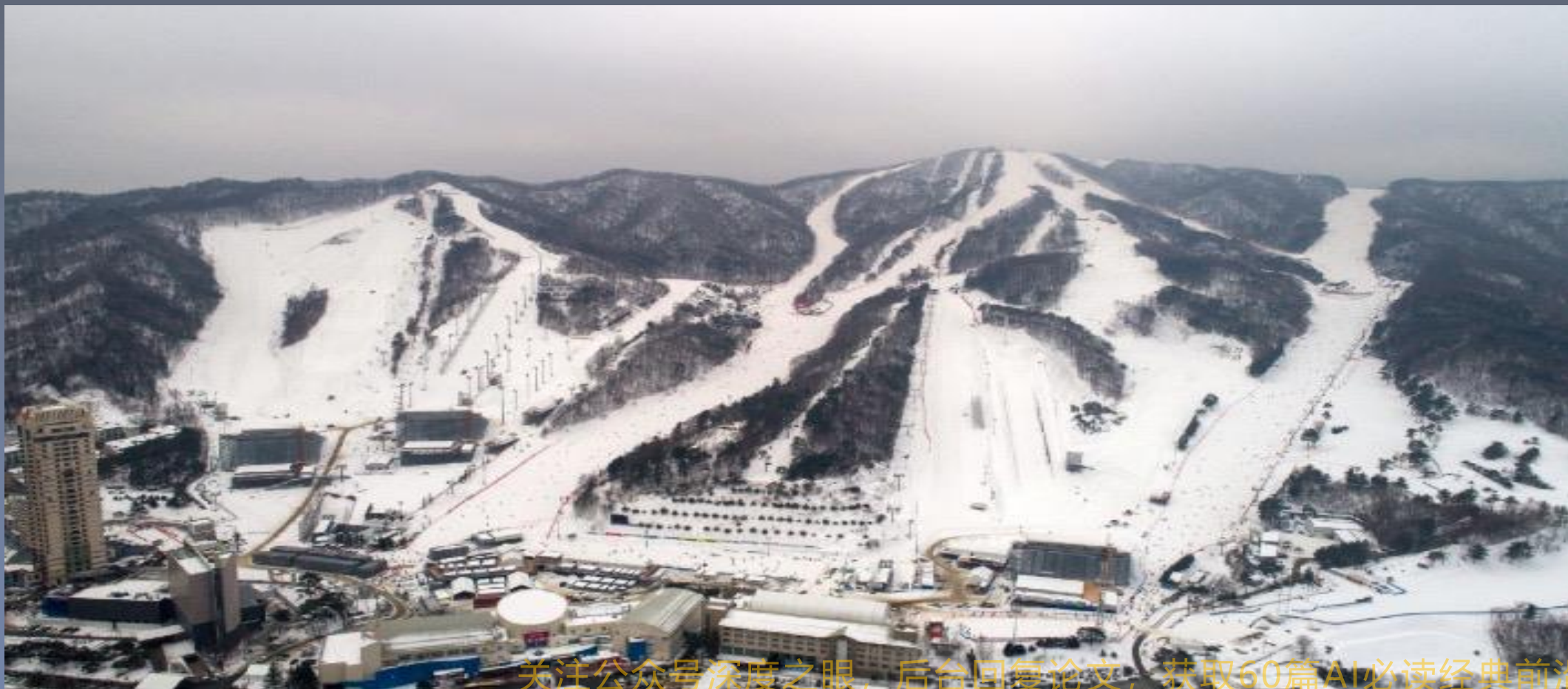
学习率 (learning rate) 控制更新的步伐



优化器

Optimizer

Momentum（动量，冲量）：结合当前梯度与上一次更新信息，用于当前更新



优化器

Optimizer



deepshare.net

深度之眼

Momentum (动量, 冲量)

指数加权平均: $v_t = \beta * v_{t-1} + (1 - \beta) * \theta_t$

$$v_{100} = \beta * v_{99} + (1 - \beta) * \theta_{100}$$

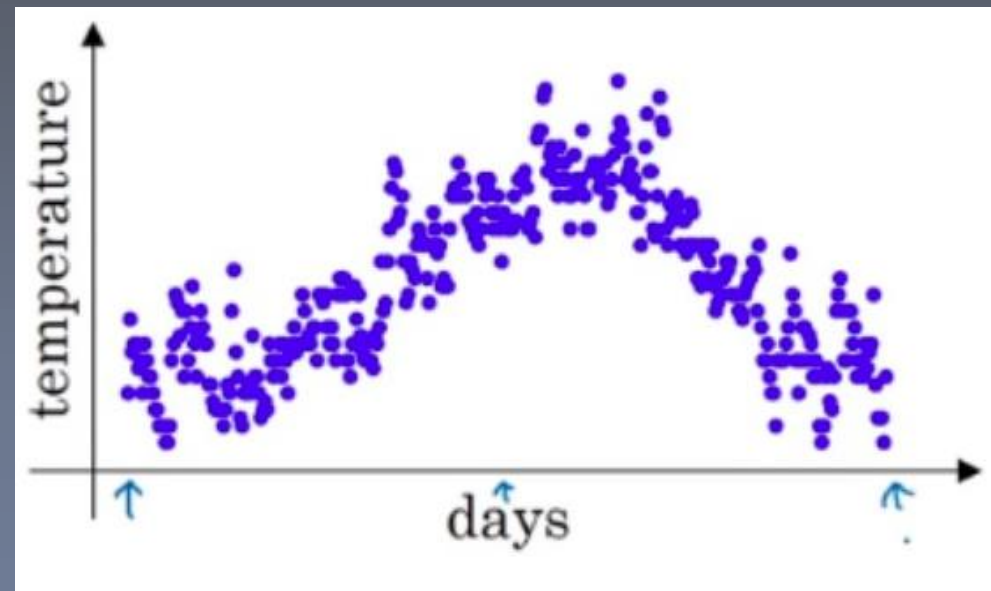
$$= (1 - \beta) * \theta_{100} + \beta * (\beta * v_{98} + (1 - \beta) * \theta_{99})$$

$$= (1 - \beta) * \theta_{100} + (1 - \beta) * \beta * \theta_{99} + (\beta^2 * v_{98})$$

$$= (1 - \beta) * \theta_{100} + (1 - \beta) * \beta * \theta_{99} + (1 - \beta) * \beta^2 * \theta_{98} + (\beta^3 * v_{97})$$

$$= (1 - \beta) * \theta_{100} + (1 - \beta) * \beta^1 * \theta_{99} + (1 - \beta) * \beta^2 * \theta_{98} + (\beta^3 * v_{97})$$

$$= \sum_i^N (1 - \beta) * \beta^i * \theta_{N-i}$$



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

优化器

Optimizer



deepshare.net

深度之眼

Momentum（动量，冲量）

梯度下降：

$$w_{i+1} = w_i - lr * g(w_i)$$

pytorch中更新公式：

$$v_i = m * v_{i-1} + g(w_i)$$

$$w_{i+1} = w_i - lr * v_i$$

w_{i+1} ：第i+1次更新的参数

lr：学习率

v_i ：更新量

m：momentum系数

$g(w_i)$ ： w_i 的梯度

$$\begin{aligned} v_{100} &= m * v_{99} + g(w_{100}) \\ &= g(w_{100}) + m * (m * v_{98} + g(w_{99})) \\ &= g(w_{100}) + m * g(w_{99}) + m^2 * v_{98} \\ &= g(w_{100}) + m * g(w_{99}) + m^2 * g(w_{98}) + m^3 * v_{97} \end{aligned}$$

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

优化器

Optimizer



1.optim.SGD

主要参数：

- **params**：管理的参数组
- **lr**：初始学习率
- **momentum**：动量系数，贝塔
- **weight_decay**：L2正则化系数
- **nesterov**：是否采用NAG

```
optim.SGD(params, lr=<object object>,  
momentum=0, dampening=0,  
weight_decay=0, nesterov=False)
```

NAG参考文献：《On the importance of initialization and momentum in deep learning》

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

优化器

Optimizer



deepshare.net

深度之眼

1. `optim.SGD`: 随机梯度下降法
2. `optim.Adagrad`: 自适应学习率梯度下降法
3. `optim.RMSprop`: Adagrad的改进
4. `optim.Adadelta`: Adagrad的改进
5. `optim.Adam`: RMSprop结合Momentum
6. `optim.Adamax`: Adam增加学习率上限
7. `optim.SparseAdam`: 稀疏版的Adam
8. `optim.ASGD`: 随机平均梯度下降
9. `optim.Rprop`: 弹性反向传播
10. `optim.LBFGS`: BFGS的改进

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

优化器



1. optim.SGD: 《On the importance of initialization and momentum in deep learning》
2. optim.Adagrad: 《Adaptive Subgradient Methods for Online Learning and Stochastic Optimization》
3. optim.RMSprop:
http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
4. optim.Adadelta: 《AN ADAPTIVE LEARNING RATE METHOD》
5. optim.Adam: 《Adam: A Method for Stochastic Optimization》
6. optim.Adamax: 《Adam: A Method for Stochastic Optimization》
7. optim.SparseAdam
8. optim.ASGD: 《Accelerating Stochastic Gradient Descent using Predictive Variance Reduction》
9. optim.Rprop: 《Martin Riedmiller und Heinrich Braun》
10. optim.LBFGS: BDGS的改进

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，学习了优化器Optimizer

在下次课程中，我们将会学习

学习率调整策略



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

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