

随 epoch 调整学习率 Scheduler

lr_scheduler.step() 1. 更新学习率 (位于每个 epoch 结束时)
学习率调整

lr_scheduler.StepLR(optimizer, step_size, gamma=0.1)

等间隔调整学习率

step_size 调整间隔 epoch 数

gamma: 系数

$$lr = lr \times gamma$$

2. Multi StepLR

按给定间隔调整学习率

列表 [0, 75, 150]



lr_scheduler.MultiStepLR(optimizer, milestones, gamma)

3. ExponentialLR.

按指数衰减学习率

lr_scheduler.ExponentialLR(optimizer, gamma)

$$lr = lr \times \gamma^{\text{epoch}}$$

4. CosineAnnealingLR.

余弦周期调整学习率

lr_scheduler.CosineAnnealingLR(optimizer, T_max, eta_min)

T_max: 下降周期.

eta_min: 学习率下限.

$$\eta_t = \eta_{\min} + \frac{1}{2} (\eta_{\max} - \eta_{\min}) \left(1 + \cos\left(\frac{T_{\text{cur}}}{T_{\text{max}}} \pi\right) \right)$$

5. ReduceLROnPlateau X:

监控指标, 当指标不再变化则调整

lr_scheduler.ReduceLROnPlateau(optimizer, ...)

mode: min/max 两种模式 (监控指标下降/上升)

factor: 调整系数

patience: "耐心", 接受几次不变化

Cool down: "冷却时间", 停止监控一段时间 (每隔 cooldown 个 epoch)

verbose: 是否打印日志

min_lr: 学习率下限

eps: 学习率衰减最小值

~~X~~ scheduler.lr_step(\checkmark), 监控的参数

6. LambdaLR.

自定义调整策略

lr_scheduler.LambdaLR(optimizer, lr_lambda)

lr_lambda: 函数/列表