网络优化 正则化

一、小批量梯度下降

选K个训练样本，$\mathcal{I}_t=\{(x^{(k)},y^{(k)})\}_{k=1}^{K}$

$$g_t(\theta)=\frac{1}{K}\sum_{(x^{(k)},y^{(k)})\in\mathcal{I}_t}\frac{\partial L(y^{(k)},f(x^{(k)};\theta))}{\partial\theta}$$

$g_t\triangleq g_t(\theta_{t-1})$

$\theta_t\leftarrow\theta_{t-1}-\alpha g_t$ $\alpha>0$ 学习率

$\Delta\theta_t\triangleq\theta_t-\theta_{t-1}$

$\Delta\theta_t$和梯度g_t并不需要完全一致

① 合适的K（批量大小）

② 学习率α

③ 参数更新方向

① 批量大小：

不影响随机梯度的期望，但影响方差：

批量↑ → 方差越小 → 噪声越小 → 训练稳定

→ 设置较大学习率

学习率和批量都是线性缩放

② 学习率调整：

学习率衰减

1）分段常数衰减

2）逆时衰减 $\alpha_t=\alpha_0\frac{1}{1+\beta t}$

3）指数衰减 $\alpha_t=\alpha_0\beta^t$

4）自然指数衰减 $\alpha_t=\alpha_0\exp(-\beta t)$

5）余弦 $\alpha_t=\frac{1}{2}\alpha_0\left(1+\cos\frac{t\pi}{T}\right)$

让学习率跳出

鞍点：随机

梯度较大

不稳定

可以适当减小

学习率

周期性调整学习率：为了让其跳出局部最小值

虽然短期损害优化过程

但长期收敛至较为局部最小值

$m=\lfloor 1+\frac{t}{2\Delta T}\rfloor$

循环学习率（方向）$\alpha_t=\alpha_{min}^{m}+(\alpha_{max}^{m}-\alpha_{min}^{m})(\max(0,1-b))$

热重启的随机梯度下降

$b=|\frac{t}{\Delta T}-2m+1|$

1）Adagrad 算法

不同参数的收敛情况分别设置

$G_t=\sum_{\tau=1}^{t}g_\tau\odot g_\tau$

$$\Delta\theta_t=-\frac{\alpha}{\sqrt{G_t+\epsilon}}\odot g_t$$

G_t↑ $\Delta\theta_t$↓

G_t↓ $\Delta\theta_t$↑

学习率都是随着迭代次数增加，学习率减小

缺点：一定次数迭代后若没有到最优点，由于此时很小，将很难再到最优点

即早停下降

2）RMSprop

避免 Adagrad 中学习率单调下降过早衰减

$$G_t=\beta G_{t-1}+(1-\beta)g_t\odot g_t$$

$$=(1-\beta)\sum_{\tau=1}^{t}\beta^{t-\tau}g_\tau\odot g_\tau$$

指数移动平均

$$\Delta\theta_t=-\frac{\alpha}{\sqrt{G_t+\epsilon}}\odot g_t$$

3) Adadelta 算法

$\Delta X^2_{t-1} = \beta_1 \Delta X^2_{t-2} + (1-\beta_1)\,\Delta\theta_{t-1}\cdot\Delta\theta_{t-1}$

通过梯度平方的指数衰减移动平均来调整学习率 Gt 和 RMSprop 一样

$$\Delta\theta_t = -\frac{\sqrt{\Delta X^2_{t-1}+\epsilon}}{\sqrt{G_t+\epsilon}} g_t$$

④更新优化方向

动量法：

$$\Delta\theta_t = \rho\Delta\theta_{t-1} - \alpha g_t$$
$$= -\alpha\sum_{\tau=1}^{t}\rho^{t-\tau} g_\tau$$

负梯度的"加权移动平均"

ρ 为动量因子.

α 为学习率

2) Nesterov 加速梯度

动量法可拆分为两步

$$\hat\theta = \theta_{t-1} + \rho\Delta\theta_{t-1}$$

$\theta_t = \hat\theta - \alpha g_t$ 不是在 $\hat\theta$ 点的梯度

$$\Delta\theta_t = \rho\Delta\theta_{t-1} - \alpha g_t(\theta_{t-1}+\rho\Delta\theta_{t-1})$$

3) Adam 算法

同时更新学习率和梯度.

RMSprop + 动量法

$$\Delta\theta_t = -\frac{\alpha}{\sqrt{\hat G_t+\epsilon}}\hat M_t \quad (\alpha=0.001,\ \alpha_t=\frac{\alpha}{\sqrt{t}})$$

$$\begin{cases} G_t = \beta_2 G_{t-1} + (1-\beta_2) g_t \odot g_t & \beta_2 = 0.99 \\ M_t = \beta_1 M_{t-1} + (1-\beta_1) g_t \end{cases}$$

$$\hat M_t = \frac{M_t}{1-\beta_1} \qquad \hat G_t = \frac{G_t}{1-\beta_2}$$

2. 参数初始化

全为0 → 对称权重

随机初始化 {太小：{多层所导梯度 sigmod 主为非线性；太大：sigmoid 梯度为0

保持每个神经元输入输出方差一致

{高斯分布；均匀分布

类别	优化算法
固定衰减 α	分段常数，逆时，指数，余弦
周期 α	为何振荡 SGDR
自适应 α	Adagrad, RMSprop, Adadelta
梯度方向优化	动量法，Nesterov，梯度截断
综合	Adam

$$\Delta\theta_t = -\frac{\alpha_t}{\sqrt{G_t+\epsilon}} M_t$$

$G_t = \psi(g_1,\cdots g_t)$

$M_t = \phi(g_1,\cdots g_t)$

Xavier 初始化

神经元数量计算方差

高斯：$N(0, \sqrt{\frac{2}{n^{(l-1)}+n^{(l)}}})$

均匀：$r = \sqrt{\frac{6}{n^{l-1}+n^{l}}}$