Lecture 2　Gradient and Stochastic Gradient Descent

Alex　2019 年 7 月 23 日

一　梯度下降

1.1　梯度

假设目标函数 $f: R^d \to R$ 可微。无约束最小化问题写作：

$$x^* = \arg\min_{x \in R^d} f(x)$$

假设函数 $f: R^d \to R$ 在 $x_0 \in R^d$ 处可微，则函数 f 在点 x_0 处的**梯度 (gradient)**，记作 $\nabla_x f(x_0)$，是一个 R^d 空间的向量：方向——函数值增长最快，即增长率最大的方向，大小——函数值的增长率大小。由于梯度的方向是函数值增长最快的方向，因此，如果我们希望到达函数值最小的地方，可以沿着**梯度的反方向**一步一步走下去。这样通过向梯度反方向不停迭代最终到达最低点的方法，称为**梯度下降法 (Gradient Descent)**：

- 初始化 x=0；
- 重复：$x \leftarrow x - \eta \nabla f(x)$
- 直到满足停止条件。

1.2　步长

梯度下降法中有一个超参数——步长 (step size)，记作 η。一种常用的设置是“固定步长”，即将 η 设置为一个恒定的值，例如 0.1，这种设定被称为 Fixed Step Size。注意，这里说的固定步长，仅仅指 η 恒定，而不是指每次迭代的步长恒定；相反，如果 η 恒定，每次迭代的步长反而是不同的，因为每次迭代的步长大小为 $\eta \cdot ||\nabla f||$，而一般每一个点的梯度大小是不一样的。

对于 Fixed Step Size，已经有理论能够保证，当该固定步长足够小时，梯度下降法能够收敛：Suppose $f: R^d \to R$ is convex and differentiable, and ∇f is Lipschitz continuous with constant $L > 0$, i.e. $||\nabla f(x) - \nabla f(y)|| \leq L||x - y||$. For any $x, y \in R^d$. Then gradient descent with fixed step size $\eta \leq 1/L$ converges. In particular,

$$f(x^{(k)}) - f(x^*) \leq \frac{||x^{(0)} - x^*||^2}{2\eta k}$$

上述定理说明，当目标函数的梯度是利普希茨连续时，任何小于利普希茨常数的倒数的步长，都能够收敛。因此，在梯度变化较快的地方 (二阶导较大)，L 较大，则理论保证的 η 较小，应该放慢脚步；在梯度变化较慢的地方 (二阶导较小)，L 较小，则理论保证的 η 较大，应该迈大步。综上，该收敛定理为我们提供了一种动态调整步长的方式。关于利普希茨连续的理解，可参考 [非凸优化基石：Lipschitz Condition-Zeap][1]。

1.3 小批量梯度下降与随机梯度下降

假设空间记作 $\mathcal{F} = \{f_w : \mathcal{X} \to \mathcal{A} | w \in R^d\}$，则我们说 hypothesis 被 w 参数化，即一个 w 就代表着一个特定的 hypothesis。经验误差最小化 (以下写作 ERM) 即寻找最佳的参数 w 以最小化：

$$\hat{R}_n(w) = \frac{1}{n}\sum_{i=1}^{n} l(f_w(x_i), y_i)$$

假设损失函数 l 可微，则对于上面的最小化问题，我们可以使用梯度下降法。梯度的表达式如下：

$$\nabla_w \hat{R}_n(w) = \frac{1}{n}\sum_{i=1}^{n} \nabla_w l(f_w(x_i), y_i)$$

可见，对于每一个 w 处的梯度，我们都要对每一个样本进行计算，然后将结果加总起来，得到" 总" 的梯度，计算复杂度为 $O(n)$。这样的方式十分不利于处理大数据，速度会很慢。因此，在每一步迭代时，我们可能并不想算" 真正的梯度"，而看看能否计算一个好算的对于真正梯度的估计：

What if we just use an estimate of the gradient?

考虑 Minibatch Gradient—— “部分梯度”：

$$\nabla_w \hat{R}_N(w) = \frac{1}{N}\sum_{i=1}^{N} \nabla_w l(f_w(x_{m_i}), y_{m_i})$$

其中 $\{(x_{m_1}, y_{m_1}), \cdots, (x_{m_N}, y_{m_N})\}$ 是原始数据的一个子集。易证，minibatch gradient 是 full gradient 的无偏估计：

$$E\Big[\nabla_w \hat{R}_N(w)\Big] = \nabla_w \hat{R}_n(w)$$

当然，N 越大，估计的越准 (N=n 则 100% 正确估计)。但 N 越大，计算耗费也就越高。因此需要权衡取舍。特别的，当 N=1 时，这种梯度下降被称为**随机梯度下降 (Stochastic gradient descent)**。

对于 minibatch 梯度下降有一些经验法则：

- N 常常选择 1 到几百的数值；
- $N = 32$ 是一个不错的选择；

[1] https://zhuanlan.zhihu.com/p/27554191

- $N \geq 10$ 时能够得到显著的提速。

对于固定步长的 SGD 并没有收敛的理论保证，但是实践中的效果往往不错。对于逐渐减小步长 (decreasing step size) 的 SGD，有如下的理论保证：记 η_t 为第 t 轮迭代时的步长，

Robbins-Monro Conditions

Many classical convergence results depend on the following two conditions:

$$\sum_{t=1}^{\infty} \eta_t^2 < \infty, \quad \sum_{t=1}^{\infty} \eta_t = \infty$$

Robbins-Monro 条件说明，当步长按照满足上述形式的速度——不太快也不太慢地减小时，SGD 能够收敛。