共轭梯度是一种通过迭代下降的共轭方向 (conjugate directions) 以有效避免Hessian矩阵求逆计算的方法。这种方法的灵感来自于最速下降方法弱点的仔细研究(详细信息请参看第4.3节),其中线性搜索迭代地用于梯度相关的方向中。图8.6说明了该方法在二次碗型目标中如何表现的,是一个相当无效的来回往复,锯齿形模式。这是因为每一个由梯度给定的线性搜索方向,都保证正交于上一个线性搜索方向。

假设上一个搜索方向是  $d_{t-1}$ 。在极小值处,线性搜索终止,方向  $d_{t-1}$  处的方向导数为零:  $\nabla_{\theta}J(\theta)\cdot d_{t-1}=0$ 。因为该点的梯度定义了当前的搜索方向, $d_{t}=\nabla_{\theta}J(\theta)$ 将不会贡献于方向  $d_{t-1}$ 。因此方向  $d_{t}$  正交于  $d_{t-1}$ 。最速下降多次迭代中,方向  $d_{t-1}$ 和  $d_{t}$  之间的关系如图8.6所示。如图展示的,下降正交方向的选择不会保持前一搜索方向上的最小值。这产生了锯齿形的过程。在当前梯度方向下降到极小值,我们必

须重新最小化之前梯度方向上的目标。因此,通过遵循每次线性搜索结束时的梯度, 我们在某种意义上撤销了在之前线性搜索的方向上取得的进展。共轭梯度试图解决 这个问题。**7** 

在共轭梯度法,我们寻求一个和先前线性搜索方向共轭 (conjugate) 的搜索方向,即它不会撤销该方向上的进展。在训练迭代 t 时,下一步的搜索方向  $d_t$  的形式如下:

 $\mathbf{d}_t = \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) + \beta_t \mathbf{d}_{t-1},$  (8.29)

其中,系数  $\beta_t$  的大小控制我们应沿方向  $d_{t-1}$  加回多少到当前搜索方向上。

Require: 初始参数 $\theta_0$ Require: 包含 $m$ 个样本的训练集初始化 $\rho_0=0$ 初始化 $\rho_0=0$ 初始化 $t=1$ while 没有达到停止准则 do 初始化梯度 $t=1$ while 没有达到停止准则 do 初始化梯度 $t=1$	算法	<b>法 8.9</b> 共轭梯度方法
初始化 $\rho_0 = 0$ 初始化 $t = 1$ while 没有达到停止准则 do 初始化梯度 $g_t = 0$ 计算梯度: $g_t \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(\boldsymbol{x}^{(i)}; \theta), \boldsymbol{y}^{(i)})$ 计算 $\beta_t = \frac{(g_t - g_{t-1})^\top g_t}{g_{t-1}^\top g_{t-1}}$ (Polak-Ribière) (非线性共轭梯度: 视情况可重置 $\beta_t$ 为零,例如 $t$ 是常数 $k$ 的倍数时,如 $k = 5$ ) 计算搜索方向: $\rho_t = -g_t + \beta_t \rho_{t-1}$ 执行线搜索寻找: $\epsilon^* = \operatorname{argmin}_{\epsilon} \frac{1}{m} \sum_{i=1}^m L(f(\boldsymbol{x}^{(i)}; \theta_t + \epsilon \rho_t), \boldsymbol{y}^{(i)})$ (对于真正二次的代价函数,存在 $\epsilon^*$ 的解析解,而无需显式地搜索) 应用更新: $\theta_{t+1} = \theta_t + \epsilon^* \rho_t$ $t \leftarrow t + 1$	$\overline{\mathrm{Re}}$	$\mathbf{quire}$ : 初始参数 $\boldsymbol{\theta}_0$
初始化 $g_0 = 0$ 初始化 $t = 1$ while 没有达到停止准则 do 初始化梯度 $g_t = 0$ 计算梯度: $g_t \leftarrow \frac{1}{m_t} \nabla_{\theta} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$ 计算 $\beta_t = \frac{(g_t - g_{t-1})}{g_{t-1}^T g_{t-1}}$ (Polak-Ribière) (非线性共轭梯度:视情况可重置 $\beta_t$ 为零,例如 $t$ 是常数 $k$ 的倍数时,如 $k = 5$ ) 计算搜索方向: $\rho_t = -g_t + \beta_t \rho_{t-1}$ 执行线搜索寻找: $\epsilon^* = \operatorname{argmin}_{\epsilon} \frac{1}{m} \sum_{i=1}^m L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}_t + \epsilon \rho_t), \boldsymbol{y}^{(i)})$ 文介 (对于真正二次的代价函数,存在 $\epsilon^*$ 的解析解,而无需显式地搜索) 应用更新: $\theta_{t+1} = \theta_t + \epsilon^* \rho_t$ $t \leftarrow t + 1$	Re	$\mathbf{quire:}$ 包含 $m$ 个样本的训练集
初始化 $t=1$ while 没有达到停止准则 do 初始化梯度 $g_t=0$ 计算梯度: $g_t \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)};\boldsymbol{\theta}),\boldsymbol{y}^{(i)})$ 计算 $\beta_t = \frac{(g_t-g_{t-1})^\top g_t}{g_{t-1}^\top g_{t-1}}$ (Polak-Ribière) (非线性共轭梯度: 视情况可重置 $\beta_t$ 为零,例如 $t$ 是常数 $k$ 的倍数时,如 $k=5$ ) 计算搜索方向: $\rho_t = -g_t + \beta_t \rho_{t-1}$ 执行线搜索寻找: $\epsilon^* = \operatorname{argmin}_{\epsilon} \frac{1}{m} \sum_{i=1}^m L(f(\boldsymbol{x}^{(i)};\boldsymbol{\theta}_t + \epsilon \rho_t),\boldsymbol{y}^{(i)})$ 受了 (对于真正二次的代价函数,存在 $\epsilon^*$ 的解析解,而无需显式地搜索) 应用更新: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \epsilon^* \rho_t$ $t \leftarrow t+1$	衣	刀始化 $ ho_0=0$
while 没有达到停止准则 do 初始化梯度 $g_t = 0$ 计算梯度: $g_t \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$ 计算 $\beta_t = \frac{(g_t - g_{t-1})^\top g_t}{g_{t-1}^\top g_{t-1}}$ (Polak-Ribière) (非线性共轭梯度:视情况可重置 $\beta_t$ 为零,例如 $t$ 是常数 $k$ 的倍数时,如 $k = 5$ ) 计算搜索方向: $\rho_t = -g_t + \beta_t \rho_{t-1}$ 执行线搜索寻找: $\epsilon^* = \operatorname{argmin}_{\epsilon} \frac{1}{m} \sum_{i=1}^m L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}_t + \epsilon \rho_t), \boldsymbol{y}^{(i)})$ (对于真正二次的代价函数,存在 $\epsilon^*$ 的解析解,而无需显式地搜索) 应用更新: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \epsilon^* \rho_t$ $t \leftarrow t+1$	衣	刀始化 $g_0=0$
初始化梯度 $g_t = 0$ 计算梯度: $g_t \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$ 计算 $\beta_t = \frac{(g_t - g_{t-1})^{\top} g_t}{g_{t-1}^{\top} g_{t-1}}$ (Polak-Ribière) (非线性共轭梯度: 视情况可重置 $\beta_t$ 为零,例如 $t$ 是常数 $k$ 的倍数时,如 $k = 5$ ) 计算搜索方向: $\rho_t = -g_t + \beta_t \rho_{t-1}$ 执行线搜索寻找: $\epsilon^* = \operatorname{argmin}_{\epsilon} \frac{1}{m} \sum_{i=1}^m L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}_t + \epsilon \rho_t), \boldsymbol{y}^{(i)})$ 承知 (对于真正二次的代价函数,存在 $\epsilon^*$ 的解析解,而无需显式地搜索) 应用更新: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \epsilon^* \rho_t$ $t \leftarrow t + 1$	衣	刀始化 $t=1$
计算梯度: $g_t \leftarrow \frac{1}{m_t} \nabla_{\theta} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$ 计算 $\beta_t = \frac{(g_t - g_{t-1})^\top g_t}{g_{t-1}^\top g_{t-1}}$ (Polak-Ribière) (非线性共轭梯度: 视情况可重置 $\beta_t$ 为零,例如 $t$ 是常数 $k$ 的倍数时,如 $k = 5$ ) 计算搜索方向: $\rho_t = -g_t + \beta_t \rho_{t-1}$ 执行线搜索寻找: $\epsilon^* = \operatorname{argmin}_{\epsilon} \frac{1}{m} \sum_{i=1}^m L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}_t + \epsilon \boldsymbol{\rho}_t), \boldsymbol{y}^{(i)})$ 贪了 (对于真正二次的代价函数,存在 $\epsilon^*$ 的解析解,而无需显式地搜索) 应用更新: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \epsilon^* \boldsymbol{\rho}_t$ $t \leftarrow t+1$	v	vhile 没有达到停止准则 do
计算 $\beta_t = \frac{(g_t - g_{t-1})^\top g_t}{g_{t-1}^\top g_{t-1}}$ (Polak-Ribière) (非线性共轭梯度: 视情况可重置 $\beta_t$ 为零,例如 $t$ 是常数 $k$ 的倍数时,如 $k = 5$ ) 计算搜索方向: $\rho_t = -g_t + \beta_t \rho_{t-1}$ 执行线搜索寻找: $\epsilon^* = \operatorname{argmin}_{\epsilon} \frac{1}{m} \sum_{i=1}^m L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}_t + \epsilon \boldsymbol{\rho}_t), \boldsymbol{y}^{(i)})$ 更为 (对于真正二次的代价函数,存在 $\epsilon^*$ 的解析解,而无需显式地搜索) 应用更新: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \epsilon^* \boldsymbol{\rho}_t$ $t \leftarrow t+1$		初始化梯度 $g_t=0$
计算 $\beta_t = \frac{(g_t - g_{t-1})^\top g_t}{g_{t-1}^\top g_{t-1}}$ (Polak-Ribière) (非线性共轭梯度: 视情况可重置 $\beta_t$ 为零,例如 $t$ 是常数 $k$ 的倍数时,如 $k = 5$ ) 计算搜索方向: $\rho_t = -g_t + \beta_t \rho_{t-1}$ 执行线搜索寻找: $\epsilon^* = \operatorname{argmin}_{\epsilon} \frac{1}{m} \sum_{i=1}^m L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}_t + \epsilon \boldsymbol{\rho}_t), \boldsymbol{y}^{(i)})$ 文介 (对于真正二次的代价函数,存在 $\epsilon^*$ 的解析解,而无需显式地搜索 ) 应用更新: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \epsilon^* \boldsymbol{\rho}_t$ $t \leftarrow t+1$		计算梯度: $g_t \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$
(非线性共轭梯度: 视情况可重置 $\beta_t$ 为零,例如 $t$ 是常数 $k$ 的倍数时,如 $k=5$ ) 计算搜索方向: $\rho_t = -g_t + \beta_t \rho_{t-1}$ 执行线搜索寻找: $\epsilon^* = \operatorname{argmin}_{\epsilon} \frac{1}{m} \sum_{i=1}^m L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}_t + \epsilon \boldsymbol{\rho}_t), \boldsymbol{y}^{(i)})$ (对于真正二次的代价函数,存在 $\epsilon^*$ 的解析解,而无需显式地搜索) 应用更新: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \epsilon^* \boldsymbol{\rho}_t$ $t \leftarrow t+1$		计算 $\beta_t = \frac{(g_t - g_{t-1})^\top g_t}{g_t^\top g_{t-1}}$ (Polak-Ribière)
执行线搜索寻找: $\epsilon^* = \operatorname{argmin}_{\epsilon} \frac{1}{m} \sum_{i=1}^m L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}_t + \epsilon \boldsymbol{\rho}_t), \boldsymbol{y}^{(i)})$ (对于真正二次的代价函数,存在 $\epsilon^*$ 的解析解,而无需显式地搜索 ) 应用更新: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \epsilon^* \boldsymbol{\rho}_t$ $t \leftarrow t+1$		(非线性共轭梯度: 视情况可重置 $\beta_t$ 为零, 例如 $t$ 是常数 $k$ 的倍数时, 如 $k=5$ )
执行线搜索寻找: $\epsilon^* = \operatorname{argmin}_{\epsilon} \frac{1}{m} \sum_{i=1}^m L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}_t + \epsilon \boldsymbol{\rho}_t), \boldsymbol{y}^{(i)})$ (对于真正二次的代价函数,存在 $\epsilon^*$ 的解析解,而无需显式地搜索 ) 应用更新: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \epsilon^* \boldsymbol{\rho}_t$ $t \leftarrow t+1$		计算搜索方向: $\rho_t = -g_t + \beta_t \rho_{t-1}$
(对于真正二次的代价函数,存在 $\epsilon^*$ 的解析解,而无需显式地搜索) 应用更新: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \epsilon^* \boldsymbol{\rho}_t$ $t \leftarrow t+1$		
$t \leftarrow t+1$		
$t \leftarrow t + 1$		应用更新: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \epsilon^* \boldsymbol{\rho}_t$
end while		
	e	and while
	2	



