**AdaGrad 算法**，如算法8.4所示，独立地适应所有模型参数的学习速率，放缩每个参数反比于其所有梯度历史平方值总和的平方根 (Duchi *et al.*, 2011)。具有损失最大偏导的参数相应地有一个快速下降的学习速率，而具有小偏导的参数在学习速率上有相对较小的下降。净效果是在参数空间中更为平缓的倾斜方向会取得更大的进步。

**算法 8.4** AdaGrad 算法

**Require:** 全局学习速率 $\epsilon$

**Require:** 初始参数 $\boldsymbol{\theta}$

**Require:** 小常数 $\delta$，为了数值稳定大约设为 $10^{-7}$

  初始化梯度累积变量 $\boldsymbol{r} = 0$

  **while** 没有达到停止准则 **do**

    从训练集中采包含 $m$ 个样本 $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ 的minibatch，对应目标为 $\boldsymbol{y}^{(i)}$。

    计算梯度：$\boldsymbol{g} \leftarrow \frac{1}{m}\nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$

    累积平方梯度：$\boldsymbol{r} \leftarrow \boldsymbol{r} + \boldsymbol{g} \odot \boldsymbol{g}$

    计算更新：$\Delta\boldsymbol{\theta} \leftarrow -\frac{\epsilon}{\delta+\sqrt{\boldsymbol{r}}} \odot \boldsymbol{g}$  （逐元素地应用除和求平方根）

    应用更新：$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$

  **end while**