8.6.1 牛顿法

优势：

在第4.3节，我们介绍了二阶梯度方法。① 与一阶方法相比，二阶方法使用二阶导数改进了优化。最广泛使用的二阶方法是牛顿法。我们现在更详细地描述牛顿法，重点在其应用于神经网络的训练。

牛顿法?是基于二阶泰勒级数展开在某点 $\boldsymbol{\theta}_0$ 附近来近似 $J(\boldsymbol{\theta})$ 的优化方法，其忽略了高阶导数：

θ_0　　一阶　　二阶

① 泰勒展开：

$$J(\boldsymbol{\theta}) \approx J(\boldsymbol{\theta}_0) + (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \boldsymbol{H}(\boldsymbol{\theta} - \boldsymbol{\theta}_0), \tag{8.26}$$

其中 $\boldsymbol{H}$ 是 J 相对于 $\boldsymbol{\theta}$ 的Hessian矩阵在 $\boldsymbol{\theta}_0$ 处的估计。如果我们再求解这个函数的临界点，我们将得到牛顿参数更新规则：

更新参数　②

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}_0 - \boldsymbol{H}^{-1} \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0). \tag{8.27}$$

算法 8.8 目标为 $J(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} L(f(\boldsymbol{x}^{(i)};\boldsymbol{\theta}), y^{(i)})$ 的牛顿法

Require: 初始参数 $\boldsymbol{\theta}_0$

Require: 包含 m 个样本的训练集

while 没有达到停止准则 **do**

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

计算Hessian：$\boldsymbol{H} \leftarrow \frac{1}{m}\nabla^2_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)};\boldsymbol{\theta}), \boldsymbol{y}^{(i)})$

计算Hessian逆：$\boldsymbol{H}^{-1}$　　求逆过程.

计算更新：$\Delta\boldsymbol{\theta} = -\boldsymbol{H}^{-1}\boldsymbol{g}$　　变化：二阶矩阵与一阶矩阵结合

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

end while
