牛顿法限制？

式(8.27)更新参数。

①

在第8.2.3节，我们讨论了牛顿法只适用于Hessian是正定的情况。在深度学习中，目标函数的表面通常具有非凸的很多特点，如鞍点。因此使用牛顿法是有问题的。如果Hessian矩阵的特征值并不都是正的，例如，靠近鞍点处，牛顿法实际上会导致更新朝错误的方向移动。这种情况可以通过正则化Hessian来避免。常用的正则化策略包括在Hessian矩阵对角线上增加常数 α。正则化更新变为

非凸目标函数.

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}_0 - [H(f(\boldsymbol{\theta}_0)) + \alpha \boldsymbol{I}]^{-1} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_0). \tag{8.28}$$

②

除了目标函数的某些特征带来的挑战，如鞍点，牛顿法用于训练大型神经网络还受限于其显著的计算负担。Hessian矩阵中元素数目是参数数量的平方，因此，如果参数数目为 k（甚至是在非常小的神经网络中 k 也可能是百万级别），牛顿法需要计算 $k \times k$ 矩阵的逆，计算复杂度为 $O(k^3)$。另外，由于参数将每次更新都会改变，每次训练迭代都需要计算Hessian矩阵的逆。其结果是，只有参数很少的网络才能在实