全局最小与局部极小

前言

上一篇文章我们介绍到了 $E=\lambda\frac{1}{m}\sum_{k=1}^{m}E_k+(1-\lambda)\sum_i w_i^2$, （5.17） 这个公式，但是问题就出在这里。若用 E 表示神经网络在训练集上的误差，则它显然是关于连接权 w 和阈值 θ 的函数。此时，神经网络的训练可以看作一个参数寻优过程，即在参数空间中，寻找一组最优参数使得 E 最小。

我们常谈到两种“最优”，一个是“局部最优”(local minimum)和“全局最小”(global minimum)。对 w^* 和 θ^* ，若存在 $\epsilon>0$ 使得

$$\forall(\mathbf{w};\theta)\subseteq\{(\mathbf{w};\theta)|\|(\mathbf{w};\theta)-(\mathbf{w}^*;\theta^*)\|\leq\epsilon\},$$

都有 $E(\mathbf{w};\theta)\geq E(\mathbf{w}^*;\theta^*)$ 成立，则 $(\mathbf{w}^*;\theta^*)$ 为局部极小解；若对参数空间中的任意 $(\mathbf{w};\theta)$ 都有 $(\mathbf{w};\theta)\geq(\mathbf{w}^*;\theta^*)$ ，则 $(\mathbf{w}^*;\theta^*)$ 为全局最小解。直观地看，局部极小解是参数空间中的某个点，其邻域点的误差函数均不小于该点的误差函数值，两者对应的 $E(\mathbf{w}^*;\theta^*)$ 分别称为误差函数的局部极小值和全局最小值。

显然，参数空间内梯度为零的点，只要其误差函数值小于邻点的误差函数值，就是局部极小点；可能存在多个局部极小值，但却只会有一个全局最小值。也就是说，“全局最小”一定是“局部极小”，反之则不成立，例如，下图中有两个局部极小，但只有其中之一是全局最小，显然，我们在参数寻优过程中是希望找到全局最小。

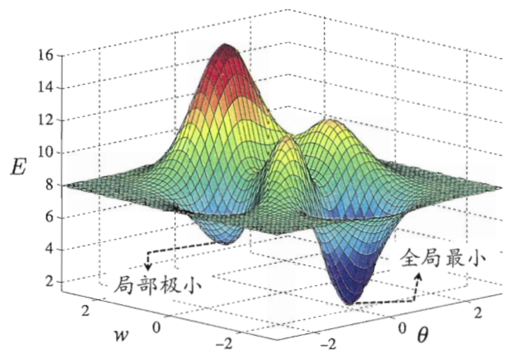

图 5.10 全局最小与局部极小

如何解决梯度下降局部极小

基于梯度的搜索是使用最为广泛的参数寻优方法（感知机更新规则式 (5.1) 和BP 更新规则式 (5.11)- (5.14) 都是基于梯度下降。）。在此类方法中，我们从某些初始解出发，迭代寻找最优参数值。每次迭代中，我们先计算误差函数在当前点的梯度，然后根据梯度确定搜索方向。例如，由于负梯度方向是函数值下降最快的方向，因此梯度下降法就是沿着负梯度方向搜索最优解。若误差函数在当前点的梯度为零，则已达到局部极小，更新量将为零，这意味着参数的迭代更新将在此停止。显

然，如果误差函数仅有一个局部极小，那么此时找到的局部极小就是全局最小；然而，如果误差函数具有多个局部极小，则不能保证找到的解是全局最小。对后一种情形，我们称参数寻优陷入了局部极小，这显然不是我们所希望的。

在现实任务中，人们常采用以下策略来试图“跳出”局部极小，从而进步接近全局最小：

- 以多组不同参数值初始化多个神经网络，按标准方法训练后，取其中误差最小的解作为最终参数。这相当于从多个不同的初始点开始搜索，这样就可能陷入不同的局部极小，从中进行选择有可能获得更接近全局最小的结果。
- 使用“模拟退火”(simulated annealing)技术[Aarts and Korst,1989]。模拟退火在每一步都以一定的概率接受比当前解更差的结果，从而有助于“跳出”局部极小。在每步迭代过程中，接受“次优解”的概率要随着时间的推移而逐渐降低，从而保证算法稳定。
- 使用随机梯度下降。与标准梯度下降法精确计算梯度不同,随机梯度下降法在计算梯度时加入了随机因素。于是，即便陷入局部极小点，它计算出的梯度仍可能不为零，这样就有机会跳出局部极小继续搜索。

此外，遗传算法(genetic algorithms)[Goldberg,1989]也常用来训练神经网络以更好地逼近全局最小。需注意的是，上述用于跳出局部极小的技术大多是启发式，理论上尚缺乏保障。