|  |  |  |  |
| --- | --- | --- | --- |
|  | 更新规则 | 说明 | 更新公式 |
| BGD | 在整个训练集上计算LOSS对参数的梯度 | 计算比较慢  不能投入新数据实时更新模型  凸函数:可以收敛到全局最小  非凸函数:可以收敛到局部最小 |  |
| SGD | 对每个样本计算LOSS对参数的梯度 | 噪音比较多,不是每次都是整体最优  更新比较频繁,LOSS振荡  可以新增样本  可收敛到局部最小 |  |
| MBGD | 对一小批样本计算LOSS对参数的梯度 | 收敛更稳定,但不能保证很好的收敛  非凸函数:鞍点或极小值振荡跳动  先设置大一点的LR,两次迭代变化低于某个值,就减小LR |  |
| Momentum | SGD+动量规则 | 加快收敛减小振荡  适应性不好 |  |
| NAG | 在前一步累计梯度上一个大的跳跃,在衡量一下做一下修正 | 计算梯度不是当前位置,而是在未来  使得RNN在很多任务上有更好的表现  在Momentum上更新方向都是走一小步 |  |
| Adagrad | 相当于增加了一个学习率递减系数  反向梯度来决定学习率衰减多少 | 对低频的参数做较大的更新,高频的做较小的更新  稀疏的数据表现很好,提高了SGD的robust  较少了学习率的手动调节  学习率会收缩最终变得非常小 |  |
| Adadelta |  |  |  |
| RMSprop | 考察近几步的反向梯度来决定学习率衰减 | 解决Adagrad学习率下降的问题 |  |
| Adam | RMSprop+Momentum |  |  |