RMSProp算法

1. 产生背景

在AdaGrad算法中，当学习率在迭代早期下降较快且解依然不佳时，AdaGrad算法在迭代后去由于学习率过小，可能较难找到一个有用的解。

1. 算法介绍

不同于AdaGrad算法中状态变量是截至时间步t所有小批量随机梯度按原始平方和，RMSProp苏纳法将这些梯度按元素平方做指数加权移动平均，即：

给定超参数,RMSProp算法在时间步t>0计算：

和AdaGrad算法一样，RMSProp算法将目标函数自变量中每个严肃的学习率按元素运算重新调整，然后更新自变量：

其中是学习率，是维持数值稳定性的较小常熟，如。因为RMSProp算法的状态变量是对平方项 的指数加权移动平均，所以可以看作是最近个时间步的小批量随机梯度平方的加权平均。这样就可以使得每个元素的学习率在迭代过程中不再一直降低（或不变）。

代码实现：



在同样学习率下，RMSProp算法可以更快逼近最优解。

1. 特点

RMSProp算法和AdaGrad算法的不同在于，RMSProp算法使用了小批量随机梯度按元素平方的指数加权移动平均来调整学习率