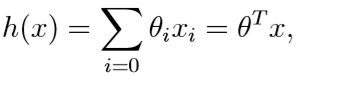
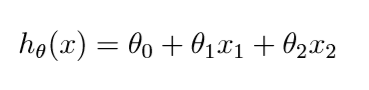
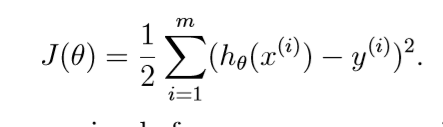
由于老是忘掉神经网络中得反向传播，因此在这里做一个记录

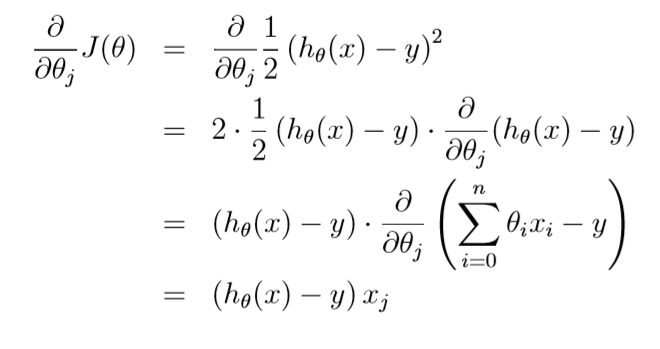
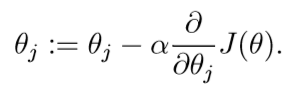
梯度下降：



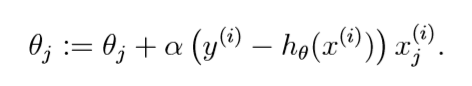
损失函数选择平方差：



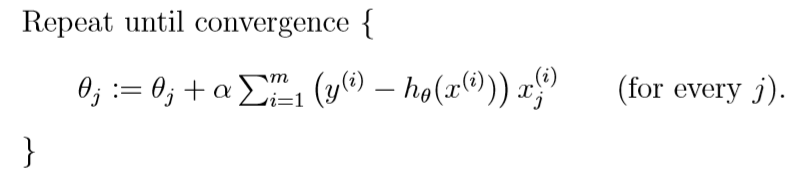
更新参数：



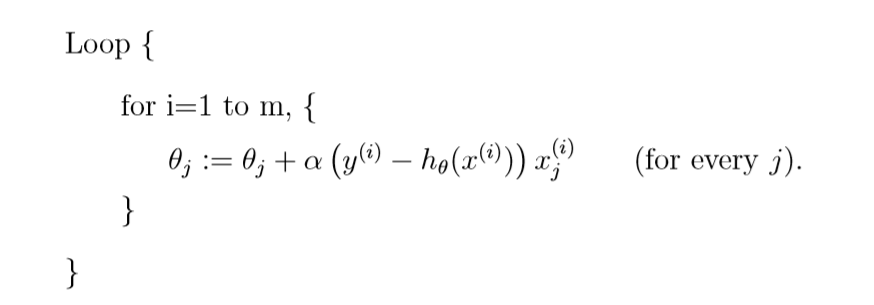
于是对每一个样本更新参数J：



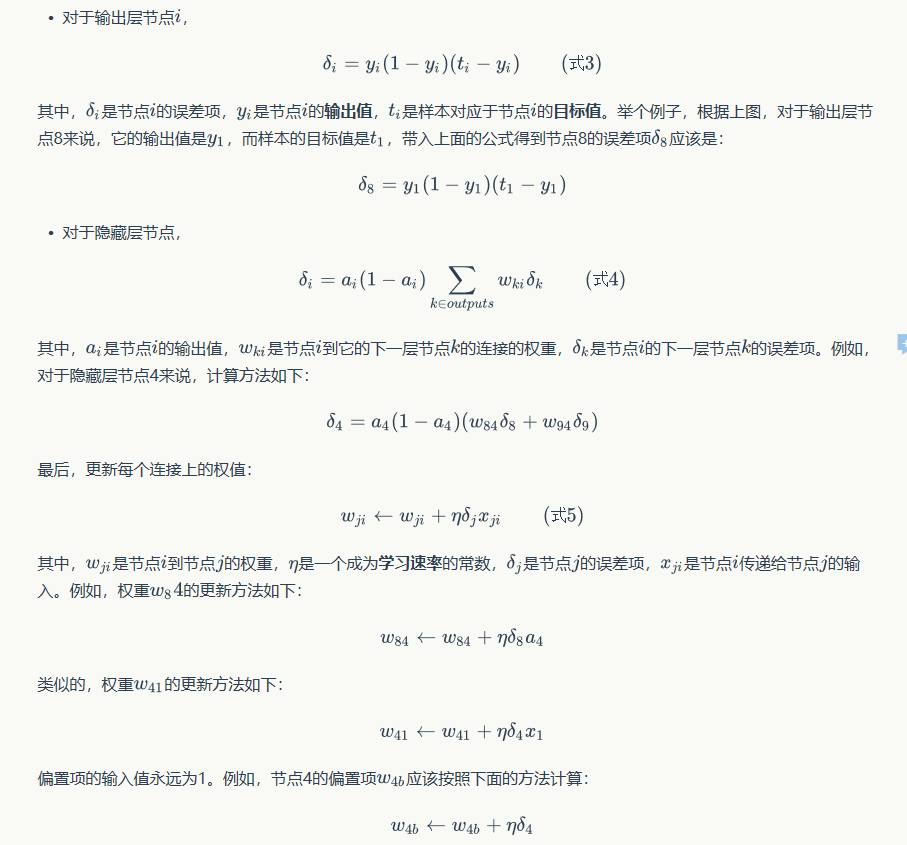
批量梯度下降：每次更新θ的迭代，要遍历训练数据中所有的样本进行计算，i表示样本编号



随机梯度下降：每次更新的迭代，只计算一个样本。这样对于一个具有数百万样本的训练数据，完成一次遍历就会对更新数百万次，效率大大提升



反向传播：



其中yi(1-yi)，ai(ai-1),是激活函数的导数值，如果没有激活函数，误差项就是