

对反向传播算法的个人理解

目标

通过训练，找出输出与输入之间的联系，使得新的输入数据经过BP网的计算结果尽可能接近我们希望的输出

一、初始化

设置输入层、输出层、各隐藏层的神经元个数，随机设置各神经元到下层神经元的权重

二、向前计算输出

1. 输入层→隐藏层：计算输入层的加权和，使用激活函数，得到第一个隐藏层的节点值。
2. 隐藏层→隐藏层：计算隐藏层的加权和，使用激活函数，得到下一个隐藏层的节点值
3. 隐藏层→输出层：计算隐藏层的加权和，使用激活函数，得到输出层的节点值
4. 向前计算结束，由于初始权重完全随机，此时的输出层结果往往与期望值相差很大，需要反向传播修改权重

三、反向传播

1. 使用 $Error = (1/2) * \sum (d_i - o_i)^2$ 公式计算各输出层节点误差，求和得到总误差
2. 从最后一层开始，逐层用总误差对各个权重求偏导，得到每个权重对总误差的影响，进而进行权重调整（在原有权重的基础上减去上述偏导与learning rate的乘积）。
learning rate设置不宜过大，这会导致网络调整幅度过大，网络出现波动；也不宜设置过小，这会使网络收敛速度过慢
3. 权值全部更新后进行下次迭代的训练，经过多次训练后各层间的权重可以较为准确地抽象出输入层特征，并传播到输出层，得到较为精确到结果