第一页

大家好，我是胡老师！ 上一节课我们讲到了，神经网络的训练过程是基于梯度下降的方式来更新网络的权重系数的。那么什么是梯度下降呢？我们先来复习一下高等数学的知识。首先，我们来看看导数的定义。X0处的导数为\*\*\*。根据公式，我们知道，导数就是指函数在某个区间的变化率，针对函数中的某个特定点x0，该点的导数就是x0点的切线的斜率。也就是这条红色直线的斜率。这个斜率越大，就表明其上升趋势越强劲。

第二页

好，现在了解了导数。我们再来看看梯度，为了表达形象，我们以2维函数为例子。梯度为函数在x轴维度的偏导数与y轴维度的偏导数组成的向量，它是一个矢量，表达的是一个方向。梯度的方向就代表着函数增长速度最快的方向，那么很显然，如果要找到目标函数最小值，就沿着梯度的反方向来走，梯度的反方向就是函数减小的速度最快方向。

第三页

现在我们再来看看神经网络，神经网络实际上是一个复杂的复合函数，对于每一组输入数据，神经网络会给出对应的输出数据，该输出数据与输入数据的期望输出不一定相同，因此我们可以构建一个损失函数来表达网络输出与期望输出之间的差异，输出数据对应的损失函数的值小于一定的阈值时,我们认为网络已经学习到了输入数据的特征，能给出一个接近输入数据的期望输出值。 也就是说，神经网络训练过程是尽可能得使损失函数的值最小。因此，为了尽可能的找到损失函数的最小值，我们运用梯度下降的方式来寻找，并在梯度下降过程中不断更新网络的权重系数。

第五页

这里，我们通过一个小例子来学习神经网络的训练过程。这里有一个只有一层隐藏层的神经网络，其输入为二维向量，输出为一维向量，也就是一个标量数值。现在呢，我们从数据集中取出一个样本数据，该数据两个维度的值分别为0.5和1.0，该数据对应的标签值为0.8，也就是说，训练好的网络在以这个数据为输入时，其应该输出一个十分接近0.8的数值。开始的时候，网络中的参数是随机的。

第六页

这里，我们通过一个小例子来学习神经网络的训练过程。这里有一个只有一层隐藏层的神经网络，其输入为二维向量，输出为一维向量，也就是一个标量数值。现在呢，我们从数据集中取出一个样本数据，该数据两个维度的值分别为0.5和1.0，该数据对应的标签值为0.8，也就是说，训练好的网络在以这个数据为输入时，其应该输出一个十分接近0.8的数值。开始的时候，网络中的参数是随机的。

第七页

接着呢， 我们来计算第一个神经元的值，为了简洁，我们这里省略了神经元的偏置和激活函数，如图所示，第一个神经元的输出为（读公式）

第九页

接着，我们计算第二个神经元的输出，第二个神经元的输出为(读公式)

第十页

然后，我们计算最后一个神经元的输出，最后一个神经元的输出为(读公式)

在得到了网络的输出数值之后，我们就可以计算输出值与真实值之间的误差，即损失函数的值。此处，我们用一个平方损失来衡量这个误差，可以得到，损失函数的数值为2.205.

得到损失函数值之后，我们基于梯度下降来更新网络的参数。我们先来求解w5的梯度。基于复合函数的链式求导法则，我们可以知道，（读公式）。

根据高等数学知识，我们知道，沿着梯度的反方向，是下降最快的方向，能够最快的找到我们的最小值，即最优解。因此，我们得到更新的权重为(读公式)。此处的eta是学习率，用来控制参数更新的步长，步长太小，会使得网络的学习速度变慢，步长太大又会导致网络没法收敛。学习率通常更加经验来自行设定。用相同的方法，我们可以更新得到更新后的w6数值。

接着，我们来计算w1的梯度，根据链式法则，（读公式）

同理，(读公式)，可得更新后的w1数值为0.895。用相同的方法，我们可以更新得到更新后的w2，w3，w4的数值。

到这里，我们的参数都更新完成了，可以看到，每一个权重都有了变化。

在权重更新之后，我们再进行第二轮的正向传播，首先得到，h1为\*\*，h2为\*\*，从而得到网络的输出值为1.3478。网络的期望值是0.8，因而参数更新之后网络的损失值为0.15。可以看到，该值小于第一轮的损失函数值。实际应用时，当网络的损失值小于某个约定的阈值时，我们认为神经网络训练完成，其已经充分学习到了数据集中的数据特征。

由以上示例可以看见，神经网络训练时，先是读入数据，计算每个神经元的输出，进而得到整个网络的输出，接着计算输出值和期望值之间的误差，即损失函数，在此基础上，计算每个权重系数的梯度，并更新权重数值。通过这种方式，我们可以使得网络输出不断的去逼近一个真实值，当损失小到一定程度后，我们就认为网络收敛了。

第十一页

最后，我会将本教程的ppt及代码放在github上，欢迎有需要的同学自取！！