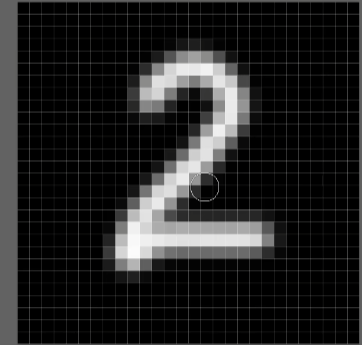
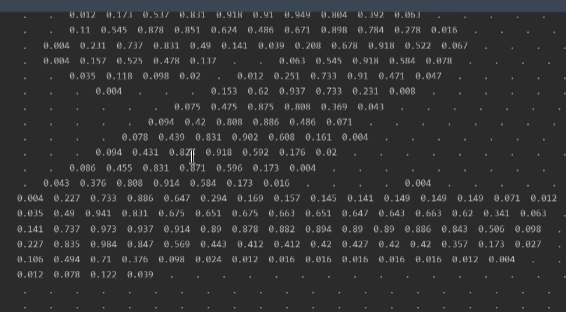
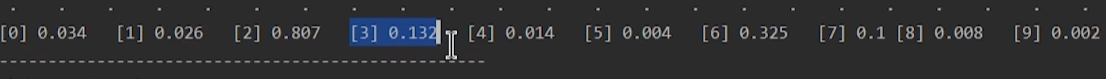
图片为2





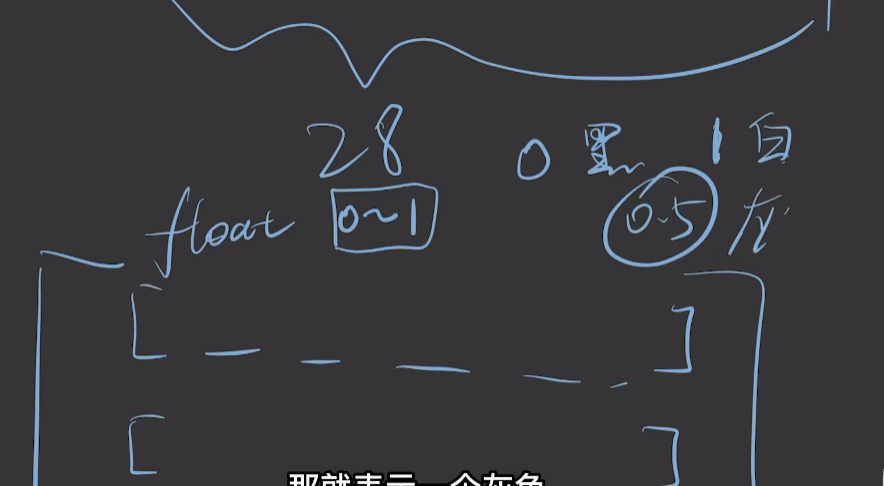
训练数据



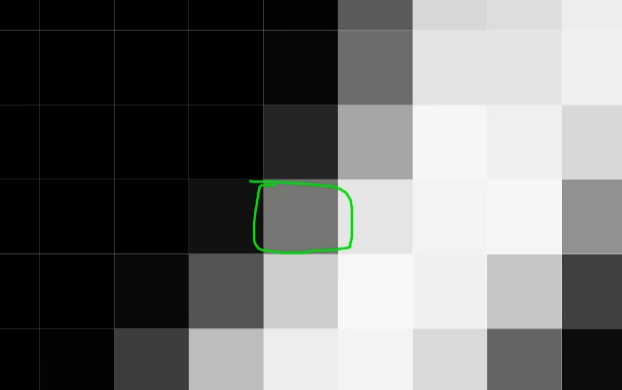


第二集视频

图片转成阵的表示法



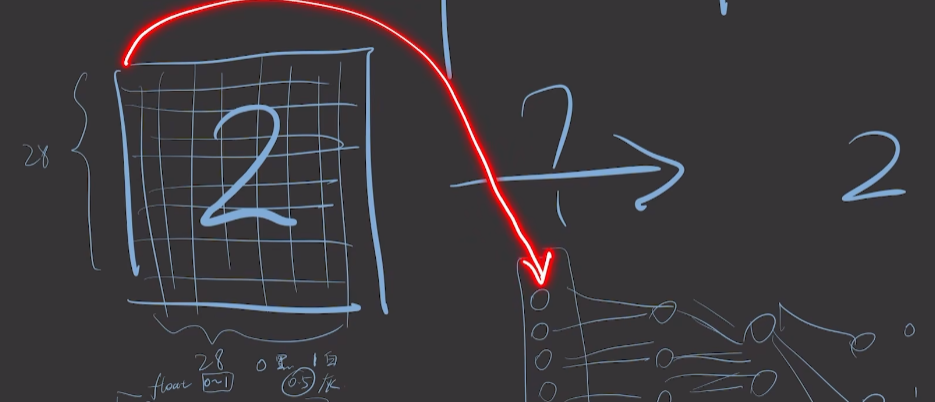
例如0.5代表下面的灰色



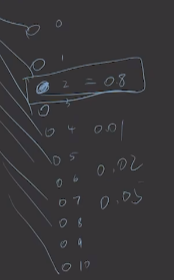
每个像素点刚刚好对应着么一个神经元，2400多个



传递0过去神经元

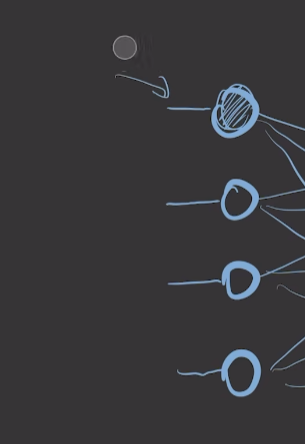


当输入2，我们就发现这个神经元最后的神经元格外的亮

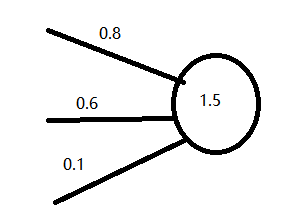


通过最后一层神经元的亮度，就知道结果是什么了

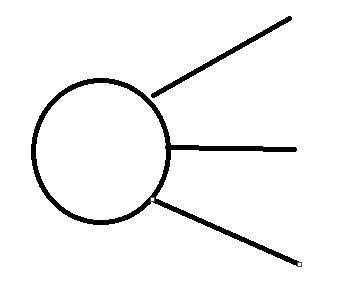
输入



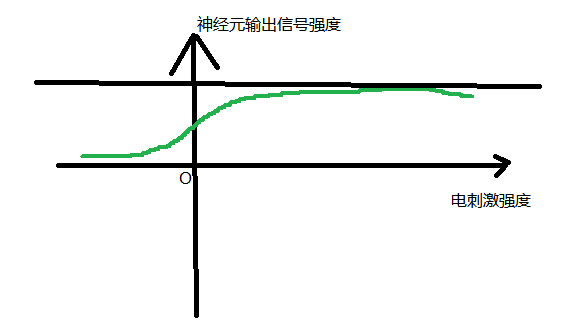
所有的网络都是由这种东西组成的



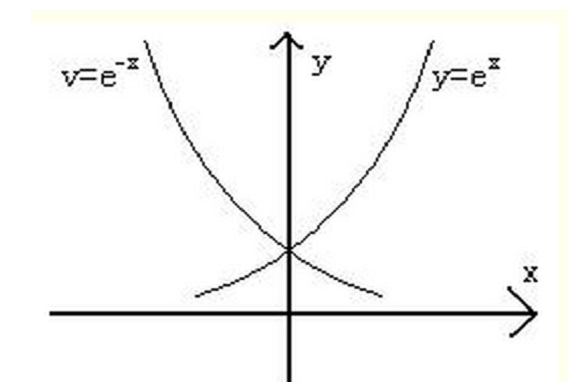
不是



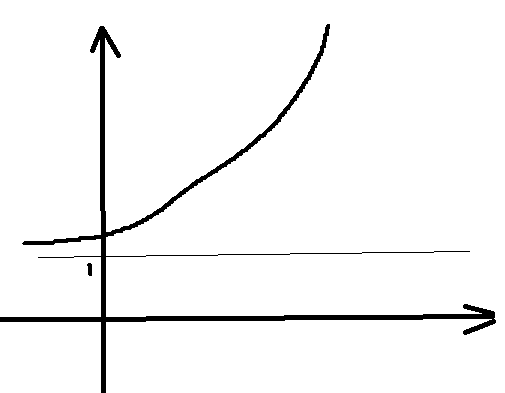
一般电信号输出函数是这个样子的，从左到右，当最左边开始，强度没啥变化，忽然间变化大大增加，继续加大，又没什么变化了，这个函数叫做sigmoid函数

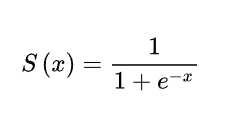


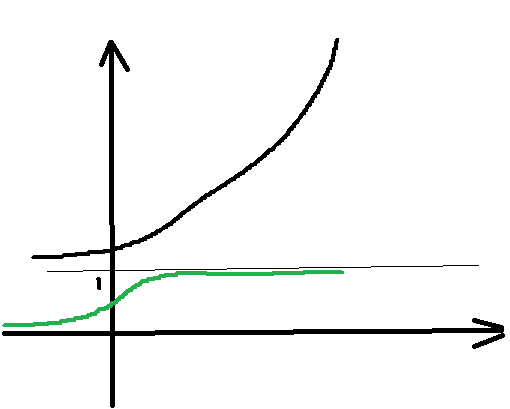
指数函数e的x次方以及e的负x次



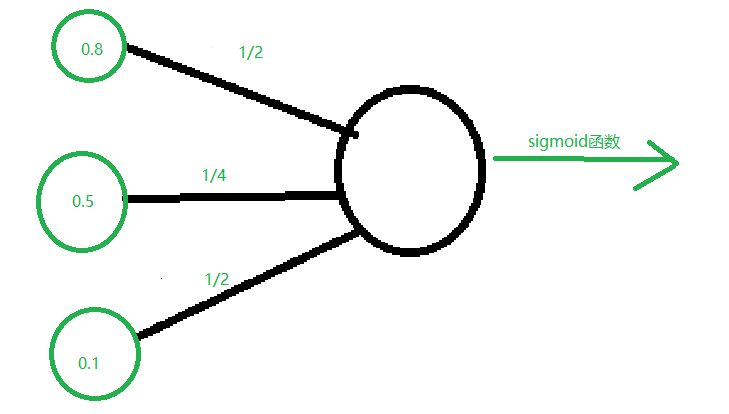
加上一个1



继续取倒数，就变成了

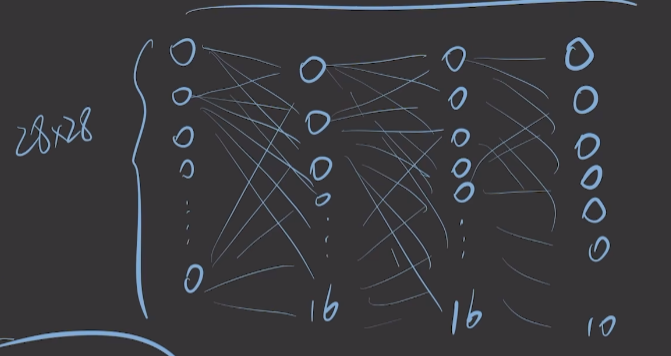


所以我们的神经网络变成了这个样子



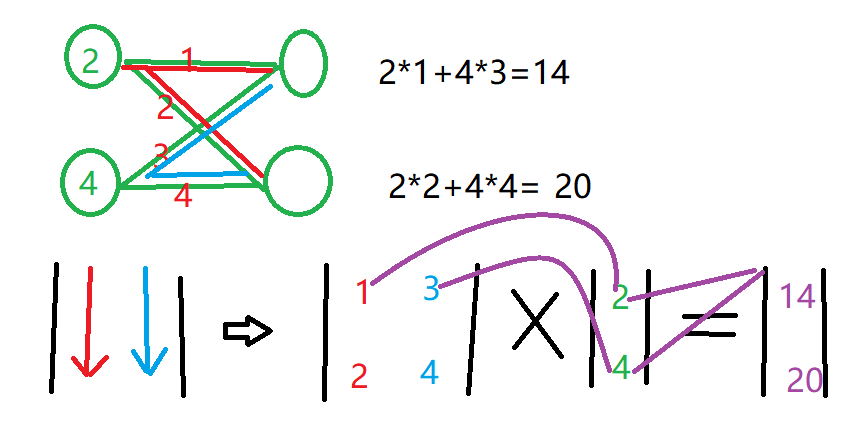
图示28\*28的

如果是下面的神经网络结构就是一共有12960条线

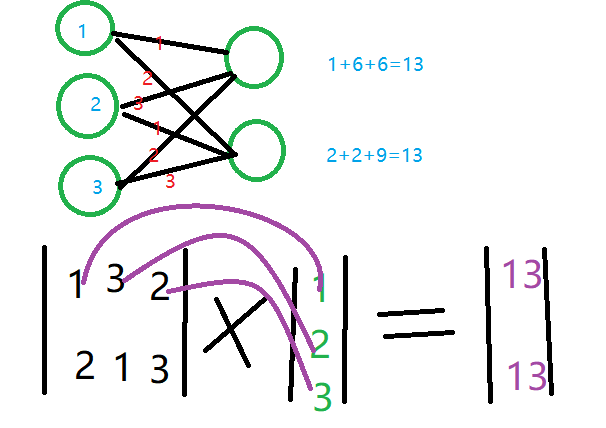


# 第2集 正向传播的矩阵乘法表示

数据是怎么从左往右传播的，刚刚好高数的矩阵乘法可以拟合这一点

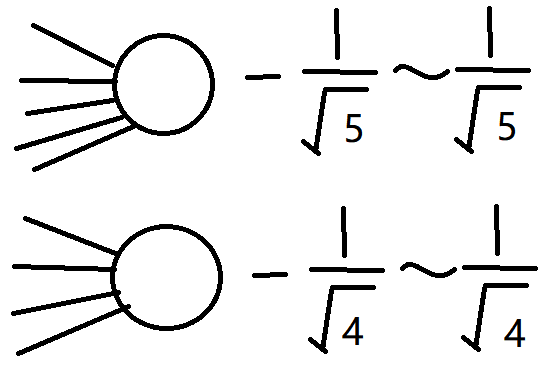


继续来一张图



# 第3集 如何自动调整单条权重？

一般如何设置权重默认值，不至于让某个权重太大，导致某个圈太亮



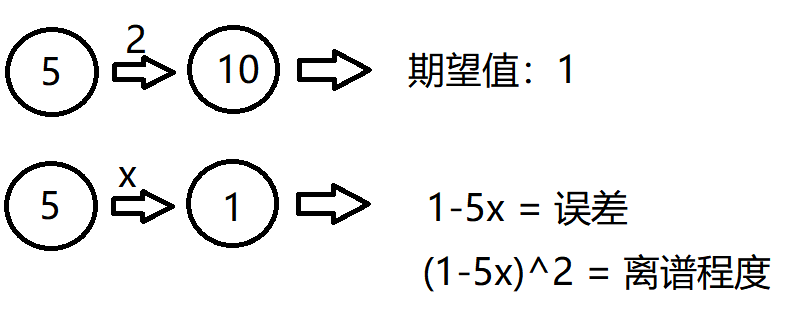
如何调整我们的权重呢？

每个权重变大变小的丽都是不一样的

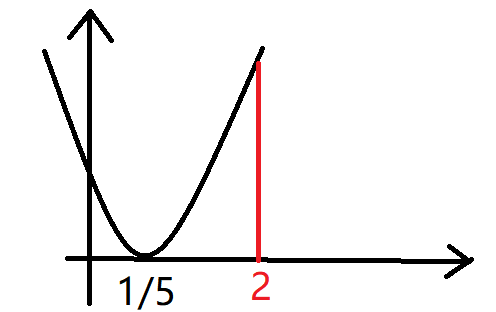
0.2是实际值，0是预期值



然后不断向后传播，得出上一层的一个神经元都有一个误差亮度，继续向后调整，然后就可以得出每一个神经元的误差亮度。



得出离谱程度函数，那么我们就知道x1/5了，y=0,就是误差最小，当x=2的时候，误差还是挺大的，不能直接移动到1/5，否则权重发生巨变。

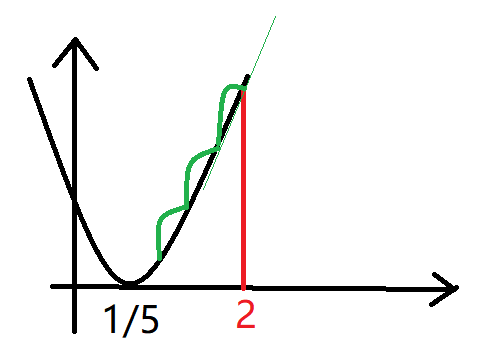


但是我们不能直接就将权重设置成0.2，我们需要一步一步的微调，例如上图x=2的时候，我们要将x一点点往左移，如果我们直接一步到位设置0.2会有什么后果呢？

因为我们的训练数据是这个样子的，第一次训练2，第二次可能训练是3，第三次训练可能是4，第四次训练可能是2，如果我们每一次都一步到位，那么神经元就会把之前训练的数据都忘掉了，我们不能训练一个数据就发生一次巨变，我们是训练权重一点点拟合，贴合。

那么我们不能一下子从2跳到0.2，那么我们该设置怎么样的步伐呢，我们可以依据离谱程度函数的2对应点的斜率，然后2--斜率=步伐。

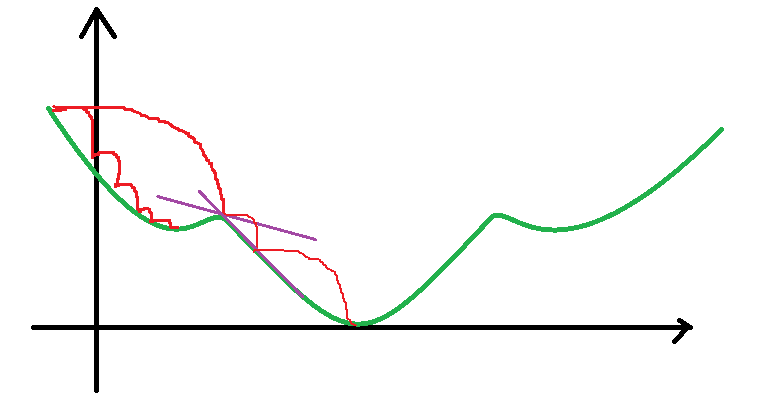
为什么不能直接变成最低点，如果只有一个点训练数据的话，一点就可以，但是我们是有很多数据的。



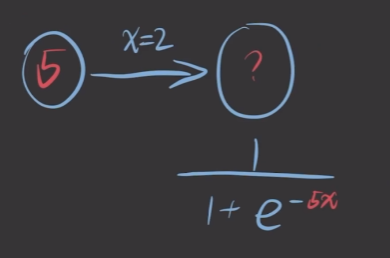
但是步伐不只是简单的1-斜率=步伐，我们要先求出导数(1-5x)^2的导数为-10(1-5x)。

代入2，就可以得出在2的斜率为k=90，数字太大了，如果我们直接2-90=-88，一下子x=-88，步子跨得太大，导致无法到达最低点，所以现在我们就可以定义一个参数z，这个z在神经网络里面就叫做学习率，函数就变成了x-zk -> x-90z,z的大小一般为0.01、0,03。

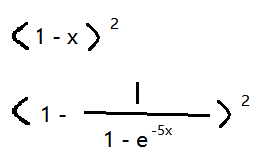
假如我们遇到了这中国咋的函数，我们设置一个很低的学习率，他可能无法到达最低点，有可能设置高学习率可以到达最低点。横坐标代表权重。



现在我们先来看一下sigmoid函数



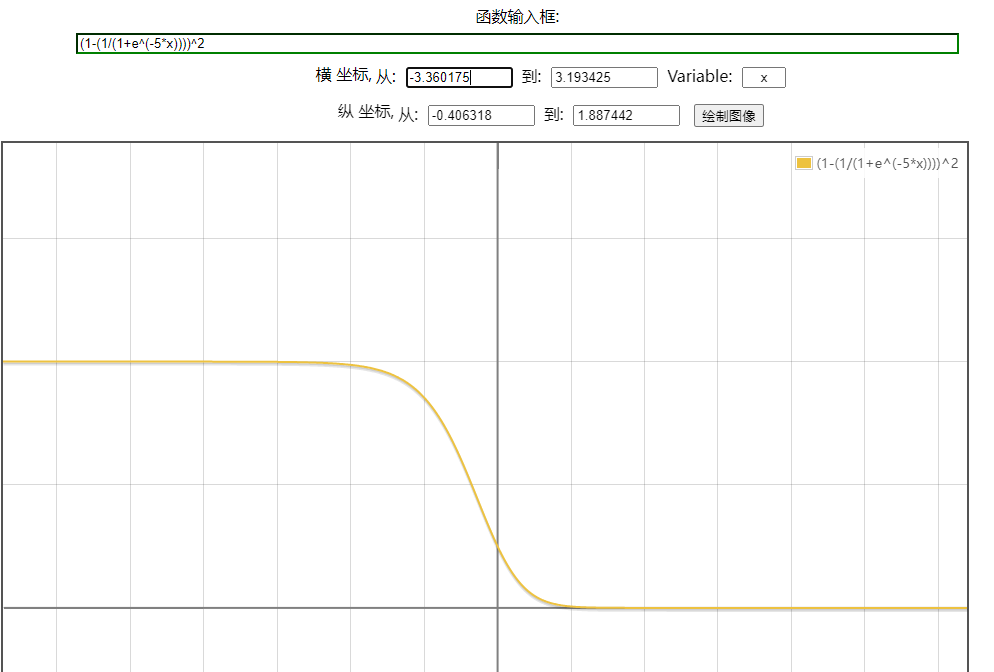
上面我们还没有加上sigmoid函数，加上sigmoid函数，我们的离谱程度函数就变成了



函数绘图器

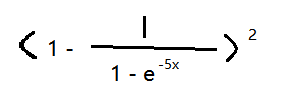
<https://zh.numberempire.com/graphingcalculator.php?functions=&xmin=-10&xmax=10&ymin=-2&ymax=5&var=x>

我们发现离谱程度函数变成了这个样子

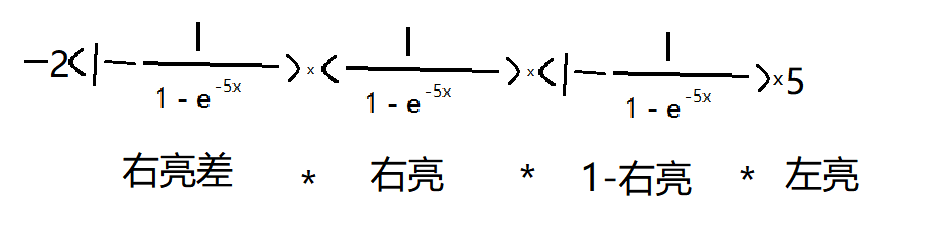


所以为什么我们要设置我们的默认值在0的附近，否则太左边了，就跳不过去了

现在我们的函数为，-5是输入值，x是权重，函数的值是斜率



我们需要对这个函数求导，因为求导才能得出斜率，有斜率才能够计算出步伐和方向的信息，将目前的权重x=2代入，求得斜率



我们将我们的权重x=2输入到导数得出斜率k，x1 = x - zk，

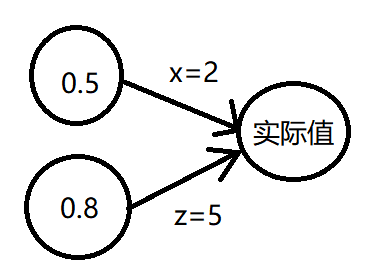
x1代表新的权重，

x代表旧权重2，

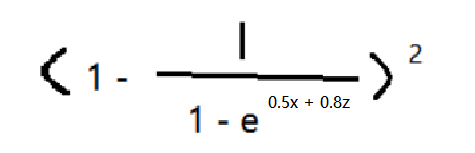
z代表学习率，

k代表离谱程度函数在权重x=2时的斜率。

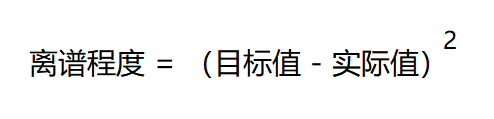
# 从两个权重的网络推广到末层全部权重的调整



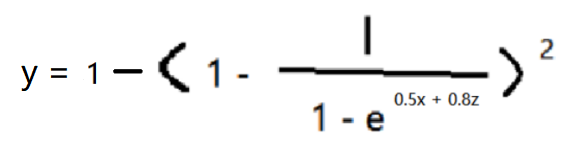
实际值如下



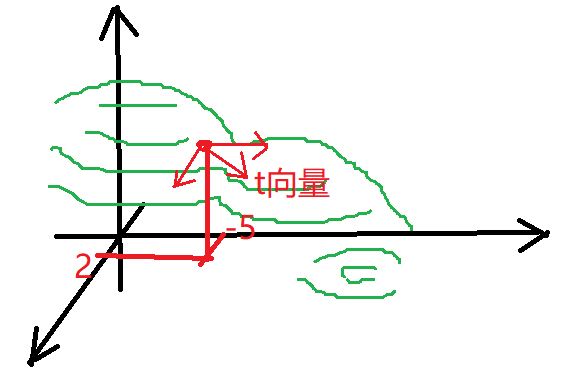
所以可以得出下面公式



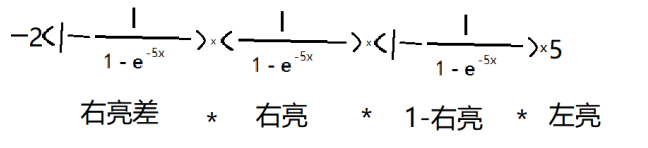
公式下图，我们发现这是一个二元一次方程，y的值跟x和z有关，这不就是一个空间函数吗



假如对应函数的小山坡，分别求出上面函数x和y方向的偏导，这两个导数分别当做向量t的值（x的导数，z的导数），这个向量就是梯度的方向和步伐。

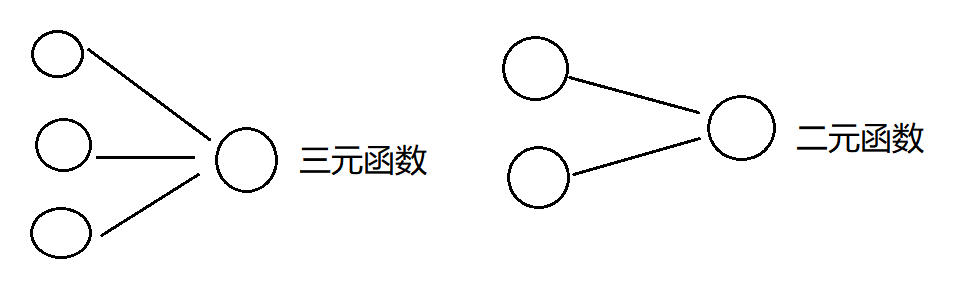


我们需要求出这个函数的导数，跟我们的汉字公式是一样的，我们将权重信息，输入值，输出值代入汉字公式，我们就可以求出这个函数的导数（斜率），求出梯度我们才能知道我们的权重该如何调整，这个是重点，权重调整依赖于下面这个公式。

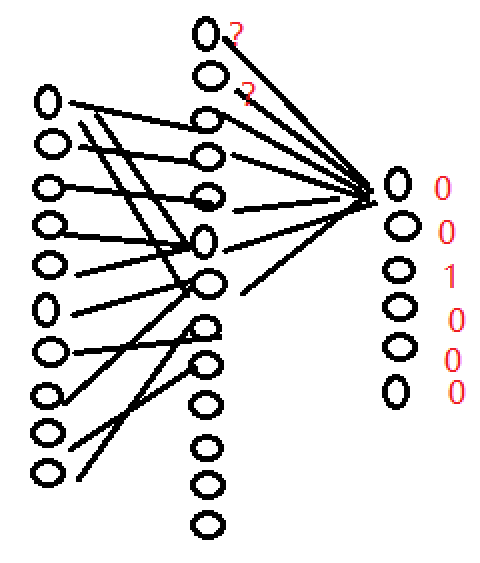


在三维空间中，任何一个点都对应着一个离谱程度，我们的目的就是寻找到离谱程度最小的那个点，所以我们的目的也是寻找离谱程度下降最快的方向，所以我们就会在三维空间中一步步漫游，直至找到最低点。

两个神经元就可以让离谱程度函数这么复杂，我们实际开发中甚至有几百个上千个神经元，那函数就变得超级复杂。

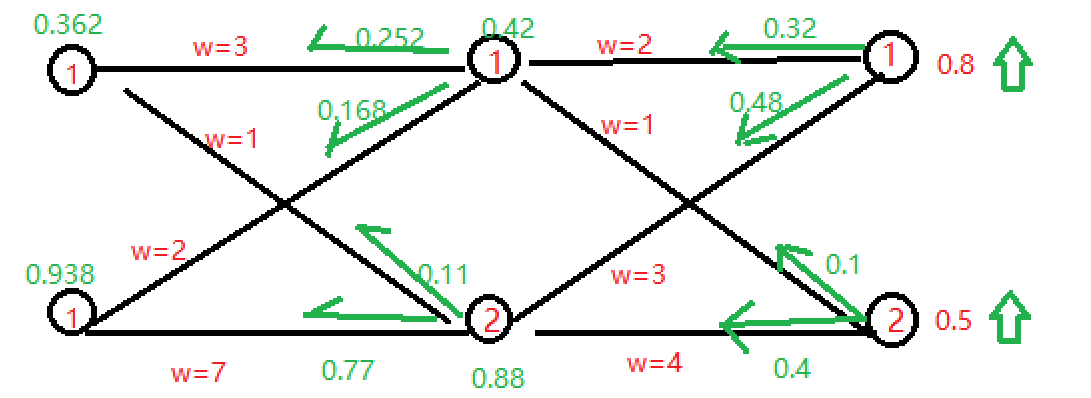


如何调整前面层的神经元的权重呢，一个很关键的一点就是我们需要之前前面几层的神经元的目标值，一般来说我们只知道最后一层的神经元的目标值是多少，但是我们不知道其他层的神经元是多少，如果我们想调整其它层神经元链接的权重，就必须知道每一层的**目标值.**

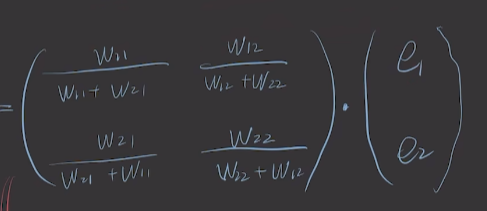


看下图就知道了

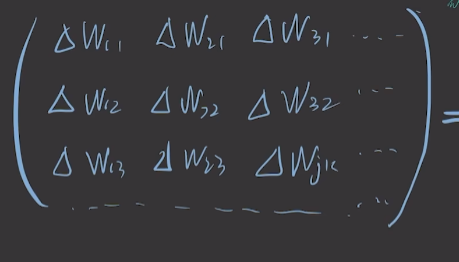
我们可以算出右终端需要上升0.8,0.5，我们通过计算梯度可以算出步伐x轴坐标x新，x新代入均方误差公式得到y新，y新减去y旧就得到了右终端需要上升0.8,0.5

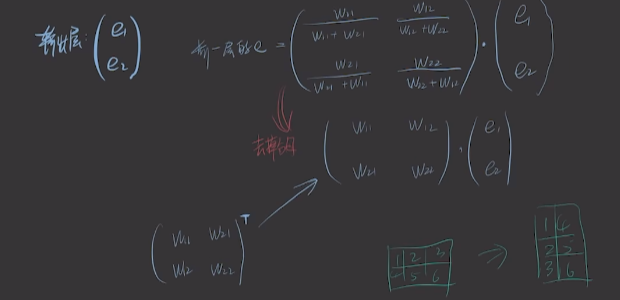


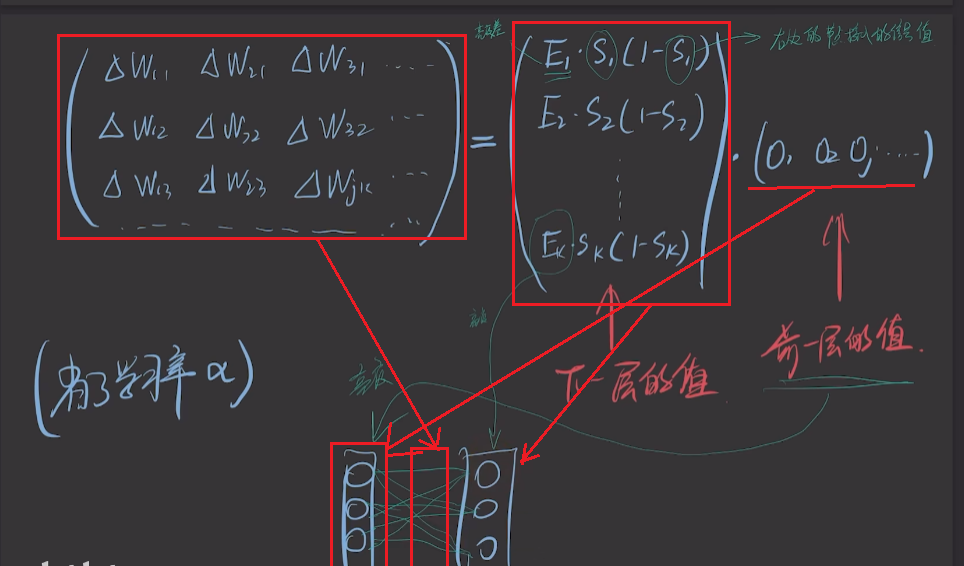
直接可以通过矩阵乘法算出上一层的e



求出每一条线的误差值







反向传播算法，怎么处理激活函数

