### 3.10（选修）直观理解反向传播（Backpropagation intuition）

这个视频主要是推导反向传播。

下图是逻辑回归的推导：

回想一下逻辑回归的公式(参考公式3.2、公式3.5、公式3.6、公式3.15) 公式3.38：

所以回想当时我们讨论逻辑回归的时候，我们有这个正向传播步骤，其中我们计算，然后，然后损失函数。

公式3.39：

神经网络的计算中，与逻辑回归十分类似，但中间会有多层的计算。下图是一个双层神经网络，有一个输入层，一个隐藏层和一个输出层。

前向传播：

计算，，再计算，，最后得到**loss function**。

反向传播：

向后推算出，然后推算出，接着推算出，然后推算出。我们不需要对求导，因为是固定的，我们也不是想优化。向后推算出，然后推算出的步骤可以合为一步：

公式3.40： (注意：逻辑回归中；为什么多了个转置：中的(视频里是)是一个列向量，而是个行向量，故需要加个转置);

公式3.41：

公式3.42： 注意：这里的矩阵：的维度是：。

， 的维度都是：，如果是二分类，那维度就是。

，的维度都是：。

证明过程： 见公式3.42，

其中维度为：、相乘得到，和维度相同，的维度为，这就变成了两个都是向量逐元素乘积。

实现后向传播有个技巧，就是要保证矩阵的维度相互匹配。最后得到和，公式3.43：

可以看出 和 非常相似，其中扮演了的角色， 等同于。

由： 得到：

注意：大写的表示的列向量堆叠成的矩阵，以下类同。

下图写了主要的推导过程：

公式3.44：

公式3.45：

公式3.46：

公式3.47：

公式3.48：

公式3.49：

吴恩达老师认为反向传播的推导是机器学习领域最难的数学推导之一，矩阵的导数要用链式法则来求，如果这章内容掌握不了也没大的关系，只要有这种直觉就可以了。还有一点，就是初始化你的神经网络的权重，不要都是0，而是随机初始化，下一章将详细介绍原因。

