

## 1、选择结构

之前学习了神经网络中很多关于算法上的一些实现细节，比如正向传播求解误差项，反向传播用于求解权重矩阵等等。这一系列都是建立在已选择好神经网络结构之上所要完成的任务。

那么我们拿到一个学习对象后，首先应当做什么，答案是选择结构，即输入层、隐藏层和输出层的组成。

输入层：就是训练数据中特征的数量。

输出层：想要根据学习模型预测出多少种类，输出层就有多少个输出单元。

隐藏层：这里我暂时把它当作一个黑箱，虽然计算步骤我已经了解清楚了，不过原理我还是一知半解（神经网络正是这样解决比较复杂的函数，当层数很多的时候，我们有一个相对简单的输入量，通过加以权重和不同的运算送到第二层，而第三层在第二层作为输入的基础上再来进行一些更复杂的运算，一层一层下去解决问题）老师们的做法是通常选择一个隐藏层，多个也没关系；隐藏层单元数量尽量控制在输入单元数的 1~3 倍，各层一致。

## 2、神经网络执行

结构构建好了，接下来做的就是执行 6 大步骤

- a、随机初始化权重矩阵，共初始化 L-1 层
- b、对每一个输入计算输出
- c、计算出总的代价值，还应考虑正则化
- d、对每一个输入项，计算误差，单个偏导数，然后总体叠加除以 m，计算平均偏导数
- e、如果处于检验阶段，则执行这一步骤，依据双侧数值计算检验偏导数是否计算正常
- f、使用高等梯度下降算法计算每次迭代更新后的权重矩阵

## 3、实战

目前已经完成了一个简单的神经网络预测模型了，输入对象为手写数字 1~10 的 20 pixel \* 20 pixel 图像，输出对象为识别出的数字。

首先进行的是 logistic 一对多回归模型，输入层为 400 个特征值，输出层为 10 个分类，然后利用逻辑分类的方法计算出 10 组  $\theta$  值，这里用的是高等梯度下降算法，每次迭代过程中，需计算出一个代价值，401 个偏导数，有特征值较多，因此不再采用之前的 fminunc，而是采用新的 fmincg 函数，该函数较之前在计算多特征值时有明显优势。

在计算出 10 组  $\theta$  值之后，就可以进行预测了，由于采用的逻辑分类的方法，因此预测结果为置信度最高的那一组结果。

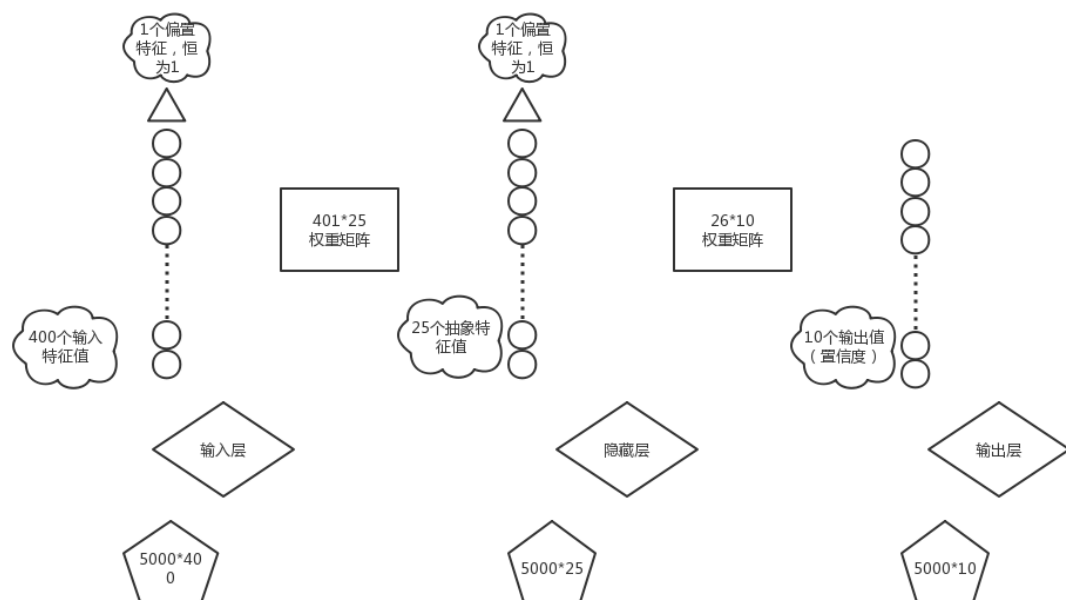
之前讲过，在普通的 logistic 无法预测非线性假设，好在刚刚那个例子属于线性

logistic 回归，因此能够正常运行。如果输入样本为 rgb 图像又该怎么办呢？这里就要用到神经网络模型了。

在第一次试验时，旨在通过正向传播的过程进行结果预测，因此代价值求解以及反向传播计算权重矩阵不再本册实验讨论范畴内。

还是刚才那个例子，输入层 400 个特征值，输出层 10 个分类，隐藏层这里设置为 25 个单元（神经网络的作用就在于此，不用改变输入层结构，而是构建隐藏层结构，进而将简单的特征值转化为复杂抽象的特征值，在转化的过程中实现更高维特征值的提取，最终实现非线性回归，一般来说隐藏层单元数逐级递减，功能越来越复杂），两个权重矩阵已经给出，分别为  $401 \times 25$  和  $26 \times 10$ ，最终得到 5000 组数据关于 10 个分组的置信矩阵，及  $5000 \times 10$ ，置信度最高那组即为本对象的预测结果。

这里给出一个图，好让我有一个直观的感受，可以看到，在每次向前激励的时候，



特征值数量减少，但更加抽象，月到最后，其表达含义越加脱离原数据特征值。因此这样一个神经网络可以预测出非线性的复杂模型。

#### 4、问题

在将这样一个模型拿去预测数据的时候，我发现正确率有时候难以达到 100%，譬如下面这张手写数字，人眼根据其特征很可能预测它为 6，实际结果为 10，先不说它到底是不是 10，就根据其特征值，计算机识别出它为 6 也很正常，毕竟我们训练了它很多次，它的特征值与真正的 6 的特征值很接近，所以要想更精确的预测出它为 10 的话，现在能做的只有加大训练量，譬如增加图片清晰度，使得 10 和 6 的特征值有较大区别，以至于计算机能够区分开来。

