车辆的状态分为四类:

* unacc (Unacceptable 状况很差)
* acc (Acceptable 状况一般)
* good (Good 状况好)
* vgood (Very good 状况非常好)

那我们又是通过什么来判断这辆车的状态好坏呢?有6个feature

* buying (购买价: vhigh, high, med, low)
* maint (维护价: vhigh, high, med, low)
* doors (几个门: 2, 3, 4, 5more)
* persons (载人量: 2, 4, more)
* lug\_boot (贮存空间: small, med, big)
* safety (安全性: low, med, high)
* 如果展示出这些数据, 我们就能清楚的看到这些数据的表示形式了.

| **buying** | **maint** | **doors** | **persons** | **lug\_boot** | **safety** | **condition** |
| --- | --- | --- | --- | --- | --- | --- |
| vhigh | vhigh | 2 | 2 | small | low | unacc |
| vhigh | vhigh | 2 | 2 | small | med | unacc |
| vhigh | vhigh | 2 | 2 | small | high | unacc |

好了, 第一个问题来了, 我们能不能直接这样喂给神经网络让它学习呢?

这里的数据有很多都是文字形式的, 比如 vhigh, small 等. 而神经网络能够读取的数据形式都是数字的。我们需要通过数据预处理将字符用数字来表示。

Method 1：

将每种情况给它一个数字代替呢 比如 (vhigh=0, high=1, med=2, low=3).但这样的方法有一个问题，就是这些情况会存在距离的问题，比如，红黄蓝，（012）红黄距离为1，红蓝距离为2，但其实这种距离毫无意义。

Method 2：类似Mnist,使用onehot编码。比如红黄蓝它们就没有这种距离上的差距概念, 而是每个类别都是独立, 特别的, 不能互相比较的.

* 0 -> [1,0,0,0,0,0,0,0,0,0]
* 1 -> [0,1,0,0,0,0,0,0,0,0]
* 2 -> [0,0,1,0,0,0,0,0,0,0]
* …
* 9 -> [0,0,0,0,0,0,0,0,0,1]