1. 每一个时间点都是一个交通状态s。
2. 神经网络要输出的就是每一个交通状态下的Q值，代表衡量从这个状态发展下去的优劣度。
3. 所使用的代表状态s的特征是[速度\_s、延迟时间\_s、行程时间\_s、气温\_s、降水概率\_s],

如果下一个状态s\_的特征是[速度\_s\_、延迟时间\_s\_、行程时间\_s\_、气温\_s\_、降水概率\_s\_],那么对应的选择行为就是[速度\_s、延迟时间\_s、行程时间\_s、气温\_s、降水概率\_s]减去[速度\_s\_、延迟时间\_s\_、行程时间\_s\_、气温\_s\_、降水概率\_s\_], 得到的各位置的值正为1，负为0，从而将行为离散化（范围是0-2^5, 选择的行为离散化向量[0,1,0,.....]化成整数即代表所在输出向量的维度。）。

所得到的奖励用延迟时间的差值来表示=延迟时间\_s - 延迟时间\_s\_。

1. 所采用的神经网络功能为输入s,，输出Q(a：s)，要输出神经元个数，即输出向量维度为2\*n\_features = 32。
2. 记忆池的存储结构是[n\_features, a, r, n\_features\_].

6、预测未来情况时，输入当前状态，输出未来各种可能性下的Q\_value,选取最大的一个就行了，其实这里的Q\_value指代的就是交通通畅指数，而不是交通拥堵指数。

**DRQN的算法步骤：**

1、初始化DRQN网络结构，参数是，初始化目标网络，参数。

2、初始化贪心系数epsilon,学习率learning\_rate,reward衰减系数gamma,迭代次数Episodes,每一个episode的迭代轮数T，训练batchsize，神经网络参数轮换周期transter\_cycle，

3、for episode in Episodes do

初始化交通状态S*t*  = 

For t 从0到T;

1>选择行为（输出一个整数，范围在（0到2^n\_features-1））：以1-epsilon的概率选择，epsilon的概率随机选择行为。

2>行为确定好后，在数据表中找到符合这个行为的所有状态s\_all,然后再s\_all中随机选择一个作为（如果在s\_all中找不到符合条件的，则重新确定行为），然后根据S*t*以及来计算奖励.

3>将经验（）放入记忆池中，

4>随机拿出batchsize个数据，分别计算q\_eval以及q\_next

5>构造 🡺 q\_target

6>根据q\_eval以及q\_target,反向传播提升网络q

7>如果迭代次数为transter\_cycle的整数倍，就更新

当前状态 = 

达到单轮次game最大迭代次数T时停止该轮次训练，重新回到交通状态初始化阶段。

4、备注：需要记录的是，每次更新Q时的损失loss.