策略说明

一、算法原理

本策略使用的是 DDPG（Deep Deterministic Policy Gradient）深度强化学习算法，因为传统的 Actor-Critic 收敛慢的问题，所以 Deepmind 提出了 Actor Critic 升级版 Deep Deterministic Policy Gradient（即 DDPG），后者融合了 DQN 的优势, 解决了收敛难的问题。

DDPG 我们可以拆开来看 Deep Deterministic Policy Gradient。Deep：首先 Deep 我们都知道，就是更深层次的网络结构，我们之前在 DQN 中使用两个网络与经验池的结构，在 DDPG 中就应用了这种思想。 PolicyGradient：顾名思义就是策略梯度算法，能够在连续的动作空间根据所学习到的策略（动作分布）随机筛选动作。Deterministic：它的作用就是用来帮助 Policy Gradient 不让他随机选择，只输出一个动作值。

随机性策略，∑π(a|s)=1∑π(a|s)=1 策略输出的是动作的概率，使用正态分布对动作进行采样选择，即每个动作都有概率被选到；优点，将探索和改进集成到一个策略中；缺点，需要大量训练数据。

确定性策略，π(s)S→Aπ(s)S→A 策略输出即是动作；优点，需要采样的数据少，算法效率高；缺点，无法探索环境。然而因为我们引用了 DQN 的结构利用 offPolicy 采样，这样就解决了无法探索环境的问题

从 DDPG 网络整体上来说 :他应用了 Actor-Critic 形式的, 所以也具备策略 Policy 的神经网络 和基于 价值 Value 的神经网络，因为引入了 DQN 的思想，每种神经网络我们都需要再细分为两个, Policy Gradient 这边，我们有估计网络和现实网络，估计

网络用来输出实时的动作，供 actor 在现实中实行，而现实网络则是用来更新价值网络系统的。再看另一侧价值网络，我们也有现实网络和估计网络，他们都在输出这个状态的价值，而输入端却有不同，状态现实网络这边会拿着从动作现实网络来的动作加上状态的观测值加以分析，而状态估计网络则是拿着当时 Actor 施加的动作当做输入。

DDPG 在连续动作空间的任务中效果优于 DQN 而且收敛速度更快，但是不适用于随机环境问题。

二、参数说明

在单路口配时算法中，状态、动作、奖励的定义分别如下：

状态：同一阶段下的相位车道的排队长度之和[南北直行排队长度和，南北左转排队长度和，东西直行排队长度和，东西左转排队长度和], 通过 traci.lanearea.getJamLengthMeters('e2 型检测器编号')获取；

动作：通过 actor 网络进行输出；

更新频率：以周期为单位，在一周期开始前，根据当前环境的观测值进行动作的选取；

动作输出：（周期时长，阶段持续时长占比）考虑到周期受到交通量大小的影响，为此为周期设置一个上下限值，防止周期过大或过小，将周期设置为基准周期的 0.6-2 倍之间，基准周期为 80s，并考虑到阶段时间过短造成不良影响，不执行时长小于 8s 的阶段。

例如：action = [1.5, 0, 0.5, 0.3, 0.2]，a[0] = [1.5]表示为此周期时长为基准周期时长的 1.5 倍，用 np.clip(a[0], 0.5, 2)进行截断；a[1:] = [0, 0.5, 0.3, 0.2] ——表示各阶段时长占总周期比例，若为 0 则表明不进行该阶段。

奖励：排队车辆数、等待时间、平均时延以及路口吞吐量的加权

排队车辆数：L = sum(l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8)

通过 traci.lanearea.getJamLengthVehicle('e2 型检测器编号')获取

等待时间：T = sum(t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8)

通过 traci.lane.getWaitingTime('道路编号')获取

平均时延：D = sum(d_1,d_2,d_3,d_4,d_5,d_6,d_7,d_8)

通过 traci.lane.getLastStepMeanSpeed('道路编号')获取

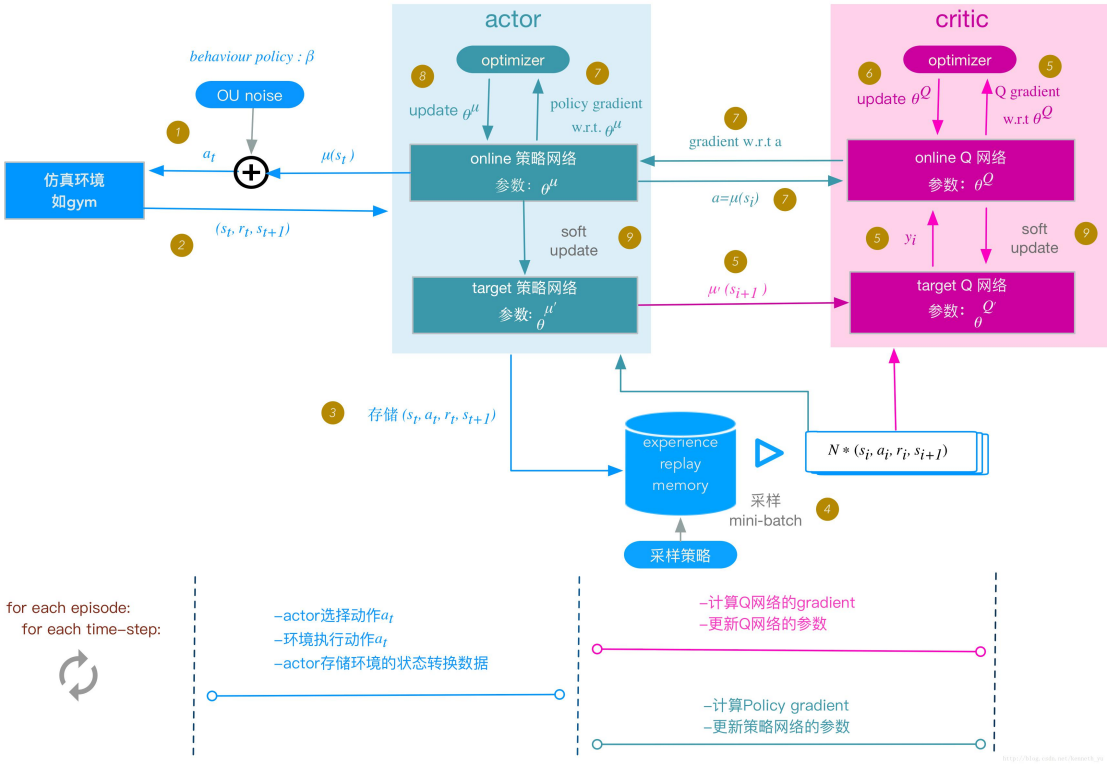
吞吐量：Throughput 通过统计通过车辆名称进行统计

通过 traci.inductionloop.getLastStepVehicleIDs('e1 型检测器编号')获取

加权公式：reward = W1*L + W2*T + W3*D + W4*Throughput

(W1 = -1, W2 = -0.02, W3 = -1, W4 = 0.2)

三、算法原理流程



四、优化效果

仿真时长为 10800s，在一个仿真内，三种算法的指标对比如下：

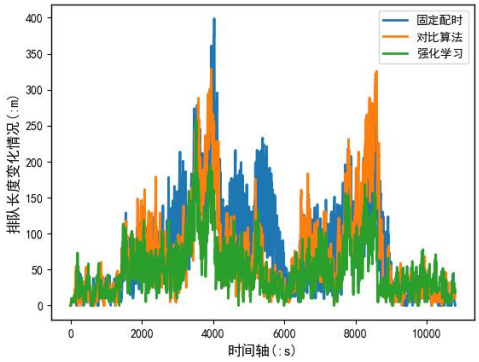


图 3. 排队长度对比图

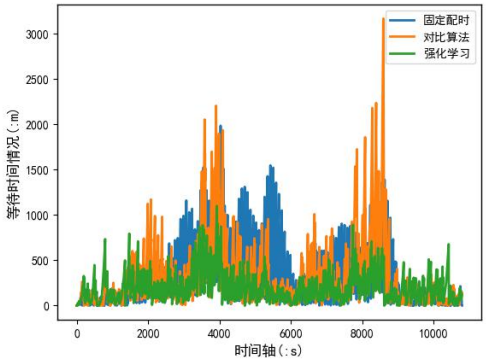


图 4. 等待时间对比图

直观上可以看到，强化学习算法的结果较对比算法和固定配时均有一定程度的提高。

仿真时长的 10800s 分为八个时段，分别是 0-1350,1351-3150,3151-4050,4051-5400,5401-6300,6301-7650,7651-8550,8551-10800（单位秒），每个时段内的车流分布服从相同到达概率的泊松分布，八个时段的到达概率各不相同，基本符合一天 24 小时内的车流量增减规律，含超低峰，低峰，正常，高峰期。下面，对八个时段的结果进行统计分析，结果如下：

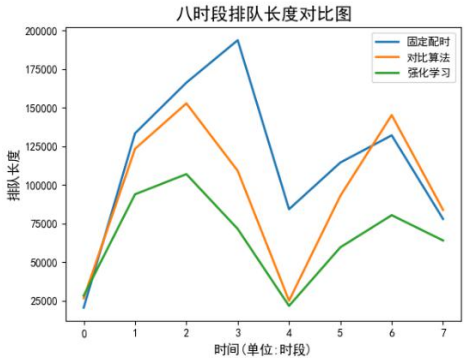


图 5. 八阶段排队长度对比图

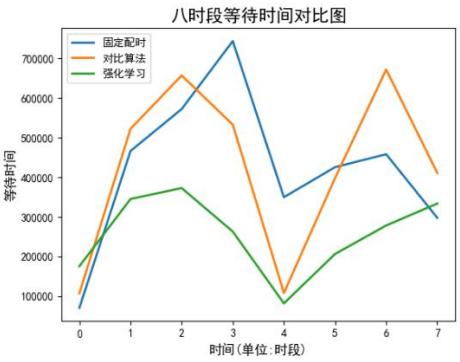


图 6. 八阶段等待时间对比图

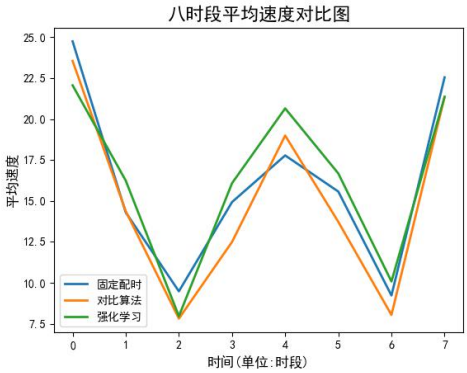


图 7. 八阶段平均速度对比图

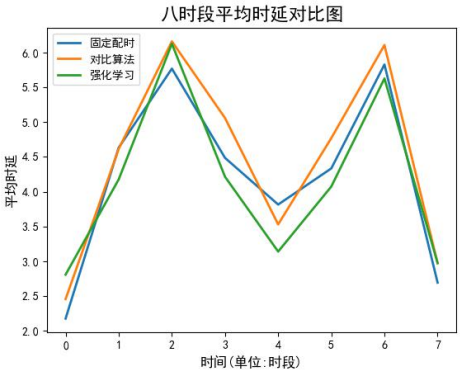


图 8. 八阶段平均时延对比图

具体指标提升情况如下：

表 1 排队长度对比

排队长度	强化学习	固定配时	对比算法	（强-固）/固	（强-对）/对
第一时段	28311.67	20582.62	26648.28	0.375513588	0.062419996
第二时段	93890.41	133364.9	123338.4	-0.295988733	-0.238757751
第三时段	106959	166139.2	152725.7	-0.356208438	-0.299665587
第四时段	71496.21	193600.1	109097.6	-0.630701595	-0.344658232
第五时段	21706.18	84266.55	25066.64	-0.742410529	-0.134061105
第六时段	59602.15	114460.8	92930.39	-0.479278774	-0.358636654
第七时段	80378.13	132037.4	145144.3	-0.391247307	-0.446219194
第八时段	64021.71	77954.63	83841.35	-0.17873118	-0.236394551

表 2 等待时间对比

等待时间	强化学习	固定配时	对比算法	（强-固）/固	（强-对）/对
第一时段	175552	70645	107072	1.484988322	0.639569635
第二时段	345154	466341	522336	-0.259867779	-0.339210776
第三时段	372796	572223	657018	-0.34851273	-0.432593932
第四时段	262992	743861	532688	-0.646450076	-0.506292614
第五时段	81768	349731	107932	-0.766197449	-0.242411889
第六时段	206388	425574	397919	-0.515036163	-0.481331628
第七时段	278340	458177	671919	-0.392505516	-0.58575364
第八时段	333713	297601	410964	0.121343678	-0.187975102

表 3 平均速度对比

平均速度	强化学习	固定配时	对比算法	（强-固）/固	（强-对）/对
第一时段	22.06616	24.75242	23.55867	-0.108525241	-0.063353024
第二时段	16.24236	14.30076	14.37426	0.135768965	0.129961949
第三时段	7.95742	9.477139	7.811386	-0.160356301	0.018694947
第四时段	16.08612	14.93147	12.4958	0.077329517	0.287321772
第五时段	20.65662	17.7826	18.99635	0.161619823	0.087399427
第六时段	16.68104	15.57423	13.72325	0.071066829	0.21553092

第七时段	10.07811	9.223101	8.034189	0.092702925	0.254402876
第八时段	21.35661	22.556	21.37065	-0.053174033	-0.000656984

表 4 平均时延对比

平均时延	强化学习	固定配时	对比算法	（强-固）/固	（强-对）/对
第一时段	2.807963	2.175902	2.456784	0.290482645	0.142942778
第二时段	4.178268	4.635115	4.617822	-0.098562176	-0.095186521
第三时段	6.127666	5.770085	6.162027	0.061971518	-0.005576219
第四时段	4.215032	4.486713	5.059812	-0.060552314	-0.166958797
第五时段	3.139618	3.815858	3.53027	-0.177218386	-0.110657787
第六时段	4.07505	4.335476	4.771	-0.060068624	-0.145870784
第七时段	5.62868	5.829859	6.109603	-0.03450829	-0.078715841
第八时段	2.974916	2.692706	2.971613	0.104805433	0.001111709

可以看出强化学习算法的结果在大多时段优于固定配时和对比算法。

五、举例说明

在执行策略 3 的过程中，在某一周期将要结束前，记录下这一周期中各阶段的排队长度最大值，作为状态输入到 DDPG 模型中，DDPG 模型根据以往的训练经验，计算出可以使奖励值最优化的配时方案（奖励值包括: 排队车辆数、等待时间、平均时延以及路口吞吐量），并下发到信号机中执行。