

Report

实现基本的驾驶智能体

问题: 在你的报告中, 说明观察到的智能体的行为。它最终到达目标位置了吗? 有什么其他值得注意的地方吗?

Answer:

由于智能体的行为是随机的, 运气好的话, 它是可以到达目标位置的, 但是到达目标位置时一般都超过了deadline。如果deadline = -100时智能体还没有到达目标位置的话, 就会尝试失败。

Inform the Driving Agent

QUESTION: What states have you identified that are appropriate for modeling the smartcab and environment? Why do you believe each of these states to be appropriate for this problem?

Answer:

1. next_waypoint (下一路径点的位置)
2. light (信号灯的状态)
3. oncoming (前面车的行驶方向)
4. left (左边车的行驶方向)

next_waypoint是根据目标位置得到的, 用于指示前进方向。

light, oncoming, left是为了不违反交通规则, 不发生撞车事故。

没有选择right是因为右边的车不会对智能体产生影响, 不需要考虑。

OPTIONAL: How many states in total exist for the smartcab in this environment? Does this number seem reasonable given that the goal of Q-Learning is to learn and make informed decisions about each state? Why or why not?

Answer:

在这个环境中一共有 $3*2*4*4=96$ 个状态，对于Q-Learning来说是合理的。这些状态包含了所有可能会使智能体违反交通规则和发生撞车事故的情况。

Implement a Q-Learning Driving Agent

QUESTION: What changes do you notice in the agent's behavior when compared to the basic driving agent when random actions were always taken? Why is this behavior occurring?

Answer:

智能体不再是随机的移动。经过学习以后，智能体能够不违反交通规则，不发生撞车，快速的到达目标位置。即使有deadline，对智能体来说也能轻松应对。这是因为我们设置的Qtable，并且根据智能体每一次的行为都对Qtable进行了更新。当智能体在某个位置出现了错误的行为或者去了错误的方向时都会被惩罚，经过几轮学习之后，智能体就能够自己规范行为并去向目标位置的方向。

Improve the Q-Learning Driving Agent

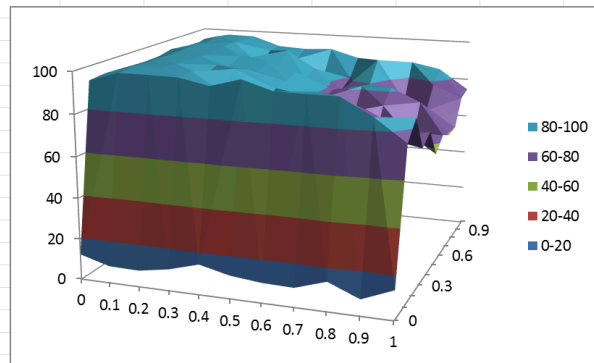
QUESTION: Report the different values for the parameters tuned in your basic implementation of Q-Learning. For which set of parameters does the agent perform best? How well does the final driving agent perform?

Answer:

alpha gamma

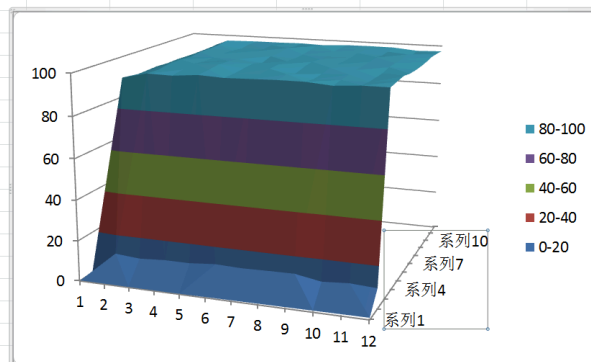
我将alpha和gamma分别从0-1以0.1为间隔遍历了所有的值，并且以能够到达目标位置且净奖励为正作为评分标准，满分为100分，并且取10次评分平均值为最终评分。输出的表格如下图：

gamma/alpha	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0	12.4	8.3	8	10.5	14.7	11.2	9.7	9.3	14.1	7.9	14.1
0.1	93.3	97.7	97.7	97.4	94.5	98.5	94.1	93.8	93.8	85.8	75.4
0.2	94.1	98.4	98.5	97.7	89.3	81.1	87.3	92.2	70	75.8	83.6
0.3	93.8	97.9	98.2	93.4	97.4	80.1	88	77.4	73	69.1	66.3
0.4	93.3	98.1	97.8	96.1	94.9	86.4	71.3	83.6	79	69.1	70.8
0.5	93.1	97.6	97.4	92.7	91.1	87.4	74.2	73.5	59.9	66.1	56.5
0.6	93.4	98.3	97.6	95.9	94	89.4	77.5	68.9	75.2	57.3	62.5
0.7	94.9	98.5	98	92.9	88.4	83.6	78.5	81.4	81.5	75	59.3
0.8	93.8	97.9	98.2	85.5	95.1	86.3	75.7	79.8	55.5	67.9	59.9
0.9	92.8	98.7	97.4	92.2	88.3	78.3	79.5	90.2	69	78.5	72.7
1	94.3	97.4	97.1	92.4	91.6	92.3	88.2	87.5	87.5	84.6	74.6



我们能看出在gamma不为0时成绩都比较好，但是gamma从0到0.1变化时成绩出现了突变，所以猜测最好成绩也有可能在0-0.1之间，于是我们对0-0.1之间又进行了评分，如下图：

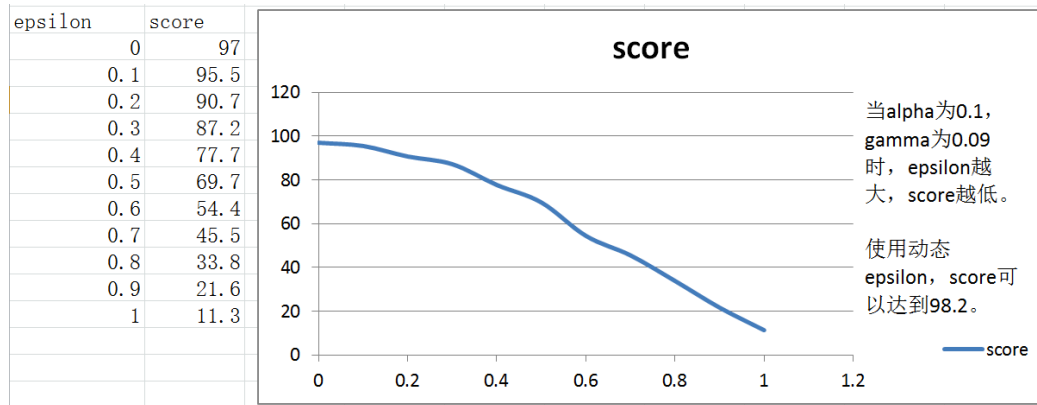
gamma/alpha	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0	11.1	10.1	10.9	10.9	12	11.6	11.9	12.1	9.9	10.8	10.1
0.01	94.6	97.3	97	98.1	97.5	97.8	98.1	98.1	97.1	98	98
0.02	93.5	97.1	97.9	98.4	97.4	97.8	98.7	97.2	97.5	97.4	98.4
0.03	93.7	98.1	98.7	99.1	97.9	98.4	96.9	97.5	97.9	97.3	98.8
0.04	94	98.1	97.8	98.7	98.2	98.6	97.9	98.4	98	97.2	97.8
0.05	93.6	97.6	97.9	97.8	96.6	98.1	98.5	98.1	98	98.1	97.4
0.06	93.8	98.2	98.1	98.1	98	97.6	97.9	97.6	98.1	98.2	97.3
0.07	92.9	98.2	97.8	98.2	98.1	97.6	97.5	97.6	98.4	98.2	97.6
0.08	93.5	98.4	97	97.4	98.6	98	98.1	97.9	97.3	98.6	98.1
0.09	92.9	99	97.6	97.7	97.5	98.7	97.9	97.9	97.5	96.5	97.8
0.1	94	96.4	97.4	97.4	97.6	97.2	97.9	98	97.9	96.8	97.2



在这个图表中我观察到gamma不等于0时，智能体的表现都是不错的，都在90分甚至95分以上，但是还是能找到表现最优的参数值alpha=0.1，gamma=0.09，此时分数为99分。并且我对每一个参数组合都进行了10次评分取平均值，所以这个结果是可信的，而不是偶然得出。

epsilon

在 $\alpha=0.1$ ， $\gamma=0.09$ 的情况下对epsilon从0-1以0.1为间隔遍历了所有的值,仍然以能够到达目标位置且净奖励为正作为评分标准，满分为100分。输出的表格如下图：



我们能看出随着epsilon值升高，分数是下降的。但是直接取 $\epsilon=0$ 或者0.1我认为还是不适用的。因为在最开始的时候epsilon的值应该大一些，智能体应该多尝试不同的情况以适应不同的环境。在训练次数小于60次的时候，`self.epsilon = (60 - self.ntrain) / 60`，同时epsilon的值不断减小，直到60次以后， $\epsilon=0$ 。经过测试，这种动态epsilon的评分可以达到98.2分，比单纯的定值都要更好。

*QUESTION: Does your agent get close to finding an optimal policy, i.e. reach the destination in the minimum possible time, and not incur any penalties?
How would you describe an optimal policy for this problem?*

Answer:

在这个算法中，通过类似于 Grid Search 的方法来确定 α ， γ 的最优值，并且再次对epsilon进行了优化。

在程序中的智能体依据优化，已经可以顺利到达目的地，并且能够保证净奖励为正。对于这个智能体来说，最优算法应该是在不违反交通规则，不发生事故的前提下快速到达目标位置。