Question 1 - Random selection of actions

Observe what you see with the agent's behavior as it takes random actions. Does the smartcab eventually make it to the destination? Are there any other interesting observations to note?

Answer: Chosing random actions at each update, the agent is still able to reach the destination with a certain probability. Figure 1 shows the accumulated number of trials, in which a "dumbcap", disregarding all traffic rules and selecting the next action completely at random, reaches the destination within the given deadline.

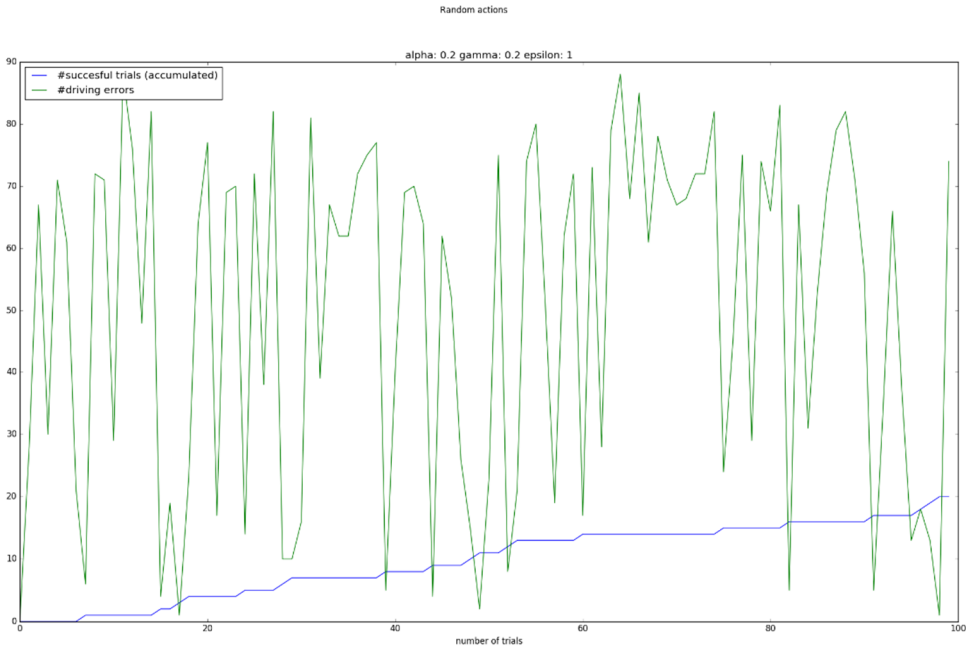

Figure 1: agent with random action selection

Any model-based smartcab should achieve a significantly better performance than the dumbcap represented by figure 1. Specifically, the success-curve of a smartcab that learns to drive perfectly should be a line after a certain number of trials, while the number of traffic violations should be zero.

Question 2 - Inform the Driving Agent

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: State information which can be directly obtained from the environment includes the traffic light state and the surrounding traffic (left,right,oncoming). Figure 2 shows the performance of an agent using only this state information:

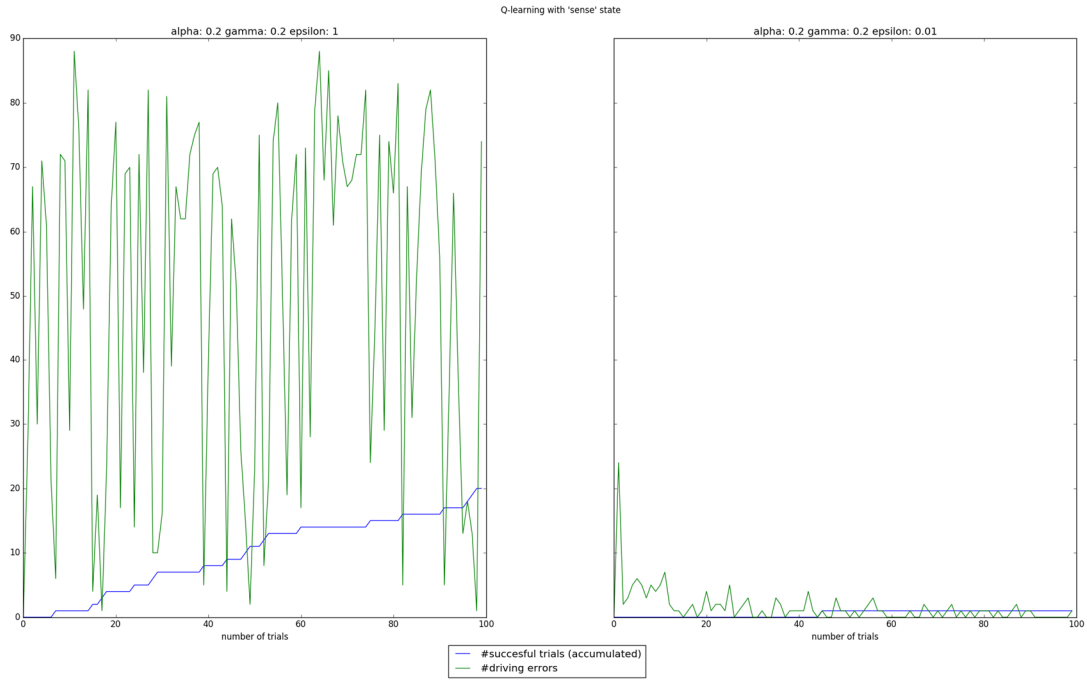

*Figure 2: Agent with *sense* state*

The bad performance of this model can be attributed to the fact, that the next waypoint was not included in the state space. As a consequence, the model cannot relate the reward to chosing the correct direction. With the inclusion of "next waypoint" into the state space, the performance gets significantly better:

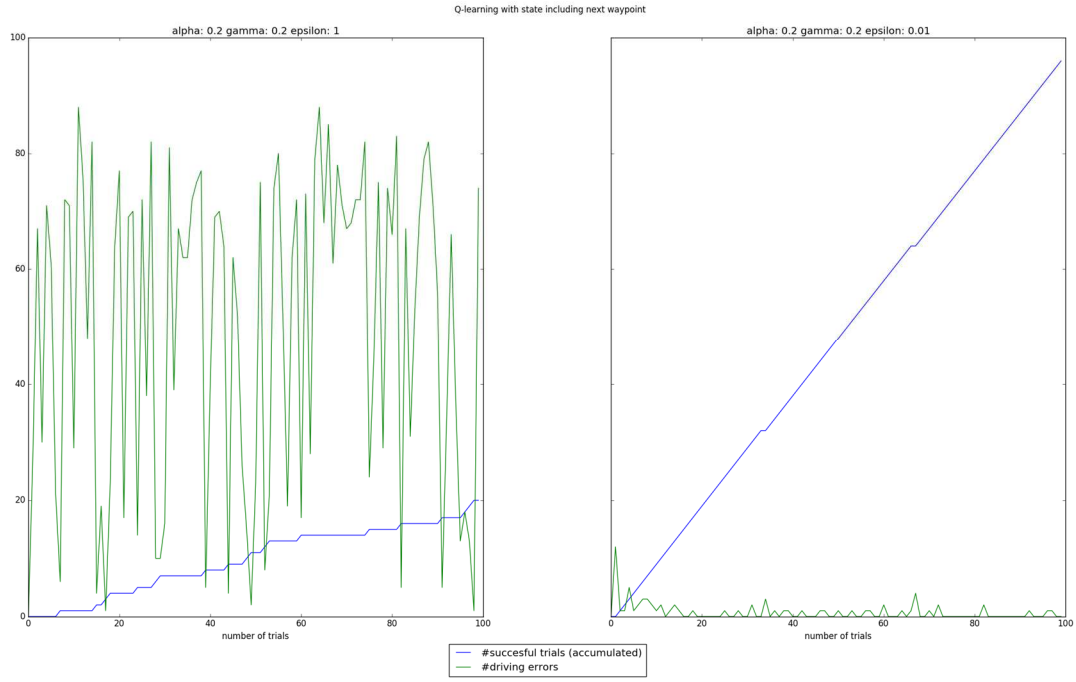

Figure 3: Agent including next waypoint information in state

Question 3 - Implement a Q-Learning Driving Agent

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: The comparison of a q-learning based smartcab with an agent that just choses direction at random reveals that the model-based agent systematically learns to reach the destination within the soft deadline and with a minimum number of traffic violations. The "success"-curve of the model represented by the right plot of figure 3 is almost a straight line, while the mean number of driving errors seems to go down over the number of trials.

Question 4 - Improve the Q-Learning Driving Agent

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: The following figure shows the performance of agents with different settings for the alpha and gamma parameters:

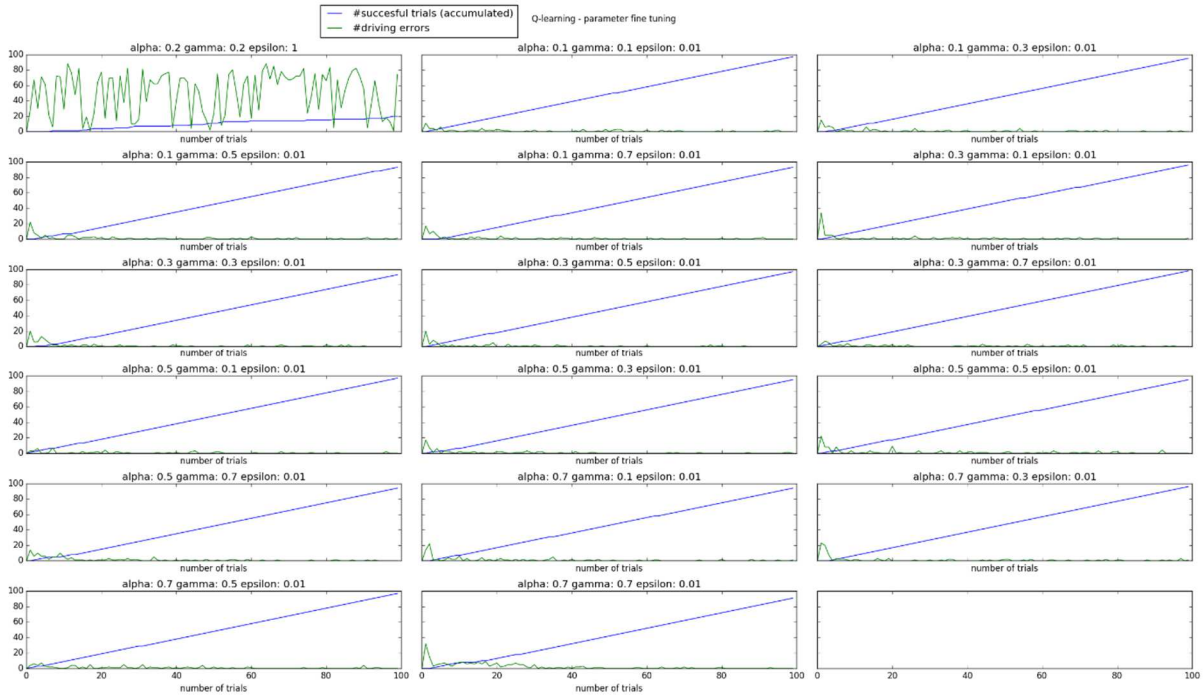

Figure 4: Q-learning with parameter finetuning

In this particular run, the best performance could be achieved with an alpha of 0.3 and a gamma of 0.7, with a succes rate of 98/100 and zero traffic violations in the final trial.

Question 5 - Finding the optimal policy

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:
An optimal policy for this particular problem could be formulated as follows:

If the next waypoint can be reached without traffic violation, move to the next waypoint. Otherwise, wait.

An agent that would follow these rules, would reach the destination in the minimum possible time. The agent above reaches the destination within the soft deadline, but not necessesarily in the minimum possible time. The reason for this can be found in the environment implementation, which in certain cases rewards both "waiting" and "moving away from the next waypoint" with 0 points. If the next waypoint cannot be reached, an agent may therefore move away from the next waypoint, which would be a violation of the optimal policy.