Imitation Learning

Earl Wong

What Is the Essence of Reinforcement Learning?

- Reinforcement learning occurs through a process of trial and error.

- Reinforcement learning is predicated on the idea of a reward function.

- As a result, the concept of “exploration” is deeply embedded in reinforcement learning.

- i.e. Try different things, get different feedback / rewards, and try to develop the best policy / actions that will maximize the cumulative return.

What Are the Shortcomings?

- Trial and error may not be an “expensive” option.
- i.e. We shouldn’t learn wilderness survival skills, by getting lost in a National Forest and “figuring it out”.
- i.e. We shouldn’t pack our own parachute for skydiving, having never done it before.
- i.e. We shouldn’t learn how to fly a drone, by crashing 200 of them into a wall.

What Are the Shortcomings?

- In addition, the reward could have a very long time horizon or be sparse, making reinforcement learning challenging.

- Long time horizon: Your computer simulation will take 6 months to complete (assuming no code crashes), before producing a result.

- Sparse: The only feedback that you will receive on your course performance, is your grade on your final project.

- Difficult applications for RL: Classic - Maze navigation, Games - Montezuma’s Revenge, Robotics - Robotic manipulation

Imitation Learning

- An alternative for addressing these shortcomings, is to observe and learn from an expert.

- Now, trial and error and reward are not part of the equation.

- Instead, an expert knows and demonstrates the correct approach.

- We then learn a policy by IMITATING the expert.

Imitation Learning: Behavior Cloning

- Observe the expert, recording the (state, action) pairs associated with the expert.
- i.e. Generate a fixed dataset, based on the expert's knowledge.
- Apply supervised learning (regression or classification) to the fixed dataset, to learn a policy that mimics the expert's behavior.

Imitation Learning: Behavior Cloning

What you have done:

- There exists a large state / observation space.
- You have trained a policy, using observed states / observations (drawn from this space) and the expert actions.
- But, what if your expert data only contains states / observations in a finite part of the space?

Imitation Learning: Behavior Cloning

- If your policy then takes an action that leads to visiting an unseen state / observation, problems could arise.

- This is because your policy has not been trained to choose the appropriate actions, given the unseen state / observation.

- In addition, subsequent actions based on “follow on” unseen states / observations, could lead you deeper into the explored space, eventually culminating in the complete failure of the task.

Solution

- DAgger - dataset aggregation.
- We acknowledge that the expert training data could have incomplete coverage.
- When we encounter new, unseen states / observations, we obtain a new expert action label for the unseen state / observation.
- We then augment the original expert training data with this new (state, action) pair.
- We then retrain on the larger dataset, with the richer state / observation coverage.

Detection

- Different heuristics can be applied to the action outputs, to help detect the unseen states / observations:

- For example, if the current input state / observation produces an action distribution with high entropy, there is a high probability that the distribution resulted from an unseen input state / observation.

- For example, if the current input state / observation produces an action distribution with a small gap between the most likely and next most likely action, there is an increased probability that the distribution resulted from an unseen input state / observation.

Detection

- A general heuristic can also be at the input = state / observation:

- For example, how far is this input state / observation from the known distribution of the expert training data?

- For example, how far is this latent input state / observation from the known latent state / observation of the expert training data?

- If the distance is large, then the state / observation is probably not contained in the expert training data.

Other Practical Issues

- In addition to performing data augmentation using DAgger, other practical issues exist.
- For example, is the newly discovered state / observation synonymous with an anomaly?
- If true, the (state, action) pair needs to be given significantly more weight.
- This can be done by increasing the back propagated loss associated with this (state, action) pair, or by adding the (state, action) pair N times to the original expert training dataset.

Other Practical Issues

- Also, will our model under fit, if we add too many additional unseen states / observations to our original expert training data?

- If we make our model large (in anticipation of this problem), will we then overfit, losing generalization?

- Losing generalization will then exacerbate the original problem of visiting unseen states / observations.

- This is because, even though the input states / observations lie in the same distribution as the expert, the resulting policy actions may not be “smooth / continuous” relative to the expert.

A Toy Problem

- openAI gym cart pole has 4 input states and 2 discrete action outputs.

- Input: car position [-2.4, 2.4], car velocity [-3, 3], pole angular position [-12, 12], pole angular velocity [-2, 2].

- Output: action 0 = left, action 1 = right.

- The expert says that if the pole angular position is between [-12, 0), choose action 0.

- The expert says that if the pole angular position is between [0, 12], choose action 1.

A Toy Problem

- Behavior cloning allows you to train a policy from a finite dataset of (state, action) pairs created from expert knowledge.

- We observe that model generalization is good = returns are high, even for pole angular positions not in the original training data.

- Next, an updated version of the game is released.

- Now, pole angular velocities are allowed to range from [-50, 50] instead of [-2, 2].

A Toy Problem

- Now, for various use cases, the returns quickly plummet, and the episode terminates quickly.

  What went wrong?

- Although the expert model was able to generalize for angular pole positions, it could not generalize for a larger pole angular velocity range.

- 1) Larger pole angular velocities have created a significant distributional shift.

- 2) Larger pole angular velocities have expanded the existing state / observation space.

A Toy Problem

- DAgger could be applied, to address this problem.

- In lieu of DAgger, we could also take the existing policy (trained on expert data) and use it as an initial condition for reinforcement learning.

- If the latter option is chosen, we no longer need to worry about how to weigh the data points from the unseen states / observations correctly.

- This is because -in the limit and assuming adequate exploration- the trial and error process of reinforcement learning will automatically address the matter.

Engineering Design

- Engineering design requires you to design a solution / choose a path with the highest probability of success (minimum risk), at the lowest cost.
- In order to accomplish this, the designer needs to fully understand the tools at his / her disposal.
- i.e. What are the strengths and weaknesses of imitation learning and reinforcement learning?
- i.e. How do I maximize the strengths, while minimizing the weaknesses?
- Should I use the expert data from imitation learning to fine tune a generic policy obtained from reinforcement learning, or, vice versa?