Project Tips

CompSci 590.2
Ron Parr
Duke University

General debugging

• Why is my learner doing badly?

• Try your code/algorithm on a smaller problem:
  • Inverted pendulum
  • Discrete MDP with 1’s – 10’s of states and indicators on state as features

• If your code can’t solve a small problem, it is unlikely to solve a big problem – look for bugs
General Debugging

• My code can solve trivial problems but can’t solve hard problems

• Try simplifying the problem:
  • Domain dependent, e.g.,:
    • Remove dimensions
    • Start closer to the goal
    • Remove randomness
  • Try with gamma=0 (agent should maximize immediate reward)
  • Try with gamma=0 and cherry-picked states

• If you can’t maximize immediate reward
  • This is a supervised learning problem
  • Check your features, function approximation architecture

You can solve easier problems, not hard ones

• Is the problem exploration?
  • Is your reward function sparse?
  • How often are you seeing non-zero rewards in your training data?
  • Consider revising your exploration strategy
  • Consider adding a shaping reward

• Is the policy/value function representation rich enough?
  • Big problems require good features and/or big nets
  • These require more samples, training time
Representation Issues

• Is your state Markovian in your features?
• Crucial for value function methods
• Very helpful for policy gradient methods

• Possible, though hard to detect this algorithmically
• Need to think carefully about your features

Unexpected behavior

• Your learner consistently does something coherent (not random flailing) but it’s not what you expected

• Is your discount factor reasonable? (Is behavior myopic?)

• Is your reward function reasonable? (Where did it come from?)
Debugging reward function issues

• Create your own “good” policy and compute the score of a trajectory
• Compute the score of a trajectory from your learned policy
• Does the “bad” trajectory have a higher score than your good one?
• If yes, then the problem is your reward function

Debugging value function issues

• Can your value function represent the reward ($\gamma = 0$)?
• If not, then little chance of representing $V^*$
• Compare value function to Monte Carlo estimates from rollouts
• Find states where value function approximation is particularly bad
• Think about your inputs/features
• Is it possible that there is a Markov property violation because two different states could look the same through the lens of your features/inputs?
Debugging policy function issues

• Come up with a reasonable policy by some other means

• Train your policy function on this as a supervised learning problem

• If your training error is large, then your policy function probably isn’t rich enough to represent reasonable policies for this domain

• If your training error is small, but your test error is large, then you are overfitting – simplify your policy function or use more training data

Deep learning issues

• Are you training enough?
• Try different mini-batch sizes
• Try different initializations
• Try different architectures

• Debug your deep learning system on small/simple problems first