TEXAS
The University of Texas at Austin

TOPIC 4 REINFORCEMENT LEARNING

Dynamic Programming

- There are a few problems with dynamic programming
  - You must know the distribution of all randomness
  - The state space could be huge
  - The action space could be huge
  - There may not be a way to solve the problem exactly
- With all this in mind we still want to be able to solve complicated dynamic programming problems
  - We'll have to make some approximations!
- In some communities this is called approximate dynamic programming, but the hipper term is reinforcement learning!

Dynamic Programming

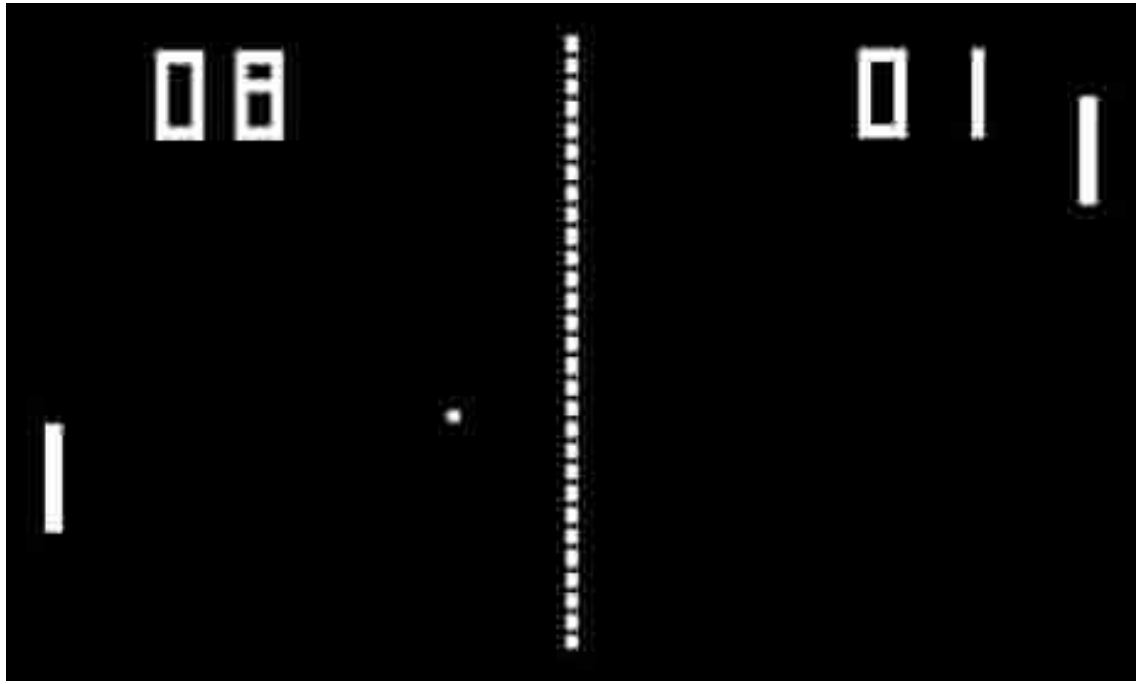
- Let's think back to the mining example we did in class
- Is it realistic to constrain the number of tons mined to be an integer?
- We made this assumption to make the problem solvable!
- What would we do to relax that assumption?
- At each time step we could use interpolation to get the value function for every (non-integer) value of s
- Then when we pick x, we use the interpolated value function to do the optimization

Dynamic Programming

- This idea only works because the state variable was only 1-dimensional
- What if the state had many dimensions
  - Location of all pieces on a chess board
  - Inventory of many products at a grocery store
- Then interpolation becomes hard
- We must do some type of approximation of the value function, like with a neural net!

Reinforcement Learning

- We're going to spend most of our time doing an example where we learn to play Pong
- But Dan, this isn't a business problem!
  - DeepMind was founded in 2010
  - By 2014 they had a general-purpose RL tool that could play most Atari games at super-human levels
  - They sold to google for $500M
  - The video game market generates ~$100Bn in revenue per year
- This is a great blog post on it
  - http://karpathy.github.io/2016/05/31/rl



Reinforcement Learning

- Although RL is good at playing games, it is good at other tasks too
  - Robots
  - Manufacturing automation
  - Portfolio construction
  - Traffic light management
  - Any dynamic program that's too hard to solve

Reinforcement Learning

- In reinforcement learning we interact directly with our environment to learn the optimal policy
  - Instead of knowing the true distribution of randomness we can just simulate from it – which is sometimes easier

Reinforcement Learning

- The general concept of RL is simple
  - Enact a random action policy
  - Interact with the environment
  - Remember which actions worked well and which didn't
  - Try to do more good actions and fewer bad actions
- The details of RL are not simple…
  - How do we actually implement more good actions?!?!

Reinforcement Learning

- Play a game of pong using OpenAI gym
- After each frame refresh, you can push a button or do nothing
- You also collect a reward after each frame (+1, 0, -1)
- We will think of a screen full of pixels as our state
  - Actually, the previous few screens full of pixels, so we can get a sense of motion!
- We want to feed the frames into a NN and have it tell us which button is best
- Then we push that button, and we get a new screen and reward
- We need to train that NN so that it gets better at telling us which button to push

Reinforcement Learning

- Instructions to get Atari working with python are on canvas
- First, let's look at a random game of pong in python with gym
- Then, let's see a game of Pong played by an RL agent that I trained for ~36 hours on my laptop