ML for Agroecology

Predicting optimal co-dependent crop planting techniques

Shannon Brownlee, Jordan Cahoon, Aryan Gulati, Megan Friedenberg, Leslie Moreno, Colin Ho, Felix Chen with faculty advisor Barath Raghavan

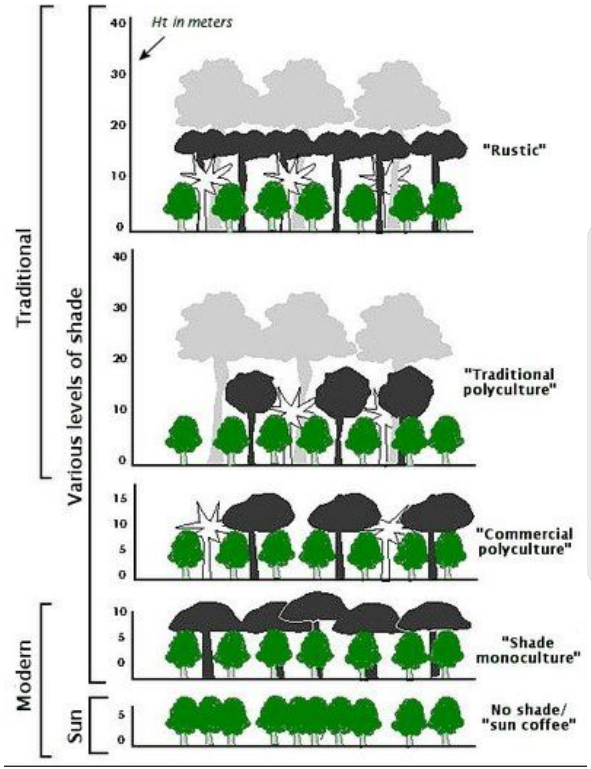
CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

Background

- Current methods of industrial agriculture are unsustainable
- Wild lands and local ecosystems required to be cleared for monoplant agriculture
- Single crop-planting leads do malnourished soil and lower outputs over time
- An alternative approach is to instead plant multiple crops within existing microclimates instead of altering them to better fit certain crops

Top to bottom: **Three sisters** (squash, corn, beans), industrialized agriculture

Purpose



Top-left to bottom right: shade-grown coffee, leafy green polyculture, three sisters

- Need efficient way to gather sustainable agriculture expertise
  - Most knowledge relies on indigenous knowledge systems
- Utilize reinforcement learning to implement an agent that determines optimal polycultures
  - Reduce barriers to implement sustainable practices
  - Better assess flora symbiotic relationships
- Professor if partnered with farmers in CA and HI who are already attempting their own polycultures

CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

Project Overview

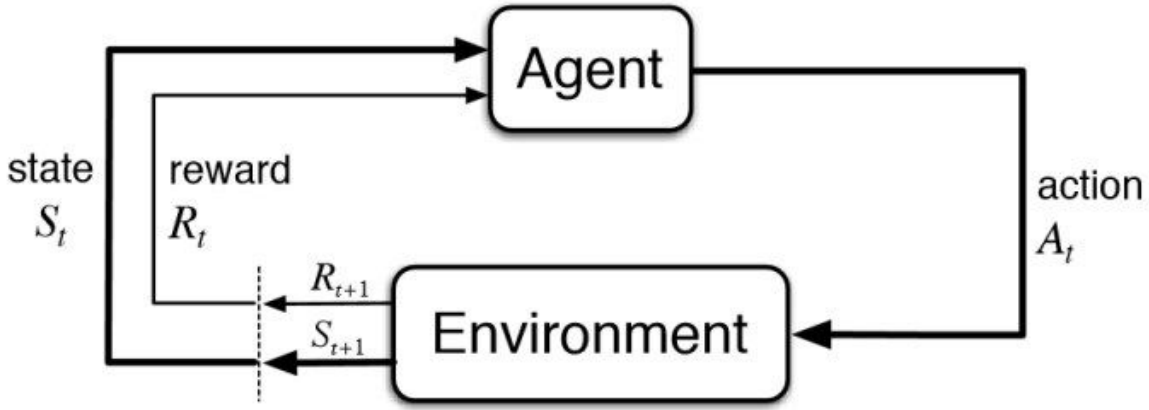
- We've created 3 main agents to tackle this problem and can compare their efficacies
  - Monte Carlo Tree Search
  - Genetic Algorithms
  - DeepQ
- We're hoping to make our model more similar to real life. To do this we're looking into
  - Using a 3D environment
  - Using Gridworld as an example game to give our agents
  - Creating a more holistic and detailed plant class

Modeling - Overview

- **Two categories of Reinforcement learning techniques**
  - Model-Based (neural networks)
    - Deep-Q
    - Genetic algorithms with neural network approximation
  - Model-free (algorithms)
    - Genetic algorithms
    - Graph search algorithms
    - Dynamic programming
- **Implemented functioning genetic and monte carlo tree search algorithms to practice interfacing with a simple environment**
  - Eventually transition to use Deep-Q model-based learning for graphics simulator

CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

Reinforcement Learning



- Agent: determines location and combination of plants at a specified time step
- Environment: polyculture field
- Reward function: calorie yield
  - Punishment: planting cost, crowding
  - Reward: diversity, symbiosis, plant maturity

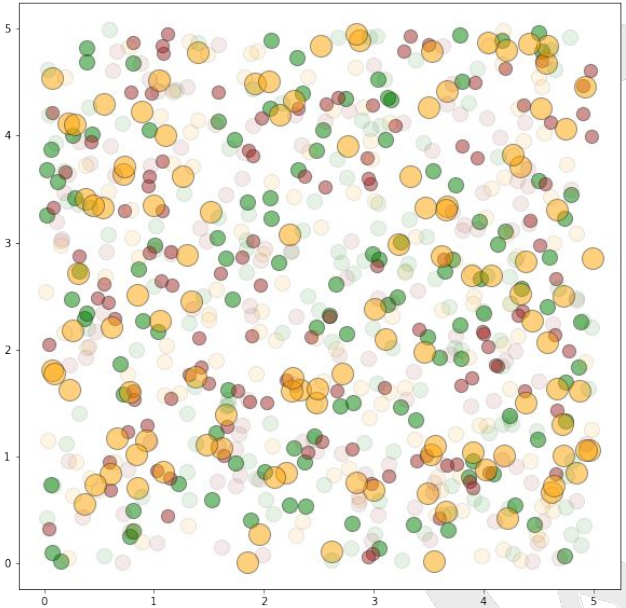
CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

Modeling

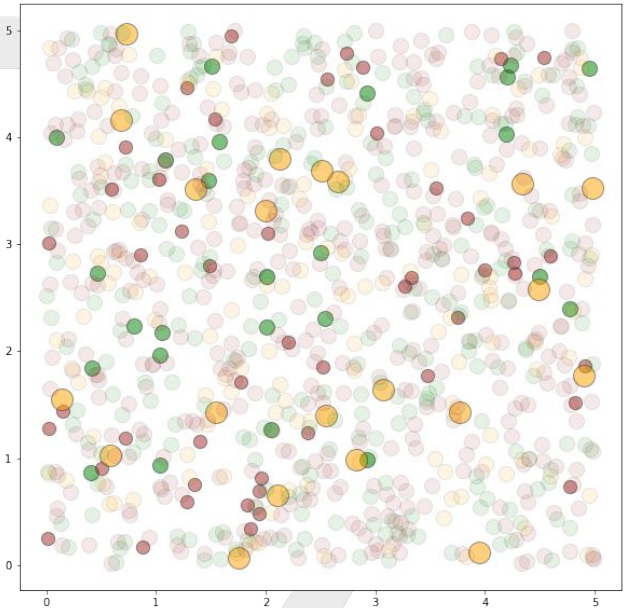
Sandbox Environment

- Plant class
  - Type (corn, beans, squash, etc.)
  - Age (squash~60 days, corn > 60)
- Plants crops on 2D field once maturity is reached
- Calorie reward at end of the season
  - Small crowding penalty
  - Three sisters reward
- One action = planting a specified number of crops
- Aim: have the highest caloric output at the end of 120 days

Example of Environment



Calorie yield: 492.1
Mature plants
Preliminary three-sister formation



Calorie yield: 106.1
Mostly immature plants
Randomized result

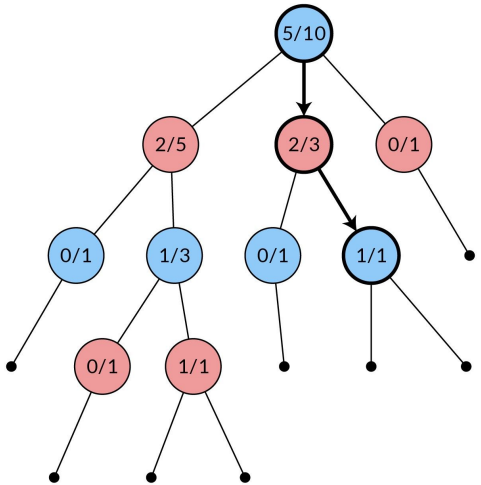
CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

Search Algorithms

CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

Search Algorithms: Why They're Essential

- The polyculture field environment is a large search space, and search algorithms with heuristics enable us to search more efficiently.
- Search algorithms are more transparent than machine learning models, and they give us a better understanding of the problem.
- Search algorithms, such as Monte Carlo Tree Search, can be used despite having incomplete information, benefiting our research due to the limited data we were working with

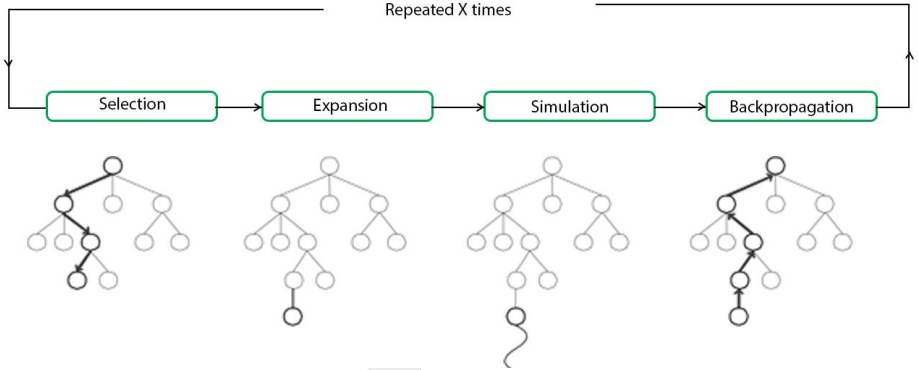


CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

Monte Carlo Tree Search (MCTS)

- Quick to optimize a large search space
- Simple to implement
- Four phases:
  - Traverse down the tree using Upper Confidence Bound until a leaf is reached
  - Expand leaf and add all action states to tree
  - Select an action and run monte carlo simulation (random actions) until end of season
  - Backpropagate and update UCB of path to action state

Algorithm overview



Upper confidence bound (Bandit Arm Algorithm)

$$S_i = x_i + C\sqrt{\frac{\ln(t)}{n_i}}$$

S_i = value of a node i
x_i = empirical mean of a node i (averaged reward along path)
C = a constant
t = time step
n_i = # of times node visited

CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

Q-Learning

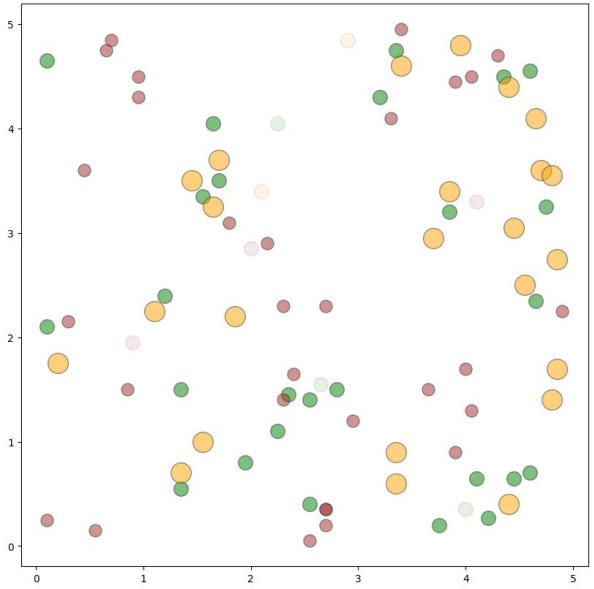
- Q-Learning: maximize policy (Q-value) to assess the next action
- Bellman equation uses a recursive relationship to define current Q-Value based on optimal future reward
    - α - learning rate
    - r - reward
    - γ - discount factor

$$\underbrace{Q(s_t, a_t)}_{\text{Q-value for current state-action pair}} = Q(s_t, a_t) + \alpha(r + \gamma(\underbrace{max_a Q(s_{t+1}, a_{t+1})}_{\text{Optimal future Q-Value}} - Q(s_t, a_t)))$$

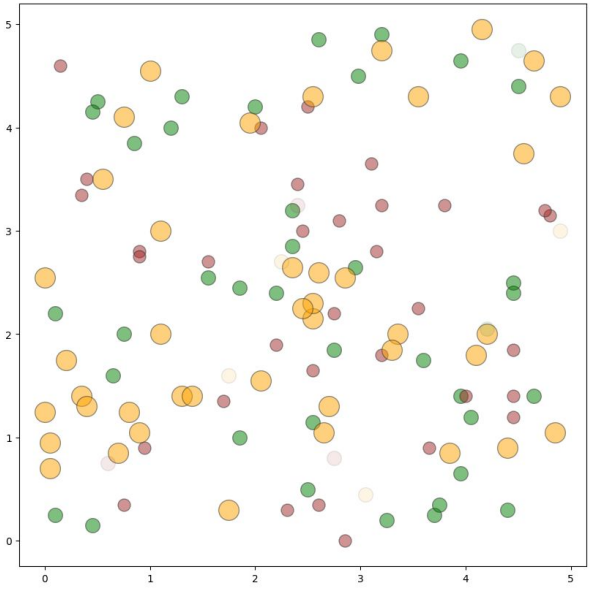
CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

Model Comparison

Random-Average: 147 calories



Q-Learning: 172 calories

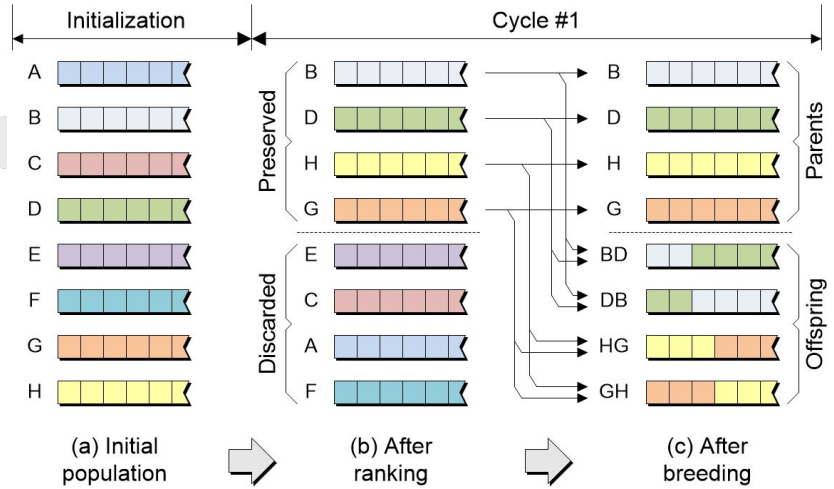


CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

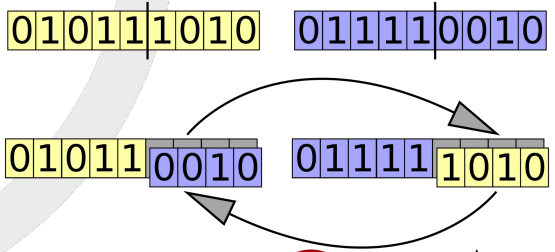
Genetic Algorithm

- Based on Darwin's Theory of Evolution and meiosis
- Five Phases
  - Initial population: Array of actions
  - Fitness function: calculates reward, impacts`probability of progressing to next generation
  - Selection: select two pairs of parent samples
  - Crossover: split samples at one point and swap to create children
  - Mutation: randomly mutate children to induce variation

Algorithm overview:



Example of crossover:



CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

Model-Based Reinforcement Learning

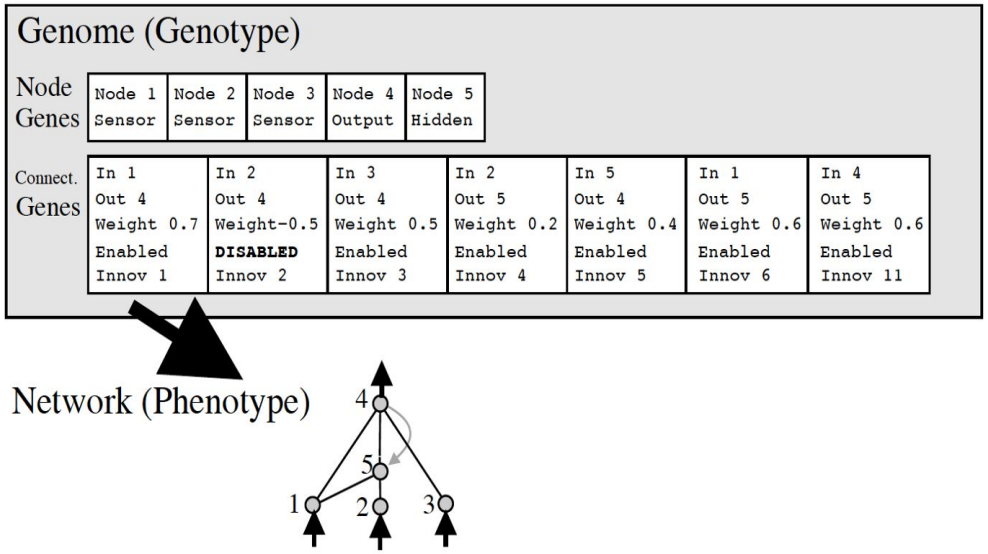
Model-based RL - Background

- Reinforcement learning is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward
- By using neural networks, we can reduce the amount of computation time because instead of looking at all possible setups and actions, we can calculate probabilities of rewards to come to more informed decisions quicker
- The main model we're looking into is DeepQ

NEAT Algorithm

- NeuroEvolution of Augmenting Topologies
- Uses the genetic algorithm process to evolve a neural network to solve the problem
  - GA process is used to optimize the weights + structure of the network

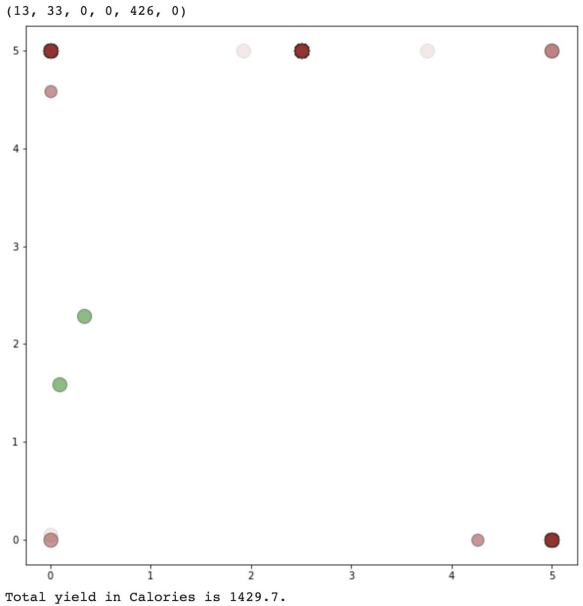
Algorithm overview:

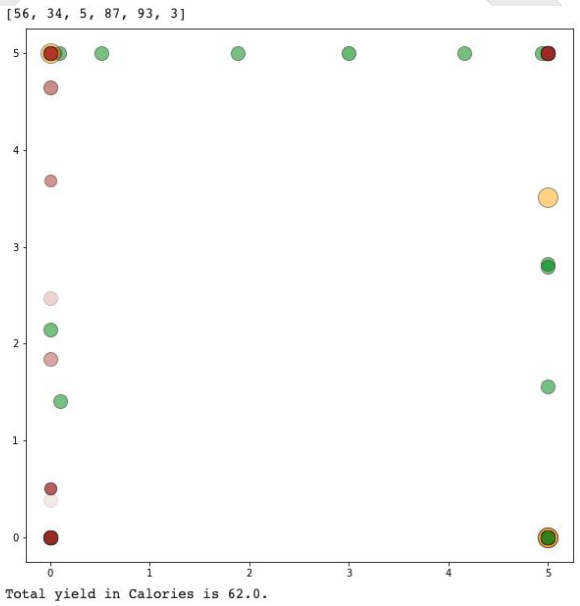


CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

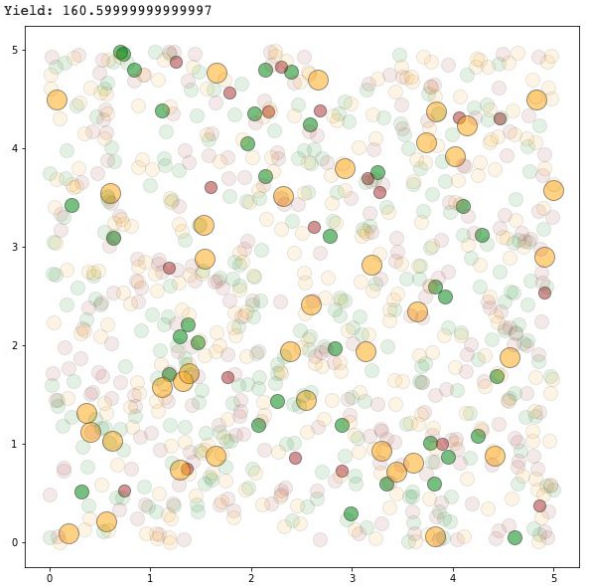
Genetic Algorithm Results

NEAT Algorithm:





Regular GA:



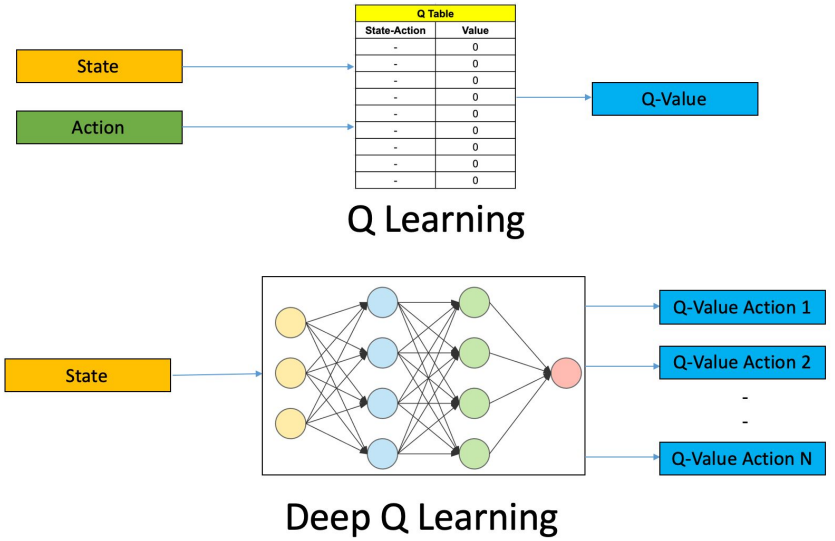
CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

Motivations for Deep Q Network

Drawbacks to Q-Learning

- Poor space complexity → Initialize Q-Table
- Simplify environment:
  - One plant for action (reduced Q-Table)
  - Lower maturity from 110 to 10 days

Deep Q Neural Networks use neural networks to optimize policy



Details on comparison and how we're going thru altering env + models

RL - Environment

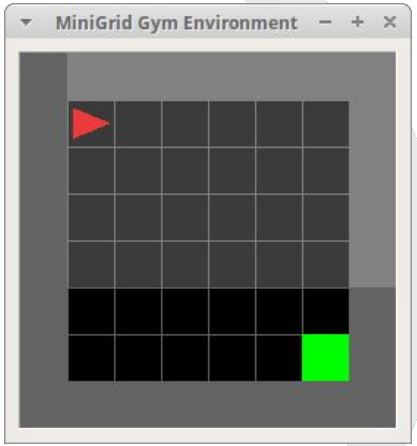
- Tried using OpenAI gym to quickly run simulations for the reinforcement algorithm
- Developing a 2D grid world where each gridspace is a crop planting
- Using an environment like this allows for multiple RL techniques to be done

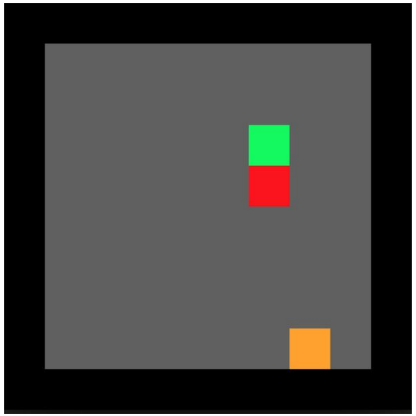
Downsides

- No configurations of gridworld not about finding maze solutions are available
- Difficult to let time consistently pass, or achieve more linear results
- Environment must be extremely tailored

RL - Environment Status

- Agent repeatedly tried to plant squash in the same square, despite crowding
- If there was no “goal” to reach, it was difficult to encourage the agent to move out of its square
- Will continue to explore other options and see how it can be modeled as a game





CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH

SEMESTER SUMMARY + NEXT STEPS

- Implemented basic agents, reviewed RL approaches, developed toy gym environment, and designed preliminarily grid world environment

Future Steps

- Putting environment and model together
  - Standardize environments for comparisons
  - Linking grid world with agents
- Machine Learning models
  - Optimizing results
  - Developing DQN
- Environment
  - Plant growth and mass
  - Managing sparse reward at end of season
  - Abstracting to 3D environments

Thanks for listening!

Please let us know if you have any questions or comments

“Only corn or only squash. That is ideal”

CAIS++
CENTER FOR AI IN SOCIETY'S STUDENT BRANCH