Description

First of all, we need to claim that this game is a simulation of a logging farm. Our main goal is to maximise the benefits of wood with minimal impact on the environment.

1. Basic mathematical settings

The basic mathematical settings are as follows:

State and transition: 10*10 cells for the planting position. Each cells can be recognised as -1 to 7 which is the age of the trees(-1 represent that the tree has just been cut off, and 0 means this cell is just seeded). The age of the trees grows naturally over time. For cells that equal 7, the next year they will seed all the cells around them(i.e., change state -1 to 0), they will not make any influence on the tree that already bean planted(i.e., state from 0-7 won't be changed at all)

Action: For each year, player can choose to cut down trees with one specific age from 1 to 7(i.e., At year 5, if player press number 6, that means all the 6-year-old trees will be cut down) or to do nothing.

AGE	-1	0	1	2	3	4	5	6	7
Profit	0	0	0.785	3.14	7.065	12.56	19.625	28.26	38.465

Reward: In this basic model, the reward is the profit of the trees that has been cut down. The relationship between value of the trees and its age is as follows:

Observation time: 16 years. If all cells are equal -1, the observation will end prematurely.

2. Algorithm:
2.1 Q learning

After Δt steps into the future the agent will decide some next step. The weight for this step is $\gamma^{\Delta t}$ calculated as , where γ (the *discount factor*) is a number between 0 and 1 ($0 \leq \gamma \leq 1$)and has the effect of valuing rewards received earlier higher than those received later (reflecting the value of a "good start"). γ may also be interpreted as the probability to succeed (or survive) at every step Δt.

$$Q : S \times A \to \mathbb{R}$$

The algorithm, therefore, has a function that calculates the quality of a state–action combination:

Before learning begins, Q is initialised to a possibly arbitrary fixed value (chosen by the programmer). Then, at each time the agent selects an action a_t , observes a reward r_t, enters a new state s_{t+1} (that may depend on both the previous state s_t and the selected action), and Q is updated. The core of the algorithm is a Bellman equation as a simple value iteration update, using the

$$Q^{new}(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\Big(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} - \underbrace{Q(s_t, a_t)}^{\text{old value}} \Big)}^{\text{temporal difference}}$$
$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{new value (temporal difference target)}}$$

weighted average of the old value and the new information:

In this case our default discount factor is 1, learning rate is 0.2 and the max episode length is 250.

2.2 Variant: Deep Q learning

The DeepMind system used a deep convolutional neural network, with layers of tiled convolutional filters to mimic the effects of receptive fields. Reinforcement learning is unstable or divergent when a nonlinear function approximator such as a neural network is used to represent Q. This instability comes from the correlations present in the sequence of observations, the fact that small updates to Q may significantly change the policy of the agent and the data distribution, and the correlations between Q and the target values.

The technique used experience replay, a biologically inspired mechanism that uses a random sample of prior actions instead of the most recent action to proceed. This removes correlations in the observation sequence and smooths changes in the data distribution. Iterative updates adjust Q towards target values that are only periodically updated, further reducing correlations with the target.

2.3 Policy Gradient

Policy gradient method start with a mapping from a finite-dimensional (parameter) space to the space of policies: given the parameter vector θ , let π_θ denote the policy associated to θ. Defining the performance function by

$$\rho(\theta) = \rho^{\pi_\theta},$$

under mild conditions this function will be differentiable as a function of the parameter vector θ. If the gradient of ρ was known, one could use gradient ascent.

3. Package

3.1 Gym

Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games.

3.2 Pygame

Pygame is an open-source-cross-platform library for the development of multimedia applications like video games using Python. It uses the Simple DirectMedia Layer library and several other popular libraries to abstract the most common functions, making writing these programs a more intuitive task.

3.3 numpy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

3.4 PyTorch

PyTorch is an open source machine learning framework based on the Torch library, used for application s such as computer version etc..

In this project we mainly use 2 Modules: Optim and nn. torch.optim is a model that implements various optimisation algorithms used for building neural networks. And module nn can be useful when raw autogrid is a bit too low-level for defining complex neural networks.

3.5 Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

4. Advanced Versions

At the beginning we wants to give agent multiple choices for each step, that means it can choose to cut tree down or not cell by cell. But unfortunately this will make the complexity much more complicated and the learning time may never terminate(Space Complexity=O(2^{100}),Time Complexity is unsolvable.). That's why we make the rule that it can only cut down the trees with same age.

Meanwhile, the main goal in this task is to protect environment. So we made some advanced version, which is more realistic. One of the common parameters we added is the value of greenhouse gas uptake(including CO2 sequestration etc.). This means each tree with different age has different value of GHG sequestration. This information is based on a book which is written by

E.Gregory McPherson and James R. Simpson. The origin figure is as follows:

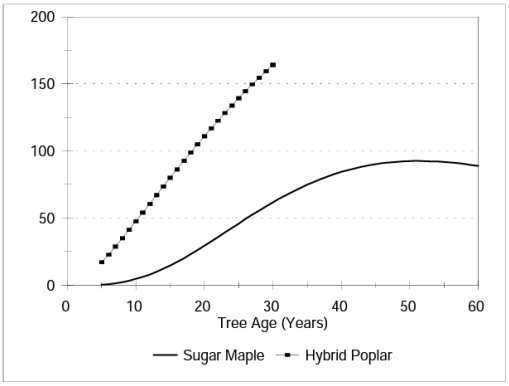

Figure 3—Growth rate and life span influence CO_2 sequestration. In this example, the total amount of CO_2 sequestered over 60 years by the slower growing maple (3,225 kg) is greater than the amount sequestered by the faster growing but shorter-lived poplar (2,460 kg). Growth curves and biomass equations used to derive these estimates are based on data from urban trees (Frelich 1992, Pillsbury and Thompson, 1995).

We can see the relationship between CO2 sequestration and tree age is close to linear. So we assume that:

Value of GHG=5 x Tree Age(Tree Age is between 0 to 7)

4.1 Limited value of GHG

In this version we decided to add a limit to the value of greenhouse gas uptake. According to the mentioned formula, the parameters can be set in following sheet:

AGE	-1	0	1	2	3	4	5	6	7
GHG Value	0	0	5	10	15	20	25	30	35

After “10 years” observation, we wants to make sure that the total amount of GHG sequestration would be more than 13000. Otherwise the normal reward would be 0.

4.2 Weighted Reward

In this version we decided to combine the timber reward with value of GHG. In this case we need to use weight function. The weight function is as follows:

reward function(weighted)= timber * weight_timber + greenhouse_gas * weight_greenhouse_gas

4.3 With environmental benefits

The Tree environment is trying to simulate the growth of trees in each cell, cutting down trees will gain monetary benefits, and retaining trees will gain environmental benefits (such as greenhouse gas uptake, or restoration of soil fertility), the goal is to find a balance between monetary benefits and environmental protection. The system show a land with a size of 10*10, action 0 means don't cut down any tree, action from 1-7 means cutting down the tree of the corresponding age.