Machine Learning aided Crop Yield Optimization

Chace Ashcraft, Kiran Karra
Johns Hopkins University/Applied Physics Lab
7701 Montpelier Rd
Laurel, Maryland 20723
kiran.karra@jhuapl.edu

Abstract
We present a crop simulation environment with an OpenAI Gym interface, and apply modern deep reinforcement learning (DRL) algorithms to optimize yield. We empirically show that DRL algorithms may be useful in discovering new policies and approaches to help optimize crop yield, while simultaneously minimizing constraining factors such as water and fertilizer usage. We propose that this hybrid plant modeling and data-driven approach for discovering new strategies to optimize crop yield may help address upcoming global food demands due to population expansion and climate change.

Introduction
Machine learning is enabling rapid progress in many areas of the physical and life sciences. However, its success depends on the availability of a large amount of data. For example, in computer vision, many databases of images (LeCun and Cortes 2010; Krizhevsky 2009; Deng et al. 2009; Netzer et al. 2011) have spurred innovation both within that field, and machine learning in general. Similarly, large datasets in human language technologies (HLT), including IMDB (Maas et al. 2011) for text, and LDC (of Pennsylvania 2021) for speech, have had similar effects on both the subfield and machine learning in a broader context. While these core subfields of machine learning have benefited from such large databases, the same cannot be said for machine learning aided agriculture.

Under many projections, Earth’s population is projected to increase over the next few decades (DESA 2019), and climate change is expected to adversely affect the way food is grown (EPA 2017). The application of machine learning to help accommodate for the projected population growth as well as adverse crop growing conditions through plant growth has been explored by both academia and industry. Precision agriculture seeks to address this, in part, by providing real-time data driven analytics to help farmers decide when to apply fertilization, water, and other necessities, or use computer vision algorithms to determine critical time points, for example when to harvest fruits or vegetables.

A less explored avenue, and one which concerns this manuscript, is active control. More specifically, we are interested in the scenario where machine learning operates in a feedback loop to dynamically control and optimize an agricultural system in operation. An example of this is the autonomous greenhouse challenge (Hemming et al. 2019, 2020). Here, the idea is to develop algorithms which monitor and control various greenhouse environmental aspects, including ventilation, irrigation, and temperature control, while maximizing crop health and yield. However, the aforementioned challenge uses an expensive greenhouse setup (provided by the competition sponsor), is not readily available to most researchers, and requires a full growing season to produce a single, small dataset that cannot be reused in “replay mode” (rewinded and altered – a common practice in machine learning to explore the state space).

An alternative approach is to use software to simulate environments in which algorithms can exert active control. Several simulators at varying levels of fidelity exist. For crop yield, the most notable one is the Python Crop Simulation Environment (PCSE), which is based on the LINTUL3 (Shibu et al. 2010) and WOFOST (Van Diepen et al. 1989) plant models. From a modeling perspective, PCSE has high fidelity, but its primary drawback is that all model parameters need to be provided a-priori. In particular, common farming tasks, such as irrigation and fertilization must be preplanned from seed sowing to harvest. The drawbacks of this from an algorithmic perspective are primarily that it is more difficult, for an ML algorithm to learn because of: 1) the lack of immediate feedback, and 2) the long time-horizon over which the algorithm needs to provide predictions. More recently, Overweg, Berghuijs, and Athanasiadis (2021) published CropGym, a reinforcement learning environment for growing wheat that utilizes the LINTUL3 (Shibu et al. 2010) crop model from PCSE. Their environment demonstrates the feasibility of using DRL to maximize harvestable biomass while minimizing fertilizer application. The work discussed here is similar, but uses a different crop model, problem formulation, and most importantly, a higher resolution time step to allow more fine grained algorithmic control over the environment. For these reasons, we believe it is still of interest to the community.

Given the necessary simulation environments and datasets, we believe that machine learning can help address the problem of increased demands on the agricultural system due to both human population growth and climate change. We believe that a growth, similar to the computer vision
revolution which followed the inception of ImageNet (Deng et al. 2009) is possible for machine learning aided agriculture. As a first step towards this front, we propose an easily accessible plant simulation environment [1] which can be used by machine learning practitioners and researchers without background expertise in agriculture, to develop policies which optimize the crop yield. In particular, this environment enables easy application of modern reinforcement learning techniques. Preliminary results using RLib’s (Liang et al. 2017) implementation of Proximal Policy Optimization (PPO) with default parameters suggest that DRL may be a successful method of learning useful control policies for precision agriculture. We hope this is a new direction in which both the DRL community and the precision agriculture community will take interest.

Plant Model

Crop models can be important tools for assessing the impact of climate change on crop production and yield. In this work, we utilize the SIMPLE crop model (Zhao et al. 2019) which includes 13 parameters to specify a crop type, with four of these for cultivar characteristics. Inputs required for the SIMPLE crop model include daily weather data, crop management, and soil water holding parameters. The model has several limitations, including the lack of response to vernalization and photoperiod effect on phenology. The model accounts for irrigation but ignores nutrient dynamics, an important component of precision agriculture. The model also makes assumptions about well drained soil, which can overlook the effects of over-watering on a crop. Regardless, Zhao et al. (2019) suggest that the model can be fairly accurate while being easy to implement. We use the SIMPLE model as a starting point for implementing a viable reinforcement learning environment simulating crop yield.

Reinforcement Learning

Reinforcement learning is a subfield within machine learning, in which an agent seeks to interact with an external environment in order to optimize a given reward function. In reinforcement learning, the environment is formally specified as a Markov decision process (MDP) \( \{ S, A, T, R \} \), where \( S \) is the state space, \( A \) is the set of actions available to the agent, \( T : S \rightarrow S \) is the transition function, and \( R : S \times A \rightarrow \mathbb{R} \) is the reward function. Fig. 1 depicts this in the context of growing crops, where the agent receives information regarding the state of the crops (for example, crop health, growth rate, etc.) and any rewards, and takes relevant actions (for example, irrigating, fertilizing, etc.) in the environment to achieve a user objective (for example, maximizing yield).

For this effort, we utilize an implemented version of PPO (Schulman et al. 2017) from the well established and vetted library RLib (Liang et al. 2017). PPO is a modern reinforcement learning algorithm which is known to work well “out-of-the-box”. PPO seeks to maximize the agent’s reward by taking the largest possible improvement step on the policy during training, while minimizing catastrophic performance collapse often associated with large training steps. This allows PPO to train quickly, while maintaining good performance. In order to automatically take advantage of these reinforcement learning algorithms, we first wrap our SIMPLE plant environment into OpenAI’s Gym interface (Brockman et al. 2016). This is simply a standardized software interface for RL environments, and enables easy interfacing to RL algorithm implementations, including PPO.

Environment Description

Our environment is configurable to use each of the parameters specified by the SIMPLE model. We set the time step of the environment to a single day, to match the SIMPLE model. At each time step, the agent receives an observation consisting of a noisy reading of plant’s biomass, the cumulative daily mean temperature since sowing, the number of days since sowing, the day’s weather (as given in Table 1), a noisy reading of the next day’s weather, and a noisy reading of plant available water (PAW). We define noisy to mean sampled from a Gaussian distribution with the actual state value as the mean and a given standard variation. The agent action is a single continuous value in the interval \([0, 100]\) representing the amount of water to apply, in millimeters (mm), to the crop. The reward function is defined as follows:

\[
R(s, a) = \begin{cases} 
C_I \times a + Y & \text{at harvest date} \\
C_I \times a & \text{otherwise} 
\end{cases}
\]

where \( s \) is the current state, \( a \) is the current action and \( Y \) is the final yield in metric tons per hectare. \( C_I \) is a hyper-parameter representing the cost to irrigate, which we set to \( C_I = -1.25 \times 10^{-2} \). The goal of PPO in this case is to maximize the reward, which is positively correlated with the yield, \( Y \), and negatively correlated with the amount of water used, while accounting for the noisy state observations.

Preliminary Results

We use the parameters for Russet Potatoes from Benton, Washington, USA given in (Zhao et al. 2019), weather data from Benton starting April 1st, 2000\(^2\) and standard deviations for weather parameters given in Table 2.

Using RLib’s implementation of PPO with default parameters and default neural network architecture, we observe the mean reward growth curves shown in Figure 2. The source code can be found at: https://github.com/iscoe/croprl

---

1Source code can be found at: https://github.com/iscoe/croprl

2From https://weather.wsu.edu
Table 1: Weather parameters used by our environment.

| Parameter | Definition                                      |
|-----------|-------------------------------------------------|
| TMAX      | Daily maximum temperature (°C)                  |
| Tmin      | Daily minimum temperature (°C)                  |
| TAVG*     | Daily mean temperature (°C)                     |
| Rain      | Daily rainfall (mm)                             |
| Solar Rad | Daily solar radiation (MJ/m²)                   |
| CO₂       | Atmospheric CO₂ concentration (PPM)             |
| VAP**     | Average daily vapor pressure (hPa)              |
| Wind**    | Average wind speed at 2 meters (m/s)            |

*Not specified as weather variable in SIMPLE, but used in our model.

**Not used in original SIMPLE model, but used to compute reference evapotranspiration in ours using the Penman-Monteith method (Allen et al. 1998).

Table 2: Standard deviations used to add noise to observation variables to create the next day’s weather forecast.

| Variable | SD     |
|----------|--------|
| Biomass  | 1      |
| Rain     | 3      |
| PAW      | 1      |
| TMAX     | 3      |
| TMIN     | 2      |
| TAVG     | 2      |
| Solar Rad| 1      |
| CO₂      | 1      |
| VAP      | 1      |
| Wind     | 6      |

Initial indications of learning are promising, but note that performance seems to plateau sub-optimally. Intuitively, optimal performance should be attained by spending the least on irrigation as possible while keeping the soil saturated. For example, using a constant action of $10 \text{ mm} / \text{m}^2$ irrigation leads to net normalized reward of 0.976 in the current implementation, while the PPO agents seems to plateau closer to 0.55.

Future Work

Given the very early nature of this project, there are many tasks future work could, or should, address. We propose two general directions we are interested in: crop modeling and advanced DRL.

For crop modeling, we should further validate the correctness and accuracy of our model using data from the field or real-world data collected from crop growing experiments. Validation is important for obtaining useful results for applied agriculture and understanding the DRL performance. We would also find it useful and worthwhile to further augment the model to account for other agent actions, such as the application of fertilizer and deciding when to harvest. Another useful improvement would be to improve the model to account for other realistic conditions, such as overwatering, or implement a more complex model entirely, such as those used by PCSE (de Wit 2021). Finally, finding and using more accurate simulation parameters, such as for daily weather, would also make our simulations more realistic.

Future work in DRL could include the application of different algorithms to the simulator, attempting hyperparameter tuning, and experimenting with various neural network architectures. Each of these can significantly affect the performance of DRL on a task. Additionally, many other observation representations and reward functions are possible, some of which may facilitate faster learning. Lastly, because DRL tends to require significant amounts of data, and real-world data is typically difficult to obtain, we believe application of techniques such as offline reinforcement learning or meta-learning would be interesting to explore.

Conclusion

We put forth the idea that with relevant datasets and software simulation environments, the problem of supporting future growing populations under climate change can be possible and furthermore aided by machine learning. As a proof of concept for this idea, we showcase a simple plant simulation model with an OpenAI Gym interface, and train an agent using a modern RL algorithm to learn a policy for optimizing crop yield. We seek to continue this work by building better plant models to increase the fidelity of our simulation and make its policies more relevant to real-world conditions.

References

Allen, R. G.; Pereira, L. S.; Raes, D.; Smith, M.; et al. 1998. Crop evapotranspiration—Guidelines for computing crop water requirements—FAO Irrigation and drainage paper 56. Fao, Rome 300(9): D05109.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym. arXiv preprint arXiv:1606.01540.

de Wit, A. 2021. Python Crop Simulation Environment. URL https://pcse.readthedocs.io/en/stable/index.html.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, 248–255. Ieee.

DESA, U. 2019. World Population Prospects 2019. United Nations. Department of Economic and Social Affairs. World Population Prospects 2019.
