Reinforcement Learning

Haoteng Zhao, Liping Di*, Liying Guo, Lin Li, Chen Zhang, Eugene Yu, Hui Li
Center for Spatial Information Science and Systems (CSISS), George Mason University
Fairfax, VA 22030, USA
* ldi@gmu.edu

2023 11th International Conference on Agro-Geoinformatics (Agro-Geoinformatics) | 979-8-3503-0351-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/Agro-Geoinformatics59224.2023.10233673

***Abstract*—task of irrigation scheduling involves sequentially establishing both the timing and quantity of irrigation to be administered to the field throughout the course of the growing season. This task can be conceptualized as a Markov decision process. Reinforcement learning (RL), a machine learning approach that leverages rewards acquired through interactions with the environment to steer behavior and progressively develops a strategy to maximize cumulative rewards, is well-suited for managing sequential decision-making processes such as irrigation scheduling. Deep RL, a combination of RL with deep learning techniques, has the potential to offer novel solutions to intricate cognitive decision-making challenges in intricate states. In this research, a deep RL-based irrigation scheduling approach will be presented to enhance the optimization of economic return in irrigation applications. This method involves computing the irrigation quantity for each step while taking evapotranspiration (ET), soil moisture, future precipitation probability, and the current crop growth stage into consideration. The simulation results showed a significant improvement in economic return, 5.7% and 17.3% for a wet season and a dry season, respectively, while water-saving effect is similar to conventional threshold-based methods.**

Keywords— reinforcement learning, optimization, water balance, irrigation scheduling, economic return

I. INTRODUCTION

Agricultural irrigation accounts for approximately two-thirds of worldwide freshwater usage[1], thereby escalating the strain on water resources due to population expansion and climate fluctuations [2-4]. Therefore, a pressing requirement exists to enhance the efficiency of irrigation systems through optimization aiming to conserve water resources, despite water in soil being curial for crop growth and high yield [5, 6]. Historically, farmers have commonly relied on traditional irrigation scheduling. This approach entails applying a consistent volume of irrigation water within specific time intervals. While straightforward, this method frequently leads to water wastage and diminished crop productivity. However, recent times have witnessed substantial exploration of more accurate irrigation strategies rooted in sensor data.[7]. Nonetheless, the majority of these approaches rely on thresholds or rudimentary models for decision-making, leading to numerous instances of imprecise or suboptimal irrigation occurrences. In the case of threshold-centered irrigation scheduling, an expert's involvement is required to translate sensor data into suitable threshold values for utilization within a scheduling model. This undertaking can become intricate due to the multitude of zones, swiftly shifting weather conditions, variances in soil composition, diverse crop varieties, and the varying water requirements across different growth stages [8]. It's also hard for these methods to find the optimized action in the term of long run return. In addition, the abundance of sensor data further compounds the complexity of real-time scheduling, potentially leading to conflicting information from various sensor types and other data origins [9]. An evident limitation of employing manually calculated thresholds or models is their inherent time-consuming nature and restricted scalability. To address these issues, researchers have delved into the realm of machine learning techniques to automate the process [10, 11]. Techniques such as linear regression or neural networks are harnessed to distill valuable insights from sensor data and construct the scheduling model. Nevertheless, even with the integration of these techniques, human supervision remains imperative for meticulous result analysis and the manual regulation of irrigation applications.

Irrigation decisions entail determining when and how much irrigation is needed, akin to a Markov decision process where the future state relies solely on the present state [12]. Reinforcement learning (RL) is well-suited for such processes due to their Markov property.[13]. Inspired by behavioral psychology[14], RL proposes a formal framework to the sequential decision-making problem. The core concept involves an artificial agent learning through interactions with its environment, much like a living organism. Leveraging accumulated experience, the artificial agent aims to optimize specified objectives, depicted in Figure 1, through cumulative rewards. This approach can be applied to a wide range of sequential decision-making problems that depend on past encounters..

This paper presents a Deep Q-Network (DQN) based RL approach[15] for irrigation scheduling using High-Resolution Land Data Assimilation System (HRLDAS)[16, 17] simulated soil moisture and ET. Nebraska, a typical crop state with large demand of irrigation, was selected as study area for simulation.

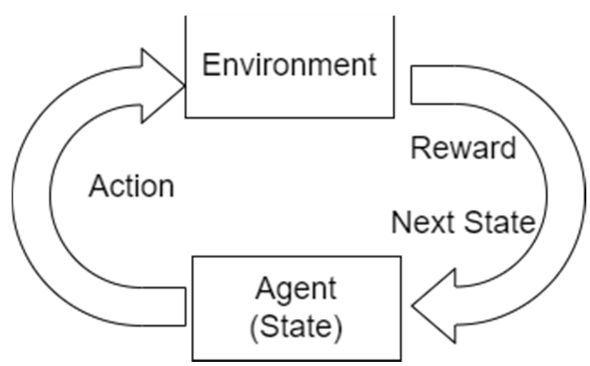


Figure 1. Interaction between agent and environment in reinforcement learning.

II. Deep Reinforcement Learning for Irrigation Scheduling

The decision-making process for irrigation exhibits the Markov property, where the determination of whether to irrigate is influenced by both the soil moisture and the crop growth condition during growing season. The current state of the soil and crop only depends on previous state of the soil, weather, crop, and previous irrigation action. As a result, the application of RL was employed to address irrigation scheduling that incorporates the Markov property.

Conventional reinforcement learning, such as Q learning, involves maintaining a Q table with dimensions (s, a), where s is the number of states and a is the number of actions, while learning from data. Fundamentally, a Q-table maps state and action pairs to a Q-value. However, in the irrigation decision scenario, the soil moisture and precipitation are not distributed discretely, meaning the number of states could be infinite, making it computationally intractable to build a table. A deep Q-learning network algorithm was developed by integrating deep neural network (DNN) into RL to tackle the continuous state space and expedite learning.

Rather than a Q-table for mapping state and action pairs to a Q value, a Q function is approximated by learning parameters (weights) of a DNN which we call a Q-network such that it can generate the optimal Q values. The fundamental concept behind a Deep Q Network is closely akin to the Q Learning algorithm. It initializes with arbitrary Q-value estimates, exploring the environment through an ε-greedy policy. Its core employs a similar notion of dual actions—current action with current Q-value and target action with target Q-value—within its update logic to enhance Q-value estimates. The integration of DNN enables handling vast amounts of observational data, rendering the proposed irrigation approach scalable.

For neural network training to achieve convergence and stability, DQN introduced a technique known as "experience replay." This method disentangles data connections and optimally utilizes historical data samples. The experience at each time-step, $e_t = (s_t, a_t, r_t, s_{t+1})$ is stored in a limited-size replay memory. When the maximum capacity is reached, the freshest experience replaces the oldest one. The aim of experience replay is to employ mini-batch training for the Q-network, approximating the Q function through samples from the replay memory. Enhancing training performance, a method known as combined experience replay (CER) is also employed. CER ensures the incorporation of the most recent experience within the samples.

For an irrigation agent, in every time step, the agent selects an action (a_t) from the set of legal actions, $A = \{a_0, a_1, a_2, \dots, a_k\}$, with each action corresponding to a specific irrigation volume. This action is then communicated to an environment connected with both AquaCrop[18] and HRLDAS model to compute the subsequent state and the reward. In the framework of DQN, the ongoing observation of the environment using HRLDAS simulations (x_t) is used to indicate the current state (s_t). The goal of the irrigation agent is to interact with the actual environment and the AquaCrop model, making action choices in a way that maximizes long-term gains. Utilizing the prevailing state and selected action, the environment emulator calculates the upcoming state and the ensuing reward, as depicted in Figure 2.

The algorithm is shown in Table 1. Initially, the replay memory and Q network are initialized. Each training episode corresponds to a complete crop season. During each time step within an episode, there's a probability of selecting a random irrigation action; otherwise, the action that can maximize the Q value function is chosen. Subsequently, the

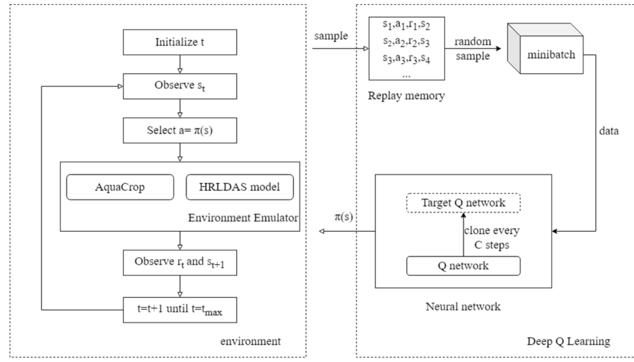


Figure 2. Architecture of DQN-based irrigation scheduling.

chosen action and its interaction with the environment emulator are employed to compute the current reward and ascertain the subsequent state. The transition (s_t, a_t, r_t, s_{t+1}) is preserved within replay memory. Ultimately, experience replay is executed by extracting a random mini-batch of transitions from the replay memory and carrying out a gradient descent step. The environment emulator, illustrated in Figure 2, is the module that interacts with AquaCrop and HRLDAS. HRLDAS is run first to generate simulated soil moisture and ET after irrigation is implemented. Subsequently, it forwards this data to AquaCrop and initiates a single cycle of crop simulation. Upon completion of the simulation, it retrieves the simulation outcomes and employs pertinent data to compute the reward and subsequent state values. The precise architecture of the neural network is as follows: the input layer is a matrix comprising feature vectors of stowage samples, while the output layer represents the approximated Q value for each action. Consequently, the input layer encompasses 9 nodes, and the output layer contains 42 nodes. In this configuration, there are two hidden layers, both fully-connected, each comprising 24 units..

TABLE I. Procedures of the DQN-Based Irrigation Scheduling Algorithm.

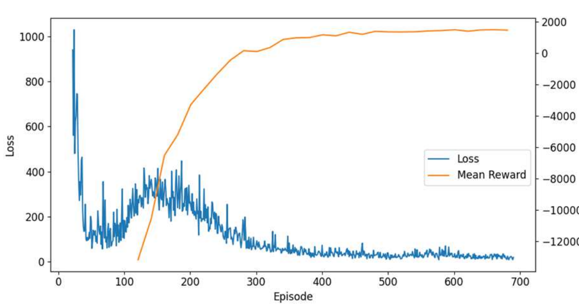
1: Initialize replay memory *M* to capacity *N*;
2: Initialize Q network with random weights θ;
3: Initialize target network with random weights $\theta' = \theta$;
4: **for** *episode* = 1 to *E* **do**
5: Initialize batch size;
6: Collect the environmental condition and initialize state s;
7: **while** *Crop growing season is not end* **do**
8: With probability ε select a random action a_t;
9: Otherwise select $a_t = max_a Q^*(s_t, a_t; \theta')$;
10: Execute irrigation action in the Environment Emulator;
11: Observe reward r_t and next state s_{t+1};
12: Store transition (s_t, a_t, r_t, s_{t+1}) into *M*;
13: **if** size of *M* > size of minibatch
14: Sample random minibatch of (s_i, a_i, r_i, s_{i+1}) from M;
15: set $y_i =$
$$\begin{cases} r_i & if\ s_{i+1}\ is\ terminal\ state \\ r_i + \gamma \max_{a_{i+1}} Q(s_{i+1}, a_{i+1}; \theta') & otherwise \end{cases}$$
16: Perform a gradient descent step on $\left(y_i - Q(s_i, a_i; \theta)\right)^2$ according to θ;

Every C steps clone $\theta' = \theta$;
if $\varepsilon > \varepsilon_{min}$: $\varepsilon \leftarrow \varepsilon * \varepsilon_{decay}$
End for
End for

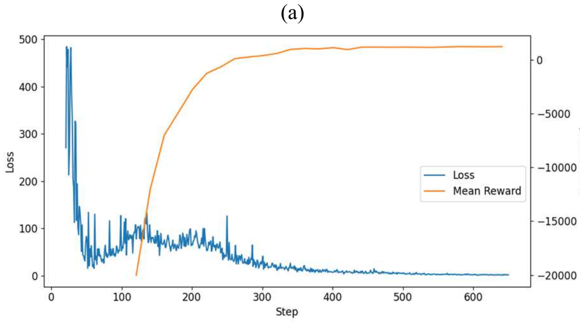
III. Results and Discussion

The loss signifies the value of the objective function during neural network training, indicating how closely the neural network approximates the discrete action value. As illustrated in Figure 3, the initial loss is substantial due to insufficient information for satisfactory approximation, and it subsequently diminishes rapidly within 50 iterations. Then the loss increased and floated around in a high value before 200 iterations as the exploration strategy is more adopted in the beginning episodes although the network already learned some good policy. After approximately 300 -500 iterations, as the parameters of each iteration exhibited minor differences, the loss value demonstrated a tendency to stabilize and fluctuate. This trend indicated that the neural network was progressively aligning itself with the action value across diverse states.

Slightly different from what was anticipated, the mean reward started from a large negative value and then increased dramatically to a positive normal value. This phenomenon stems from the fact that initially, the exploration strategy was more prominently employed (ε was close to 1 at beginning), leading to the acquisition of a sufficient range of reward values for every state. This increased exploration was prone to result in actions that yielded lower rewards, often a negative value as in this case costs on the irrigation action have a strong chance to exceed the gain in yield. Subsequently, a shift towards the exploitation strategy occurred, favoring actions with the highest rewards (ε decayed to the minimum ε). After approximately 200-300 iterations, a noticeable and enduring rise in mean rewards was observed, settling above 0.



(a)



(b)

Figure 3. Training curve tracking the loss and mean reward for (a): Kelly 2019, and (b): Kelly 2020.

The results of the DQN irrigation scheduling are contrasted with those of the conventional threshold-based strategy and Q learning method. Results of real irrigation and no irrigation cases are also included for reference. The Table 2 shows the amount of irrigation water, yield, and the economic returns of different approaches in different seasons.

Table 2 shows that the DQN irrigation scheduling could further improve the economic return by saving irrigation water and increasing yield compared with threshold-based methods and real irrigation scheduling. The economic return improved 5.7% and 17.3% in wet season and dry season, respectively, indicating a significant difference among different weather conditions. During a wet season with abundant rainfall, corn only needed minimal irrigation to grow properly and reached full potential of yield; thus, both threshold-based and reinforcement learning based irrigation scheduling methods could only take effect on water-saving as the yield in real case was close to its full potential. However, during a dry season, although Q-learning and DQN irrigation scheduling showed an inferior ability in water-saving comparing to threshold-base methods, a significant increase in yield greatly improves the total economic return. This is because the reward function was mainly designed for maximizing the economic return, resulting more water irrigated if it's necessary for improving the economic return.

TABLE II. Comparison of irrigation methods for corn field Kelly in different season.

Season and methods	Yield (ton/ha)	Irrigation Water (mm)	Irrigation Times	Net Return (USD)
2019 (Wet Season)				
Deep Q networks	13.525	131	10	3477.4
Q learning	13.45	122.5	10	3470.0
ET-WB based	13.3	130.3	11	3405.52
SM based	13.3	121.8	11	3419.12
Ground Truth	13.4	206.8	13	3291.12
No irrigation	11.1	0	0	3108.0
2020 (Dry Season)				
Deep Q networks	14.865	243	11	3663.4
Q learning	14.725	237.5	11	3633.0
ET-WB based	13.6	226.7	10	3345.28
SM based	13.6	222.5	10	3352.0
Ground Truth	13.7	278.4	17	3220.56
No irrigation	9.4	0	0	2632.0

Although the DQN irrigation scheduling is not always the one that saves most water in irrigation comparing to Q learning and the conventional approaches, but it successfully achieves the highest economic return which we set as the goal of irrigation scheduling. Generally, the conventional approaches performs well in water-saving as they are designed to eliminate most unnecessary water. However, with deep reinforcement learning, the similar amount water can be better allocated on proper dates during the whole growing season to reach a higher yield or higher economic return.

IV. Conclusion and Future Work

In summary, our research has explored the application of deep reinforcement learning for optimizing irrigation scheduling in agriculture. The combination of reinforcement learning and deep neural networks in deep reinforcement learning allows for the effective approximation of the state-action pair value (Q value) and the solution of the sequential decision-making problem for maximizing final rewards based on the Markov property.

To further investigate the potential of deep reinforcement learning in irrigation scheduling, we analyzed threshold-based approaches and rainfall forecast integration. Building on our findings, we developed a DQN algorithm-based irrigation scheduling model that optimizes irrigation actions for a best seasonal reward. The rapid convergence of the training process of the DQN algorithm showcases the efficacy of our model.

Our simulated results indicate that our DQN strategy can conserve irrigation water by 20-40% compared to real irrigation decisions, which is comparable to conventional threshold-based approaches. However, our DQN strategy offers a much higher yield increase, resulting in the goal of maximizing economic return being achieved, which was increased by 5.7% and 17.3% for wet season and dry season, respectively.

Our approach provides a promising avenue for optimizing irrigation scheduling and reducing water waste in agriculture. Further research can explore the potential of deep reinforcement learning in addressing other challenges in agriculture and beyond, taking advantages of recent progress in in-season crop mapping[19-23] and agricultural data availability[24-27].

ACKNOWLEDGMENT

This research was supported by NSF WaterSmart project (CNS-1739705, PI: Liping Di).

REFERENCES

[1] C. A. Dieter, *Water availability and use science program: Estimated use of water in the United States in 2015*. Geological Survey, 2018.

[2] L. Lin *et al.*, "A review of remote sensing in flood assessment," in *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 2016: IEEE, pp. 1-4.

[3] L. Lin *et al.*, "Improvement and validation of NASA/MODIS NRT global flood mapping. Remote Sens. 11, 205," ed, 2019.

[4] M. S. Rahman *et al.*, "Impact of climate change on soil salinity: a remote sensing based investigation in coastal Bangladesh," in *2018 7th International Conference on Agro-geoinformatics (Agro-geoinformatics)*, 2018: IEEE, pp. 1-5.

[5] H. Zhao *et al.*, "Impacts of soil moisture on crop health: A remote sensing perspective," in *2021 9th International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 2021: IEEE, pp. 1-4.

[6] W. R. Walker, *Guidelines for designing and evaluating surface irrigation systems*. 1989.

[7] Z. Gu, Z. Qi, R. Burghate, S. Yuan, X. Jiao, and J. Xu, "Irrigation scheduling approaches and applications: A review," *Journal of Irrigation and Drainage Engineering,* vol. 146, no. 6, p. 04020007, 2020.

[8] H. Zhao, L. Di, and Z. Sun, "WaterSmart-GIS: A Web Application of a Data Assimilation Model to Support Irrigation Research and Decision Making," *ISPRS International Journal of Geo-Information,* vol. 11, no. 5, p. 271, 2022.

[9] C. Zhang *et al.*, "Crop-CASMA: A web geoprocessing and map service based architecture and implementation for serving soil moisture and crop vegetation condition data over US Cropland," *International Journal of Applied Earth Observation and Geoinformation,* vol. 112, p. 102902, 2022.

[10] S. Tsang and C. Y. Jim, "Applying artificial intelligence modeling to optimize green roof irrigation," *Energy and Buildings,* vol. 127, pp. 360-369, 2016.

[11] E. G. Yu *et al.*, "Full Stack Web Development of a Geospatial Information Service System for Intelligently Irrigated Agriculture," in *2019 8th International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 2019: IEEE, pp. 1-6.

[12] E. Lee, K. Raju, and A. W. Biere, "Dynamic irrigation scheduling with stochastic rainfall," *Agricultural water management,* vol. 19, no. 3, pp. 253-270, 1991.

[13] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[14] R. S. Sutton, *Temporal credit assignment in reinforcement learning*. University of Massachusetts Amherst, 1984.

[15] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602,* 2013.

[16] F. Chen *et al.*, "Description and evaluation of the characteristics of the NCAR high-resolution land data assimilation system," *Journal of applied Meteorology and Climatology,* vol. 46, no. 6, pp. 694-713, 2007.

[17] H. Zhao, L. Di, Z. Sun, E. Yu, C. Zhang, and L. Lin, "Validation and Calibration of HRLDAS Soil Moisture Products in Nebraska," in *2022 10th International Conference on Agro-geoinformatics (Agro-Geoinformatics)*, 2022: IEEE, pp. 1-4.

[18] P. Steduto, T. C. Hsiao, D. Raes, and E. Fereres, "AquaCrop—The FAO crop model to simulate yield response to water: I. Concepts and underlying principles," *Agronomy Journal,* vol. 101, no. 3, pp. 426-437, 2009.

[19] C. Zhang *et al.*, "Rapid in-season mapping of corn and soybeans using machine-learned trusted pixels from Cropland Data Layer," *International Journal of Applied Earth Observation and Geoinformation,* vol. 102, p. 102374, 2021.

[20] C. Zhang *et al.*, "Towards automation of in-season crop type mapping using spatiotemporal crop information and remote sensing data," *Agricultural Systems,* vol. 201, p. 103462, 2022.

[21] L. Lin *et al.*, "Validation and refinement of cropland data layer using a spatial-temporal decision tree algorithm," *Scientific Data,* vol. 9, no. 1, p. 63, 2022.

[22] H. Li, L. Di, C. Zhang, L. Lin, and L. Guo, "Improvement of In-season Crop Mapping for Illinois Cropland Using Multiple Machine Learning Classifiers," in *2022 10th International Conference on Agro-geoinformatics (Agro-Geoinformatics)*, 2022: IEEE, pp. 1-6.

[23] Z. Yu *et al.*, "Selection of landsat 8 OLI band combinations for land use and land cover classification," in *2019 8th International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 2019: IEEE, pp. 1-5.

[24] P. Hao, L. Di, E. Yu, L. Guo, Z. Sun, and H. Zhao, "Using machine learning and trapezoidal model to derive All-weather ET from Remote sensing Images and Meteorological Data," in *2021 9th International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 2021: IEEE, pp. 1-4.

[25] C. Zhang *et al.*, "Integrating OGC Web Processing Service with cloud computing environment for Earth Observation data," in *2017 6th International Conference on Agro-Geoinformatics*, 2017: IEEE, pp. 1-4.

[26] Z. Sun *et al.*, "GeoFairy: Towards a one-stop and location based Service for Geospatial Information Retrieval," *Computers, Environment and Urban Systems,* vol. 62, pp. 156-167, 2017.

[27] M. S. Rahman *et al.*, "Comparison of selected noise reduction techniques for MODIS daily NDVI: An empirical analysis on corn and soybean," in *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 2016: IEEE, pp. 1-5.