APPLICATION OF DEEP REINFORCEMENT LEARNING (DRL) IN UAV

Buriyev Sardor Norovich,
Assistant Teacher of Optical Communication Systems and Network Security Department, Karshi branch of the TUIT
Polvonov Kholbek Nematovich,
Senior Lecturer of Optical Communication Systems and Network Security Department, Karshi branch of the TUIT
Tursunov Akhmadjon Akmal ugli, student
Karshi branch of the Tashkent University of Information Technologies

Abstract

Today, artificial intelligence (AI) is one of the most common areas of research in combat unmanned aerial vehicles (UAVs). AI integration in a UAV can enhance it and various capabilities, particularly decision-making and target detection. We consider the principle of operation of the most optimal algorithm. In this article, we will describe the successes and achievements that can be achieved in unmanned aerial vehicles (UAV) using this algorithm.

Keywords: Artificial intelligence (AI), unmanned aerial vehicle (UAV), decision-making, Deep Reinforcement Learning (DRL), Deep Q-Networks (DQN)

Аннотация

Сегодня искусственный интеллект (ИИ) является одним из наиболее распространенных направлений исследований боевых беспилотных летательных аппаратов (БПЛА). Интеграция ИИ в БПЛА может улучшить его и различные возможности, в частности принятие решений и обнаружение целей. Рассмотрим принцип работы наиболее оптимального алгоритма. В этой статье мы опишем успехи и достижения, которых можно достичь в беспилотных летательных аппаратах (БПЛА) с помощью этого алгоритма.

Ключевые слова: искусственный интеллект (ИИ), беспилотный летательный аппарат (БПЛА), принятие решений, глубокое обучение с подкреплением (DRL), Deep Q-Networks (DQN).

AI can be used to enhance the choices of UAVs. For instance, reinforcement learning algorithms allow UAVs to make operational decisions in real time in a complex environment and can reach for optimal decisions that satisfy the objectives of the mission while operating under the existing hardware limitations [1]. Besides of that applying the fuzzy cognitive decision-making methods is possible to cooperative search for multiple UAVs, consequently increasing the search efficiency [2].

The step-by-step working process of a typical Deep Reinforcement Learning (DRL) algorithm like Deep Q-Networks (DQN):

1. Initialize the Environment: State the operating environment where the agent (UAV for example) will be running. This involves describing the state space, action space, and reward/penalty functions, as well as setting the conditions to terminate the training.

2. Initialize the Deep Q-Network (DQN): Construct a DQN model that takes state as an input and returns the Q-values for each action in the action space as output. The DQN framework is usually backed by a number of layers (e.g., for image input) and activation functions (e.g., ReLU) for learning deeper representations.

3. Initialize Hyperparameters: Fix hyperparameters such there as learning rate, discount factor (gamma), exploration rate (epsilon), batch size, target network update frequency, and replay buffer size. Such hyperparameters are related to the learning process and algorithm convergence procedure.

4. Initialize Replay Buffer: Develop a replay memory also to hold the encountered experience in interactions (state, action, reward, next state) when agent interacts with the environment. By inserting replay buffer, the learner violating the dependency relation between successive experiences is achieved, and stability of learning is also enhanced.

5. Initialize Target Network: Develop a target network, a companion to the DQN that, rather than for training, is used to calculate estimated target V-values. The modification of the target network parameters, for example, after every step N, eliminates overfitting and prevents any bias towards overestimation.

6. Training Loop:

For each episode:

- Reset the environment and be back to the integral.

- While the episode is not terminated:- While the episode is not terminated:

- Choose an action with a epsilon-greedy policy (which is to explore with the probability of epsilon, exploit with probability of 1-epsilon).

- Implement this in an environment where you can see the new state of the environment and its reward.

- Vortpolje – potvrđi danu u replay bufferu (situacija, akcija, nagradu, sljedeća situacija).

- Test a small slice from the replay buffer experience minibatch.

- Calculate the target Q-values using the target network and Bellman equation:

$$Q_{target}(s, a) = r + Y max_{a'} Q_{target}(s',a')$$

- Update the DQN parameters by minimizing the loss between predicted Q-values and target Q-values:

$$Loss = \frac{1}{N}\sum_i (Q(s_i, a_i) - Q_{target}(s_i, a_i))^2$$

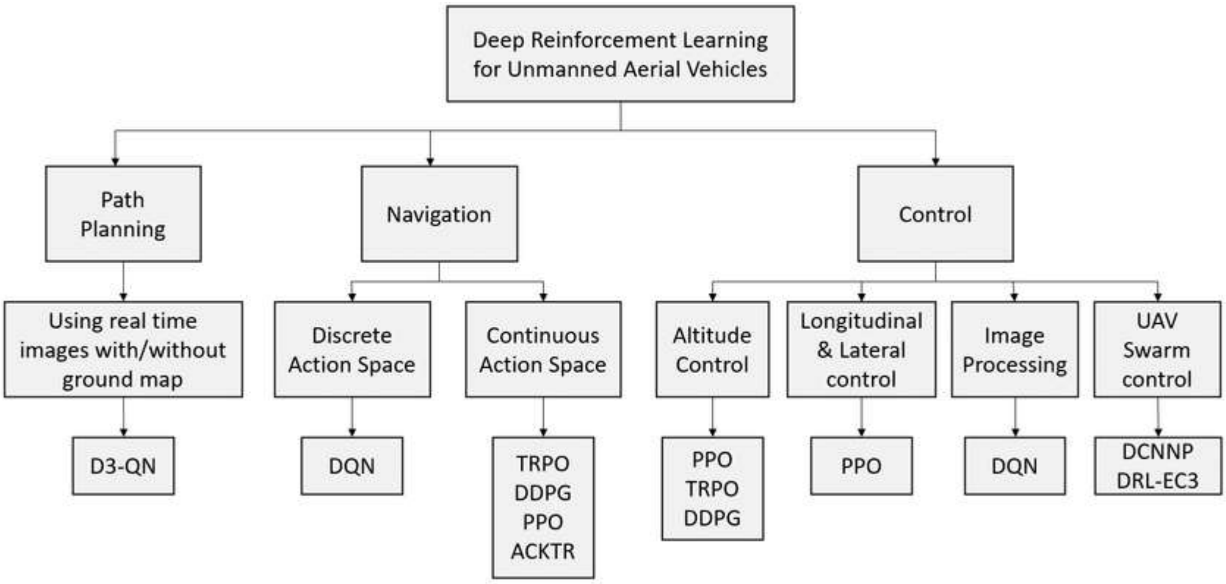
- Apply gradient descent (take for example the Adam optimizer) to make changes in the weights of the DQN.

- If required, go forth and update target network parameters (e.g., after every N cycles).

7. Evaluation: Then, after you have trained the scheme with a fixed amount of episodes or until convergence specifications are met, examine the network performance on test set. Track metrics that will be indicators of how well the agents, such as mean cumulative reward, % of successful runs, or other domain-specific assessments, perform.

8. Deployment: After that, send the trained individual and validated DQN to the real-world environment or simulation for the implementation purposes (e.g., navigation of UAV without human aid in search for a target, execution of surveillance tasks from increased efficiency).

On the other hand, this sequential process presents the common working flow of a DRL algorithm like DQN which trains an agent to find the best policies during reinforcement learn tasks.



•Energy Efficiency: The UAV path planning can be smartly optimized using DRL in conjuncture with the constraints of node energy expenditure and task deadline, leading optimal energy consumption rate for cooperative computing[1].

•Adaptability: DRL algorithms, thanks to their ability to dynamically adjust their routes in the presence of time-varying indeterministic factors, allow the UAV to adapt its flight planning accordingly [3].

•Flexibility: DRL can handle complex surroundings, such as crowded cities, by a mixture of global and local individual map representations of the environment[4].

•Collaboration: DRL thus sets the basis for a coordination between UAVs, separating the task of data collection appropriately, and making sure that the scattered IoT resources are extensively harvested evenly [4].

•Scalability: DRL is capable to modify and tune up the number of UAVs, IoT devices or the maximum flying time without the need in expensive and resource consuming recomputations or relearning control tactics [4].

•Real-time Response: The DRLs can provide real-time responses to whatever changes of environmental status, which helps the UAV stay on an optimal route.

•Improved Performance: DRL has been proven capable of delivering substantial performance improvement to the existing UAV path planning algorithms, manifested in more balanced and energy-efficient use of energy as demonstrated in the simulations [3].

These merits, in turn, make DRL a promising method of UAV route planning, by allowing the creation of the practical, adaptable and scalable solutions for the range of applications, for instance task allocation with collaborative computing and data collecting.

Thus, in a nutshell, AI integration in UAVs for warlike settings is a considerably study field that can be used to improve of capabilities of such UAVs. One of the most significant impacts of AI technologies in the field of UAVs is related to AI technologies particularly decision-making and target identification, can greatly improve the operational efficiency and effectiveness of UAVs.

Reference:

1. https://www.semanticscholar.org/paper/0fd7cdea1f24efb0c0f3051555433417823bb392
2. https://www.semanticscholar.org/paper/029c1d3cb1fe5ff63c9821ed2437c122ba994a52
3. https://www.semanticscholar.org/paper/c6d53346bf56b22cf18a8a4e3f2b9f7febf37022
4. https://arxiv.org/abs/2010.12461