International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 07 | July 2020 www.irjet.net p-ISSN: 2395-0072

Autonomous Drones using Reinforcement Learning

Khush G Chandawat[1*], L Pavan Venkat[2*], Prof Poornima K[3], Prof Priya D[4]

[1,2]*Student, Dept. of Information Science and Engineering, RV College of Engineering, Bengaluru, India*
[3,4] *Assistant Professor, Dept. of Information Science and Engineering, RV College of Engineering, Bengaluru, India*

Abstract - *Unmanned aerial vehicles (UAV) or drones are becoming very useful. They are usually used in operations in unknown environments, there might not be a particular mathematical model for every environment .This paper offers a framework or a model for using reinforcement learning study for navigating of UAV efficiently in such environments. The simulation and actual implementation achieved shows how efficiently the UAV or drone can learn to navigate through unknown territory .Regarding technical factors of using reinforcement learning is getting to know a set of guidelines to a UAV or drone gadget and drone flight management had been also addressed. This will allow continuing studies of the use of a UAV with increasing abilities in more important packages, which consist of wildfire monitoring, or seek and rescue missions.*

Key Words: **UAV, RL, RDS, CNN**

1. INTRODUCTION

Unmanned aerial vehicles (UAV) or drone are used many in undiscovered territories or environments such as surveillance , monitoring of wildfire ,search and rescue mission etc., drone can host wide range of sensors these drone have high flexibility and low operational cost. The main issue faced is that of the accuracy of the model which depends on prior knowledge of the environment .The implementation of these kind of mathematical model are very difficult because the availability of the data and knowledge is very limited or unavailable regarding the environment.

Therefore using a reinforcement learning model is an efficient approach to solve this complex problem because using RL helps the UAV or drone and the UAV team to continuously learn from the changing environment where the model of the environment keeps improving.

The main feature of drone are:-

1. Reduce labor cost
2. Reduce bandwidth requirements
3. Ensure safety
4. Enabling new kind of technology

The main purpose of this paper is to show the implementation of RL algorithm on a drone to navigate to different kind of environment

2. REINFORCEMENT LEARNING AND Q LEARNING AND VARIOUS TOOLS

2.1 What is RL?

RL is very useful when the data and knowledge to train the model is very limited or unavailable for the environment. The training is done in sequence of decision made. Through trial and error, the agent learns to navigate in an uncertain, potentially complex environment to achieve its goal.

The policy is designed for the agents. It is the set of rules to be followed by the agent and the goal of the agent is to maximize the reward by the end.

The assumption is that the ecosystem statifies Markovian property; the reward to the agent is directly dependent on the current state of the agent.

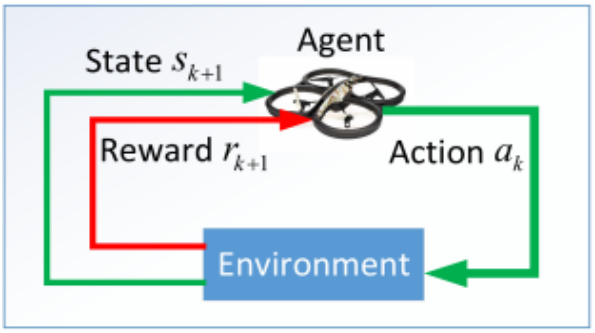

Fig -1: RL algorithm process

2.2 Q-Learning

Q-Learning falls under the paradigm of Reinforcement Learning. The main objective of the agent is to discover and learn a set of actions based on its current states, which is known coverage, to achieve the objective to maximizes the amount of reward it gets over time .In a state - movement fee characteristic Q (sk, ak), It gives how appropriate or good it is to select an movement in a given state or instant, using this the agent decides which movement or step to be take. The agent can iteratively compute the optimal action to be taken under the present state that maximizes the reward. In this paper, we observe a famous Reinforcement Learning algorithm referred to as Q-gaining knowledge of,

wherein Q-deck is used to cache the agent, computes gold trendy fee characteristics and information them right into a Q-deck which is a tabular database. This understanding can be used to recall to determine which motion to take to optimize or maximize its rewards over the next episodes. For each new release version, the estimation of the top-exceptional nation - movement cost function is up to date following the Bellman equation

$$Q_{k+1}(s_k, a_k) \leftarrow (1-\alpha)Q_k(s_k, a_k) + \alpha[r_{k+1} + \gamma \max_{a\prime} Q_k(s_{k+1}, a\prime)],$$

where $0 \leq \alpha \leq 0$ and $0 \leq \gamma \leq 0$ are learning rate and discount factor of the learning algorithm, respectively. To keep balance between exploration and exploitation actions, the paper uses a simple policy called ϵ greedy, with $0 < \epsilon < 1$, as follows:

$$\pi(s) = \begin{cases} \text{a random action } a, & \text{with probability } \epsilon; \\ a \in arg\max_{a\prime} Q_k(s_k, a\prime), & \text{otherwise.} \end{cases}$$

Fig-2: RL algorithm formula

2.3 ROS

ROS stands for robotic running device, is an open source framework that began as part of Stanford University Project STAIR. It affords programmers a clean manner to work with all styles of robotic platforms. It has more than one libraries that permit to code in exceptional languages including C++, Python and Lisp. It's taken into consideration an working system for robots because of the truth it gives similar functionalities to the ones you would count on from a traditional working device, hardware abstraction, and low-level device manipulate. In one-of-a-kind words, the programmer http://stair.Stanford.Edu/ doesn't need to care about enforcing code for a specific hardware due to the fact ROS looks after that. The same way a web developer is not coding the way to set the pixels brighter or decrease inside the display screen, due to the truth that is achieved through the operating system communicating with the screen (hardware).Being open supply facilitates the community to apply libraries that fit with the needs, in this paper we are able to integ brilliant drone robots, with ROS. We can analyze the specific drones in the marketplace and check for projects and libraries that build on pinnacle of ROS to engage and manage the ones drones.

2.4 Open AI Gym

Drones are aerial robots, and superior robots need to be able to paint autonomously the use of AI (artificial intelligence).We realize the way to provide commands to handle verbal exchange among nodes in our drone via ROS, but how can those instructions be automatic? right here is wherein OpenAI is available. OpenAI is a non-season edfit enterprise created with the aid of Elon Musk, that works on enforcing the OpenAI open source framework. To get a bit bit of context, Elon is likewise the founding father of Tesla, a vehicle employer focused on electric powered and self reliant cars. This framework tries to simplify the complexity of AI algorithms via providing a personal first-class framework that permits programmers to train their fashions. OpenAI has a simply beneficial toolkit referred to as OpenAI gymnasium that is focused on a gadget analyzing algorithm this is becoming substantially famous because of the top notch consequences the ones algorithms have become whilst tested in opposition to conventional computer video games. The tool getting to know algorithms are known as reinforcement gaining knowledge of algorithms.

2.5 Gazebo

Gazebo is a 3-d simulator, at the same time ROS serves as an interface with the drone and gazebo. Which combines each results in a powerful robot simulator. With Gazebo you are capable of creating a 3-D scenario on your pc with robots, environmental limitations and plenty of various objects. Physical engines such as illumination, gravity, inertia, etc. are used by gazebo additionally .Where the drone could observe and tested in dangerous or difficult situations in the absence of any damage or harm to the robot. Maximum of the time it's far quicker to run a simulator in preference to beginning the complete state of affairs on your actual robot. In the beginning Gazebo became designed to assess algorithms for robots. For plenty of programs it is crucial to check your robotic utility, like mistakes in managing battery lifestyles, localization, navigation and greed.

3. Implementation Simulation and Results

The reward function: a step returns a reward of positive 100 or a negative 100 which is dependent on the episode whether or not it has been respectively finished or stopped. The middle steps cross the lower back a reward of negative 1 plus Δdg, That is, the distinction from the space-to-the-purpose is recognized in the previous step. This Δdg is used to stimulate movements that are intentional. The simplest difference from the previous work is that violating the geofence can generate terrifying rewards. The drone needs to fly in the vicinity of the delimited area by the geo-fence, this is, a digital orthogonal field. Geo-fence boundaries are specified by the maximum and minimum values of the field's dimensional components.

3.1 State Space

In the first few states, each architecture is provided and tested. A good convolution neural network(CNN) must be implemented for each and every triple state. A convolutional neural network (CNN), the first model of a joint neural community (JNN-2d), and a development model that can manage three vertical actions of a drone JNN-3-D. Each module is designed and developed to handle the 3 different entering states. A 30x100 resolution picture/image is inputted by CNN. The image is an

important part of the depth photo (20 x 100) and is also a vertical 10 x 10 resolution block placed in the desired relative perspective.

The input for the JNN model is the same image, but with more details from the USA system. United States system uses the following scalar values: current position of the drone (px, py coordinates), space and purpose (dx, dy, dt, dt are Euclidean distances calculated from dx and dy), and distance Second geo-fence To (dxmin, dxmax, dymin, dymax).

Finally, the images of the JNN network have been simplified to hold only depth photos (20 x 100 pixels) and the drone role has also been removed from the state, avoiding overfitting. Instead, scalar frames are transformed using the geofence's third dimension. Where P is the role of the drone, G is the distance traveled, and GF is the distance to the geofence. Location and distance are second for CNN and JNN 2d and 3D for JNN 3D structure.

The JNN network integrates the CNN layer with dense layers of various state scalars. The CNN layer produces the image of the Kingdom and then becomes part of the scalars, hence the name JNN. Then the two community flows are aggregated and passed through the final dense layer layer to the output. The JNN-2 has a weight of approximately 887,203 and the JNN-3-D has a weight of 642,982. Despite the fact that JNN-3-D has a 3D input kingdom, it shortens the length to photo simplification by 27%.

3.2 Action space

The different turning angles allow lots of viable instructions with the aid of successive turns. However, the horizontal turn has the ability to stop the drone before any turning functionally. It gives a swinging nature and should not be tried.

Vertical movement is inspired by previous studies. The motion of a 6-dimensional drone was recorded to handle the touchdown technique. There are four horizontal layers that move along the x-y axis and up and down movement which help to the fore stall and to descend. Each movement of the command was converted into a two-dimensional movement, which is a shift of about 1 meter. The author states a significant vibration. This leads to faster movement and leads to the introduction of stop motion, so four different images are taken on the spot along the route of the table. These four different snapshots were used to select subsequent step agent moves. Keep in mind that these are a set of motions which no longer consist of any ascend.

The final set of different vertical functions in 3D space has 6 practical choices for replacing individual 3-axis drone velocities. There is a change of ± 0.5 metre per second increment due to drone tempo.

3.3 Improving Training Efficiency

The training and the stimulating environments run on a system powered by Intel i7 processor, 16 Gigabytes of RAM and a NVIDIA GeForce GTX TITAN Black with 6 GB of VRAM. The simulator takes 40 hours to complete each cycle because it relies on the rendering of the surrounding environment to embed the agent.

In general, stopping training usually gives you a comfortable version where the rest of the model and weight are saved. However, we found that a small percentage of random tuning of the training parameter duration leads to improvement. It is very expensive to repeat or expand training. There is no guarantee that the rewards will be high. Therefore, developing inside the training has been carried out in which the model weights are stored every time a positive reward is obtained. These rewards episodes are saved as checkpoints.

This will create a model each cycle. At the end of the training cycle, the rest of the models with nn weights and all the other models taken at the final checkpoint named "check pt 457845.m5" as it back propagates the model weights for the cycle with the satisfactory result of the training outputs are then completed by closing the model and the good model and consequences are explored to expose the different checkpoints.

3.4 Experimental Results

Environment and the parrot uav is simulated in Gazebo. Gazebo is a simulator, with powerful image support and developed for such situations. Gazebo released a test of the MAV model and found that the simulated flight environment was close to the real MAV behavior.

In the chosen scenario, the purpose of the model and drone is to find different ways to get from point A to point B. Usually, different cycles start from the point [0,0,0] axis. The rectangles suggest that obstacles between flight cycles will either stop at the same location or arrive while hitting a specific obstacle in the geo-fence where the MAV is not visible. However, the cycle will stop as soon as the maximum training step is reached.

4. CONCLUSION

The increasing usage of drones has created a need to improve the performance of drones to work in different types of environments .In this paper shows to technique to use Q-learning algorithm to train a UAV to navigate through unknown environment .The simulation and implementation shows similar results where the drone can navigate through unknown environment .This paper shows a method through which a RL drone can navigate through unknown environment without a model available. In future

,we will continue to work on using UAV on real world application.

5. FUTURE WORK

Our future goal is to work on using UAVs where these learning capabilities help the UAV to have better coordination and effectiveness on solving real world problems.

6. REFERENCES

[1] Nikolai Smolyanskiy ,Alexey Kamenev, Jeffrey Smith, Stan Birchfield ,Toward Low-Flying Autonomous MAV Trail Navigation using Deep Neural Networks for Environmental Awareness,IEEE /RSJ conference(2017).

[2] Alessandro Giusti , Jérôme Guzzi1 , Dan C. Cire¸san , Fang-Lin He , Juan P. Rodríguez,A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots, IEEE ROBOTICS AND AUTOMATION(2015)

[3] H. Alvarez, L. Paz, J. Sturm, and D. Cremers,Collision avoidance for quadrotors with a monocular camera,International Symposium on Experimental Robotics(2016)

[4] F. Liu, C. Shen, G. Lin, and I. Reid,Learning depth from single monocular images using deep convolutional neural fields,IEEE Transactions on Pattern Analysis and Machine Intelligence(2015).

[5]F. Munoz, E. Quesada, E. Steed, H. M. La, S. Salazar, S. Commuri, and L. R. Garcia Carrillo, Adaptive consensus algorithms for real-time operation of multi-agent systems affected by switching network events, International Journal of Robust and Nonlinear Control, (2017).