Automated Drones for Radiation Source Searching with Reinforcement Learning

2018 Symposium on Radiation Measurements and Applications (SORMA XVII)

Zheng Liu[a], Gregory R. Romanchek[a], Shiva Abbaszadeh[a]

[a]Dept. of Nuclear, Plasma, and Radiological Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 61801

Introduction

In the search of anomalous radiation sources, different survey approaches have been studied such as manually scanning of the area by human operated detectors, as well as automatically scanning the area with robots under the navigation of pre-defined survey paths. However, neither of the manual scanning method and the survey path method can achieve flexibility and efficiency at the same time. Recent developments of reinforcement learning and drone technology provide an alternative data-driven solution to conduct radiation detection automatically. In this paper, we integrates a drone, a radiation detector, and reinforcement learning into an automated radiation source detection platform such that it automatically searches for radiation sources efficiently without any human intervention. The hardware components of the automated radiation source detection platform are introduced, and simulation results of applying reinforcement learning for automated radiation source detection are presented.

Methods

Drones for radiation detection

As shown in Fig. 1, a mobile radiation sensor (D3s gamma/neutron detector and a phone) is attached to a drone (DJI Inspire 2) to construct an automated radiation detection platform. The phone can run a trained reinforcement learning algorithm to analyze radiation measurements from the detector and navigate the drone to search for anomalous radiation sources.

Fig. 1 Drone with mobile radiation sensor attached

Methods (cont'd)

Reinforcement learning for source detection

A reinforcement learning (RL) algorithm (Q-learning [1] + recurrent neural network (RNN) [2]) was developed to navigate the radiation detection platform in source detection tasks. Fig.2-left illustrates a general problem setup in reinforcement learning: in an environment, an agent obtains observations (S), makes actions (A), and receives rewards (R) step by step. The Q-learning (a family of algorithms in reinforcement learning) aims to find a function $Q(S, A)$ that guides the agent to take the optimal action such that the cumulative reward is maximized. The specific simulation problem setup is as follows (Fig. 2-right):

- Environment (50x50 grid): background radiation + randomly placed single anomalous source
- Agent action: move in one of four directions
- State: (agent action, radiation measurement)
- Reward: 0.5 if closer to source, -1.5 otherwise.

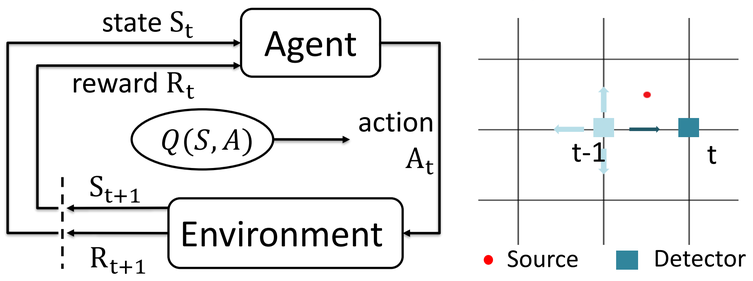

Fig. 2 Left: General problem setup for reinforcement learning. Right: Specific simulation environment for the radiation source detection task.

We implemented RNN (Fig. 3) to represent $Q(S, A)$ such that current action was chosen based on previous several steps' actions/measurements.

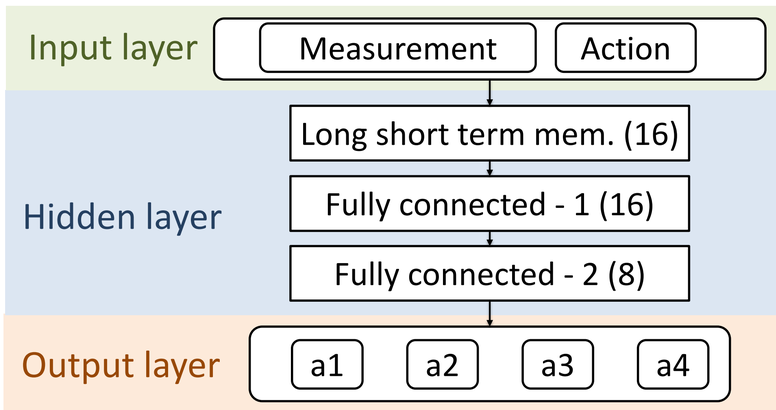

Fig. 3 RNN that approximates the $Q(S, A)$.

To stabilize the training, we further applied double Q-learning [3] and experience replay [1]. Radiation measurements were deterministic in training and Poisson-random in testing. Here is the algorithm:

Algorithm Recurrent neural network Q-learning

Initialize reward function Q with weights θ
Initialize target reward function $\hat{Q}$ with weights $\hat{\theta}$
Initialize replay memory D
for episode = 1,M **do**
  Initialize sequence of states $\phi_1 = (s_{-k+2}, s_{-k+3}, \ldots, s_1)$
  for t=1,T **do**
    Select action a_t using ϵ-greedy policy based on Q
    Execute action a_t, obtain reward r_t and state s_{t+1}
    Update sequence of states $\phi_{t+1} = (s_{t-k+2}, \ldots, s_{t+1})$
    Store transition $\{\phi_t, a_t, r_t, \phi_{t+1}\}$ in D
    Sample random minibatch of transitions $\{\phi_j, a_j, r_j, \phi_{j+1}\}$ from D
    Set

$$y_j = \begin{cases} r_j \text{ if episode terminates at step } j+1 \\ r_j + \gamma \hat{Q}(\phi_{j+1}, \underset{a}{argmax} Q(\phi_{j+1}, a)) \text{ otherwise} \end{cases}$$

    Perform a gradient descent step on $\text{HuberLoss}(y_j, Q(\phi_j, a_j; \theta))$ for θ
    Every C steps reset $\hat{Q} = Q$
    Terminate the episode if source is found
  end for
end for

Results

Training results for source detection

Fig. 4 shows the RNN-Q-learning algorithm's training results in the metric of 'epoch length' and 'cumulative rewards'. Each training epoch would be terminated if the agent found the source or moved more than 100 steps. The algorithm converged after 40000 epochs' training. After convergence, the agent toke in average 40 steps to find the source and achieved a mean cumulative reward of 14.

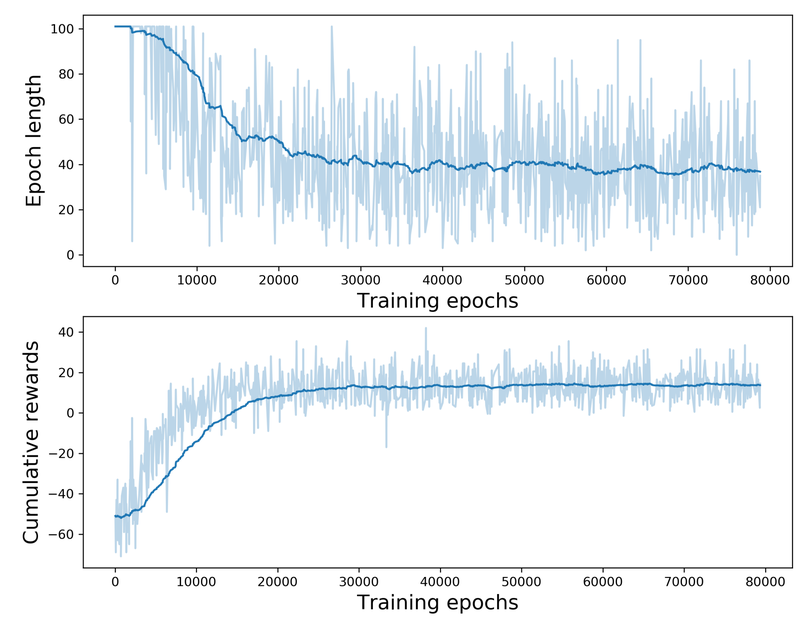

Fig. 4 Training results of the RNN-Q-learning algorithm.

Fig. 5 shows 9 testing examples after the algorithm was trained for 70000 epochs. The source strength was 1600 times higher than the background radiation, and measurements were Poisson variables. The RNN-Q-learning algorithm was able to find the source efficiently in all 9 testing cases.

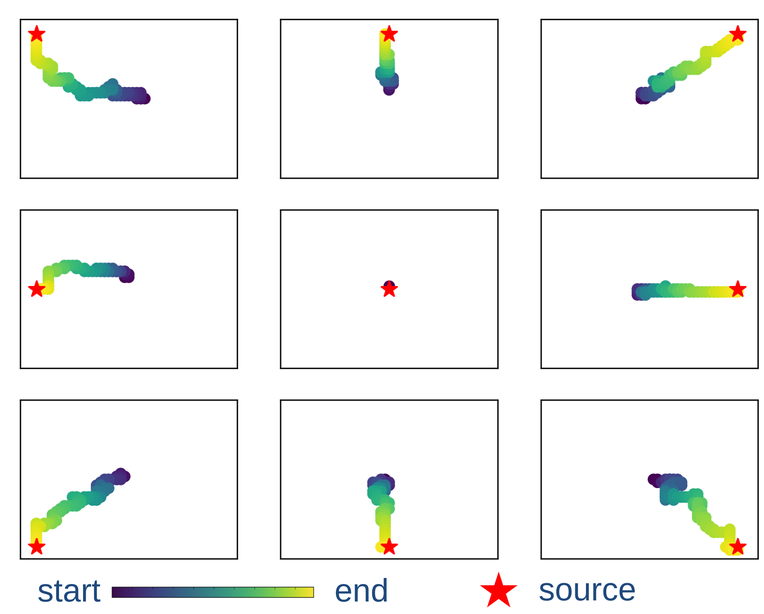

Fig. 5 Testing results of the RNN-Q-learning algorithm.

Conclusions:

- A drone-based mobile radiation senor system was developed.
- A reinforcement learning (RL) algorithm was developed to navigate a mobile sensor searching for radiation sources.

Acknowledgements:

Whom should we acknowledge in this poster?

References

[1] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529.

[2] Graves, Alex. "Generating sequences with recurrent neural networks." arXiv preprint arXiv:1308.0850 (2013).

[3] Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep Reinforcement Learning with Double Q-Learning." AAAI. Vol. 16. 2016.