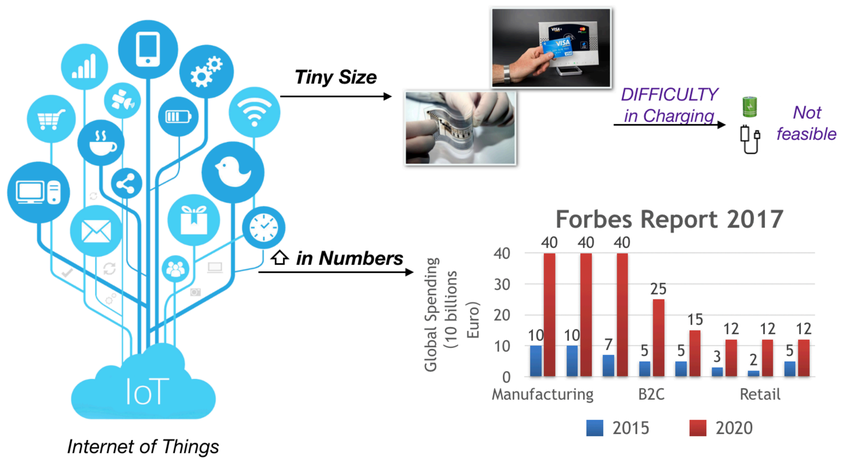
ROBOT PATH PLANNING IN WIRELESS COMMUNICATION
Using Reinforcement Learning

Anushka Vashishtha [FYP18049]
Supervisor: Dr. Chuan Wu

Motivation: Charging tiny IoT devices via UGV



Conclusion

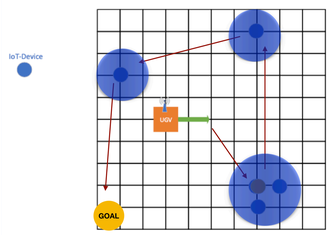
- ❖ Q-learning should be used when operation area is small as it finds the optimal path faster in that case. Dyna-Q should be incorporated with it in scenarios where no information about the model is present.
- ❖ In all the other cases, deep Q-learning should be used as it supports larger grid size and is more efficient in achieving an optimal path with low energy consumption.

Key Steps

Creating charging region model and deploying UGV for charging → Applying Multi Integer Non-Linear Programming, Q-learning and Deep Q-learning for effective robot path planning → Choosing one with the best performance

Charging Region Model

Presents operation area containing robot, goal & charging regions of IoT devices. Robot has to charge the devices by entering their charging regions & ultimately reach the goal for maintenance. Moving energy of robot is reduced through path planning.



Multi Integer Non-Linear Programming

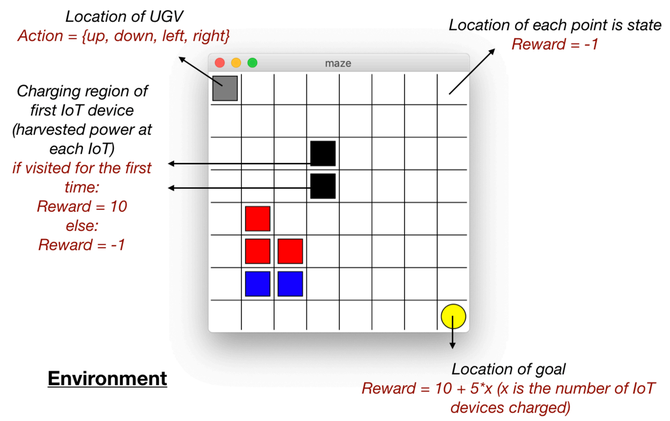
$$E_M = \min_{\mathbf{v},\mathbf{X},\{\lambda_m\}} \left(\frac{\alpha_1}{a} + \alpha_2\right) \mathrm{Tr}(\mathbf{D}^T\mathbf{W})$$

s.t.
$v_1 = v_M = 1,\ (select\ starting\ and\ end\ points)$
$v_m \in \{0,1\},\ \forall 2 \le m \le M-1,\ (selection\ is\ binary)$
$\sum_{m\in\mathcal{C}_i} v_m \ge 1,\ \forall i = 1,\cdots,K,\ (charge\ all\ IoT\ users)$
$W_{m,j} \in \{0,1\},\ \forall m,j,\ W_{m,m} = 0,\ \forall m,\ (flow\ selection\ is\ binary)$
$\sum_{j=1}^{M} W_{1,j} = 1,\ \sum_{j=1}^{M} W_{j,1} = 0,\ (flow\ from\ starting\ point)$
$\sum_{j=1}^{M} W_{M,j} = 0,\ \sum_{j=1}^{M} W_{j,M} = 1,\ (flow\ to\ end\ point)$
Presents lower bound
$\sum_{j=1}^{M} W_{m,j} = v_m,\ \sum_{j=1}^{M} W_{j,m} = v_m,\ \forall m = 2,\cdots,M,$
$(flow\ passing\ selected\ points; no\ flow\ passing\ abandoned\ points)$
$\lambda_m - \lambda_j + \left(\sum_{l=1}^{M-1} v_l - 1\right) W_{m,j} + \left(\sum_{l=1}^{M-1} v_l - 3\right) W_{j,m}$
$\le \sum_{l=1}^{M-1} v_l - 2 + J\left(2 - v_m - v_j\right),\ \forall 2 \le m,j \le M-1,\ m \ne j,$
$v_m \le \lambda_m \le \left(\sum_{l=1}^{M-1} v_l - 1\right) v_m,\ \forall m \ge 2.$
$(guarantee\ flow\ connected; Sub-Tour\ elimination)$

Parameters

$\mathbf{v}$	Visit a point in the grid or not (Boolean variable)
$\mathbf{X}$	Link between two points in the grid or not (Boolean matrix)
α_1, α_2	Toning parameters (0.29, 0.41): pioneer's 3DX robot experiment result at MIT(constant)
a	Velocity of UGV (constant)
$\mathbf{D}$	Distance exists between two points in the grid or not (Boolean matrix)
$\mathbf{W}$	Summation of X values (matrix)
M	length & width of the grid (variable)
K	Total number of IoT devices (variable)

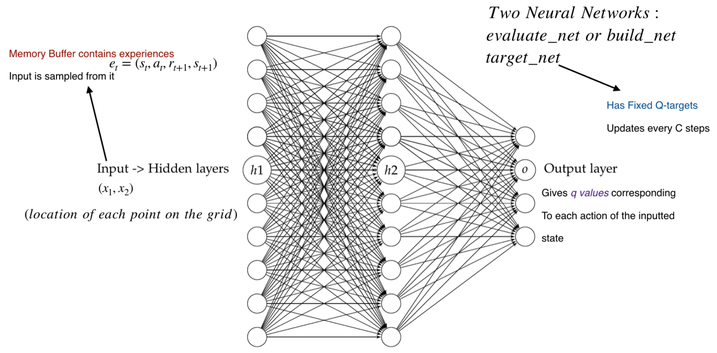
Q-learning & Deep Q-learning

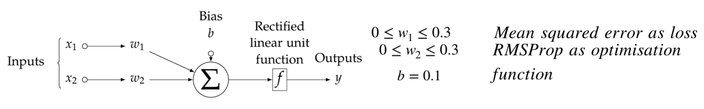


Incorporate Dyna-Q when not enough information about the model is present as it uses simulated environment for model learning before Q-learning application.

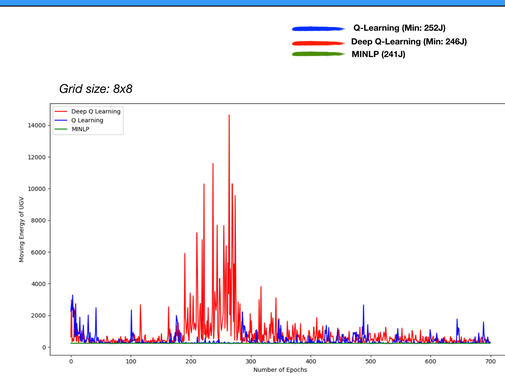
Deep Q-Network

Neural Network Architecture



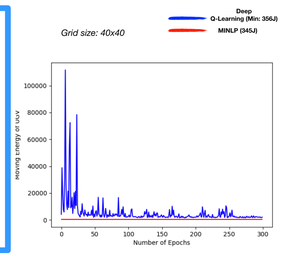


Results



- ❖ Q-learning trains faster
- ❖ Moving energy for deep Q-learning is lower for larger epochs as it approximates to a stable optimal Q-function due to experienced replay which eliminates correlation between present and next state

Larger grid: Deep Q-learning finds a path faster for operation area with 40x40 grid size. Whereas, Q-learning is not efficient due to slow Q-table updates via value iteration.



Future Work

- ❖ Variable power given by robot to an IoT device according to how far away it is from an IoT device
- ❖ Limited energy present in UGV
- ❖ Federated Deep Reinforcement Learning using multiple agents
- ❖ Continuous charging model