Cooperative Transportation using Multi-agent Reinforcement Learning

Aditya Barve, Komal Kangutkar, Sayali Salunke

Guide: Prof. S. P. Gosavi

“To train a team of software agents to carry out transportation tasks in a randomly generated 2D grid world using reinforcement learning.”

System Requirements

Small agents

Cooperative

Decentralised

Local communication

Limited field of view

Anonymity

Literature Survey

A significant part of the research on multi-agent learning concerns reinforcement learning techniques. Benefits and challenges of MARL were learnt from the research papers listed below:

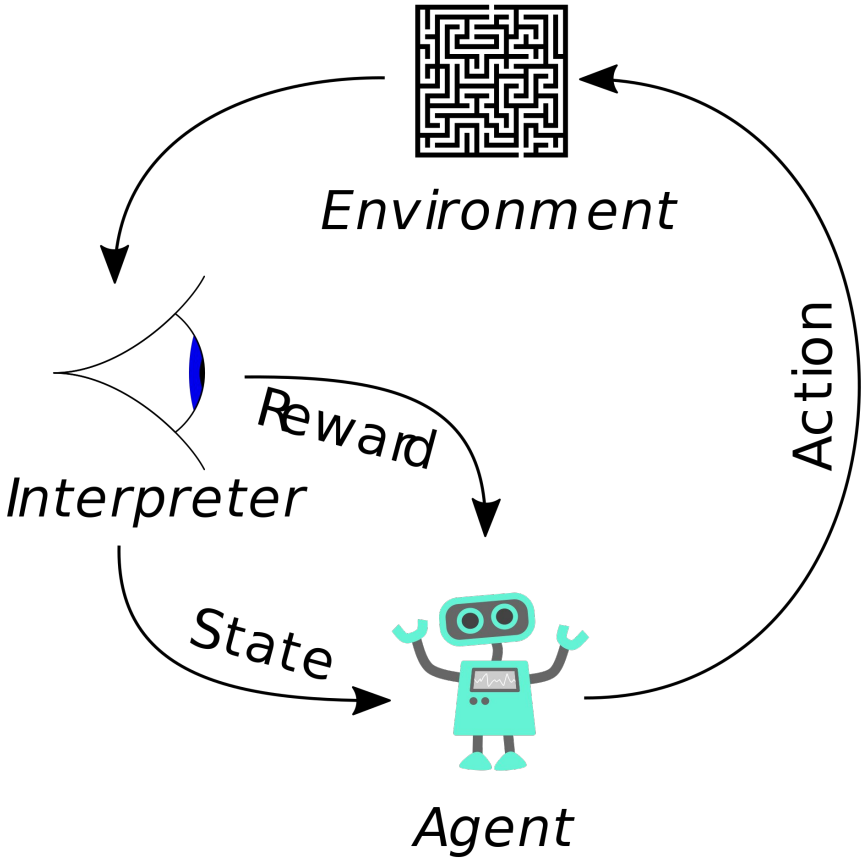
1. A Comprehensive Survey of Multi-Agent Reinforcement Learning (Buşoniu et al)
2. Coordinating Multi-Agent Reinforcement Learning with Limited Communication (Zhang et al)
3. A Survey of Multi-Agent Pathfinding Problem (Zhou et al)

Reinforcement Learning

Learn from trial and error

Decisions affect future inputs

Learn human-like skills



Learning Algorithm

$N \leftarrow Network.initialize(\text{random weights})$

for $i = 1$ to n **do**

$S_i \leftarrow Simulation.initialize()$

//Run simulation to generate Agent experiences

$S_i.run()$

//Instantaneous Learning

for all $A \in$ Agents **do**

$N.train(A.experiences)$

$S_i.experiences \mathrel{+}= A.experiences$

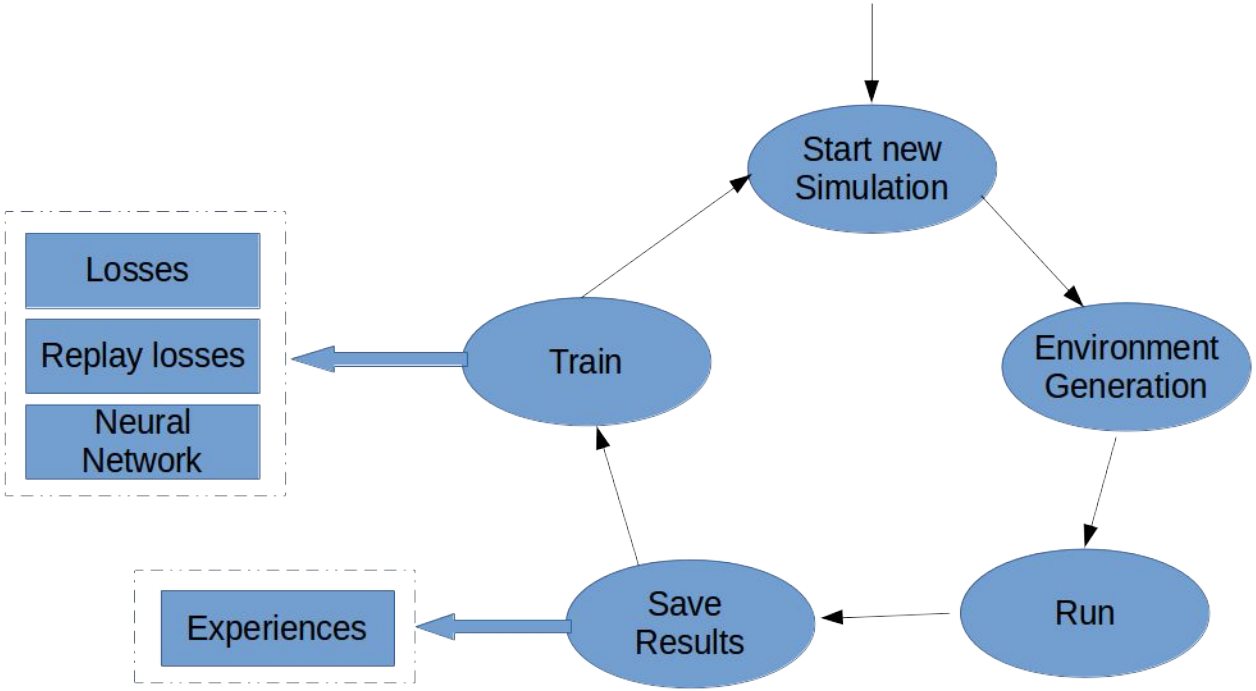
//Experience Replay

Simulation $S' \leftarrow$ choose random from $S_{1..i}$

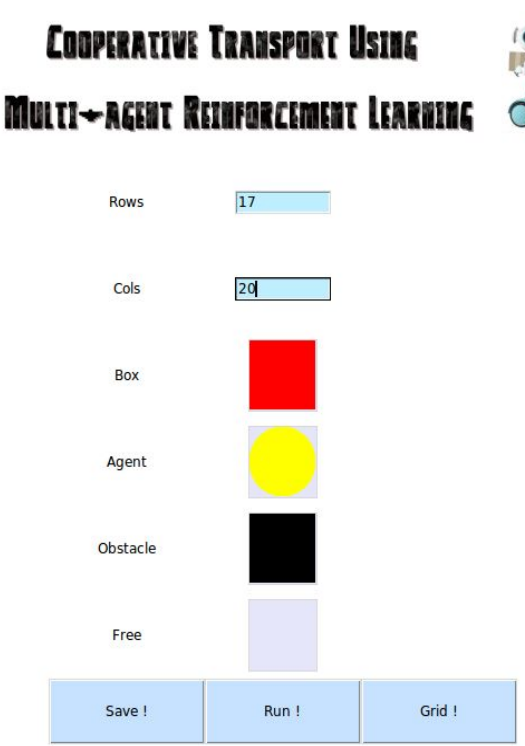
Experience $E' \leftarrow$ choose random subset from $shuffled(S'.experiences)$

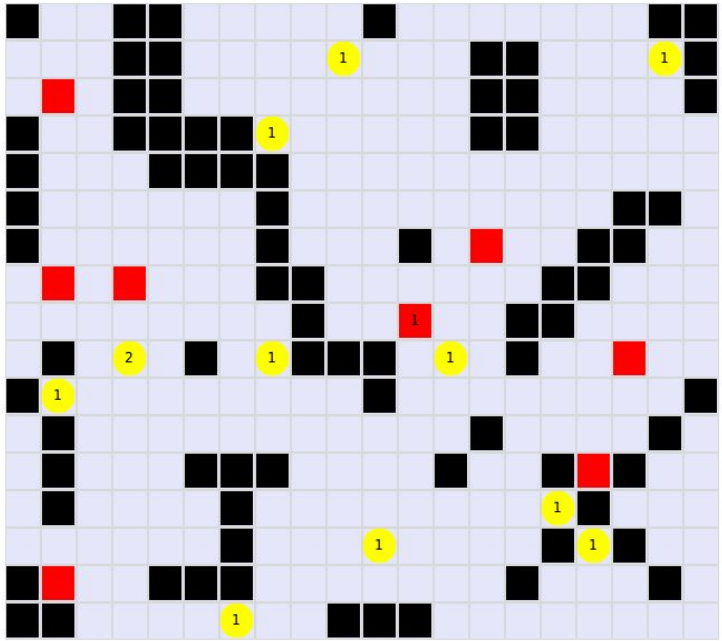
$N.train(E')$

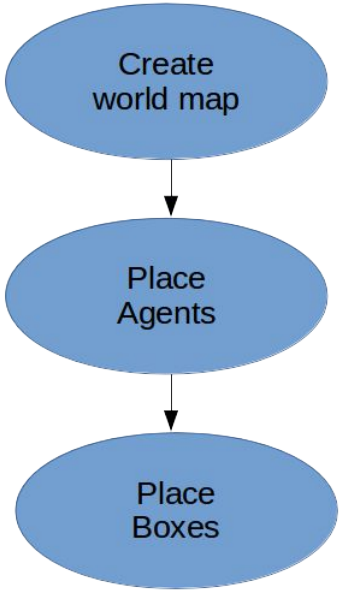
System Overview



Environment







Agent

Actions

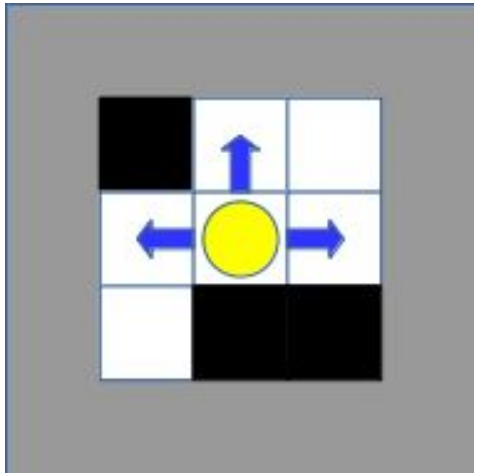
1. Move actions
2. Box actions

Roles

1. Explore environment
2. Deliver boxes

Attributes

1. Lifting capacity
2. Internal map
3. Map Gaussian
4. Location Cache
5. Network
6. Percept



Percept

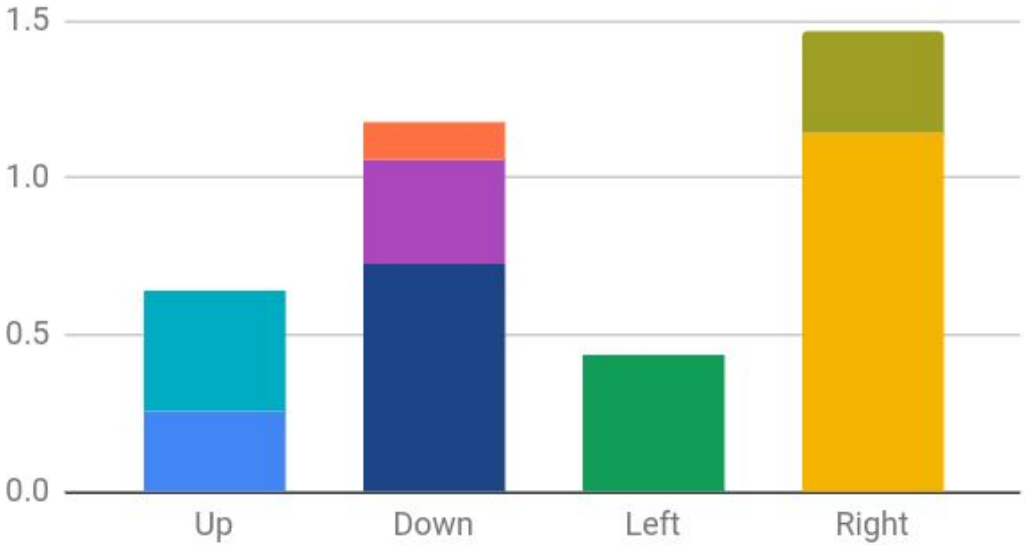
Box

Attributes

1. Weight
2. Destination

Alternating steps

1. Voting step
2. Group movement step

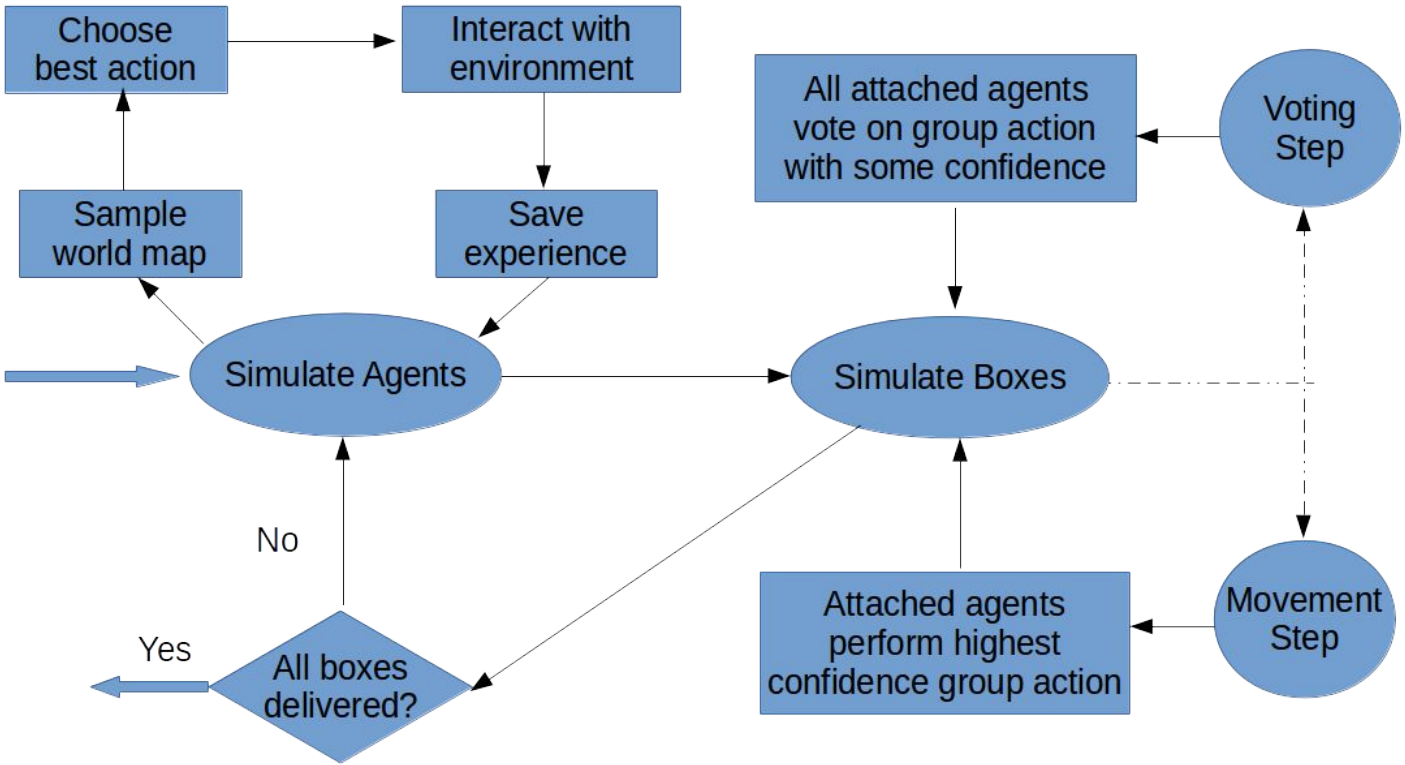


Voting consensus

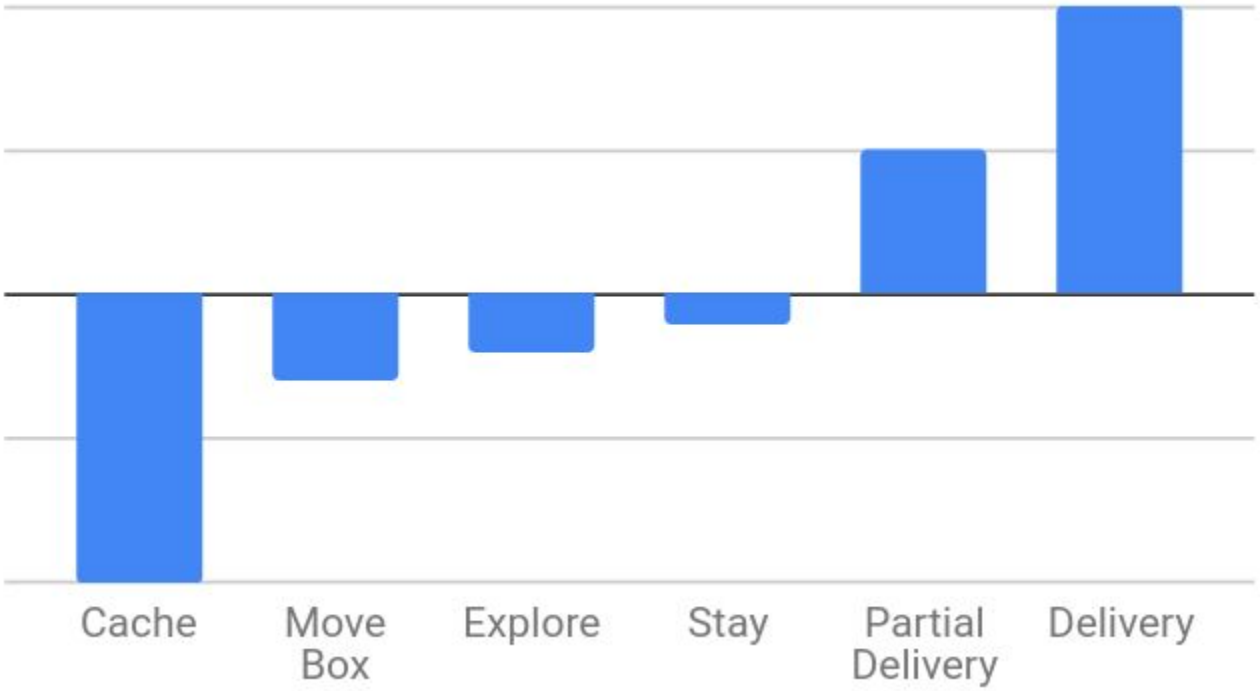
Demo

Embed video

Simulation

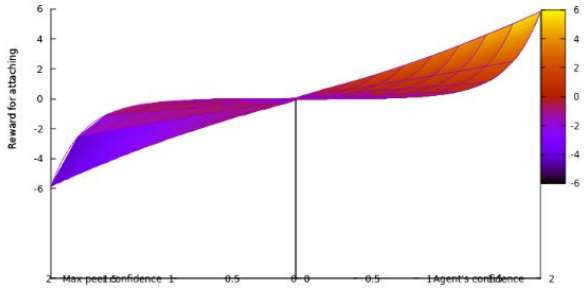

Choose best action
Interact with environment
All attached agents vote on group action with some confidence
Voting Step
Sample world map
Save experience
Simulate Agents
Simulate Boxes
No
Yes
All boxes delivered?
Attached agents perform highest confidence group action
Movement Step

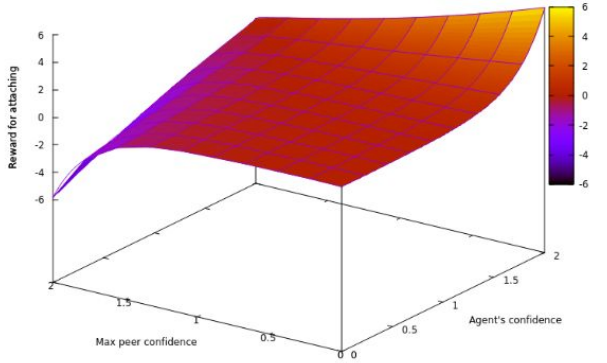
Reward System



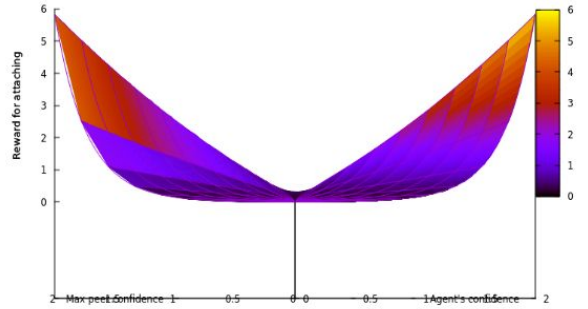
Attachment Reward

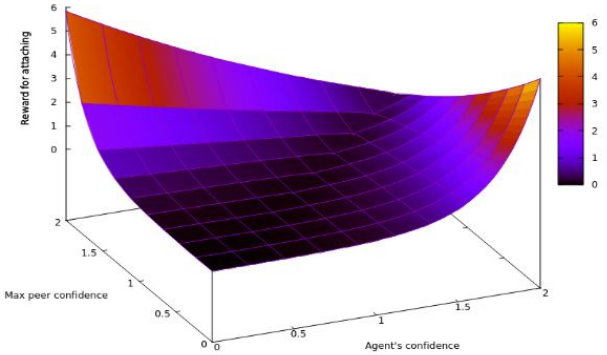
No remaining box weight to be lifted



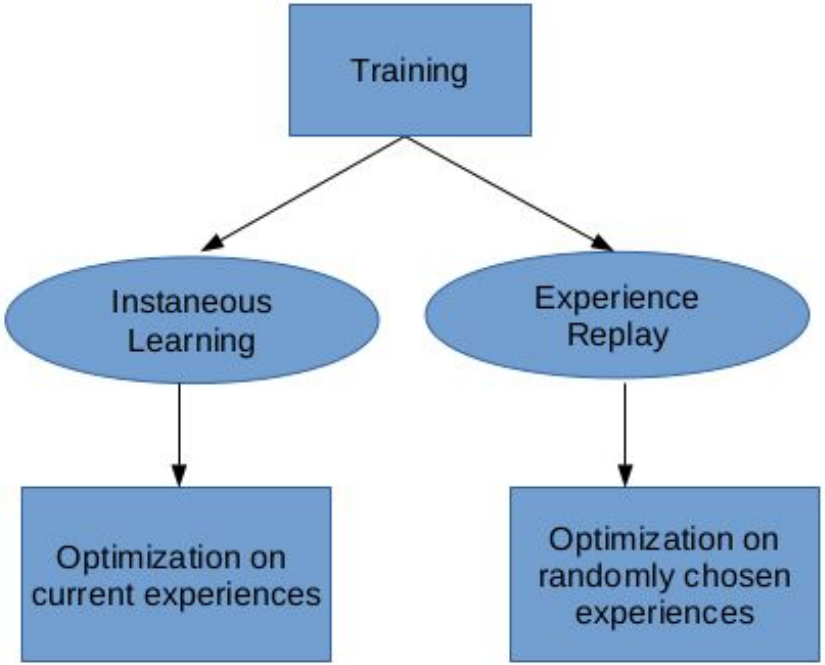


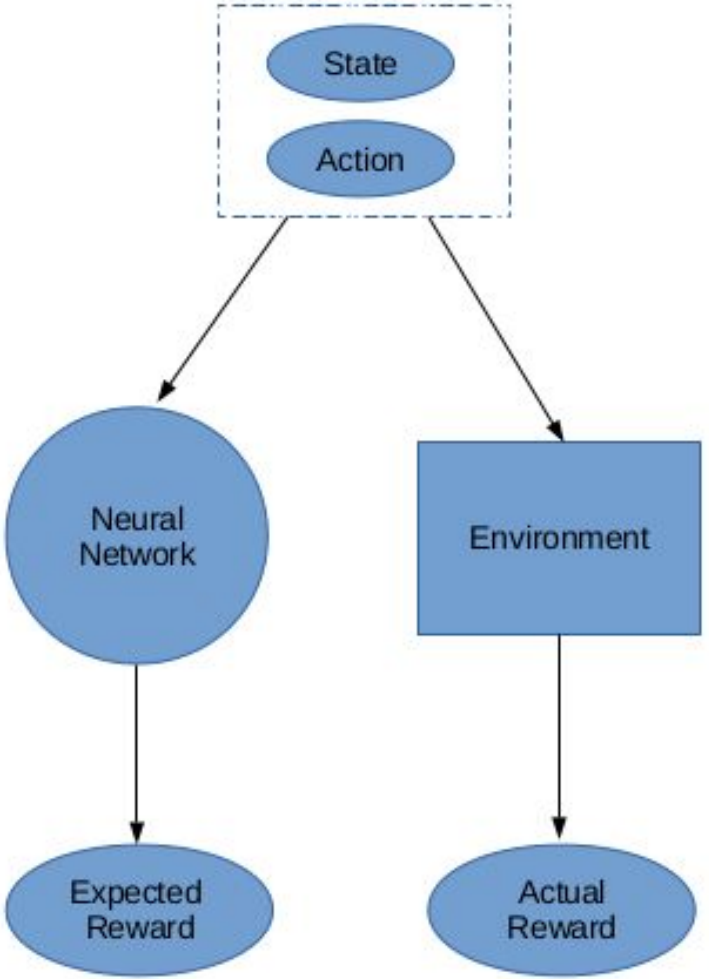
Some remaining box weight to be lifted





Q-Network


Training
Instaneous Learning
Experience Replay
Optimization on current experiences
Optimization on randomly chosen experiences


State
Action
Neural Network
Environment
Expected Reward
Actual Reward

Network parameters

Agent's personal information

1. Position
2. Percept
3. Lifting capacity
4. Map Gaussian
5. Internal map size

Agent's action information

1. Movement and box actions
2. New position

Information read from box

1. Goal
2. Number of lifters
3. Remaining weight
4. Peer confidence

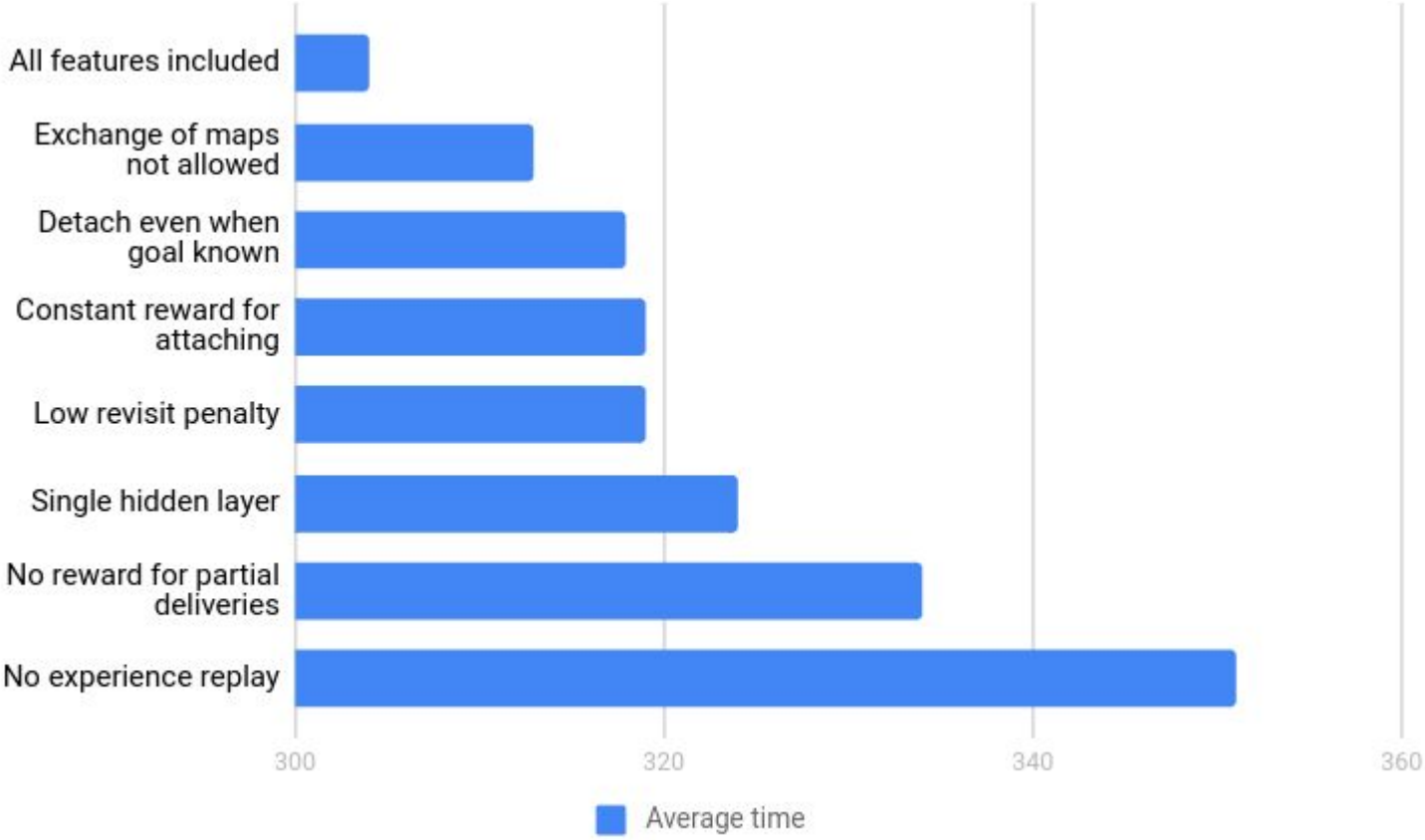
Network specific information

1. Explore network – cached locations
2. Box network – time attached to box

Features which improve system performance

- Agents **exchange maps** of the environment when they come in direct contact
- Agents **stay attached** to the box if they know the goal
- **Attachment reward is proportional** to the confidence of attaching agent and its attached peers
- Agents receive a **large penalty for revisiting** recently visited locations
- Neural network has **two hidden layers**
- Agents are **rewarded for partial deliveries**
- Neural network is trained using **experience replay**

Analysis and Results



Conclusion

Our system **reverses** the traditional transportation model. Teams of agents are trained using **MARL** to deliver boxes in unknown environments. We show how various features affect the **efficiency** of the system.

Future Work

Exchange of box information

Beacon system

Specialized agents

Long range communication

References

1. "Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G. Barto
2. "Deep Learning" by Ian Goodfellow Yoshua Bengio, Aaron Courville
3. "A Comprehensive Survey of Multi-Agent Reinforcement Learning" Lucian Buşoniu, Robert Babuška, Bart De Schutter
4. "Cooperative Manipulation and Transportation with Aerial Robots" Nathan Michael, Jonathan Fink, and Vijay Kumar
5. "Coordinating Multi-Agent Reinforcement Learning with Limited Communication" Chongjie Zhang, and Victor Lesser
6. "A Survey of Multi-Agent Path-finding Problem" Xin Zhou
7. "Playing Atari with Deep Reinforcement Learning'' Volodymyr Mnih, Koray Kavukcuoglu, David Silver

Thank you
Any questions?