Continual Reinforcement Learning for Robotic Tasks

Initial Presentaiton

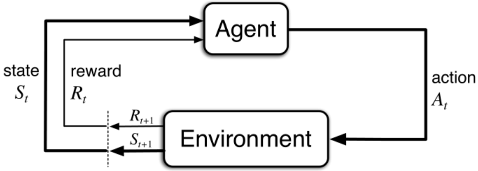
Philemon Schöpf

Supervisors: Antonio Rodriguez Sanchez, Sayantan Auddy

2022-03-15

Reinforcement Learning

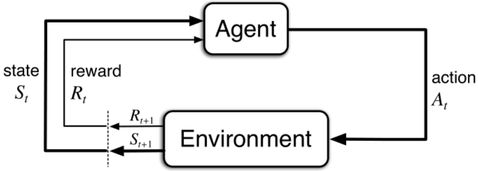
- Learn by interacting with an environment[1]
- Find a policy that takes the optimal action for each state (input)



RL loop[1, p.48]

Reinforcement Learning

- Learn by interacting with an environment[1]
- Find a policy that takes the optimal action for each state (input)
- Ideally suitable to robotics
  - No (labeled) training data required
  - Physical simulation of environment easier



RL loop[1, p.48]

Continual Learning

- Learning of multiple tasks in succession
- Adapt existing model with new experiences
  - Model has to retain old information
  - Old training data is **not accessible** while training new tasks
- Critically important for human intelligence
- Still a major issue in machine learning[2]

Catastrophic Forgetting

- Old skills are forgotten as network is trained on new data
  - No incentive to keep old skills
- Traditional approach: *replay* of old training examples to incentivize preserevation
  - Keep old training data: storage inefficient, privacy issues
  - Not scalable to "lifelong learning"

Previous Work

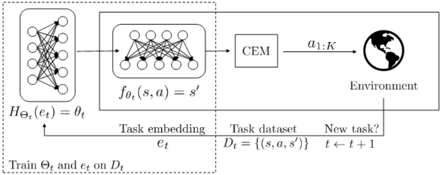
- Regularization: penalize changes in prediction[2]

$$\theta_s^*, \theta_o^*, \theta_n^* = \text{argmin}_{\hat{\theta}_s, \hat{\theta}_o, \hat{\theta}_n}(\lambda \boldsymbol{\mathcal{L}_{old}(Y_o, \hat{Y}_o)} + \mathcal{L}_{new}(Y_n, \hat{Y}_n) + \mathcal{R})$$

- Dynamic architectures: change/expand network while training[2]
- Hypernetworks: NN that output another NN[3]

Hypernetworks

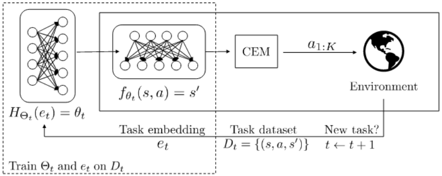
- Hypernetwork generates weights
- Main network generates state-action paris



Hypernetwork Architecture for RL[3,4]

Hypernetworks

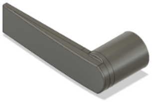
- Hypernetwork generates weights
- Main network generates state-action paris
- Task embeddings and hypernetwork learn via backpropagation
- Separate task identification and solving
  - "If–then-else" as a NN

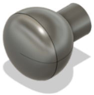


Hypernetwork Architecture for RL[3,4]

DoorGym

- Simulated environment for door opening
- Different types of door handles
  - Round turn knob
  - Lever knob
  - Pull knob
- Train agent able to open all 3 kinds of doors
  - One type of door learned at a time



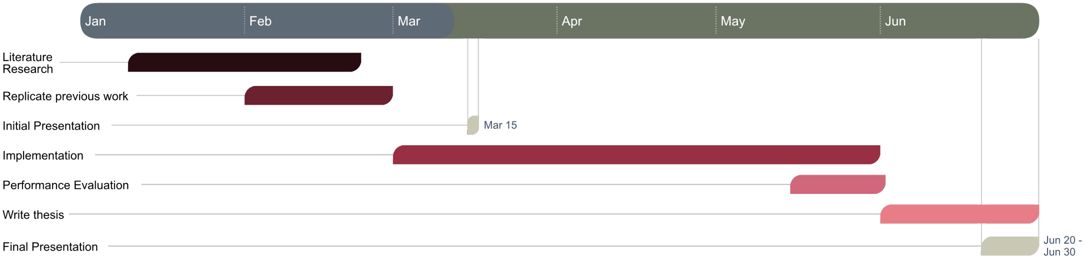


DoorGym demo

Goals

- Use the hypernetwork approach for reinforcement learning
- First: task-incremental CL[5]
    - Type of door is known at inference time
- Expanding: domain- and class-incremental CL
    - Domain-incremental CL: type is unknown
    - Class-incremental CL: type is unknown *and* we want to infer type
- Inference e.g. via object recognition from renders

Planned Timeline



References

R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, 2nd Edition, The MIT Press, 2018.
URL http://incompleteideas.net/book/the-book-2nd.html

G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, S. Wermter, Continual lifelong learning with neural networks: A review, CoRR abs/1802.07569 (2018).
URL http://arxiv.org/abs/1802.07569

Y. Huang, K. Xie, H. Bharadhwaj, F. Shkurti, Continual model-based reinforcement learning with hypernetworks, CoRR abs/2009.11997 (2020).
URL https://arxiv.org/abs/2009.11997

J. von Oswald, C. Henning, J. Sacramento, B. F. Grewe, Continual learning with hypernetworks, CoRR abs/1906.00695 (2019).
URL http://arxiv.org/abs/1906.00695

G. M. van de Ven, A. S. Tolias, Three scenarios for continual learning, CoRR abs/1904.07734 (2019).
URL http://arxiv.org/abs/1904.07734