A Foundational Basis of Meta-Reinforced Learning

Georgia Tech College of Engineering
H. Milton Stewart School of Industrial and Systems Engineering

ISYE Summer Scholars Program 2024 | Dr. Xiaoming Huo and Anna Zhao

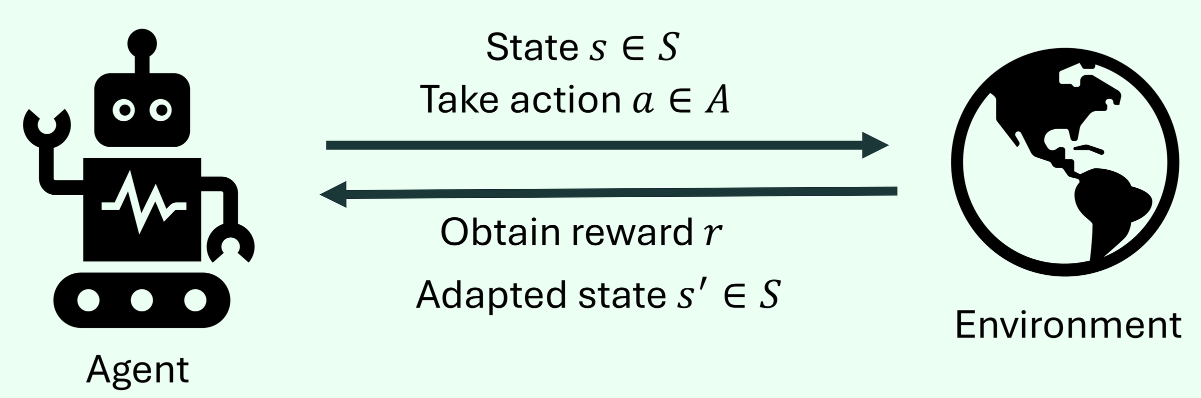
Introduction

Why Meta Reinforcement Learning?

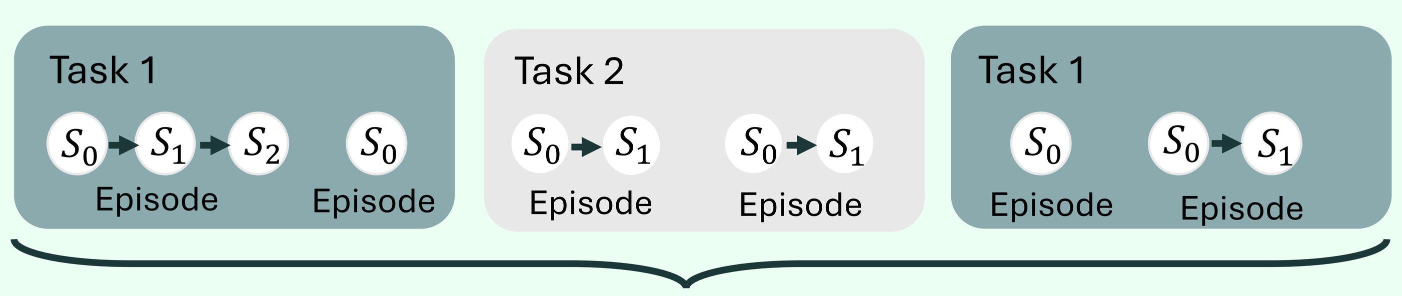
To answer this question we first need to address reinforcement learning itself- machine learning (ML) has three different methodologies:

Supervised Learning	Unsupervised Learning	Reinforcement Learning
❖ Data provided is labelled input and output ❖ Classification	❖ Data provided is unlabeled and is sorted through pattern recognition ❖ Clustering	❖ An agent interacting with its environment ❖ Learns through trial and error (reward or failure)

Reinforcement Learning Task Overview



Meta Learning



Meta training allows for a model to be exposed to multiple tasks with separate datasets to optimize performance for adaption.

Motivation and Definition

Traditional reinforced learning (RL) runs into issues with

- ❖ Needing a large amount of data
- ❖ General inefficiency
- ❖ Balancing exploration (trying new actions) versus exploitation (using known actions that produce reward)

Meta-Reinforcement Learning

Divided into two parts,

- ❖ Inner Loop
  - ❖ The agent uses meta-knowledge from the outer loop for quick adaption to new tasks/environments
  - ❖ Effectiveness is measured based on the agent's performance
- ❖ Outer Loop
  - ❖ Develop and tune meta-knowledge
  - ❖ Examine how trends within tasks to enhance meta-knowledge
  - ❖ Updates based on inner loop feedback

Model Agnostic Meta Learning

Model Agnostic Meta Learning or MAML was made to address some shortcomings in the basic formula of Meta-RL and continue innovation.

This algorithm was made to keep the number of learned parameters constant or enforce unneeded constraints on the model- accessible to all models and easily combined. The goal is to achieve rapid adaption using few data points.

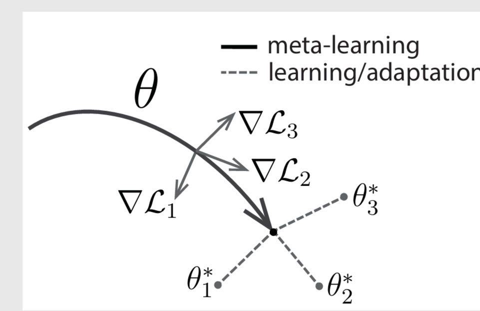


Fig 1. With θ as an agent, the path of growth is shown as learning tasks $\nabla L_1, \nabla L_2$, and ∇L_3 also for θ to transform into θ_1^*, θ_2^*, and θ_3^*.

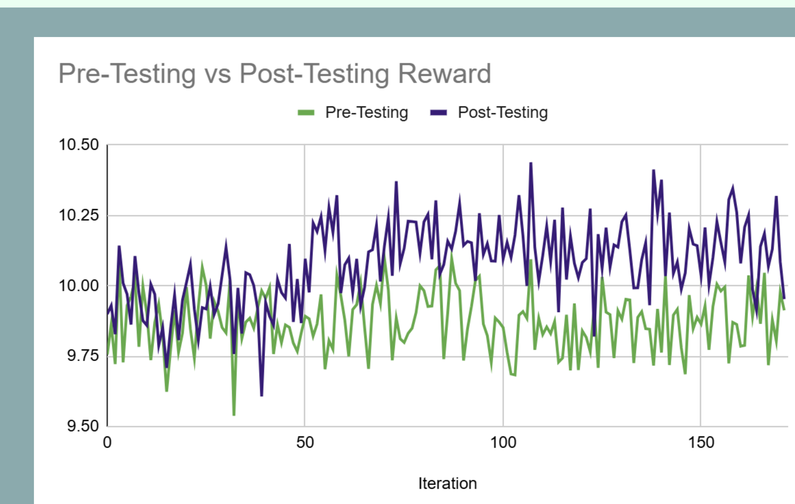


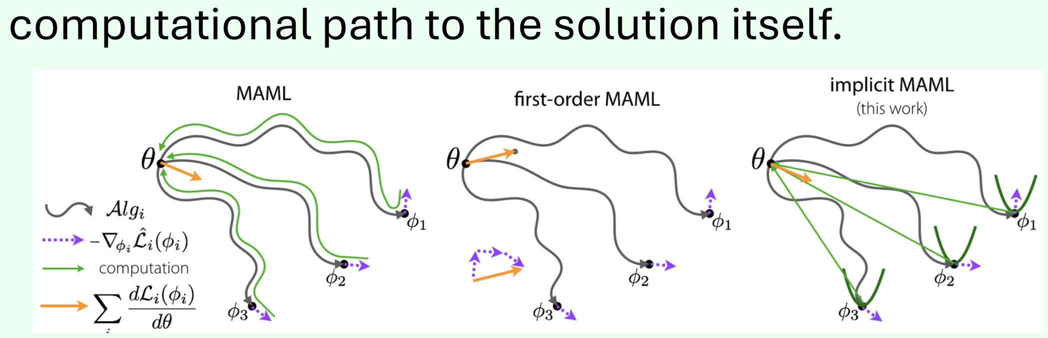
Fig 2. Looking at the chart, it is evident that testing had a significant impact between rewards pre-testing and post-testing, showing its efficiency.

Results

- ❖ MAML is able to return the most accurate results in the least amount of time during a locomotion simulation task
- ❖ In 2D navigation, MAML performed as well as Oracle and well above the pretrained model within 2 gradient steps
- ❖ For RL, it had the overall best accuracy between 1 and 5 gradient updates and varying degrees of accuracy compared to other meta models

Implicit Model Agnostic Meta Learning

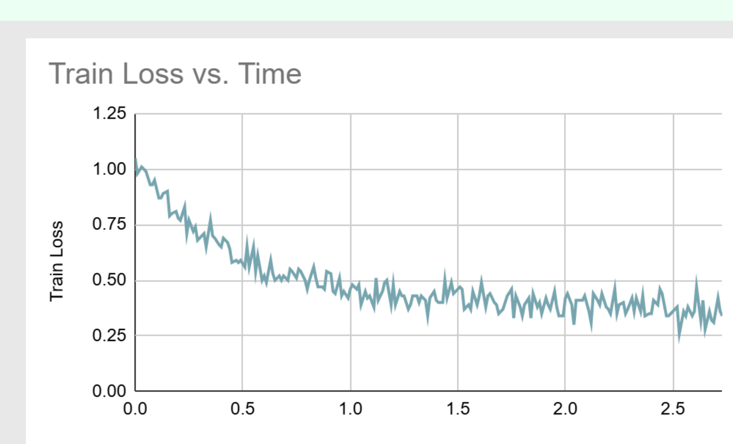
Implicit Model Agnostic Meta Learning (iMAML) was made to reduce computational cost and memory footprints. It does so through the use of implicit differentiation, saving only the solution to the inner loop optimizer and not the computational path to the solution itself.



Results

- ❖ Compared to MAML in few-shot image recognition, iMAML consistently uses less gradient descent steps and memory storage than MAML
- ❖ It also produces fewer terminal errors during runtime

Fig 3. This chart displays how the loss (difference between the model's predictions and the true values) decreases as the model continues running.



Meta Gradient Reinforced Learning

Meta Gradient Reinforced Learning, MGRL, sought to optimize the agent's reward or cumulative reward through treating it like a parametric function with tuneable meta-parameters like the discount factor γ or the bootstrapping label θ.

Results

- ❖ Any use of MGRL outpaced IMPALA in accuracy on all fronts
- ❖ Meta tuning more than one factor significantly improved performance, and even more so with organic human starting positions

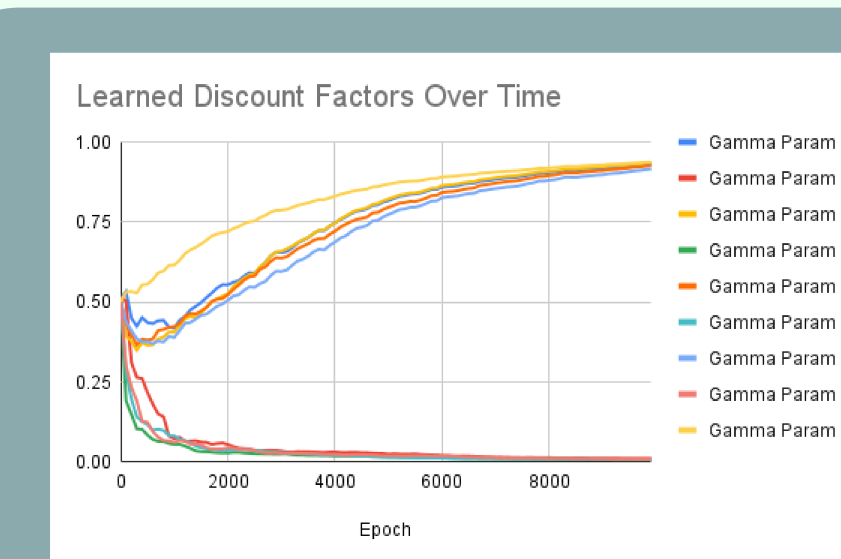
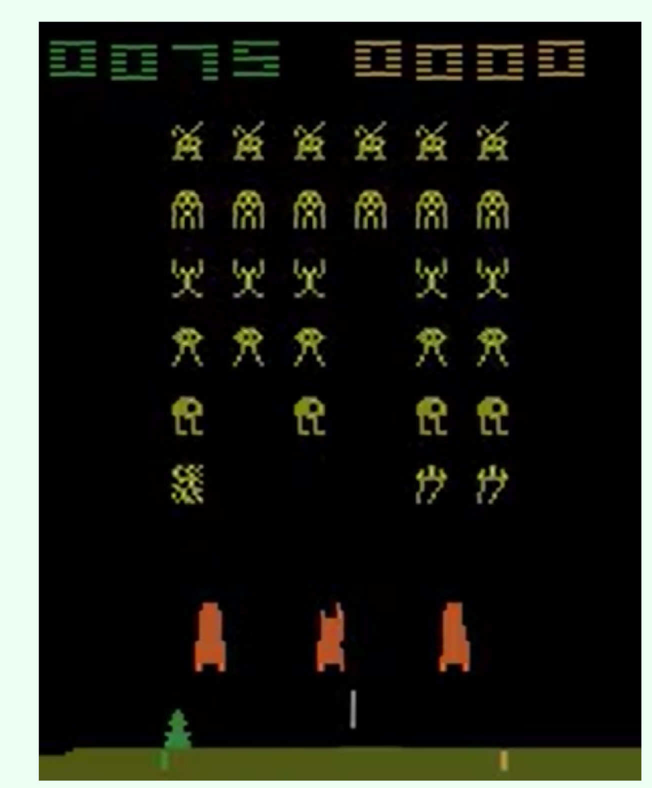


Fig 5. As the discount factor determines the urgency of a reward, the figure shows the model adjusting and looking for the most optimal factors in a model.

Deep Q Learning



Deep Q Learning combines Q learning, a value-based reinforcement learning algorithm that learns the optimal action-value function named Q, and deep learning neural networks.

This is an element that has become a large facet of RL and can be used for aspects of Meta RL in its inner loop.

Fig 6. Here it is being used on the side to play a classic Atari Game, Space Fighters

References

Finn, C., Abbeel, P., & Levine, S. (2017, March 9). *Model-Agnostic Meta-Learning for fast adaptation of deep networks*. arXiv.org. https://arxiv.org/abs/1703.03400

Huggingface. (n.d.). *deep-rl-class/units/en/unit3/deep-q-algorithm.mdx at main · huggingface/deep-rl-class*. GitHub. https://github.com/huggingface/deep-rl-class/blob/main/units/en/unit3/deep-q-algorithm.mdx

ICML 2019 Meta-Learning Tutorial. (n.d.). https://sites.google.com/view/icml19metalearning

Jiang, S., Ge, Y., Yang, X., Yang, W., & Cui, H. (2024). UAV control method combining reptile Meta-Reinforcement learning and Generative Adversarial Imitation learning. *Future Internet, 16*(3), 105. https://doi.org/10.3390/fi16030105

Metaopt. (n.d.). *GitHub - metaopt/torchopt: TorchOpt is an efficient library for differentiable optimization built upon PyTorch*. GitHub. https://github.com/metaopt/torchopt/tree/main

Meta-RL Tutorial. (n.d.). https://sites.google.com/view/meta-rl-tutorial-2023/home

Rajeswaran, A., Finn, C., Kakade, S., & Levine, S. (2019, September 10). *Meta-Learning with Implicit Gradients*. arXiv.org. https://arxiv.org/abs/1909.04630

Xu, Z., Hado, V. H., & Silver, D. (2018, May 24). *Meta-Gradient Reinforcement learning*. arXiv.org. https://arxiv.org/abs/1805.09801

Yunjey. (n.d.). *GitHub - yunjey/pytorch-tutorial: PyTorch Tutorial for Deep Learning Researchers*. GitHub. https://github.com/yunjey/pytorch-tutorial/tree/master