Transfer Learning Algorithms in Reinforcement Learning

Giordano Arcieri

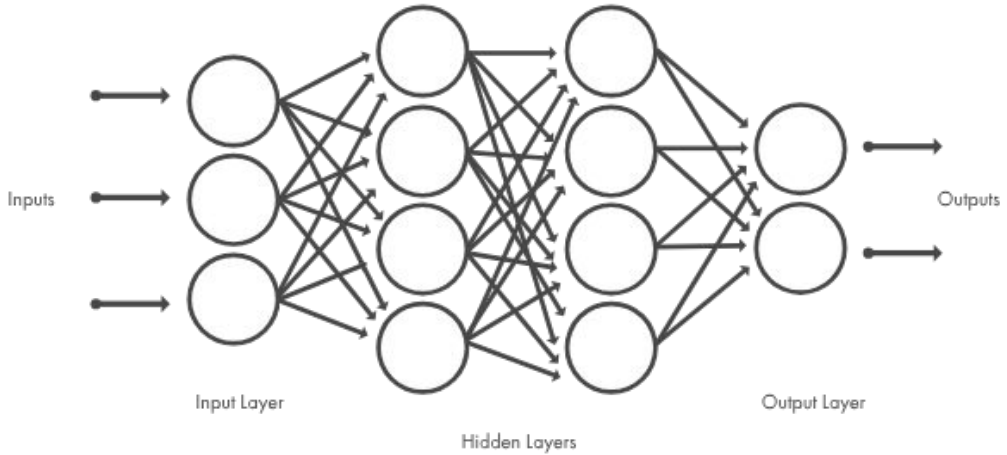
Introducing Deep Learning

Weights/Bisais - constants of the function

Forward Propagation (O(L * n^2))

Backward Propagation (O(L * n^2))

Gradient Descent (O(T * L * n^2))



Introducing Reinforcement Learning

Agent - student trying to maximize reward

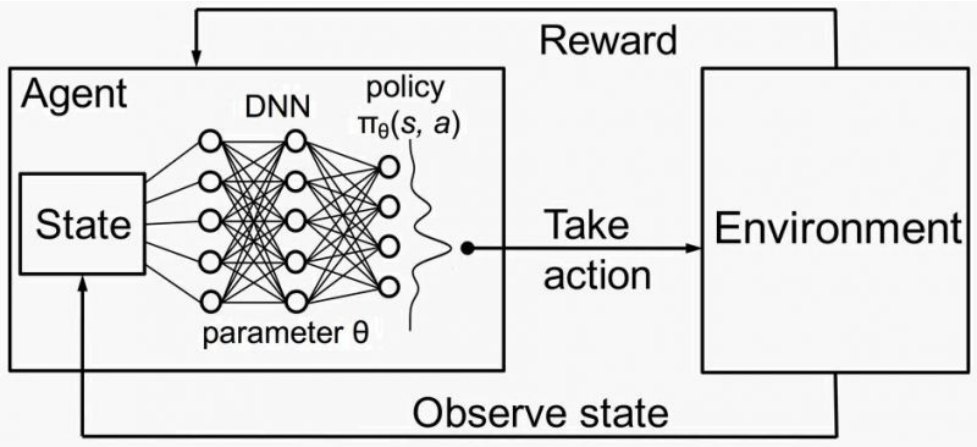
State - current state of the environment

Reward - how good was the action?

Policy - maps states to action in order to maximize rewards

Value - maps states to the value of being in that state

PPO - a DRL algorithm to train the two neural networks

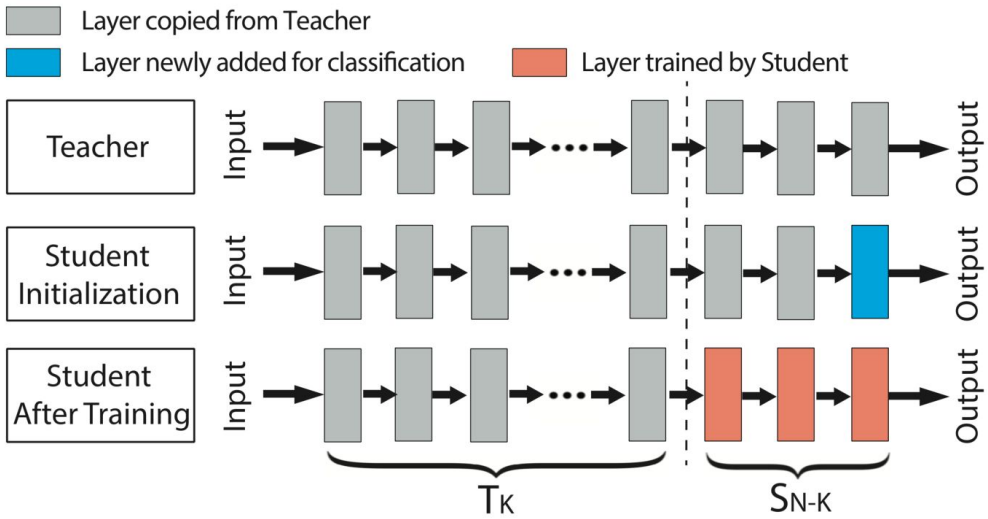


V(s) = how good is it to be in state s

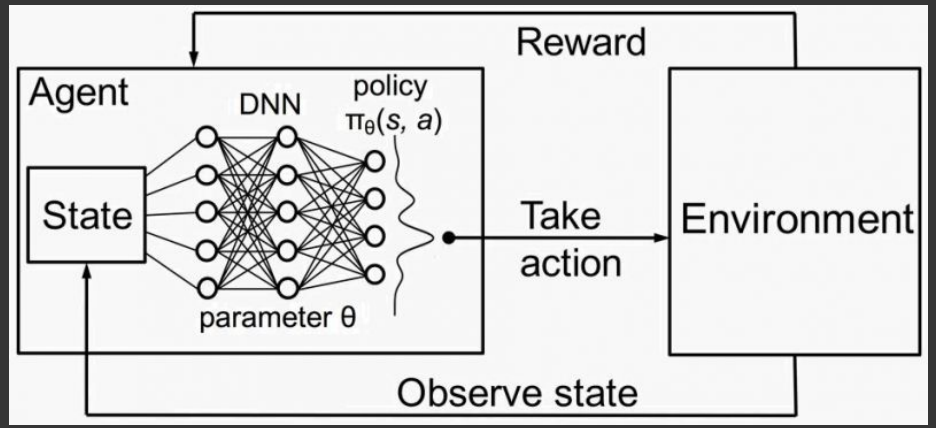
Introducing Transfer Learning

Algorithm:

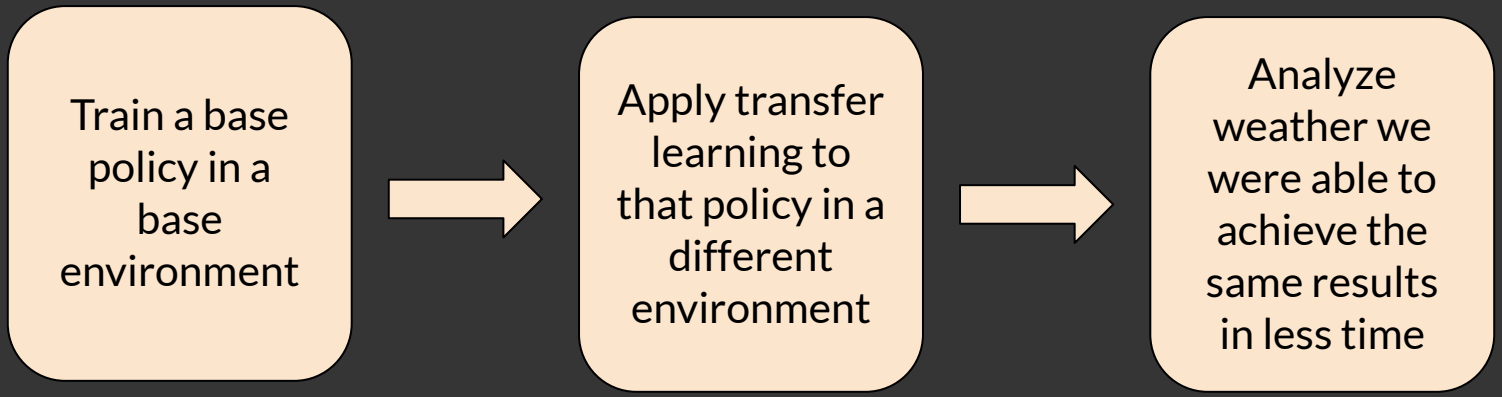
- **Start with a trained base policy**
- **Freeze first k layers O(k)**
- **Reinitialized the last n-k layers O(n-k)**
- **Retrain the policy on a new environment O(T * (n−k) * m^2)**



Can Transfer Learning be applied to an RL Policy to decrease the training time for a new task?



Methodology



Methodology

Policy Architecture:

- **[24 (Input Layer) -> 64 -> 64 -> 64 -> 64 -> 64 -> 4 (Output Layer)]**

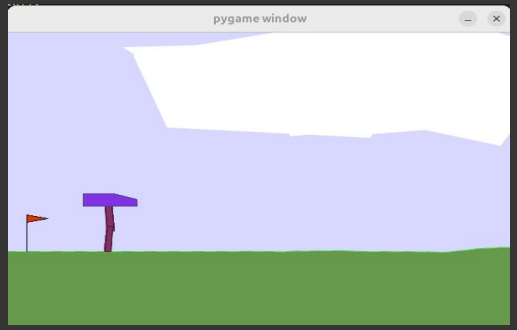
Train base policy on a base environment

Transferred Policy: Freeze first 5 layers and re-initialize last 2

Train transferred policy on a new environment

Environments

Base Env



Slippery Env

Bumpy Env

Slippery and bumpy Env

Base Model

Training Time: 20M iterations

Base Model Results:

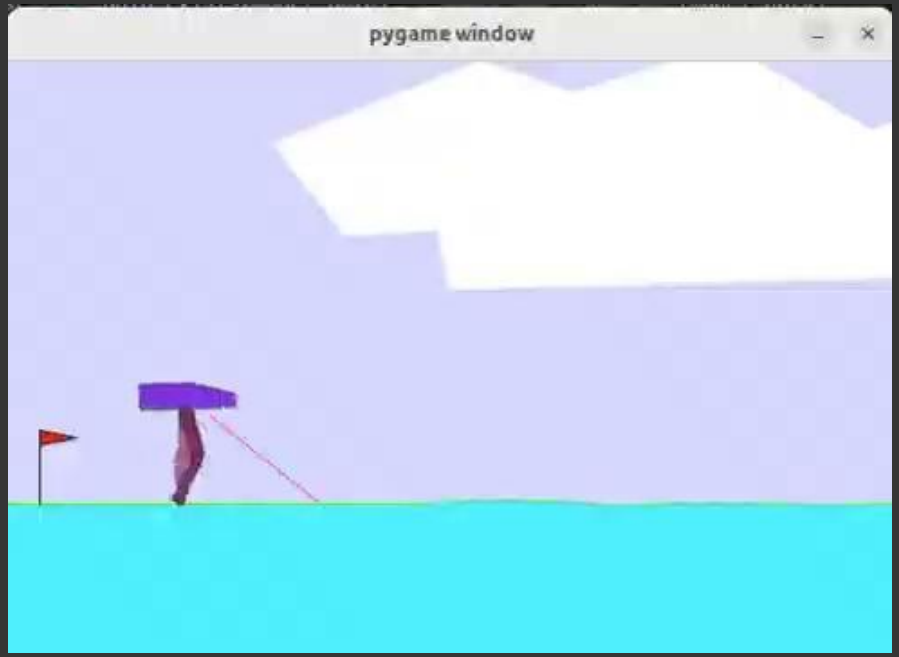
- Mean Rewards: 161.24
- Mean Time Elapsed: 1252.48

Transferred Model

Training Time: 8M iterations

Transferred Model Results:

- Mean Rewards: 158.2
- Mean Time Elapsed: 1228.4

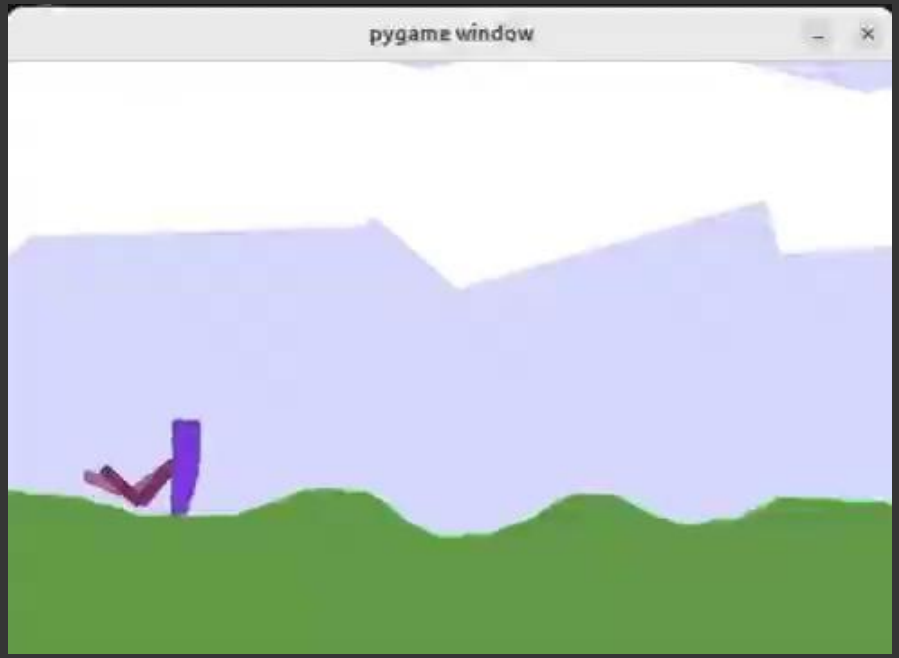


Transferred Model

Training Time: 8M iterations

Transferred Model Results:

- Mean Rewards: 18.36
- Mean Time Elapsed: 573.4

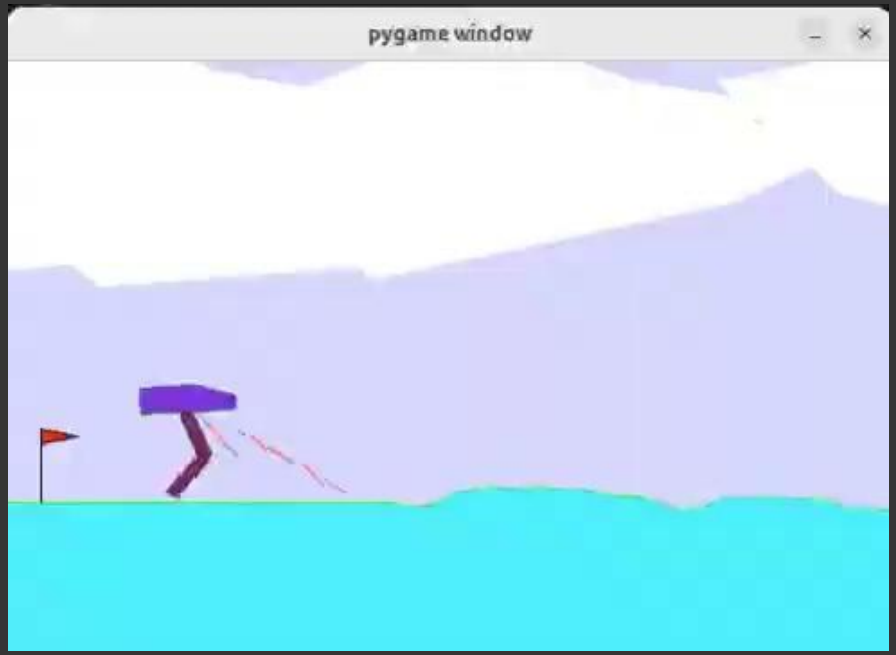


Transferred Model

Training Time: 8M iterations

Base Model Results:

- Mean Rewards: -11.12
- Mean Time Elapsed: 404.97



Analysis

Base Policy Rewards: 161.25

Transferred Policy (Slippery Env) Rewards: 158.2

Transferred Policy (Bumpy Env) Rewards: 18.36

Transferred Policy (Hard Env) Rewards: -11.12

Although we saw some possibility of transfer learning working it did not really cut training time and the same level of rewards was never achieved

What I would do differently?

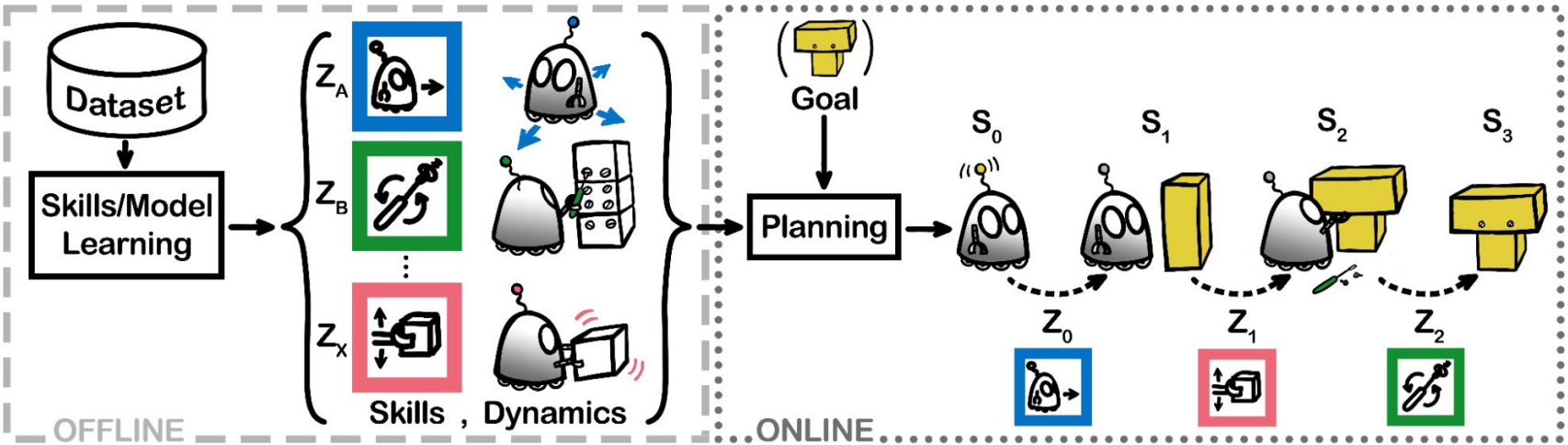
- Overall this experiment failed and I was not really able to conclude whether transfer learning is effective

- I believe this was due to over-fitting, representation relevance, overparameterized.

- A much more common way that transfer learning is applied to RL is by learning skills and reusing them when they can be used

Much better experiment

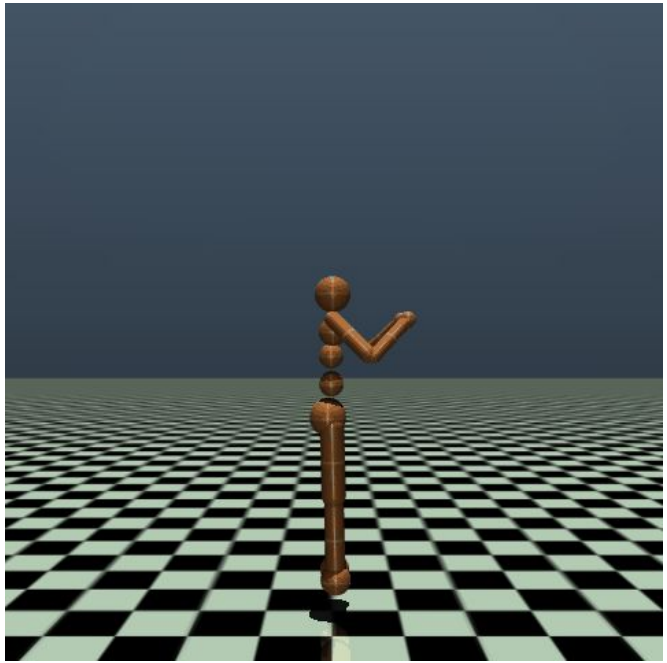
It is much more common to have robots learn “skills”. This way when they are being trained they can find themselves in a similar position and use a “skill”.



New Experiment

I started working on this new experiment:

- Base goal: Learn some skills
  - Standing up
  - Walking
  - Running
  - Jumping
- New goals:
  - Navigate a maze



References

I did create a github link to this project along with a small research paper on this experiment:
https://github.com/giordano-arcieri/TransferRL

Papers:
Proximal Policy Optimization Algorithms arXiv:1707.06347 [cs.LG]
A Comprehensive Survey on Transfer Learning arXiv:1911.02685 [cs.LG]
Transfer Learning in Deep Reinforcement Learning: A Survey arXiv:2009.07888 [cs.LG]
Skill-based Model-based Reinforcement Learning arXiv:2207.07560 [cs.LG]

Tools:
Gym's Bipedal Walker Env https://gymnasium.farama.org/environments/box2d/bipedal_walker/
Stable Baselines PPO Model https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html

Question?

How does RL work?

How does PPO work?

Why did Transfer Learning not work?