Deep Reinforcement Learning (Sp25)

Instructor: Dr. Mohammad Hossein Rohban

Quiz Solutions - Lecture 27

Solution by : Benyamin Naderi

Q1

How can we improve zero-shot transfer performance?

Answer: Zero-shot transfer performance can be improved by enhancing generalization and representation learning so that the model can effectively handle unseen tasks or domains.
Explanation: In zero-shot transfer, a model is expected to perform well on tasks or environments it has never encountered during training. Improving zero-shot transfer involves strategies that help the model learn more general, task-invariant features and reasoning capabilities.

- **1. Better representation learning:** Train models to extract task-agnostic or semantic representations using techniques like contrastive learning, self-supervised learning, or large-scale pretraining.
- **2. Meta-learning:** Use meta-learning methods (e.g., MAML, PEARL) that optimize for adaptability and generalization across diverse tasks, helping models handle new tasks without additional training.
- **3. Domain randomization and augmentation:** Expose the model to a wide range of variations (e.g., environments, dynamics, visuals) during training so that it learns to generalize beyond seen conditions.
- **4. Goal or context conditioning:** Condition policies or models on high-level task descriptors, goals, or context embeddings that help guide behavior in unseen situations.
- **5. Regularization and invariant learning:** Apply methods such as adversarial regularization or invariant risk minimization (IRM) to encourage robustness to domain shifts.
- **6. Knowledge transfer from large pre-trained models:** Leverage foundation models (e.g., pre-trained vision-language or policy models) that already encode broad world knowledge useful for zero-shot settings.

Q2

What are the three main challenges in RL transfer learning?

A)Source domain representations may fail in the target domain

B)Target domain may prohibit actions possible in the source

C)Exploration needed during finetuning clashes with deterministic optimal policies.

Answer: A), B), and C) — all three are valid challenges in reinforcement learning (RL) transfer learning.
Explanation: Transfer learning in RL aims to leverage knowledge learned from a source environment (or task) to improve learning in a new target environment. However, this process faces several key challenges:

- **A) Source domain representations may fail in the target domain:** Representations learned in one environment may not generalize well if the target domain differs in dynamics, reward functions, or observation spaces. This leads to poor transfer or even *negative transfer*.
- **B) Target domain may prohibit actions possible in the source:** The action space may differ between domains — some actions available in the source may be invalid or unsafe in the target environment. This mismatch complicates policy reuse and adaptation.
- **C) Exploration needed during finetuning clashes with deterministic optimal policies:** In RL, effective fine-tuning requires exploration in the target domain to adapt to new conditions. However, policies transferred from the source domain often become highly deterministic, reducing their ability to explore and discover better strategies.

Q3

What are four ways knowledge can be stored and reused across reinforcement learning tasks?

1. **1. Policy transfer:** Reusing a trained policy (or part of it) from a source task as the initialization or prior for a new task. This allows faster adaptation by leveraging previously learned action mappings:

$$\pi_{\text{target}} \leftarrow \pi_{\text{source}}$$

2. **2. Value function transfer:** Transferring learned value functions or Q-functions to guide exploration or bootstrapping in new environments. This helps the agent estimate rewards and expected returns more effectively at the start of training.
3. **3. Representation (feature) transfer:** Reusing learned state or feature representations from previous tasks. Shared neural encoders or latent embeddings capture general patterns useful for multiple tasks:

$$\phi(s)_{\text{target}} \leftarrow \phi(s)_{\text{source}}$$

4. **4. Model transfer:** Transferring components of a learned environment model (e.g., dynamics or reward models) for model-based RL. This allows the agent to simulate or plan efficiently in the new task using prior knowledge about the environment's structure.

Q4

What is the most effective way to maximize forward transfer in sim-to-real reinforcement learning?
i) Using identical parameters in both simulation and real-world environments
ii) Training with randomized domain variations (dynamics, appearance, etc.)
iii) Increasing the number of layers in the neural network
iv) Deploying the policy without any prior simulation trainin

Answer: ii) Training with randomized domain variations (dynamics, appearance, etc.)
Explanation: In **sim-to-real reinforcement learning**, the goal is to train an agent in simulation and then transfer the learned policy to the real world without significant performance degradation. The main challenge is the *reality gap* — differences between simulated and real-world physics, noise, visuals, and dynamics.
Domain randomization is the most effective strategy to maximize forward transfer. During training, the agent is exposed to a wide range of randomized simulation conditions, such as:

- Physical dynamics (mass, friction, gravity)
- Visual appearance (lighting, textures, colors)
- Sensor noise or latency

By training under this diversity, the policy learns to be robust to variations and generalizes well to the real environment.

Q5

How does adversarial domain adaptation enforce feature invariance?

Answer: Adversarial domain adaptation enforces **feature invariance** by training a feature extractor and a domain discriminator in an adversarial manner.
Explanation: The goal is to learn a shared feature space where source and target domain features are indistinguishable.

- The **feature extractor** F maps input data from both domains into a latent representation.
- The **domain discriminator** D tries to classify whether a feature comes from the source or target domain.
- The feature extractor is trained *adversarially* to fool the discriminator — i.e., it minimizes:

$$\mathcal{L}_{\text{adv}} = -\mathbb{E}[\log D(F(x))]$$

  while D maximizes its classification accuracy.

This min–max game forces F to produce domain-invariant features:

$$\min_F \max_D \ \mathcal{L}_{\text{adv}}(F, D)$$