Deep Reinforcement Learning (Sp25)

Instructor: Dr. Mohammad Hossein Rohban

Quiz Solutions - Lecture 28

Solution by : Benyamin Naderi

Q1

Which of the following best describes the goal of MAML in the context of few-shot learning?

A)To train a model that performs well on a specific task using a large dataset.

B)To find a universal model that performs equally well on all possible tasks without adaptation.

C)To find an initial model that can quickly adapt to new tasks with just a few gradient steps and few data points.

D)To learn task-specific models independently and average their parameters.

Correct Answers: C

Answer: C) To find an initial model that can quickly adapt to new tasks with just a few gradient steps and few data points.
Explanation: Model-Agnostic Meta-Learning (MAML) is designed to train a model initialization that can be rapidly adapted to new tasks using only a small number of training examples and gradient updates. In few-shot learning, this means the model learns a good starting point that generalizes across tasks and can quickly specialize to a new one.

- **Option A:** Incorrect. MAML does not aim to perform well on a single specific task or rely on a large dataset; instead, it focuses on generalizing across multiple small tasks.
- **Option B:** Incorrect. MAML does not attempt to create a universal model that performs well on all tasks without adaptation. The key idea of MAML is *fast adaptation*, not universal performance.
- **Option C: Correct.** The goal of MAML is precisely to learn a good initialization that can quickly adapt to new tasks using few data points and few gradient updates.
- **Option D:** Incorrect. MAML does not independently train task-specific models and then average their parameters (that is closer to methods like Reptile or federated averaging). Instead, it optimizes a shared initialization through meta-learning.

Q2

Explain why MAML requires computing second-order derivatives during training?

A)To train a model that performs well on a specific task using a large dataset.

B)To find a universal model that performs equally well on all possible tasks without adaptation.

C)To find an initial model that can quickly adapt to new tasks with just a few gradient steps and few data points.

D)To learn task-specific models independently and average their parameters.

Answer: MAML requires computing second-order derivatives because the meta-learning process involves differentiating through the inner-loop gradient update used to adapt the model to a specific task.
Explanation: In MAML, training occurs in two stages for each task:

1. **Inner loop:** The model parameters θ are updated to task-specific parameters θ' using a gradient descent step:
$$\theta' = \theta - \alpha \nabla_\theta \mathcal{L}_{\text{task}}(\theta)$$
2. **Outer loop:** The meta-objective then updates the original parameters θ based on how well θ' performs on the task:
$$\min_\theta \sum_{\text{tasks}} \mathcal{L}_{\text{task}}(\theta')$$

To compute the gradient of the outer loss with respect to θ, we must differentiate through the inner update step. This involves taking the derivative of a gradient (i.e., $\nabla_\theta(\nabla_\theta \mathcal{L})$), which introduces **second-order derivatives** or **Hessian terms**.
In summary: MAML requires second-order derivatives because the meta-optimization depends on how the inner-loop gradient update changes with respect to the model's initial parameters. This is what allows MAML to learn an initialization that is highly sensitive to task-specific adaptation.

Q3

What distinguishes meta-learning from multi-task learning?

A)Meta-learning uses deeper neural network

B)Meta-learning optimizes the learning process itself

C)Multi-task learning requires more training data

Answer: B) Meta-learning optimizes the learning process itself.
Explanation: Meta-learning, also known as "learning to learn," focuses on improving the learning algorithm itself so that the model can learn new tasks more efficiently from limited data. It operates at a higher level — instead of learning directly from data, it learns how to update model parameters or how to adapt quickly to new tasks.

- **Option A:** Incorrect. The difference between meta-learning and multi-task learning is not about using deeper neural networks. The architecture depth is not the defining characteristic.
- **Option B: Correct.** Meta-learning explicitly aims to optimize the *learning process* — i.e., how models learn — by adjusting learning rules, initializations, or update mechanisms to generalize better across tasks.
- **Option C:** Incorrect. While multi-task learning can involve more data since it trains on multiple tasks simultaneously, this is not the defining distinction between the two paradigms.

In summary: Meta-learning learns *how to learn*, whereas multi-task learning learns to perform several tasks simultaneously using shared representations.

Q4

In MAML-style meta-RL, the inner loop update is:

The correct inner-loop update in MAML-style meta-RL is:

$$\boxed{\phi_i \leftarrow \theta - \alpha \nabla J_i(\theta)}$$

Q5

How does contextual policy (a|s,) relate to meta-RL?
i) functions like the learned in meta-RL
ii) Contextual policies are only for supervised learning
iii) Meta-RL never uses state information

Answer: i) ω functions like the learned ϕ_i in meta-RL.
Explanation: In meta-reinforcement learning, the agent aims to quickly adapt its policy to new tasks. After the inner-loop adaptation, each task i has its own adapted policy parameters ϕ_i, derived from the shared initialization θ.
A **contextual policy** $\pi(a \mid s, \omega)$ extends this idea by introducing a latent variable or context vector ω that represents task-specific information. This ω captures aspects of the current task or environment — effectively encoding the same role as the adapted parameters ϕ_i in meta-RL.

$$\pi(a \mid s, \omega) \quad \text{is analogous to} \quad \pi_{\phi_i}(a \mid s)$$

where ω (context) provides task-specific adaptation without explicitly updating parameters via gradient descent.
Why the others are incorrect:

- **ii) Contextual policies are only for supervised learning:** Incorrect. Contextual policies are widely used in reinforcement learning to handle multiple tasks or changing environments.
- **iii) Meta-RL never uses state information:** Incorrect. Meta-RL explicitly depends on state information for policy decisions and adaptation; the state s is a fundamental part of the policy $\pi(a \mid s)$.

Q6

Goal-conditioned policies can handle non-goal-oriented tasks.

- True
- False

Answer: False
Explanation: Goal-conditioned policies are specifically designed to handle *goal-oriented tasks*, where the policy is conditioned on a goal variable g. The policy is represented as:

$$\pi(a \mid s, g)$$

where s is the current state and g defines the desired target or goal state.
These policies learn to generalize over different goals, enabling the agent to adapt its behavior depending on the given goal. However, in **non-goal-oriented tasks** (i.e., tasks without an explicit goal specification), there is no goal variable g to condition on, so goal-conditioned policies are not necessary or directly applicable.