Advanced Topics in Deep Reinforcement Learning

RL Algorithms

Theresa Eimer

Challenges for RL Algorithms

Challenges for RL Algorithms

- Acquiring good training data

Challenges for RL Algorithms

- Acquiring good training data
- Good policy updates

Challenges for RL Algorithms

- Acquiring good training data
- Good policy updates
- Stable improvement during training

Challenges for RL Algorithms

- Acquiring good training data
- Good policy updates
- Stable improvement during training
- Being efficient with the interactions we collect

Challenges for RL Algorithms

- Acquiring good training data
- Good policy updates
- Stable improvement during training
- Being efficient with the interactions we collect
- Dealing with large action & observation spaces

Challenges for RL Algorithms

- Acquiring good training data
- Good policy updates
- Stable improvement during training
- Being efficient with the interactions we collect
- Dealing with large action & observation spaces
- Learning a range of different policies

Challenges for RL Algorithms

- Acquiring good training data
- Good policy updates
- Stable improvement during training
- Being efficient with the interactions we collect
- Dealing with large action & observation spaces
- Learning a range of different policies

Challenges for RL Algorithms

- Acquiring good training data
  Interaction, exploration, stable improvement
- Good policy updates
- Stable improvement during training
- Being efficient with the interactions we collect
- Dealing with large action & observation spaces
- Learning a range of different policies

Challenges for RL Algorithms

- Acquiring good training data
  Interaction, exploration, stable improvement
- Good policy updates
  Optimizers, losses, hyperparameters
- Stable improvement during training
- Being efficient with the interactions we collect
- Dealing with large action & observation spaces
- Learning a range of different policies

Challenges for RL Algorithms

- Acquiring good training data
  Interaction, exploration, stable improvement
- Good policy updates
  Optimizers, losses, hyperparameters
- Stable improvement during training
  Training data, hyperparameters, good updates
- Being efficient with the interactions we collect

- Dealing with large action & observation spaces

- Learning a range of different policies

Challenges for RL Algorithms

- Acquiring good training data
  Interaction, exploration, stable improvement
- Good policy updates
  Optimizers, losses, hyperparameters
- Stable improvement during training
  Training data, hyperparameters, good updates
- Being efficient with the interactions we collect
  Buffers, models, good updates
- Dealing with large action & observation spaces

- Learning a range of different policies

Challenges for RL Algorithms

- Acquiring good training data
  Interaction, exploration, stable improvement
- Good policy updates
  Optimizers, losses, hyperparameters
- Stable improvement during training
  Training data, hyperparameters, good updates
- Being efficient with the interactions we collect
  Buffers, models, good updates
- Dealing with large action & observation spaces
  Exploration, good updates
- Learning a range of different policies

Challenges for RL Algorithms

- Acquiring good training data
  Interaction, exploration, stable improvement
- Good policy updates
  Optimizers, losses, hyperparameters
- Stable improvement during training
  Training data, hyperparameters, good updates
- Being efficient with the interactions we collect
  Buffers, models, good updates
- Dealing with large action & observation spaces
  Exploration, good updates
- Learning a range of different policies
  Exploration, efficiency, good updates

Challenges for RL Algorithms

- Acquiring good training data
  Interaction, exploration, stable improvement
- Good policy updates
  Optimizers, losses, hyperparameters
- Stable improvement during training
  Training data, hyperparameters, good updates
- Being efficient with the interactions we collect
  Buffers, models, good updates
- Dealing with large action & observation spaces
  Exploration, good updates
- Learning a range of different policies
  Exploration, efficiency, good updates

Generalist algorithms

What makes up an RL algorithm?

Interaction
Paradigm

What makes up an RL algorithm?

Data Acquisition

Exploration

Policy

Interaction
Paradigm

What makes up an RL algorithm?

Data Acquisition

Exploration

Policy

Interaction Paradigm

Interaction Backup

Buffers

Models

What makes up an RL algorithm?

Data Acquisition

- Exploration
- Policy

Policy Updating

- Objective functions
- Optimizers
- Losses
- Hyperparameters

Interaction Paradigm

Interaction Backup

- Buffers
- Models

What makes up an RL algorithm?

Data Acquisition

Exploration

Policy

Policy Updating

Objective functions

Optimizers

Losses

Hyperparameters

Interaction Paradigm

Timing

Interaction Backup

Buffers

Models

What makes up an RL algorithm?

Data Acquisition

- Exploration
- Policy

Policy Updating

- Objective functions
- Optimizers
- Losses
- Hyperparameters

Interaction Paradigm

Timing

Interaction Backup

- Buffers
- Models

Off-Policy-ness

Fully on-policy

Fully off-policy

Off-Policy-ness

Fully on-policy

No re-use of interactions

Fully off-policy

No new interactions

Off-Policy-ness

Fully on-policy

No re-use of interactions

Fully off-policy

No new interactions

Policy Search (without value baseline)

Offline RL

Off-Policy-ness

Fully on-policy

No re-use of interactions

Fully off-policy

No new interactions

Policy Search (without value baseline)

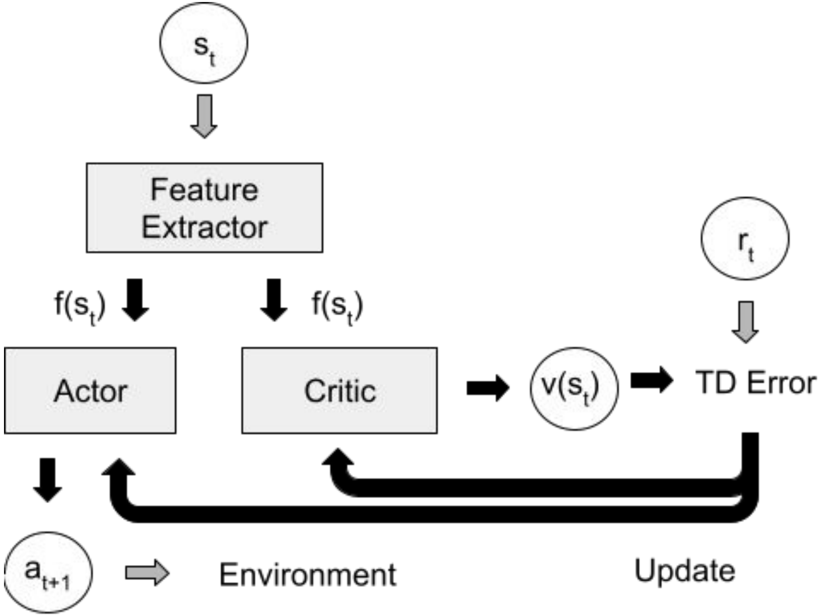
Policy Search with baseline

Off-policy online RL

Offline RL

Model-Free On-Policy Learning

- State of the art: Policy Gradient with value baselines
- Actor-critic architectures learn policy and baseline jointly



Model-Free On-Policy Learning

- State of the art: Policy Gradient with value baselines
- Actor-critic architectures learn policy and baseline jointly
- Actor predicts the base for action probability distribution
- Critic predicts the value function
- No interaction backup, instead updates across current rollouts
- Important ideas:

Model-Free On-Policy Learning

- State of the art: Policy Gradient with value baselines
- Actor-critic architectures learn policy and baseline jointly
- Actor predicts the base for action probability distribution
- Critic predicts the value function
- No interaction backup, instead updates across current rollouts
- Important ideas:
  - Parallel rollouts for efficiency
  - Limiting policy updates, e.g. via clipping in PPO [Schulman et al. 2017]
  - Exploration via intrinsic rewards and action noise
  - Separating representations for policy & values [Andrychowicz et al. 2021, Raileanu & Ferguson 2021]

Model-Free On-Policy Learning

- State of the art: Policy Gradient with value baselines
- Actor-critic architectures learn policy and baseline jointly
- Actor predicts the base for action probability distribution
- Critic predicts the value function
- No interaction backup, instead updates across current rollouts
- Important ideas:
  - Parallel rollouts for efficiency
  - Limiting policy updates, e.g. via clipping in PPO [Schulman et al. 2017]
  - Exploration via intrinsic rewards and action noise
  - Separating representations for policy & values [Andrychowicz et al. 2021, Raileanu & Ferguson 2021]
- Example Algorithms: PPO, TRPO, A2C/A3C, Impala

Model-Free Multi-Agent On-Policy Learning

- Two added important factors:
  - Knowledge about the other agents
  - Centralized vs decentralized information

Model-Free Multi-Agent On-Policy Learning

- Two added important factors:
  - Knowledge about the other agents
  - Centralized vs decentralized information
- We can model our belief about how the other agents will act:

Model-Free Multi-Agent On-Policy Learning

- Two added important factors:
  - Knowledge about the other agents
  - Centralized vs decentralized information
- We can model our belief about how the other agents will act:
  - Using the agent as a model for others [Raileanu et al. 2018]
  - Combining agents' predictions [Rashid et al. 2018]
  - Learning an explicit model for opponents [Yu et al. 2022]

Model-Free Multi-Agent On-Policy Learning

- Two added important factors:
  - Knowledge about the other agents
  - Centralized vs decentralized information
- We can model our belief about how the other agents will act:
  - Using the agent as a model for others [Raileanu et al. 2018]
  - Combining agents' predictions [Rashid et al. 2018]
  - Learning an explicit model for opponents [Yu et al. 2022]
- Especially in cooperative settings, we can choose to share information via:

Model-Free Multi-Agent On-Policy Learning

- Two added important factors:
  - Knowledge about the other agents
  - Centralized vs decentralized information
- We can model our belief about how the other agents will act:
  - Using the agent as a model for others [Raileanu et al. 2018]
  - Combining agents' predictions [Rashid et al. 2018]
  - Learning an explicit model for opponents [Yu et al. 2022]
- Especially in cooperative settings, we can choose to share information via:
  - Centralized critics [Yu et al. 2022]
  - Experience sharing [Gerstgrasser et al. 2023]

Model-Free Multi-Agent On-Policy Learning

- Two added important factors:
  - Knowledge about the other agents
  - Centralized vs decentralized information
- We can model our belief about how the other agents will act:
  - Using the agent as a model for others [Raileanu et al. 2018]
  - Combining agents' predictions [Rashid et al. 2018]
  - Learning an explicit model for opponents [Yu et al. 2022]
- Especially in cooperative settings, we can choose to share information via:
  - Centralized critics [Yu et al. 2022]
  - Experience sharing [Gerstgrasser et al. 2023]
- But: experience sharing makes training harder to implement, this may not be worth it [De Witt et al. 2020]

Model-Free Multi-Agent On-Policy Learning

Note: Off-policy MARL exists, is just less common

- Two added important factors:
  - Knowledge about the other agents
  - Centralized vs decentralized information
- We can model our belief about how the other agents will act:
  - Using the agent as a model for others [Raileanu et al. 2018]
  - Combining agents' predictions [Rashid et al. 2018]
  - Learning an explicit model for opponents [Yu et al. 2022]
- Especially in cooperative settings, we can choose to share information via:
  - Centralized critics [Yu et al. 2022]
  - Experience sharing [Gerstgrasser et al. 2023]
- But: experience sharing makes training harder to implement, this may not be worth it [De Witt et al. 2020]

Model-Free Online Off-Policy Learning

- We learn from previous experiences and continuously generate new ones

Model-Free Online Off-Policy Learning

- We learn from previous experiences and continuously generate new ones
- Two options:
  - Predicting Q-values and extracting a policy from them
  - Off-policy actor-critic, e.g. SAC [Haarnoja et al. 2018]

Model-Free Online Off-Policy Learning

- We learn from previous experiences and continuously generate new ones
- Two options:
  - Predicting Q-values and extracting a policy from them
  - Off-policy actor-critic, e.g. SAC [Haarnoja et al. 2018]
- Important ideas:

Model-Free Online Off-Policy Learning

- We learn from previous experiences and continuously generate new ones
- Two options:
  - Predicting Q-values and extracting a policy from them
  - Off-policy actor-critic, e.g. SAC [Haarnoja et al. 2018]
- Important ideas:
  - Buffer sampling strategies
  - Exploration beyond epsilon-greedy
  - Regular parameter resets [Nikishin et al. 2022]
  - Better representations, e.g. via self-prediction [Schwarzer et al. 2021]
  - Distributional RL [Bellemare et al. 2023]
  - Architecture choices for Q-networks
  - Alternative Losses [Farebrother et al. 2024]

Off-Policy Example: SOTA DQNs

- Rainbow DQN [Hessel et al. 2017, Obando-Ceron & Castro 2021]
  - Noisy Nets
  - Distributional policy
  - Prioritized Replay
  - Target Network
  - Dueling Architecture
  - Multi-step learning

Off-Policy Example: SOTA DQNs

- Rainbow DQN [Hessel et al. 2017, Obando-Ceron & Castro 2021]
- Updates in recent years: BBF [Schwarzer et al. 2023]
  - Target network
  - Prioritized replay
  - Self-predictive representations
  - High replay ratio
  - Periodic parameter resets
  - Larger networks
  - Weight decay

Off-Policy Example: SOTA DQNs

- Rainbow DQN [Hessel et al. 2017, Obando-Ceron & Castro 2021]
- Updates in recent years: BBF [Schwarzer et al. 2023]
- Why so little overlap?

Off-Policy Example: SOTA DQNs

- Rainbow DQN [Hessel et al. 2017, Obando-Ceron & Castro 2021]
- Updates in recent years: BBF [Schwarzer et al. 2023]
- Why so little overlap?
  - Some components proved to be not universally well-performing
  - Others are subsumed in a more general form in BBF
  - Better configurations
  - Potentially also better evaluations

Off-Policy Example: SOTA DQNs

- Rainbow DQN [Hessel et al. 2017, Obando-Ceron & Castro 2021]
- Updates in recent years: BBF [Schwarzer et al. 2023]
- Why so little overlap?
  - Some components proved to be not universally well-performing
  - Others are subsumed in a more general form in BBF
  - Better configurations
  - Potentially also better evaluations

How do we judge a state of the art algorithm?

Offline Learning

- Unfortunately, off-policy online methods often don’t work as well without any new online interactions [Ostrovski & Castro 2021]

Offline Learning

- Unfortunately, off-policy online methods often don’t work as well without any new online interactions [Ostrovski & Castro 2021]
- We can simply imitate from the dataset
  - Often not considered RL per se, but very strong in the offline setting
  - Example algorithms: BC, MARWIL

Offline Learning

- Unfortunately, off-policy online methods often don't work as well without any new online interactions [Ostrovski & Castro 2021]
- We can simply imitate from the dataset
  - Often not considered RL per se, but very strong in the offline setting
  - Example algorithms: BC, MARWIL
- We can change RL algorithms to work better on offline data
  - Conservative predictions [Kumar et al. 2020]
  - Supervised learning [Chen et al. 2021]

Offline Learning

- Unfortunately, off-policy online methods often don't work as well without any new online interactions [Ostrovski & Castro 2021]
- We can simply imitate from the dataset
  - Often not considered RL per se, but very strong in the offline setting
  - Example algorithms: BC, MARWIL
- We can change RL algorithms to work better on offline data
  - Conservative predictions [Kumar et al. 2020]
  - Supervised learning [Chen et al. 2021]
- Big challenge: how to evaluate a policy?
  - Access to online evaluation is assumed (potentially not realistic)
  - We have to estimate policy quality, e.g. via errors [Zitovsky et al. 2023]

Model-Based Learning

- In any interaction paradigm or algorithm, we can use a model of the environment instead of the actual environment

Model-Based Learning

- In any interaction paradigm or algorithm, we can use a model of the environment instead of the actual environment
- Advantages:

Model-Based Learning

- In any interaction paradigm or algorithm, we can use a model of the environment instead of the actual environment
- Advantages:
  - Interactions are cheap
  - We can look at alternatives -> planning with the model
  - Sometimes the model might be easier to learn than the policy

Model-Based Learning

- In any interaction paradigm or algorithm, we can use a model of the environment instead of the actual environment
- Advantages:
  - Interactions are cheap
  - We can look at alternatives -> planning with the model
  - Sometimes the model might be easier to learn than the policy
- Common model task: predict reward from (s,a)

Model-Based Learning

- In any interaction paradigm or algorithm, we can use a model of the environment instead of the actual environment
- Advantages:
  - Interactions are cheap
  - We can look at alternatives -> planning with the model
  - Sometimes the model might be easier to learn than the policy
- Common model task: predict reward from (s,a)
- Models are usually learned with supervised learning

Model-Based Learning

- In any interaction paradigm or algorithm, we can use a model of the environment instead of the actual environment
- Advantages:
  - Interactions are cheap
  - We can look at alternatives -> planning with the model
  - Sometimes the model might be easier to learn than the policy
- Common model task: predict reward from (s,a)
- Models are usually learned with supervised learning
- Probably best model-based algorithm currently: Dreamer V3 [Hafner et al. 2023]

My Understanding of RL Algorithms

- ❑ I know the elements of RL algorithms
- ❑ I can describe at least one algorithm per domain
- ❑ I know at least one important idea per domain

- ❑ I know the important ideas for each domain
- ❑ I understand why they are important research directions
- ❑ I can explain where the combination of some ideas can work

- ❑ I understand the basic idea of most algorithms
- ❑ I theorize where one algorithm could be better than another
- ❑ I can propose an algorithm given a problem setting

Examples for Seminar Session Ideas

- Applying algorithms to our settings (aka creating baselines)
- Investigating differences between methods
- Applying an algorithm to a domain it’s not usually used in (e.g. offline DQN)
- Thinking about why a certain direction of algorithm research might (not) work for one of our settings (e.g. highly parallel evaluations or model-based RL)
- Suggesting a change to a common algorithm that would make sense for us
- …