Seminar in Deep Reinforcement Learning

Topics and Resources

What should a good presentation include?

1. Motivate the topic - why are people even doing research in this area?
2. Make sure everybody gets the basics - there is no point in explaining the details of an algorithm to someone who did not get the idea
3. Draw novel, thought provoking connections - Does the topic you are presenting relate to something that the author did not think about? In which way? What do you see as outcome of this connection?
4. Teach something new - the lecture is a success if everybody learned something interesting. Given the huge amount of papers in Deep Reinforcement Learning no one in the room will be aware of all papers in your topic - find something interesting and present it. Make sure however to not miss point 2.

18.02. Introduction to Deep Reinforcement Learning (DRL)

In the first seminar we will have a broad overview over what deep reinforcement learning (DRL) is and why we are even interested in investigating DRL algorithms. We will shortly refresh the basics of value based methods and policy gradient based methods, and where function approximation in the form of deep learning comes in. We will introduce the seminal algorithms that started the field (DQN and A3C) and discuss their Pros and Cons. This introductory seminar will also provide an overview over the sub-topics discussed in the following weeks for a better orientation in the research area.
Further, the coding challenge will be introduced.

25.02. Deep Learning and Neural Architecture

This lecture should give an introduction into function approximation using deep learning. It should cover basic motivations for using deep learning such as the universal function approximation theorem and newer insights such as the deep double descent phenomena. The lecture should also give an overview over basic neural network architectures, including

- Convolutional Neural Networks
- Recurrent Neural Networks (LSTMs)
- Residual Networks
- Maybe also the newer Transformer architecture

The lecture might also provide an overview of deep learning frameworks, there Pro and Cons (Tensorflow vs. PyTorch).
To connect the lecture to the rest of the seminar, consider incorporating the Dueling Network Architecture for Deep Reinforcement Learning or Stabalizing Transformers for Reinforcement Learning

03.03. On Policy vs. Off-Policy vs. Batch-Policy Learning

This lecture should cover and contrast the different learning setups

- On-Policy Learning, i.e., policy gradient methods and the benefits and pitfalls of learning from the most up to date policy
- Almost-On-Policy Methods, i.e., off policy correction algorithms such as Retrace-lambda or the V-trace algorithmus in IMPALA, and the benefits thereof
- Off-Policy Learning, such as Q-learning
- Batch-Policy learning, where no interactions with the environment are allowed at training time. Show benefits and limitations of the current approaches.

The lecture should also cover the deadly triad, its implications and modern work-arounds

10.03. DRL in continuous action space

This lecture should highlight the difference between discrete action space environments (such as Atari, DMlab and board games) and continuous action space environments (such as MuJoCo robotic control environments). It should introduce the basic off-poliy algorithms (DDPG, TD3) as well as on-policy algorithms (TRPO, PPO, including the idea of trust regions) for this setup.
It should further discuss why the entropy regularization as found in A3C does not work in continuous action space (standard deviations of Gaussian explode) and consequently introduce the maximum entropy reinforcement learning setup (SAC).
If time, you might also touch on distributional policy optimization

17.03. Hierarchical DRL

This lecture should cover the idea of abstraction over time. That is, as humans we have skills/options - a prolonged set of atomic actions - that we use over and over again. Can a learning algorithm find/use such a temporal hierarchy? Papers in this direction include:

- The option-critic architecture
- Data-Efficient Hierarchical Reinforcement Learning
- FeUdal Networks for Hierarchical Reinforcement Learning
- Hierarchical Reinforcement Learning with Advantage-Based Auxiliary Rewards
- DAC: The Double Actor-Critic Architecture for Learning Options

24.03. DRL and stochastic planning in games

This lecture should cover two areas: Planning given a perfect model, foremost Monte Carlo Tree Search, and Self-Play, where agents learn by playing against themselves - often approximated by fictitious play. Papers for these subjects are:

- A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play
- Deep Reinforcement Learning from Self-Play in Imperfect-Information Games
- A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning
- Emergent Tool Use from Multi-Agent Interaction

An interesting case study might also be the AlphaStar algorithm.

31.03. Model based vs. model free DRL

While most original DRL algorithms (DQN, A3C, PPO) are model free, this lecture should introduce the idea of explicitly learning a model of the environment for planning (either Model Predictive Control or Monte Carlo Tree Search[1], see e.g., Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model) or learning in a "dream" (World Models, Model Based Reinforcement Learning for Atari). The lecture should also introduce successor features, a middle ground between model based and model free DRL. A paper that might also be interesting is Algorithmic Framework for Model-based Deep Reinforcement Learning with Theoretical Guarantees.

[1] MCTS should be introduced in the lecture on games the week before

07.04. DRL in partial observability

This lecture should introduce partially observable markov decision processes (POMDPs) and the difficulties that arise when the state is not fully given to the agent. It should include an example of why this setup requires stochasticity (stochasticity mitigates the worst case when acting under uncertainty). It should also cover the prominent idea to deal with POMDPs: explicitly learning a believe state, which in itself gives the necessity for a memory of some sort (e.g. an RNN) and a learned state transition model (explain why). New papers in this area include Deep Variational Reinforcement Learning for POMDPs, Variational Recurrent Models for Solving Partially Observable Control Tasks and Discriminative Particle Filter Reinforcement Learning for Complex Partial observations

21.04. Multi-Armed Bandits

Multi Armed Bandits is a well established field with many algorithms providing different sorts of guarantees. Why bother talking about it in a Reinforcement Learning Seminar? A reinforcement learning setup with an episode length of 1 is essentially a (contextual) multi-armed bandit problem. Further, many RL problems - especially all problems where a reward is only given at the end of the episode - can be framed as a multi-armed bandit (MAB) problem (with the action sequence becoming one action in the MAB). The advantage of framing as MAB problem is that there exist optimality guarantees on the exploration-exploitation trade-off, the disadvantage is that action spaces can explode quite rapidly. The lecture should introduce the difference between context free and context based MABs, different exploration strategies - epsilon greedy, softmax exploration, upper confidence bound exploration, Thompson sampling - as well as their regret guarantees.

28.04. DRL as tool for non-differentiable optimization

A neat feature of RL is that the algorithms are designed to optimize a score (maximize the reward), which leaves a lot of room to the engineer to what the reward is. More specifically, if one wishes to minimize a non-differentiable function f(x), one can train a DRL algorithm to alter x with a reward of -f(x). As such, RL opens the possibility to use deep learning in many new applications, including:

- Generation of Discrete Token Sequences - see SeqGAN
- Neural Architecture Search
- Graph Problems - see Attention, Learn to Solve Routing Problems!

The lecture should also contrast RL to the other big family for non-differentiable optimization - evolutionary algorithms. See also: Evolution Strategies as a Scalable Alternative to Reinforcement Learning and Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning

05.05. Meta-Learning

The lecture on Meta-Learning should cover the learning to learn paradigm - learn on many tasks in order to learn faster on a new task. Papers include:

- RL2: Fast Reinforcement Learning via Slow Reinforcement Learning
- Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks
- Learning to Balance: Bayesian Meta-Learning for Imbalanced and Out-of-distribution Tasks
- Meta-Learning without Memorization

A good reference is also the ICML tutorial on meta learning.

19.05. Continual Learning

This lecture should cover two areas, on the one hand the continual learning paradigm - explaining catastrophic forgetting, transfer learning and knowledge distillation - and on the other hand curriculum learning - how to automatically generate curricula of tasks such that the agent can continue to improve its performance. Papers include:

- Reinforced Continual Learning
- Experience Replay for Continual Learning
- Uncertainty-guided Continual Learning with Bayesian Neural Networks
- Intrinsic Motivation and Automatic Curricula via Asymmetric Self-Play
- Reverse Curriculum Generation for Reinforcement Learning
- Mix & Match Agent Curricula for Reinforcement Learning

26.05. Review Seminar

In the last seminar, we will summarize the different insights we have seen, connecting them back to the overview of the research field from the first seminar. We will discuss unsolved problems and potential future work.
Also, we will discuss your results on the coding challenge.