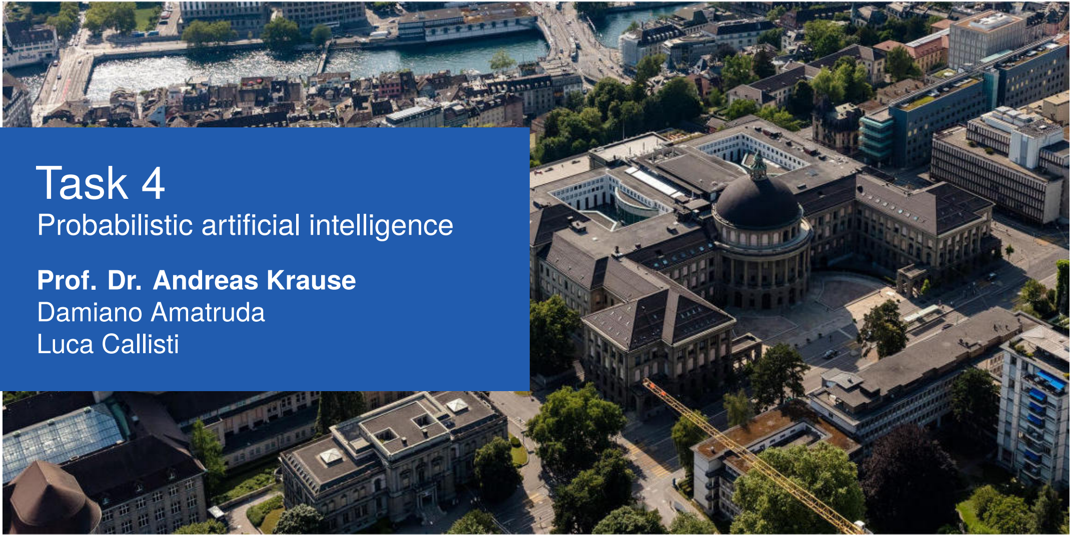
ETH zürich



Task 4

Probabilistic artificial intelligence

Prof. Dr. Andreas Krause
Damiano Amatruda
Luca Callisti

Soft Actor-Critic

We solve this task using Soft Actor-Critic, getting public score 0.0 and private score 0.0.

Soft Actor-Critic is an off-policy method that uses entropy regularization for exploration.

We use three kinds of neural networks:

1. *Actor*: Approximates the policy π_θ with a distribution over the action space;
2. *Critic*: Represents two Q-functions Q_{ϕ_1}, Q_{ϕ_2}.
3. *Critic Target*: Represents two target Q-functions $Q_{\phi_1^{\text{target}}}, Q_{\phi_2^{\text{target}}}$, used for bootstrapping.

The agent gets a reward at each time step proportional to the entropy of the policy at that time step:

$$\pi^* = \arg\max_{\pi} \mathop{\mathbb{E}}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} \gamma^t \Big(R(x_t, a_t, x_{t+1}) + \alpha H(\pi(\cdot|x_t)) \Big) \right],$$

where $\alpha > 0$ is the temperature, a coefficient for exploration-exploitation trade-off.

Critic Learning

Let us consider the Bellman equation for the entropy-regularized Q-function:

$$\begin{aligned} Q^\pi(x,a) &= \mathop{\mathbb{E}}_{\substack{x'\sim p(\cdot|x,a)\\ a'\sim\pi(\cdot|x')}} \left[R(x,a,x') + \gamma \left(Q^\pi(x',a') - \alpha \log \pi(a'|x') \right) \right] \\ &\approx r + \gamma \left(Q^\pi(x',\tilde{a}') - \alpha \log \pi(\tilde{a}'|x') \right), \quad \tilde{a}' \sim \pi(\cdot|x'). \end{aligned}$$

We train the critic using the loss:

$$L_Q(\phi_1,\phi_2) = \left(Q_{\phi_1}(x',\tilde{a}'_\theta) - y(x',\tilde{a}'_\theta) \right)^2 + \left(Q_{\phi_2}(x',\tilde{a}'_\theta) - y(x',\tilde{a}'_\theta) \right)^2,$$

$$y(x',\tilde{a}'_\theta) = r + \gamma \left(\min_{i\in\{1,2\}} Q_{\phi_i^{\text{target}}}(x',\tilde{a}'_\theta) - \alpha \log \pi_\theta(\tilde{a}'_\theta|x') \right), \quad \tilde{a}'_\theta \sim \pi_\theta(\cdot|x'),$$

where we take the minimum over the two target approximations in order to reduce overestimation bias.

Then, we update the critic target using a bootstrapping estimate in order to reduce variance:

$$\phi_i^{\text{target}} \leftarrow (1-\tau)\phi_i^{\text{target}} + \tau\phi_i.$$

Actor Learning

To sample the action, we use a Gaussian with mean and variance from the policy neural network:

$$\tilde{a}_\theta(x, \xi) = \tanh(u) = \tanh(\mu_\theta(x) + \sigma_\theta(x) \odot \xi), \quad \xi \sim \mathcal{N}(0, I).$$

We optimize the policy using the reparameterization trick:

$$\theta = \arg\max_\theta \left(\min_{i \in \{1,2\}} Q_{\phi_i^{\text{target}}}\big(x, \tilde{a}_\theta(x, \xi)\big) - \alpha \log \pi_\theta(\tilde{a}_\theta(x, \xi)|x) \right)$$

Finally, we update the temperature α according to the loss:

$$L_\alpha(x) = \alpha(-\log \pi_\theta(\tilde{a}_\theta|x) - H_{\text{target}}).$$

To avoid numerical instabilities in the calculation of $\log \pi(a|x)$, we use the formula:

$$\begin{aligned}
\log \pi_\theta(a|x) &= \sum_i \log p(u_i|x) - \sum_i \log\left(1 - \tanh^2(u_i)\right) \\
&\approx \sum_i \log p(u_i|x) - \sum_i 2\left(\log 2 - u_i - \log\left(1 + e^{-2u_i}\right)\right).
\end{aligned}$$

Bibliography

[1] Tuomas Haarnoja et al. *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*. 2018. arXiv: 1801.01290 [cs.LG]. URL: https://arxiv.org/abs/1801.01290.

[2] Tuomas Haarnoja et al. *Soft Actor-Critic Algorithms and Applications*. 2019. arXiv: 1812.05905 [cs.LG]. URL: https://arxiv.org/abs/1812.05905.

[3] OpenAI. *Soft Actor-Critic (SAC)*. n.d. URL: https://spinningup.openai.com/en/latest/algorithms/sac.html.

[4] OpenAI. *Soft Actor-Critic (SAC) - Core.py*. GitHub Repository. 2024. URL: https://github.com/openai/spinningup/blob/master/spinup/algos/pytorch/sac/core.py.