Ludwig-Maximilians-Universität München
Institut für Informatik
Lehrstuhl für Mobile und Verteilte Systeme
Prof. Dr. Claudia Linnhoff-Popien

mobile and distributed systems group

Praktikum Autonome Systeme
Wintersemester 2020/21
Übungsblatt 5 – Policy-based Deep Reinforcement Learning

Ziel der heutigen Praxisveranstaltung ist das Tuning und die Modifikation von A2C (synchroner Actor-Critic) zur Lösung von einfachen OpenAI Gym Umgebungen. Für dieses Übungsblatt wird zusätzlich das aktuelle `torch`-Package benötigt (für die Installation siehe: https://pytorch.org/get-started/locally/).

Aufgabe 1: A2C Hyperparameter Tuning

Laden Sie für dieses Übungsblatt das ZIP-Archiv `autonome-systeme-uebung5.zip` runter. In diesem Archiv finden Sie zusätzlich die Datei `a2c.py` und eine für OpenAI Gym angepasste `main.py`.

In der Datei `a2c.py` finden Sie eine vollständige Implementierung des A2C Algorithmus in der Klasse `A2CLearner`. Die A2C hat in dieser Implementierung zwei *Output Heads* (siehe Architektur in Abbildung 1). Machen Sie sich mit der Klasse vertraut: an welchen Stellen finden Sie den *Actor* und den *Critic*?

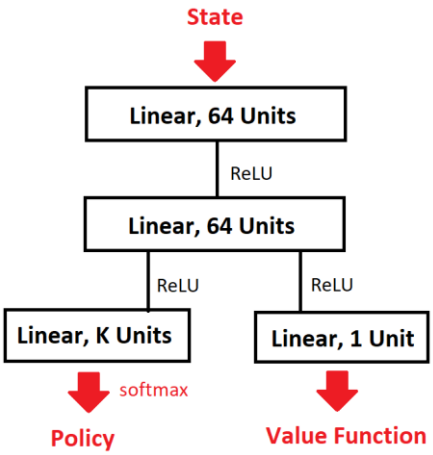

Abbildung 1: Netzwerkarchitektur des A2C.

Durch das Deep Learning kommen neue Hyperparameter hinzu (siehe `main.py`). Finden Sie ein geeignetes Hyperparameter-Setting, mit denen der `A2CLearner` die OpenAI Gym Domänen `CartPole-v1`, `Acrobot-v1` und `MountainCar-v0` löst (**Vorsicht:** Diese Aufgabe erfordert viel Zeit und Resourcen. Nutzen Sie dazu ggf. die Slurm-Engine im CIP-Pool).

Zusatzaufgabe: Implementieren Sie ein Verfahren (z.B. Random Search, Evolutionary Optimization), das automatisch nach einem geeigneten Hyperparameter-Setting sucht.

Aufgabe 2: A2C Modifications

Testen Sie die folgenden Änderungen gegen die Original-Version des A2C (s. Aufgabe 1) in den OpenAI Gym Domänen `CartPole-v1`, `Acrobot-v1` und `MountainCar-v0`.

1. REINFORCE

Ändern Sie den `advantage` aus Zeile 88 in `a2c.py`, indem Sie den Critic weglassen bzw. auf 0 setzen.

2. Temporal-Difference Actor-Critic

Ändern Sie den `advantage` aus Zeile 95 in `a2c.py` von $A(s_t, a_t) = \sum_{k=0}^{T} r_{t+k} - \hat{V}_\theta(s_t)$ zu $A(s_t, a_t) = r_t + \hat{V}_\theta(s_{t+1}) - \hat{V}_\theta(s_t)$.

3. Separate Actor and Critic Networks

Trainieren Sie zwei getrennte neuronale Netze für den Actor und den Critic.