
Wintersemester 2019/2020

Praktikum Selbstlernende Systeme

Aufgabenblatt 3

Schicken Sie Ihre Lösung in der Form die in der ersten Veranstaltung festgelegt wurde bis zum **Sonntag, den 15.12.2019 um 24:00 Uhr** an obenstehende E-Mail-Adresse.

WICHTIG: Verwenden Sie für **ALLE** Aufgaben eine `screen_size` von 16

1 DDQN (1 Punkte)

1. Trainieren Sie Ihren DQN-Agenten mit einem Double Deep Q-Network.
2. Visualisieren Sie den Lernfortschritt wie gehabt.
3. Der trainierte DDQN-Agent soll durch ein Python File (RunDDQN.py) gestartet werden können. Der Agent soll die trainierte Gewichte einlesen und diese benutzen ohne weiter zu lernen.
4. Das Training des DDQN-Agenten soll durch ein Python File (TrainDDQN.py) gestartet werden können. Der Agent soll hierbei ein neues Netz initialisieren.

2 Dueling DQN (4 Punkte)

1. Trainieren Sie Ihren DQN-Agenten mit einem Dueling Deep Q-Network.
2. Verwenden Sie hierfür einen Lambda Layer.
3. Visualisieren Sie den Lernfortschritt wie gehabt.
4. Der trainierte DDQN-Agent soll durch ein Python File (RunDuelDQN.py) gestartet werden können. Der Agent soll die trainierte Gewichte einlesen und diese benutzen ohne weiter zu lernen.
5. Das Training des DDQN-Agenten soll durch ein Python File (TrainDuelDQN.py) gestartet werden können. Der Agent soll hierbei ein neues Netz initialisieren.

3 CNN Dueling DDQN mit Prioritized experience Replay (5 Punkte)

1. Erweitern Sie Ihren Agenten so, dass er ein Convolutional Neural Network mit folgenden Layern benutzt:
 - a) 1 Input Layer
 - b) 1 Hidden Convolution Layer mit 16 Filtern der Größe 5x5 und Stride 1 + ReLu
 - c) 1 Hidden Convolution Layer mit 32 Filtern der Größe 3x3 und Stride 1 + ReLu
 - d) 1 hidden Layer (64 Knoten + ReLu)
 - e) 1 Output Layer (8 Knoten + Linear)
2. Als Input verwenden Sie auf jeden Fall den *unity_density*-Layer
 - Sie finden alle Layer die von pysc2 bereitgestellt werden hier:
`timesteps[0].observation.feature_screen`
 - Achten Sie darauf, dass Ihr Input das richtige Format hat: `shape=(x, y, channels)`
3. Der Agent soll Double Q-Learning benutzen.
4. Erweitern Sie das Netz zu einem Dueling DQN.
5. Verwenden Sie ein Prioritized Experience Replay.
 - a) Hierfür dürfen Sie das bereits bestehende PER von baselines verwenden
<https://github.com/openai/baselines>
 - b) Um baselines auf Windows zu Installieren, folgen Sie dieser Anleitung:
<https://arztsamuel.github.io/en/blogs/2018/Gym-and-Baselines-on-Windows.html>
 - c) Verwenden Sie folgende Werte:
 - $\alpha = 0,4$
 - $\beta = 0,6$
 - $\beta_{inc} = 5e - 6$
 - $\epsilon_{replay} = 1e - 6$
6. Visualisieren Sie den Lernfortschritt wie gehabt.
7. Der trainierte CNN-Agent soll durch ein Python File (RunCNN.py) gestartet werden können. Der Agent soll die trainierte Gewichte einlesen und diese benutzen ohne weiter zu lernen.
8. Das Training des CNN-Agenten soll durch ein Python File (TrainCNN.py) gestartet werden können. Der Agent soll hierbei ein neues Netz initialisieren.

Viel Erfolg bei der Bearbeitung!