

Praktikum: Selbstlernende Systeme

Verwendung des Boilerplate-Projekts

25.10.2018

Universität Augsburg

Institut für Informatik

Lehrstuhl für Organic Computing

- Klonen sie das Repository:
`https://git.rz.uni-augsburg.de/krausjul/sls-boilerplate.git`
- Verwenden sie für die Authentifizierung ihre RZ-Kennung.
- Benennen sie das Verzeichnis von sls-boilerplate in sls-`<RZ-Kennung>` um.
- Verwenden sie dieses Verzeichnis für die Bearbeitung aller Aufgaben des Praktikums und verwalten sie es in einem eigenen Repository.

- Öffnen sie das Projekt mit der IDE ihrer Wahl (Empfehlung: PyCharm). Achten sie darauf `.../sls-<RZ-Kennung>` (nicht `.../sls-<RZ-Kennung>/sls`) als Root-Verzeichnnis zu wählen!
- Weisen sie dem Projekt das vorher erstellte Conda-Environment `pysc2` als Interpreter zu.
 - Pycharm: File -> Settings... -> Project: `sls-<RZ-Kennung>`
-> Project Interpreter -> *Zahnrad* -> Add -> Conda
Environment -> Existing Environment

Das Boilerplate implementiert Komponenten für die Interaktion mit StarCraft 2 und die Evaluierung der Lernergebnisse:

- env: Schnittstelle zur PySC2 API
- runner: Koordination des schrittweisen Ablaufs und regelmäßiger Tasks (z.B. Speichern der Ergebnisse)
- runScript: Konfiguration und Start eines durchzuführenden Laufs (Welcher Agent?, Wieviele Episoden?, ...)
- agents: Ein Agent erhält in jedem Zeitschritt den aktuellen Zustand der Umgebung (SC2) und leitet daraus eine durchzuführende Aktion ab, die er zurückgibt
Ihre Aufgabe wird es sein verschiedene Agenten zu implementieren.

- Jeder Agent den sie implementieren soll von AbstractAgent erben.
- AbstractAgent bietet einige Hilfsmethoden und globale Variablen für den Zugriff auf die relevanten SC2-Einheiten und Aktionen und definiert abstrakte Methoden die ihre Agenten implementieren müssen.
 - step: Bekommt den aktuellen Spielzustand übergeben und gibt eine auszuführende Aktion zurück.
 - save_model: Speichert das Gelernte in eine Datei. ¹
 - load_model: Läd vorher Gelerntes aus einer Datei. ¹

¹nicht nötig für den geskripteten Agenten (Aufgabe 1)

- Für die Evaluierung wird Tensorboard benutzt.
- Der während einer Episode erreichte Score dient als Performance-Maß.
- Die Ergebnisse eines Laufs werden unter `./graphs/YYYYMMDD_hhmm_type_agent` abgelegt.
- Um die Ergebnisse anzuzeigen, öffnen sie ein Terminal, aktivieren sie ihr Conda Environment und nutzen den Befehl
`$ tensorboard --logdir PATH`
Wobei PATH entsprechend zu ersetzen ist.