

Verteiltes Deep Reinforcement Learning System zum Trainieren von Game AI

- Praktikum 23.04.2021 -

[Aufgabe 1: Umwelt für Experimente auswählen](#)

[Aufgabe 2: Experimente planen und ausführen](#)

[Aufgabe 3: Ergebnisse visualisieren](#)

[Aufgabe 4: Agenten beobachten \(ggf. Video rendern\)](#)

Ganz Wichtig: Arbeitet euch durch die Dokumentation!

[ML-Agents Toolkit Release 15 Dokumentation](#)

Aufgabe 1: Umwelt für Experimente auswählen

- Selbst erstellte Umwelt vom Vortag oder eine der Beispielumwelten auswählen
- Am besten eine Umwelt mit folgenden Eigenschaften (Trainingsaufwand!):
 - Diskreter Aktionsraum
 - Kein visueller Input
 - Vorsicht! visueller Input braucht auf LiDo 3 und Colab einen X Server
 - Das betrifft Unity Umwelten
 - Keine Multi-Agent Umwelt

Aufgabe 2: Experimente planen und ausführen

- Eine Baseline trainieren (zählt nicht als Experiment)
 - ergo Ausgangslage
 - 3 Wiederholungen trainieren lassen
- Min. 4 Experimente
- Mit welcher Intention wird das Experiment durchgeführt?
 - Bitte nicht willkürlich experimentieren
- Mögliche Bereiche zum Experimentieren:

- Veränderung der Dynamiken der Umwelt?
- Hyperparameter modifizieren
- Belohnungsfunktion ändern
- Beobachtungsraum erweitern oder einschränken
- (Modellarchitektur anpassen, das ist nicht ganz so trivial bei ml-agents)
- Welche Rechenkapazitäten stehen zur Verfügung?
 - Privater Rechner?
 - Google Colab (Vorsicht vor den Timeouts -> Checkpoints!)
 - PC-Pool (nicht uneingeschränkter Zugriff, Accounts benötigt)
 - LiDo 3 (Accounts müssen von den Betreuern beantragt werden)
- Experimente ausführen
 - Nützlich: batch/shell Scripte
- Experimente wiederholen
 - min. 3 Durchläufe (für das Praktikum soll das genügen)
 - das betrifft auch die Baseline!

Aufgabe 3: Ergebnisse visualisieren

- Trainingsstatistiken plotten
 - Durchschnitt über die jeweiligen Durchläufe mit Varianz
- Plots nach Aussagekraft auswählen und in einem Dokument bündeln
 - z.B. Slideshow oder DIN A4
 - je nachdem was der Übersicht und der Vergleichbarkeit dient

Aufgabe 4: Agenten beobachten (ggf. Video rendern)

- Beschreibt oder visualisiert falls möglich das Agentenverhalten
- Video aufnehmen
 - z.B. OBS, snagit, fraps