

Organic Computing 2

Lösungsvorschlag Blatt09

Lukas Huhn Qiang Chang Victor Gerling

22. Juli 2019

Universität Augsburg

Institut für Informatik

Lehrstuhl für Organic Computing

1. Aufgabe 01

Aufgabe 01

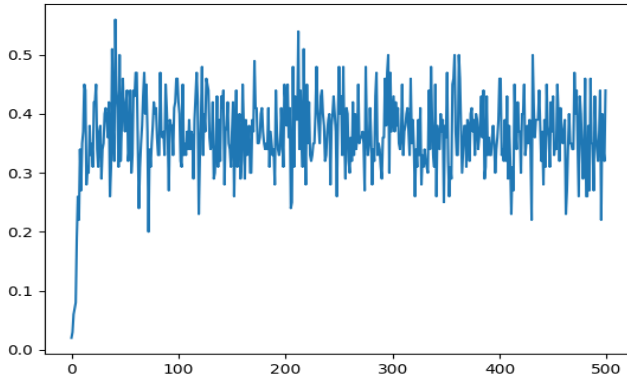
Passen Sie Ihre Q-Tabelle und die Interaktion mit ihr (Durchsuchen, Initialisieren, Updating etc.) so an, dass Intervalle statt genauer Zustandswerte sowie eine Zeile pro Aktion benutzt werden!

- States werden in Binärstrings umgewandelt, mit jeweils eigenen Min, Max und Step-values für die verschiedenen Attribute.
- Zum Durchsuchen wurde ein Python Dictionary + Regex verwendet (langsam).
- Beim Update der Tabelle wurde für den Wert in $Q(s, a)$ der Durchschnitt aller matchenden State-Action Regeln genommen.

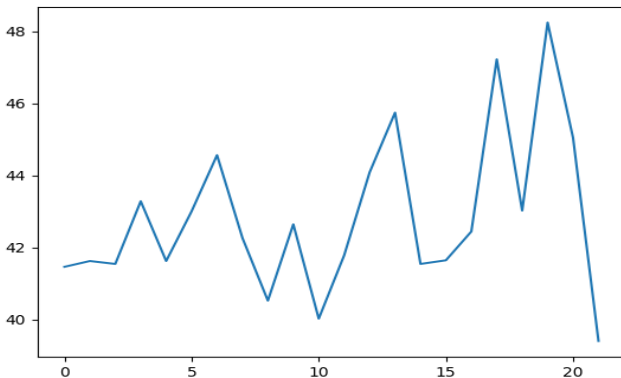
Implementieren Sie einen einfachen GA für die Zustandsbeschreibungen Ihrer Q-Tabelle und lassen Sie diesen alle paar Schritte eine Iteration durchführen! Wann läuft Ihr GA? Wie wurde er implementiert? Welches Fitnessmaß verwenden Sie?

- Nach 50 Schritten wird der GA auf dem jetzigen Action-Set ausgeführt.
- Population: 4, Selection: 2 Parents mittels Roulette-Wheel und Softmax-Distribution
- Uniform crossover
- Mutation mit 0, 1 und (mit 0.04)
- Fitnessmaß: Werte der Q-Table

Zeichnen Sie ein Episode-Returndurchschnitt-Diagramm zu FrozenLake-v0.



Zeichnen Sie ein Episode-Returndurchschnitt-Diagramm zu CartPole-v1.



Nach wie viel Episoden ist der Lernprozess zufriedenstellend abgeschlossen? Begründen Sie!

- Frozenlake: Bei ca. 1000, hier wird jedoch kaum generalisiert, da es ohnehin nur 16 Zustände gibt.
- Cartpole: Hier konnte keine zufriedenstellende Lösung gefunden werden (evtl. falsche Parameter + ECS Lookup ist zu ineffizient)