
Wintersemester 2019/2020

Praktikum Selbstlernende Systeme

Aufgabenblatt 5

Schicken Sie Ihre Lösung in der Form die in der ersten Veranstaltung festgelegt wurde bis zum **Sonntag, den 02.02.2019 um 24:00 Uhr** an obenstehende E-Mail-Adresse.

1 Gruppenfindung

1. Trage dich in eine der vier Gruppen (Protoss, Zerg, Terran oder Random) im Digicampus ein
2. Nimm Kontakt mit deinen Teamkameraden auf

2 A2C (15 Punkte)

1. Implementieren Sie einen Agenten der das Minigame MoveToBeacon mittels einer *Advanced-Actor-Critic*-Lernkomponente löst
2. Der Agent hat in jedem Schritt 8 mögliche Aktionen (Himmelsrichtungen)
3. Verwenden Sie hierfür ein neuronales Netz mit folgenden Layern:
 - a) Input
 - b) Hidden Convolution Layer (16 Filter mit Größe 5x5 und Stide=1 + ReLu)
 - c) Hidden Convolution Layer (32 Filter mit Größe 3x3 und Stide=1 + ReLu)
 - d) 1 Hidden Layer (128 Knoten + ReLu)
 - e) 1 Output Layer (8 Knoten + Softmax)
 - f) 1 Output Layer (1 knoten + Linear)

4. Verwenden Sie eine *screen_size* von 16
5. Implementieren Sie die Loss Function (und das Netz) in Tensorflow
6. Der Agent soll mit mind. 8 Workern trainieren
7. Verwenden Sie einen *n-step-Return* mit $n = 5$
8. Verwenden Sie als Value-Konstante $c_{val} = 0.5$
9. Verwenden Sie als Entropie-Konstante $c_H = 0.005$
10. Verwenden Sie als Learningrate $lr = 0.0007$
11. Der Agent bekommt in jedem Schritt folgenden Reward:
 - +1 wenn er das Ziel erreicht
 - -0.01 sonst
12. Verwenden Sie eine *minibatch_size* von 64
13. **WICHTIG:** Initialisieren Sie die jeweiligen Environments der worker im dazugehörigen Thread
14. Visualisieren Sie in einem Graphen den durchschnittlichen Reward aller Worker pro Episode
15. Der trainierte A2C-Agent soll durch ein Python File (RunA2C.py) gestartet werden können. Der Agent soll die trainierte Gewichte einlesen und diese benutzen ohne weiter zu lernen.
16. Das Training des A2C-Agenten soll durch ein Python File (TrainA2C.py) gestartet werden können. Der Agent soll hierbei ein neues Netz initialisieren.

3 A2C - FullyConv (5 Punkte)

1. Erweitern Sie ihren A2C Agenten so, dass er direkt zum Beacon läuft
2. Verwenden Sie hierfür folgenden Netz:
 - a) Input
 - b) Hidden Convolution Layer (16 Filter mit Größe 5x5 und Stide=1 + ReLu)
 - c) Hidden Convolution Layer (32 Filter mit Größe 3x3 und Stide=1 + ReLu)
 - d) Hidden Convolution Layer (1 Filet mit Größe 1x1 und Stride=1 + Softmax) - Actor Output
 - e) 1 Hidden Layer (256 Knoten + ReLu)
 - f) 1 Output Layer (1 knoten + Linear)

3. der Agent kann als mögliche Aktion jeden Pixel als Ziel wählen
4. Visualisieren Sie in einem Graphen den durchschnittlichen Reward aller Worker pro Episode
5. Der trainierte A2C-FullyConv-Agent soll durch ein Python File (RunA2C-FC.py) gestartet werden können. Der Agent soll die trainierte Gewichte einlesen und diese benutzen ohne weiter zu lernen.
6. Das Training des A2C-FullyConv-Agenten soll durch ein Python File (TrainA2C-FC.py) gestartet werden können. Der Agent soll hierbei ein neues Netz initialisieren.

Viel Erfolg bei der Bearbeitung!