



Matthias Tschöpe, Kunal Oberoi

Übung 11

Was ist ein Bot ?

- Unter einem **Bot** (von englisch *robot* ,Roboter‘) versteht man ein Computerprogramm, das weitgehend automatisch sich wiederholende Aufgaben abarbeitet, ohne dabei auf eine Interaktion mit einem menschlichen Benutzer angewiesen zu sein.

Weight Matrix and Static weights Heuristic Function

- 1. "Static weights
Heuristic Function"
- 2. "Board-Weight
Matrix"

| | | | | | |
|----|----|----|----|----|----|
| -3 | 2 | 2 | 2 | 2 | -3 |
| -4 | -1 | -1 | -1 | -1 | -4 |
| -1 | 1 | 0 | 0 | 1 | -1 |
| -1 | 0 | 1 | 1 | 0 | -1 |
| -1 | 0 | 1 | 1 | 0 | -1 |
| -1 | 1 | 0 | 0 | 1 | -1 |
| -4 | -1 | -1 | -1 | -1 | -4 |
| -3 | 2 | 2 | 2 | 2 | -3 |

```
private int boardWeight[][] = {  
    { 100, -10, 11, 6, 6, 11, -10, 100  
    { -10, -20, 1, 2, 2, 1, -20, -10  
    { 10, 1, 5, 4, 4, 5, 1, 10  
    { 6, 2, 4, 2, 2, 4, 2, 6  
    { 6, 2, 4, 2, 2, 4, 2, 6  
    { 10, 1, 5, 4, 4, 5, 1, 10  
    { -10, -20, 1, 2, 2, 1, -20, -10  
    { 100, -10, 11, 6, 6, 11, -10, 100  
};
```



Arten von Bots

- Random Bot
- Greedy Bot

Random Bot

- Beim Random Bot spielen zwei Spieler (Computer) zufällig gegeneinander

Übungsaufgabe Random Bot

- Implementieren sie eine Klasse Random Bot der form
 - **class random_bot:**
 - **def __init__(self, spieler_farbe):**
 - Legen Sie hierbei die spielerfarbe , pos_felder und current_choice an
 - **def set_next_stone(self):**
 - Hinweis: Damit erhält man aus den möglichen Felder Matrix alle Indizes als Liste
 - `pos_zeilen = [(idr, idz) for idr, row in enumerate(self.pos_felder) for idz, zelle in enumerate(row) if zelle == "pos_f"]`

Greedy Bot

- 1. naiver Greedy Bot (maximiert Anzahl der eigenen Steine)
- 2. weighted Greedy Bot (maximiert Anzahl der eigenen Steine bzgl gegebener Gewichtsmatrix)

Übungsaufgabe Greedy Bot

- **class greddy_bot:**
 - **def __init__(self,spieler_farbe):**
 - Legen Sie hierbei die spielerfarbe
 - **def set_next_stone(self,spielfeld,pos_felder):**
 - Hinweis: Damit erhält man aus den möglichen Felder Matrix alle Indizes als Liste
 - `pos_zeilen = [(idr, idz) for idr, row in enumerate(self.pos_felder) for idz, zelle in enumerate(row) if zelle == "pos_f"]`
 - Die Methode `check_rules`, könnte durch aus hilfreich sein bei lösen dieser Aufgabe

Bot mit Neuronalen Netze

- Fragerunde

- 1. Spielfeld flatten, wie groß ist der Input Vektor?
- 2. Was wäre Sinnvoller Output?
- 3. Architektur festlegen (berücksichtigen der Aktivierungsfunktion)
- 4. Für welche Klassifizierer geeignet?
- 5. Wie werden die Klassifizierer verwendet?
- 6. Nachteile ?
- 7. Woher bekommen wir Ground Truth?
- 8. Möglichkeiten für einen "perfekten" Klassifizierer?

■ Othello with Neural Network (Advanced)

- <https://arxiv.org/pdf/1711.06583.pdf>
- <https://www.ai.rug.nl/~mwiering/GROUP/ARTICLES/paper-othello.pdf>