

Projektbericht: Maschinelles Lernen am Spiel „Vier Gewinnt“

Business Analytics Project von Jonathan Cawalla, Lena Gräwe, Ahmad
Haschemi und Lena Knickmeier

1. Einleitung

1.1) Aufgabenstellung **ahmad**

- Angelehnt an die von Google implementierte KI für AlphaGo eine KI implementieren
- Anwendung von Maschinellern Lernen

2. Umsetzung der Trainingsumgebung

2.1) Vier Gewinnt **Jon**

- Suche nach fertigen Implementationen und Anpassung
- Viele der gefundenen Implementationen waren unbrauchbar, da sie nicht in der Größe anpassbar waren.
- Viel Zeit in eine Implementation investiert, die sich zwar in der Größe anpassen ließ, jedoch war es uns nicht möglich, eine zweite KI korrekt an das Spiel anzubinden.
- Eigene Implementierung des Spiels, um nicht noch mehr Zeit zu verlieren
- Turnier-Modus: Zur Auswertung der KI(s) gedacht. KI lernt in diesem Modus nicht weiter, sondern zeigt nur wie gut diese mit dem bisherigen Training spielen kann.

2.2) Nicht-lernende KI für das Training **Lena K**

- Um eine Basis zu haben, mit der die KI(s) lernen können, haben wir eine KI geschrieben, die allein mit if-Abfragen und Schleifen arbeitet.
- Den Algorithmus haben wir aus einer anderen Implementation übernommen.

2.3) Datenbank und Binär-Codierung **Jon**

- Datenbank, die intern mit zwei ineinander verschachtelten HashMaps implementiert ist. Dabei enthält die erste HashMap, den aktuellen State (Spielfeld) und die darin enthaltene HashMap, die Actions (Züge, also Spalten in die geworfen wird) enthält und das Value (Bewertung) für diese Action.
- Methoden zum Einfügen, Updaten sowie Speichern und Laden der Datenbank.
- Hash-Codierung Veränderung
- Speichern und Laden → Lernen unterbrechen /fortsetzen
- Perfektes Hashing ^^
- BC: Zum Speichern der Spielzustände sind zwei Methoden implementiert, die zwei-dimensionale Arrays in eine Dimension umwandeln und umgekehrt, ohne dass dabei Informationen verloren gehen.

3. Q-Learning

3.1) Erläuterung der Methode, Literaturrecherche **Lena K**

- KI erhält für Spielzüge positive oder negative Rewards und sammelt dadurch künstlich Erfahrungen.
- Rewards werden anhand des Spielausgangs vergeben.

3.2) Verwandte arbeiten **lena**

3.3) Umsetzung **Jon**

- KI geniert mit einem Zuggenerator den besten möglichen Zug basierend auf einer Datenbank, die Belohnungen oder Bestrafungen für die KI speichert, je nachdem, ob sie gewonnen oder verloren hat.
- KI betrachtet bei jedem Zug, die darauffolgenden nächsten Züge und den maximal möglichen Wert, den sie damit erreichen kann und wertet dementsprechend den aktuellen Zug mit einem gewissen Lernparameter, der bestimmt wie wichtig langfristig mögliche Züge sind.
- Inverse Gegner Zug
- ϵ - Greedy

4. Neuronales Netz

- 4.1) Erläuterung der Methode, Literatur Recherche **ahmad**
- Die nichtlineare Diskriminanzfunktion wird durch Werte der Knoten (Neuronen) und das Gewicht der verbindenden Kanten (Synapsen) berechnet. Durch Training wird dieses Gewicht aktualisiert, um die beste Strategie herauszufinden.
 - Neuronale Netze gilt als eine klassische Methode des Maschinellen Lernens, und kann zum Vergleich mit RL geeignet sein.
- 4.2) Umsetzung **lena** G

5. Test und Auswertung

- 5.1) Q-Learning gegen NormalKI **Dienstag**
- 4x5 und 6x7
 - Anfangen/ abwechselnd
 - ϵ - Greedy und inverse betrachtung des Gegners → Verbesserung?
 - Ausgehend von bestimmten spielzuständen
 - Zählen wie oft er nicht weiß was zu tun ist (zustand nicht in Datenbank)
- 5.2) Neuronale KI gegen NormalKI
- 4x5 **lena g**
 - 6x7 **jon**
 - Spielzüge analysieren **lena g**
- 5.3) Q-Learning gegen Neuronales Netz **lena k**

6. Ausblick

- 6.1) NN **lena g**
- 6.2) Q
- Weiter trainieren **lena k**
 - Datenbank komprimieren **jon**
 - Durch schnitt gegen maimum tauschen **jon/lena k**