**Ausblick**

Die Ergebnisse dieses Business Analytics Projekts zeigen, dass es mit der Methode des Q-Learnings und auch mit der Methode von Neuronalen Netzen möglich ist, eine selbst lernende Künstliche Intelligenz für das Spiel „Vier-Gewinnt“ zu implementieren, aber es gibt noch weitere Möglichkeiten dieses Projekt in Zukunft fortzuführen bzw. zu erweitern, diese werden im Folgenden dargestellt.

**Q-Player**

Der Q-Player erzielt auf einem auf einem 4x5 Feld hervorragende Ergebnisse. Die einzige Frage die bleibt ist, warum er ab einem gewissen Grad wieder schlechter wird. Hierzu sollte die Berechnung der Werte für die Zustände neu betrachtet werden. Evtl. genüg es schon auf die Durchschnittsberechnung zu verzichten und nur den maximalen Wert der nächsten Zustände zu betrachten, nachdem für den Gegner vom für ihn bestmöglichen Zug ausgegangen wird. Ein anderer Ansatz ist es nicht mehr davon auszugehen, dass der Gegner den bestmöglichen Zug macht, sondern an dieser Stelle einen Mittelwert bildet. Beide Alternativen müssten experimentell erforscht werden.

Auch auf dem 6x7 kann der Q-Player von einer Verbesserung an dieser Stelle profitieren. Zunächst muss jedoch sichergestellt werden, dass er überhaupt alle für ihn wichtigen Spielzustände erforschen kann. Da die gewählte Implementierung der Datenbank hier an ihre Grenzen stößt, liegt hier Verbesserungspotenzial um den Q-Player auch auf einem 6x7 Feld zu trainieren. Die Datenbank wird sehr groß und ab einer gewissen Größe reicht der Arbeitsspeicher nicht mehr aus. Dann wird die Datenbank zu Teilen auf die Festplatte geschrieben und das Abfragen von Spielzuständen dauert sehr lange. Ein Ansatz zur Komprimierung der Datenbank scheint lohnenswert. Der Schlüssel der HashMap, die intern verwendet wird, ist ein 2-dimensionales Int-Array. Dies ist vom Speicherverbraucht nicht sehr effizient, da die Anzahl der Zustände in der Datenbank sehr schnell sehr groß wird. Bei 3.200.00 Trainings gibt es 21.055.352 Einträge in der Datenbank. Einen kleineren Schlüssel zu verwenden könnte den Speicherverbrauch reduzieren.

Allerdings kann es sein, dass auch so nicht alle Zustände in einer händelbaren Datenbank untergebracht werden können. Wie oben erwähnt beschäftigt sich Olszowka in seiner Arbeit mit der Problematik rund um die Anzahl der Zustände bei einem 6x7 Spielfeld. Er geht sogar davon aus, dass es nicht einmal möglich ist, die KI alle Zustände erforschen zu lassen. Diese Aussage basiert jedoch auf der Annahme, dass es etwa Zustände gibt. [vgl. ] Tatsächlich sollte die Anzahl bedeutend kleiner sein, da die Steine nicht beliebig auf dem Feld verteilt werden dürfen. Zu beachten sind u.a. folgende Punkte:

* Die Spieler dürfen nur abwechselnd werfen, d.h. die Anzahl der Steine von Spieler eins und zwei unterscheiden sich maximal um eins.
* Steine könne nicht „schweben“. Sie fallen immer so weit nach unten wie es ihnen möglich ist.
* Wenn vier Steine einer Farbe neben einander liegen ist das Spiel beendet, d.h. es werden keine weiteren Steine mehr platziert.

Trotzdem wird es sich lohnt, sich mit Optionen zur Zustandsreduzierung zu befassen. Olszowka schlägt die Verwendung Feature-Vektoren vor. Dabei wird nicht mehr jeder Spielzustand einzeln gespeichert. Stattdessen erhält der der Q-Player einen Vektor mit Eigenschaften die im jeweiligen Zustand enthalten sind. Wichtig ist aber, dass nicht zu viele Informationen verloren gehen. [vgl. ]

**NNPlayer**

Den *NNPlayer2* betreffend, wäre es von Bedeutung, sich die Datasets genauer anzuschauen. Zurzeit werden sie durch zufällige Spielzüge der *NormalKI* generiert und man hat bisher keinen Einfluss darauf, bestimmte Spielzüge auszusortieren, d. h. es ist möglich, dass auch Spielzüge gespeichert werden, die zum Verlieren führen, wie in Kapiel 5.2. vermutet wurde. Von daher wäre es sinnvoll, nur die Spielzüge zu speichern, die zum Sieg oder zu einem Unentschieden führen können. Außerdem könnte die *turn*-Methode im *NNPlayer2* umgestaltet werden und mit zwei Neuronalen Netzen gearbeitet werden. Eines für die Spiele des 1. Spielers und eines für Spiele des 2. Spielers und je nach Situation im Turnier, sollte dann das passende Netz ausgewählt werden, um so die Gewinnchancen zu erhöhen bzw. auszutesten.