

Erlernen der Spielweise des Spiels „Connect6“ anhand dessen Spielregeln

Studiengang: MAS | Vertiefung: MAS Data Science

In dieser Masterarbeit wurde ein neuronales Netzwerk, der Q-Learning Algorithmus und das Spiel „Connect6“ in Java implementiert. Mittels dieser Implementation wurde versucht, die Spielweise des Spiels ausschliesslich mit den Spielregeln zu erlernen.

Problemstellung

In der heutigen Zeit existieren für diverse Programmiersprachen Bibliotheken, welche die Ansätze im Bereich des maschinellen Lernens zur Verfügung stellen. Anhand dieser Bibliotheken können Problemstellungen effizient und ohne grössere Programmierung behandelt werden. Jedoch besteht zugleich das Risiko, dass die Anwender den verwendeten Ansatz nicht detailliert verstehen. Aus diesem Grund besteht das Ziel dieser Masterarbeit darin, einen vollständigen Ansatz aus dem Bereich des maschinellen Lernens zu implementieren und eine entsprechende Problemstellung zu behandeln.

Vorgehen

Da keine vordefinierten Daten zur Verfügung stehen, wird ein „reinforcement learning“ Ansatz implementiert. Bei diesen Ansätzen besteht die Möglichkeit, die benötigten Daten während dem Lernprozess zu erstellen und zugleich zu verwenden. Der implementierte Ansatz besteht aus dem Zusammenspiel zwischen einem Mehrschichten-Perzeptron mit Backpropagation und dem Q-Learning Algorithmus. Die dazugehörige Problemstellung besteht darin, die Spielweise des Spieles „Connect6“ ausschliesslich anhand den Spielregeln zu erlernen. Beim Spiel „Connect6“ handelt es sich um ein Zweipersonen-Strategiespiel mit dem Ziel, sechs identische Spielsteine in eine Reihe zu bringen. Im Vergleich zu anderen ähnlichen Spielen, können die Spieler zwei Spielzüge hintereinander tätigen, ausser beim ersten Spielzug. Die vollständige Implementation wird vollständig mittels der Programmiersprache Java umgesetzt.

Nach der Implementation werden unterschiedliche Modelle trainiert, eine Trainingseinheit besteht in der Regel aus 10'000 Spielen. Bei den einzelnen Versuchen werden unterschiedliche Parameterwerte für die Lernrate, den Discount-Faktor, der Batch-Grösse und der Anzahl Epochen verwendet. Nach jedem Versuch werden die Spielresultate miteinander verglichen und

wenn nötig die Implementation verbessert. Nach den ersten drei Versuchen konnte festgestellt werden, dass kein Modell die Spiellogik erlernt hat. Aus diesem Grund wurde für die restlichen Versuchen die Spielbrettgrösse reduziert. Durch die Reduktion wird die Komplexität des Spiels verringert. Unabhängig von den einzelnen Verbesserungen oder den Variationen der einzelnen Parameterwerten konnte bei den restlichen Versuchen nicht festgestellt werden, dass ein Modell die Spiellogik erlernt hat. Aus diesem Grund wurde beim letzten Versuch die Anzahl der Spiele auf 200'000 erhöht. Auch durch diese Erhöhung erlernte das Modell die Spiellogik nicht.

Fazit

Während der Implementationsphase und Simulationsphase konnte mein Wissen bezüglich der Programmierung und den eingesetzten Ansätzen vertieft werden. Diese Vertiefung entstand durch Recherchen oder durch Fehler, die während der Implementation entstanden. Des Weiteren kam ich zum Entschluss, dass bei der Implementation eines solchen Ansatzes viele unterschiedliche Aspekte berücksichtigt werden müssen, die in der Regel in der Theorie nicht erwähnt werden. Da jedoch die Problemstellung nicht gelöst werden konnte, ist das Ziel dieser Masterarbeit nicht erfüllt worden. Für die Nicht-Erfüllung existieren unterschiedliche Gründe, ein wesentlicher Grund ist, dass zu wenige Simulationen durchgeführt werden konnten. Durch die vorhandene Komplexität des Spieles müssten einige Millionen Spiele simuliert werden, um die Spiellogik zu erlernen.



Lukas Ferilli