# Machine Learning in Computerspielen

Studiengang: BSc in Informatik | Vertiefung: Web and Business Applications
Betreuer: Dr. Jürgen Eckerle
Experte: Dr. Federico Flueckiger



Google DeepMind entwickelte das Programm Alpha Go, welches den weltbesten Profispieler Lee Sedol im Spiel «Go» schlagen konnte. Das Programm verwendete dazu die Lernmethode Reinforcement Learning in Verbindung mit künstlichen Neuronalen Netzen, bei dem ein Agent mit der Umwelt interagiert und dabei belohnt oder bestraft wird. Diese Lernmethode wird in folgender Arbeit angewendet und ist deshalb so interessant, da sie Analogien zum menschlichen Lernverhalten aufweist.

## Ausgangslage

Mari Flow ist ein von Seth Bling programmierter Machine Learning Agent, der das Spiel «Super Mario Kart» mittels eines rekurrenten Neuronalen Netzes spielt. Der Input des Netzes beinhaltet nur das verpixelte Bild des Spiels. Als entsprechender Output liefert das Netz schliesslich die möglichen Steuerungstasten. Durch Aufnahme des Bildausschnittes sowie der dabei gedrückten Tasten, wird das Netz durch Supervised Learning mit dem Mensch als Supervisor trainiert.
Das Problem bei diesem Ansatz ist, dass der Agent nicht vorrangig darauf abzielt, zu gewinnen oder möglichst schnell zu fahren, sondern lediglich die Aktionen zu imitieren versucht, durch die er trainiert wurde. Die Fähigkeiten des Agenten werden aus diesem Grunde nie diejenigen des Menschen übertreffen.

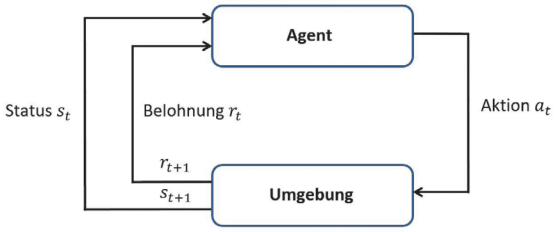
## Ziel

Ziel der Bachelor Thesis war die Entwicklung eines KI Agenten, welcher das Spiel «Super Mario Kart» selbstständig erlernen und spielen kann. Dafür wird der Agent durch Reinforcement Learning trainiert. Der Agent soll ohne jegliche Vorkenntnisse durch Trial and Error selbst lernen, den besten und schnellsten Weg zu finden.

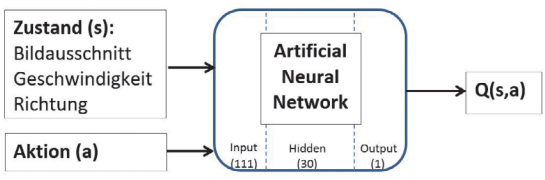
## Ergebnis

Das Ergebnis ist ein funktionierender Prototyp, welcher das Spiel «Vektor Racer» meistert. Das Spiel «Vektor Racer» wurde aus dem Grund als Simulationsumgebung gewählt, da es ähnliche Input-Daten generiert, wie das «Super Mario Kart». Dabei wird ein dreischichtiges Neuronales Netz zum erlernen der Q-Funktion eingesetzt. Als Eingang werden der Bildausschnitt in 100 Pixel, die Geschwindigkeit, die Richtung sowie die neun möglichen Aktion eingelesen. Die Anzahl der Eingänge liegt somit bei 111. Als Output wird nur ein Ausgang benötigt, welcher den Q-Wert des Zustandsaktionspaares aufzeigt.
Je nachdem, wie man die Belohnung/Bestrafung einstellt, kann man das Lernverhalten des Agenten sehr genau steuern. Wird der Agent belohnt solange er auf der Strecke in die richtige Richtung fährt, so haben meine Tests gezeigt, dass der Agent möglichst langsam fährt, um so viel Belohnung wie möglich zu erhalten. Belohnt man aber den Agenten entsprechend seiner Geschwindigkeit, so konnte ich feststellen, dass er tatsächlich versucht, schneller zu fahren. Die Trainingszeit liegt bei dieser Version deutlich höher, da die Wahrscheinlichkeit höher ist, dass er aus der Strecke gerät. Dieser trainierte Agent fährt am Ende allerdings wesentlich schneller, als der vorherige.

Angelo Mario Campanile
am.campanile@bluewin.ch




Reinforcement Learning




Programmstruktur