Computer spielen Computerspiele mit Hilfe von reinforcement learning am Beispiel Vier Gewinnt

Pablo Lubitz

BACHELORARBEIT

eingereicht am
Universitäts-Bachelorstudiengang

Informatik

in Bremen

im September 2019

Betreuung:

Thomas Barkowsky, Holger Schultheis

Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Bremen, am 10. September 2019

Pablo Lubitz

Inhaltsverzeichnis

Vorwort

TODO

Kurzfassung

TODO

Abstract

TODO engl.

Kapitel 1

Einleitung

In der Bachelorarbeit soll ein Programm entwickelt werden das mit Hilfe von maschine learning trainiert um das Spiel Vier Gewinnt zu erlernen. Es wird ein maschine learning Ansatz gewählt da hierdurch automatisiert individuelle Gegner geschaffen werden können die das können unterschiedlicher Spieler widerspiegeln.

1.1 Motivation

Warum spielen wir? Wir spielen um Fähigkeiten zu erlernen, ein Kind lernt Objekte nach Formen zu sortieren oder eine Katze lernt spielerisch das Jagen von Beute. Doch auch erwachsene Menschen spielen noch und können hierdurch ihre Fähigkeiten verbessern. Um den bestmöglichen Effekt zu haben muss das Spiel eine gewisse Schwierigkeit haben die aber noch zu bewältigen sein muss. Da aber jeder Mensch, egal wie erfahren, das Spiel spielen soll ist es oft Hilfreich einen Gegner zu haben der ein ähnliches Level an Erfahrung mitbringt. Wenn nun diese Rolle des Gegners nicht von einem Menschen besetzt werden kann ist es naheliegend einen Computergegner zu erschaffen. Die Frage die ich in dieser Bachelorarbeit versuche zu beantworten ist: Ist es möglich automatisiert verschieden starke Computergegner aus einer Simulation zu extrahieren.

1.2 Problemstellung

Ein Computergegner der für dieses Spiel Vier Gewinnt erschaffen wird kann auf verschiedenen Algorithmen basieren er könnte zum Beispiel in jedem Zug einfach zufällig eines der sieben möglichen Einwurflöcher bedienen oder er könnte durch einen Minimax Algorithmus zu jedem Zeitpunkt den bestmöglichen Zug errechnen. Das Problem hieran ist, dass jeder einzelne Algorithmus programmiert werden muss und somit ein enormer Arbeitsaufwand entstehen kann. Ein Lösungsansatz für diese Problem sind selbstlernende Algorithmen die mit Hilfe von reinforement learning und neuronalen Netzen selbstständig ähnlich wie ein Mensch erlernen wie das Spiel zu gewinnen ist. Hierraus können dann automatisch Computergegner generiert werden.

1.3 Umsetzung

Es wird eine digitale Version des Spiels Vier Gewinnt erstellt und eine Schnittstelle erschafft die es ermöglicht auszuwählen ob ein Mensch oder der Computer die Rolle der Spieler übernimmt. Spielt der Computer soll zwischen verschiedenen Algorithmen ausgewählt werden können. Dies ist wichtig da ein selbstlernender Algorithmus viele Spiele durchlaufen muss um einen Lernfortschritt aufzuzeigen. Somit kann in der Simulation dann trainiert werden ohne jedes einzelne Spiel gegen einen Menschen spielen zu müssen. Als Trainingsgegner wird ein Minimax Algorithmus dienen. Nachdem die Simulation dann durchlaufen ist können verschieden gute Varianten des selbstlernenden Algorithmus extrahiert werden indem die aktuelle Version zu verschiedenen Trainingsstadien abgespeichert wird. Die Auswahl der Varianten wird anhand der prozentualen Gewinnchance durchgeführt.

1.4 Evaluation

Um zu testen wie gut der maschine learning Algorithmus das Spiel in seinen verschiedenen Trainingsstadien gelernt hat werden die Varianten gegen den Minimax Algorithmus mit unterschiedlicher Suchtiefe antreten. Die unterschiedlichen Suchtiefen stellen dann die Spieler mit bestimmt guten Kenntnissen über das Spiel dar. Somit kann die Stärke der verschieden stark trainierten Varianten an Gegnern getestet werden die nicht weiter dazu lernen und somit einen guten Vergleich bieten. Zu erwarten ist, dass die weniger trainierten Varianten früher mehr gegen die Gegner mit ansteigender Suchtiefe verlieren werden als die mehr trainierten Varianten. Hierdurch soll sich dann zeigen, dass durch das maschine learning Gegner auf automatisierte Weise geschaffen werden können die einen fein granularen Schwierigkeitsanstieg bieten können.

1.5 Related work

Kapitel 2

Grundlagen

In dieser Arbeit werden einige Technologien genutzt um die Problemstellung zu lösen. Die wichtigsten dieser Technologien werden hier erklärt um das weitere Verständniss des Lesers zu gewährleisten.

2.1 Vier Gewinnt

Vier Gewinnt ist ein Spiel für 2 Spieler in dem abwechselnd Spielsteine in ein vertikales Spielfeld eingeworfen werden. Jeder Spieler kann sich in seinem Zug zwischen einem von sieben Einwurflöchern entscheiden. Wird ein Spielstein eingeworfen so fällt dieser auf die niedrigste der sechs Positionen die noch nicht durch andere Spielsteine gefüllt ist. Sind schon sechs Spielsteine in ein bestimmtest Einwurfloch gesteckt wurden so darf hier kein weiterer Spielstein eingeworfen werden. Ein Spieler gewinnt das Spiel wenn er es geschafft hat, dass sich vier seiner Spielsteine in einer Reihe befinden. Eine Reihe kann horizontal, vertikal oder diagonal gebildet werden. Werden alle 42 Positionen mit Spielsteinen befüllt ohne eine Reihe von vier Steinen des selben Spielers zu bilden geht die Partie unentschieden aus.

2.2 Reinforcement Learning

Reinforcement Learning (Bestärkendes Lernen) ist eine Maschine-Learning Methode bei der ein Agent mit Hilfe eines Environments, Policy, State, Actions und eines Rewards lernt eine ihm gegebene Aufgabe zu lösen. Die Aufgabe ist in diesem Fall das Spiel Vier Gewinnt zu meistern.

2.2.1 Agent

Bei dem Agenten handelt es sich um den Acteur der die ihm gegebene Aufgabe meistern soll. Er sieht das Environment und wählt mittels der Policy die ihm am besten erscheinende Action aus. Hier ist der Agent einer der beiden Spieler von Vier Gewinnt.

2.2.2 Environment

Das Environment beschreibt das gesamte Problem dass der Agent versucht zu lösen. In Vier Gewinnt ist dies das Spielfeld und der Gegner.

2.2.3 Policy

Mit der Policy wird das Verhalten beschrieben nach dem der Agent entscheidet welche der möglichen Actions die aktuel beste ist um den Reward zu maximieren.
TODO

2.2.4 State

Der State beschreibt den actuellen Zustand des Environments. Für Vier Gewinnt beschreibt der State welche der 42 Felder mit welchen Steinen befüllt sind.

2.2.5 Actions

Actions sind die möglichen Aktionen zwischen denen sich der Agent je nach State entscheiden muss. Die Actions in Vier Gewinnt beschreiben die sieben Löcher in die der Agent Steine werfen kann.

2.2.6 Reward

2.2.7 psychologischer Hintergrund

Reinforcement Learning funktioniert nach dem Prinzip von Trial-and-Error.
TODO

2.2.8 Q-Funktion

TODO

2.3 Software

Um die Problemstellung zu bearbeiten wurde eine Implementierung des Spiels in Python genutzt und aus dieser mit openai gym ein Environment geschaffen. An diesem Environment trainiert dann ein Agenten der durch Keras-RL erschaffen wurde.

2.3.1 Python

TODO

2.3.2 Tensorflow

TODO

2.3.3 Keras-RL

TODO

Kapitel 3

Konzept

Wie soll alles gemacht werden

3.1 Plan

Was ist der Plan?

Kapitel 4

Implementierung

Vorgehensweise bei der Implementierung

4.1 Agent

4.2 Environment

4.3 Policy

4.4 Reward

4.5 State

4.6 Actions

Kapitel 5

Evaluierung

Was kann evaluiert werden?

5.1 Auswertung

Kapitel 6

Fazit

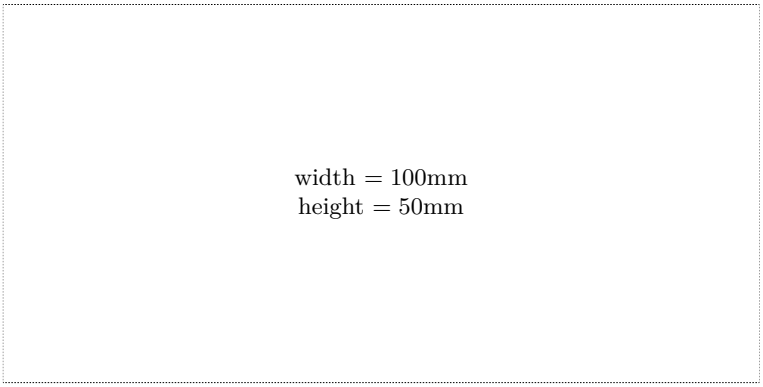
Was wurde festgestellt auswertung ausblick

6.1 Besonderheiten

Quellenverzeichnis

Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —