

Checkers Simulation 2.0

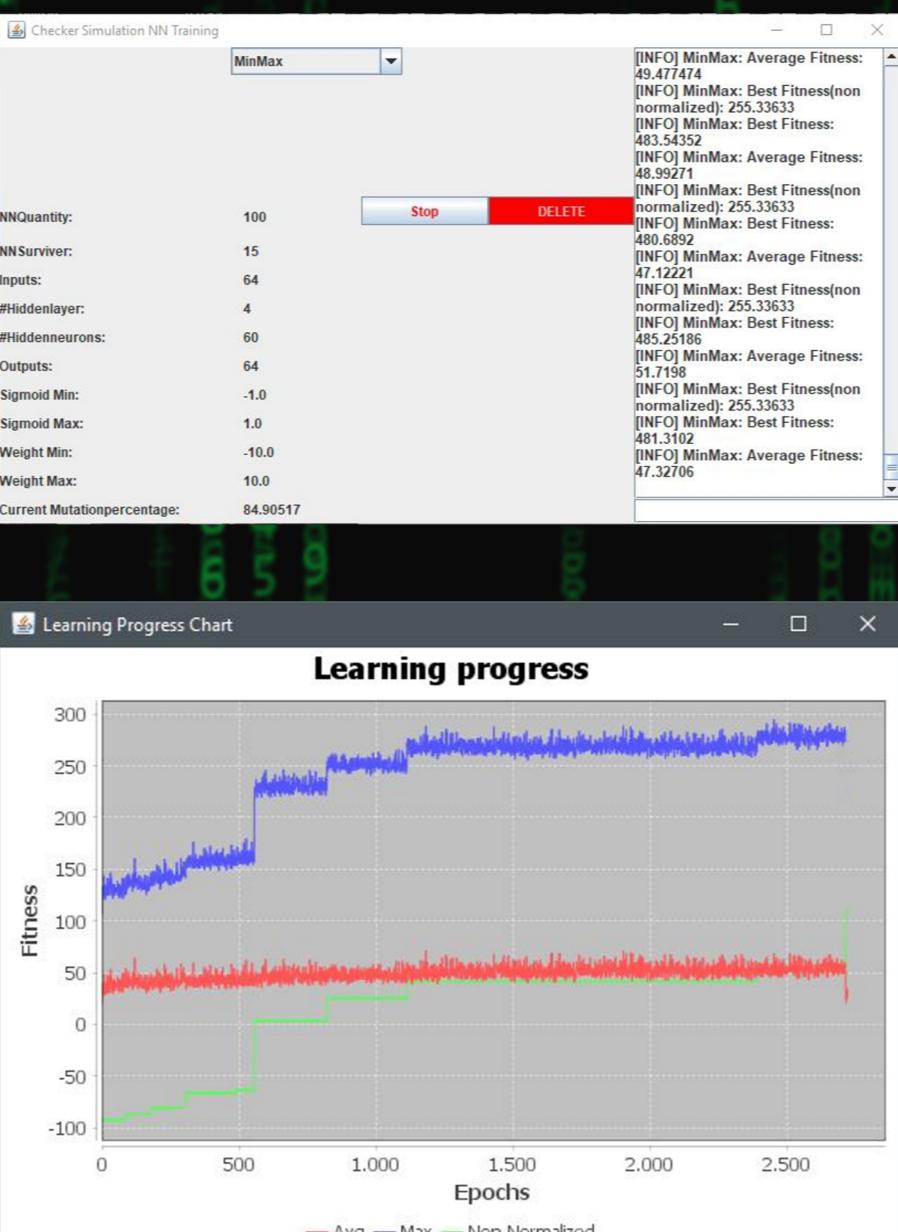
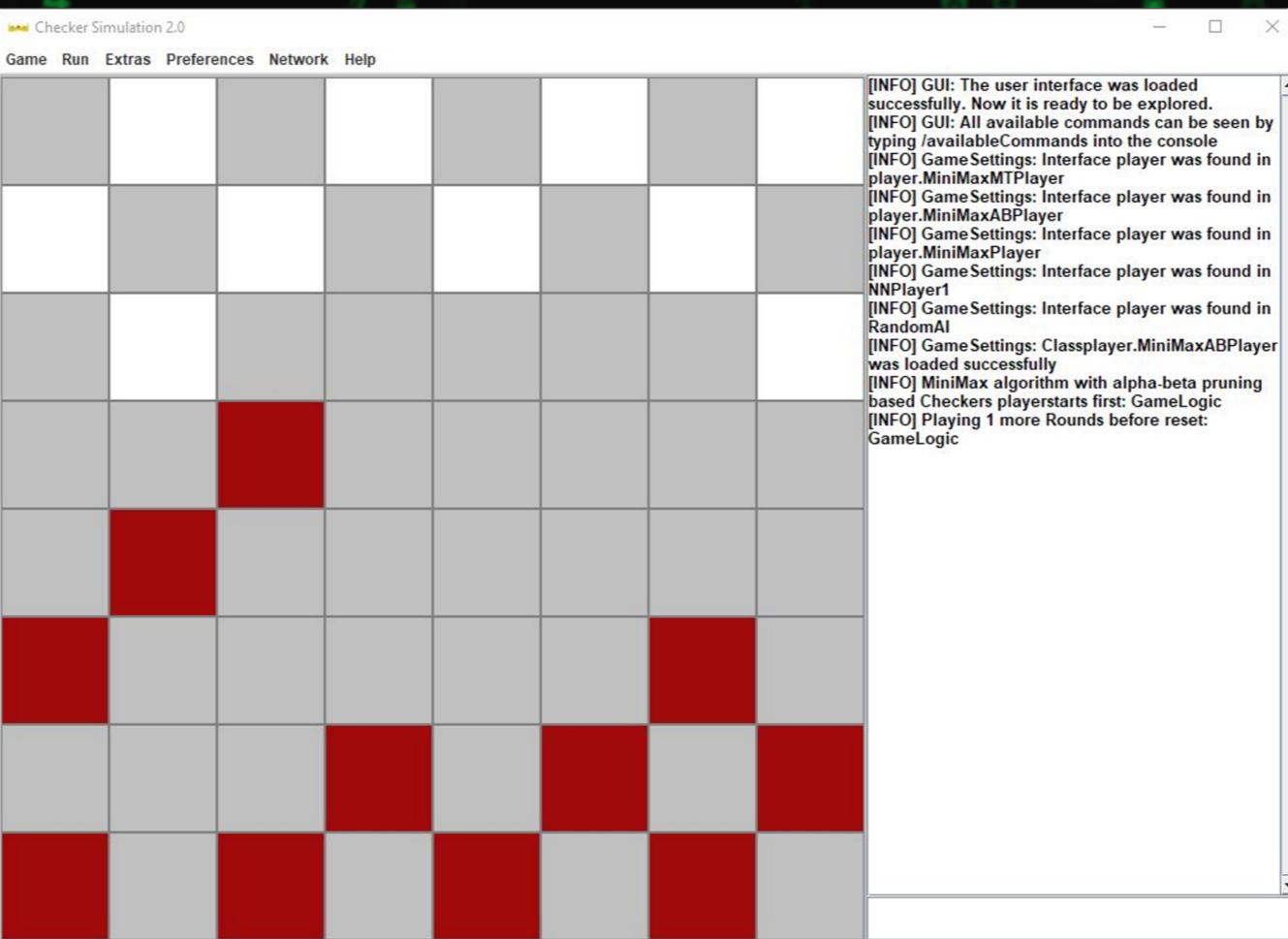
Unser Projekt

Im Bereich der Informatik gibt es besonders bei komplexen Sachverhalten eine Vielzahl an Lösungsmöglichkeiten. Der beste Weg diese Neuheiten gegenüber dem schon Existierenden zu testen, ist, besonders bei Spielen, die reale Anwendung der Algorithmen. So können einfach Effizienz, Spielstärke und andere wichtige Kriterien untersucht werden. In unserem Projekt wollen wir genau diesen Vergleich zwischen konventionellen Algorithmen und neuronalen Netzwerken durchführen. Dafür war der erste nötige Schritt ein Programm zu entwickeln, das alle diese Aufgaben bewältigen kann. Es braucht also eine Funktion, die es ermöglicht Dame zu spielen, einfach neue Spielerarten hinzuzufügen und Daten von den Spielen zu sammeln. Das alles sollte zudem möglichst einfach über eine graphische Oberfläche zu bedienen sein.

Der zweite Teil war die Programmierung der Dame-Spieler selbst.

Alle Spieler werden mit dem Programm über ein "Player" Interface angebunden. Das heißt, dass man als Programmierer den Spieler nur mit einigen festgelegten Methoden ausstatten muss und er dann vom Programm während der Laufzeit geladen und genutzt werden kann. Das hat den Vorteil, dass man keine einzige Zeile Programmcode verändern muss um einen Spieler hinzuzufügen. Der Nutzer muss nur die Datei(en) des fertigen Spielers nehmen und sie in den passenden Ordner ziehen.

Bei der Implementation unserer Spieler haben wir uns zum einen auf das MiniMax-Prinzip und zum anderen auf neuronale Netze konzentriert. Um diese Netze auch zu trainieren, bedurfte es eines zweiten Programms: CS 2.0-NNT (Neural Network Training). Hiermit kann man verschiedene Netzstrukturen in verschiedenen Szenarien trainieren und ihren Lernfortschritt überwachen. Zusätzlich verfügt dieses Programm über die Möglichkeit den Fortschritt der "Trainingssessions" im JSON Format zu speichern. Dadurch kann man jedes Training jederzeit beenden und fortsetzen, ohne den Fortschritt zu verlieren.

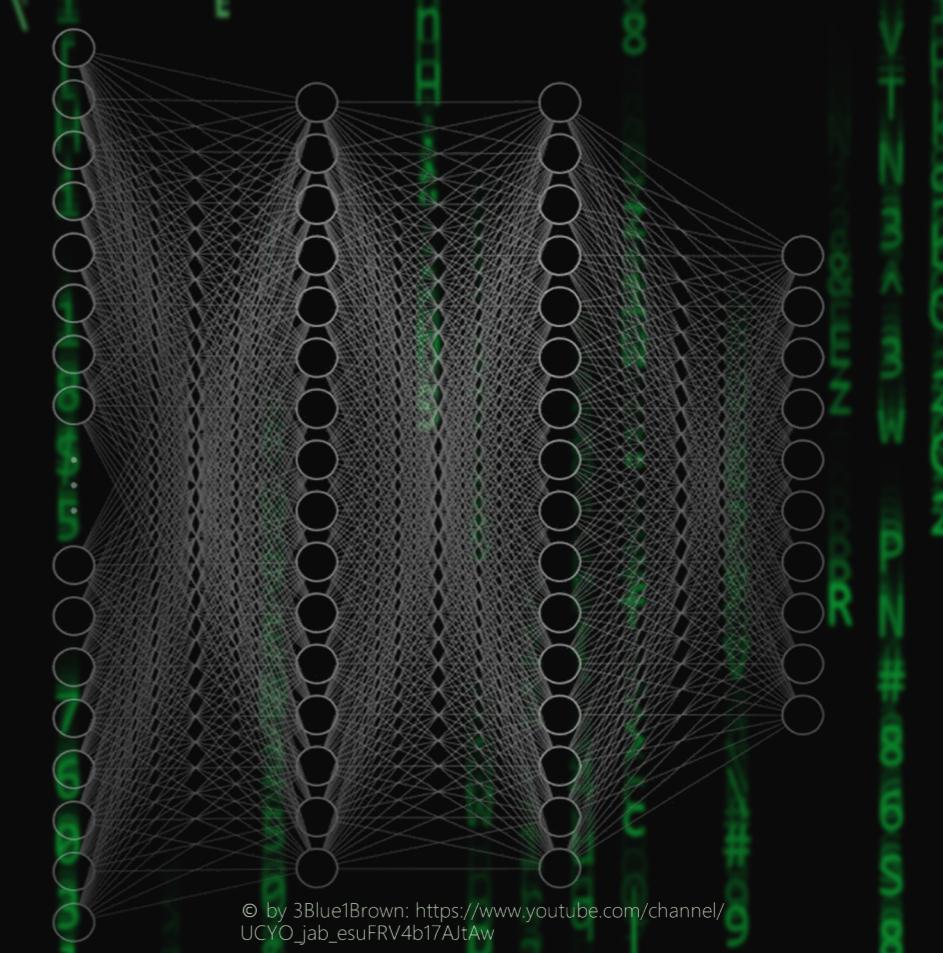


Sind neuronale Netze die ultimativen Problemlöser?

Das ist eine der Fragen die wir uns am Anfang unseres Projektes stellten. Wenn man in diversen Medien sieht, dass Firmen wie Google, NVIDIA oder OpenAI, die monatlich Durchbrüche in der Erforschung neuronaler Netzwerke für autonomes Fahren, komplexe Spiele wie GO oder Anderes erreichen, bekommt man den Eindruck ein neuronales Netz kann alles was ein Mensch kann. Allerdings können wir Aufgrund unserer gewonnenen Erfahrung versichern das es in naher Zukunft keine künstlichen Gehirne geben wird.

Auch wenn ein neuronales Netzwerk (NN) die Strukturen im Gehirn nachahmt fehlt die große Leistung die ein Gehirn erbringen kann und besonders die vorgefertigten Strukturen, schon "programmierte" Instinkte und die Jahrtausende lange Verbesserung durch die Evolution.

Um auf die Ursprungsfrage zurückzukommen:
Nein, NNs können nicht jede Aufgabe bewältigen und es ist auch nicht sinnvoll sie für Alles zu nutzen. NNs sind für die Probleme gut, die schwierig logisch zu abstrahieren und in Code darzustellen sind. Auch wenn NNs diese Prozesse vereinfachen, ist dabei noch eine Menge Arbeit nötig um die optimale Struktur des NNs zu finden, Daten geeignet aufzubereiten, Daten überhaupt zu sammeln und eine Schnittstelle für das NN zur Außenwelt zu schaffen.



© by 3Blue1Brown: https://www.youtube.com/channel/UCYO_jab_esuFRV4b17AitAw

Marco Adamczyk
Can Ward
Till Brinkmann

Herr Frerkes

Gymnasium
Steinhagen