Vorwort:

Machine-Learning ist eines der am meist wachsenden Felder in der Computer Science und rund um die Welt sehr gefragt. Mit den unzähligen Möglichkeiten wie Machine-Learning der Technischen Entwicklung der Menschheit behilflich sein kann ist für diese Technologie auch kein Ende in Sicht. Schon seit ich meinen ersten Utopie Film mit Künstlicher Intelligenz gesehen habe interessierte mich das Thema. Ende des Jahres 2020 hatte ich das Gefühl, dass meine programmier-Kenntnisse genug fortgeschritten waren, um mich dem Thema Machine-Learning anzunehmen. Im Frühjahr 2021 stand dann die Maturaarbeit an und ich musste mich für ein Thema entscheiden. Und ich hatte die Idee meine andere Passion Gaming mit Machine-Learning zu verbinden und mir das Ziel gesetzt meinem PC beizubringen wie man Computerspiele spielt.

Einleitung:

Was ist Rocket League:

Damit ich einem Computer ein Spiel beibringen kann muss es einige Kriterien erfüllen. Um das Projekt im Rahmen der Maturaarbeit fertigstellen zu können muss das Spiel eine sinnvolle Komplexität haben. Auch muss ich das Spiel möglichst einfach über Computerprogramme ansprechen können und auch Spieldaten daraus auslesen können um daraus Trainingssets für das Neuronale Netzwerk gewinnen zu können. Alle diese Kriterien erfüllt RocketLeague das von Psyonix entwickelt wurde. Das Spielprinzip ähnelt einer vereinfachten Form von Fussball indem zwei Teams versuchen mit Autos einen Ball in das gegnerische Tor zu befördern. Dieses Spiel eignet sich sehr gut für solch ein Projekt weil es auf einem immer gleichen Spielfeld stattfindet und die Grundbedingungen immer gleich bleiben was es für ein Neuronales Netzwerk deutliche einfacher macht bedeutsame Entscheidungen zu treffen. Auch wurde ein Framework namens RLBot für Rocket League entwickelt welches einem erlaubt einigen Programmiersprachen unter anderem Python erlaubt RocketLeague Controllerinputs zu geben und einem auch erlaubt die aktuellsten Spielwerte zu erhalten wie zum Beispiel wo sich der Ball oder die Spieler befinden. Dies erlaubt mir möglichst viel Zeit der praktischen Arbeit in die Entwicklung des Bots stecken zu können und somit muss ich mich nicht noch mit anderen Problemen befassen.

Fragestellung:

In meiner Maturaarbeit wollte ich herausfinden, wie gut solch ein RocketLeague Bot im Rahmen der Maturaarbeit werden kann und ob er schlussendlich sogar einen Menschlichen Spieler schlagen kann.

Auch welche Komponenten und Probleme eines solchen Projektes sich effektiv sinnvoll mit Machine-Learning lösen lassen und welche eine weak-ai also eine künstliche Intelligenz die eine spezifische Aufgabe gut kann beziehungsweise ein simples neuronales Netzwerk nicht sinnvoll lösen kann.

Um eine gute Künstliche Intelligenz zu erzeugen ist natürlich unfassbar wichtig, dass man den richtigen neurale Netzwerke typ wählt und fast noch wichtiger, dass man die richtigen Daten wählt um diese zu trainieren. Dazu gehört das Netzwerk nicht zu overfitten keine unnötigen oder unverständlichen Daten in das Netzwerk zu füttern

Zusätzlich finde ich noch interessant herauszufinden welche spielerischen Vorteile ein vom Computer gesteuerter Bot über den Menschen hat.

Begriffe

Hauptteil:

Librarys:

Erstmals benutze ich die RLbot Python library die mir sehr unkompliziert erlaubt meinen code mit RocketLeague zu verknüpfen. Einerseits erlaubt mir RLbot eine Klasse zu verwenden die dessen get\_output() funktion jeden frame läuft und es mir erlaubt darin meine routinen laufen zu lassen, daten direkt aus dem Spiel zu benutzen und schlussendlich einen controllerinput zurück zu geben. Zusätzlich hat diese Library unglaublich praktische debugging featueres die mir erlauben Gegenstände im Spiel zu teleportieren eigene Spielregeln fest zu legen und polygone und text direkt in das Spiel zu zeichnen. Auch liefert mir RLbot eine vorgefertigte Vektor 3 Klasse die ich aber noch leicht angepasst habe um sie optimal nutzen zu können.

Da das programmieren von guten neuronalen Netzwerken sehr viel Zeit in Anspruch nimmt habe ich mich dafür entschieden eine Library namens Tensorflow dazu zu benutzen. Damit kann ich immer Netzwerke verwenden die optimal funktionieren und unkompliziert Änderungen an diesen vornehmen was essentiell dafür ist herauszufinden wieso sich ein Netzwerk nicht so verhält wie es soll.

Um meine Neuronalen Netzwerke optimal überwachen zu können und um sie zu optimieren benutze ich Matplotlib um graphen zu zeichnen. Dies erlaubt mir auch Phänomene wie overfitting zu erkennen oder das Netzwerk zu optimieren, da ich finde, dass die visuelle darstellung dieser Daten deutlich einfacher zu verstehen und zu interprätieren ist.

Zusätzlich benutze ich für das einfachere arbeiten mit den Daten der Neuronalen Netzwerke Numpy. Ich verwende dieses hauptsächlich für die unzähligen array funktionen die es beinhaltet.