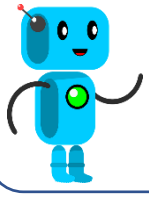
Fachhochschule Südwestfalen
University of Applied Sciences

Snake AI programmieren

Ferienkurs an der FH-SWF

Fachhochschule Südwestfalen
University of Applied Sciences



Hier könnt ihr nun testen, wie gut Ihr im Snake spielen seid. Versucht einige Runden lang einen möglichst guten Highscore zu erzielen und merkt euch euer Ergebnis.

1. Erreichte Punktzahlen:

2. Ist es möglich die Geschwindigkeit der Snake zu verändern, wenn ja, wie?

3. Ändert nun die Geschwindigkeit und versucht euren bisherigen Highscore zu knacken.

Fachhochschule Südwestfalen
University of Applied Sciences

„Maschinelles Lernen“ programmieren

Was ist „Maschinelles Lernen“?

Künstliche Intelligenz beschäftigt sich mit der Frage, ob Computer oder Roboter in der Lage sind, Dinge, die wir Menschen (zurzeit noch) besser können, zu erledigen. Dinge, wie das Finden eines Schlüssels in der Hosentasche oder das Finden einer Tür in einem Raum fallen uns Menschen leicht, den Robotern jedoch sehr schwer. Hierbei ist besonders die Lernfähigkeit des Menschen ein großer Vorteil gegenüber den Robotern. Was aber, wenn wir eine Möglichkeit finden einer Maschine das Lernen beizubringen? Mit dieser Frage beschäftigen wir uns und werden Anhand des Spieles „Snake“ versuchen dem Computer beizubringen, besser als wir Menschen zu spielen.

Gibt es Unterschiede beim „Maschinellen Lernen“?

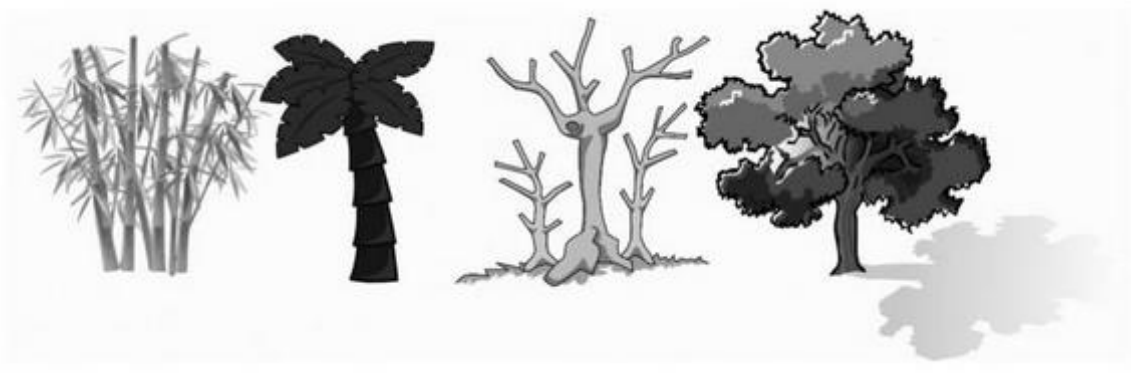
Beim „Maschinellen Lernen“ gibt es im Wesentlichen drei verschiedene Lernmethoden, welche auf unterschiedlichen mathematischen Formeln basieren.

Überwachtes Lernen

Das überwachte Lernen bedarf eines Lehrers. Den darf man sich jetzt allerdings nicht als eine Person vorstellen, welche die ganze Zeit das Programm überwacht. Es geht darum, dem Programm eine große Menge von Ein- und Ausgaben zur Verfügung zu stellen, die bereits über den korrekten Funktionswert verfügen. Ein Beispiel kann ein großer Datensatz von Bildern mit Hunden und Katzen sein, wo für jedes Bild hinterlegt wurde, ob es ein Hund oder eine Katze ist. Diese Daten werden dann benutzt, um dem Computer beizubringen, auf welchen Bildern Hunde und auf welchen Katzen zu sehen sind. Wenn dann neue Bilder hinzukommen, hofft man, dass dieser Anhand der bisherigen Bilder die neuen Bilder selbstständig einordnen kann.

Unüberwachtes Lernen

Während beim überwachtem Lernen dem Computer eine Sammlung von Zielwerten gegeben wird, gibt es Fälle, in denen es auf diese Weise nicht möglich ist, das Verhalten positiv oder negativ zu bewerten. Zum Beispiel bei der Abbildung mit den unterschiedlichen Pflanzen. Wie würdet Ihr Sie einsortieren?



Wie man sehen kann, kann man die Pflanzen auf unterschiedliche Weise einordnen. Jeder Datensatz hier hat unterschiedliche Merkmale und man untersucht sie auf Ähnlichkeiten. Genau so geht der Computer beim unüberwachtem Lernen vor.

Bestärkendes Lernen

Da der Computer nicht weiß, was richtig oder falsch ist bzw. wie die optimale Strategie aussieht, um z.B. eine Wohnung zu saugen, benötigt man eine Möglichkeit es Ihm beizubringen. Man weiß ja, was ein gutes Ergebnis ist, und was ein Schlechtes, z.B. wenn ein Putzroboter am Treppenabsatz herunterfällt und nicht stehen bleibt, oder er immer wieder die gleichen Stellen putzt. Für solche Problemfälle gibt es Lösungen aus dem Bereich des „Bestärkenden Lernen“ bzw. englisch Reinforcement Learning. Hierbei erhält ein Agent (Teil des Programms, welches für das Training des Computers zuständig ist) ständig Rückmeldungen in Form von Belohnung und Bestrafung, wodurch er mit der Zeit eine optimale Strategie für unser Problem lernen soll. Stößt der Roboter seinen Kopf also an der Wand, erhält er eine Bestrafung. Stoppt er rechtzeitig und dreht um, eine Belohnung. In diesem Beispiel bekommt er jeweils +10 Punkte oder -10 Punkte. Stellt es euch vor als würde man einem Kind Süßigkeiten geben oder wieder wegnehmen.

Schauen wir uns das bestärkende Lernen noch einmal genauer an einem besseren Beispiel an. Stellt euch vor, ihr habt einen Hundewelpen bei euch zuhause. Der Hund ist der Akteur. Er hat mehrere Aktionen wie Bellen, zur Tür gehen, sich verstecken, uns ignorieren und liegen bleiben und quasi alles was ihr euch noch vorstellen könnt, wenn man diesem sagt, dass es jetzt „Gassi“ geht. Der Zustand ist „Gassi gehen“, der Hund liegt aber noch im Körbchen. Die Umgebung ist das Zimmer.

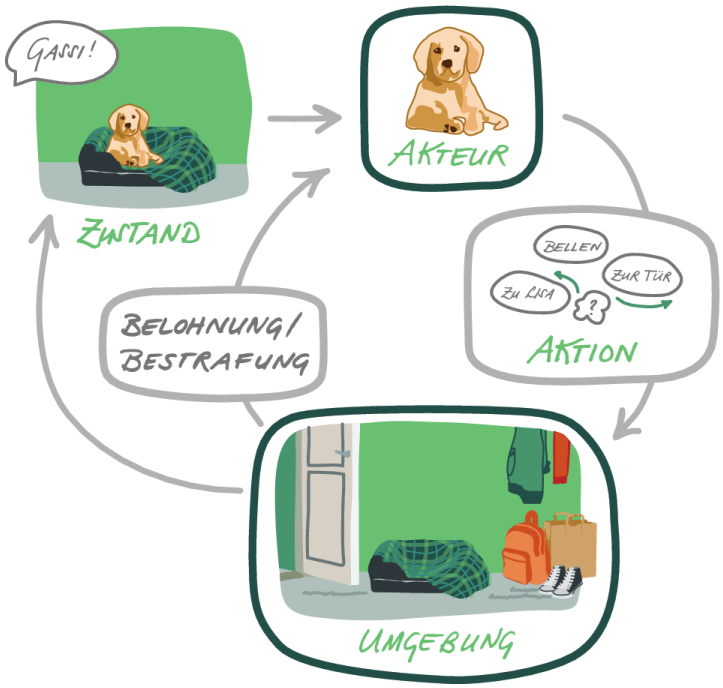
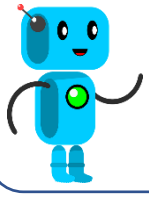


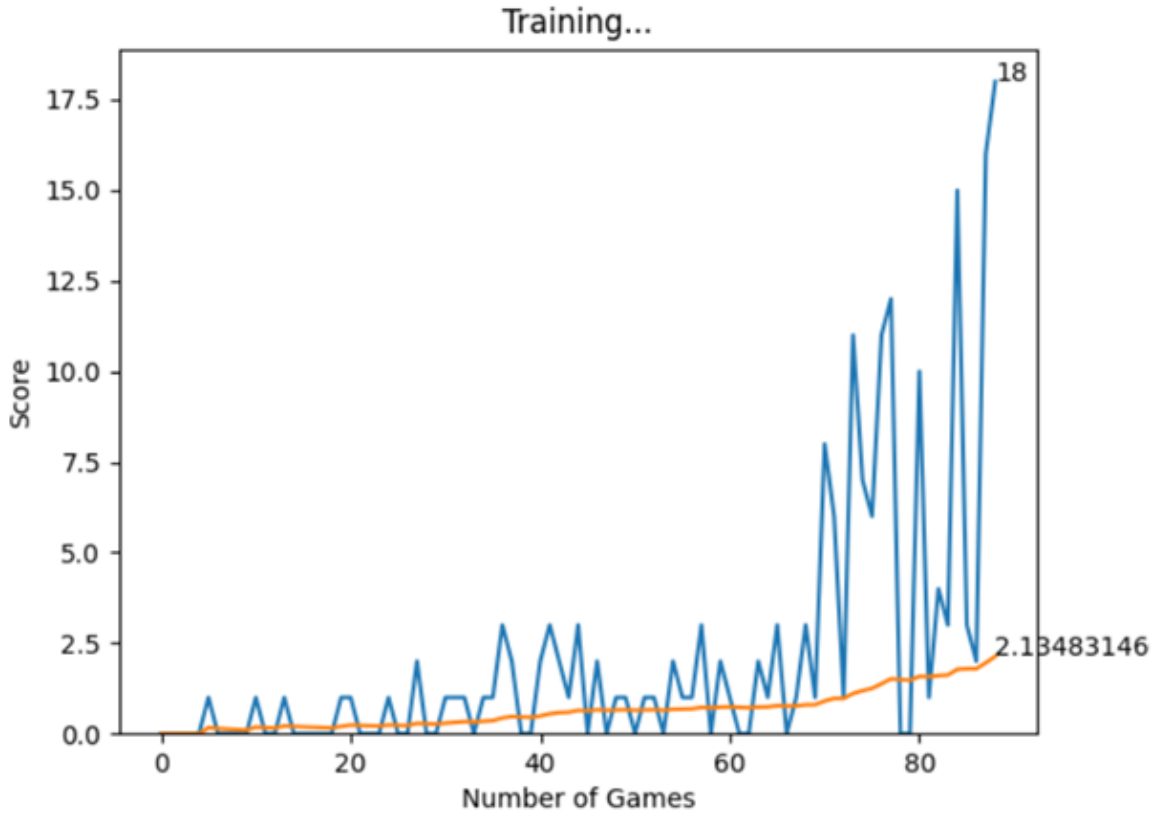
Abbildung 1: Reinforcement Learning am Beispiel vom Hundewelpen.
Quelle: Herrmann, T., Peiss, L.F. (2019). Verstärkendes Lernen. In: Kersting, K., Lampert, C., Rothkopf, C. (eds) Wie Maschinen lernen. Springer, Wiesbaden. https://doi.org/10.1007/978-3-658-26763-6_26

Was wollen wir in der Situation? Wir möchten, dass der Hund mit uns zur Tür kommt, damit wir ihm das Geschirr anlegen und „Gassi gehen“ können. Kommt er mit uns zur Tür, bekommt er ein Leckerchen. Kommt er partout nicht zu uns, sind wir erstmal sauer. Dadurch lernt der Welpe, dass das „Gehorchen“ zu einer positiven Erfahrung führt. Das gleiche tun wir mit unserer KI. Nur kann es mit der KI sehr lange und mitunter tausende oder mehr Durchläufe dauern.



Hier wird es nun etwas schwieriger. Erkennt ihr was die KI tut?
Schaut euch das Spiel und die Durchläufe dafür genau an!

1. Hier seht ihr eine Übersicht der Trainingsanzeige. Was genau sagt die Anzeige aus? Erklärt kurz die untenstehende Übersicht.



Wie funktioniert das Snake-Spiel mit der KI?

Unsere KI sieht genau das gleiche wie wir nur im kleineren Rahmen. Mit jeder Bewegung wird geschaut, welcher Status aktuell in welcher Kategorie vorherrscht. Es gibt insgesamt elf Arten in drei Kategorien. Die Kategorien sind Gefahr, Richtung und Essen.
Eine 1 bedeutet der jeweilige Status trifft zu. Eine 0 bedeutet, dass dieser nicht zutrifft.

Kann ich mir den Kopf stoßen?

Für die Schlange wird bei jeder Bewegung überprüft ob und in welcher Richtung Gefahr lauert.
Ist geradeaus, rechts oder links eine Wand oder ein Teil der Schlange?
Im Code ist diese Art als [danger straight, danger right, danger left] zu finden.

Welche Richtungen gibt es?

Die KI kennt vier Richtungen. Links, rechts, oben und unten. In diesen vier Richtungen wird immer wieder geprüft, wo sich das Essen oder eine Gefahr befindet.
[direction left, direction right, direction up, direction down]

Wie bewegt sich die Schlange?

Damit die Schlange sich nicht selbst isst, sieht sie selbst nur von ihrem Kopf aus in die Richtungen links, rechts und geradeaus. Sie kann also nicht rückwärts in sich hineinlaufen.
Die Schlange selbst kennt also drei Richtungen.
Dafür gibt es:
[1,0,0] -> Straight (In gleicher Richtung bleiben)
[0,1,0] -> Right Turn (Rechtsdrehung)
[0,0,1] -> Left turn (Linksdrehung)

Die Schlange bewegt sich auch in dieser Reihenfolge. Immer erst geradeaus, danach falls Essen oder Gefahr in der Nähe ist, rechts, dann links. Dann wieder geradeaus bis zu dem nächsten Hindernis und darauffolgend wieder rechts und dann links. Nach diesem Muster geht die KI größtenteils vor.

Wo ist das Essen?

Die KI sucht ständig nach dem Essen, da sie ja +10 Punkte für jedes erfolgreich „gefangene“ Essen bekommt. Es wird immer wieder geschaut, wo sich die „Beute“ befindet. Ist sie oben, oder unten, ist sie links oder rechts?
[food left, food right, food up, food down]

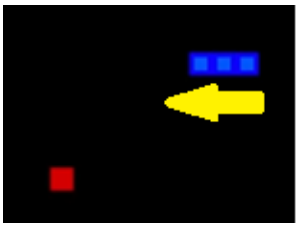
Aufgabe: Gemeinsamkeiten zwischen den Tieren

Schaut euch noch einmal Abbildung 1: Reinforcement Learning am Beispiel vom Hundewelpen. Quelle: Herrmann, T., Peiss, L.F. (2019). Verstärkendes Lernen. In: Kersting, K., Lampert, C., Rothkopf, C. (eds) Wie Maschinen lernen. Springer, Wiesbaden. https://doi.org/10.1007/978-3-658-26763-6_26 an. Wie könnt ihr das Modell auf das Snake-Spiel übertragen? Was ist was? Wo sind Gemeinsamkeiten?


Fachhochschule
Südwestfalen
University of Applied Sciences

Was bedeutet das jetzt für die Programmierung?

Schaut euch einmal die untenstehende Grafik an.



Wenn man die drei Kategorien einmal nach dem Muster durchgeht, sind folgende Status verfügbar.

KATEGORIE 1 (GEFAHR):

[0,0,0] -> Keine Gefahr, deswegen trifft danger straight, right und left nicht zu!

KATEGORIE 2 (RICHTUNG):

[1,0,0,0] -> Direction left stimmt. Die KI sieht, dass wir uns auf dem Bildschirm nach links bewegen. Das Essen ist zu weit weg, um schon in eine andere Richtung zu schauen. Diagonal können wir nicht schauen!

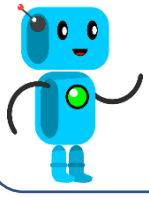
KATEGORIE 3(FUTTER):

[1,0,0,1] -> Um an das Futter zu gelangen müssen wir später nach links und dann noch nach unten! Nach oben oder rechts müssen wir uns nicht bewegen.

Aufgabe: Könnt ihr wie eine KI denken?

Die KI denkt, ähnlich wie wir. Nur ist das Denken komplizierter dargestellt. Versucht einmal im Kopf genau nachzuvollziehen, wie ihr selbst auf die Lösung in dem Szenario unten kommt. Stückelt eure Gedanken auf. In welchen Schritten denkt ihr, wenn ihr jetzt diese Schlange in die Richtung des Futters lenken müsstet?

Schreibt eure Überlegung genau auf! Könntet ihr auch nachvollziehen, wie das Muster für die KI nach dem obigen Schema lauten würde?



Nun schauen wir uns einmal den Programmcode genauer an!
Was könnt ihr noch erkennen? Achtet auf die Kommentare.

Kennt ihr schon das RGB-Farbmodell?

Das Farbmodell RGB (**Rot**, **Grün**, **Blau**) ordnet jeder Farbe einen spezifischen numerischen Wert zwischen 0-255 zu. Dies passiert für jede der Primärfarben Rot, Grün und Blau. Die Zahl zeigt das Mischverhältnis von Rot, Grün und Blau an, mit dem eine bestimmte Farbe erzeugt wird.

Beispiele dafür könnt ihr hier sehen:

Farbe	Rot-Wert (R)	Grün-Wert (G)	Blau - Wert (B)
Wassermelone	234	62	112
Mangogelb	249	175	31
Blutorange	230	6	45
Grüner Apfel	155	181	49
Glutrot	198	44	58
Brombeermousse	78	19	57
Blaubeere	0	88	153
Tannengrün	0	121	80

In unserem Code finden wir auch ein Farbschema nach dem RGB.
Sucht die Stellen in der Datei game.py.

1. Ändert einmal die Farbe des „Futters“ für unsere Schlange. Ändert dabei nicht die Bezeichnung, da es sich hier um eine Variable handelt die später im Code genutzt wird. Passt an dieser Stelle bitte erstmal nur die einzelnen Farbwerte an. In welcher Zeile wird die Farbe festgelegt? Welche Farbe habt ihr genutzt?

2. Ändert nun noch ein paar andere Farben. Passt zum Beispiel die Farbe des Hintergrunds oder die der inneren und äußeren Haut der Schlange an. Was habt ihr geändert und welche Farben habt ihr dafür genutzt? Sucht euch gerne auch noch neue Farben heraus. Über eine Suchmaschine im Browser findet ihr bestimmt einen RGB-Generator.

3. Ändert nun die Schriftart in „neuropol.ttf“. Die Datei dafür liegt in dem Ordner. Ihr braucht die Schriftart jedoch nur bei euch im Code anzupassen. Wo und wie habt ihr das gemacht? Hat sich viel geändert?

4. Passt nun auch noch die Schriftgröße an. Diese ist in dem gleichen Befehl zu finden. Was passiert? Welche Werte sind zu groß und welche zu klein?

5. Ändert nun noch einmal die Geschwindigkeit der KI, damit wir schneller die Ergebnisse beobachten können. Am besten auf einen Wert zwischen 100 und 130. Beobachtet in eurer Trainingsübersicht den Graphen bei circa 100 Spielen. Macht dies für fünf Durchläufe.
   Wo liegt der durchschnittliche Wert? Ist dieser immer gleich? Welcher ist jeweils circa der höchste Score?