

# Wie lernen Künstliche Intelligenzen?

**Definition:** Künstliche Intelligenz (KI) bezeichnet die Fähigkeit von Maschinen und Computern, menschenähnliche kognitive Fähigkeiten zu simulieren. Es beinhaltet die Entwicklung von Algorithmen und Systemen, die es Computern ermöglichen, zu lernen, zu verstehen, zu schlussfolgern, Probleme zu lösen und Entscheidungen zu treffen, ähnlich wie es der menschliche Verstand tut.

Durch die Vielfältigkeit der Informatik, also die vielen Möglichkeiten und Wege ein Problem zu lösen, gibt es auch für das Lernen einer KI mehrere Ansätze, mit denen sich dieses Poster beschäftigt.

## Brute-Force-Algorithmus

**Definition:** Brute Force ist eine Methode, um Probleme zu lösen oder Ergebnisse zu erzielen, indem alle möglichen Kombinationen oder Optionen systematisch durchprobiert werden, bis die richtige Lösung gefunden wird.

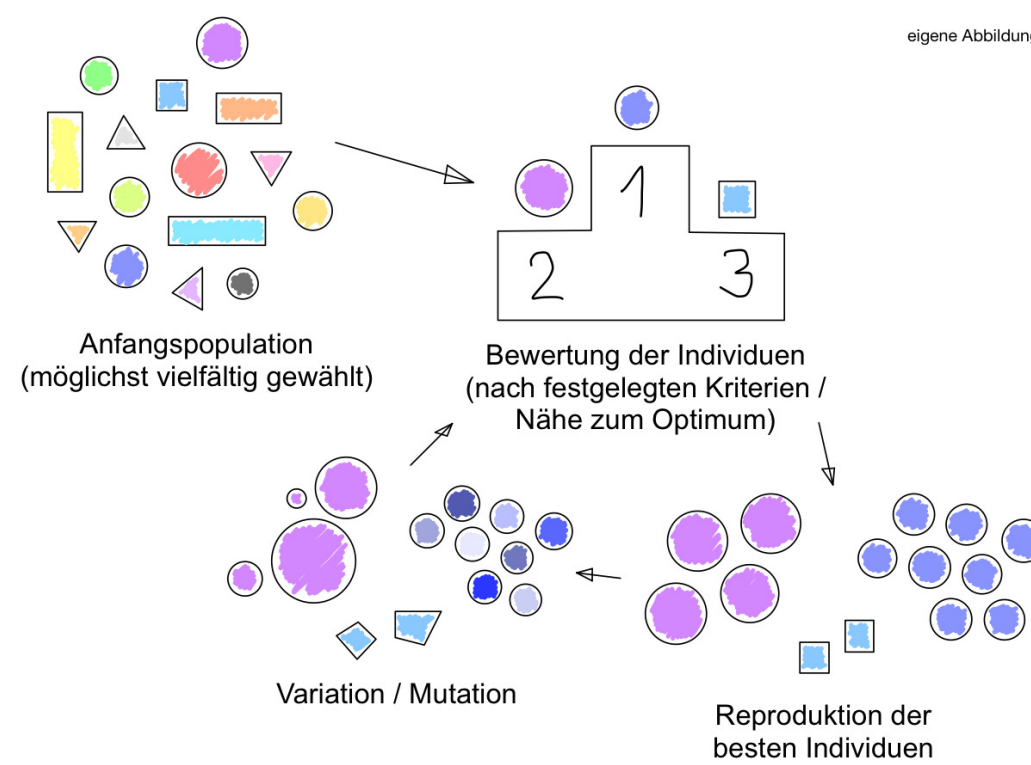
Vorteile	Nachteile
<ul style="list-style-type: none"> <li>einfach zu implementieren und zu verstehen</li> <li>garantierte Lösung (wenn Suchraum endlich)</li> <li>auf Vielzahl von Problemen anwendbar (universell)</li> </ul>	<ul style="list-style-type: none"> <li>hohe Rechenleistung und hoher Speicherbedarf bei großen Suchräumen</li> <li>unpraktisch bei unendlichen Suchräumen</li> <li>viele Kombinationen werden probiert, die definitiv nicht zur Lösung führen (ineffizient)</li> </ul>

→ Brute Force nicht immer beste oder effizienteste Methode für Lösung komplexer Probleme. In der KI daher oft fortschrittlichere Techniken und Algorithmen eingesetzt, wie z.B. maschinelles Lernen, um die Suche im Suchraum zu optimieren und effizientere und intelligentere Lösungen zu finden

**Definition:** Maschinelles Lernen ist eine Technik, bei der Computer-Algorithmen entwickelt werden, um aus Daten zu lernen und aufgrund dieser Erfahrungen automatisch Muster zu erkennen, Vorhersagen zu treffen und Entscheidungen zu treffen, ohne dass sie explizit programmiert werden müssen.

## Evolutionärer Algorithmus / Evolutionary Algorithm

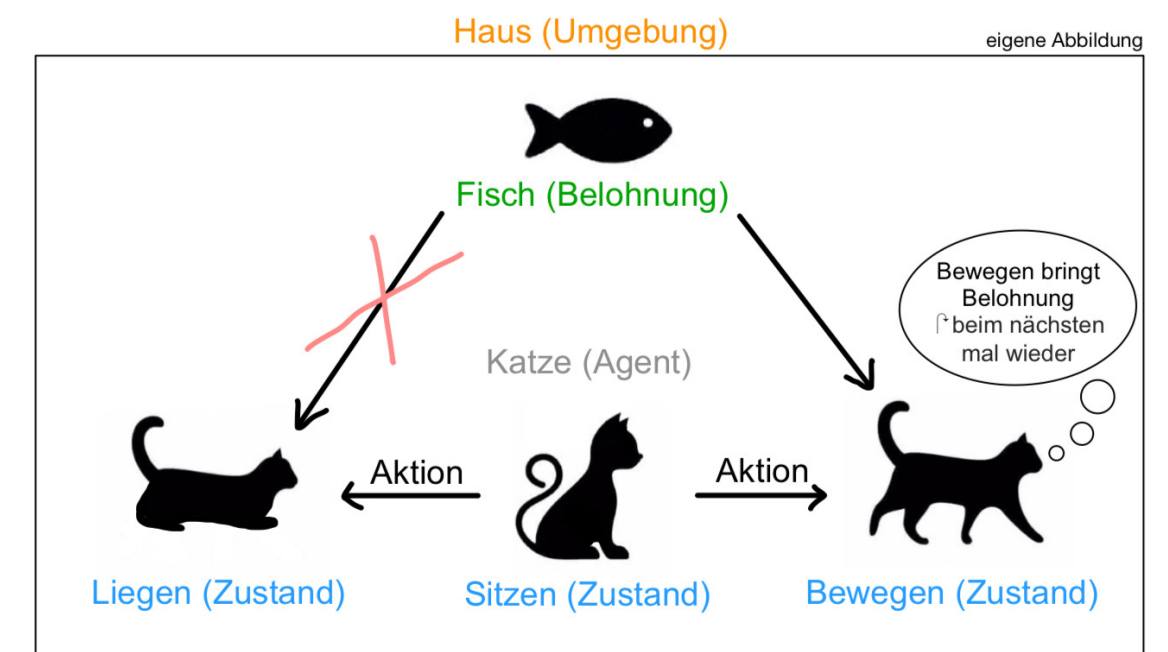
**Definition:** Der evolutionäre Algorithmus ist ein Optimierungsalgorithmus, welcher von der natürlichen Selektion inspiriert ist. Er verwendet Prinzipien der Evolution, um Lösungen für komplexe Probleme zu finden.



Vorteile	Nachteile
<ul style="list-style-type: none"> <li>breite Anwendbarkeit</li> <li>erfordert keine Ableitungen oder analytische Funktionen</li> <li>kombiniert Exploration (Suche nach neuen Lösungen) und Ausbeutung (Verbesserung bestehender Lösungen)</li> </ul>	<ul style="list-style-type: none"> <li>endgültige Lösung die unbedingt die beste, sondern nur besser als der Rest</li> <li>Wiederholen von Fehlern nicht ausgeschlossen</li> <li>hohe Rechenleistung und hoher Speicherbedarf bei komplexen Problemen</li> <li>stark von Parametern abhängig</li> </ul>

## Bestärkendes Lernen / Reinforcement Learning

**Definition:** Reinforcement Learning ist ein maschinelles Lernverfahren, bei dem ein Agent durch Ausprobieren lernt, belohnt wird und optimale Entscheidungen trifft.



Vorteile	Nachteile
<ul style="list-style-type: none"> <li>leicht an sich ändernde Umgebungen anpassbar</li> <li>Algorithmen können über menschliches Wissen hinaus agieren und lernen</li> <li>in vielen Bereichen anwendbar</li> <li>kann komplexe Strategien erlernen</li> </ul>	<ul style="list-style-type: none"> <li>lange Lernzeiten</li> <li>hohe Rechenkosten in komplexen Umgebungen</li> <li>unerwartete Verhaltensweisen möglich</li> <li>schwierige Parametereinstellung</li> </ul>

## Evolutionärer Algorithmus & Reinforcement Learning angewendet

Mit dem Thema dieses Posters haben sich meine Gruppe und ich auch in unserer Seminarfacharbeit beschäftigt. Sie trägt den Titel: "Verschiedene Methoden zur Strategieoptimierung und deren Vergleich" und es folgt eine kleine Zusammenfassung unserer Arbeit.



Abbildungen: Künstliche Intelligenzen "AlphaZero" und "AlphaGo" besiegen Weltmeister in Schach und Go  
→ Spiele bieten gute Grundlage um an KI's zu forschen, weil abgegrenzte Umgebung mit klaren Regeln

Eine **Spielsituation** ist ein einzelner Zustand eines Würfels (also wie viele der Seiten belegt sind und wie viele Augen auf diesen verteilt wurden). Das Ziel unseres Algorithmus war es, eine optimale Strategie für einen der Spieler zu finden. Eine **Strategie** beschreibt für ein Spiel eine Handlungsvorschrift für einen Spieler, die für jede mögliche Spielsituation eine Aktion vorgibt, sodass die größtmögliche Gesamtauszahlung erreicht wird. Durch **Auszahlungen** wird der Algorithmus belohnt oder nicht belohnt. Um Strategien zu vergleichen, wird eine „**Fitness**“ definiert. Sie beträgt den Durchschnitt der Gesamtauszahlungen in einem Spiel gegen alle möglichen gegnerischen Würfel.

Unser finales Diagramm vergleicht die beiden Strategiefindungsansätze mit verschiedenen Arten der Strategierepräsentation (RL nutzt eine Funktion) und wie man sieht, ist Reinforcement Learning die beste Art und Weise dieser zwei (oder sogar drei, Brute Force mit einbezogen) Algorithmen, KI's lernen zu lassen. Was man noch alles Interessantes in diesem Diagramm entdecken kann, ist vielleicht sogar eine Frage an mich wert. ;)

Für unsere Seminarfacharbeit wurde ein einfaches Würfelspiel gewählt, welches klare Regeln und einen Glücksfaktor kombiniert.

### 1. Phase: Erstellung des Würfels

Jeder Spieler darf  $p$  Punkte auf  $s$  Seiten verteilen. Die Spieler sind abwechselnd an der Reihe und vergeben für jede Seite eine beliebige Anzahl an Augen.

### 2. Phase: Würfeln

Beide Spieler würfeln ein Mal. Die höhere Zahl gewinnt.

Für die Repräsentation der Spielstrategien wurden mehrere Ansätze genutzt und bewertet:

- Tabellarische Repräsentation
- Lineare Funktion (mit zwei Ansätzen basierend auf Vektoren)
- Neuronales Netz

Die beiden Algorithmen wurden von uns in Python implementiert und es wurden Fitnessgraphen erstellt, welche die Fitness der besten Strategie bei steigender Anzahl an ausgewerteten Spielen zeigt.

