

BA- oder Projekt-Themen – Stand 08/2023

ChatGPT für Mathe-Aufgaben

[zunächst vielleicht als INF- oder WI-Projekt, evtl. ausbaubar]

ChatGPT von OpenAI braucht wohl heutzutage nicht mehr vorgestellt werden. Die Medien überschlagen sich mit Meldungen, was ChatGPT alles kann, wo es evtl. noch Schwächen hat.

Fragestellung für Projekt:

- Kann **ChatGPT** eine Lernunterstützung für Mathe-Aufgaben liefern? Hierzu sollen ChatGPT geeignete Mathe-Aufgaben vorgelegt werden und die Antworten auf Eignung, Qualität, Stärken und Schwächen bewertet werden.
- Unterfrage: Kann ChatGPT mit mathematischen Gleichungen umgehen und wenn ja, wie?

Arbeitsschritte:

- Vertraut machen mit den Möglichkeiten von ChatGPT
- Umgang von ChatGPT mit mathematischen Gleichungen: Recherche, evtl. Transformation (Gleichung als Programmcode)
- Fundus von Aufgaben, z.B. aus MathWeb/WMINT, <https://wmint.de/mathematik>, zusammenstellen. (Hierzu kann ich später noch Unterstützung anbieten)
- Lösungen sammeln und bewerten. Konsistenz und Reproduzierbarkeit?
- Inwieweit können die Antworten den Lernprozess von Studierenden unterstützen? Kann ein Studierender beim Chat mit ChatGPT Nützliches erfahren, das ihn/sie beim Lösen solcher Aufgaben unterstützt, das Erklärungen bietet?
- Diskussion und Zusammenfassung

Einstiegspunkte in Literatur / Links:

- <https://chat.openai.com/chat>: ChatGPT-Zugang
- WI-Prof an HS Aschaffenburg hat ChatGPT seine Wirtschaftsinformatik-Klausuraufgaben vorgelegt und findet in <https://www.youtube.com/watch?v=SS24iTTOn7o>, dass ChatGPT die meisten mit Bravour löst.
- [Bing-KI mit GPT-4](#): Man muss diesen Link in MS Edge-Browser öffnen und mit einem Microsoft-Konto angemeldet sein. Diese Bing-KI mit GPT-4 scheint besser Mathe zu können als ChatGPT.
- Diskussion über ChatGPT in [diesem LinkedIn-Beitrag](#) der TH Köln
- <https://wmint.de/mathematik> : MathWeb/WMINT

Voraussetzungen

- Interesse an Mathe-Didaktik und an ChatGPT (und Varianten)

Stochastische RL-Algorithmen Python für Blackjack + (Kuhn) Poker

- Ausgangslage: Vorangegangene BAs und MAs haben Blackjack [Meissner2021] und Poker [Zeh2021a,b] in GBG implementiert. Allerdings stellte sich heraus, dass die Algorithmen in GBG (keine stochastische Policy-Lerner) nicht so gut geeignet sind für solche stark stochastischen Spiele.
- In Python (Open AI Spinning Up, StableBaselines3) sind auch stochastische Policy-Optimizer vorhanden.
- Zielsetzung BA: Herausfinden, wie geeignet diese Policy-Optimizer für stochastische Spiele sind. Verstehen, wie sie arbeiten, wo die Vor- und Nachteile liegen.
- Herangehensweise
 - Einarbeiten in die Spiele Blackjack, KuhnPoker, evtl. Poker (BA + MA liegen vor)
 - Erste Erfahrung mit einer vereinfachten Python Blackjack-Implementierung in OpenAI Gym [[OpenAIBlackJack](#)] sammeln
 - Evtl. diese aufbohren auf volles Blackjack.
 - Analog in Python Kuhn-Poker als OpenAI Gym Env realisieren
 - Testen verschiedener SB3 Algos [SB3] darauf → Leistung, Vor- und Nachteile
 - Evtl. Konzept für Interface zw. trainierten Python Agenten ↔ GBG (Java). Hier kann auf Konzepten der [Projektarbeit von Niklas Fabig](#) aufgebaut werden.

Voraussetzungen

- Kenntnisse Python (und evtl. Java).
- Interesse an Reinforcement Learning

Literatur:

[Meissner2021] Meißner, Simon: „[Untersuchung des Spiel- und Lernerfolgs künstlicher Intelligenzen für ein nichtdeterministisches Spiel mit imperfekten Informationen: Blackjack in der Game-Learning-Umgebung GBG](#)“, Bachelorarbeit, TH Köln, 2021

[Zeh2021a] Zeh, Tim: „Implementierung von Poker im General Board Game Framework“, Projektarbeit, TH Köln, 2021

[Zeh2021b] Zeh, Tim: „Untersuchung von allgemeinen KI-Agenten für das Spiel Poker im General Board Games Framework“, Master Thesis, TH Köln, 2021.

[Fabig2021] Fabig, Niklas: „[Portierung einer Java-Reaktorsimulation nach Python und Performanz-Vergleich mit verschiedenen Python-Java-Bridges](#)“, Projektarbeit, TH Köln, 2021 ([lokale Kopie](#))

[OpenAIBlackJack] https://github.com/openai/gym/blob/master/gym/envs/toy_text/blackjack.py
OpenAI Gym's simple, yet effective Blackjack environment

[SB3] <https://stable-baselines3.readthedocs.io/en/master/>: [Stable Baselines3 \(SB3\)](#) is a set of reliable implementations of reinforcement learning algorithms in PyTorch.