### **Hands-On Projekt**

### **a): Wahl und Beschreibung der RL Umgebung**

Finanzmärkte sind seit langem das Fundament der Weltwirtschaft, wobei der Aktienhandel als ein Bereich hervortritt, der das Interesse sowohl von Einzel- als auch von institutionellen Anlegern weckt. Die letzten Jahrzehnte haben einen tiefgreifenden Wandel in den Handelspraktiken erlebt, insbesondere durch das Aufkommen von algorithmischen und Hochfrequenzhandelsparadigmen. Diese Transformation unterstreicht die wachsende Nachfrage nach intelligenten Systemen, die in der Lage sind, informierte Handelsentscheidungen in Echtzeit zu treffen. Auch wenn historische Aktiendaten keine garantierte Prognose für zukünftige Marktbewegungen bieten, stellen sie eine pragmatische Plattform dar, um die Stärken und Schwächen eines Handelsalgorithmus zu bewerten. Es stellt eine intellektuelle Herausforderung dar, zu versuchen, einen Markt zu entschlüsseln, der von einer Vielzahl von unvorhersehbaren Faktoren beeinflusst wird, von geopolitischen Ereignissen bis hin zu unternehmensspezifischen Nachrichten. Um diesem Bedarf gerecht zu werden, führen wir *CustomStockTradingEnv* ein, eine speziell für den Aktienhandel auf Basis historischer Daten konzipierte Umgebung. Das *CustomStockTradingEnv* scheint besonders für spezifische Anwendungsfälle geeignet zu sein, in denen Einzelaktien mit einer begrenzten Menge an Daten und technischen Indikatoren gehandelt werden sollen. Es bietet eine direkte und unkomplizierte Möglichkeit, RL-Modelle zu trainieren und zu testen, ohne sich mit den Nuancen eines umfangreicheren Frameworks auseinandersetzen zu müssen. *FinRL* hingegen wäre für Anwendungsfälle, wie das Handeln von Portfolios oder das Implementieren komplexerer Handelsstrategien, besser geeignet. Es bietet mehr Flexibilität und Funktionalität, kann aber auch komplexer in der Einrichtung und Verwendung sein. Da es in diesem Kontext nicht um Handelsstrategien oder Portfolioallokation geht, liegt es nahe, das *FinRL* Framework nicht zu nutzen.

**b): Entwicklung des RL-Agenten**

Ein charakteristisches Merkmal der Methode ist die Nutzung eines kontinuierlichen Aktionsraums, definiert im Bereich von -1 bis 1. Jeder Wert in diesem Aktionsraum repräsentiert den Prozentsatz des Portfolios, den der Agent beabsichtigt zu kaufen oder zu verkaufen. Diese Wahl eines kontinuierlichen Raums ermöglicht fein abgestimmte Handelsentscheidungen und bietet dem Agenten eine größere Flexibilität im Vergleich zu diskreten Handlungsoptionen. Das zentrale Element dieser Forschung ist die innovative Gestaltung der Belohnungsfunktion. Anstelle der konventionellen Methode, bei der Agenten für erzielte Gewinne belohnt werden, wurde die Belohnungsfunktion so konzipiert, dass sie Agenten bestraft, wenn ein Verlust im Portfolio-Wert auftritt. Bei einem Anstieg oder einer Konstanz des Portfolio-Werts werden weder Belohnungen noch Strafen vergeben. Diese spezifische Ausgestaltung der Belohnungsfunktion zielt darauf ab, den Agenten zu konservativen Handelsentscheidungen zu ermutigen und gleichzeitig das Eingehen unnötiger Risiken zu minimieren. Zur Implementierung und zum Training des Agenten wurden die RL-Algorithmen A2C und PPO verwendet. Beide Algorithmen haben sich in der Literatur als effizient im Umgang mit kontinuierlichen Aktionsräumen erwiesen.

**c): Bewertung und Visualisierung**

**Regularisierung**   
Die Einführung einer Regularisierung durch Transaktionskosten hatte den Effekt, dass der Agent konservativer im Handel wurde, um Overtrading und damit verbundene Kosten zu vermeiden. Dies kann jedoch dazu führen, dass der Agent potenziell profitable Handelsmöglichkeiten verpasst. Während das Ziel der Regularisierung darin besteht, eine realistischere Handelssimulation zu schaffen und Risiken zu minimieren, kann es auch unerwünschte Nebenwirkungen haben, wie eine verringerte Gesamtrendite. Es ist wichtig, das Gleichgewicht zwischen Risikominimierung und Rentabilität zu finden und die Regularisierungsparameter entsprechend anzupassen.

**Aktionsraum verfeinern.**Die Änderung des Aktionsraums im Sinne der Diskretisierung hat als möglicher Verbesserungsansatz für die Performance des Agenten bei der Umsetzung leider nicht geklappt.

**Trainingszeitraum**  
Im Zuge der Verbesserung des Trainings, wurde der Beobachtungszeitraum verlängert auf 2010-01-01 bis 2023-01-01 verlängert. Die Evaluierung wurde von 2023-01-02 bis heute angesetzt. Dies erwies sich als kleinere Verbesserung in der Gesamt-Performance.

**Anpassung Rewardfunktion**Die überarbeitete Reward-Funktion bietet eine verfeinerte Herangehensweise an die Bewertung von Handelsentscheidungen. Während die ursprüngliche Funktion lediglich den absoluten Gewinn zwischen zwei aufeinanderfolgenden Zeitpunkten betrachtete, berücksichtigt die neue Funktion zusätzlich den Drawdown, also den relativen Rückgang vom Höchststand des Portfolios. Dies erlaubt es, Risiken besser zu bewerten, da hohe Drawdowns mit einer Strafe belegt werden. Durch diese Dualität – Belohnung für Gewinne und Bestrafung für Risiko – wird der Agent dazu angeregt, sowohl rentable als auch risikoarme Handelsstrategien zu verfolgen.

**Nicht umgesetzte Verbesserungen**Die fortlaufende Optimierung unseres Agenten für den Handel von Einzelaktien bietet zahlreiche Ansatzpunkte. Erweitertes Feature Engineering kann dabei helfen, Marktdaten nuancierter zu repräsentieren und so die Erkennung von Mustern zu verfeinern. Ein systematisches Tuning von Hyperparametern, gegeben die Empfindlichkeit von RL-Modellen, kann erhebliche Auswirkungen auf die Performance haben. Dabei könnte auch der Einsatz alternativer RL-Modelle wie DDPG oder A3C in Betracht gezogen werden. Eine Erweiterung der Trainingsdaten, die verschiedene Marktzyklen abdeckt, fördert eine allgemeinere Strategiebildung. Ergänzt durch eine periodische Neu-Trainierung bleibt der Agent aktuell und anpassungsfähig. Schließlich kann das Training in diversen Umgebungen die Robustheit des Agenten steigern.

**Quellen**

<https://github.com/Chubbyman2/reinforcement-learning-stock-trader/tree/main>

<https://chat.openai.com/>

<https://www.gymlibrary.dev/content/basic_usage/>