Hochschule Düsseldorf
University of Applied Sciences

Learning to Cooperate: Reinforcement Learning in the Iterated Prisoners Dilemma

Jonah Gräfe

Das iterierte Gefangenendilemma

Das wiederholte Gefangenendilemma (Iterated Prisoner's Dilemma, IPD) ist ein klassisches Problem der Spieltheorie, das die Spannung zwischen Kooperation und Wettbewerb in wiederholten Interaktionen modelliert. In jeder Runde entscheiden sich zwei Spieler unabhängig voneinander entweder für die Kooperation (C) oder für die Defektion (D). Ihre Entscheidungen bestimmen ihre Auszahlungen auf der Grundlage einer Auszahlungsmatrix:

Player A/B	Cooperate (C)	Defect (D)
Cooperate (C)	(3, 3)	(0, 5)
Defect (D)	(5, 0)	(1, 1)

Table 1. Auszahlungsmatrix

Basis-Strategien

Strategie	Beschreibung
Random	Entscheidet zufällig
AlwaysCooperate	Kooperiert immer
AlwaysDefect	Defektiert immer
Provocative	Defektiert nach 2-mal kooperieren
TitForTat	Imitiert den Gegner
TitForTwoTats	Defektiert, wenn Gegner 2-mal defektiert
TwoTitsForTat	Defektiert 2-mal, wenn Gegner defektiert
TitForTatOppesite	Imitiert den Gegner gegenteilig
Spiteful	Defektiert für immer, wenn Gegner defektiert
GenerousTitForTat	Imitiert den Gegner aber vergibt zu 10%
Adaptive	Defektiert, wenn Gegner >50% in den letzten 10 Runden defektiert hat
Pavlov	Defektiert, wenn Entscheidungen in vorheriger Runde unterschiedlich waren
Gradual	Defektiert genauso oft wie Gegner defektiert
WinStayLoseShift	Ändert Strategie wenn reward < 1
SoftMajority	Defektiert, wenn Gegner >50% defektiert

Dazu kommen "Suspicious"-Strategien, als erstes defektieren. So kommt man auf insgesamt 24 Strategien.

Deep Q-Learning

Q-Learning-Agenten

1. **Q-Werte:** Der Q-Wert stellt die erwartete kumulative Belohnung für das Ausführen einer Aktion in einem bestimmten Zustand und das anschließende Befolgen der optimalen Strategie dar.
2. **Aktualisierung:**

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_a Q(s',a) - Q(s,a)]$$

- s, a, s': Aktueller Zustand, durchgeführte Aktion und der nächste Zustand.
- r: Erhaltene Belohnung
- α: Lernrate
- γ Diskontierungsfaktor

Deep Q-Learning-Agenten

1. **Neuronales Netz:** Bildet Zustände auf Q-Werte für alle möglichen Aktionen ab. Dies ersetzt die Q-Tabelle.
2. **Erfahrungswiedergabe:** Der Agent speichert vergangene Erfahrungen in einem Wiederholungspuffer.
3. **Training**: Minimierung des MSE zwischen vorhergesagten Q-Werten und Ziel Q-Werten mit Hilfe Adam-Optimierers

Q-Netzwerk

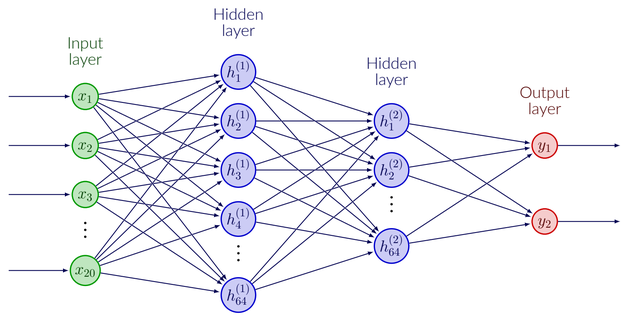

Figure 1. Q-Netzwerk mit n Input, 64 und 32 Hidden und 2 Output Neuronen.

Zufällige Strategien

In folgenden Darstellungen (Figure 2-5) wurden 10000 Epochen, bestehend aus jeweils einem Spiel mit 100 Runden gegen eine zufällige aber feste Reihenfolge von Basis-Strategien gespielt.

RandomStrategies

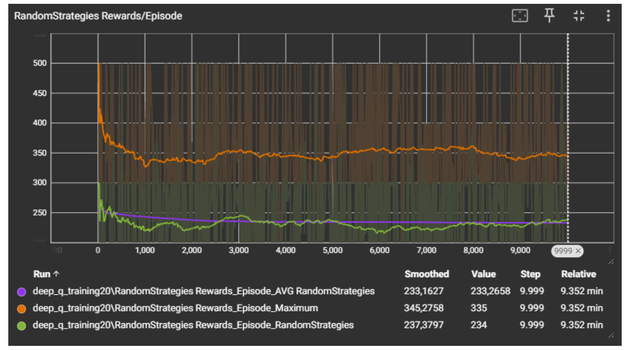

Figure 2. Geglättete Ergebnisse der zufälligen Strategien. Lila: Durchschnittsbelohnung, Orange: Maximum, Grün: Belohnung

Q-Learning Training

QLearningAgent ($\alpha = 0.01, \gamma = 0.5$)

$$Q = \begin{bmatrix} 6.0 & 3.99581385 \\ 7.99162798 & 4.22964461 \end{bmatrix}$$

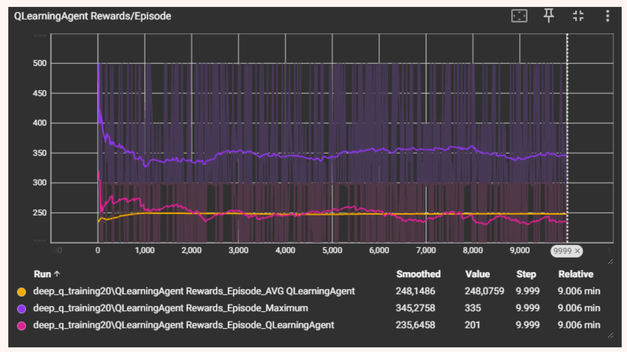

Figure 3. Geglättete Ergebnisse des QLearningAgents. Gelb: Durchschnittsbelohnung, Pink: Belohnung, Lila: Maximum

Deep Q-Learning Training

DeepQLearningAgent ($\alpha = 0.001, \gamma = 0.99$)

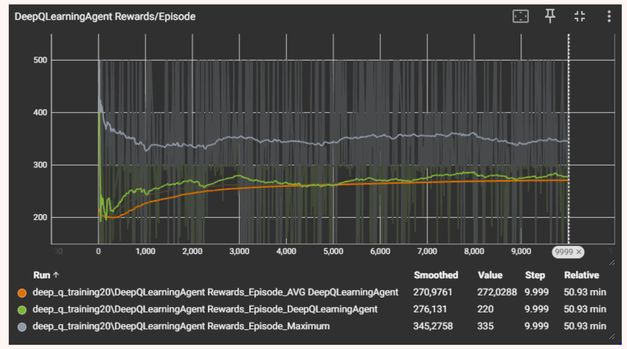

Figure 4. Geglättete Ergebnisse des DeepQLearningAgents. Orange: Durchschnittsbelohnung, Grün: Belohnung, Grau: Maximum

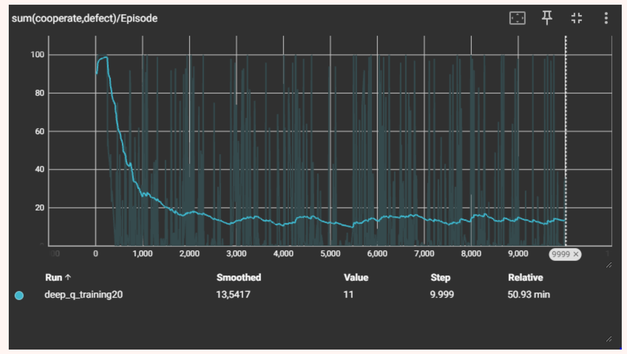

Figure 5. Geglättete Summe von Kooperation (0) und Defektion (1) in jeder Epoche

Turnier-Ergebnisse

Jeder der folgenden Agenten hat in einem Turnier 100-Mal gegen jede Basis-Strategie gespielt:

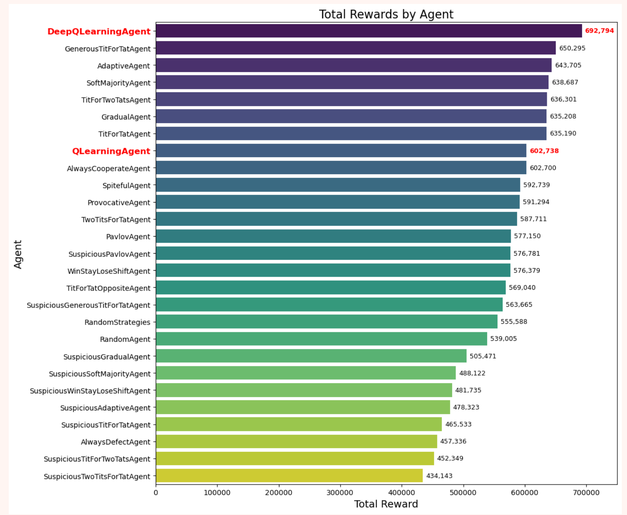

Figure 6. Barplot der Gesamtbelohnungen aller Agenten