Duale Hochschule Baden-Württemberg Mannheim

Reinforcement Learning Hands-On-Projektarbeit

Training einer KI in der Frozen Lake Umgebung

Studiengang Wirtschaftsinformatik

Studienrichtung Data Science

Verfasser(in):	Thi Quynh Anh Vu
Matrikelnummer:	1039624
Kurs:	WWI-20-DSB
Studiengangsleiter:	Prof. Dr. Bernhard Drabant
Abgabedatum:	30.07.2023

Ehrenwörtliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Titel "*Training einer KI in der Frozen Lake Umgebung*" selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Mannheim, den 30.07.2022 Thi Quynh Anh Vu

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

A	Action
F	Frozen
S	Start
S	State
G	Goal
H	Holes

1 Einführung in die Frozen-Lake-Umgebung

Die Arbeit beschäftigt sich mit der sogenannten Frozen-Lake-Umgebung. Das Ziel besteht darin, einer Künstlichen Intelligenz beizubringen, wie sie die Frozen-Lake-Umgebung mithilfe von Reinforcement Learning lösen kann. Diese Umgebung ist ein Teil von Toy-Text-Umgebungen.[1]

Die vorliegende Umgebung umfasst die Aufgabe, einen zugefrorenen See (engl. **Frozen (F)** Lake) vom **Start (S)** zum Ziel (engl. **Goal (G)**) zu überqueren, ohne dabei in Löcher (engl. **Holes (H)**) zu fallen. Der Agent hat dafür vier mögliche Aktionen: LINKS, UNTEN, RECHTS oder OBEN. Hierbei muss der Agent erlernen, Lücken zu vermeiden, um mit möglichst wenigen Aktionen zum Ziel zu gelangen. Außerdem wird der Agent erst belohnt, wenn der Zustand G erreicht ist. Allerdings kann es aufgrund der glatten Oberfläche des Frozen-Lakes passieren, dass sich der Agent nicht immer in die vorgesehene Richtung bewegt [vgl. 1].

Die Observation repräsentiert die aktuelle Position des Agenten. Die Anzahl der möglichen Observationen ist von der Größe der Karte anhängig. In diesem Fall wird die Größe der Karte, auch als Frozen-Lake bezeichnet, als 7x7 definiert. Daher gibt es insgesamt 49 mögliche Observationen. Im Rahmen dieses Projekts wird der Agent allerdings nur auf die Version von `is_slippery = False` trainiert. Detaillierter Code befindet sich im Github-Repository unter dem folgenden Link https://github.com/anhvu202/Hands-On-Projekt-RL.

[1] Dokumentation unter https://gymnasium.farama.org/environments/toy_text (besucht am 19.07.2023)

2 Q-Learning und Epsilon-Greedy-Algorithmus

Um eine Strategie im Frozen-Lake zu erlernen, die den Gesamtgewinn maximiert, wird das *Q-Learning* als Lernalgorithmus im Rahmen des Projekts verwendet. Das „Q“ im Q-Learning steht für *Qualität* und es beschreibt, wie nützlich eine bestimmte Aktion für den Erhalt einer zukünftigen Belohnung ist. Zu Beginn des Q-Learning-Prozesses wird eine *Q-Tabelle* erstellt, welche die Form **[State (S), Action (A)]** hat und deren Werte auf Null initialisiert werden. Die *Q-Werte* werden nach einer Episode aktualisiert und gespeichert. Diese Q-Tabelle dient als Referenztabelle für die Agenten, um die beste Aktion, basierend auf dem Q-Wert, auszuwählen. Der Q-Wert für das Feld des aktuellen Zustands (S) und der Aktion (A) wird mithilfe der folgenden Formel aktualisiert [vgl. 3]:

$$Q(S_t, A_t) = (1 - \alpha).Q(S_t, A_t) + \alpha.(R_t + \lambda.max_{\alpha}Q(S_{t+1}, \alpha)) \tag{2.1}$$

Ein Agent interagiert auf zwei Arten mit seiner Umgebung: Ausbeutung (engl. Exploiting) oder Erkundung (engl. Exploring). *Exploiting* bedeutet, dass der Agent basierend auf den zur Verfügung stehenden Informationen eine Entscheidung trifft. Hierbei nutzt er die Q-Tabelle, um alle potenziellen Aktionen für einen gegebenen Zustand anzuzeigen. Anschließend wählt er die Aktion mit dem höchsten Q-Wert aus. Im Gegensatz dazu bedeutet *Exploring*, dass der Agent Aktionen nach dem Zufallsprinzip auswählt, anstatt sich auf die maximale zukünftige Belohnung zu konzentrieren. Dieses zufällige Handeln ist wichtig, da es dem Agenten ermöglicht, neue Zustände zu erkunden und zu entdecken, welche möglicherweise nicht durch einen Exploitingsprozess ausgewählt werden würden. Die Balance zwischen Exploiting und Exploring wird mithilfe von Epsilon (ϵ) eingestellt. Dieser Wert bestimmt, wie oft der Agent die Umgebung erkundet bzw. Ausnutzung bevorzugt [vgl. 3]. Die Technik wird allgemein als *Epsilon-Greedy-Algorithmus* bezeichnet. Jedes Mal, wenn der Agent eine Aktion ausführen muss, hat er eine Wahrscheinlichkeit ϵ, eine zufällige Aktion auszuwählen und eine Wahrscheinlichkeit von $1 - \epsilon$, die Aktion mit dem höchsten Wert auszuwählen. Der Wert von ϵ kann sich am Ende jeder Episode um einen festen Beitrag basierend auf dem linearen Abfall oder basierend auf dem aktuellen Wert von ϵ (exponentieller Abfall) reduzieren [vgl. 2].

3 Evaluation

Nach dem Training des Q-Learning-Algorithmus in Verbindung mit dem Epsilon-Greedy-Algorithmus lässt sich erkennen, dass die Aktionen UNTEN und RECHTS am häufigsten ausgewählt wurden. Dieses Verhalten ist nachvollziehbar, da der Agent seinen Weg von der oberen linken Ecke auf der Karte zum unteren rechten Ziel finden muss. Um zu überprüfen, ob der Agent gelernt hat, wird die kumulierte Summe der Belohnungen sowie die Anzahl der Schritte, die bis zum Ende der Episode erforderlich sind, grafisch dargestellt (siehe Abbildung 3.1).

Die Abbildung zeigt einen klaren Anstieg der kumulierten Belohnungen mit zunehmenden Trainings, während die Anzahl der Schritte zur Lösung der Aufgabe abnimmt. Der Agent benötigte etwa 300 Episoden, um das Lernen zu konvergieren. Dies bedeutet, dass er nach dieser Anzahl von Trainingsdurchläufen eine stabile und effektive Handlungsstrategie entwickelt hat.

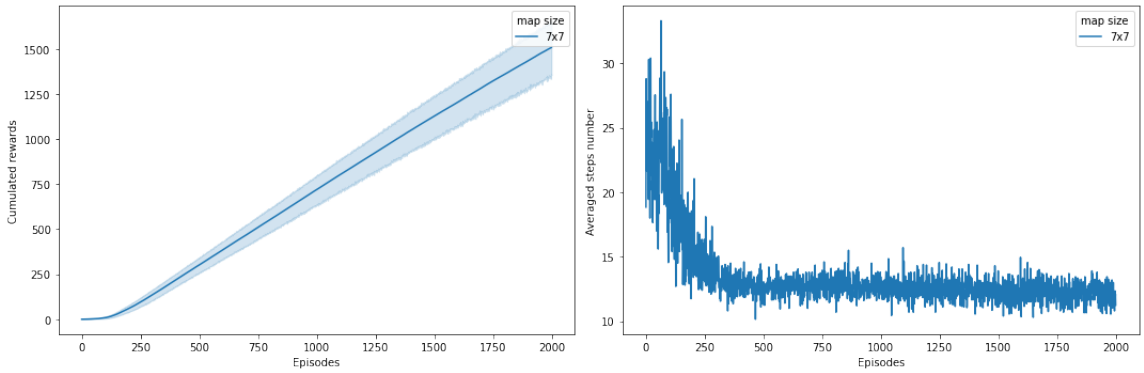

Abbildung 3.1: Die kumulierte Summe der Belohnungen sowie die Anzahl der Schritte während des Trainings
Quelle: Eigene Darstellung

4 Fazit

Q-Learning ist ein grundlegender Algorithmus im Bereich des Reinforcement Learning. Dazu wird ein Kompromiss zwischen der Erforschung unbekannter Zustands-Aktionspaare und der Ausbeutung der besten Handlungen durch die Implementierung des Epsilon-Greedy-Algorithmus erzielt.

Die Evaluation zeigt, dass der Agent über eine Fähigkeit zur Anpassung und Verbesserung seiner Handlungsstrategien verfügt. Es ist wichtig, diese Erkenntnisse zu nutzen, um die Leistung des Agenten weiter zu optimieren sowie mögliche Anwendungsfelder zu erkunden, in denen der Agent seine erlernten Fähigkeiten erfolgreich einsetzen kann. Weitere Forschung könnte darauf abzielen, den Lernprozess noch effizienter zu gestalten und die Konvergenzzeit weiter zu verkürzen. Zusätzlich wäre es interessant, das Training auf einer glatten Oberfläche durchzuführen, um zu prüfen, wie sich die erlernten Strategien in realen, weniger idealisierten Umgebungen verhalten.

Literaturverzeichnis

[1] *Frozen Lake - Gym documentation*. URL: https://www.gymlibrary.dev/environments/toy_text/frozen_lake/ (besucht am 19. 07. 2022).

[2] Maxime Labonne. „Q-learning for Beginners - towards data science“. In: (30. März 2022). URL: https://towardsdatascience.com/q-learning-for-beginners-2837b777741 (besucht am 26. 07. 2022).

[3] Andre Violante. „Simple reinforcement learning: Q-learning - towards data science“. In: (1. Aug. 2022). URL: https://towardsdatascience.com/simple-reinforcement-learning-q-learning-fcddc4b6fe56 (besucht am 26. 07. 2022).