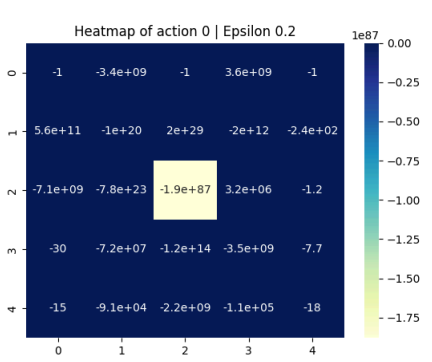
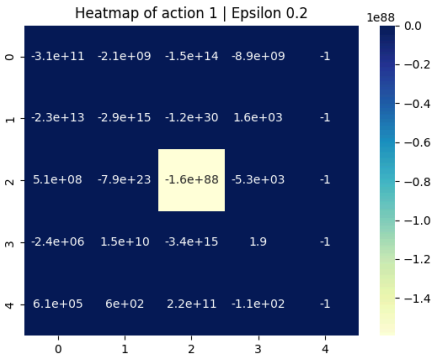
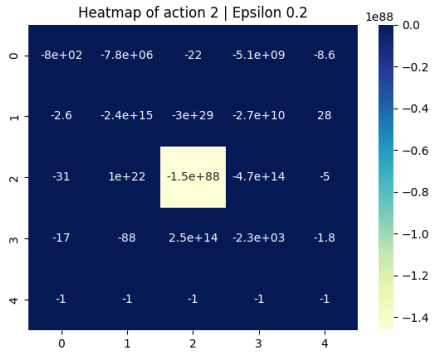
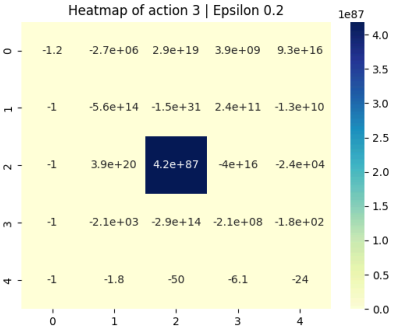
Aufgabe 4.1

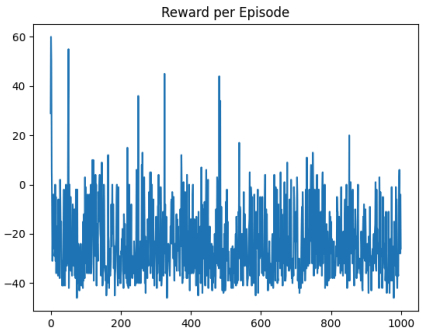
a) 1000 Episoden , Gamma = 0.9



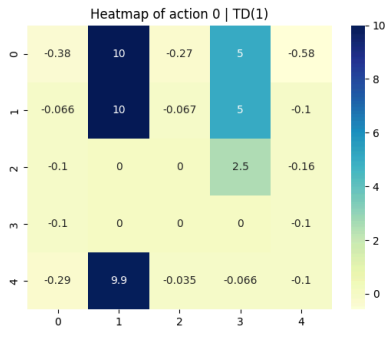


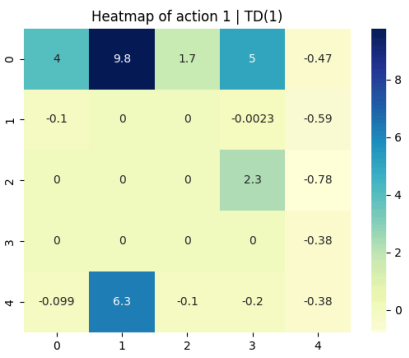


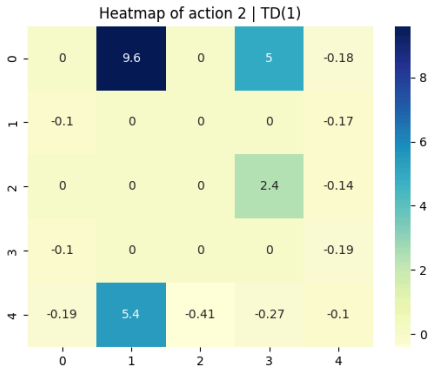


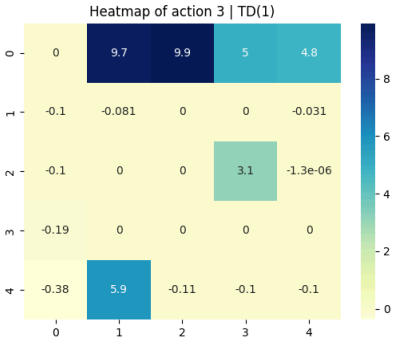


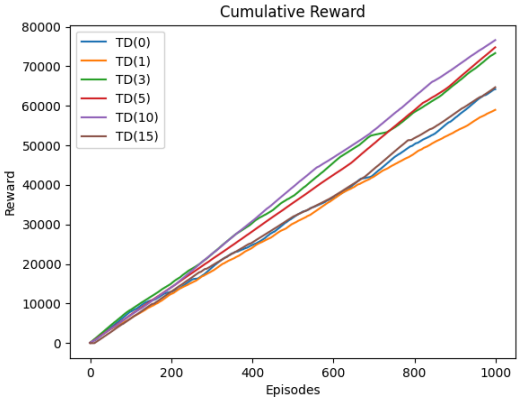
b) 1000 Episoden, Epsilon = 0.1, Alpha = 0.1

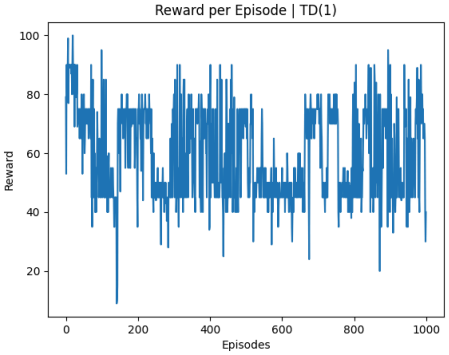




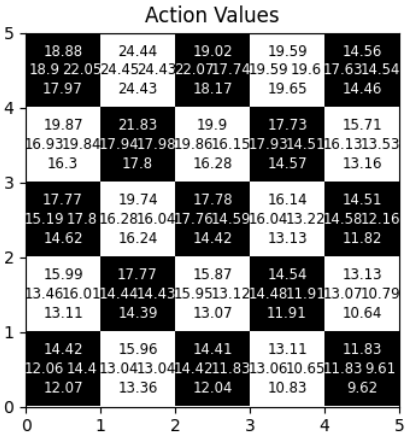


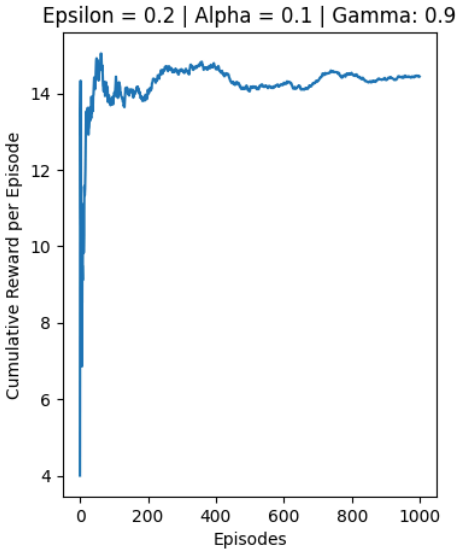






c)

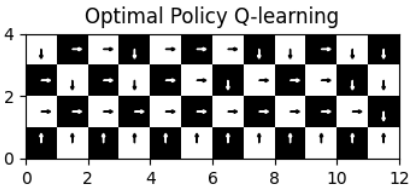


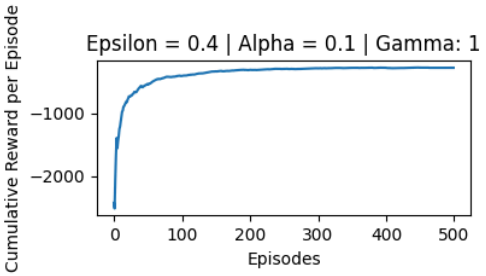


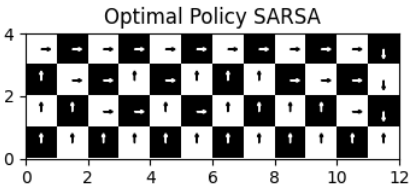
1.
Q-Learning wird als Off-Policy-Algorithmus kategorisiert, da die anpassung unserer action-values durch eine greedy-policy von unserem neuen state aus passiert, die wahl der Aktion die wir machen allerdings von einer anderen policy kommen kann wie z.B. epsilon-greedy.

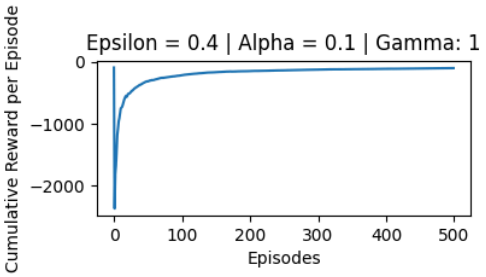
Ein Problem durch die Aktualisierung durch die greedy-policy besteht darin, dass wir womöglich, bei zu geringer Exploration, nicht oder nur langsam die optimale Policy finden, da unsere action-values immer in die momentan beste Richtung hin verbessert werden.

Aufgabe 4.2
a)









1.
On-Policy.Methoden:

Vorteile:

- Angepasst an konkrete Umgebung

Nachteile:

- Abhängig von Explorativer Starts

Szenario: Explorativer Start möglich

Off-policy-Methoden:

Vorteile:

- Nutzung von Explorativer Strategie während des lernens für die optimale Policy
- Nutzung bereits gemachter Erfahrungen von alten Policies

Nachteile:

- Große unterschiede zwischen Behavior Policy und Ziel Policy

Szenario:
Wenn Erfahrungsdaten von alten Policies vorliegen

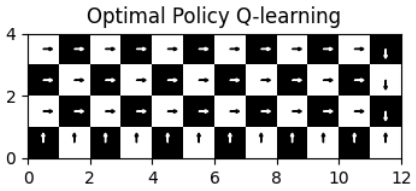
2.
Bei SARSA wird unsere momentane action-value basierend auf der action-value des neuen States angepasst, somit wird, in dem Fall dass wir in den Bereich der Klippe kommen,

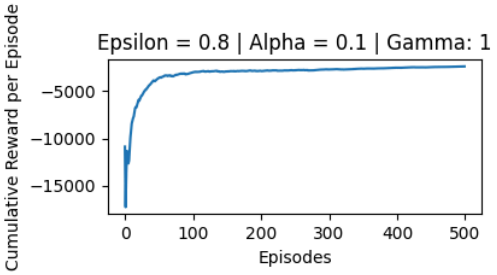
potentiell unsere nächste Aktion sehr negativen Reward bringen und somit die Aktion die uns in die nähe der Klippe bringen bestraft.
Dadurch ist es bei SARSA besser, weit weg von der Klippe zu bleiben und somit den negativen Reward nicht zu bekommen.
Beim Q-Learning werden die action-values basierend auf den besten Aktionen die wir im neuen State machen können angepasst, somit sind Fehler nicht so schlimm, da diese nicht die beste Aktion sind und somit die action-values nicht so stark beeinflussen.
Beim Q-Learning können wir von der Klippe stürzen, wenn diese Aktion vom "off-policy" Teil des Q-Learning ausgewählt wird, nicht jedoch vom gierigen updaten unserer action-values.

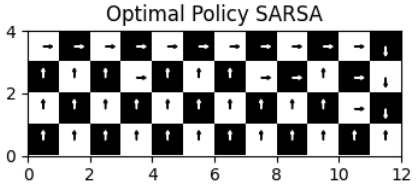
3.
Da wir bei Q-Learning näher an der Klippe vorbeilaufen, ist es Wahrscheinlicher dass wir in die Klippe stürzen, als bei SARSA wo wir weiter von der Klippe entfernt sind, da in beiden Fällen nur durch eine zufällig ausgewählte Aktion im epsilon-greedy Teil wir eine Aktion durchführen die uns in die Klippe bringt.

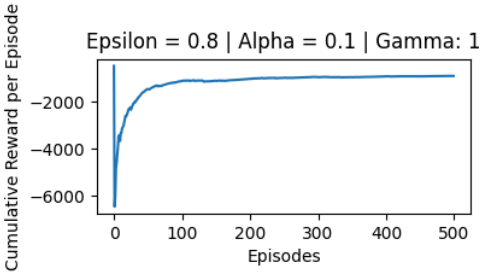
4.
Wenn im Q-Learning die Aktionen rein gierig ausgewählt werden, so bleibt Q-Learning trotzdem anders als SARSA. Bei SARSA werden die Aktionen ja epsilon-greedy ausgewählt was sich dann zu dem Q-Learning unterscheiden würde. Zudem wird bei SARSA basieren auf der ausgewählten Aktion die action-values verändert, was dann nicht notwendigerweise die optimalen sind wie es bei Q-Learning passiert.

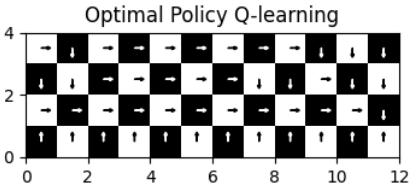
b)

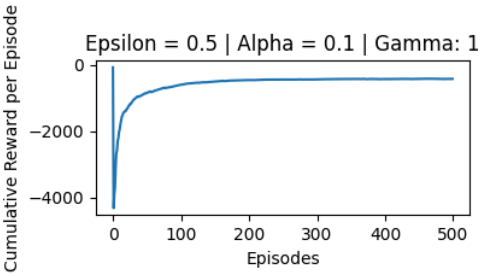


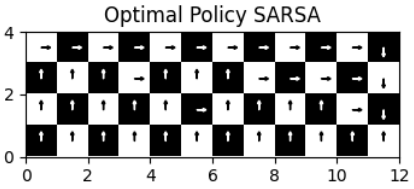


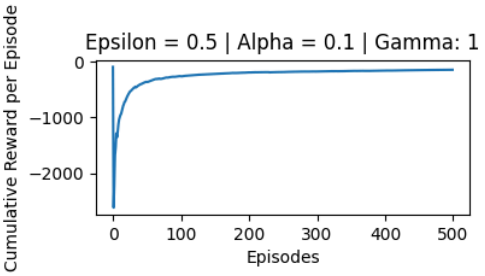


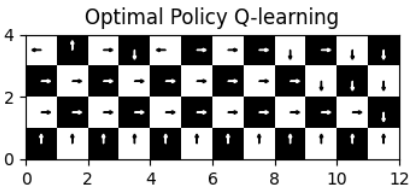


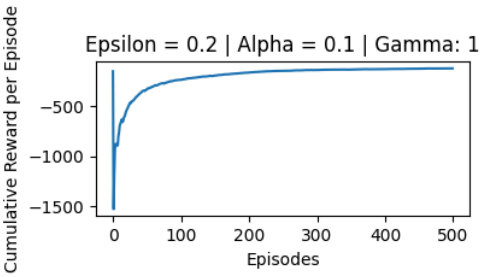

Optimal Policy Q-learning

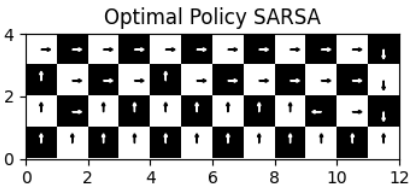

Epsilon = 0.5 | Alpha = 0.1 | Gamma: 1
Cumulative Reward per Episode
Episodes

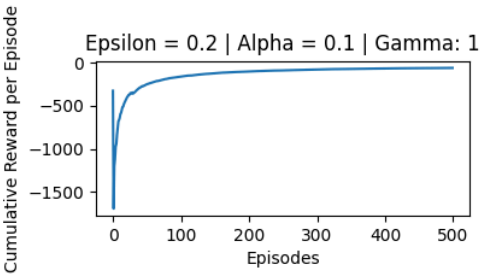

Optimal Policy SARSA

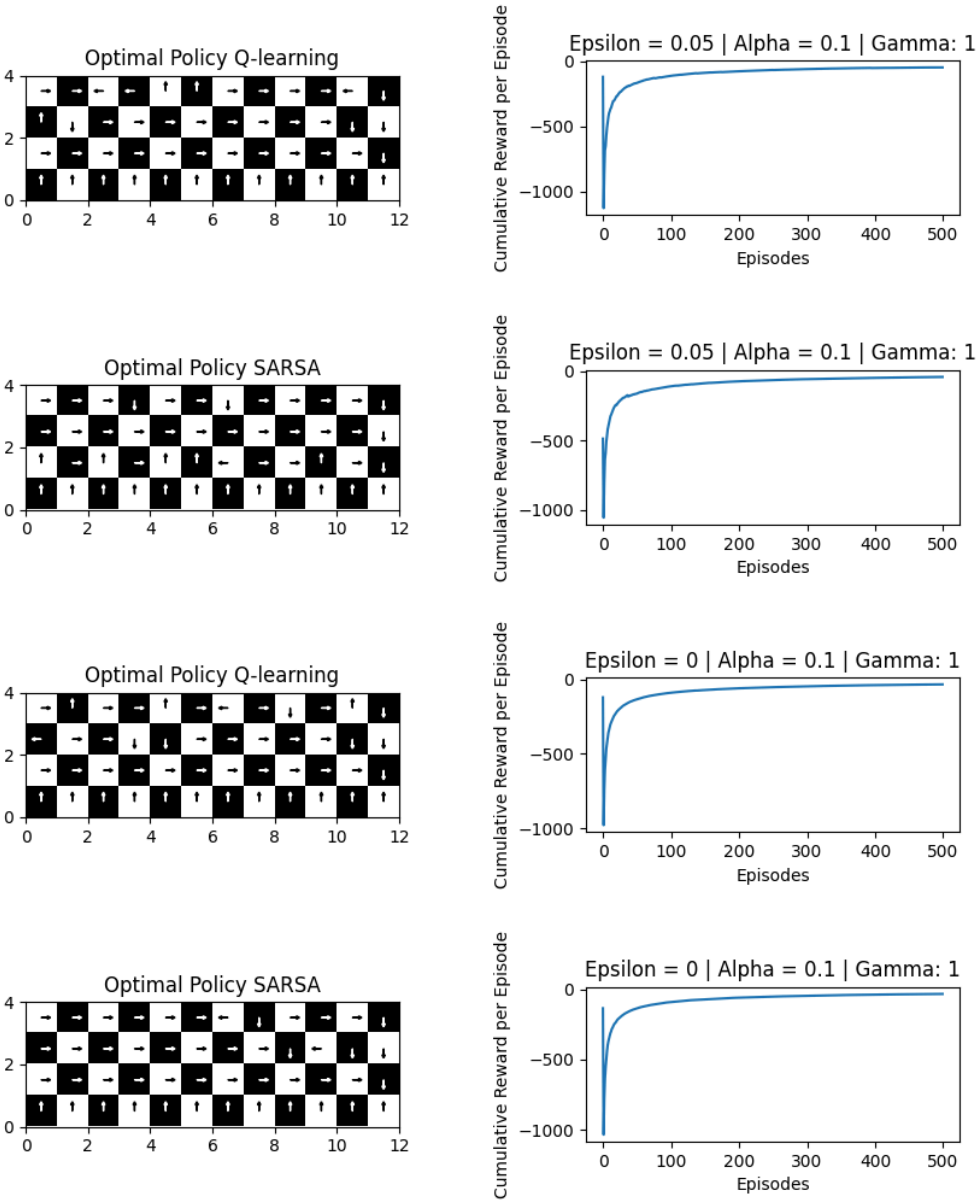

Epsilon = 0.5 | Alpha = 0.1 | Gamma: 1
Cumulative Reward per Episode
Episodes


Optimal Policy Q-learning


Epsilon = 0.2 | Alpha = 0.1 | Gamma: 1
Cumulative Reward per Episode
Episodes


Optimal Policy SARSA


Epsilon = 0.2 | Alpha = 0.1 | Gamma: 1
Cumulative Reward per Episode
Episodes



a)
Betrachtet wurden die Epsilon Werte 0.8, 0.5, 0.2, 0.05, 0.
Bei Epsilon > 0.05 wird der obere (sichere) Pfad von SARSA genommen.
Erst bei Epsilon = 0.05 verändert sich der Pfad von SARSA dazu, den Mittleren Weg zu gehen.
Bei Epsilon = 0 stimmen die Pfade der beiden Algorithmen überein, indem beide den optimalen Pfad nehmen.
Der Pfad bei Q-Learning blieb unverändert von den Epsilon Werten.
Durch niedrigere Epsilon Werte wird weniger erkundet und somit seltener in die Klippe gestürzt. Somit steigt der Reward bei fallendem Epsilon.

b)
Mit abnehmendem epsilon passt sich SARSA näher dem optimalen Pfad an, da durch die geringere Exploration es weniger wahrscheinlich wird in die Klippe zu stürzen und somit viel negativen Reward zu bekommen, wodurch Aktionen die in die nähe der Klippe führen bestraft werden, bei action-value update.

Da Q-Learning den optimalen Pfad unabhängig von Epsilon findet, führen größere Epsilon Werte dazu, dass wir eher in die Klippe fallen und somit einen negativen Reward bekommen.

c)
Eliminierung von Exploration:

Vorteil: höherer Reward (z.B. durch weniger in die Klippe fallen)

Nachteil: in Komplexeren Situationen wird nicht die beste Strategie gefunden, wenn es eine lokal bessere Strategie gibt

Modifikation / anderer Ansatz:

- epsilon mit der Zeit senken
- verschiedene Starts wählen
- Aktionen gewichtet wählen, basierend darauf wie häufig sie vorgekommen sind