Assignment 1

for **Deep Reinforcement Learning WS2024**

by
Kamil
000000
Fynn Castor
540055

the **20. Oktober 2024**

1 Excercise 1.

1.1 Excercise 1a.

It seems that the code is working. Due to the rather large pick of $\epsilon = 0.5$ the algorithm does not really take advantage of the early exploration and is around 60% of best arm chosen. Which falls in line with the expected limit for this value of $50\%+50\%\cdot\frac{1}{4} = 62.5\%$.

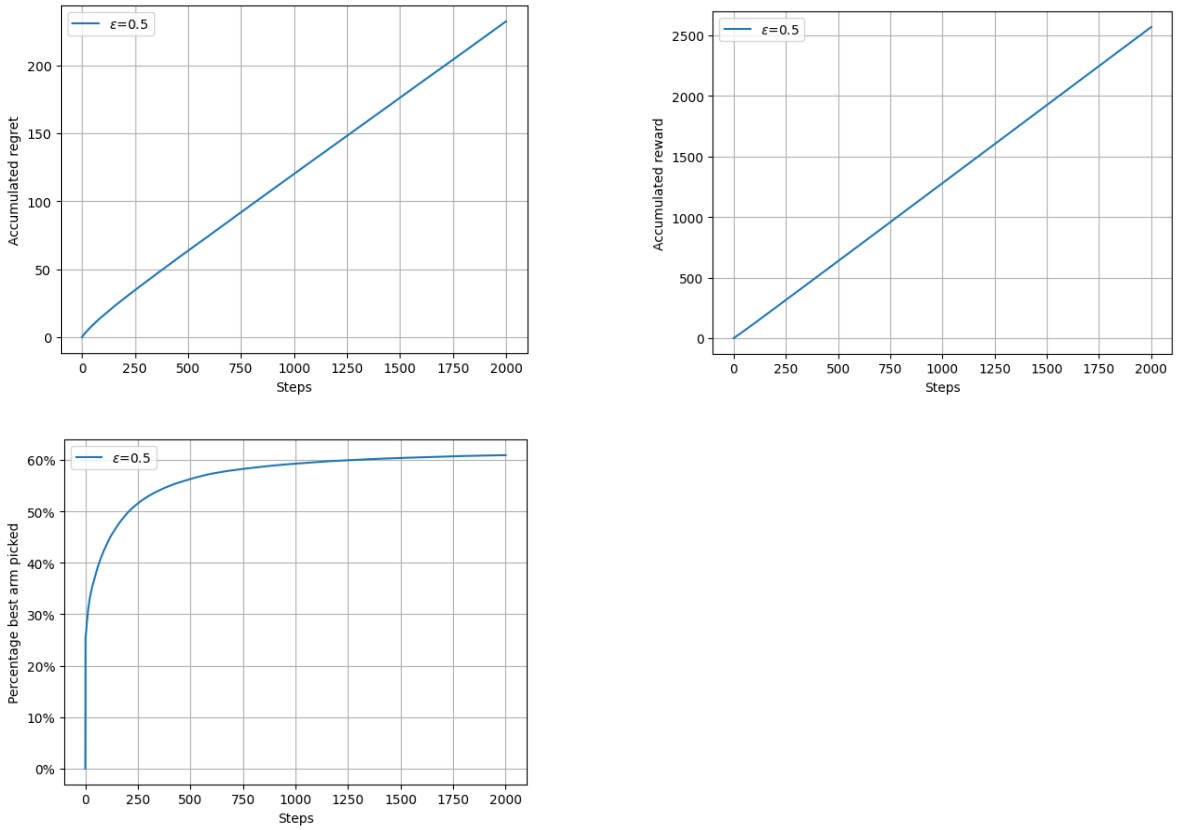

Abbildung 1: Regret, reward, and percentage of best arm chosen for 4-armed bandit with normal arm distribution, where $(\mu, \sigma) \in \{(1, 0.2), (1.2, 0.4)(1.1, 0.6), (1.4, 0.8)\}$, averaged of 1000 trials

1.2 Excercise 1b.

As said in the ex. 1a for $\epsilon = 0.5$ the algorithm can't really exploit it's early exploration due to half of the actions being picked at random. For $\epsilon = 0.01$ the algorithm suffers from the lack of early exploration and only manages to catch to the regret of $\epsilon = 0.5$ after 1500 steps, while still being outperformed by $\epsilon = 0.1$ and $\epsilon = 0.05$, which both seem to have an appropiate balance between exploaration and exploation, with $\epsilon = 0.1$ having slightly better performance.

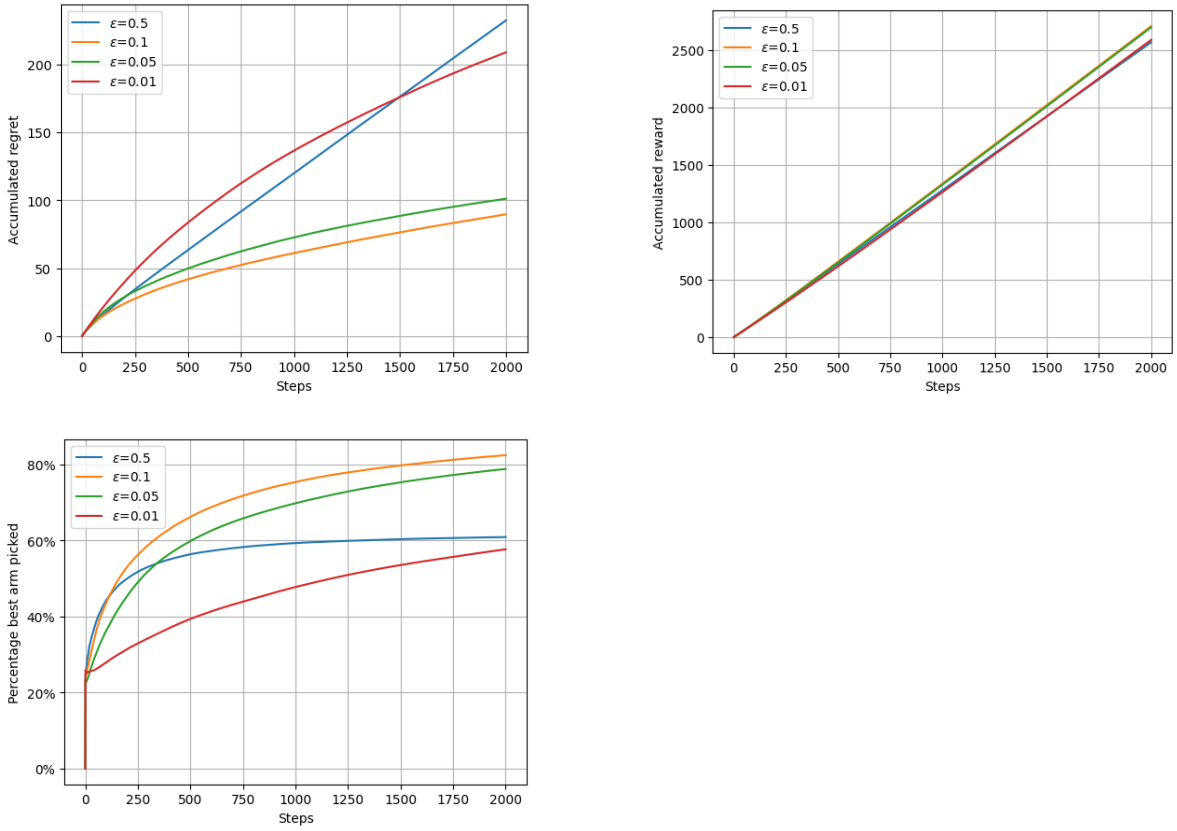

Abbildung 2: Regret, reward, and percentage of best arm chosen for 4-armed bandit with normal arm distribution, where $(\mu, \sigma) \in \{(1, 0.2), (1.2, 0.4)(1.1, 0.6), (1.4, 0.8)\}$, averaged of 1000 trials, with $\epsilon \in \{0.5, 0.1, 0.05, 0.01\}$

1.3 Excercise 1c.

Due to

$$Q_1(a) = Q_0(a) + \frac{1}{1}(R - Q_0(a)) = R$$

an overly optimistic estimate of the arm means taken as $Q_0(a)$ forces the algorithm to continue exploring during its greedy actions until all actions have been taken at least once. This in turn leads to some early exploration of each action, without having to make a tradeoff in regards to the choice of ϵ. Unfortunately due to the only small differnces in mean and the compared to that bigger σ's it does not have an immense impact

1.4 Excercise 1c.

As function a function for ϵ deterioration we chose the exponential function $f(n) = \epsilon_0 e^{-\lambda \cdot n}$. The parameter ϵ_0 hereby provides our starting ϵ whereass λ controls the speed of deterioration. We found this function on wikipedia after our initial idea of using the

multiplicative inverse of sigmoid function was not overly successful. As a benchmark we used the best perform $\epsilon = 0.1$ from ex 1b. One can see in the graphs that the choice of λ has an immense impact on the performance of the algorithm. If it's chosen too large the algorithm gets to greedy immediately and does not explore anymore. On the other hand a too small value does not exploit early enough. In the end our previous benchmark was quite clearly outperformed by $\lambda = 0.01$

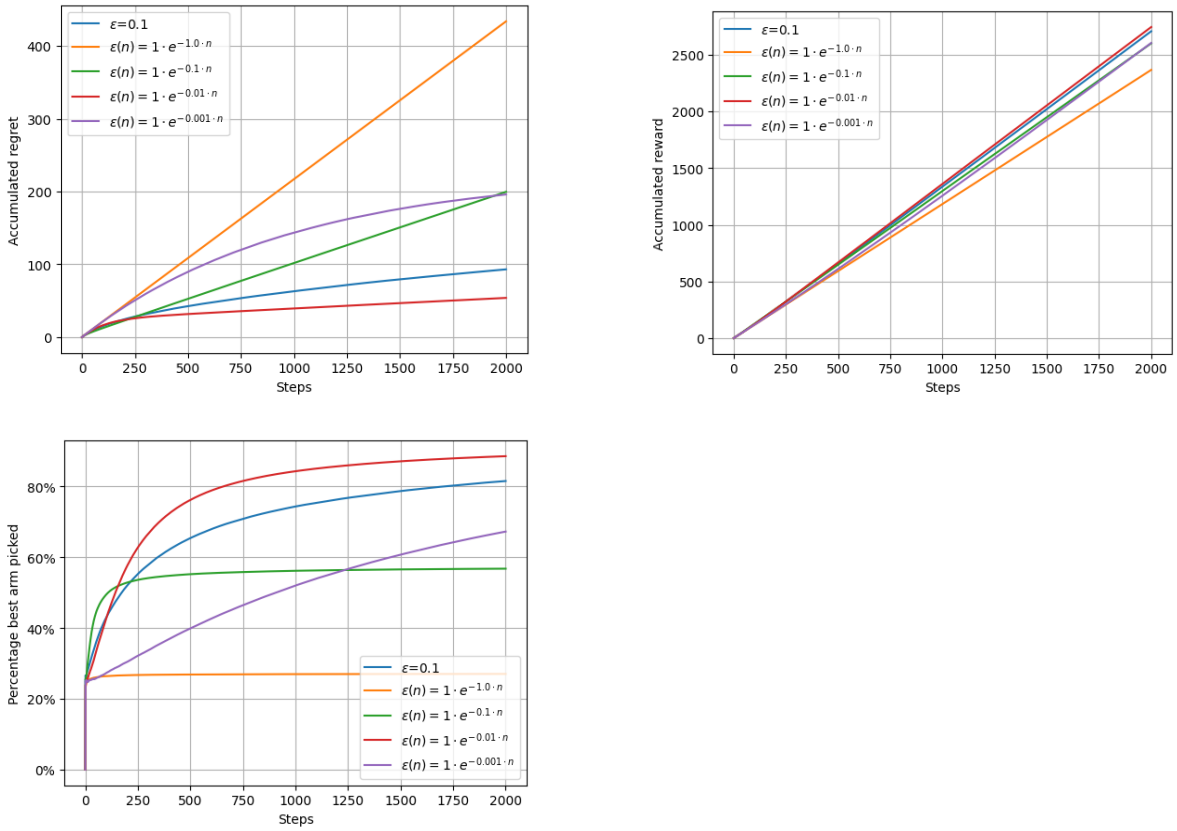

Abbildung 3: Regret, reward, and percentage of best arm chosen for 4-armed bandit with normal arm distribution, where $(\mu, \sigma) \in \{(1, 0.2), (1.2, 0.4)(1.1, 0.6), (1.4, 0.8)\}$, averaged for 1000 trials