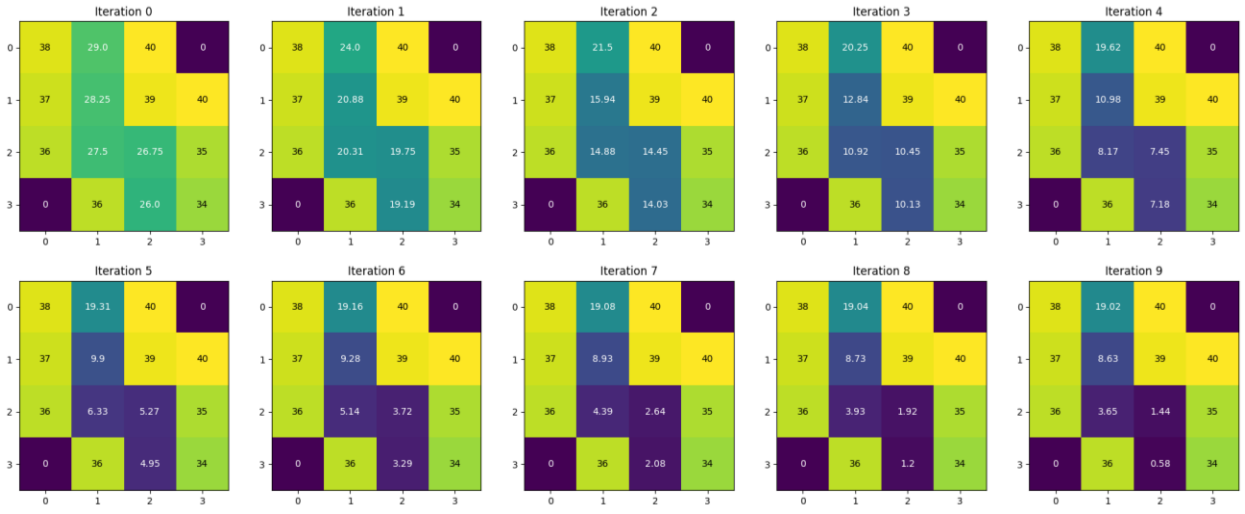
Opdracht 2.2
Adaptive Systems

Student:	Storm Joannes
Studentnummer:	1760581
Opleiding:	HBO-ICT Artificial Intelligence
Instelling:	Hogeschool Utrecht
Code:	2022_TICT_VINNO1-33_3_V
Datum:	14-01-2024

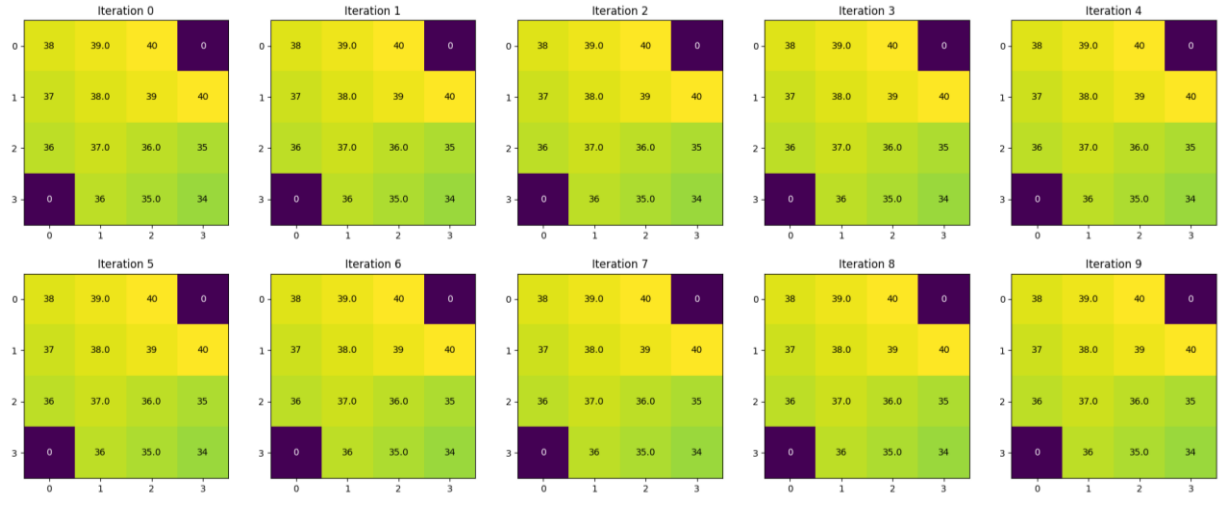
A. Temporal difference learning

De temporal differences worden berekend met 10 epochs, en een learning rate van 0,5.

De eerste figuur zijn de eind waardes van iedere episode temporal difference learning met een discount van 0.5. De episode werd beëindigd als de current position een terminal state was. Dit was in het geval van iedere iteratie positie (0, 3).



In de volgende figuur zijn de iteraties te zien van Temporal difference learning met een discount van 1. Ook hier eindigde iedere iteratie bij de terminal state (0, 3).

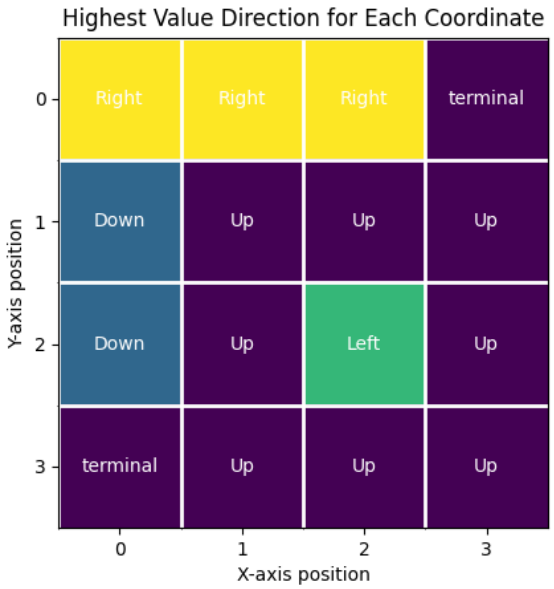


Het verschil tussen deze twee grafieken is duidelijk te zien door het verschil in waarden van de afgelopen posities. Het is duidelijk te zien dat de agent met de hogere discount veel sneller convergeert. Dit is te zien doordat de waardes in de plots vanaf de eerste iteratie al niet meer veranderen. In tegenstelling tot de agent met de lagere discount, neemt deze veel langer de tijd om te convergeren, en zoekt deze meer balans tussen de korte- en langetermijnbeloningen.

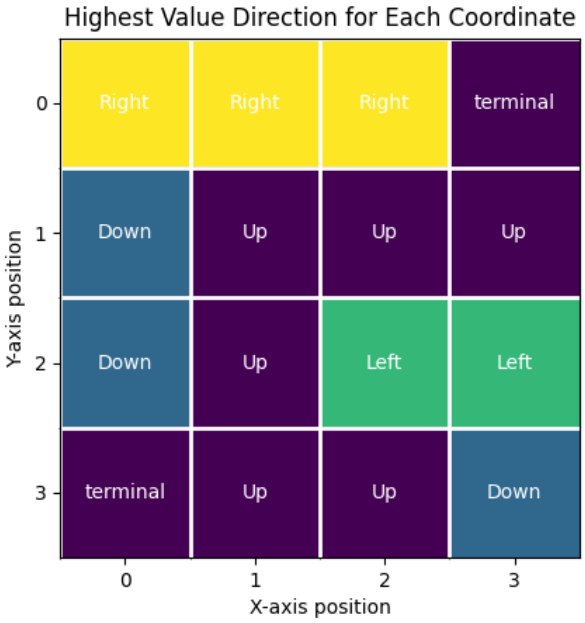
B. SARSA

De instellingen waarmee wij dit model runnen is een learning rate van 0.25, een epsilon van 0.1 en het aantal epochs van 20.

De eerste is discount 1:



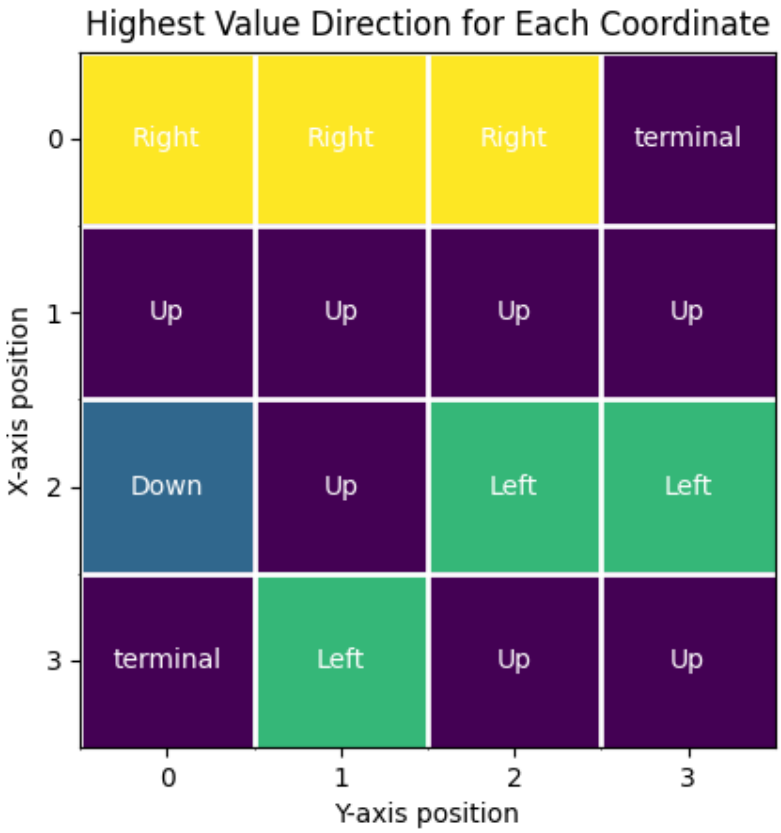
De volgende agent is een SARSA met een discount van 0,9. Hierin zie je een lichte verandering van acties die worden gekozen doordat de agent anders heeft geleerd. Echter kom je wel (op één positie na) altijd bij de terminal states uit.



Omdat het verschil in discount laag is, is ook het verschil tussen de twee plots niet heel groot.

C. Q-learning

We hebben hier Q-learning ookwel bekend als SARSAMAX toegepast met een learning rate van 0.25, een epsilon van 0.1 en het aantal epochs van 10.000, na dit aantal epochs veranderde de resultaten niet meer. Daarnaast waren de resultaten van een discount van 0.9 en 1 hierbij hetzelfde.
We hebben na de 10.000 epochs voor iedere positie de waardes van de omliggende posities bekeken waarna we een bepaling konden maken van welke stappen de agent zou nemen in de matrix. Hieruit is de volgde plot gekomen:



We kunnen hier zien dat de posities weergegeven in de plot acties weergeven om zich naar de terminal state te begeven. Ook zien we dat door Q-learning niet alle posities naar de langetermijnbeloning kijken, maar bijvoorbeeld op (2, 0) en (3, 1) er wordt gekozen voor de kortere route voor een beloning. Echter zien we tussen SARSA en Q-learning niet een heel groot verschil, doordat de twee ook best op elkaar lijken.

Om precies alle waardes voor iedere iteratie te bekijken kun je de extra images in de repo bekijken, of de code draaien.