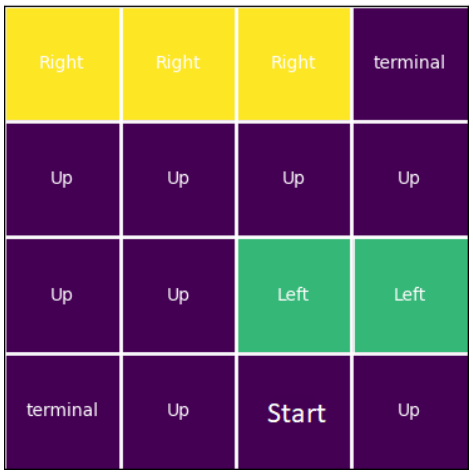
Opdracht 2.1
Adaptive Systems

Student:	Storm Joannes
Studentnummer:	1760581
Opleiding:	HBO-ICT Artificial Intelligence
Instelling:	Hogeschool Utrecht
Code:	2022_TICT_VINNO1-33_3_V
Datum:	22-01-2024

A. Monte-Carlo policy evaluation

i.

De optimale policy is als volgt:



Discount 1:

	38	40	
	37		
	36	35	
		34	

ii.

	19	40	
	8.5		
	3.25	0.63	
		-0.69	

iii.

Doordat je de optimale policy gebruikt, bepalen we momenteel de waarde van 6 states.
Dit kun je oplossen door een manier te bedenken waarop hij wel de andere states bekijkt om mogelijk betere actions te ontdekken. Dit kan bijvoorbeeld door een random actie te introduceren, waarbij de agent niet kijkt naar de waardes van de mogelijke acties, maar een random actie kiest.

iv.

Het gebruik van any-visit Monte-Carlo prediction in tegenstelling tot first-visit maakt zeker een verschil voor het resultaat. Bij first-visit Monte-Carlo wordt een staat alleen geüpdatet als het de eerste keer is dat die state wordt bezocht tijdens een episode. Bij any-visit wordt de state elke keer dat deze wordt bezocht geüpdatet. Hierdoor kan any-visit meer informatie bieden en mogelijk leiden tot nauwkeurigere uitkomsten.

B. Temporal difference learning

i.

Bij Monte-Carlo policy evaluation worden de returns berekend door het totale beloningspad vanaf het begin van een episode tot aan het einde (terminal state) te volgen. Omdat de returns direct worden waargenomen en toegevoegd aan de waarde van de bezochte staten, hoeven de terminal states niet te worden geïnitialiseerd. In tegenstelling hiertot berekent temporal difference learning de waarde-updates per stap, en daarom moeten de terminal states vooraf geïnitialiseerd worden, zodat de updates vanaf het eindpunt kunnen beginnen.

ii.

Voordeel:
Temporal difference learning kan online worden toegepast, wat betekent dat het model na elke stap kan worden bijgewerkt. Dit maakt het geschikt voor situaties waarin de volledige episode niet hoeft te worden afgewacht voordat het model wordt bijgewerkt.

Nadeel:
Temporal difference learning heeft de neiging om meer variabiliteit in schattingen te hebben dan Monte-Carlo-methoden. Omdat het slechts één stap vooruit kijkt, kan het gevoeliger zijn voor ruis en onzekerheden in de omgeving. Dit kan leiden tot minder stabiele schattingen in vergelijking met Monte-Carlo-methoden die het totale beloningspad overwegen.

C. On-policy first-visit Monte-Carlo Control

i.

De Q-function (in mijn code terug te vinden als ‘value_func’) berekent voor de gegeven omliggende states de nieuwe waarde. De berekening hiervan is, de reward, plus de discount, maal zijn huidige waarde. Hiermee valt de beste actie in de policy te bepalen.

De Q-functie wordt gebruikt in model-free control methoden, zoals Q-learning. Deze methoden leren optimale beslissingen te nemen zonder een volledig model van de omgeving te hebben. Bij het uitvoeren van deze methoden is het doel om de Q-waarde te maximaliseren over alle mogelijke acties voor een gegeven staat, dit is de verwachtte totale beloning van het kiezen voor een bepaalde staat. Hierdoor kan de beste actie volgens de huidige policy worden bepaald.

ii.

Bij model-free control, zoals bij Q-learning, heeft de agent geen gedetailleerd model van de omgeving met overgangsprobaliliteiten en beloningen. Hierdoor kan de agent de waarde van individuele states niet direct berekenen zoals bij model-based reinforcement learning met een specifieke value function. In plaats daarvan leert de agent rechtstreeks vanuit interactie met de omgeving, schattend en updatend op basis van waargenomen beloningen door middel van trial-and-error.

In het geval van de volgende opdracht, door de lunar lander simulatie een x aantal episodes uit te voeren zodat deze de omgeving kan leren kennen.

iii.

'On-policy' betekent dat het reinforcement learning-algoritme leert en beslissingen maakt op basis van de huidige policy. Met andere woorden, het verbetert zijn eigen policy terwijl het actief is en gebruikt de geüpdatete policy om acties te selecteren.

iv.

Ja, een deterministische policy kan een nadeel zijn in deze omgeving. In deze maze zijn er mogelijk meerdere wegen naar een doel, en als de agent altijd dezelfde acties kiest, kan het vastlopen in een suboptimaal pad. Een stochastische policy (waarbij de agent soms willekeurige acties kiest) zou de agent helpen verschillende wegen te verkennen en mogelijk betere oplossingen te vinden.