Opdracht voor logbook

- Zoek en annoteer eenvoudig reinforcement learning voorbeeld
  - ❑Beschrijf kort environment, toestanden, acties, observaties en beloning/optimalisatie
  - ❑Annoteer Q-learning methode of een variant daarvan

Self-driving Taxi

Doel
Het doel van de self-driving taxi is om klanten van op te halen en van de ene locatie naar de andere locatie te brengen.

Environment & State
Het environment bestaat uit een 5x5 grid met wat obstakels. De taxi moet klanten van naar vier verschillende locaties kunnen brengen: R, Y, B en G (zie de afbeelding hieronder).

Bij elke zet die een taxi zet, veranderd de state. De huidige state is hierboven te zien. De taxi bevindt zich op de coördinaten (3,1). Aan de westerse kant is er een muur. Bij elke state-verandering horen de keuzes van de actions dus beperkt te zijn.

Actions
Bij elke state moet de taxi een beslissing maken. Het kan de volgende acties uitvoeren:

- Noord
- Oost
- Zuid
- West

- Ophalen
- Afzetten
-

Je kunt aan de afbeelding zien dat er ook obstakels zijn. Op de huidige state zou de taxi bijvoorbeeld niet naar west kunnen rijden.

Rewards
Om de taxi op de rechte pad te leiden, moeten we beloningen uitreiken bij goede stappen en penalties geven bij slechte stappen.

Als reward kunnen we gewoon getallen gebruiken. Denk bijvoorbeeld aan: elke goede stap die de cab zet, krijgt het 1 erbij. Bij een verkeerde zet gaat er 1 van af. Dus des te hoger je getal, des te beter het rijdt.

- Als de taxi een klant op een goede locatie afzet, krijgt het een +1 bij het totaal.
- Als de taxi een verkeerde actie uitvoert (bijvoorbeeld op verkeerde locatie afzetten), krijgt het -1
- Als de taxi tegen een obstakel aankomt, krijgt het -1

Q-learning
Q-learning is een reinforcement algortime die probeert om de beste acties te zoeken bij de huidige state. Het leert om de totale reward te maximaliseren. Bij onze situatie leert de agent. Een q-learning algoritme die ik heb gevolden is de volgende:

$$Q(state, action) \leftarrow (1-\alpha)Q(state, action) + \alpha\Big(reward + \gamma \max_a Q(next\ state, all\ actions)\Big)$$

De Q staat voor quantity. We proberen met deze algoritme de q-value voor iedere state te updaten. Je pakt de oude q-value en update die met de nieuw geleerde value. De nieuw geleerde value is een combinatie van de reward die de agent voor zijn action gekregen heeft en maximum reward van de volgende state.

Wat we dus doen, is het kiezen van de juiste actie voor de huidige state door het kijken naar de rewards voor de huidige Q(state, action) en de maximale rewards voor de volgende state. Zo probeert onze agent de route met de beste rewards te vinden.