Adviesrapport Reinforcement Learning

Verkeers Cluster oplossing met reinforcement learning

Studenten: Carlo Keizer, Tom Blokland, 1744029
Datum: 04-06-2021

Versie: 1

Inhoud

Huidig model

Huidige stoplichten werken met sensoren. Als een auto op een bepaalde rijbaan voor een stoplicht staat kan een sensor dit waarnemen en dit gebruiken voor een stoplichtensysteem.

Een aantal huidige oplossingen voor het probleem zijn wiskundige modellen. Op de Haagse Hogeschool werken studenten aan een optimale stoplichten cluster met grafentheorie. [1]

Het advies voor het implementeren van Q learning en RL wordt gegeven ten opzichte van dit soort wiskundige modellen.

resultaten

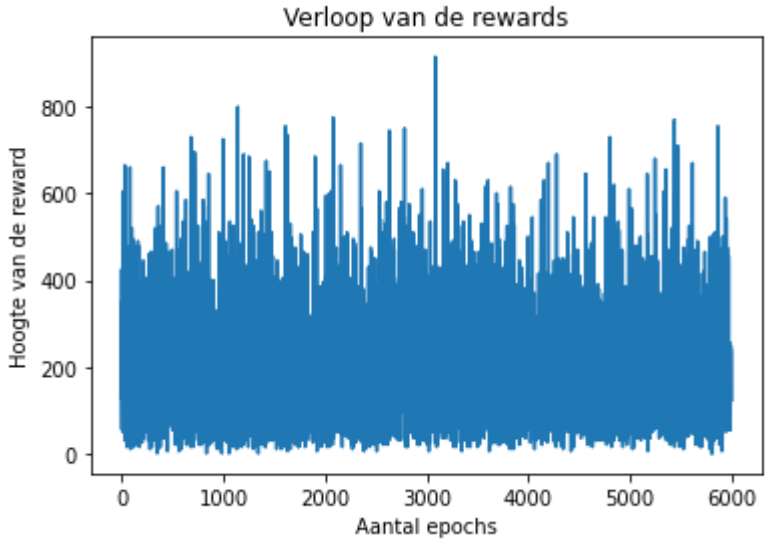

Figure 1: rewards during 10.000 episodes

In het tweede notebook hebben wij voor een game een reinforcement learning model getraind. Tijdens het trainen kwamen wij er achter dat ons model nauwelijks tot niet convergeert met verschillende parameters op 1000 epochs. Om zeker te weten of wij iets verkeerd deden of dat wij ons model niet lang genoeg lieten trainen, hebben wij het model een keer voor 6000 epochs getraind met de standaard parameters.. De resultaten hiervan zijn te zien in de grafiek hierboven. Ook in deze grafiek is te zien dat de reward van ons model niet convergeert. Hieruit blijkt dus dat het belangrijk is om de juiste parameters te kiezen, dan om het langer te trainen en te hopen dat het model convergeert.

Een ander vermoeden wat wij hebben is dat ons model te lang blijft exploreren. Oftewel het model blijft willekeurige acties kiezen, in plaats van dat er in verloop van het trainingsproces over wordt gegaan naar de beste actie. Hierdoor is het niet mogelijk voor het model om te convergeren.

Een ander vermoeden wat we hebben is dat het model de beste actie al kiest zonder dat het model iets geleerd heeft. Oftewel het model exploiteert te snel. Hierdoor kiest het model actie waarvan de q-waarde niet goed is berekend. waardoor het model nooit goed zal convergeren.

De actie die wij uit dit resultaat kunnen nemen, is dat wij de epsilon kunnen aanpassen en kijken hoe daarop gereageerd word.

Op basis van een paper over het fine tunen van DQN is de epsilon in dit project verlaagd. [2] Zij tonen aan dat in vergelijkbare context een gemiddelde epsilon van 0.3 optimaal is voor de reward. De huidige epsilon loopt van 0.1 tot 1.0 dat misschien iets te hoog is, en daarom kan het model te greedy zijn. Daarom is het experiment opnieuw gerund waarbij de epsilon van 0.1 tot 0.6 loopt binnen 750 runs. Dat betekend dus dat de laatste 250 runs die altijd op 0.6 blijft.

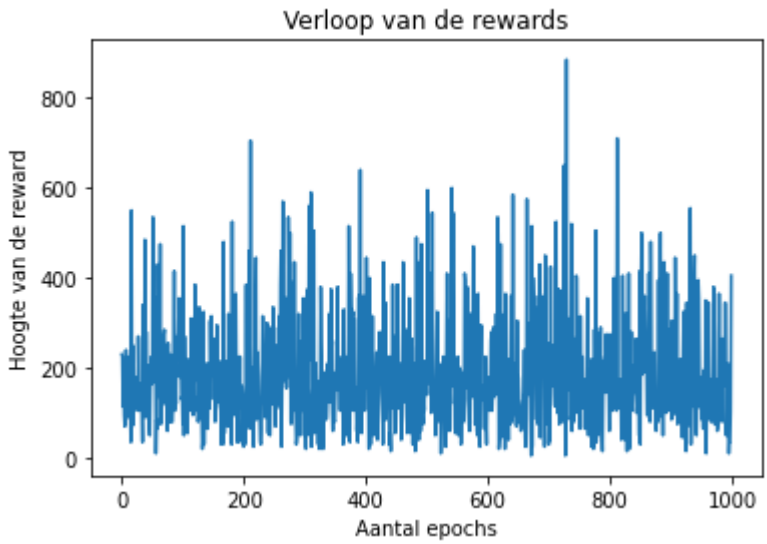

Figure 2: rewards during 1000 episodes with lower epsilon

Het blijkt dat in de eerste 1000 episodes het model niet beter wordt bij een lagere Epsilon. Dit zou te maken kunnen hebben met het feit de Epsilon in de paper een niet-lineaire functie in plaats van een lineaire functie in ons onderzoek.

Voor- en nadelen

Er zijn een aantal voordelen en nadelen voor het gebruik van Reinforcement Learning boven andere algoritmen. Een aantal daarvan worden verder toegelicht in de volgende hoofdstukken.

Voordelen

- Ten opzichte van de huidige oplossingen, is een RL model meer acuuraat. Er kunnen dus meer auto's passeren in een bepaalde tijd vergeleken met de huidige oplossingen.
- Een hoge accuracy
- Ook al is het model een black box model en is het moeilijk te bewijzen of het goed werkt, kan dit ook als voordeel worden gezien. Er hoeft niet precies bewezen te worden hoe het model werkt om een hoge reward te krijgen. Dus je hoeft ook geen statistische modellen te maken van real life data.
- Afhankelijk van de input data kan er met heel veel variabelen rekening gehouden worden. Bij foto's wordt rekening gehouden met alles dat binnen die foto zichtbaar is.
- Reinforcement Learning heeft geen grote gelabelde datasets nodig. Er hoeft dus geen input en output data verzameld worden van verkeers clusters.

Nadelen

- Ten opzichte van huidige oplossingen, leert een RL model van het gedrag van het verkeer. Maar het gedrag van de mens is niet altijd constant de verschillende tijden, dagen, seizoenen, vakantie en het weer hebben invloed op het gedrag van de mens. Het kan dus voorkomen dat het gedrag van de mens tijdens het trainen van het model anders is dan wanneer het model in gebruik wordt genomen. Dit kan ervoor zorgen dat model niet altijd optimaal werkt. Dit probleem is te voorkomen om het model voor een langere tijd te trainen, zodat het mel elk soort gedrag van de mens optimaal werkt.
- Een reinforcement model leert tijdens een trainingsproces heel goed te anticiperen op zijn omgeving. Maar als er in zijn omgeving een verandering plaatsvindt, kan het model niet goed meer functioneren. Dit kan komen omdat er bijvoorbeeld een rijbaan met een stoplicht is bijgekomen. Dit zorgt er voor dat het aantal states gewijzigd wordt. Waardoor het model niet meer goed werkt. Het gevolg hiervan is dat het model opnieuw getraind moet worden. Dit probleem is niet te voorkomen en hier zal van te voren rekening mee gehouden moeten worden.
- Het ontwikkelen van een reinforcement model kost veel tijd ten opzichte van de huidige oplossingen, er zijn veel verschillende parameters die aangepast kunnen worden. En de parameters hebben ook invloed op elkaar. Dit resulteert erin dat wanneer je een reinforcement model wilt ontwikkelen, dat het model vaak met veel verschillende parameters uitgevoerd moet worden. Wanneer het om een simpel probleem gaat is een reinforcement model hier redelijk snel mee klaar, maar wanneer het om een complex probleem gaat met meer states. Dan zal het model hier langer over doen. De combinatie van het model veelvuldig uitvoeren met verschillende parameters en het model wat er lang over doet om te trainen, zal er

dus in resulteren dat het ontwikkelen van een reinforcement model veel tijd gaat kosten.
- Dit nadeel komt dus voor wanneer het model veel states heeft, om dit probleem zoveel mogelijk te voorkomen, zou je de environment zo simpel mogelijk moet houden. wat er voor zorgt dat de trainingstijd van het reinforcement model afneemt.
- Het is niet te bewijzen of het model goed werkt of niet. (Er zijn vuistregels voor Parameters maar niet vaste regels)
- Data waarover het model een beslissing maakt is alleen afkomstig van een afgebakend gebied. Files die ver achter de stoplichten voorkomen worden hierdoor niet meegenomen

Ethische overwegingen

Wanneer je wilt dat een verkeers cluster een hoge doorstroming heeft kan het voorkomen dat het model ervoor kiest om op drukke momenten de drukste wegen groen licht te geven. Hierdoor zouden de minder drukke stoplichten nooit groen licht krijgen. Dit is ethisch niet verantwoord, want hierdoor zou het model gaan discrimineren op de minder drukke stoplichten. Een oplossing hiervoor is beschreven onder reward/penalty in het paragraaf Praktische overwegingen.

Praktische overwegingen

er moet een duidelijke een “eenzijdige” reward gedefinieerd worden. Het model kijkt simpelweg naar deze reward en houdt geen rekening met de context (bijvoorbeeld dat ambulances voorrang moeten krijgen, of dat het OV voorrang moet krijgen)

Reward/Penalty

Het is lastig om een goede reward te definiëren, vaak is het zo dat een reinforcement model onverwacht gedrag gaat vertonen, maar het model blijft zich wel aan de afgesproken regels houden. Bijvoorbeeld wanneer het model een reward krijgt voor elke auto die een groen licht passeert. Dan zou het kunnen gebeuren dat er op het drukste moment van de dag, het drukste stoplicht altijd op groen komt te staan. In theorie doet het model het dan perfect, want er worden dan zoveel mogelijk auto’s doorgelaten. Maar in de praktijk blijkt dan dat er nog altijd auto’s zijn die helemaal geen groen krijgt. Een oplossing hiervoor zou kunnen zijn om het model een steeds hogere penalty te geven afhankelijk van hoe lang er al een auto voor een rood licht staat te wachten. Hierdoor zou de penalty op een gegeven moment sterker moeten gaan meetellen dat de rewards. waardoor het model de keuze moet maken om de auto’s die het langste wachten het eerste door te laten.

Data

Om het model te kunnen trainen is er een omgeving nodig om data uit te extraheren die wordt gebruikt om het model te trainen. Dit zou een computersimulatie kunnen zijn of er kunnen echte sensoren worden aangebracht op een kruispunt om zo realistische data te kunnen verkrijgen. Er zal per rijbaan/stoplicht bijgehouden worden hoeveel auto's er voor het stoplicht staan te wachten met een bepaald maximum. Een state zou dan gezien kunnen worden als momentopname waarbij voor elk stoplicht de lengte van de wachtrij wordt gegeven. Met de volgende formule kan het aantal mogelijk states berekend worden.

$$amount\ of\ states\ =\ maximum\ amount\ of\ cars\ on\ one\ lane^{\ amount\ of\ lanes}$$

Als we hier uitgaan van een kruispunt waar je vanuit elke richting alle richtingen op kan gaan, dus 3 x 4 stoplichten. en met een maximum wachtrij van 15 dan komt hier het volgende uit 15^12 = 1.3 x 10^14. Dit zijn erg veel states, waardoor het trainen van model erg veel tijd gaat kosten. We zouden een wachtrij kunnen opdelen in chunks/bins van 4. Als we dit doen komt hier het volgende uit 4^12 = 16.777.216. Dit zijn al een stuk minder states en dit zou voor een sneller trainingsproces moeten zorgen. Een praktische overweging zou dus zijn om de lengte van de wachtrij op de delen in 4 bins.

Om het model toe te passen moet er data verzameld worden om de huidige state en omgeving vast te stellen.

DQN en state space

Als data verzameld wordt met sensoren van hoe de omgeving eruit ziet, dan is het misschien beter om een network de bellman functie laten benaderen. De states zijn zo complex en daar zijn er zoveel van dat de reward voor elke state moeilijk berekend kan worden.

Bij DQN is het nodig, zo niet verstandig, om limieten erg globaal instellen (bijvoorbeeld: rijen in bins {1,2,3}). Een voorbeeld voor het gebruik zonder globale bins bij bins van 12 voor aantal auto's per rijbaan, en bins van 12 bij aantal rijbanen kom je al op 1000.000.000.000 states uit.

Dynamische omgeving

Als het model getraind is op 1 stoplicht cluster, dan kan die het minder goed doen op een ander stoplicht cluster.

De situatie kan dus verschillen bij verschillende soorten stoplicht clusters, maar ook op verschillende tijden, dagen, seizoenen, vakantie en weer.

Planning

Door de grootte van de state space wordt dus geadviseerd om een DQN te gebruiken. Om het model toe te passen moeten een aantal taken uitgevoerd worden om dit zo goed mogelijk te doen.

Voor het project met space invaders was de state space 80x105 = 8400. Dit had een learning time van ongeveer 1 uur. Om het model te trainen en eventueel aan te passen adviseren wij de state space niet ver boven 8400 te initialiseren.
Er is een overzicht gemaakt van hoeveel bins er gemaakt moeten worden om onder de 8400 states te blijven. De formule hiervoor is:

$$aantal\ bins\ =\ floor\ (\ \sqrt[aantal\ rijstroken]{max\ bin\ threshold}\ )$$

$$where\ max\ bin\ threshold\ =\ 8400$$

Stel dat de maximale aantal auto's op een rijstrook is 10, dan kunnen de bins als volgt ingesteld worden om de state space te verkleinen.

Tabel 1: max possible bins and state space given amount of lanes

aantal rijstroken	aantal states	max aantal bins	nieuwe aantal states	max wachtrij	bin threshold
1	12	nan		12	8400
2	144	91	8281		
3	1728	20	8000		
4	20736	9	6561		
5	248832	6	7776		
6	2985984	4	4096		
7	35831808	3	2187		
8	429981696	3	6561		
9	5159780352	2	512		
10	61917364224	2	1024		
11	74300	2	2048		

	8370688				
12	8916100448256	2	4096		
13	1069932053 79072	2	8192		
14	1,28392E+15	1	1		
15	1,5407 E+16	1	1		
16	1,84884E+17	1	1		

In een paper [3] waar een een soortgelijk project wordt gedaan met dezelfde package Gym worden de gebruikte parameters voorgelegd. De parameters voor het model zouden volgens deze default waarden moeten worden geïnitialiseerd.

Tabel 2: parameter settings used to train DQN

Parameter	**Value**
Reply memory size	750
Mini-batch size	32
the size of most recent frames	4
Discount factor	0.95
Update frequency for target network	10000
Learning rate	0.00025
Initial -greedy policy value	0.6
Final -greedy polci value	0.1
Gradient momentum	0.95
Squared gradient momentum	0.95
Memory size	400000

De memory size kan wel hoger gekozen worden als gebruik gemaakt wordt bij sensoren in plaats van images. De omgeving is dan kleiner en er kan dan meer opgeslagen worden.

Om de simulatie in praktijk goed te laten werken moet data verzameld worden en toegepast worden in de simulatie. Variabelen als snelheid van de auto's, reactietijd en drukte heeft veel invloed op de environment. De conclusie uit de simulatie is meerzeggend over de werkelijkheid als de simulatie ook op de werkelijkheid lijkt.

Daarnaast zal het model periodiek getraind moeten worden als de omgeving veranderd. Als er bijvoorbeeld nieuwe rijstroken worden aangelegd zal de state veranderen en moet het model dus opnieuw getraind worden.

De dynamische omgeving heeft dus invloed op het model. Omdat het model anders werkt op een andere omgeving moet het model ook gefinetuned en getraind worden op verschillende verkeers clusters.

Omdat er geen settings gevonden zijn om het model te convergeren kan er geen advies geven worden over mogelijke parameters.

Een laatste, maar alsnog belangrijke overweging is het fine tunen.
Het model moet niet te veel gefinetuned worden, anders beperk je het model tot een specifieke omgeving. Als het model voor meerdere diverse kruispunten moet werken dan is een algemener model beter. [3]

Conclusie

Met de notebooks die gebruikt zijn om QN en DQN te gebruiken voor reinforcement learning blijkt dat QN wel werkt, gegeven dat het een simpele omgeving betreft, en een DQN niet werkt met de parameters die uitgeprobeerd zijn. Er is geprobeerd om de aantal episodes te verhogen om een hogere reward te verkrijgen, maar de reward bleef even laag en even inconsistent. Op basis van vooronderzoek over het effect van de Epsilon op de reward bij DQN's [2] is geprobeerd de Epsilon te verlagen. Dit verhoogde de rewards niet op binnen de eerste 1000 episodes. Een mogelijke oorzaak is dat er een lineaire epsilon decay wordt gebruikt, en in de paper een niet-lineaire epsilon decay wordt gebruikt.

Reinforcement Learning heeft veel potentie voor complexe situaties waarin er weinig gelabelde data beschikbaar is en waarbij optimalisatie zonder randvoorwaarden gewenst is. Echter werkt het minder goed in dynamische omgevingen, bij situaties waarin de minimale states erg hoog zijn of bij situaties waarin zekerheid over resultaten prioriteit heeft.

Het is ook belangrijk om erop te letten dat er mogelijk gediscrimineerd kan worden op verschillende soorten doorstromingen.

In praktijk zijn er een aantal zaken om te overwegen. Een eis bij Reinforcement Learning is dat er een duidelijke en “eenzijdige” reward gedefinieerd moet worden. Dan is het belangrijk om in de reward rekening te houden met meerdere factoren, zoals dat de doorstroming gelijk verdeeld zijn.

De data die wordt gebruikt om het model te trainen. kan worden verkregen uit een computersimulatie, of uit een echt kruispunt door middel van sensoren. Een state van het model kan hierbij gezien worden als een moment opname van het verkeers cluster.

Ook is het belangrijk om de states te verminderen. Bij een kruispunt met maximaal 15 auto’s per rijbaan en 12 rijbanen (4x3) zijn de aantal states zodanig hoog dat te lang duurt om het model te runnen. Omdat de state space zo groot is wordt geadviseerd om DQN te gebruiken om zo een Neural Network de Bellman functie te benaderen in plaats van een QN te gebruiken. Ook is dit de reden waarom er voor nu alleen gekeken wordt naar het verkeer van auto’s en niet het verkeer van fietsers.

Om de state space te verminderen is er in “Planning” in een tabel de maximale aantal bins genoteerd om de state space onder 8400 te houden. 8400 is de state space dat binnen dit project gebruikt is binnen de casus “space invaders” en wordt als limiet gedefinieerd. Hoe meer bins, hoe hogere potentie het model heeft om de reward te optimaliseren. Maar er moet dus een afbakening gemaakt worden om het in praktijk toe te kunnen passen.

Ook moet het model goed om kunnen gaan met dynamische situaties. Als er nieuwe wegen worden aangelegd of er iets veranderd in de omgeving zal het model opnieuw getraind moeten worden. Ook zal het model waarschijnlijk slechter presteren op kruispunten die net iets anders zijn. Het model is dus geschikt voor elk kruispunt als het op een kruispunt is getraind.

Een manier om met dynamische situaties om te gaan is het model niet te veel finetunen, wordt het model beperkt tot een specifieke kruispunt terwijl een algemener model beter werkt. [3]

Bronvermelding

[1] https://docplayer.nl/25044815-Grafentheorie-verslag-a-g-okker-hermansyah.html

[2] https://theses.ubn.ru.nl/bitstream/handle/123456789/5216/Nieuwdorp%2C_T._1.pdf?sequence=1

[3] https://pure.tue.nl/ws/files/46933213/844320-1.pdf