Environment en werking DQN

# Hoofdconcepten (TODO wetenschappelijke notatie van letters)

## De agent en de omgeving

De basisbouwstenen van een probleem in *reinforcement learning* (RL) bestaan uit de agent en de omgeving. De agent is een actor die verschillende keuzes kan maken in een specifieke omgeving. De omgeving bevindt zich in een zekere staat st op moment t zoals te zien in figuur 1. De agent heeft keuze tot verschillende acties at in de omgeving. Het resultaat van een actie zorgt voor een transitie in staat (st+1) op een volgend tijdstip. Na de uitvoering van een welbepaalde actie zendt de omgeving een signaal terug in de vorm van een beloning r.

Diagram

Description automatically generatedHet uiteindelijke doel is dat de agent een sequentie van acties leert dat resulteert in de maximale beloning. Dit wordt gedaan aan de hand van de beleidsfunctie π. Deze functie geeft op basis van elke staat de best mogelijke actie terug voor die staat. Door verschillende acties uit te proberen, ondervindt de agent verschillende beloningen of straffen (negatieve beloningen). Door herhaaldelijk te interageren met de omgeving leert de agent wat te doen in welke situatie.

Figuur : Agent en omgeving

In het geval van zelfrijdende auto’s kan de agent worden voorgesteld als de actor, de omgeving als de openbare weg, de staten verschillende verkeersituaties en de acties in de vorm van manoeuvres dat de auto kan uithalen. Wanneer het doel van de auto is om punt B te bereiken van punt A waarbij de auto zich aan de verkeersregels moet houden, zal de auto een negatieve beloning ontvangen wanneer het een verkeersovertreding maakt.

## Markov-beslissingsproces

Een Markov-beslissingsproces (MDP), vernoemt naar de Russisch wiskundige Andrey Markov (1856-1922) biedt een wiskundige omkadering om besluitvormingen voor te stellen die deels stochastisch en deels onder de controle van een besluitvormer vallen. MDP wordt gebruikt in RL om sequentiële beslissingsproblemen te modelleren. Optimalisatieproblemen die oplosbaar zijn met dynamisch programmeren (link dynamisch programmeren) kunnen worden geschetst aan de hand van MDP’s.

Background pattern

Description automatically generated

Figuur 2: Voorbeeld van een eenvoudige MDP met drie toestanden (groene cirkels) en twee acties (oranje cirkels), met twee beloningen (oranje pijlen).

MDP’s beschikken over de Markoveigenschap, die geldt als volgt: de volgende staat hangt enkel af van de huidige staat en de beschikbare actie en niet van historisch geheugen van voorgaande staten of van informatie van ergens anders. (bronvermelding) Dit is cruciaal in RL-problemen omdat dit het beredeneren van toekomstige staten mogelijk maakt met enkel de informatie van de huidige staat.

Formeel wordt een MDP in RL voorgesteld door middel van de *5-tuple (S, A, T*a, Ra, ϒ):

* S is een eindige verzameling van toegelaten staten van de omgeving
* A is een eindige verzameling van acties
* Ta is de kans dat actie a in staat s op tijdstip t zal overgaan in de staat s’ op tijdstip t+1
* Ra is de beloning nadat een actie a zorgt voor de overgang van staat s naar staat s’
* ϒ is de verminderingsfactor die het verschil in belang van huidige en toekomstige beloningen voorstelt

Extra Uitleg gamma zoals in theorieboek

De verminderingsfactor ϒ geeft weer hoeveel belang er moet gehecht worden aan beloningen in de toekomst ten opzichte van beloningen in het huidig moment. Wanneer de opdracht continu is en voor een lang tot oneindig aantal seconden moet lopen, wordt de verminderingsfactor ϒ kleiner dan 1 gezet. Bij een spel waarbij een voorwerp moet gebalanceerd worden op een kar of een paal zal de lagere waarde van ϒ ervoor zorgen dat de beloningen in de tegenwoordige tijd harder doorwegen.

Bij episodische taken zoals in figuur 3, waarbij een agent het einde van een doolhof wil bereiken, zal de verminderingsfactor rond de waarde 1 geplaatst worden aangezien de taak eindig en de finale staat waarbij de agent het einde van het doolhof bereikt heeft, het einddoel is.

Chart, box and whisker chart

Description automatically generated

Figuur : doolhofomgeving

## 

## Beleid π

## Pad τ

## 

## Winstwaarde R

## Staatwaarde V

## Staat-Actiewaarde V

## RL objectief

## Exploratie

## Off-Policy learning

## Q-learning

# Deep Value-based Reinforcement Learning

## Hoog dimensionele problemen

## DQN

# ProtoType Tree

# Interpreteerbare DQN met ProtoType Tree

# Alternatieve interpreteerbare oplossingen in Reinforcement learning