Indhold

[What is Reinforcement Learning 4](#_Toc159934102)

[The Reinforcement Learning Framework 4](#_Toc159934103)

[The RL Process 4](#_Toc159934104)

[The reward hypothesis: the central idea of Reinforcement Learning 5](#_Toc159934105)

[Markov Property 5](#_Toc159934106)

[Observations/States Space 5](#_Toc159934107)

[Action Space 6](#_Toc159934108)

[Rewards and the discounting 7](#_Toc159934109)

[**Type of tasks** 8](#_Toc159934110)

[Episodic task: 8](#_Toc159934111)

[Continuing tasks: 9](#_Toc159934114)

[**The Exploration/Exploitation trade-off** 9](#_Toc159934118)

[**Two main approaches for solving RL problems** 10](#_Toc159934119)

[The Policy π: the agent’s brain 10](#_Toc159934120)

[Policy-Based Methods 11](#_Toc159934127)

[Policy-Based Methods TLDR: 12](#_Toc159934130)

[Value-based methods 12](#_Toc159934131)

DRL

Noter d. 27/2

# **What is Reinforcement Learning**

* Idéen bag Reinforcement Learning, er at en agent (AI) vil lære fra environment ved at interagere med det (Gennem trial and error) og derved modtage rewards (positiv eller negativ) som feedback på performance.

“Reinforcement Learning is just a **computational approach of learning from actions.”**

# **The Reinforcement Learning Framework**

## Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, design Automatisk genereret beskrivelseThe RL Process

* + The RL Process: a loop of state, action, reward and next state

Et billede, der indeholder tekst, skærmbillede, design

Automatisk genereret beskrivelse

* + Vores **Agent** modtager en state fra **Environment –** der modtages første frame af vores spil (Environment)
  + Ud fra den state , laver Agenten en **action** – Agenten rykker til højre
  + Environment går til at nyt state – en ny frame i spillet
  + Environment giver en reward til Agenten – den er ikke død (positiv reward +1)
* Dette RL loop outputs en sekvens af **state, action, reward** og **next state.**

## The reward hypothesis: the central idea of Reinforcement Learning

* Den essentielle idé er at maksimere den forventede gevinst. Dette koncept stammer fra tanken om, at enhver målsætning kan beskrives som maksimering af den forventede kumulative Reward. I RL stræber vi efter at lære at træffe handlinger, der maksimerer den forventede samlede belønning for at opnå optimal adfærd.

## Markov Property

* Markov Property er afgørende i RL, hvor processen ofte betegnes som en **Markov Decision Process** (MDP). Markov Property indebærer, at vores agent kun behøver den aktuelle tilstand for at træffe beslutninger om handlinger, og ikke behøver at tage højde for hele historikken af tidligere tilstande og handlinger.

## Observations/States Space

* Observations/States repræsenterer den information, vores agent modtager fra **environment**. Dette kan f.eks. være et billede i tilfælde af et videospil eller værdien af en bestemt aktie for en handelsagent. At forstå og arbejde med disse Observations er afgørende for at opnå succesfuld Reinforcement Learning.

Der er en forskel i Observation (o) og state (s)

* + State (s) er en complet beskrivelse af ”the state” af hele environmentet(ingen gemte informationer). Her snakkes der om et fully observed environment, fx skak.
  + Observation (o) er en partial beskrivelse af state. Her snakkes der i partially observed environment, fx Super Mario. Vi kan kun beskrive det vi kan se i framen.

Et billede, der indeholder tekst, skærmbillede, kvadratisk, grafisk design

Automatisk genereret beskrivelse

## Action Space

* Action space er alle actions der er mulige at foretage i et environment.

Actions kan komme fra en *discrete* eller *continuous space:*

* + *Discrete space*: antallet af mulige actions er begrænset

f.eks In Super Mario Bros, we have only 4 possible actions: left, right, up (jumping) and down (crouching). Again, in Super Mario Bros, we have a finite set of actions since we have only 4 directions.

* + *Continuous space*: antallet af mulige actions er uendelige

f.eks A Self Driving Car agent has an infinite number of possible actions since it can turn left 20°, 21,1°, 21,2°, honk, turn right 20°…

Et billede, der indeholder tekst, skærmbillede, grafisk design

Automatisk genereret beskrivelse

Taking this information into consideration is crucial because it will **have importance when choosing the RL algorithm in the future.**

## Rewards and the discounting

* Reward er fundamental I RL da det er den eneste feedback for en agent. Dette gør at agent ved om en foretaget action var god eller dårlig.
* The cumulative reward at each time step **t** can be written as:

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, linje/række

Automatisk genereret beskrivelse

To discount the rewards, we proceed like this:

1. We define a discount rate called gamma. **It must be between 0 and 1.** Most of the time between **0.95 and 0.99**.

* The larger the gamma, the smaller the discount. This means our agent **cares more about the long-term reward.**
* On the other hand, the smaller the gamma, the bigger the discount. This means our **agent cares more about the short term reward (the nearest cheese).**

2. Then, each reward will be discounted by gamma to the exponent of the time step. As the time step increases, the cat gets closer to us, **so the future reward is less and less likely to happen.**

* Our discounted expected cumulative reward is:

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, diagram

Automatisk genereret beskrivelse

# **Type of tasks**

* En opgave er en instans af et Reinforcement Learning problem. Vi kan have to typer af opgaver: **episodic** og **continuing**.

## Episodic task:

## I denne case har vi et startpunkt og et slutpunkt (en terminal state). Dette skaber rammerne for en episode: en liste af States, Actions, Rewards og nye States

## F.eks i Super Mario: en episode begynder ved start af et nyt level og slutter når Mario dør eller ender målet

## Continuing tasks:

## Dette er de tasks, der fortsætter for evigt (no terminal state). I denne case må en agent lære at fortage den bedste action og samtidig interagere med environmentet

## F.eks en agent der foretager automated stock trading. For denne tasks er der intet startpunkt og terminal state. Agenten fortsætter indtil vi selv vælger at stoppe den.

## Et billede, der indeholder tekst, skærmbillede, grafisk design, Multimediesoftware Automatisk genereret beskrivelse

# **The Exploration/Exploitation trade-off**

**Exploration:**

* Exploration er når agentent udforsker environment ved at foretage random actions til at udforske og få information omkring environmentet

**Exploitation:**

* Exploitation er når vores agent exploiter kendt information til at maximere Reward

**Eksempel:**

* Remember, the goal of our RL agent is to maximize the expected cumulative reward. However, **we can fall into a common trap**.

Let’s take an example:

Et billede, der indeholder skærmbillede, clipart, Grafik, design

Automatisk genereret beskrivelse

In this game, our mouse can have an **infinite amount of small cheese** (+1 each). But at the top of the maze, there is a gigantic sum of cheese (+1000).

However, if we only focus on exploitation, our agent will never reach the gigantic sum of cheese. Instead, it will only exploit **the nearest source of rewards,** even if this source is small (exploitation). But if our agent does a little bit of exploration, it can **discover the big reward** (the pile of big cheese).

This is what we call the exploration/exploitation trade-off. We need to balance how much we **explore the environment** and how much we **exploit what we know about the environment.**

Et billede, der indeholder tekst, skærmbillede, design

Automatisk genereret beskrivelseTherefore, we must **define a rule that helps to handle this trade-off**. We’ll see the different ways to handle it in the future units.

# **Two main approaches for solving RL problems**

Med andre ord, hvordan bygger vi en RL agent der kan vælge de actions der maximerer det forventede samlede reward

## The Policy π: the agent’s brain

## Policy π er agentens hjerne, det er den funktion der fortæller os hvilken action der skal foretages ved given state vi er i. Derved definere den agentens opførsel på given tidspunkt.

## Policy *Think of policy as the brain of our agent, the function that will tell us the action to take given a state*

## Denne policy er funktionen vi gerne vil lære, vores mål er at finde den optimale policy π\*, den policy der maximerer forventet return når agenten følger den. Vi finder den π\* (optimale policy) gennem træning.

## Det er 2 måder at træne vores agent til at finde den π\* (optimale policy)

## Direkte, ved at lære agenten hvilken action at foretage givet nuværende state: Policy-Based Methods.

## Indirekte, her lærer vi agenten hvilket step er mere ”værdifuldt” og derfra foretage en action der fører til den mest værdifulde state: Value-Based Methods.

## Policy-Based Methods

In Policy-Based methods, **we learn a policy function directly.**

This function will define a mapping from each state to the best corresponding action. Alternatively, it could define **a probability distribution over the set of possible actions at that state.**

We have two types of policies:

* *Deterministic*: a policy at a given state **will always return the same action.**

## Et billede, der indeholder tekst, Font/skrifttype, Grafik, skærmbillede Automatisk genereret beskrivelse

* *Stochastic*: outputs **a probability distribution over actions.**

## Policy Based

## Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, design Automatisk genereret beskrivelseEt billede, der indeholder tekst, skærmbillede, Font/skrifttype, viol Automatisk genereret beskrivelsePolicy-Based Methods TLDR:

## 

## Value-Based methods

## I Value-Based Methods, I stedet for at lære en policy funktion, lærer vi en value funktion der mapper en state til den forventede value af at være i den state.

## Value af den state er den forventede discounted return agenten kan få hvis den stater i den state, og derefter opfører sig efter vores policy (“Act according to our policy” just means that our policy is “going to the state with the highest value”).

## Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, linje/række Automatisk genereret beskrivelse

## Here we see that our value function defined values for each possible state.

## Et billede, der indeholder skærmbillede, kvadratisk, design Automatisk genereret beskrivelse*Thanks to our value function, at each step our policy will select the state with the biggest value defined by the value function: -7, then -6, then -5 (and so on) to attain the goal.*

## Takket være vores value function, ved hvert step vores policy vil vlkge den state med det største value defineret af value funktionen: -7, -6, også -5 (osv.) til at opnå vores mål.

## Et billede, der indeholder tekst, skærmbillede, Font/skrifttype Automatisk genereret beskrivelseEt billede, der indeholder tekst, skærmbillede, Font/skrifttype, Rektangel Automatisk genereret beskrivelseValue-Based Methods TLDR:

# The “Deep” in Reinforcement Learning

You’ll see the difference is that, in the first approach, **we use a traditional algorithm** to create a Q table that helps us find what action to take for each state.

In the second approach, **we will use a Neural Network** (to approximate the Q value).

## Value based RL