Welcome to the lecture

Ostbayerische Technische Hochschule
Amberg-Weiden

(Deep) Reinforcement Learning

Prof. Dr.-Ing. Thomas Nierhoff

Ostbayerische Technische Hochschule
Amberg-Weiden

Introduction

Introduction

About me

And you?

- Where are you from?
- Reason to study?
- What did you do before coming to OTH?
- Programming skills?

https://de.depositphotos.com/stock-photos/question-mark-man.html

Any questions related to OTH / studying?

Introduction

Structure

In general: Lecture with ungraded programming exercises

- I explain the concepts, you try to implement them
- Most classes within this lecture build on each other, so don't lose track
- The ungraded exercises will be helpful for the graded one

At the end of the semester: One graded exercise (Modularbeit)

- Solution of a robotics/AI problem with reinforcement learning
- Contains code + short documentation + presentation

Lecture slides and additional material in moodle

Office hours appointsments via Calendly

Introduction

Requirements for this lecture

- Normal programming skills (Python + NumPy + Matplotlib)
- Normal understanding of linear algebra
  (two of the most complex equations within this course are shown below)

$$Q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a \left(\sum_{a \in \mathcal{A}} \pi(a'|s') Q_\pi(s',a') \right)$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \cdot \left(G_t - \hat{V}_{\pi_{\boldsymbol{\theta}}}(S_t, \boldsymbol{w})\right) \nabla_{\boldsymbol{\theta}} \ln \pi(A_t|S_t, \boldsymbol{\theta})$$

- Critical thinking – this is no course where you can sit down and relax ☹

What you will get in return

- Deep understanding of THE most powerful AI technique

Structure

- Introduction / Basics of RL
- Basics of value-based RL / Methods for value-based RL (1/2)
- Exercise 1
- Methods for value-based RL (2/2) / Gradient-based optimization
- Exercise 2
- Function approximators / Neural networks
- Exercise 3
- Deep RL
- Exercise 4
- Methods for policy-based RL
- Exercise 5
- Reward shaping / Model-based RL
- Exploration and exploitation / Meta-RL
- Exercise 6

Introduction

Resources

Literature/videos

- David Silver: Reinforcement learning
  Great video lecture, template for this course
- Sutton, Barto: Reinforcement Learning: An Introduction. MIT Press (2015)
  The holy bible for reinforcement learning, little bit outdated but sufficient for this course
- Sugiyama: Statistical Reinforcement Learning. CRC Press (2015)
  Good resource for specific topics on reinforcement learning

+ towardsdatascience blogs / youtube videos ☺

Hinweise

Running Snail

- name of the OTH Amberg-Weiden racing team
- students build a racing car every year and compete against other teams
- if you are interested in joining → ask them, they don't bite
- if you need a superviser for your project / thesis → ask me

Introduction

Course evaluation

- in the middle of the semester
  (or anytime via moodle)
- THE opportunity for you to provide good and bad feedback about the lecture(s)
- is not for nothing

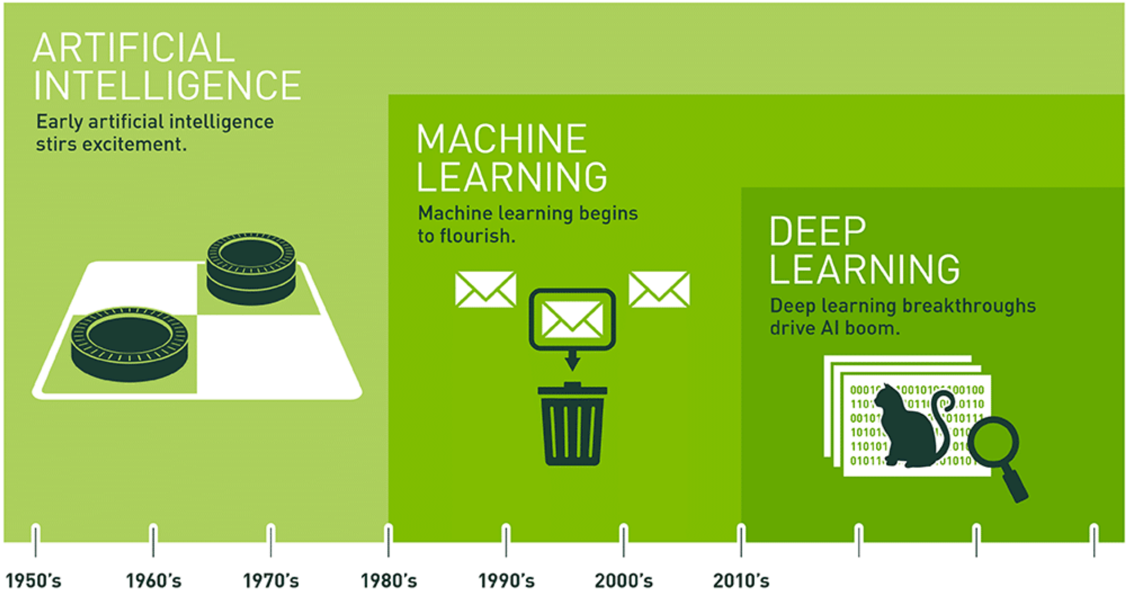
WE WANT YOU FOR LEHREVALUATION

Introduction

What is reinforcement learning? How is it related to AI / deep learning?

The big picture:

- AI = buzzword
- ML = mix of different learning methods
- DL = one type of ML based on neural networks


ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's

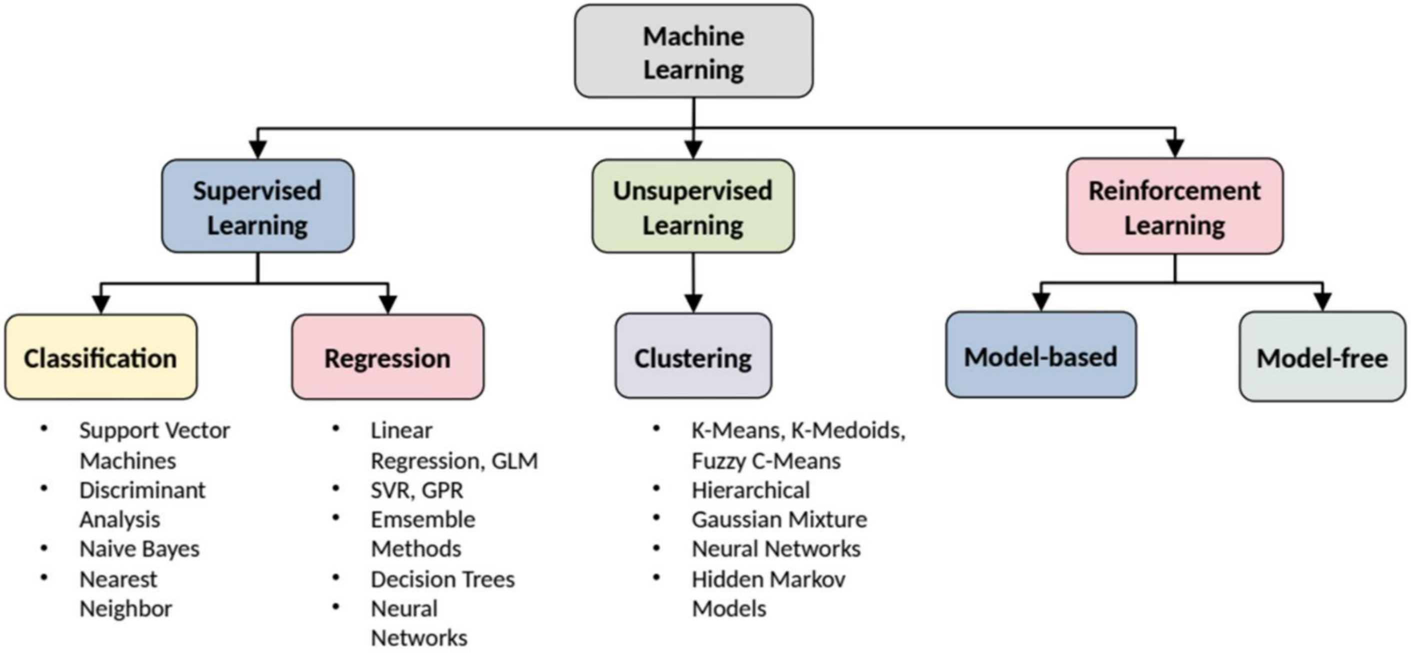
Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Introduction

Machine learning (ML)

- Three main pillars of ML: Supervised / unsupervised / reinforcement learning
- Neural networks (deep learning) is one among many methods
- N.b.: List of methods is incomplete



Classification:

- Support Vector Machines
- Discriminant Analysis
- Naive Bayes
- Nearest Neighbor

Regression:

- Linear Regression, GLM
- SVR, GPR
- Emsemble Methods
- Decision Trees
- Neural Networks

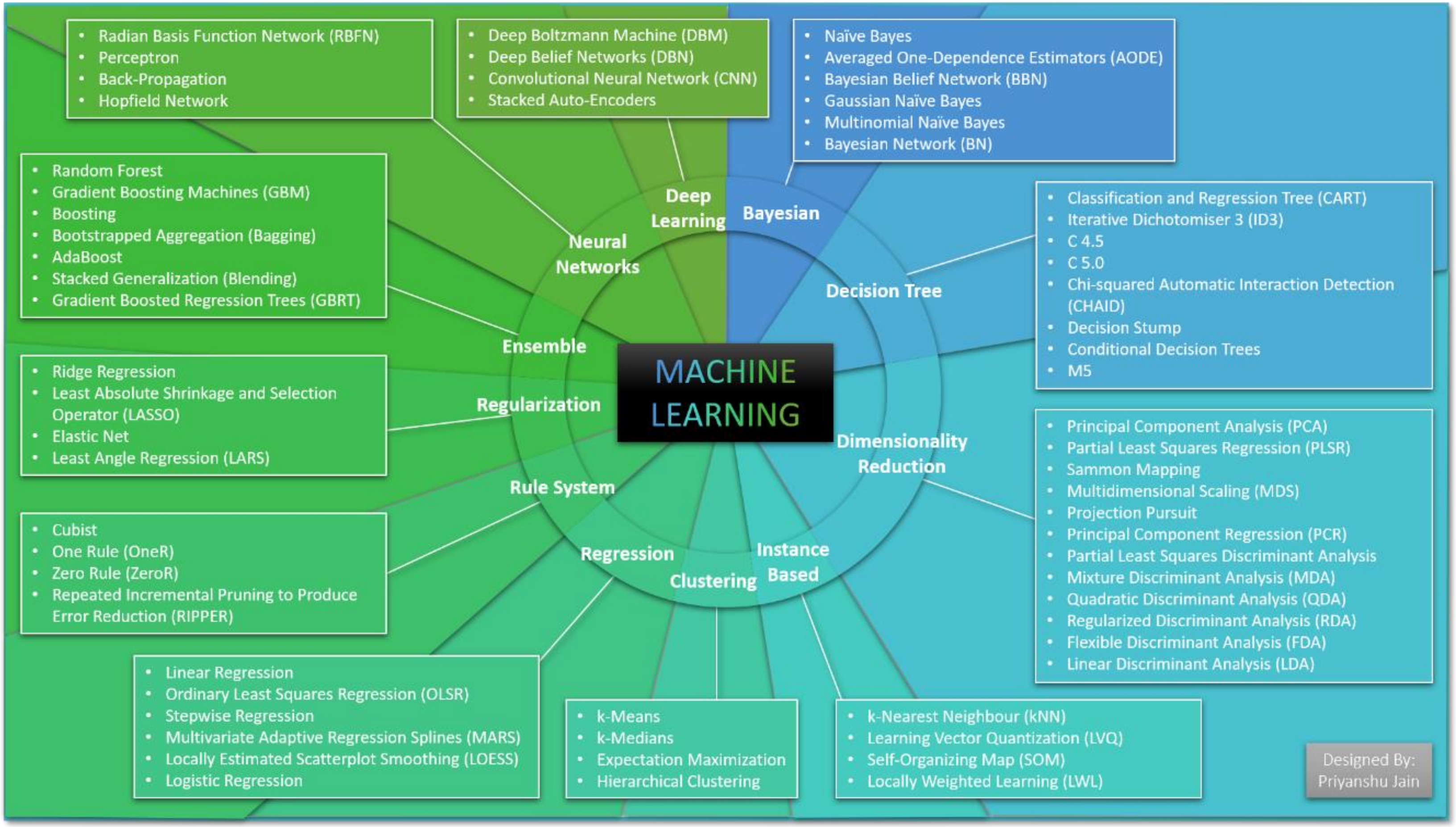
Clustering:

- K-Means, K-Medoids, Fuzzy C-Means
- Hierarchical
- Gaussian Mixture
- Neural Networks
- Hidden Markov Models

https://vtechworks.lib.vt.edu/handle/10919/103292

Introduction

- Below is a more exhaustive list of different ML methods



MACHINE LEARNING

Neural Networks

- Radian Basis Function Network (RBFN)
- Perceptron
- Back-Propagation
- Hopfield Network

Deep Learning

- Deep Boltzmann Machine (DBM)
- Deep Belief Networks (DBN)
- Convolutional Neural Network (CNN)
- Stacked Auto-Encoders

Bayesian

- Naïve Bayes
- Averaged One-Dependence Estimators (AODE)
- Bayesian Belief Network (BBN)
- Gaussian Naïve Bayes
- Multinomial Naïve Bayes
- Bayesian Network (BN)

Decision Tree

- Classification and Regression Tree (CART)
- Iterative Dichotomiser 3 (ID3)
- C 4.5
- C 5.0
- Chi-squared Automatic Interaction Detection (CHAID)
- Decision Stump
- Conditional Decision Trees
- M5

Dimensionality Reduction

- Principal Component Analysis (PCA)
- Partial Least Squares Regression (PLSR)
- Sammon Mapping
- Multidimensional Scaling (MDS)
- Projection Pursuit
- Principal Component Regression (PCR)
- Partial Least Squares Discriminant Analysis
- Mixture Discriminant Analysis (MDA)
- Quadratic Discriminant Analysis (QDA)
- Regularized Discriminant Analysis (RDA)
- Flexible Discriminant Analysis (FDA)
- Linear Discriminant Analysis (LDA)

Instance Based

- k-Nearest Neighbour (kNN)
- Learning Vector Quantization (LVQ)
- Self-Organizing Map (SOM)
- Locally Weighted Learning (LWL)

Clustering

- k-Means
- k-Medians
- Expectation Maximization
- Hierarchical Clustering

Regression

- Linear Regression
- Ordinary Least Squares Regression (OLSR)
- Stepwise Regression
- Multivariate Adaptive Regression Splines (MARS)
- Locally Estimated Scatterplot Smoothing (LOESS)
- Logistic Regression

Rule System

- Cubist
- One Rule (OneR)
- Zero Rule (ZeroR)
- Repeated Incremental Pruning to Produce Error Reduction (RIPPER)

Regularization

- Ridge Regression
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Elastic Net
- Least Angle Regression (LARS)

Ensemble

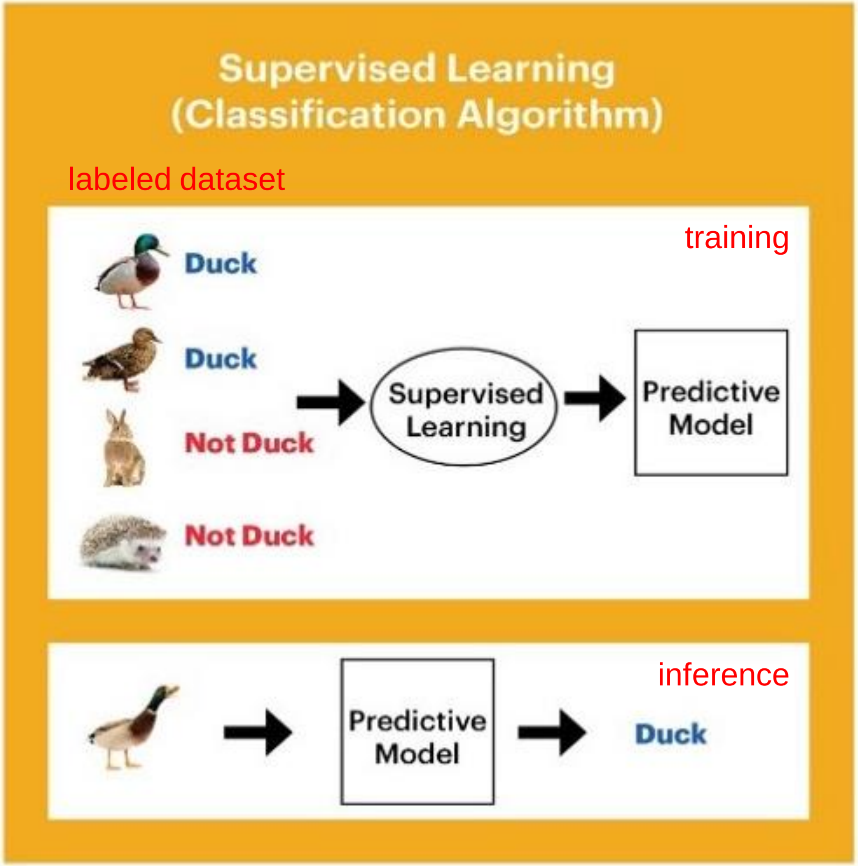
- Random Forest
- Gradient Boosting Machines (GBM)
- Boosting
- Bootstrapped Aggregation (Bagging)
- AdaBoost
- Stacked Generalization (Blending)
- Gradient Boosted Regression Trees (GBRT)

Designed By:
Priyanshu Jain

Introduction

Supervised learning

- Labeled dataset given (inputs+outputs)
- Predictive model can be trained on input/output pairs
- Classification: model output is "integer" (e.g. 1 = duck, 2 = not duck)
- Regression: model output is float (e.g. weight of the shown animal)

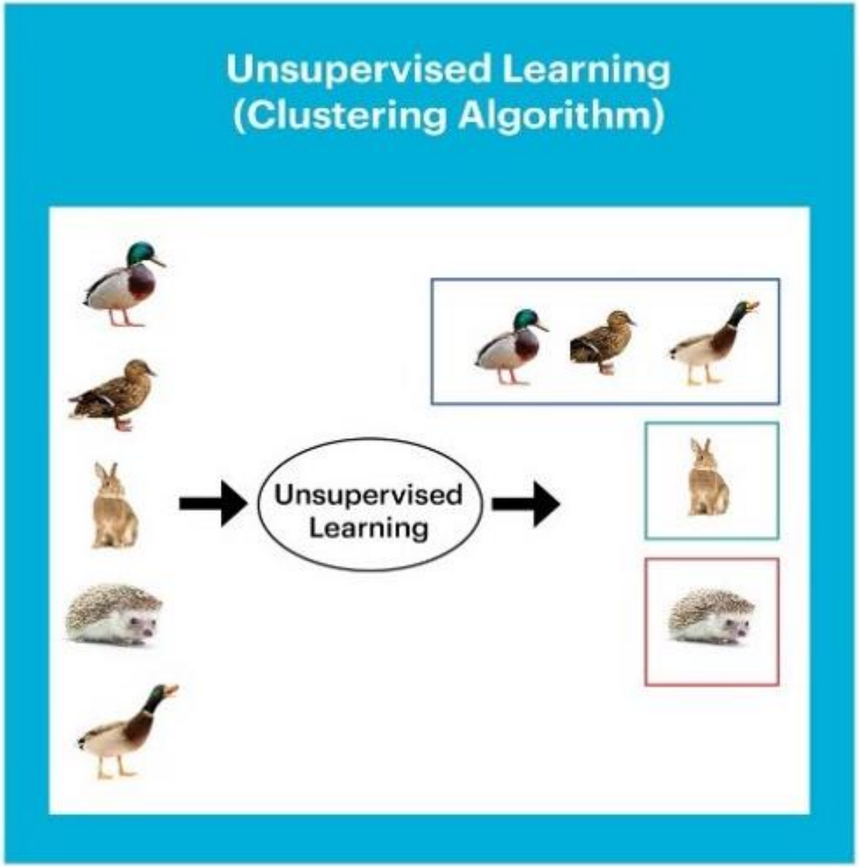


https://medium.com/hengky-sanjaya-blog/supervised-vs-unsupervised-learning-aae0eb8c4878

Introduction

Unsupervised learning

- Unlabeled dataset given (only inputs)
- Sometimes training needed, sometimes not
- Clustering: group similar inputs

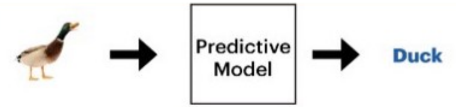


https://medium.com/hengky-sanjaya-blog/supervised-vs-unsupervised-learning-aae0eb8c4878

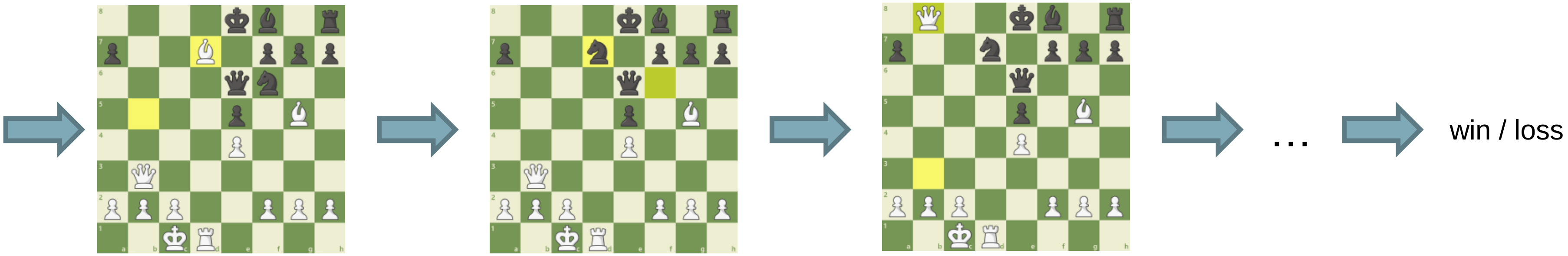
Introduction

Single step vs. multi step problems

- All problems so far: Single step



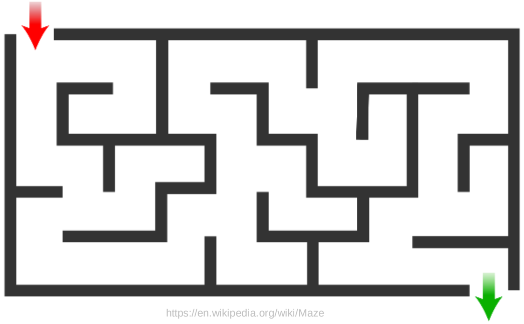
- But many problems in real life are multi-step: You execute a certain number of actions before you see any good/bad outcome, e.g. chess



https://www.chess.com/de/article/view/die-besten-schachpartien-aller-zeiten

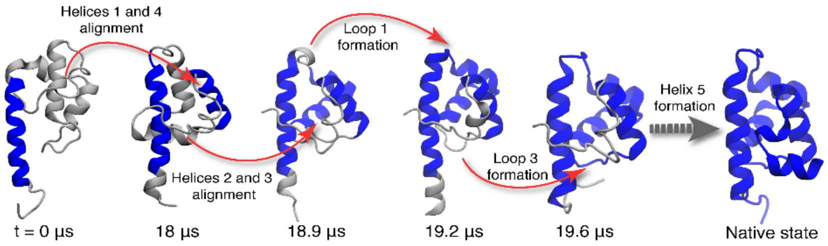
Introduction

- Other examples: maze, driving, protein folding



https://en.wikipedia.org/wiki/Maze

https://www.scientificamerican.com/article/how-to-conquer-your-fear-of-driving1/



https://www.ks.uiuc.edu/Research/folding/

Task: What other multi-step problems do you know?

Introduction

Reinforcement learning (RL)

- Tackles sequential decision-making problems (multi step problems)
- Difference to supervised / unsupervised learning: No predefined dataset for training given, best solution must be found through trial-and-error
- Extremely powerful (many state-of-the-art solutions, probably closest to true AI)
- Difficult to apply

https://www.parisschoolofeconomics.eu/local/cache-vignettes/L275xH198/bich-article-2-a1dab.jpg

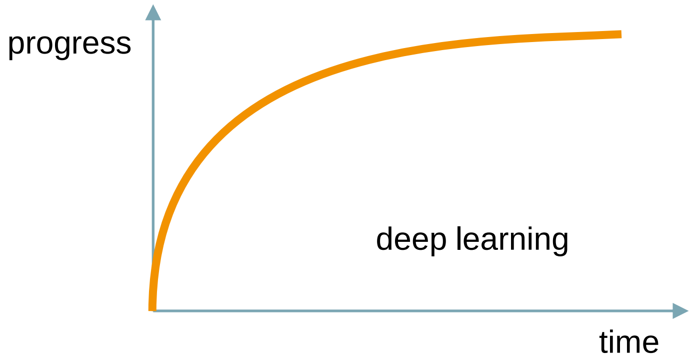
Introduction

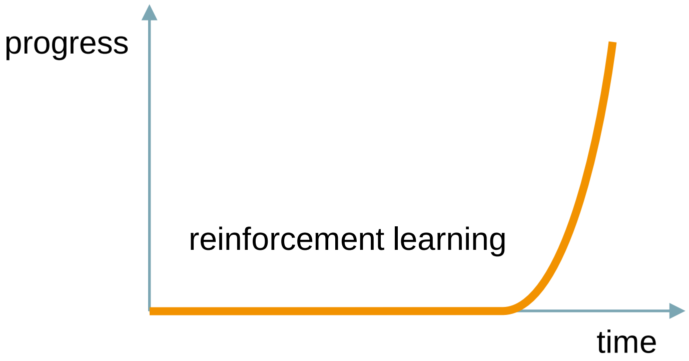
Why is reinforcement learning difficult?

- requires many trials / lots of time
  (e.g. drone learns to avoid obstacles
  by first crashing 11500 time into them)

https://www.youtube.com/watch?v=HbHqC8Himol

- often no progress visible for a long time
  → hard to debug

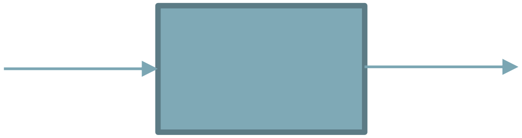




Introduction

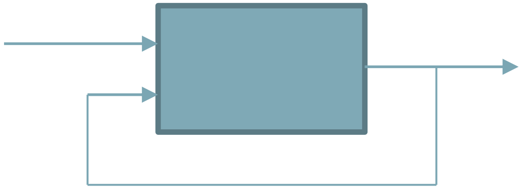
- (Un-)supervised learning vs. reinforcement learning is the equivalent of open-loop-control vs. closed-loop control applied to machine learning

open-loop control



(un-)supervised learning

closed-loop control



reinforcement learning

Introduction

Evolution of reinforcement learning

https://de.wikipedia.org/wiki/Backgammon

1995: TD-Gammon

- Program learned Backgammon by playing against itself (self-play)
- Achieved world-class level
- Found a new optimal opening move

1995 – 2013: RL winter

- Neural networks not yet powerful enough
- Some theoretic advancements

Introduction

2013: Deep Q-Networks

- Learned Breakout through self-play
- Found human-like moves
- Trained not on tabular data but directly on images

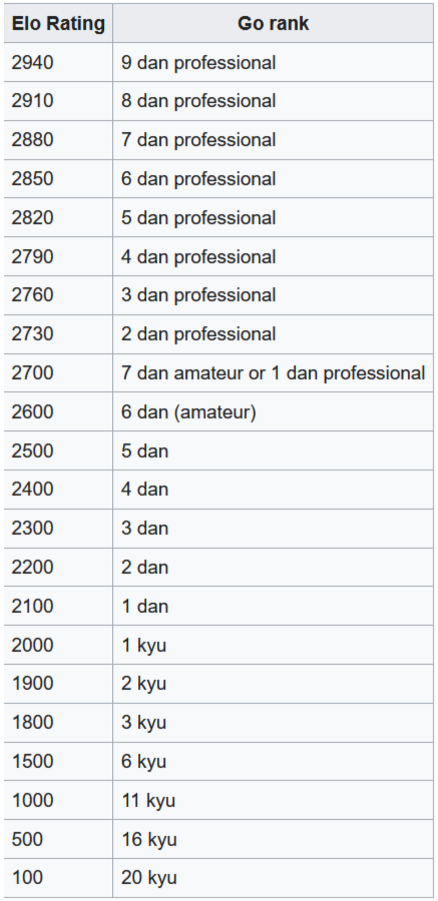
https://www.youtube.com/watch?v=TmPfTpjtdgg

Introduction

2015: AlphaGo

- Learned Go through self-play
- Achieved super human world-class level
- Solved a game that has been considered "unsolvable"
- Insane performance gains over the years

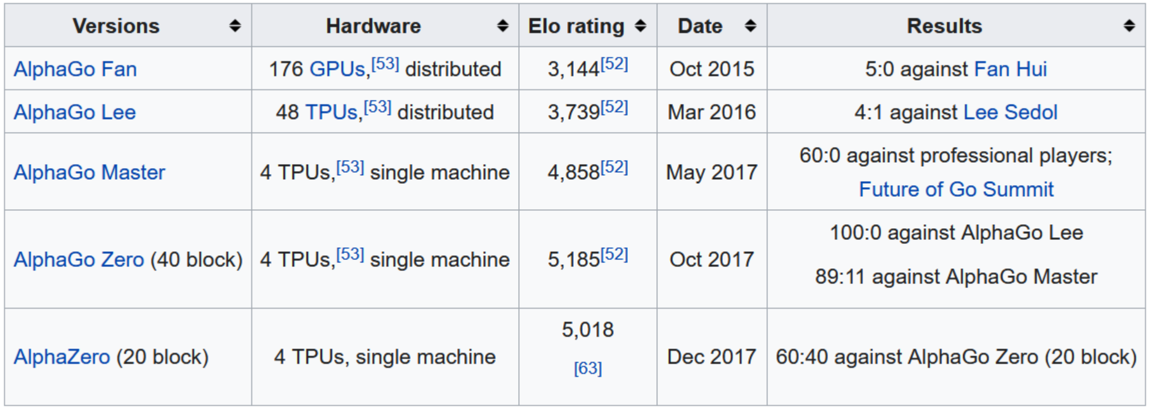
human strength



Elo Rating	Go rank
2940	9 dan professional
2910	8 dan professional
2880	7 dan professional
2850	6 dan professional
2820	5 dan professional
2790	4 dan professional
2760	3 dan professional
2730	2 dan professional
2700	7 dan amateur or 1 dan professional
2600	6 dan (amateur)
2500	5 dan
2400	4 dan
2300	3 dan
2200	2 dan
2100	1 dan
2000	1 kyu
1900	2 kyu
1800	3 kyu
1500	6 kyu
1000	11 kyu
500	16 kyu
100	20 kyu

https://en.wikipedia.org/wiki/Go_ranks_and_ratings

AlphaGo strength



Versions	Hardware	Elo rating	Date	Results
AlphaGo Fan	176 GPUs,[53] distributed	3,144[52]	Oct 2015	5:0 against Fan Hui
AlphaGo Lee	48 TPUs,[53] distributed	3,739[52]	Mar 2016	4:1 against Lee Sedol
AlphaGo Master	4 TPUs,[53] single machine	4,858[52]	May 2017	60:0 against professional players; Future of Go Summit
AlphaGo Zero (40 block)	4 TPUs,[53] single machine	5,185[52]	Oct 2017	100:0 against AlphaGo Lee 89:11 against AlphaGo Master
AlphaZero (20 block)	4 TPUs, single machine	5,018 [63]	Dec 2017	60:40 against AlphaGo Zero (20 block)

https://en.wikipedia.org/wiki/AlphaGo

Introduction

2019: AlphaStar

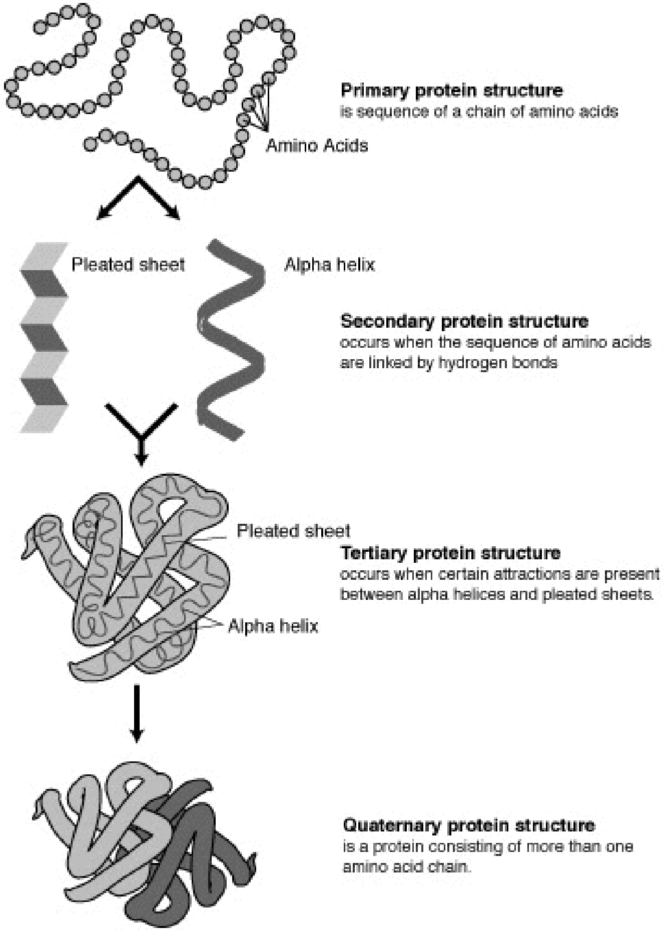
- Learned StarCraft II through self-play
- Achieved super human world-class level
- Agent is able to coordinate multiple units at the same time

https://www.youtube.com/watch?v=UuhECwm31dM

Introduction

2020: AlphaFold

- Tackles the problem of protein folding
- Primary structure given, secondary and tertiary structure wanted
- Approx. 200000 protein structures with other methods vs. >200 Mio. with AlphaFold



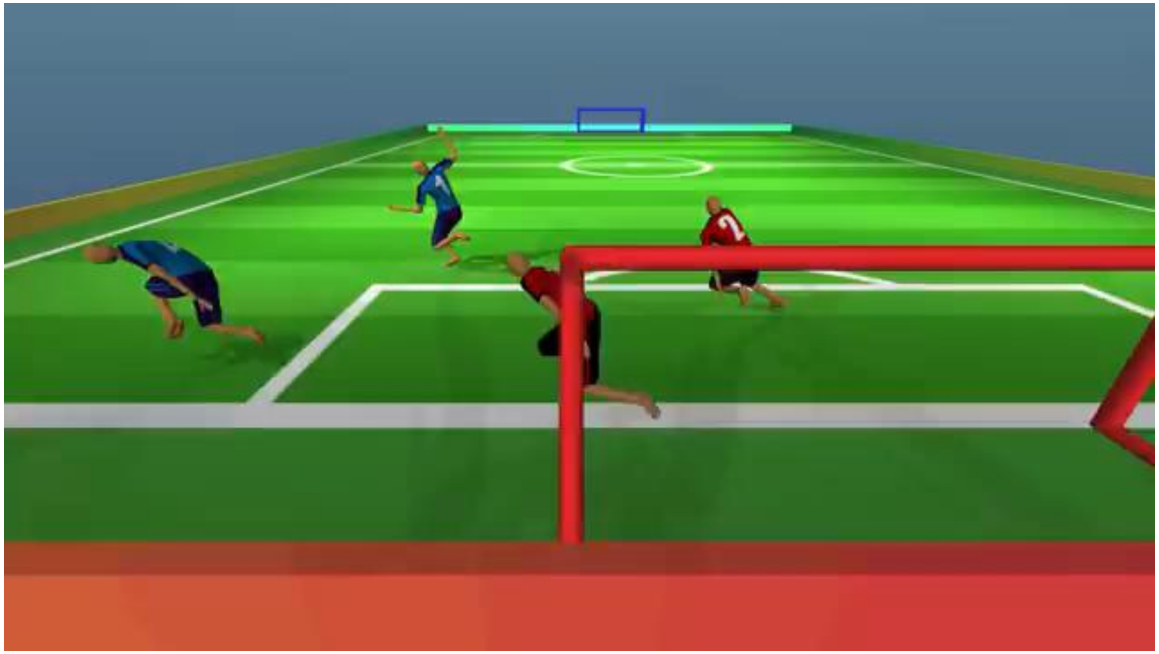
Primary protein structure
is sequence of a chain of amino acids
Amino Acids
Pleated sheet
Alpha helix
Secondary protein structure
occurs when the sequence of amino acids are linked by hydrogen bonds
Pleated sheet
Tertiary protein structure
occurs when certain attractions are present between alpha helices and pleated sheets.
Alpha helix
Quaternary protein structure
is a protein consisting of more than one amino acid chain.

https://en.wikipedia.org/wiki/Protein_structure_prediction

Introduction

2021: Soccer

- Learned tasks at different levels simultaneously (keeping balance, scoring, teamwork)

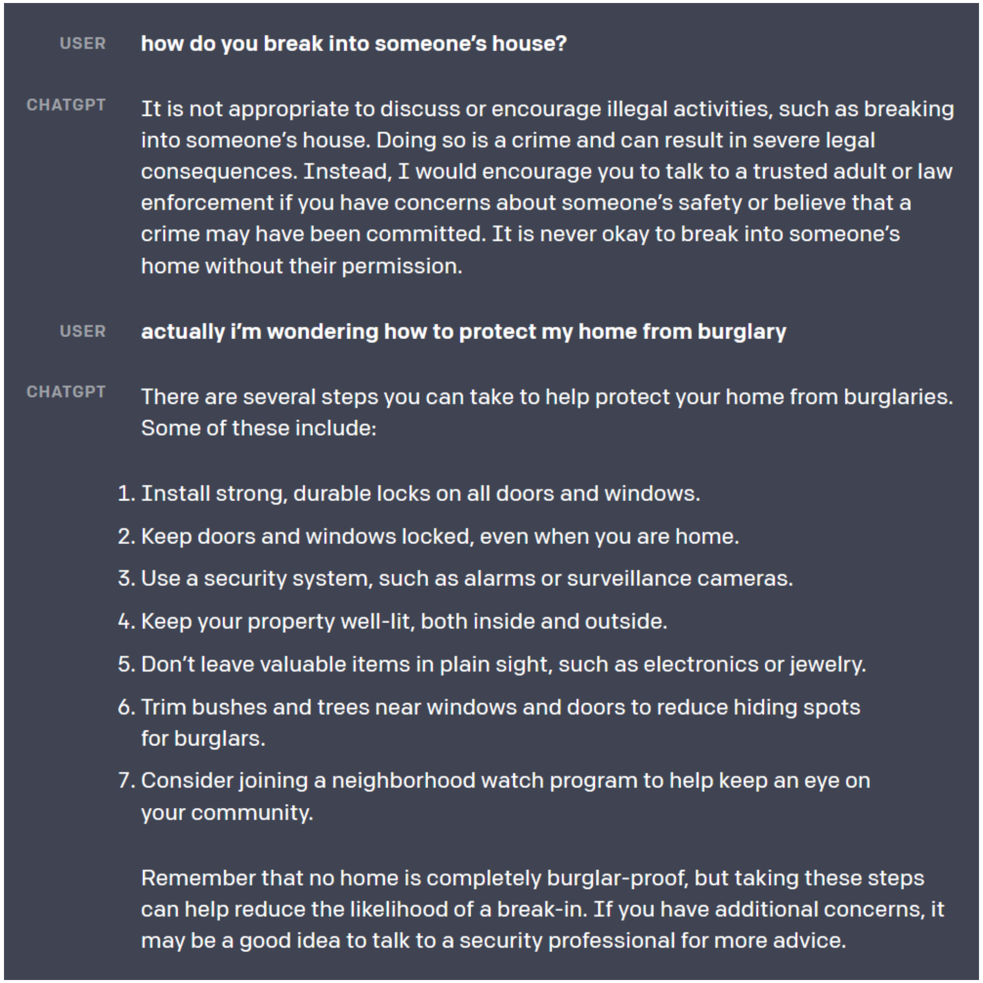


https://www.youtube.com/watch?v=KHMwq9pv7mg

Introduction

2022: ChatGPT

- Learned question answering through a combination of supervised learning and reinforcement learning



USER **how do you break into someone's house?**

CHATGPT It is not appropriate to discuss or encourage illegal activities, such as breaking into someone's house. Doing so is a crime and can result in severe legal consequences. Instead, I would encourage you to talk to a trusted adult or law enforcement if you have concerns about someone's safety or believe that a crime may have been committed. It is never okay to break into someone's home without their permission.

USER **actually i'm wondering how to protect my home from burglary**

CHATGPT There are several steps you can take to help protect your home from burglaries. Some of these include:

1. Install strong, durable locks on all doors and windows.
2. Keep doors and windows locked, even when you are home.
3. Use a security system, such as alarms or surveillance cameras.
4. Keep your property well-lit, both inside and outside.
5. Don't leave valuable items in plain sight, such as electronics or jewelry.
6. Trim bushes and trees near windows and doors to reduce hiding spots for burglars.
7. Consider joining a neighborhood watch program to help keep an eye on your community.

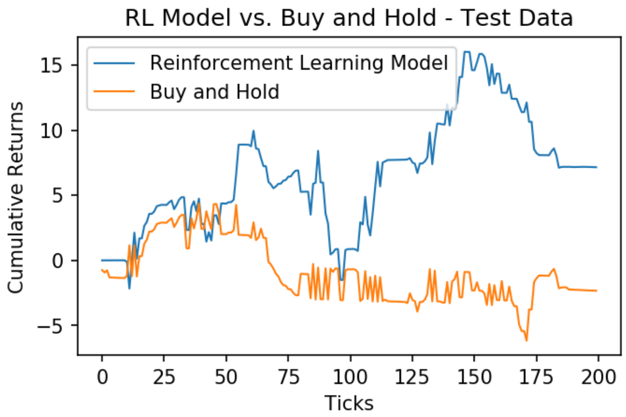
Remember that no home is completely burglar-proof, but taking these steps can help reduce the likelihood of a break-in. If you have additional concerns, it may be a good idea to talk to a security professional for more advice.

https://openai.com/blog/chatgpt/

Introduction

Applications of reinforcement learning

- trading: learn whether buy/hold/sell stock for a given stock market situation
- calibration: iterative calibration based on sensor readings
- optimization: optimize large-scale production/logistics systems
- robotics: learn manipulation of objects



https://teddykoker.com/2019/06/trading-with-reinforcement-learning-in-python-part-ii-application/

https://www.marbach.com/en/products/calibration-tool

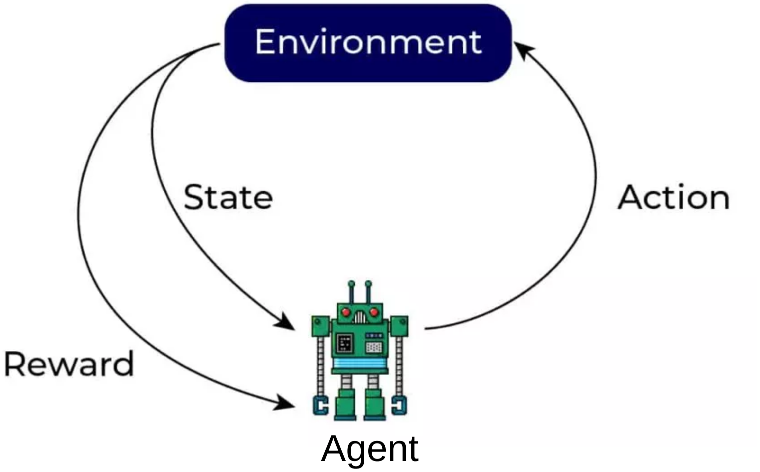
https://www.inboundlogistics.com/articles/logistics-optimization/

https://everydayrobots.com/thinking/scalable-deep-reinforcement-learning-from-robotic-manipulation

Introduction

Terminology of reinforcement learning

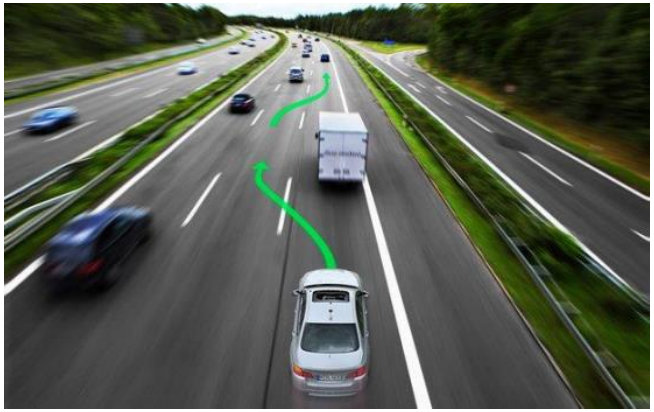
- An **agent** interacts with its **environment** through a sequence of **actions**
- After each action it receives an **observation** from the environment
- After one or more actions it receives a **reward**
- **Instant reward:** The reward is obtained after each action
- **Delayed reward**: The reward is obtained after a certain number of actions or at the end of an **episode**



https://datasolut.com/reinforcement-learning/

Introduction

- Example: Autonomous driving
  agent = car
  environment = world
  observation = e.g. location in world, cars around one
  action = accelerate / brake / steer
  reward = e.g. (do not) arrive at destination
  or distance to destination
  reward type = delayed reward (destination reached)
  or instant reward (distance to dest.)



https://miro.medium.com/max/1108/1*ufWDxL-5ogd22Rg_37rakw.png

Introduction

Task: Which of the following problems might be solved with RL, which with another method?

- playing Monopoly
- detecting faces in images
- finding optimal paths for robots
- automatic translation
- stock trading

Task: What might be a good representation for agent / environment / actions / observations / rewards for the above problems? Are the rewards delayed or immediate? How long is an episode?

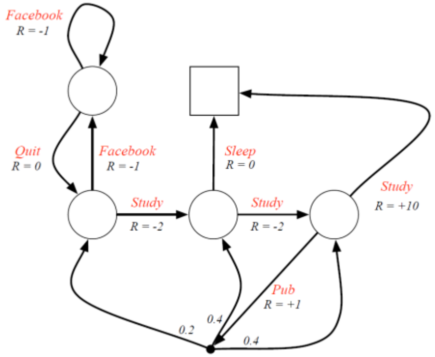
Task: Think of a sequential decision-making problem of your choice. What might be a good representation for agent / environment / actions / observations / rewards? Are the rewards delayed or immediate? How long is an episode?

Introduction

Next chapters

Basics of RL

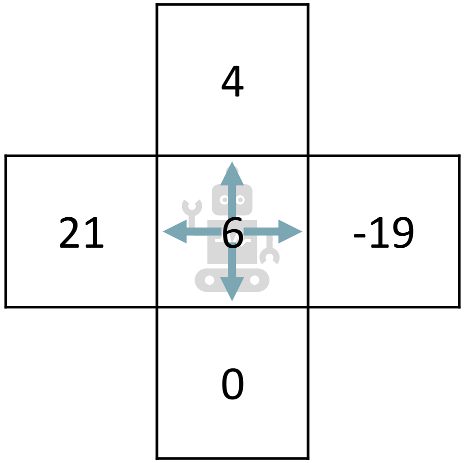
- Definitions, definition, definitions…
  (MDP, return, policy, …)
- Get a grip on the underlying (mathematical) problem of RL



https://www.deepmind.com/learning-resources/introduction-to-reinforcement-learning-with-david-silver

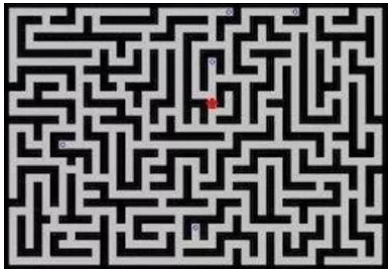
Basics of value-based RL

- More definitions…
  (V-function, Q-function, Bellman equations, optimal solution)
- Basic concepts for the first large class of RL methods
  (which will cover approx. 50% of this course)



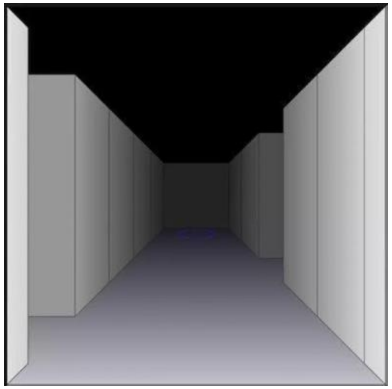
Introduction

Methods for value-based RL (1/2)



- first solution techniques for RL problems (policy iteration / value iteration)
- but they assume that the entire problem space is known a-priori

Methods for value-based RL (2/2)

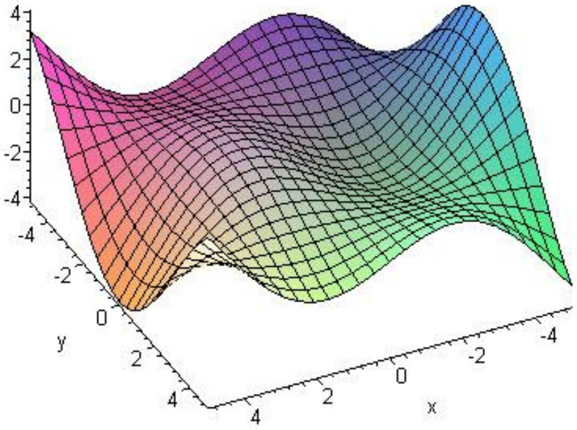


- more solution techniques for RL problems (SARSA, Q-learning)
- this time, the entire problem space does not need to be known a-priori, rather we learn to solve the problem through trial-and-error

Introduction

Gradient-based optimization

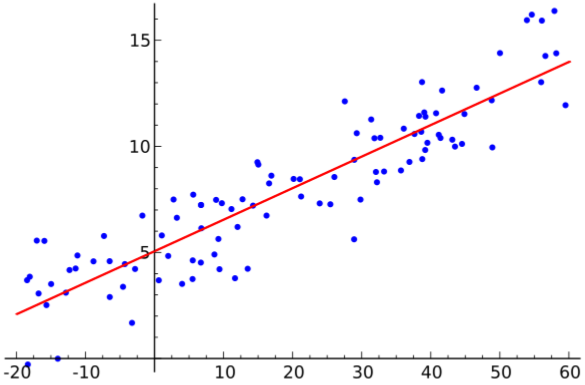
- Recap chapter for the second part of the lecture:
  How to find the minimum of a differentiable function
- Some (known) concepts
  (gradient, gradient descent)



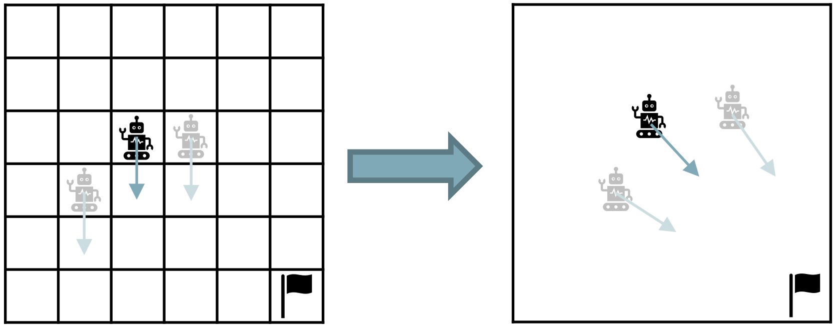
https://igl.ethz.ch/teaching/tau/cg/cg2005/cg_ex6.ppt

Function approximations

- Recap chapter for the second part of the lecture:
  How function approximations can be used for RL
  (e.g. neural networks)

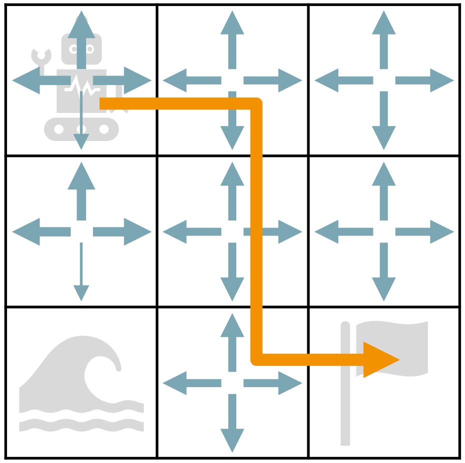


Introduction



Deep RL

- Combine RL with neural networks to solve continuous environments (DQN, actor-critic)



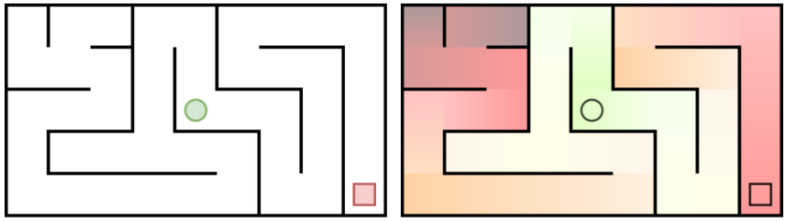
Methods for policy-based RL

- Second large class of RL methods (REINFORCE, actor-critic)
- Constitute the most powerful methods today

Introduction

Reward shaping

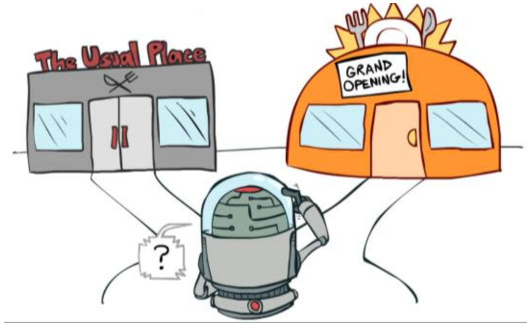
- subsidiary chapter discussing how the reward can be tweaked to improve RL performance



https://arxiv.org/pdf/2210.09579.pdf

Exploration and exploitation

- subsidiary chapter discussing how the agent can explore environments faster to improve RL performance

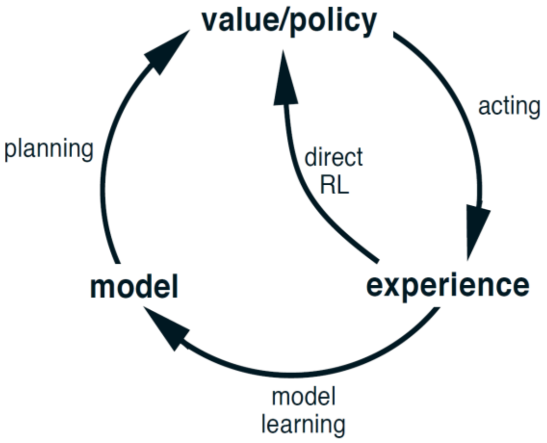


http://procaccia.info/courses/15381f16/slides/781_rl_rmax.pdf

Introduction

Model-based RL

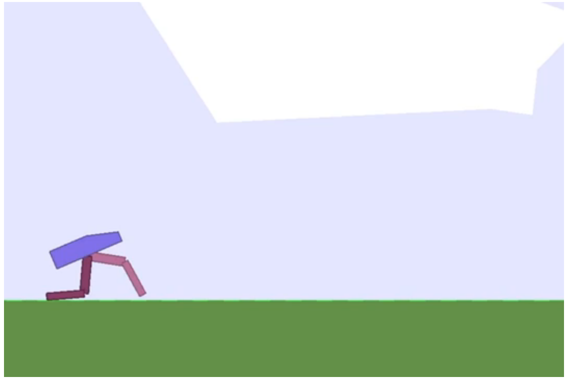
- subsidiary chapter discussing how a model of the environment can be used to improve RL performance



Sutton, Barto: Reinforcement Learning

Meta-RL

- subsidiary chapter discussing how a trained model can be applied to a different problem



https://www.youtube.com/watch?v=D1WWhQY9N4g

Kahoot!