Reinforcing the path II
(The path to home-sweet-home)

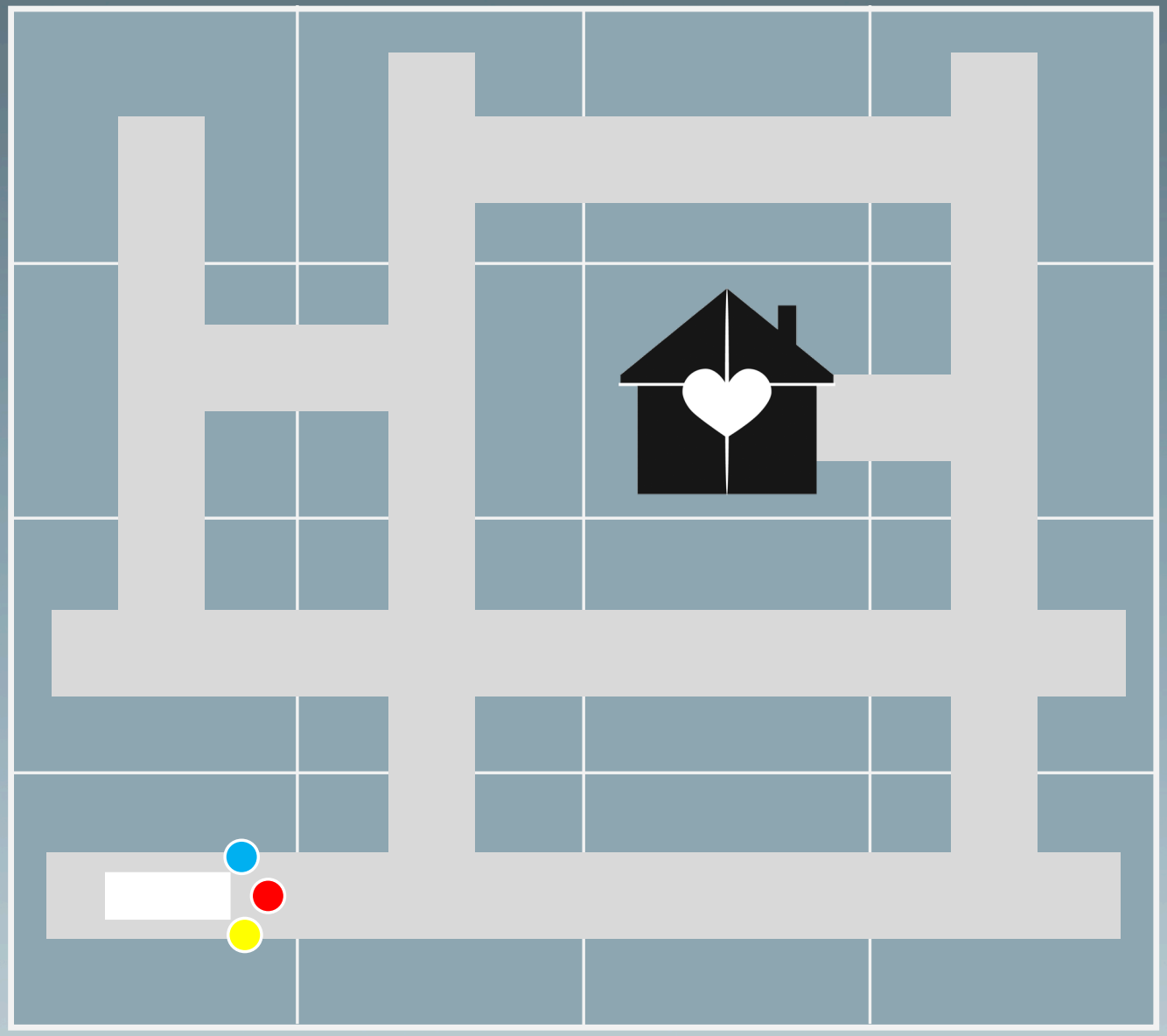
TechAngelist Meetup

In order of appearance:

The Force (always there)
Dr. M. Naci Akkøk (me)
Master Yoda
Some animals like
Python & Anaconda
And now a "deep" car with a deeper learning

April 18th 2018 @17:30 – 19:00

ORACLE



The path home	The journey schedule
• **Step #00** – Most important first: PIZZA! (Don´t forget to say hello to people) (Take something to drink for the long journey)	• Depends upon how hungry you are
• **Step #01** – Review: The math & algorithms of reinforcement learning + Gridworld (Berkeley) (The force behind Artificial Intelligence & Machine Learning)	• About 20 minutes
• **Step #02** – Neural Networks and Deep Q-learning (Donkeys always remember the path home. Let´s see if a car can do that too)	• About 25 minutes
Break (about 15 minutes, strongly recommended)	
• **Step #03** – A bit about the tools, sources, links (In case you want to try – see the meetup page)	• About 10 - 15 minutes
• **Step #04** – Code Review & Play!	• About 30 minutes
• **Step #99** – A cloud AI/ML platform, Q&A	• Depends upon how tired you are

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Reinforcing the path

Oslo Maskinlæring Meetup

In order of appearance:

The Force (always there)
Dr. M. Naci Akkøk (me)
Master Yoda
Some animals like
Python & Anaconda

January 22nd 2018 @17:30 – 19:00

ORACLE

The path

- **Step #00** – Most important first: PIZZA!
  (Don´t forget to say hello to people)
  (Take something to drink for the long journey)
- **Step #01** – The math & algorithms of reinforcement learning
  (The real force behind Artificial Intelligence & Machine Learning)

Break (strongly recommended)

- **Step #02** – The tools, the sources, the links
  (In case you want to try)
  (Some already posted on the Meetup page)
- **Step #03** – Gridworld
  (The Berkeley implementation)
- **Step #99** – Q&A

The journey schedule

- Depends upon how hungry you are
- About 30-45 minutes
- About 15-20 minutes
- About 30 minutes
- Depends upon how tired you are

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Q-Learning Review: The challenge, the goal

(Somebody invented Reinforcement Learning for a reason)

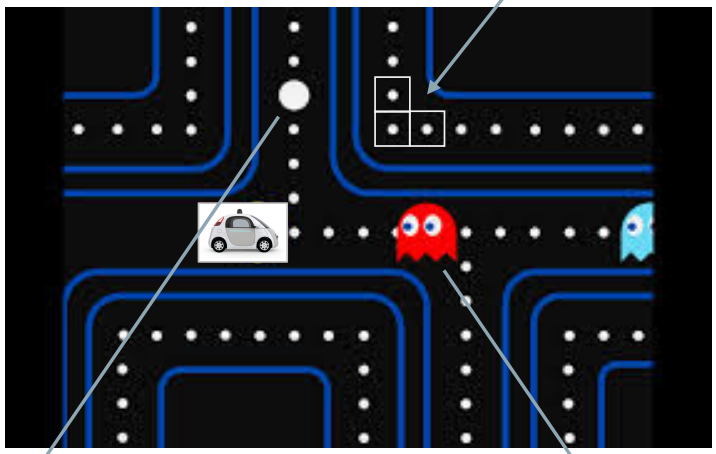
Copyright © 2015 Oracle and/or its affiliates. All rights reserved.
Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |

GRIDWORLD(S)

Each step is a move to a neighboring cell (up, down, left, right) in some sort of a grid

Pacman (our “agent”) needs to learn to eat/win and to survive (avoid being eaten).

And we need to teach it, or help it learn!



WIN - REWARD
R = +1

LOOSE - PENALTY
R = -1

BASIC IDEA:

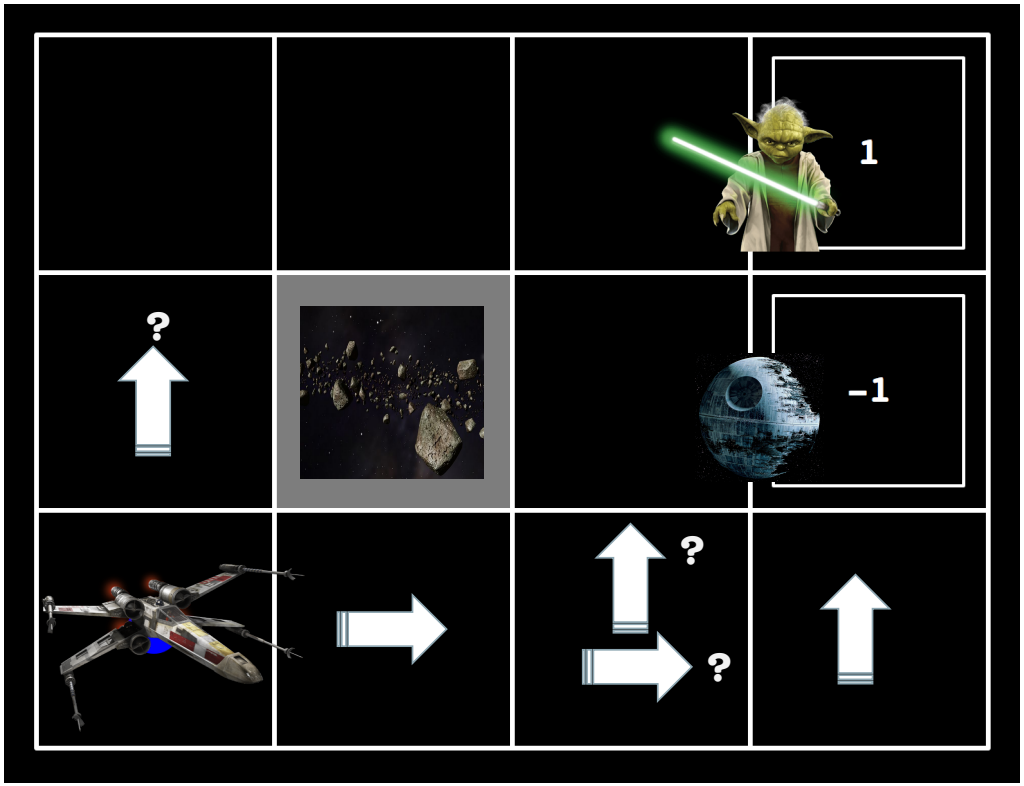
Let him step around around with an algorithm of rewards for the right choices and and penalties for the wrong ones so that it eventually learns the best strategy to win.

ORACLE

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

OUR GRIDWORLD

R=?



Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

The Bellman Equation

(The somebody who invented Reinforcement Learning was Richard Bellman @ 1957)
(And you think you are learning something new ☺)

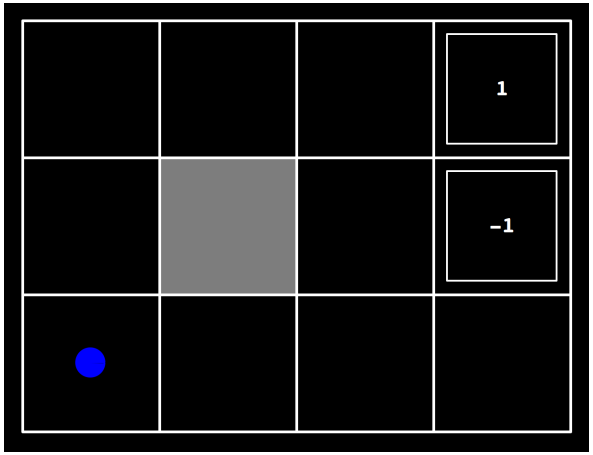
Richard Bellman called it a Policy Function, not the Bellman Equation.
It is also called the Dynamic Programming Equation.
It is (was) a mathematical optimization method.

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |

Reinforcement Learning – Formula #1 (Bellman Optimization Equation)

s, s' $(s \rightarrow s')$	State (current state and next state)
$V(s)$	Value at current state
a	Action, decision (direction)
$R(s,a)$	Reward of an action at current state (from current state)
$P(s,a,s')$	Probability of action from current state to next state

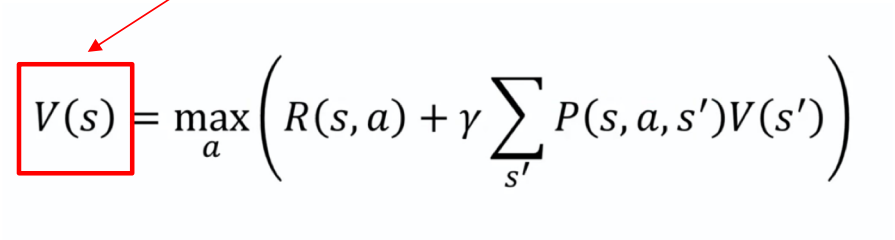
$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} P(s,a,s')V(s') \right)$$



Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Reinforcement Learning – Formula #1 (Bellman Optimization Equation)

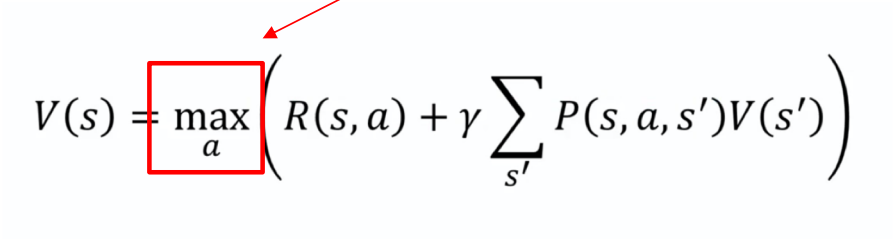
Utility, *value, or expected benefit at current state,*



Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Reinforcement Learning – Formula #1 (Bellman Optimization Equation)

*Utility, value, or expected benefit at current state, equivalent to the **maximum over all next actions** (decisions)*



$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} P(s,a,s')V(s') \right)$$

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Reinforcement Learning – Formula #1 (Bellman Optimization Equation)

Utility, value, or expected benefit at current state,
equivalent to the maximum over all next actions (decisions)
*of the sum of the **reward** of taking the next action from the current state*

$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} P(s,a,s')V(s') \right)$$

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Reinforcement Learning – Formula #1 (Bellman Optimization Equation)

Utility, value, or expected benefit at current state,
equivalent to the maximum over all next actions (decisions)
of the sum of the reward of taking the next action from the current state
*and **sum across all next states of the probabilities** of taking that action*
from this state to next state x the utility of the next state,

$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} P(s,a,s')V(s') \right)$$

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Reinforcement Learning – Formula #1 (Bellman Optimization Equation)

Utility, value, or expected benefit at current state,
equivalent to the maximum over all next actions (decisions)
of the sum of the reward of taking the next action from the current state
and sum across all next states of the probabilities of taking that action
from this state to next state x the utility of the next state,
*all regulated through multiplying it with a **discount factor** (less than 1)*
to reduce the effect of future rewards less than the immediate reward.

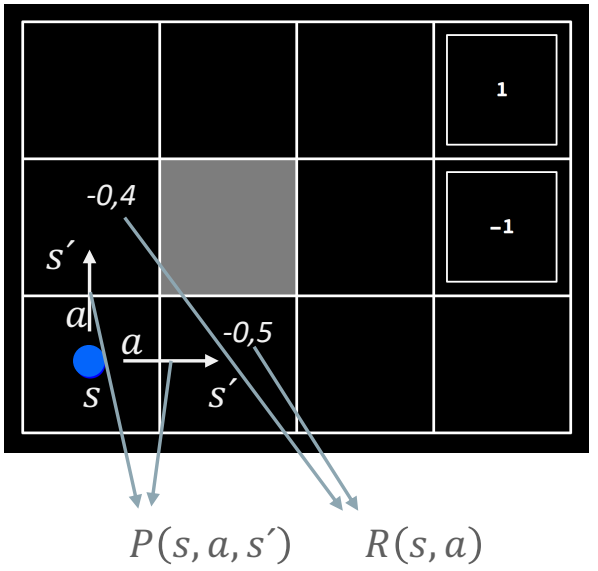
$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} P(s,a,s')V(s') \right)$$

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Reinforcement Learning – Formula #1 (Bellman Optimization Equation)

Utility, value, or expected benefit at current state,
equivalent to the maximum over all next actions (decisions)
of the sum of the reward of taking the next action from the current state
and sum across all next states of the probabilities of taking that action
from this state to next state x the utility of the next state,
all regulated through multiplying it with a discount factor (less than 1)
to reduce the effect of future rewards less than the immediate reward.

$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} P(s,a,s')V(s') \right)$$



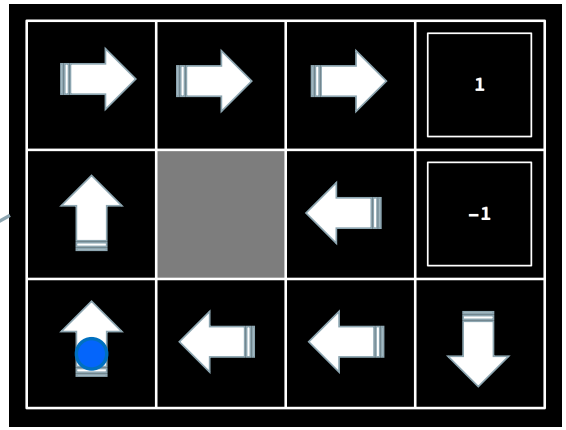
Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

GRIDWORLD

Behavior with R

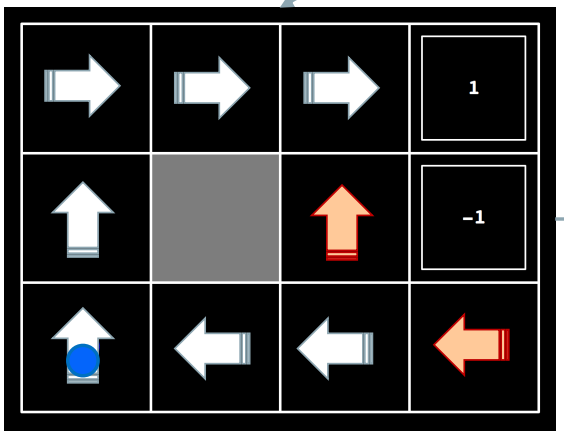
(Living Penalty)

R(s) = 0

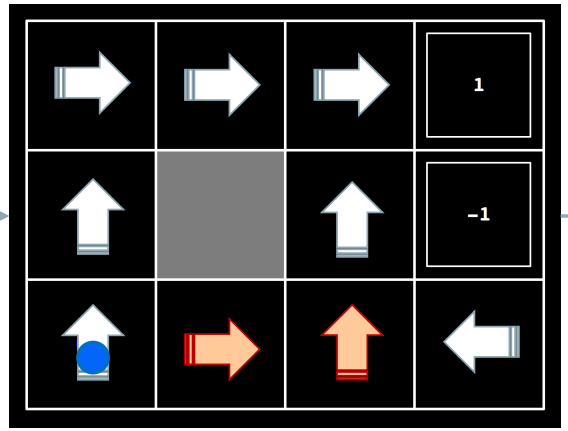


Penalty far too high for
wandering about!
Be suicidal!
Jump through fire!

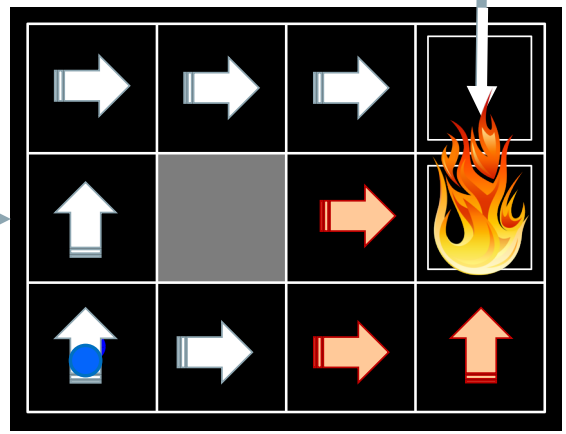
R(s) = -0,4



R(s) = -0,5



R(s) = -2,0



Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Q-Learning

(Just as you get the grasp of the Bellman Equation,
you realize that that is good but the hi-tone is missing…)

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |

Static
Pre-coded “rules”
Deterministic:

Bellman Equation

Plan

Dynamic
Data-based
Stochastic:

Markov Decision Processes (MDP)

Not everything is under the agent´s control.
MDP is Bellman++

Policy

Markov Property: Future depends only upon present state, not how the agent got there

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Q-learning

(Q for "quality" or "que")

Introduce stochasticity. Look at the value (or quality) of the **action**.

$$Q(s,a) = R(s,a) + \gamma \underbrace{\sum_{s'} P(s,a,s')V(s')}$$

Still an expected value.
Replace $V(s')$

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s,a,s') \max_{s'} Q(s',a')$$

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Temporal Difference (TD)

The value is no longer calculated but obtained empirically by walking around (training). The Q-value changes from one run to the next. There is a “temporal difference”.

The “next” $Q(s,a)$

$$TD(s,a) = R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)\Big)$$

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha\, TD_t(a,s)$$

We are adjusting (learning) our Q-value!

With respect to the ”old” $Q(s,a)$

And this indicates how quickly we are learning (our learning rate).

Quiz: What does it mean when learning rate = 0?

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

The animals of the show

(Python, Spyder and Anaconda)

(It is difficult to believe but they are tame. Or tameable.)

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.
Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |

Python Spyder Anaconda

Let´s have a Brief look

From the meetup site: Download and install **Python 2.7**, preferably with a proper IDE and a toolset that makes life easier. Many use **Anaconda Navigator** as a framework of Python tools and tutorials. It comes withe a popular Python IDE called **Spyder** (used in many Python and Scientific Computing courses at the University of Oslo).

You can download it here:
https://www.anaconda.com/download/

Python 3.6 is the newer alternative, but then you have to convert the code.
Use **2to3** and patience (`print`, exceptions and `tkinter` won´t convert automatically).

Managing environments in Anaconda: https://conda.io/docs/user-guide/tasks/manage-environments.html#activate-env
GitHub Pycolab: https://github.com/deepmind/pycolab

ORACLE

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Gridworld

(Let´s play around a bit)

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.
Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |

Gridworld (Berkeley)

Start first in manual mode and try out: `python gridworld.py -m`
(Use *up*, *down*, *left* and *right* keys to navigate the agent)

Get help: `python gridworld.py -h`

Try the following:

```
-d DISCOUNT, --discount=DISCOUNT
        (Discount on future. Default 0.9)
 -r R, --livingReward=R
        (Reward for living for a time step. Default 0.0)
-i K, --iterations=K
        (Number of rounds of value iteration. Default 10)
-s S, --speed=S
        (Speed of animation, S>1.0 is faster, 0.0<S<1.0 is slower. Default 1.0)
-v, --valueSteps
        (Display each step of value iteration)
```

UC Berkeley – Introduction to AI: http://ai.berkeley.edu/home.html
Project 3, Reinforcement Learning: http://ai.berkeley.edu/reinforcement.html#Q4

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Homework

(Fun stuff we do not have time for)

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.
Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |

Try the Berkeley exercise

(in http://ai.berkeley.edu/reinforcement.html)

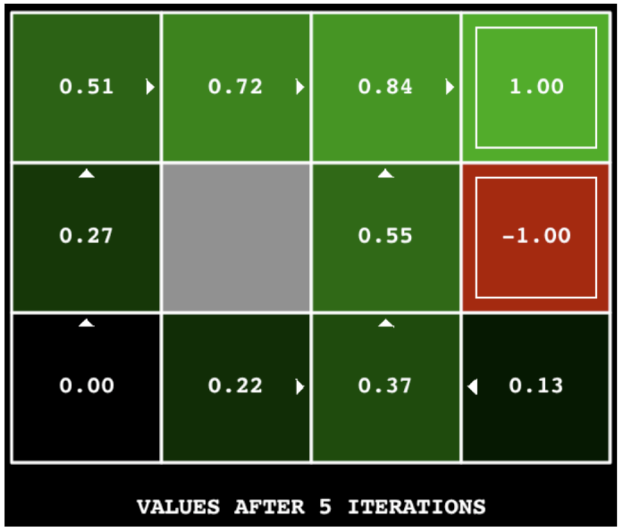
Question 1 (6 points): Value Iteration

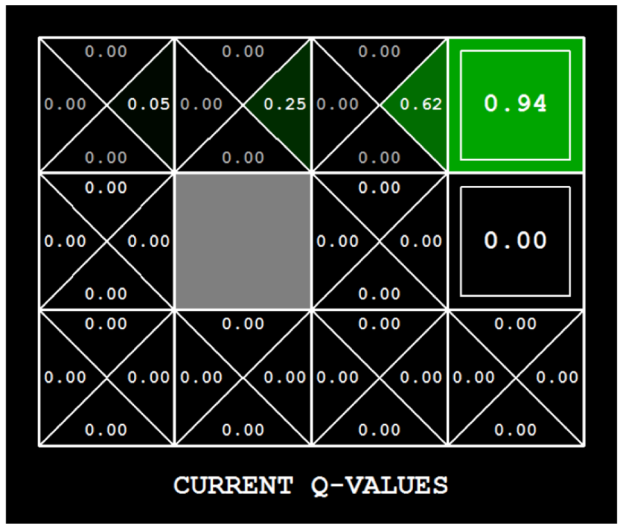
Write a value iteration agent inValueIterationAgent (partially specified in invalueIterationAgents.py). We´ll synthesize a current value reflecting the next ***k*** rewards.

- **computeActionFromValues(state)** computes the best action according to the value function given byself.values.
- **computeQValueFromValues(state, action)** returns the Q-value of the (state, action) pair given by the value function given byself.values.

These quantities are displayed in the GUI: values arenumbers in squares, Q-values are numbers in square quarters, andpolicies are arrows out from each square.

(Read the full text)





Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Q & A

(and goodbye on a kind note and a recommendation ☺)

Artificial Intelligence A-Z™: Learn How To Build An AI https://www.udemy.com

BEST SELLER in Artificial Intelligence | Business

Hadelin de Ponteves• AI Entrepreneur

Combine the power of Data Science, Machine Learning and Deep Learning to create powerful AI for Real-World...

120 lectures 17 hours All Levels English [Auto-generated]

kr150 ~~kr2,410~~

4.4 (3,174 ratings)

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.
Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |

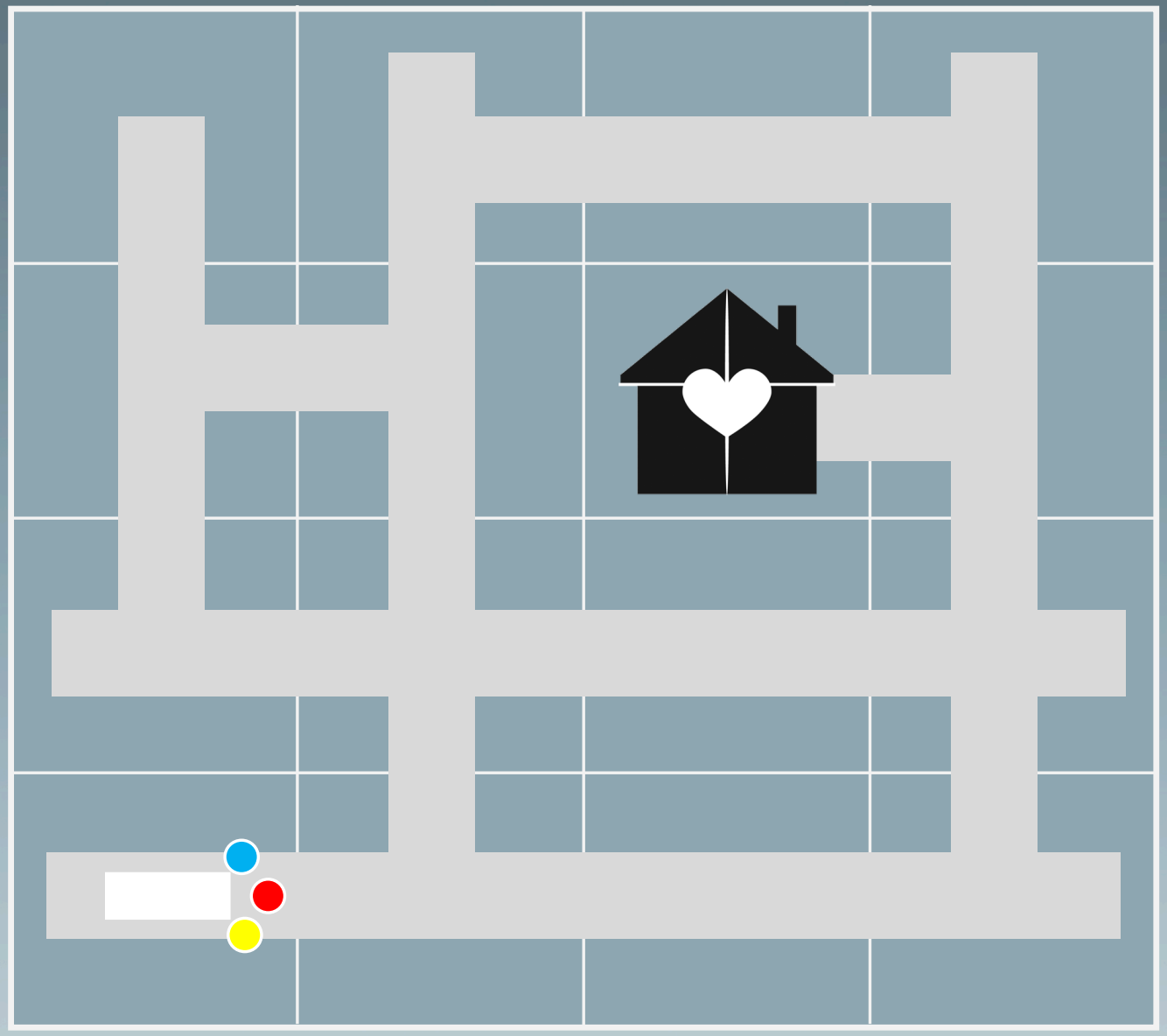
Reinforcing the path II

TechAngelist Meetup

In order of appearance:

The Force (always there)
Dr. M. Naci Akkøk (me)
Master Yoda
Some animals like
Python & Anaconda
And now a "deep" car with a deeper learning

April 18th 2018 @17:30 – 19:00



ORACLE

Recommended Tools (animals, fruits)

(If you want to try out the code)

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.
Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |

Python
Spyder
Anaconda
PyTorch
Kivy

Let´s have a Brief look

In addition to **Python 2.7**, download and install the two important libraries used in the examples.

PyTorch (Tensor-library, like TensorFlow):

- Choose your environment and download from http://pytorch.org/:
- If you´re new to tensors, you should ABSOLUTELY go through the tutorials first. Start here: http://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html

Kivy (The application framework library):

- Download from here: https://kivy.org/#download
- The intro examples and tutorials are simple and straightforward. Go to the home page and try things out and/or access tutorials/documentation: https://kivy.org/#home

ORACLE

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

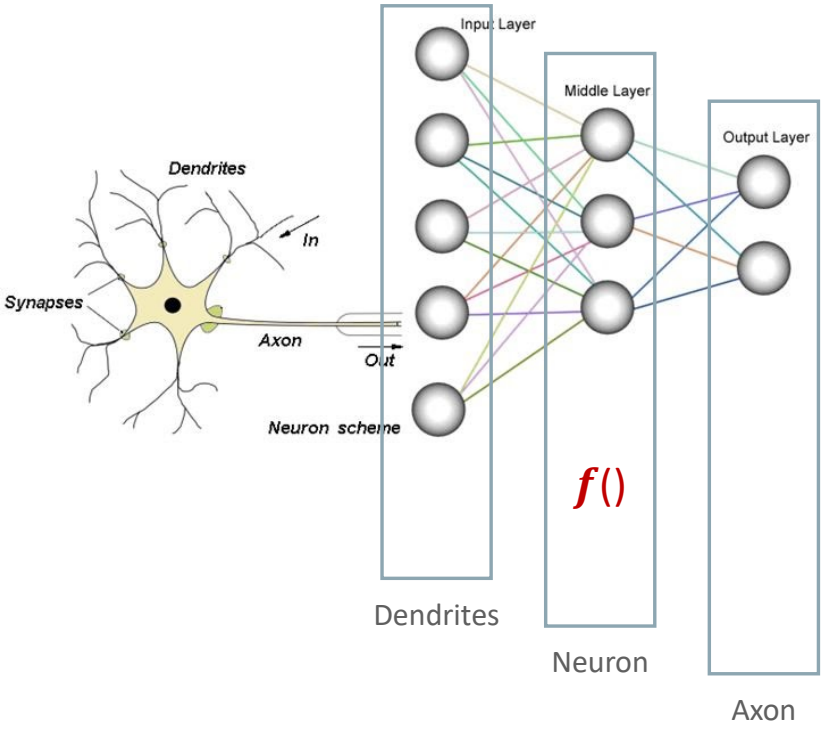
Enough work.
Let us start the journey home…

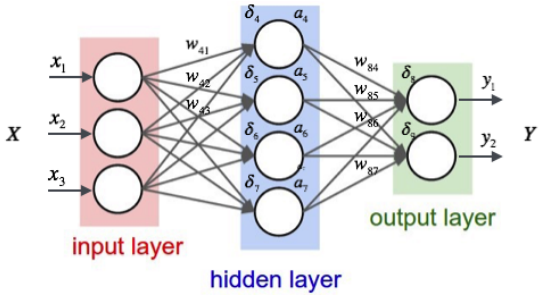
(Through the tangle of a neural network)

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.
Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |

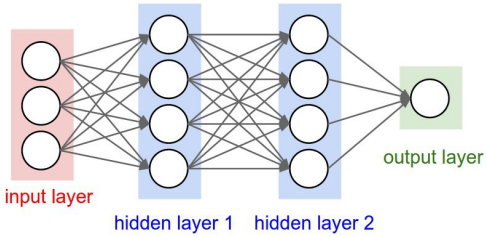
Shallow, deep, deeper: (Artificial) Neural Networks

See for example: http://neuralnetworksanddeeplearning.com/chap1.html **by M. Nielsen**
or a lighter summary: https://medium.com/technology-invention-and-more/everything-you-need-to-know-about-artificial-neural-networks-57fac18245a1
or (recommended) with code tutorial: http://pytorch.org/tutorials/beginner/blitz/neural_networks_tutorial.html

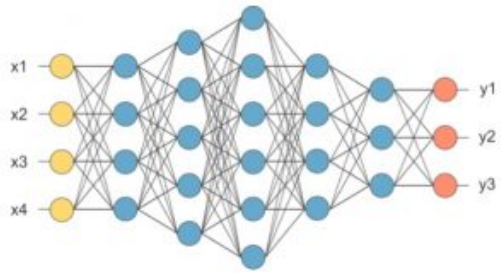




Shallow: 1 hidden layer.
(Note several outputs)



Deep: 2 hidden layers.
(Note one output)

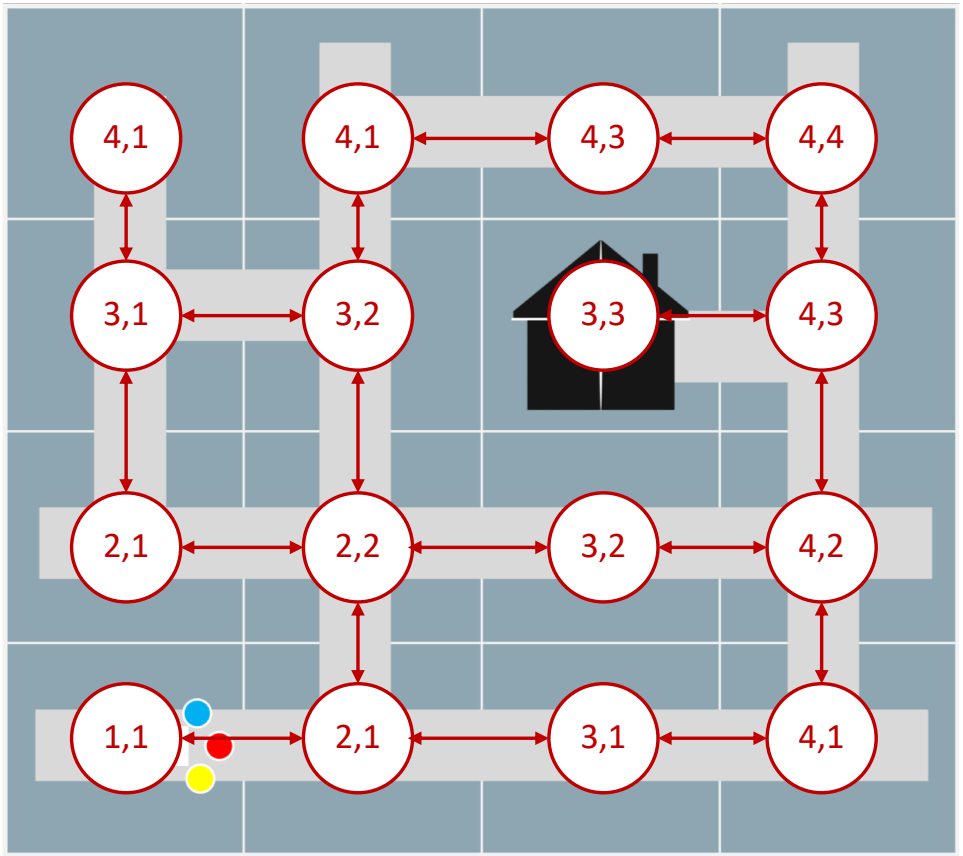


Deeper: 5 hidden layers.
(Note three outputs)

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Q-learning: Our grid-world (or a bi-directional graph)!

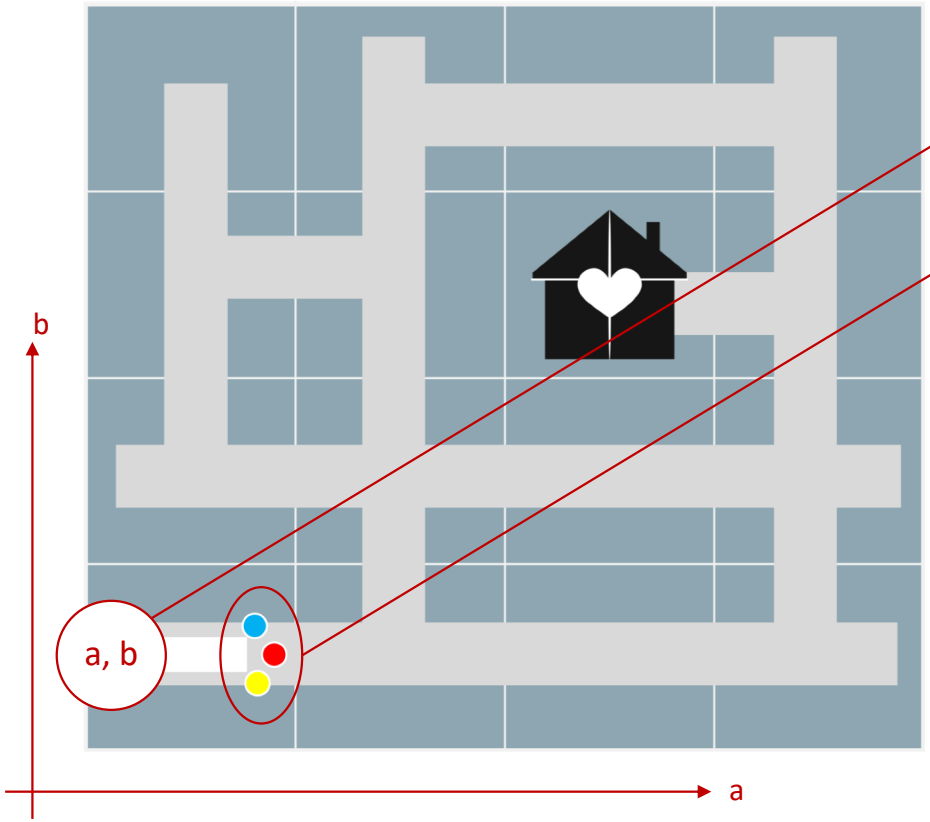
For a nice summary of Q-learning and A* path optimization, see: http://mnemstudio.org/path-finding-q-learning-tutorial.htm



Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Deep Q-learning: Add a neural network!

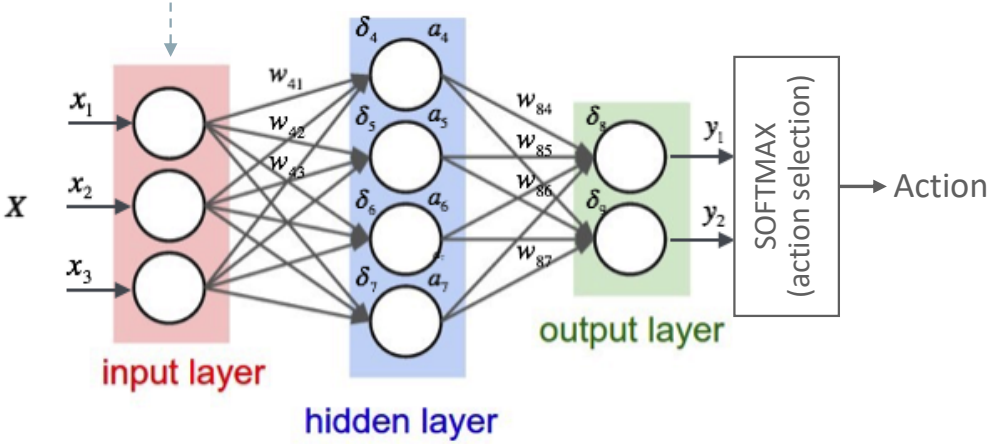
For a nice summary of Q-learning and A* path optimization, see: http://mnemstudio.org/path-finding-q-learning-tutorial.htm



State & Reward (weights) @ (a, b)

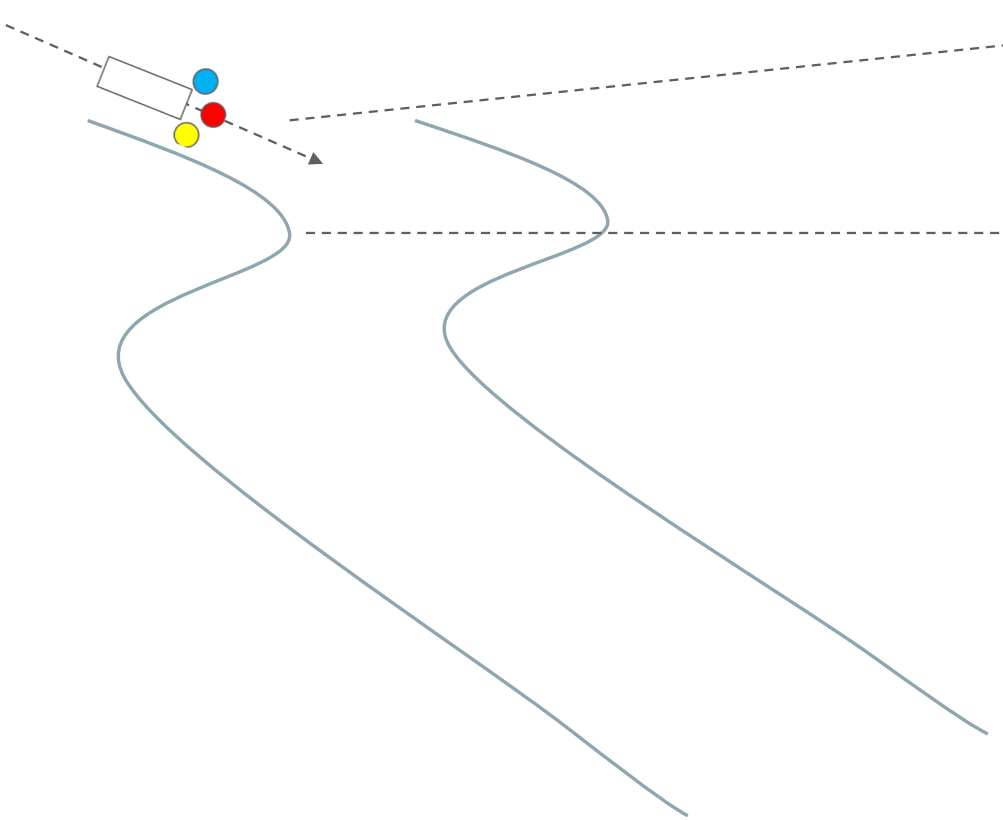
INPUT:
Sensors 1, 2 & 3 @ (a, b), reflects experience: Density of the sand

Vector/Tensor @ (a, b)



Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Important Principle: Never Stop Exploring!



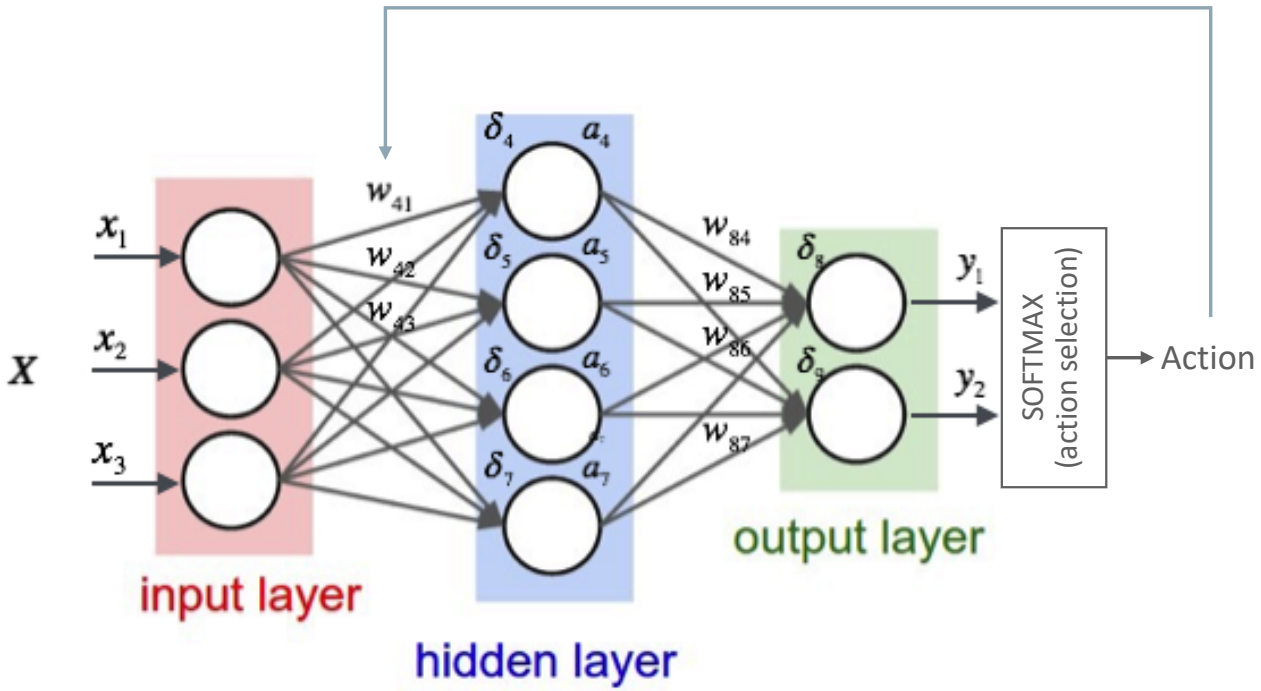
Car travels without major change in the input.
It will learn that it is doing very well: False perception!

A curve may come!
What to do?

- We do not want to bias the learning due to monotonous learning.
- Instead we store experiences in a “memory” and select a randomly (for example uniformly) distributed number of experiences to learn from:
- We “break habit”.
- Called Experience Replay

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Learning: Backwards Propagation



Learning:

Calculate "cost" as difference between actual and predicted Q values: **Re-evaluations**.

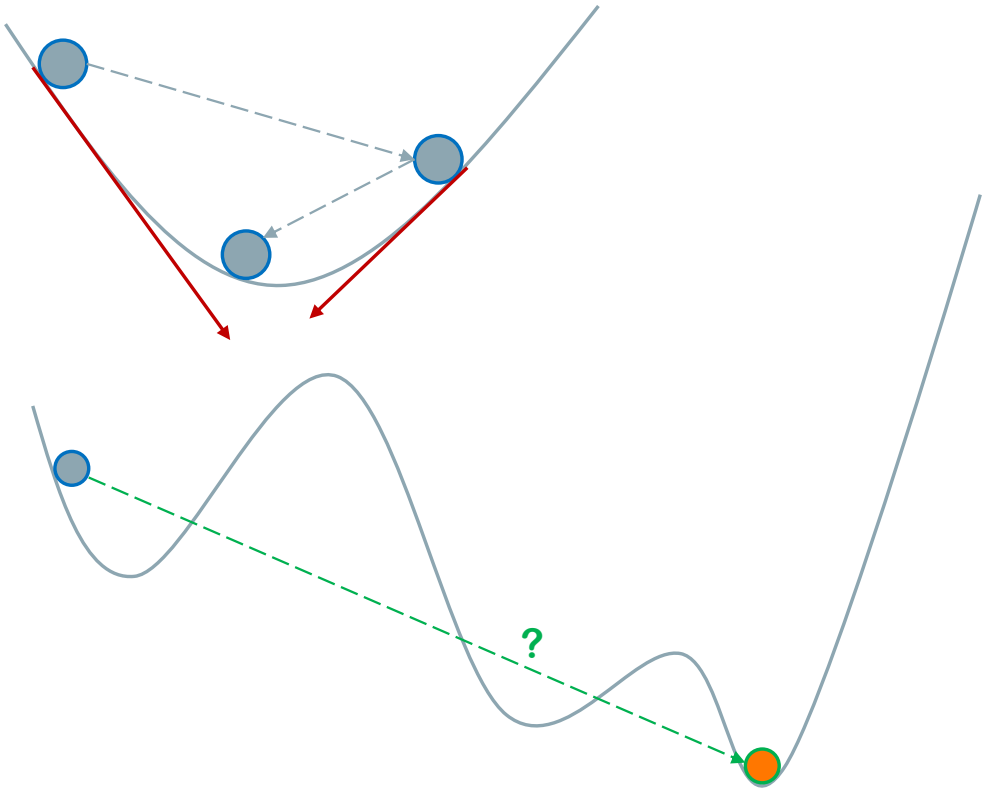
Update weights to look for the minimum cost.

But that is a lot of comparisons. In very large networks even the fastest computers will take more than a life time to do that.

What to do?

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Learning w/Backwards Propagation: Finding the minimum cost (Gradient Descent)



Calculating the minimum by comparing every single value in a deep neural network is going to take ages.

We use **Gradient Descent**.

But we risk finding a local minimum, not the actual minimum!

Again, we **introduce some randomness** so as to avoid getting stuck in one locality. We jump form location to location randomly some percentage of the time.

Remember: Never stop exploring!

ORACLE

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

A beautiful AI/ML cloud platform you can try out for free.

(There should be a pamphlet somewhere around here)

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.
Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |

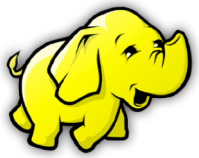
AI/Machine learning platforms and tools @ORACLE

ORACLE INTELLIGENT BOTS | ORACLE DATA VISUALIZATION | ORACLE POLICY AUTOMATION

ORACLE PL/SQL | R (ORE) | (Python v/Spark)

ORACLE Advanced Analytics (OAA)

Structured Tables
Corporate & Legacy data



Unstructured
Video/audio
Often external

ORACLE R Advanced Analytics for Hadoop (ORAAH)

ORACLE Big Data Connectors

ORACLE Big Data SQL

ORACLE AI Platform Cloud Service

ORACLE Semantic Graph & Property Graph w/built in analytics & intelligence

ORACLE Spatial, Network Model

ORACLE

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

ANNOUNCING Oracle AI Platform Cloud Service

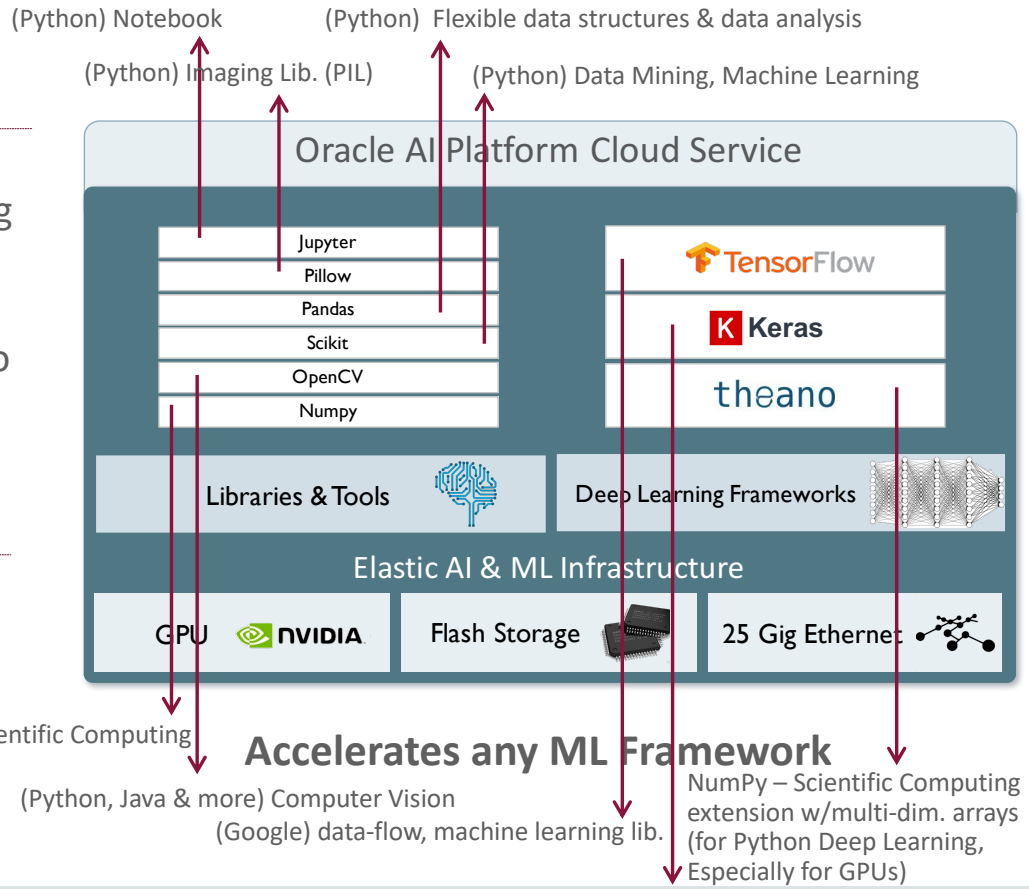
Simplify Application Development Using AI

Complete AI Development Platform

- Quick set up of auto-scaling AI development system
- Pre-installed AI libraries, tools, samples, deep learning frameworks
- OpenCL, Nvidia CUDA and cuDNN drivers
- Integrated with Oracle Object Cloud, easily connect to existing Hadoop cluster
- Seamlessly build AI-powered apps across Oracle PaaS

Best-in-Class AI Compute Infrastructure

- GPU Bare-Metal shape: 2x Tesla P100 GPUs based on NVIDIA's Pascal Architecture
- Pre-Announcing Volta GPUs with up to 8 GPUs
- NVMe High Speed Flash Storage
- 25 Gig Ethernet network



ORACLE

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

ANNOUNCING CONVERSATIONAL AI PLATFORM

Enhance Experiences with Conversational Interactions

SUPPORT FOR ALL MAJOR CHAT AND VOICE MESSAGING

- Native integration for FB Messenger, Slack, WeChat
- Native integration to Alexa, Google Home, Siri, Cortana
- Native & JavaScript SDK for Native mobile apps & Web

BREADTH OF AI SERVICES

- Deep learning algorithms fine tuned for conversational UI
- Cognitive services across speech, vision and language
- Dialog & context services to create great user experiences

DEEP MACHINE LEARNING FOR CONVERSATIONAL INTERACTION

- Usage and user experience insights; operational insights

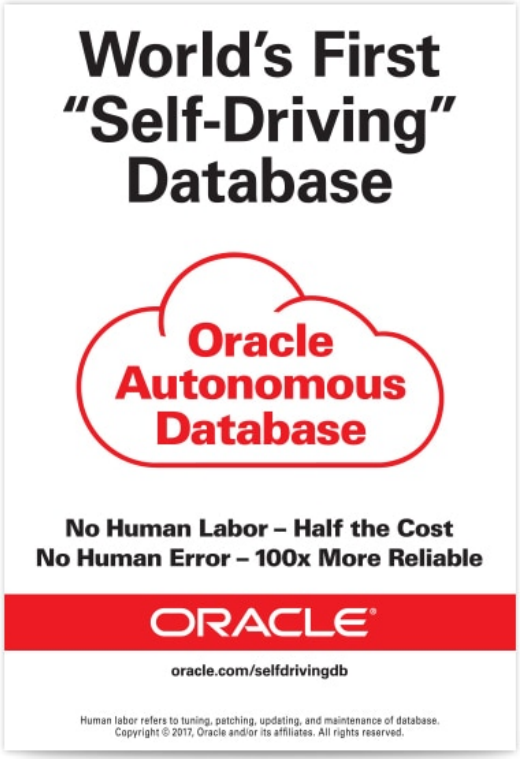
Confidential – Oracle Internal/Restricted/Highly Restricted

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

AI / ML is built into the tools!

ORACLE®

DATA VISUALIZATION



ORACLE

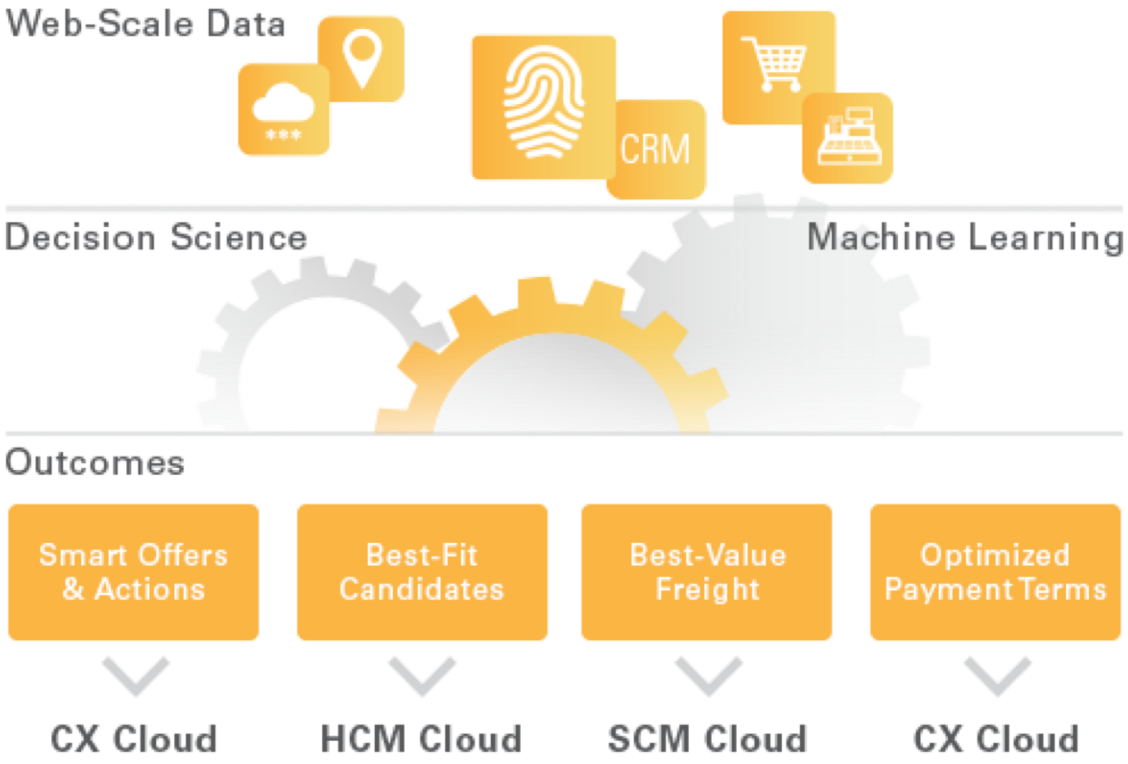
Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

AI / ML is built into the tools!

- Oracle Mobile Cloud includes conversational AI through bots to automate human interactions using natural language, sentiment, speech, images, and machine learning.
- Oracle Autonomous Database Cloud includes machine learning to enable self-driving, and self-repairing data warehousing and OLTP to improve security and reliability.
- Oracle Analytics Cloud allows analysts to use machine learning techniques to better visualize data and understand patterns without having to become a data scientist.
- Oracle Security and Management Cloud includes embedded machine learning to automate detection, prevention, and response to security breaches, performance anomalies, and vulnerabilities.

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Adaptive Intelligent Applications (AI / ML built into applications)



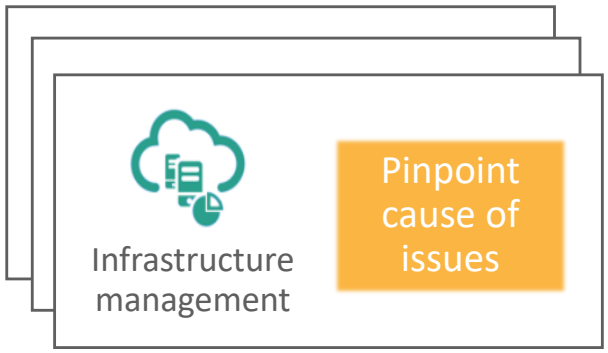
- **Data and Context-Driven**
- Process enormous amounts of web-scale data in real-time and synthesizes context to deliver the best and most personalized outcomes.
- **Action-Oriented and Omni-Channel**
- Anticipate user behaviors and deliver actionable and individualized outcomes. Data and events are tracked from everywhere so recommendations are coordinated across channels.
- **Continuously Adaptive and Self-Learning**
- Continuously improves outcomes as the system reacts, learns and adapts. There is a value exchange between the end user and the adaptive intelligent app—the more the end user interacts, the more value they receive.
- **Built for End-Users**
- Immediate value out of the box, with no need for a legion of data scientists!

ORACLE

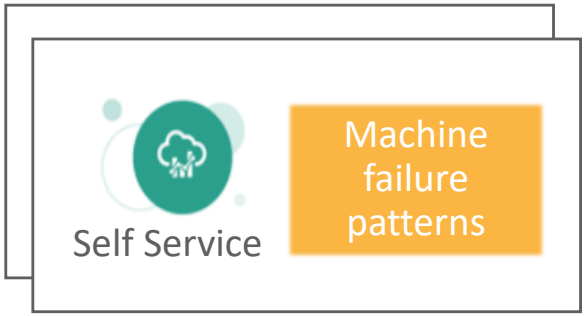
Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Machine Learning use cases and built-in ML examples

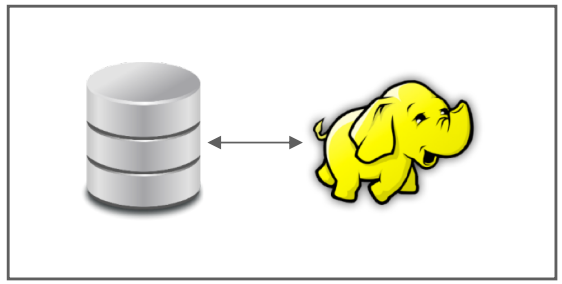
1. Standard apps



2. Business Intelligence



3. Platform
(never forget
The platform ☺)



NB! Oracle embeds AI and ML capabilities within its own business and IT services.

ORACLE

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.

Q & A

(the reference & the recommendation again ☺)
(and info if you want to contact me: naci.akkok@oracle.com)

Artificial Intelligence A-Z™: Learn How To Build An AI https://www.udemy.com

BEST SELLER in Artificial Intelligence | Business

Hadelin de Ponteves• AI Entrepreneur

Combine the power of Data Science, Machine Learning and Deep Learning to create powerful AI for Real-World...

120 lectures 17 hours All Levels English [Auto-generated]

kr150 ~~kr2,410~~

4.4 (3,174 ratings)

ORACLE

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.
Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |