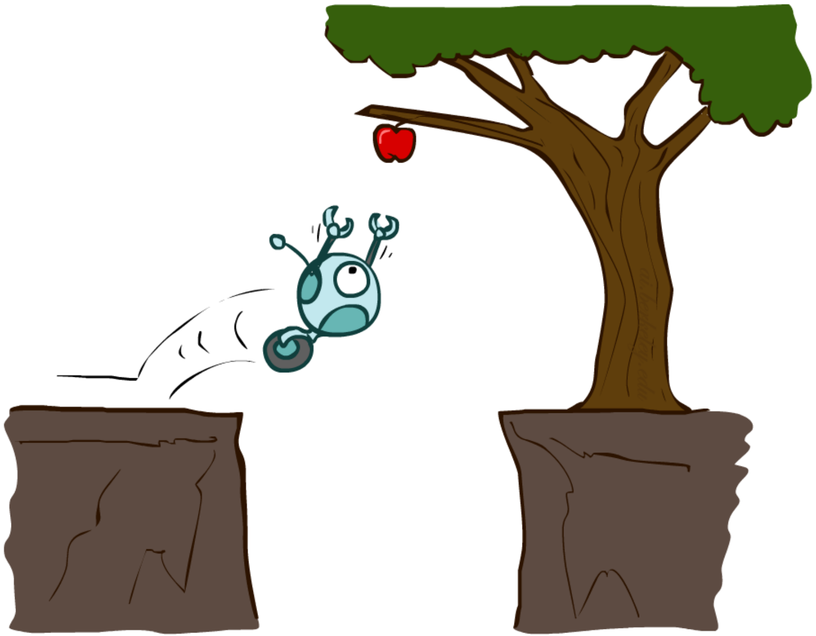
COMP1694

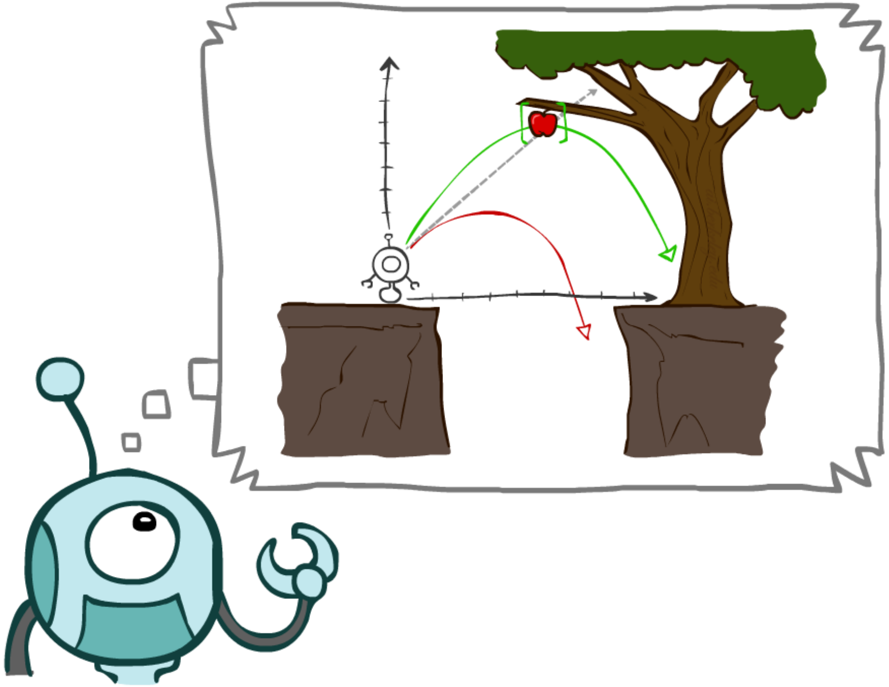
Artificial Intelligence

Lecture 9: Machine Learning

an introduction



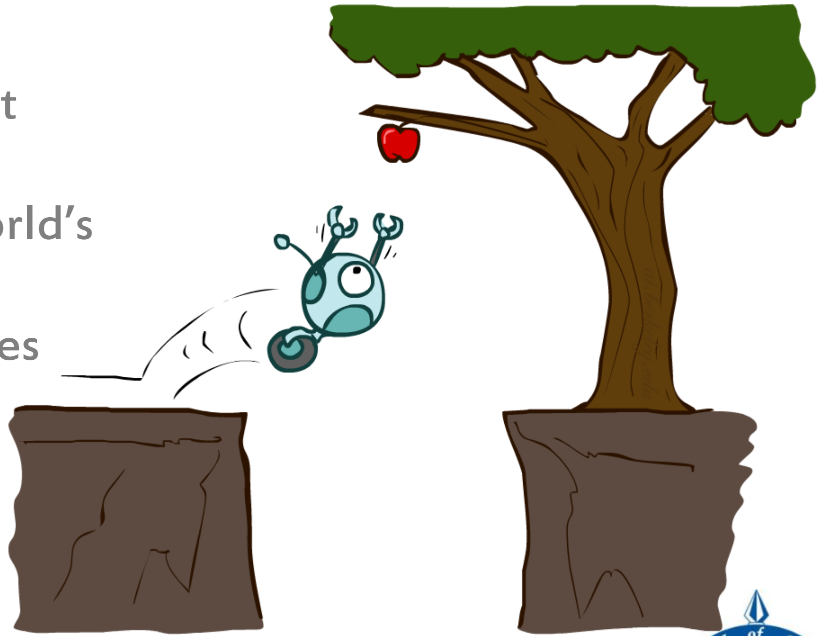
Agents that Plan



Reflex Agents

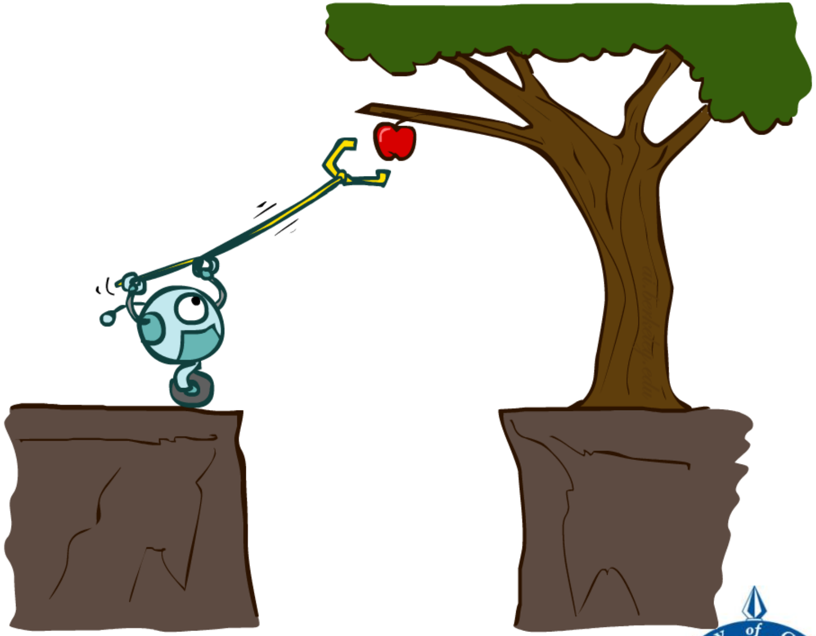
- Reflex agents:
  - Choose action based on current percept (and maybe memory)
  - May have memory or a model of the world's current state
  - Do not consider the future consequences of their actions
  - Consider how the world IS

- Can a reflex agent be rational?



Planning Agents

- Planning agents:
  - Ask “what if”
  - Decisions based on (hypothesised) consequences of actions
  - Must have a model of how the world evolves in response to actions
  - Must formulate a goal (test)
  - Consider how the world WOULD BE



- Optimal vs. complete planning
- Planning vs. replanning

Alone in the world!

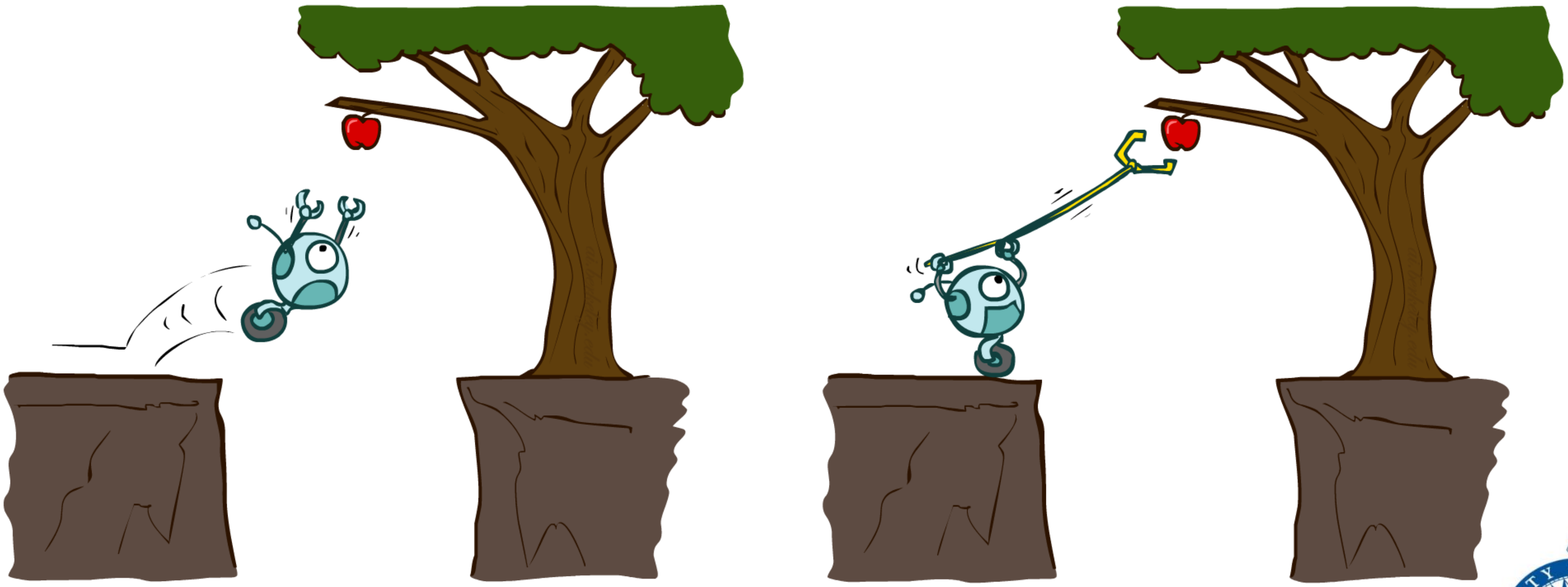


image ai.berkeley.edu

The Crawler!

Big Data

Data ≠ Knowledge

What is Machine Learning?

Machine Learning
principle

Study (design and analysis) of algorithms that

improve their performance ***P***

at some task ***T***

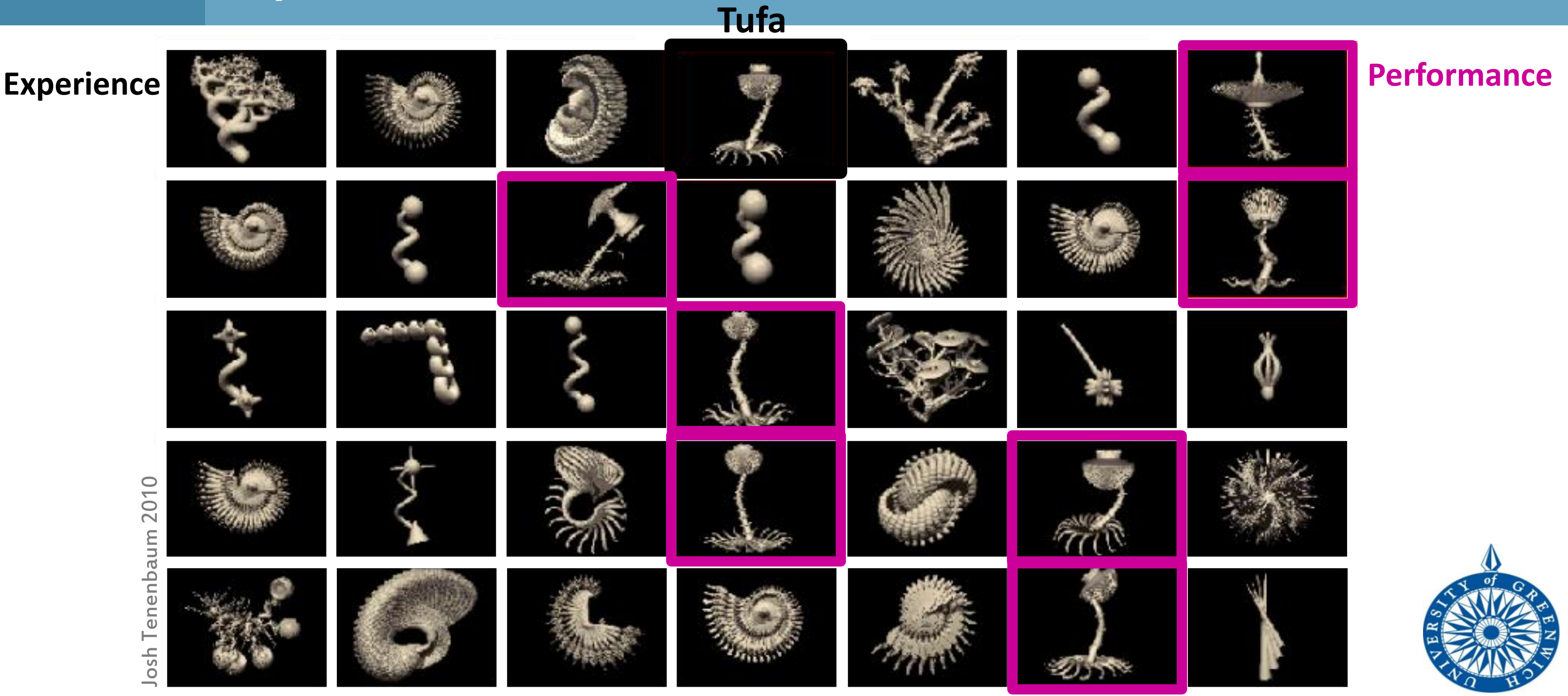
with experience ***E***

well-defined learning task: <P,T,E>

Data
experience → Learning Algorithm → Knowledge
Performance on task

Tom M. Mitchell, *Machine Learning*, McGraw Hill, 1997

Human Learning
spot the Tufas



Human Learning
what's the animal?

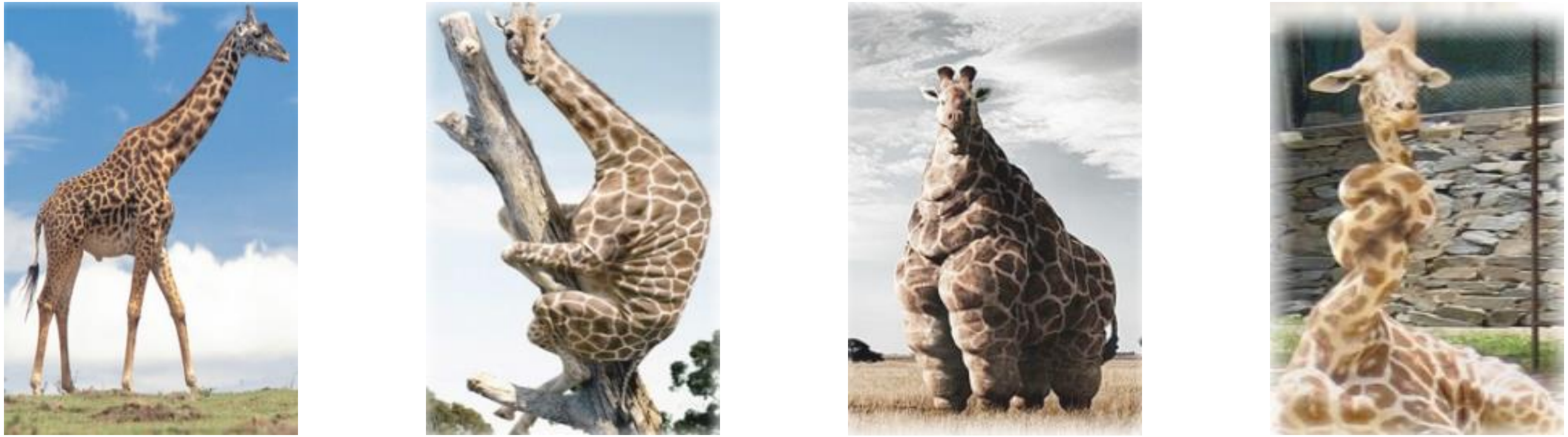


image Thomas Serre (2012), http://helper.ipam.ucla.edu/publications/gss2012/gss2012_10539.pdf

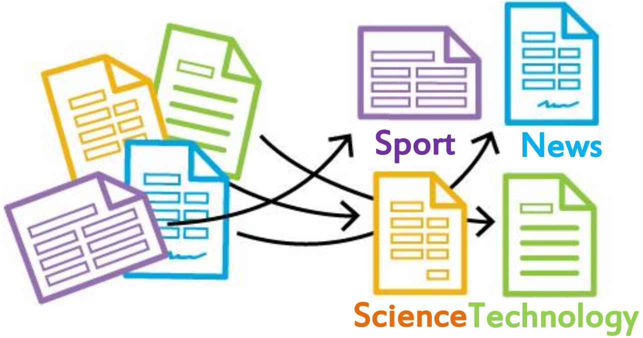
Learning is...

- Memorising something.
- Learning facts through observation and exploration.
- Improving motor and/or cognitive skills through practice.
- Organising new knowledge into general, effective representations

Learning denotes changes in a system that enable the system to do task(s) from the same projection more efficiently next time.

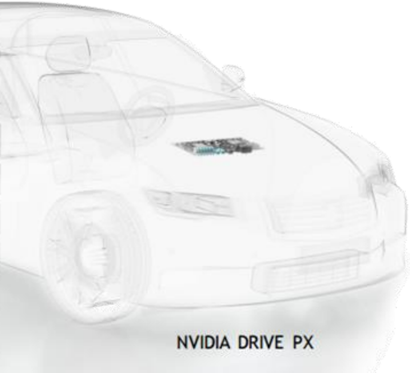
Herbert Simon, 1983

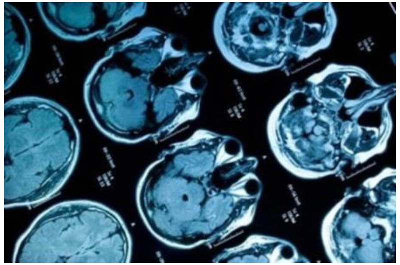
Machine Learning in Action



Google +

DeepMind
AlphaGo





Can criminal behaviour be predicted using brain scans?

University of New Mexico and Mind Research Network

Machine Learning
principle

Study (design and analysis) of algorithms that

improve their performance ***P***

at some task ***T***

with experience ***E***

well-defined learning task: <P,T,E>

Data
experience
→ Learning Algorithm →
Knowledge
Performance on task

Tom M. Mitchell, *Machine Learning*, McGraw Hill, 1997

Tasks – Experience – Performance

Kinds of Learning
popular examples...

- Supervised learning
  - given a set of example input/output pairs, find a rule that does a good job of predicting the output associated with a new input
  - training data includes both the input and desired results
- Unsupervised learning
  - given a set of examples, but no labelling of them, group them into clusters
  - no output datasets are provided, instead the data is clustered into different classes
- Reinforcement learning
  - agents interacting with the world make observations, take actions and are rewarded/punished – goal is to take actions to obtain maximum reward
  - learning from actions/reactions

Supervised Learning

Feature Space $\mathcal{X}$

Label Space $\mathcal{Y}$

Words in document

“Sports”
“News”
“Science”
“Technology”
…

Discrete Labels
Classification

Market Data up to time t

Share Price
“£ 28.4”

Continuous Labels
Regression

Task: Given $X \in \mathcal{X}$, predict $Y \in \mathcal{Y}$

Unsupervised Learning

AKA "learning without a teacher"

Feature Space $\mathcal{X}$

Words in document

Word distribution (probability of a word)

Task: Given $X \in \mathcal{X}$, learn $f(X)$

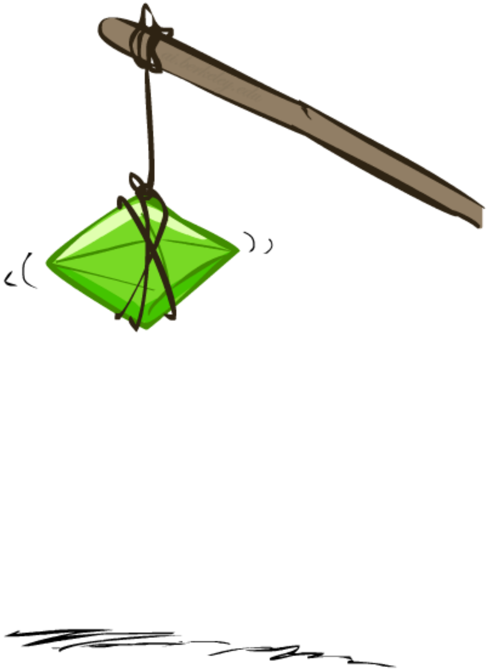
Unsupervised learning

- Clustering – group similar things eg. images [Gong, Y, et al (2015)]

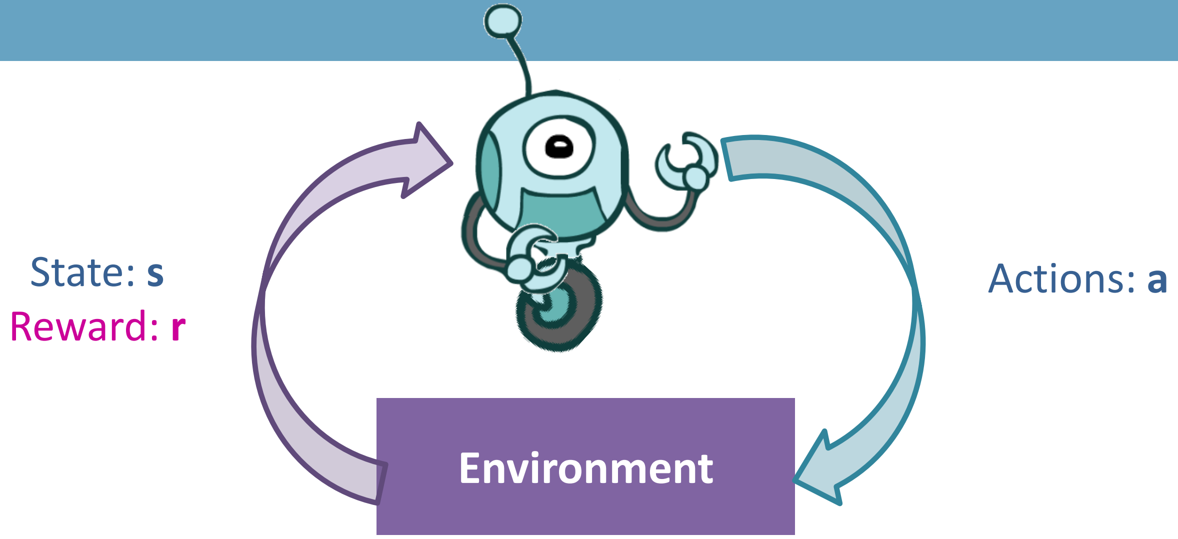
Unsupervised learning

- Dimensionality Reduction / Embedding

Reinforcement Learning



Reinforcement Learning



- **Basic idea:**
  - Receive feedback in the form of rewards
  - Agent's utility is defined by the reward function
  - Must (learn to) act so as to maximize expected rewards
  - All learning is based on observed samples of outcomes!

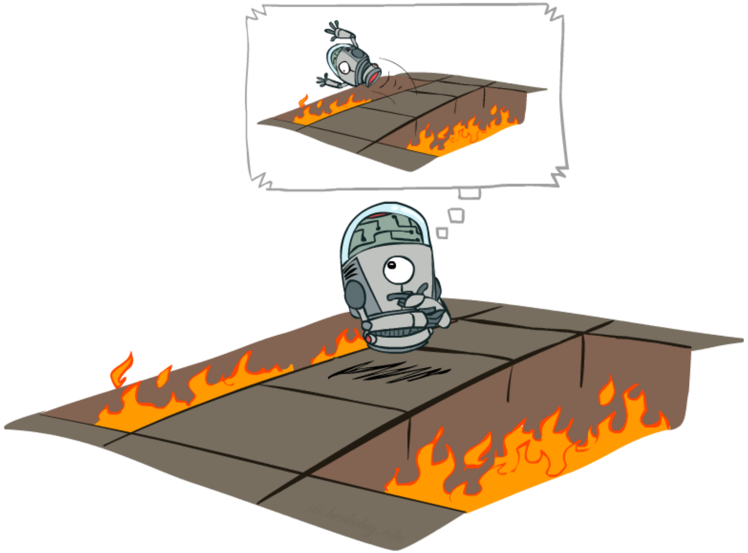
Reinforcement Learning

- Assume a Markov Decision Process (MDP):
  - A set of states $s \in S$
  - A set of actions (per state) A
  - A model T(s,a,s')
  - A reward function R(s,a,s')

- Still looking for a policy $\pi(s)$

- New twist: don't know T or R
  - I.e. we don't know which states are good or what the actions do
  - Must actually try actions and states out to learn

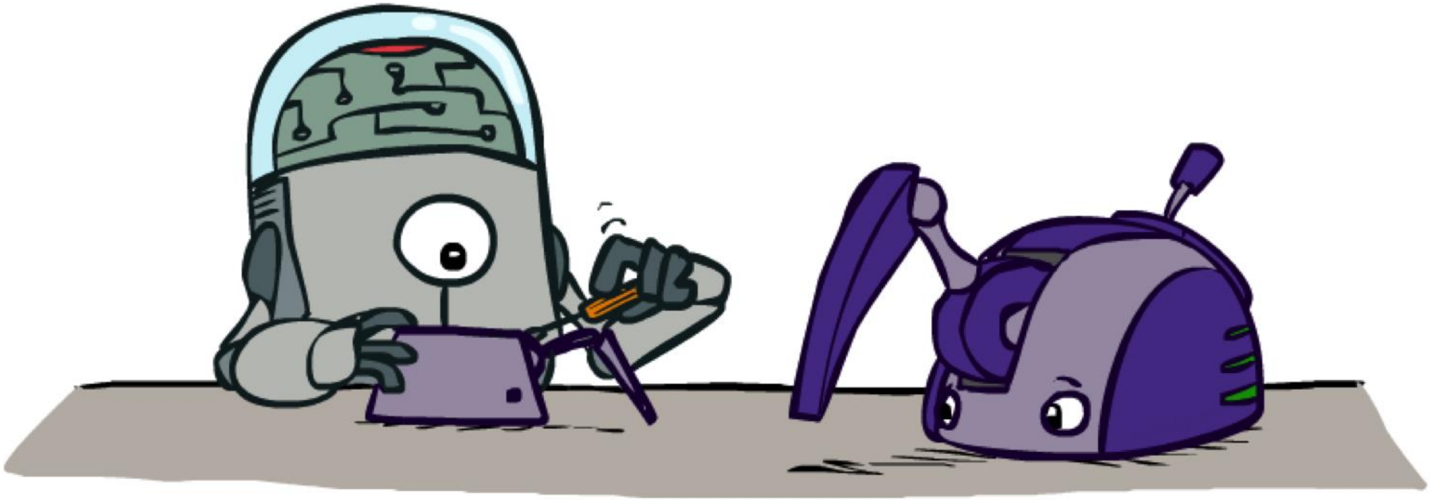
Offline (MDPs) vs. Online (RL)



Offline Solution

Online Learning

Model-Based Learning



Model-Based Learning

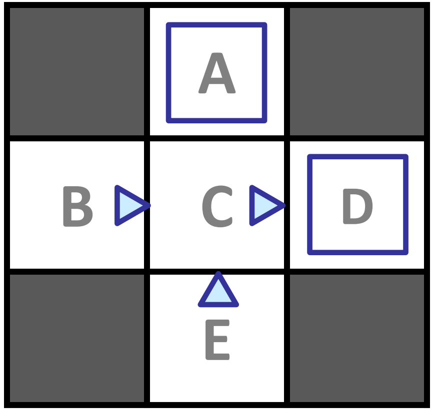
- Model-Based Idea:
  - Learn an approximate model based on experiences
  - Solve for values as if the learned model were correct

- Step 1: Learn empirical MDP model
  - Count outcomes s' for each s, a $\hat{T}(s, a, s')$
  - Normalize to give an estimate of
  - Discover each $\hat{R}(s, a, s')$ when we experience (s, a, s')

- Step 2: Solve the learned MDP
  - For example, use value iteration, as before

Example: Model-Based Learning

Input Policy π



Assume: $\gamma = 1$

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Learned Model

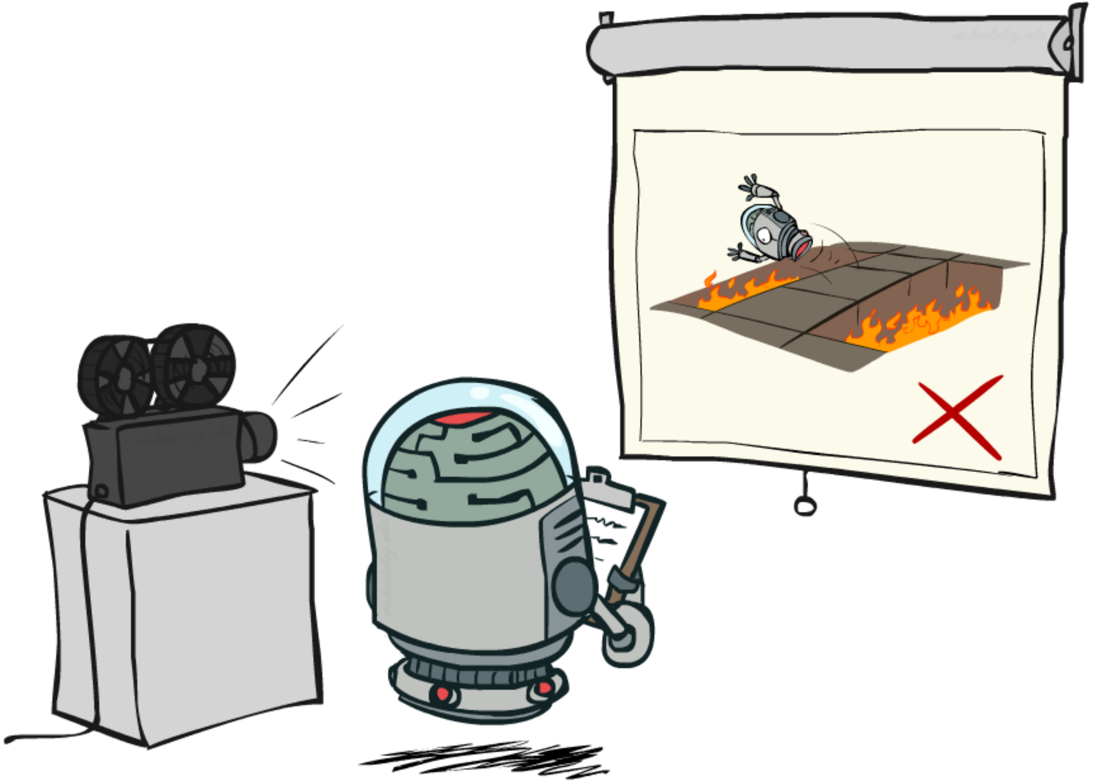
$\hat{T}(s, a, s')$

T(B, east, C) = 1.00
T(C, east, D) = 0.75
T(C, east, A) = 0.25
…

$\hat{R}(s, a, s')$

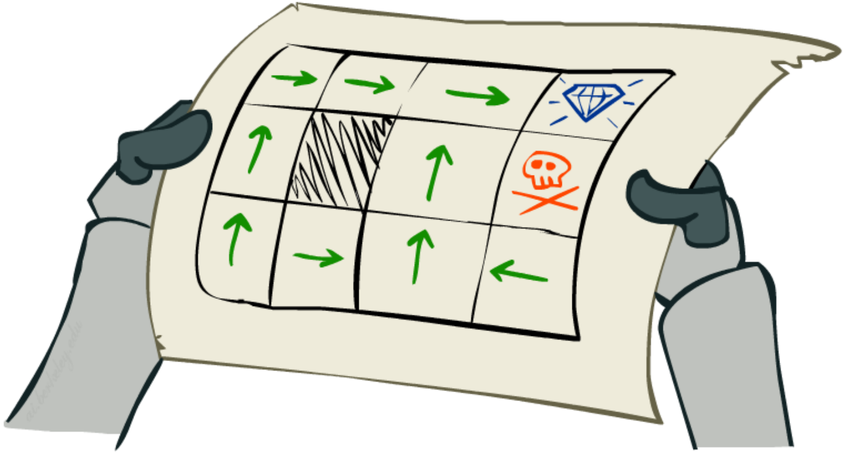
R(B, east, C) = -1
R(C, east, D) = -1
R(D, exit, x) = +10
…

Passive Reinforcement Learning



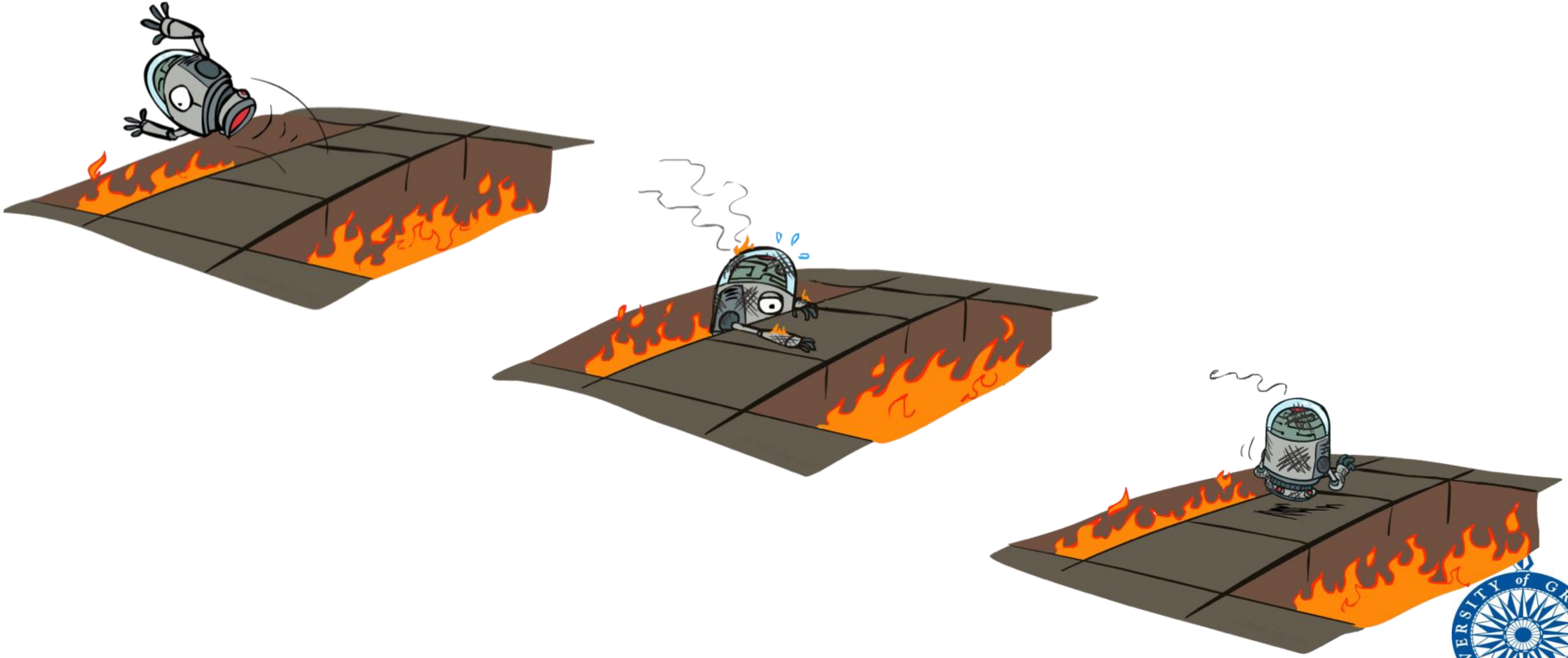
Passive Reinforcement Learning

- Simplified task: policy evaluation
  - Input: a fixed policy $\pi(s)$
  - You don't know the transitions T(s,a,s')
  - You don't know the rewards R(s,a,s')
  - Goal: learn the state values



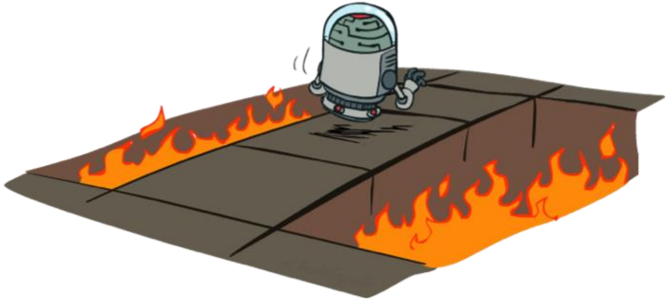
- In this case:
  - Learner is "along for the ride"
  - No choice about what actions to take
  - Just execute the policy and learn from experience
  - This is NOT offline planning! You actually take actions in the world.

Active Reinforcement Learning



Active Reinforcement Learning

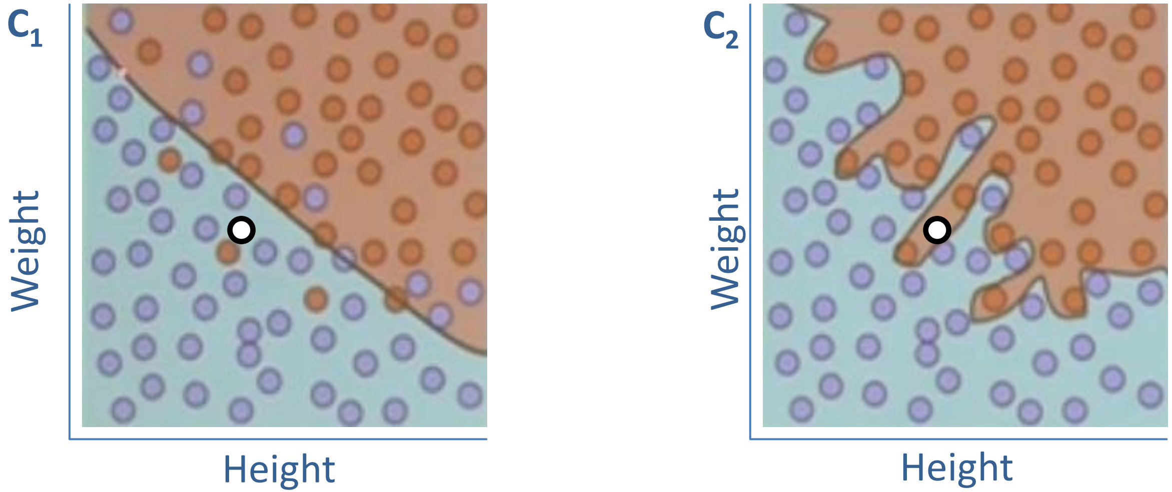
- Full reinforcement learning: optimal policies (like value iteration)
  - You don't know the transitions T(s,a,s')
  - You don't know the rewards R(s,a,s')
  - You choose the actions now
  - Goal: learn the optimal policy / values



- In this case:
  - Learner makes choices!
  - Fundamental tradeoff: exploration vs. exploitation
  - This is NOT offline planning! You actually take actions in the world and find out what happens…

Tasks – Experience – Performance

Training Data vs Test Data



Training Data

- Footballer
- Not

Test Data

- A good ML algorithm:
  - does not **overfit** training data (classifier2)
  - **generalises** well to test data

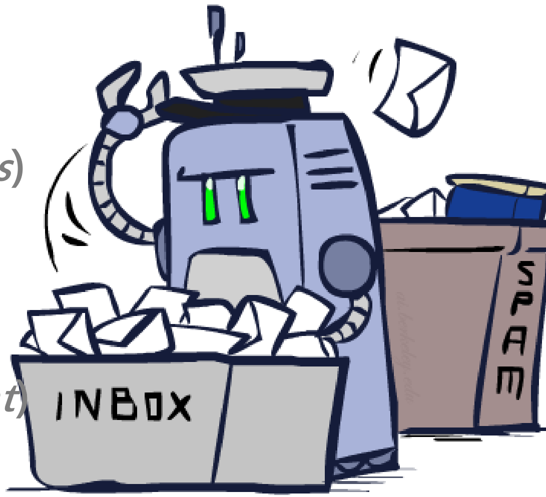
Tasks – Experience – Performance

Performance Measures

- For a random test data, measure the closeness between true label Y and prediction $f(X)$
  - Binary classification : $\text{Loss}(Y, f(X)) = 1_{\{f(X)) \neq Y\}}$ 0 or 1 loss
  - Regression: $\text{Loss}(Y, f(X)) = (f(X)) - Y)^2$ Square loss

Recap

- Machine learning deals with the problem of extracting ***features*** from data so as to solve many different ***predictive*** tasks:
  - Decision making (*e.g. robotics, games, compiler tuning, trading*)
  - Forecasting (*e.g. Energy demand prediction, finance*)
  - Imputing missing data (*e.g. Netflix recommendations*)
  - Detecting anomalies (*e.g. Security, fraud, virus mutations*)
  - Classifying (*e.g. Credit risk assessment, cancer diagnosis*)
  - Ranking (*e.g. Google search, personalization*)
  - Summarizing (*e.g. News zeitgeist, social media sentiment*)



When to apply machine learning

- Human expertise is absent (*e.g. Navigating on Mars*)
- Creating rational NPCs and agents (e.g. virtual reality games)
- Humans are unable to explain their expertise (*e.g. Speech recognition, vision, language*)
- Solution changes with time (*e.g. Tracking, temperature control, preferences*)
- Solution needs to be adapted to particular cases (*e.g. Biometrics, personalisation*)
- The problem size is to vast for our limited reasoning capabilities (*e.g. Calculating webpage ranks, matching ads to facebook pages*)

Further Reading
and interesting resources

- Learning
  - Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig, 3rd Edition, Chapter 5, Section 18
  - eBook available here: http://cessa.khu.ac.ir/wp-content/uploads/2015/12/Artificial-Intelligence-A-Modern-Approach-3rd-Edition.pdf
- Reinforcement learning
  - Russell and Norvig, Chapter 5 section 21
  - https://webdocs.cs.ualberta.ca/~sutton/book/ebook/the-book.html

Further Reading
and interesting resources

- Tom Mitchell's book on Machine Learning 1997
  - (things have moved on since then, but a good read) http://personal.disco.unimib.it/Vanneschi/McGrawHill_-_Machine_Learning_-Tom_Mitchell.pdf
- Machine learning in action
  - DeepMind: https://www.youtube.com/watch?v=TnUYcTuZJpM
  - AlphaGo, Mastering the game of Go with Deep Neural Networks & Tree Search: https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf
  - Google DeepMind playing Breakout, Pac-Man
- **AiGameDev - Community** of game developers focusing on AI http://aigamedev.com/

Further Reading
useful APIs

- WEKA
  - **Weka** is a collection of machine learning algorithms for data mining tasks
    http://www.cs.waikato.ac.nz/ml/weka/