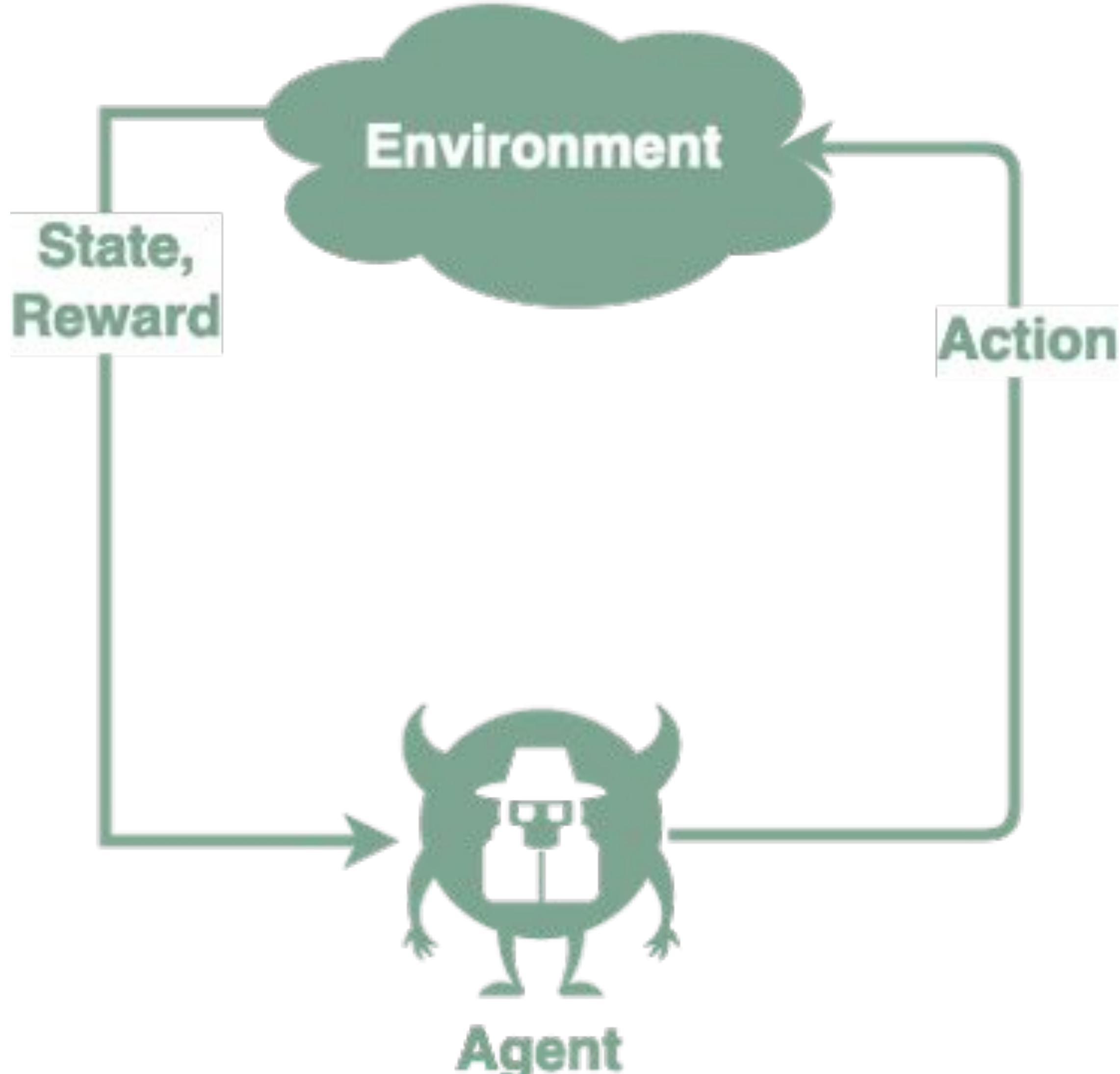
Reinforcement Learning

Timotheus Kampik

SIGNAVIO

www.signavio.com

What is RL?



What is RL?

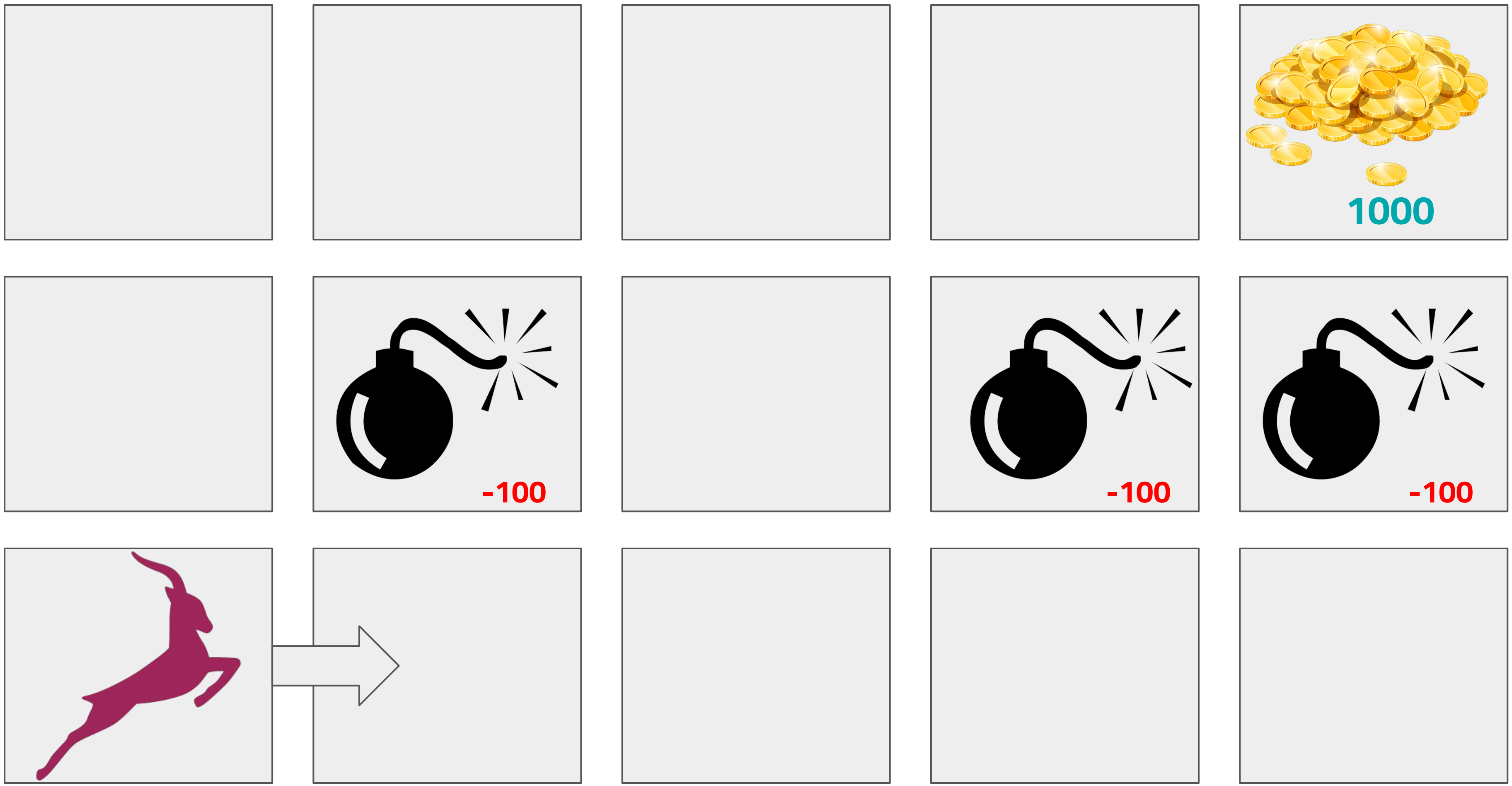
- An agent learns through iterative interactions with an environment
- "Trial and error" approach (very roughly)
- RL log entry: tuple (State, Action, Time, Reward)
- How to select actions that maximize long-term rewards?
- How to design rewards?

Exploration vs. Exploitation

- **Explore**:
  try to find better actions
- **Exploit**:
  execute action with highest expected utility, given the knowledge we have
- Explore too much
  ⇒ regret caused by lack of commitment
- Exploit too much

  ⇒ regret caused by lack of knowledge
  ⇒ get stuck in local maximum

Grid World / Markov Decision Processes



Bellman equation & Markov Decision Processes

Bellman equation:

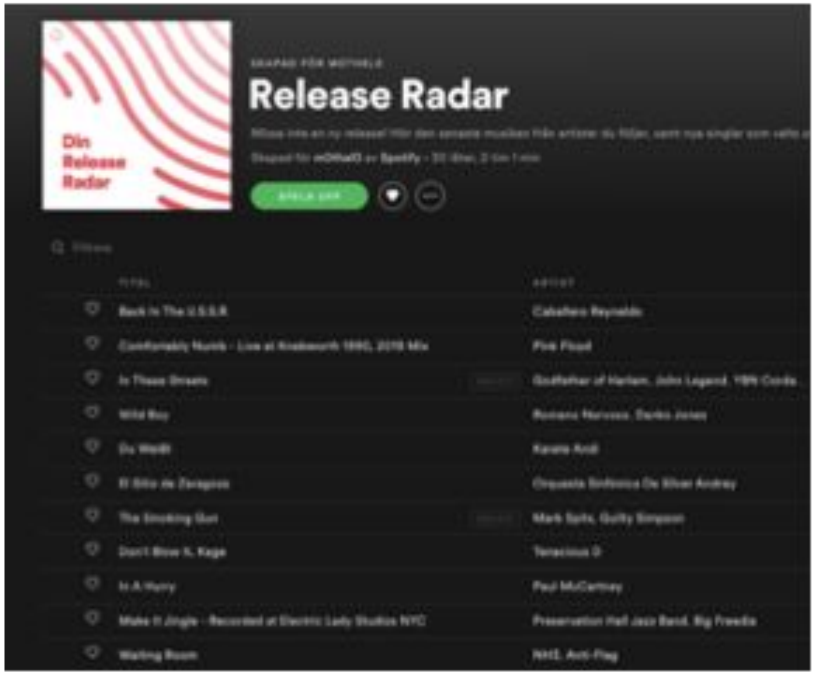
$$U(s)=\max_{a\in A(s)}(R\ (s, a) + \gamma\ U(s'))$$

- s: Current state
- *A(s):* all possible actions at state s
- s' : Future state
- R(s,a)*:* Immediate reward of S after action a
- γ*:* Discount factor

⇒ Take the action that maximizes the immediate reward plus all time-discounted future rewards

Applications - The Obvious Ones

Applications - The Useful Ones



McInerney, James, et al. "Explore, exploit, and explain: personalizing explainable recommendations with bandits." *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 2018.

Hwangbo, Jemin, et al. "Learning agile and dynamic motor skills for legged robots." *arXiv preprint arXiv:1901.08652* (2019).

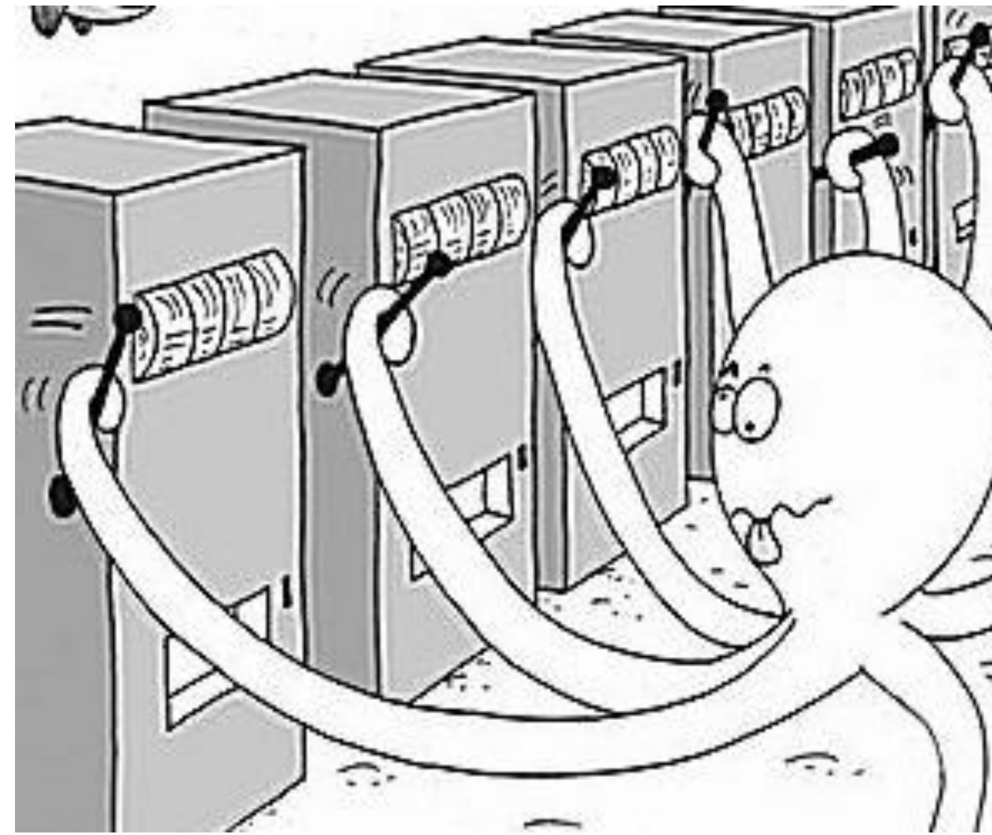
Algorithm I: (Deep) Q-Learning

- Learn action-utility (Q)
- Does not require underlying formal model (only state and actions)
- Example: The expected reward of my actions [a, b, c] in state S is [3, 4, 5].
  ⇒ If the agent **exploits**, it takes action c,
  If it explores, it takes a random action
- Problem: infinite state spaces:

  ⇒ Approximate, discretize
- Often: neural network “under the hood”

Algorithm II: Multi-Armed Bandits I

- N possible actions
- Each action has unknown expected reward (random variable)
- Goal: find best (or at least "good" action)



http://www.primarydigit.com/blog/multi-arm-bandits-explorationexploitation-trade-off

Algorithm II: Multi-Armed Bandits II

- N arms, $0<\varepsilon<1$
- At iteration i, $0 < i < N$:
  - Pull arm i.
  - Log reward returned by arm i.
- At iteration i, $i > N$:
  - If $\varepsilon>\mathrm{random}(0,1)$: Pull random arm
  - Else: Pull arm with highest expected reward
  - Update expected reward of pulled arm

SIGNAVIO

Thanks!

SIGNAVIO

EMEA

Signavio GmbH
Kurfürstenstrasse 111
10787 Berlin
Germany

+49 30 856 21 54-0
+49 30 856 21 54-19

The Americas

Signavio, Inc.
800 District Avenue
Burlington, MA 01803
United States of America

+1 978 320 5040

APAC

Signavio Pte. Ltd.
168 Robinson Road
#12-01 Capital Tower
Singapore 068912

SG: +65 6631 8334
AU: +61 3 9008 4272

info@signavio.com

www.signavio.com