Machine Learning - Other supervision strategies
(69152) DRL

Master in Robotics, Graphics and Computer Vision

Ana C. Murillo

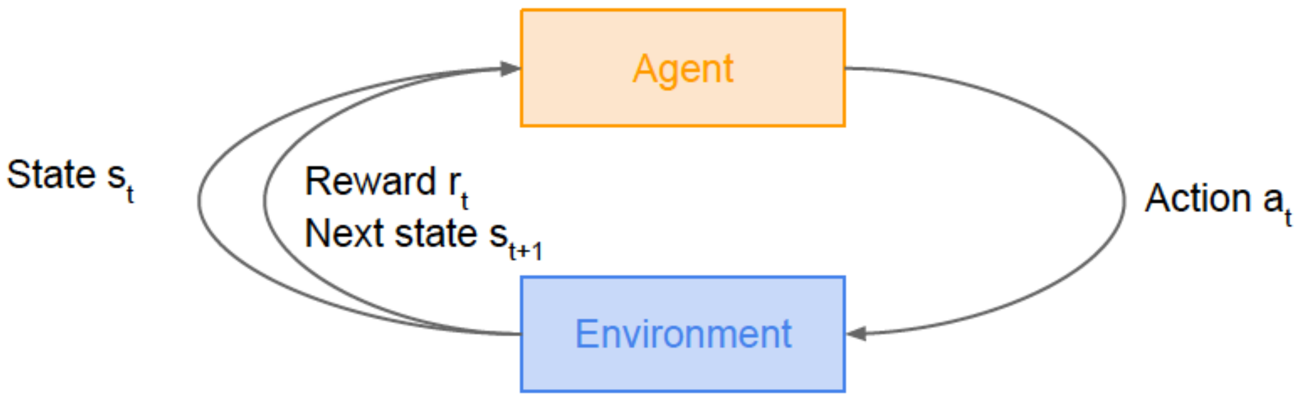
Universidad Zaragoza

Next …

- More advanced DL models
- … and different supervision strategies
  - DRL
  - Unsupervised
  - Recurrent architectures

Deep Learning & RL

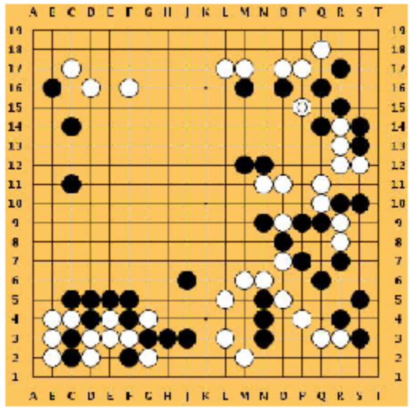
- **Reinforcement Learning ingredients:**
  - **Agent** that interacts with the environment
  - … by performing **actions**
  - Environment provides numeric **reward**
  - We want to *learn which actions to take* to maximize the reward



Adapted from Li, Johnson, Yeung. http://cs231n.stanford.edu 2017

Deep Learning & RL

- **Reinforcement Learning ingredients:**
  - Objective?
  - State?
  - Action?
  - Reward?



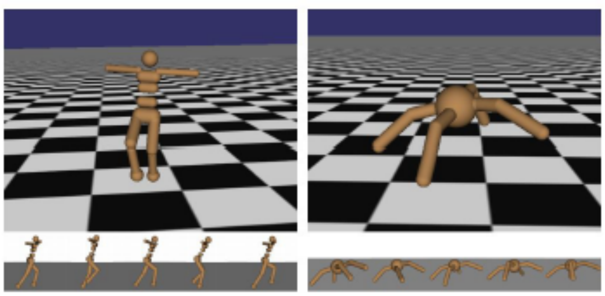
How to model a given problem?

Atari: *Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." arXiv preprint arXiv:1312.5602 (2013).*
Go: *Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 2016*

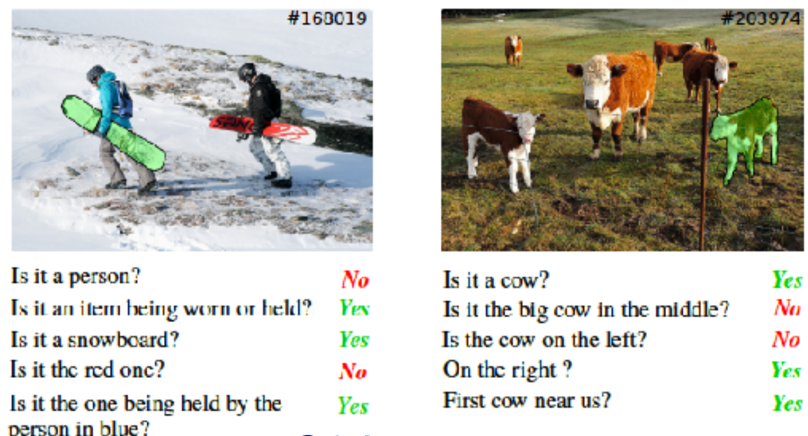
Deep Learning & RL

- **Reinforcement Learning ingredients:**
  - Objective
  - State
  - Action
  - Reward

How to model a given problem?



[Learning to run & stand up]



[Q&A games]

***Locomotion**: Duan, Yan, et al. "Benchmarking deep reinforcement learning for continuous control." ICML 2016.*
***GuessWhat**: Harm de Vries, et al. GuessWhat?! Visual object discovery through multi-modal dialogue. CVPR 2017.*

Deep Learning & RL

Once we get to “represent” the problem …

How does it work?

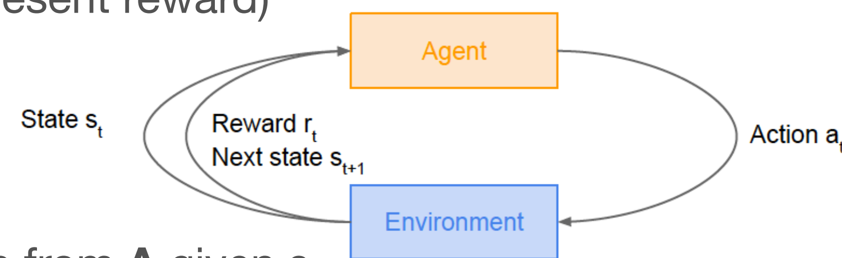
Deep Learning & RL

- **Reinforcement Learning**:
  - Involves problems about making decisions and/or predictions about the future
    - Examples: *Video games, Board games, Robotics, Recommender systems, …*
  - Our goal is to learn the best *behaviour policy*

Deep Learning & RL

- **RL** —> Markov Decision Process (Markov property*)
  - *s*: state ∈ **S**
  - *a*: action ∈ **A**
  - **R**: reward given a pair (s,a)
  - ℙ probability of transition from s, given a, to next state
  - γ: discount factor (future vs present reward)

** Current state completely characterises the state of the world*



- Policy **π**:
  - function to map from **S** to **A**
    i.e, specify what action to take from **A** given *s*
  - **optimal policy —> Find it!**
    - » maximizes cumulative *discounted* reward $\sum_{t>0} \gamma^t r_t$

Deep Learning & RL

How can we learn the best policy?

Many options …

- Learn values of each action
- Learn policy directly
- Learn a model - infer policy by planning

Deep Learning & RL

- **Goal**: Find **optimal policy** (π)
- *How do we use the policy?*
    - Following a policy produces sample paths of actions
    - Each action modifies the state of the environment

Deep Learning & RL

- **Goal**: Find **optimal policy** (π)

How good am I doing?

- **Value function ($V^\pi(s)$)** at state s is the expected cumulative reward after following the policy π from that state s

- **Q-value function ($Q^\pi(s,a)$** at state s and action a, is the expected cumulative reward performing a in s and then following the policy

$$V^\pi(s) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r_t | s_0 = s, \pi\right] \qquad Q^\pi(s,a) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r_t | s_0 = s, a_0 = a, \pi\right]$$

Deep Learning & RL

- **Goal**: Find **optimal policy** (π)

How good am I doing?

- **Value function ($V^{\pi}(s)$)** at state s is the expected cumulative reward after following the policy π from that state s

- **Q-value function ($Q^{\pi}(s,a)$** at state s and action a, is the expected cumulative reward performing a in s and then following the policy

Deep Q-learning?

Q-learning: *Use function approximator to estimate the action-value function:*

$$Q^{\pi}(s,a;\theta) \approx Q^{\pi}(s,a)$$

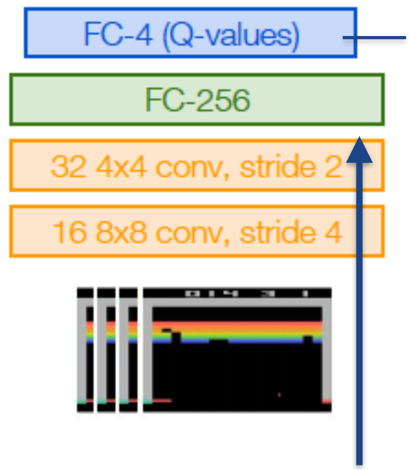
function approximator —> deep neural network —> deep q-learning

Deep Learning & RL: Deep Q-Learning example

- **Q-value function Q^{π}(s,a)** at state s and action a: expected cumulative reward performing **a** in **s** and then following the policy. **Approximation with Q^{π}*(s,a; θ)***
- **Forward pass?** Q-values for all actions from current state
- **Problems with the consecutive information feed** correlated samples - inefficient learning

Lab 5

Q^{π}(s,a; θ)
θ represents the neural net



Last FC layer has **4-d output** (if 4 actions) corresponding to $Q(s_t,a_1)$, $Q(s_t,a_2)$, $Q(s_t,a_3)$, $Q(s_t,a_4)$

Current state s_t: 84x84x4 (last 4 frames after grayscale conv., downsampling and cropping)

Atari: *Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." arXiv:1312.5602 (2013).*

Slide Adapted from Li, Johnson, Yeung. http://cs231n.stanford.edu 2017

Deep Learning & RL

- Sometimes **Q-learning** is not enough …
  - too complex/too many dimensions,
    hard to learn every pair (s,a)
  - Alternative: learn directly the policy —>
    policy gradient approaches

- Learn *Policy Gradients* **and** *Q-learning*: ***actor-critic***
  - training both the policy (*actor*)
    and the Q-function (*critic*),
    only on the pairs (s,a) generated by the *actor*

Deep Learning & RL

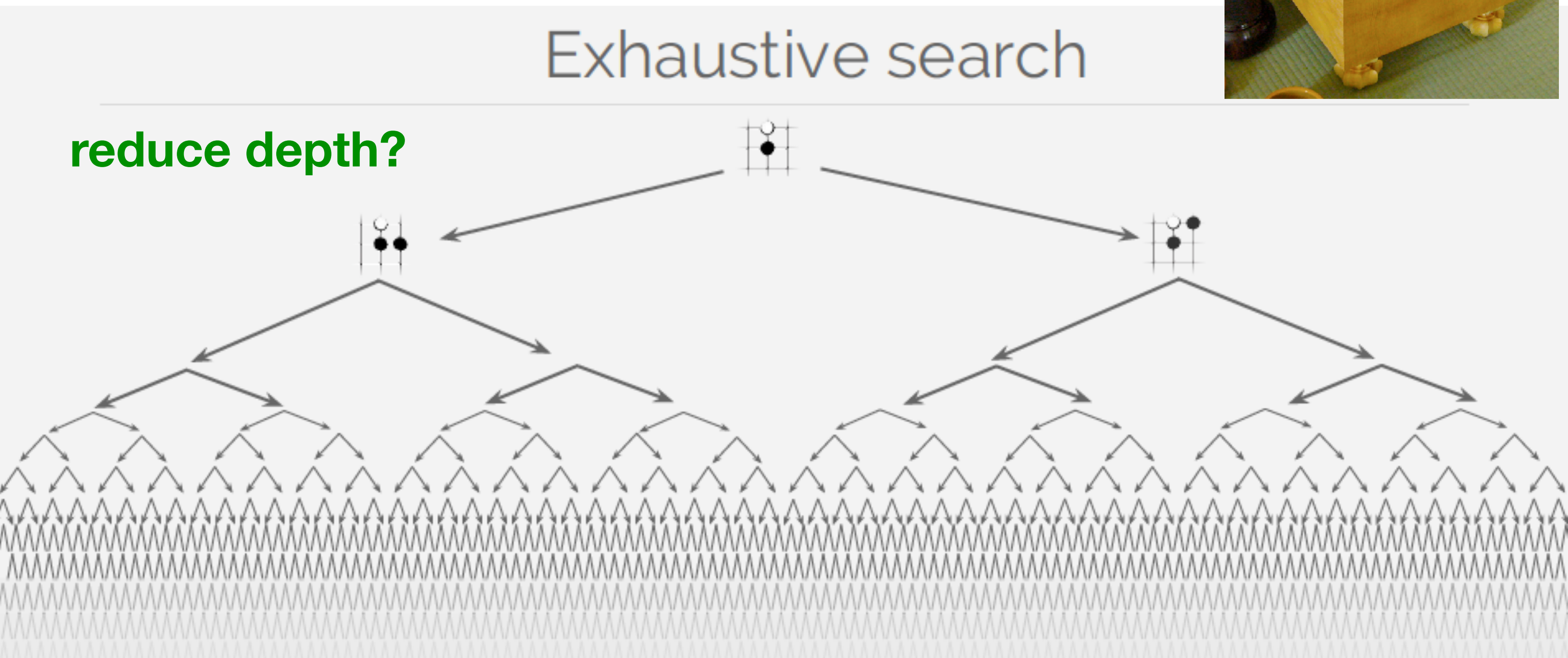
- Learn Policy Gradients and Q-learning: actor-critic
- **Training both** the policy (actor) and the Q-function (critic), only on the pairs (s,a) generated by the actor

A well known Deep RL example

Go: Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 2016

Deep Learning & RL

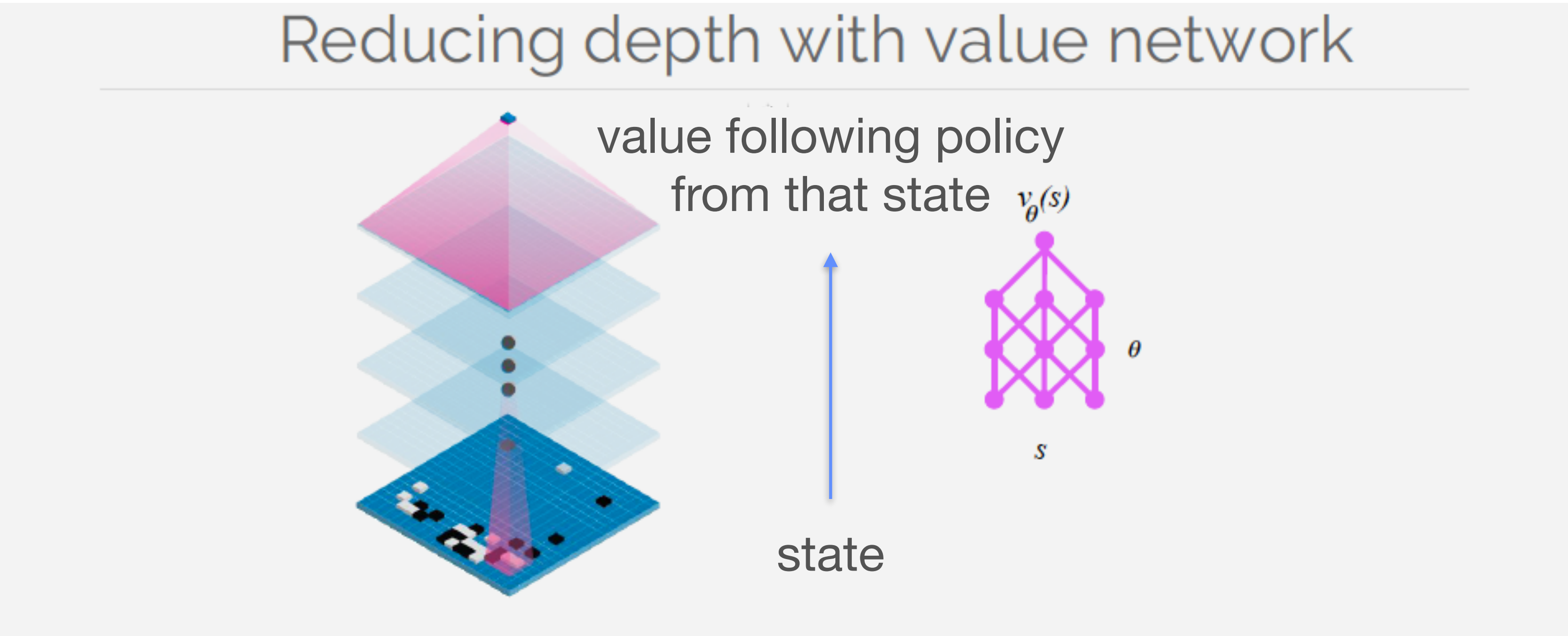
- supervised + reinforcement
- tree **search** + **deep RL**



slide adapted from RLSS_2017 Deep reinforcement learning — Hado van Hasselt

Deep Learning & RL

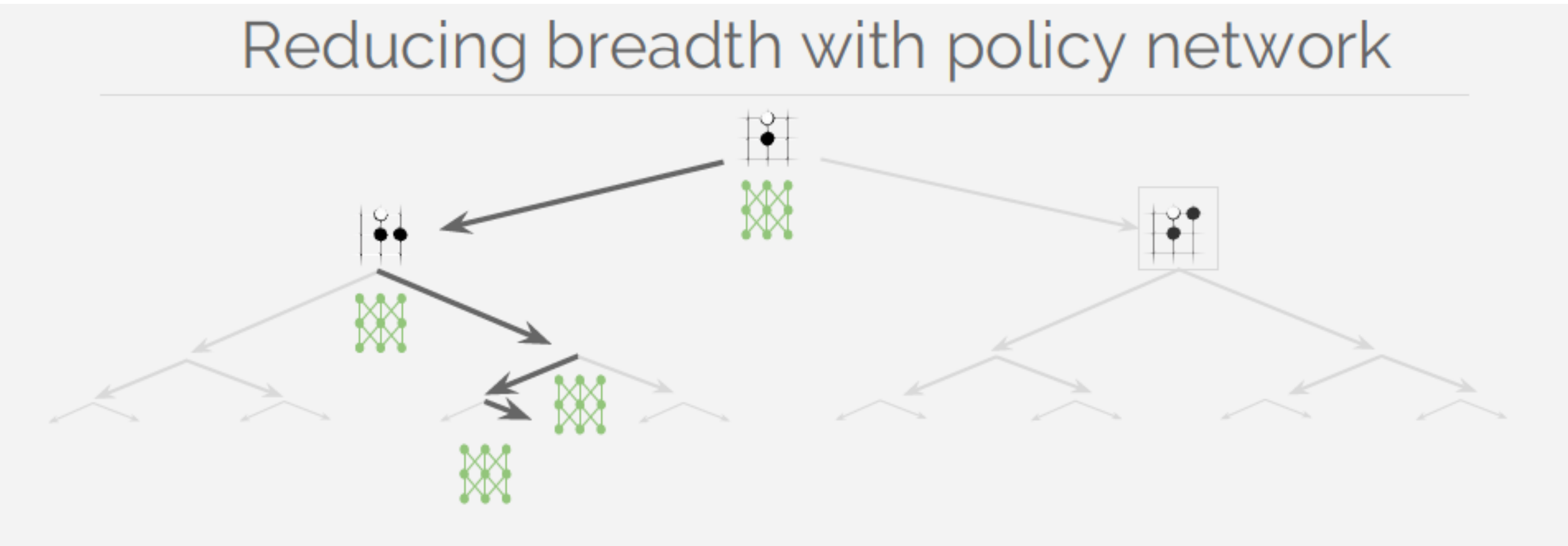
- supervised + reinforcement
- tree **search** + **deep RL**
  - both the policy (*actor*) and the **Q-function** (*critic*)



slide adapted from RLSS_2017 Deep reinforcement learning — Hado van Hasselt

Deep Learning & RL

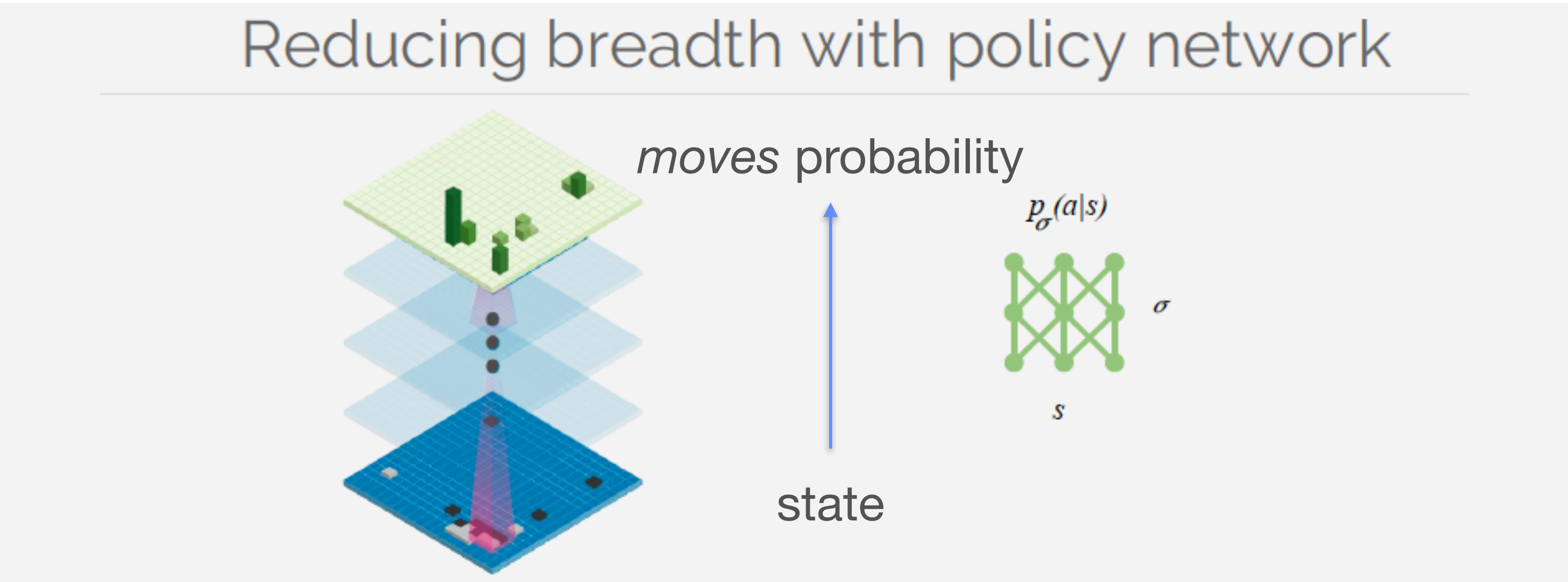
- supervised + reinforcement
- tree **search** + **deep RL**
  - both the **policy** (*actor*) and the Q-function (*critic*)



slide adapted from RLSS_2017 Deep reinforcement learning — Hado van Hasselt

Deep Learning & RL

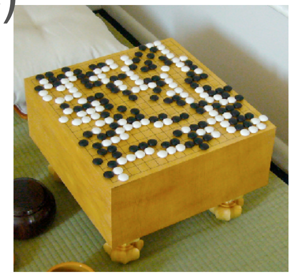
- supervised + reinforcement
- tree **search** + **deep RL**
  - both the **policy** (*actor*) and the Q-function (*critic*)

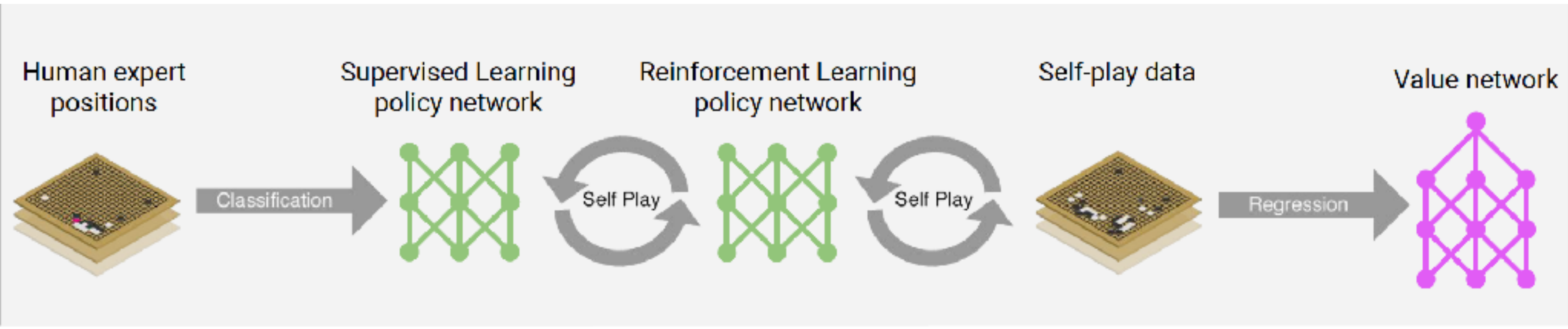


slide adapted from RLSS_2017 Deep reinforcement learning — Hado van Hasselt

Deep Learning & RL

- supervised + reinforcement
- tree search + deep RL
  - both the policy (*actor*) and the Q-function (*critic*)

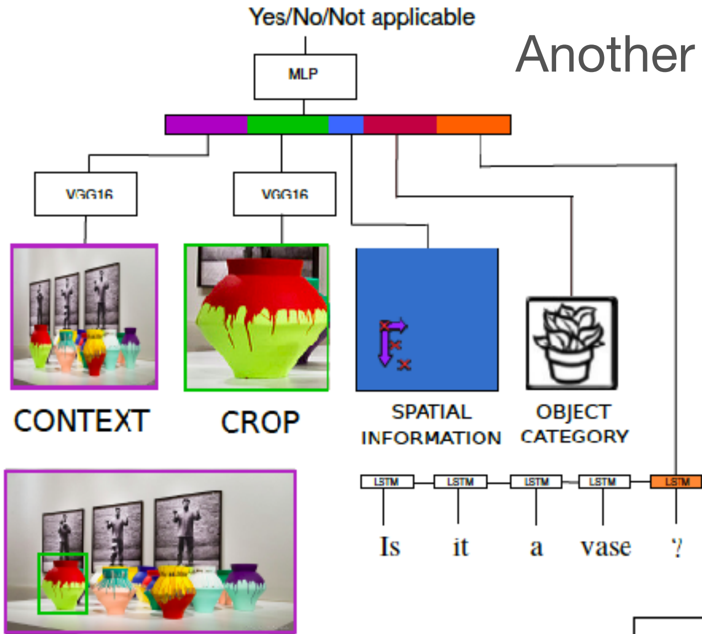




slide adapted from RLSS_2017 Deep reinforcement learning — Hado van Hasselt

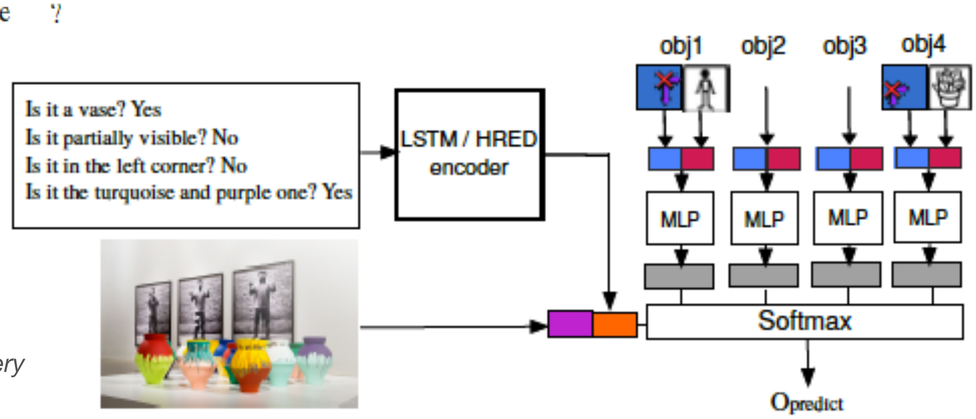
Deep Learning & RL

Another **Deep RL example:** learning 2 players



Oracle & ***Guesser*** networks

guesser:
- model of image with segmented objects (all MLPs share weights)
- generates questions



oracle:
- model of image, question, crop, spatial info and category
- *knows the answers*

Harm de Vries, et al. GuessWhat?! Visual object discovery through multi-modal dialogue. CVPR 2017.

Deep Learning & RL

- Still a lot of on-going research
  - Policy gradients: general - high variance - requires a lot of samples.
  - Q-learning: does not always work (if it does, pretty efficient)

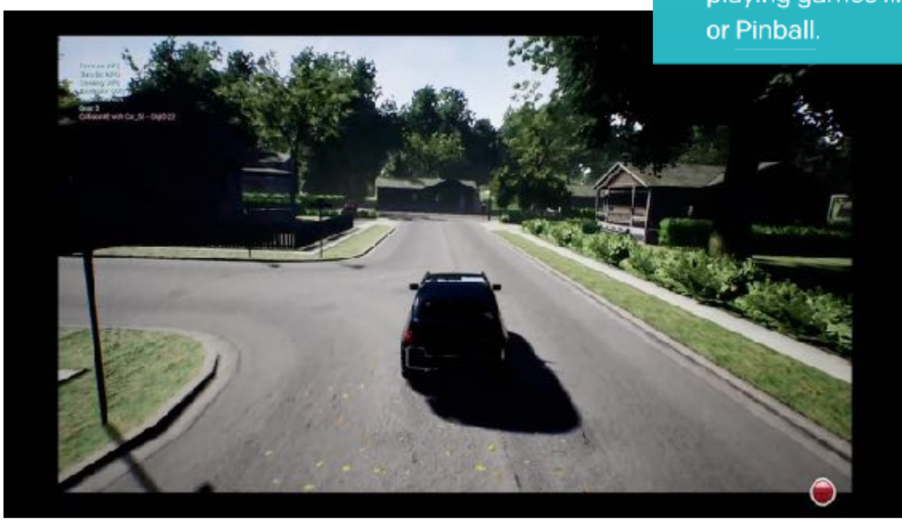
Some of the challenges?

Deep Learning & RL

- Still a lot of on-going research
  - Policy gradients: general - high variance - requires a lot of samples. *How do we sample?*
  - Q-learning: does not always work (if it does, pretty efficient). *How do we explore all the pairs s,a?*

Deep Learning && RL: simulation frameworks

- An essential piece …



[OpenAI GYM]

[Microsoft AirSIM]

[SONY_GT]

Next …

- Other supervision strategies:
    - Unsupervised
    - GANs

?

Bibliography - Resources for some of the materials today

- Stanford classes on deep learning for Computer Vision (http://cs231n.stanford.edu) and Deep Learning (https://cs230.stanford.edu/)

- Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016. http://www.deeplearningbook.org

- Deep Learning Summer School Montreal: https://mila.quebec/en/cours/deep-learning-summer-school-2017/

  - CS 294: Deep Reinforcement Learning and CS294-129 Designing, Visualizing and Understanding Deep Neural Networks. UC Berkeley.

  - DRL Bootcamp: https://sites.google.com/view/deep-rl-bootcamp/lectures

  - Open AI Gym