Robotics: Science and Systems

Machine Learning for Robot Control

Zhibin (Alex) Li
School of Informatics
University of Edinburgh

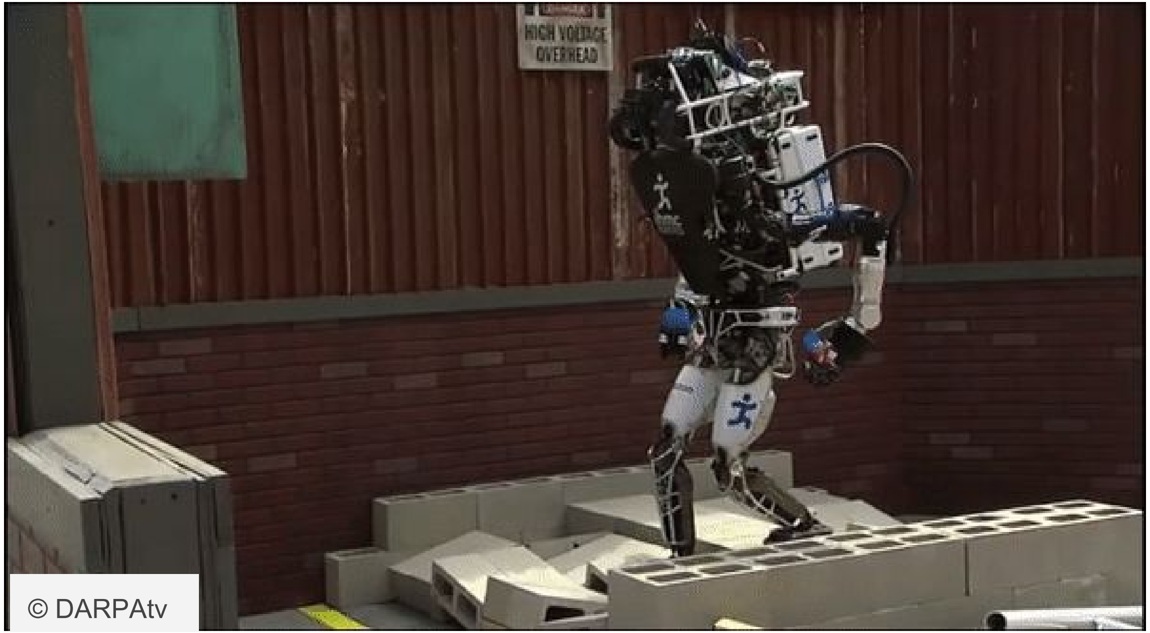
Content

- Why learning?
- Common learning paradigms
- Deep learning for robot control
- A brief summary

* only cover the machine learning cases for robot control only

Why learning?

What are missing in DRC?



© DARPAtv

1. Operators manually put footsteps for robots, which introduces human errors
2. No use of hands during locomotion
3. No reaction while falling, no autonomous behaviors
4. ⋮

   ⋮

Lesson learned from DRC: Robots need to be more autonomous

High level human supervision ⟷ (Limited communication) ⟷ Low level control for task execution

How to fill the gap?

High level human supervision ⟷ (Limited communication) ⟷ Machine learning ⟷ Low level control for task execution

Why learning?

Suitable to handle cases that are:

1. stochastic, uncertainty
2. a system or process that is hard to be modeled, strongly nonlinear or state-dependant (time-varying)
3. multiple scenarios (difficult to be numerated), multi-modality
4. high dimensional action space (multiple actions can lead towards the same goal, need to predict future)

Why learning?

Advantage:

1. automate the process of designing policies, free people from tedious routines of coding;
2. explore some new solutions, uncover our blind spots, create more inventions;
3. potentially to facilitate progress of science and technology, if used appropriately.

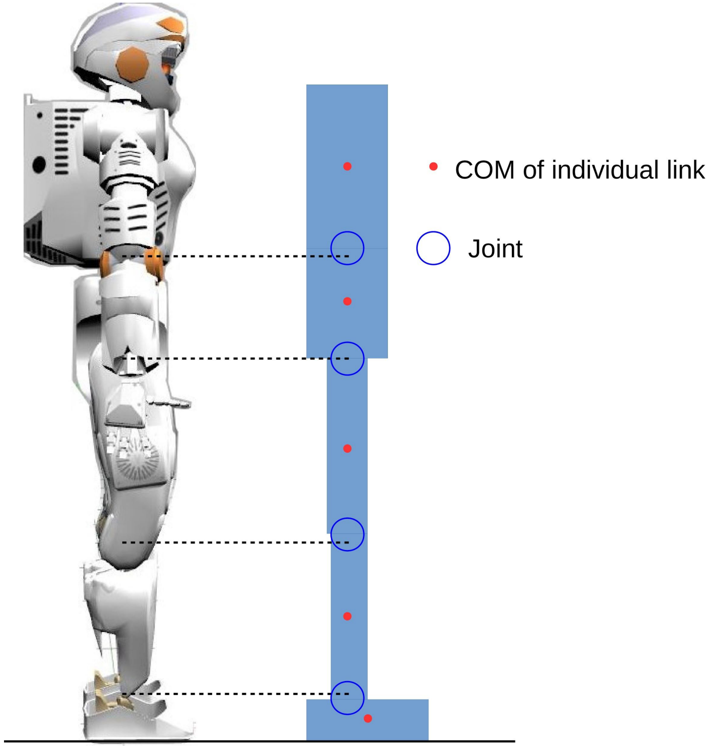
Common learning paradigms

- Supervised learning: know how to classify data (labeled), train neural network to learn how to classify it.
- Unsupervised learning: don't know how to classify data (unlabeled), train neural network to find out the underlying principles how to classify it.
- Reinforcement learning: don't know how to classify data, but a reward will be given to if end result is good, or a penalty if the result is bad (eg, Atari video game).

Deep learning for robot control

Go beyond video games, a case study of using deep learning for robot balancing and locomotion control.

Here the example uses Deep Deterministic Policy Gradient (DDPG), a model free, actor-critic reinforcement learning algorithm.

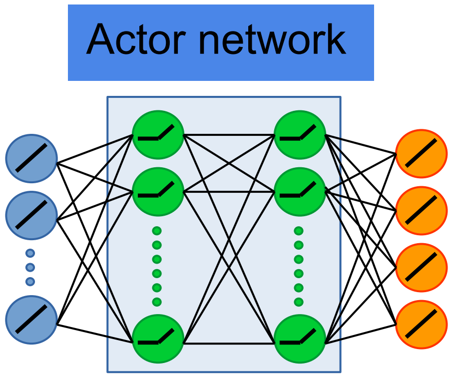


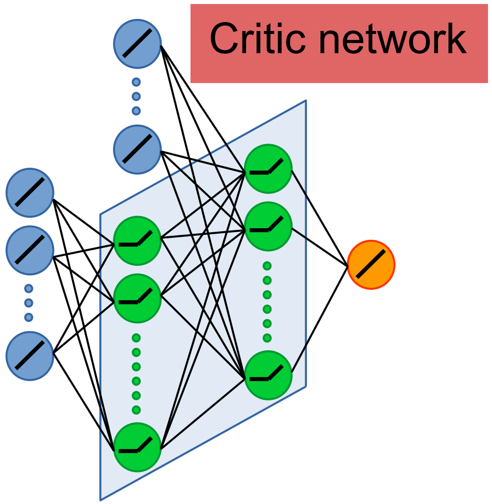
Chuanyu Yang, Taku Komura, Zhibin Li, ``Emergence of Human-Comparable Balancing Behaviors by Deep Reinforcement Learning ,'' in IEEE-RAS International Conference on Humanoid Robots, 2017.

Deep learning

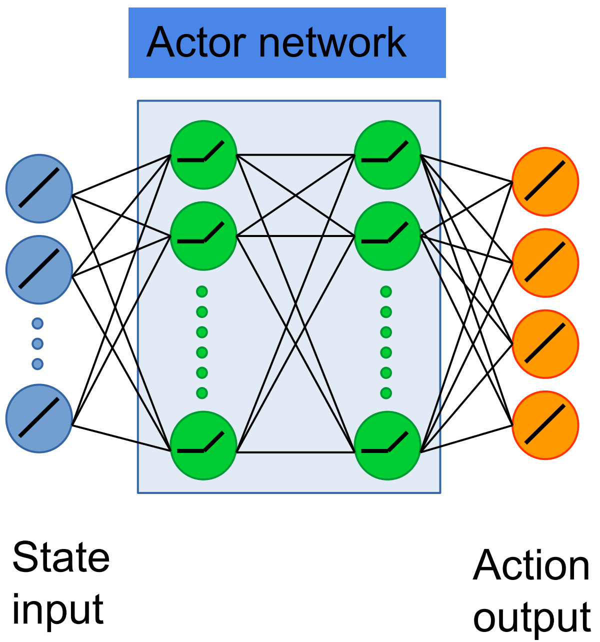
It consists two networks:

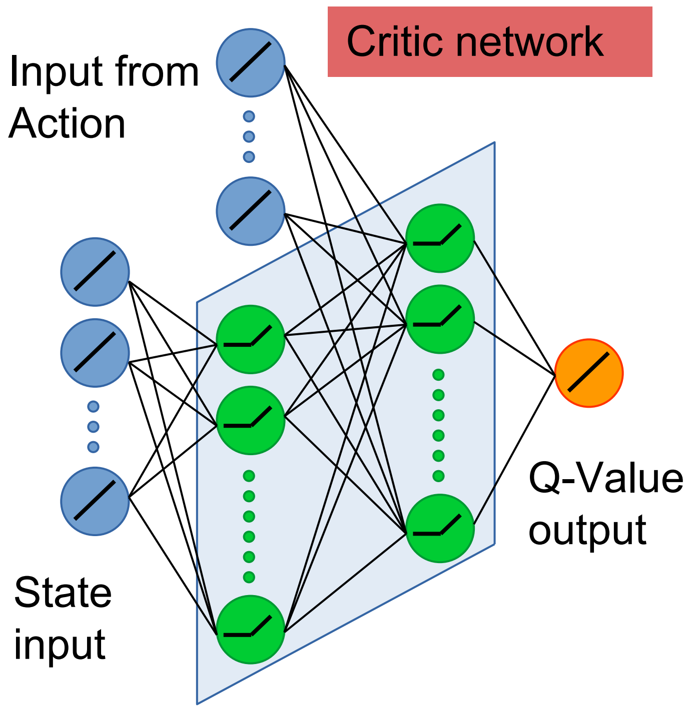
1. Actor network learns the policy that generates the optimal action;
2. Critic network evaluates the performance of the policy learnt by the actor network.



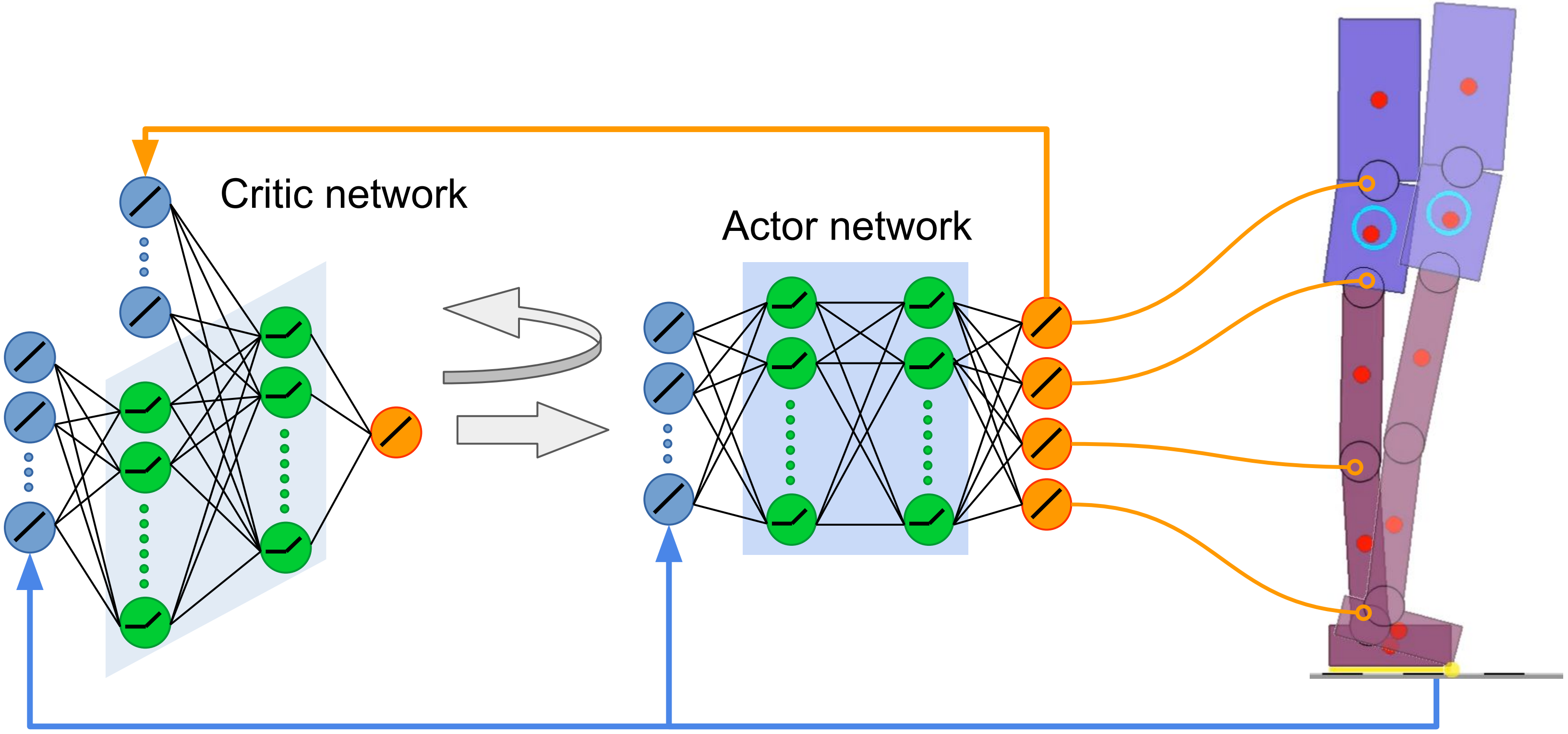


Deep Deterministic Policy Gradient (DDPG)

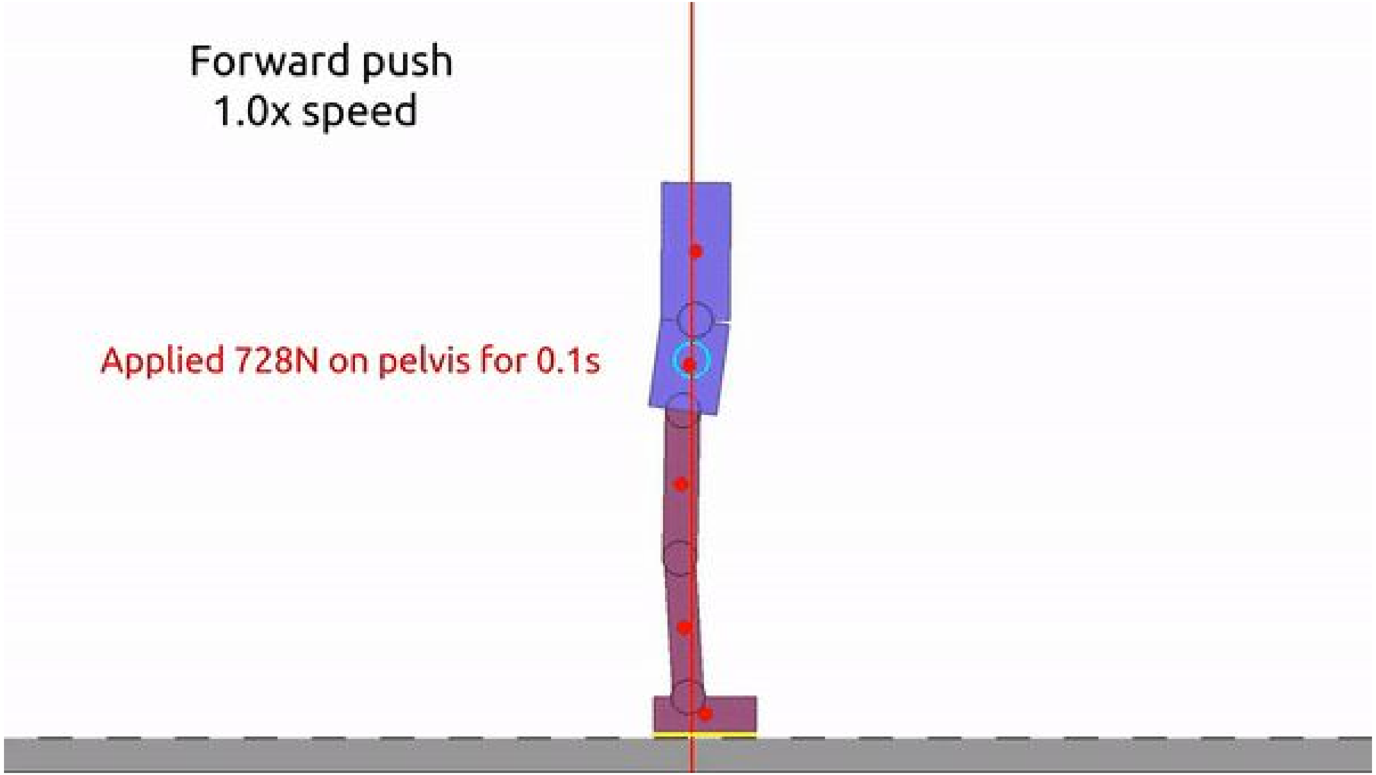




Learn an optimal policy



Deep learning (DL) for push recovery

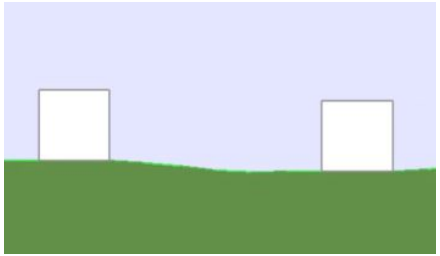


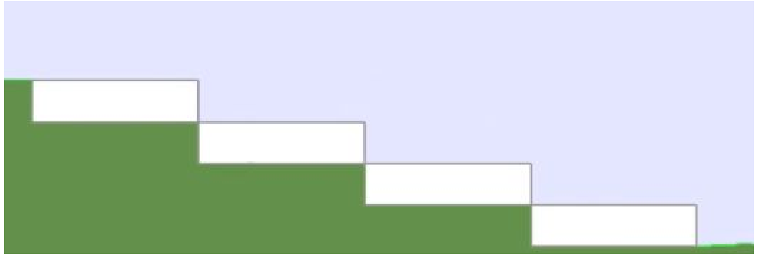
DL for rough terrain locomotion

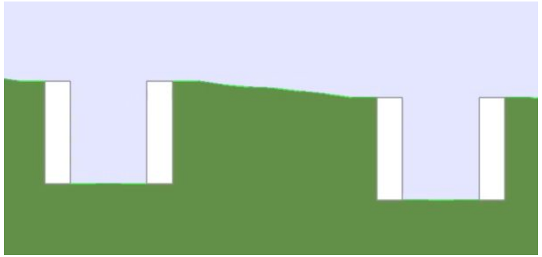
Next step:

- from balancing to walking, stepping, running, jumping, and leaping;
- using perception for coordinating control strategies.

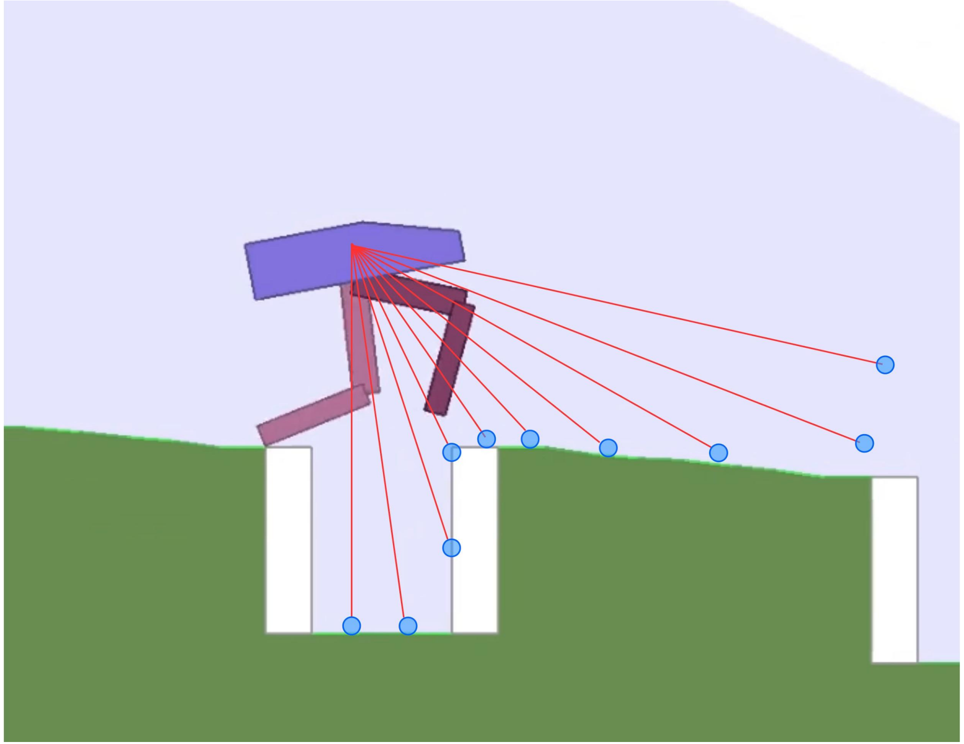
Adding recurrent neural network, mix DDPG with Recurrent Deterministic Policy Gradients (RDPG).







DL for rough terrain locomotion

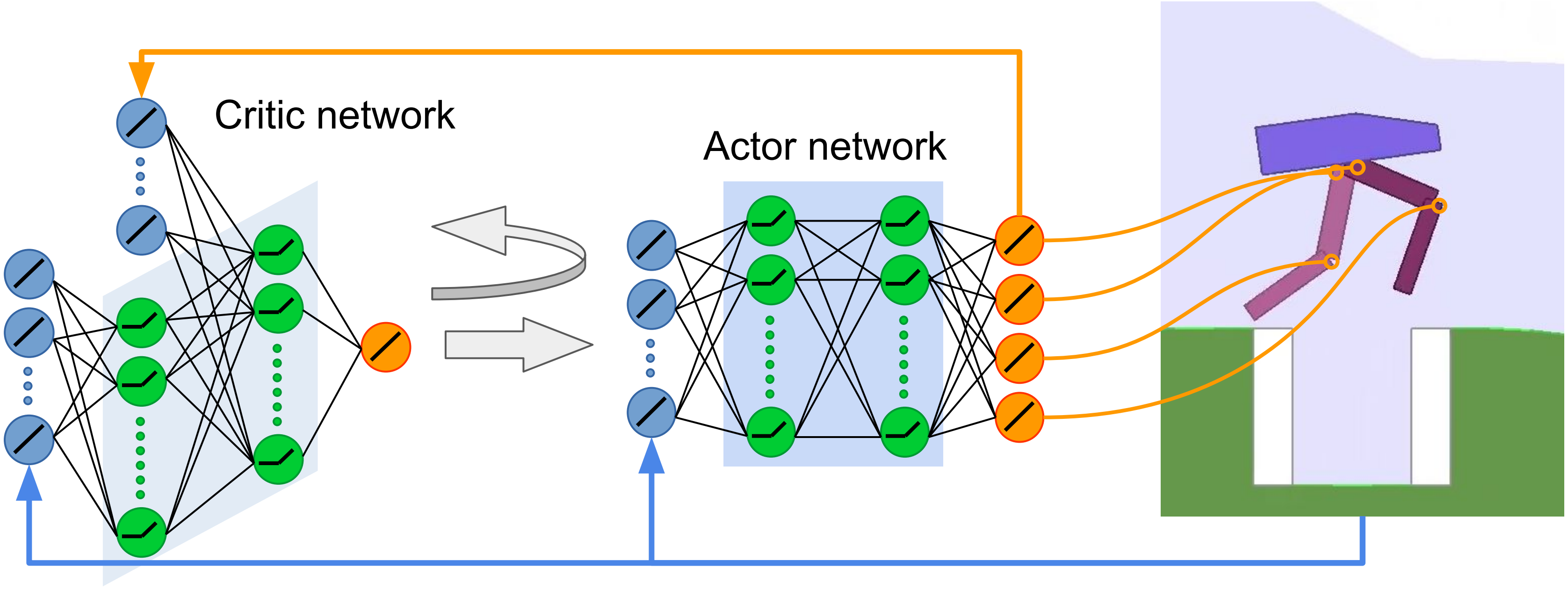


Feedback: body orientation, horizontal & vertical speed, joint position & velocity, foot-ground contact.

Perception: partially observable height map by 10 lidar scans.

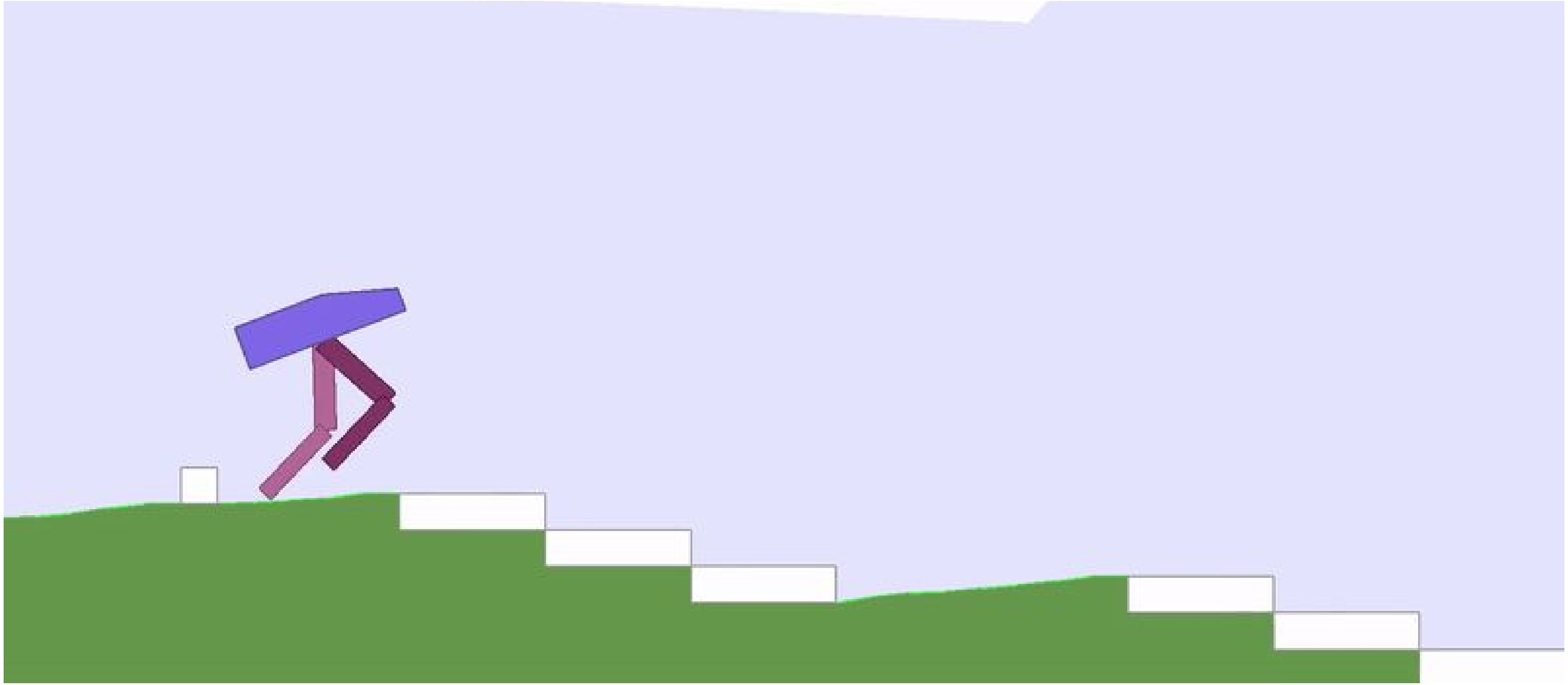
Doo Re Song, Chuanyu Yang, Christopher McGreavy, Zhibin Li, ``Recurrent Network-based Deterministic Policy Gradient for Solving Bipedal Walking Challenge on Rugged Terrains,' submitted to 'IEEE International Conference on Robotics and Automation, 2018.

Learning optimal policies

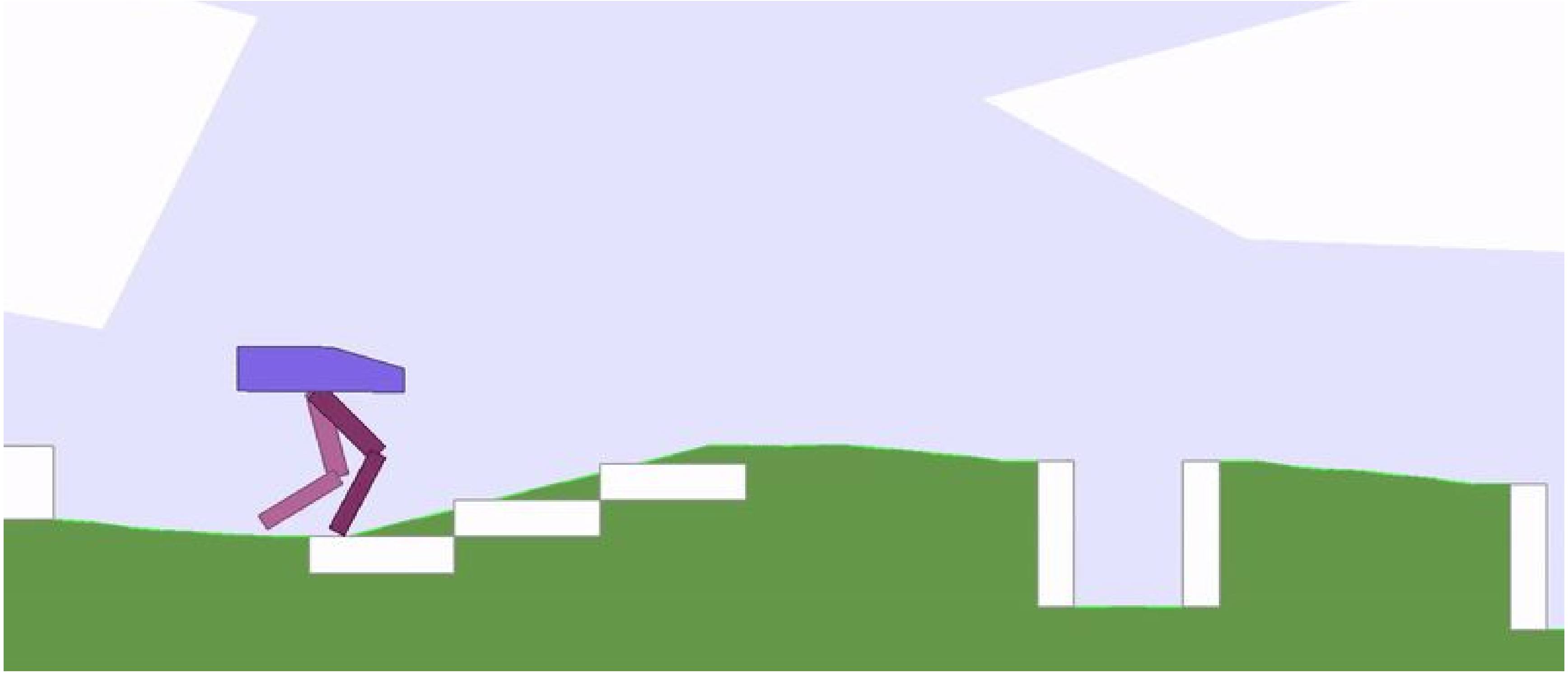


DL locomotion: obstacles

DL locomotion: stairs



DL locomotion: pitfalls



Is learning everything?
A brief summary

A catering example

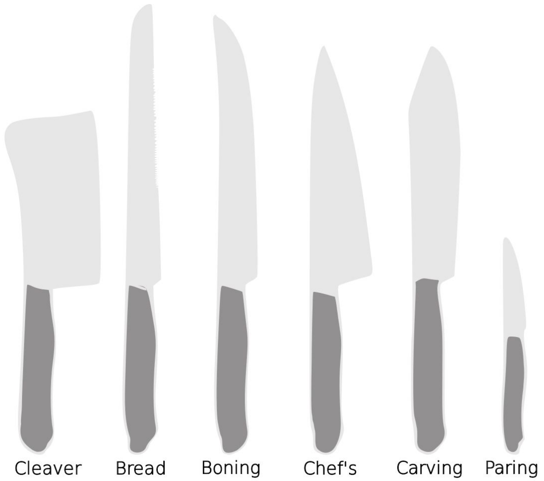


Image by Themightyquill

A right attitude towards machine learning

What a future super intelligence would think about nowadays' learning and math- or model-based tools?

- Mathematical tools, physics laws and so on are created by the top intellectuals in human history, it is unwise, from a super intelligence point of view, to use a week AI (what we have now) to learn from scratch in a very primitive way.

A right attitude towards machine learning

- Learning from scratch can be okay for non-physical learning process, eg computer graphics or playing GO inside computer, 30 iteration and 30,000 or 30 million iteration does not make a big difference, if time is not a problem.
- However, for real physical systems, eg robots, it can be hard because robots do break before learning is done. We, resilient biological systems, also can be injured before reaching successful performance.

Right tool for the right thing! It is complementary to other control tools we have learned.