

“A beginning is the time for taking the most delicate care that the balances are correct.”

Frank Herbert, *Dune*



CMPUT 628

Deep RL

Plan

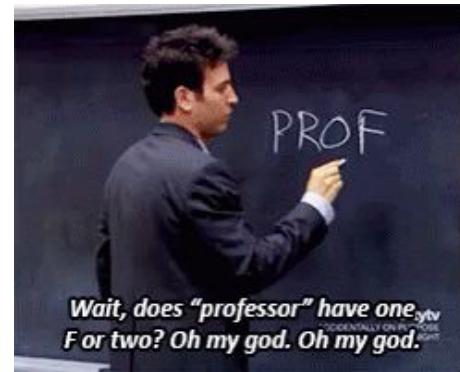
- Introduction
- Course logistics
- Intro to Deep RL

Please, interrupt me at any time!

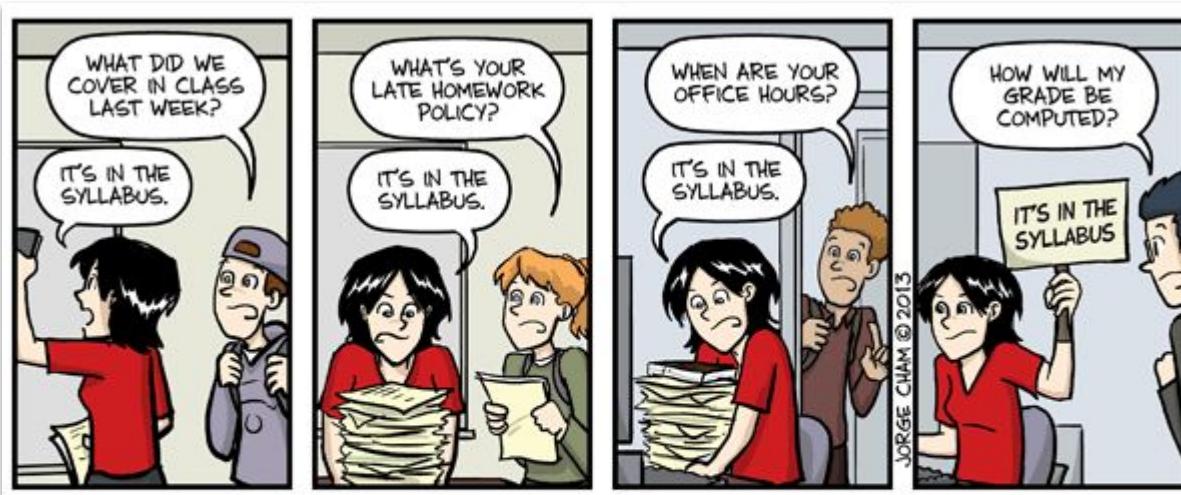


About myself

- Name: Marlos C. Machado
- I was born in Brazil
- I have been living in Edmonton for 10+ years
- I have 2 kids
- Ph.D. working on reinforcement learning
 - Interned at Microsoft Research, IBM Research, and DeepMind
- Worked 4 years at Google Brain and DeepMind
 - Among several other things, we deployed RL to fly balloons in the stratosphere
- Have been doing deep RL pretty much since it “started”



Course overview and logistics



IT'S IN THE SYLLABUS

- Canvas: [link](#)
- Slack: [link](#)
- My website: [link](#)
- Google drive: [link](#)

University of Alberta

CMPUT 628: Deep Reinforcement Learning
LEC B1
Winter 2026

Instructor: Merlin C. Machado
Teaching Assistant: Hon Tik (Rick) Tsai

Office: UOONM 714H
E-mail: machado@ualberta.ca
Web Page: <https://canvas.ualberta.ca/courses/30281>

Office hours: Slack and eClass, asynchronously. In-person meetings might be available upon request.

Lecture room & time: GSB 7-11, Tuesday and Thursday 11:00 - 12:20

Attendance isn't mandatory, although it is strongly encouraged.

Slack invitation link: We will use Slack as an optional alternative to eClass for communication and question-answering. The invitation link will be provided to the students on eClass.

COURSE CONTENT

Course Description: This course provides an introduction to the fundamentals of deep reinforcement learning, a subfield of machine learning that focuses on learning the types of problems and solution methods. It is focused on how solution methods in deep reinforcement learning involve using neural networks to parameterize artifacts such as value functions, policies, or models. In this course, we will take the view that deep reinforcement learning studies the design of algorithms that allow for modern deep learning techniques. This is in contrast to being focused on the function approximation, function learning, and simply seeing reinforcement learning as a consumer of supervised learning techniques.

The course provides an introduction to the general concepts underlying most of the traditional deep reinforcement learning algorithms that have been used to learn to achieve goals in problems with high-dimensional observations and actions. The course will cover the design choices that underlie model-based reinforcement learning and the general idea of learning a value function that maps from particular choices. Time permitting, it will then discuss some high-profile deep model-based reinforcement learning algorithms and key policy gradient methods, with the major bulk of the

Communication and classes

- Official: Canvas [\[link\]](#)
 - Announcements, slides, lecture notes, assignments

I do my best to have key announcements in my slides too

- Email address*: machado@ualberta.ca
- Unofficial: Slack [\[invitation link\]](#)
- My website [\[link\]](#)

The image contains three screenshots of course-related web pages:

- Top Screenshot:** CMPUT 628 Winter 2026 - Deep Reinforcement Learning course page on Canvas. It shows the course title, professor (Marlos C. Machado), and a 'Deep Reinforcement Learning Foundations' section with a colorful map.
- Middle Screenshot:** The course syllabus page for CMPUT 628. It includes sections for Welcome, Course Information, and Lecture notes & slides.
- Bottom Screenshot:** Marlos C. Machado's teaching section on his website. It lists the course as 'Teaching' and 'Current' under 'CMPUT 628: Deep Reinforcement Learning - W26'. It provides lecture times (Mon/Wed 11:00 - 12:20, GSB 7-11), a link to the course page on eClass, and course syllabus information.

Attendance

It is **not** mandatory.

I find it a little arrogant to imagine that you cannot succeed without me.

That being said, we don't even have a textbook. Sure, #\$\$@&%*! happens, but in general, using office hours and sending me messages asking about things I only said in class is a waste of everyone's time, just come to class.

I won't record or livestream my classes. I feel much more free to be honest this way.

Office hours

- Mine: *After class* (tentative)
- Rick Tse: TBD
- Canvas, Slack, and email: Asynchronous

Please refrain from sending me emails and private messages. I'll be slower to answer and your question will likely benefit someone else.



Pre-requisites

- No formal requirements.
- I expect you have been exposed to the basic ideas of:
 - Reinforcement learning (CMPUT 365, CMPUT 655)
 - Machine learning (CMPUT 466/566)
- Python
 - We'll use PyTorch (and I won't teach you PyTorch)

You should either be familiar with these topics or be ready to pick them up quickly as needed by consulting outside resources.



Learning resources

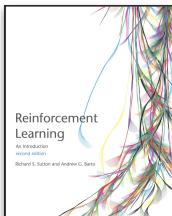
There is no required textbook.

I'll provide very incomplete lecture notes.

I recommend you read the original papers of the algorithms we discuss.

I'll try to tell you at the end of class what we'll be discussing next, in case you want to prepare.

Supplementary Textbooks:

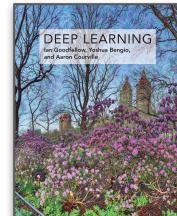


Reinforcement Learning: An Introduction

R. S. Sutton & A. G. Barto

MIT Press. 2nd Edition.

<http://www.incompleteideas.net/book/the-book-2nd.html>



Deep Learning

Ian Goodfellow, Y. Bengio, & A. Courville

MIT Press.

<https://www.deeplearningbook.org/>

Tentative

GRADE EVALUATION

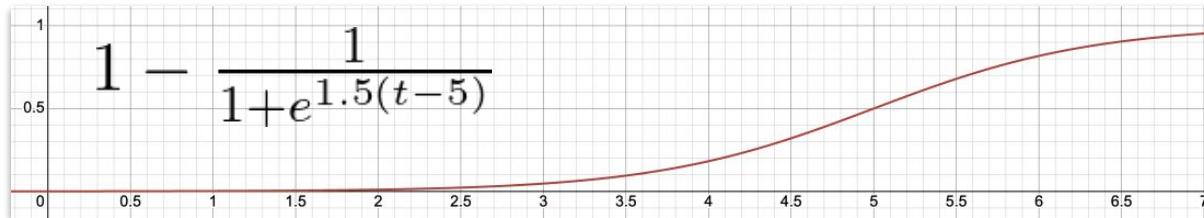
Assessment	Weight	Date
Assignment 1	10%	January 24, 2026
Assignment 2	10%	February 7, 2026
Assignment 3	15%	February 28, 2026
Assignment 4	15%	March 14, 2026
Midterm exam	20 %	March 19, 2026
Paper review	15%	April 9, 2026
Seminar	15%	March 24, 2026 – Apr 9, 2026

Closed book

Next slides

Police for late work

The late submission penalty is based on the following logistic function, which determines the percentage deducted from your final grade:



You have pretty much a two-day grace period. I won't be giving extensions unless it is something really really serious.



Paper review and seminar

- The last 6 classes will be two 30-minute seminars presented by groups of two. That's why there are only 24 of you.
- I'm still finalizing the details, but you'll be presenting and writing about a *generally relevant* paper written in the last 2–3 years.
- The order in which people will present their papers will be *randomly* decided after you give me your groups.
- The review of the same paper will be done at the end of the course.

Numerical grades to letter grades

I'll actually decide this at the end of the term based on overall class performance. But here's a guarantee for the anxious ones:

Points	97	90	85	80	75	70	60
Grade	A+	A	A-	B+	B	B-	C+

I'm sharing this with you but notice that I absolutely *won't* round grades and *no extra marks will be given*. If you end up with 89.7 and the threshold is 90, you'll get an A-.

Academic integrity

You are graduate students, don't cheat, there are consequences.

You are allowed to discuss the assignments with your classmates. Note, however, that you are not allowed to exchange any written text or code or to give and/or receive detailed step-by-step instructions on how to solve the proposed problems.

LLMs are fine to some extent. They can assist or hinder learning. You are grown-ups, don't take shortcuts. Writing is a key ability you should develop as grad student.

“Short cuts make long delays”, J. R. R. Tolkien The Fellowship of the Ring



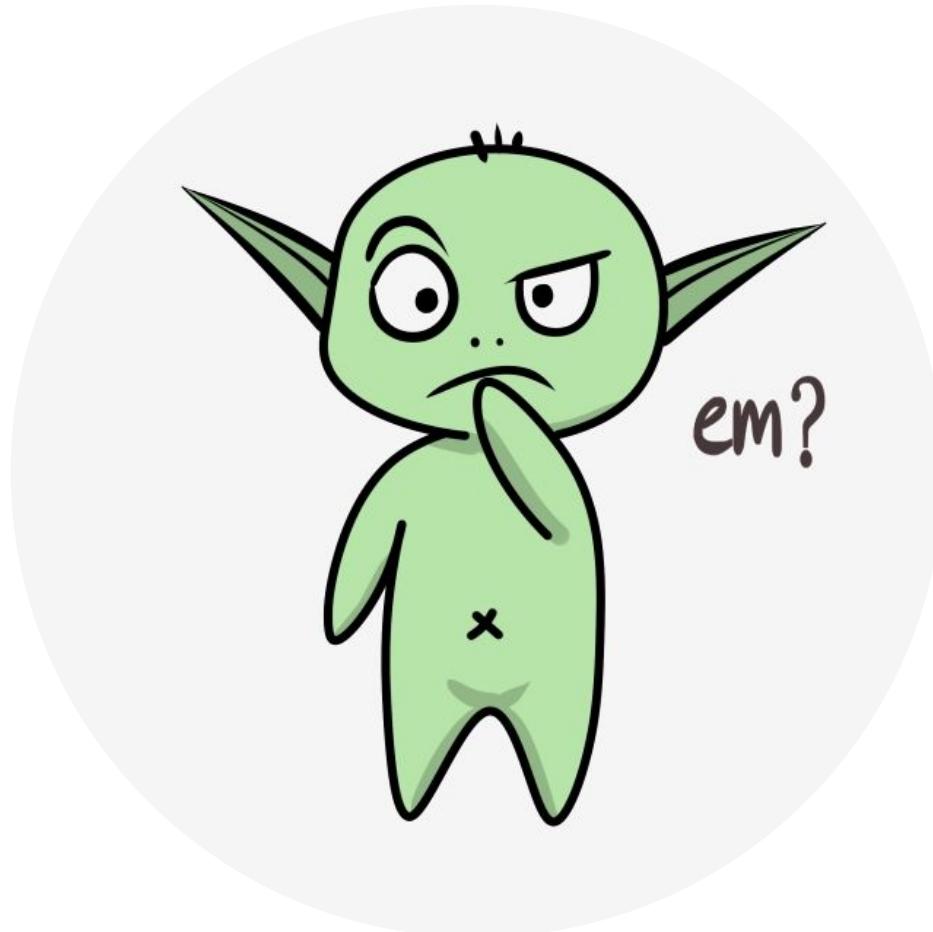
Warning: You can still leave

- I know, I know, *Deep Reinforcement Learning* sounds fun, modern, and hyp-ey

But...

- But this course won't be so well-structured as you (or I) would hope
- I won't teach you how to code fancy deep RL algorithms
- I'm not as much fun as you might think
- I don't care about grades – I might have a reputation :-)
 - *There won't be a practice midterm*
- I don't care if this course ends up being difficult





What is this course about?

Reinforcement learning

Learning to achieve goals in the world

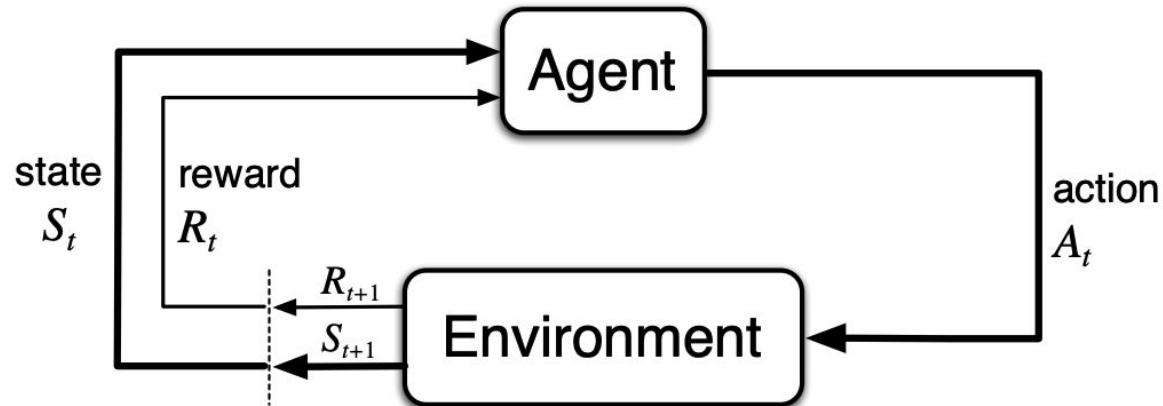


Figure 3.1: The agent–environment interaction in a Markov decision process.

It works!



Video compression
[Mandhane et al., 2022]

Matrix multiplication
[Fawzi et al., 2022]

Hardware design
[Mirhoseini et al., 2021]



Cooling systems
[Luo et al., 2022]

Thermal power generators
[Zhan et al., 2022]

Managing inventories
[Madekaet al., 2022]

...

There are many approaches to tackle this problem

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a) \left[r + \gamma v_\pi(s') \right]$$

$$q_*(s,a) = \sum_{s',r} p(s',r|s,a) \left[r + \gamma \max_{a'} q_*(s',a') \right]$$

$$V(S_t) \leftarrow V(S_t) + \alpha \left[G_t - V(S_t) \right] \quad V(S_t) \leftarrow V(S_t) + \alpha \left[R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right]$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \right]$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]$$

$$Q_1(S_t, A_t) \leftarrow Q_1(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q_2(S_{t+1}, \arg \max_a Q_1(S_{t+1}, a)) - Q_1(S_t, A_t) \right]$$

... and much much more

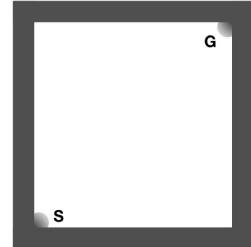
We need to *approximate* our estimates

$$v(s; \mathbf{w}) \approx v_\pi(s) \quad q(s, a; \mathbf{w}) \approx q_\pi(s, a)$$

For a long long time, the standard solution was linear function approximation:

$$v_\pi(s) \approx \hat{v}(s, \mathbf{w}) \doteq \mathbf{w}^\top \mathbf{x}(s) \doteq \sum_{i=1}^d w_i \cdot x_i(s)$$

Linear function approximation



State space: $\langle x, y \rangle$ coordinates (continuous, no grid) and $\langle \dot{x}, \dot{y} \rangle$ velocity (continuous).

Start state: Somewhere in the bottom left corner, where a suitable $\langle x, y \rangle$ coordinate is selected randomly.

Action space: Adding or subtracting a small force to \dot{x} velocity or \dot{y} velocity, or leaving them unchanged.

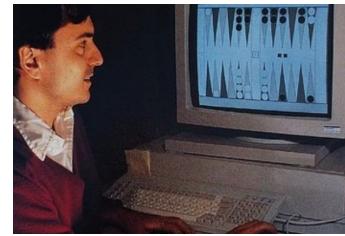
Reward function: +1 when you hit the region in G.

Features?

$$\mathbf{s} = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} \quad \mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} \quad \hat{v}(\mathbf{s}, \mathbf{w}) = \mathbf{s}^\top \mathbf{w}$$



It is not that people didn't use neural networks

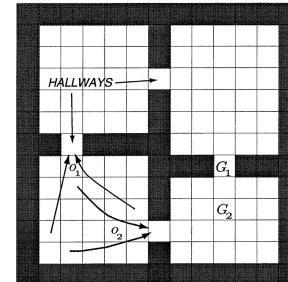
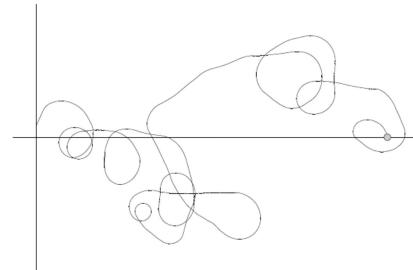
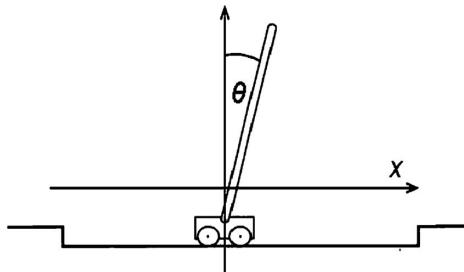


<https://achievements.ai/timeline/td-gammon-program-gerald-tesauro/>

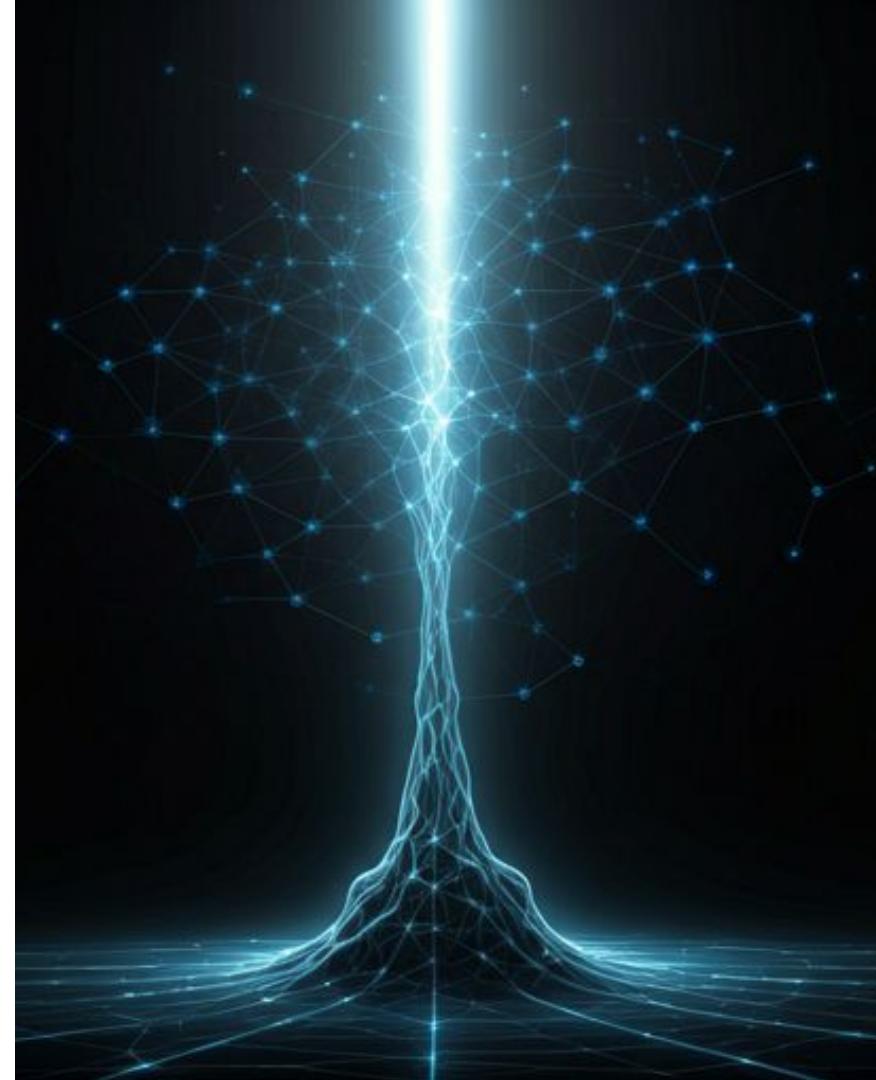
- Anderson (1986) applied RL + NNs to the Tower of Hanoi puzzle
Anderson (1987) already praised a NN's ability to learn representations: "the ability of the two-layer network to discover new features and thus enhance the original representation is critical to solving the balancing task"
- Lin (1991) provided the first implementation of Q-Learning with NNs, including the use of an experience replay buffer.
- Tesauro (1992, 1994, 1995, 2002) was responsible for the first major empirical result combining NNs and RL. He developed an RL agent that eventually was able to play backgammon at the level of the world's strongest players.
- Neural Fitted Q-Iteration (Riedmiller, 2005) is another key algorithm, quite famous even before the whole rise of deep reinforcement learning.

But, in a sense, we didn't know how (nor needed) to use NNs

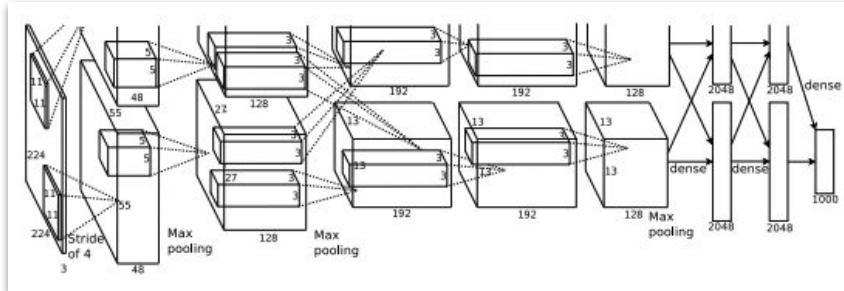
- RL algorithms were mostly simple consumers of supervised learning in a plug-and-play fashion (e.g., TD-Gammon was $TD(\lambda)$ with non-linear FA).
- In the supervised learning community, neural networks were not the state-of-the-art, things like SVMs and boosting were what everyone used. Initially, they were not that conducive to be done with SGD $\backslash(\cup)_/\backslash$
- The problems we looked at were often too simple (and low-dimensional).



The rise of deep learning



ImageNet



Classification Results (CLS)



<https://medium.com/@prudhvi.gnv/imagenet-challenge-advancement-in-deep-learning-and-computer-vision-124fd33cb948>

Krizhevsky et al. (2012)

The rise of deep RL



Different RL problems started to be proposed as well

Journal of Artificial Intelligence Research 47 (2013) 253–279

Submitted 02/13; published 06/13

The Arcade Learning Environment: An Evaluation Platform for General Agents

Marc G. Bellemare

University of Alberta, Edmonton, Alberta, Canada

MG17@CS.UALBERTA.CA

Yavar Naddaf

*Empirical Results Inc., Vancouver,
British Columbia, Canada*

YAVAR@EMPIRICALRESULTS.CA

Joel Veness

Michael Bowling

University of Alberta, Edmonton, Alberta, Canada

VENESS@CS.UALBERTA.CA

BOWLING@CS.UALBERTA.CA

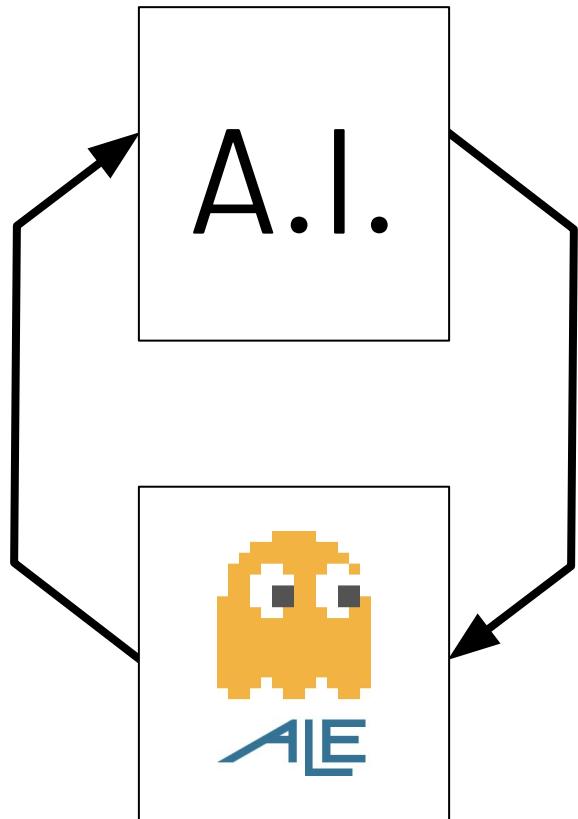
Abstract

In this article we introduce the Arcade Learning Environment (ALE): both a challenge problem and a platform and methodology for evaluating the development of general, domain-independent AI technology. ALE provides an interface to hundreds of Atari 2600

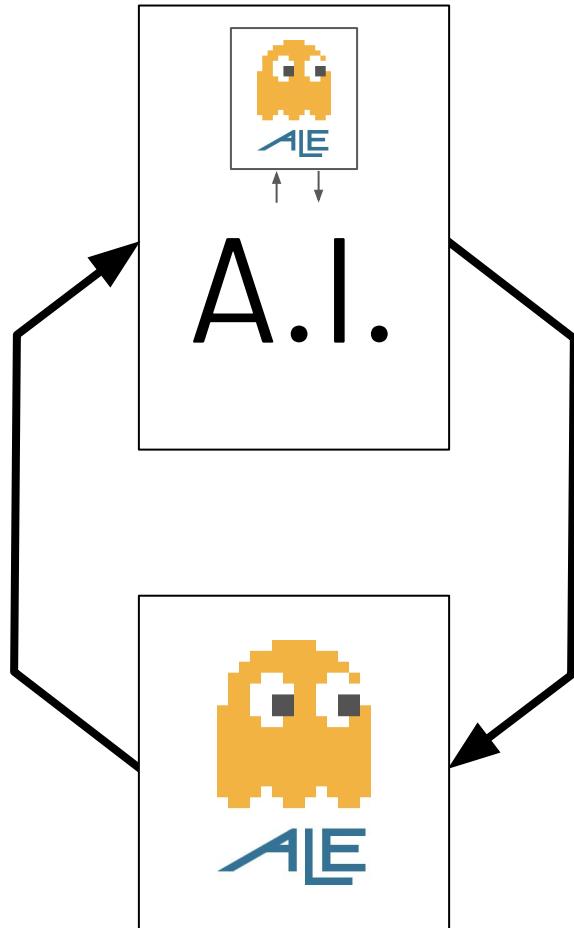
Arcade Learning Environment

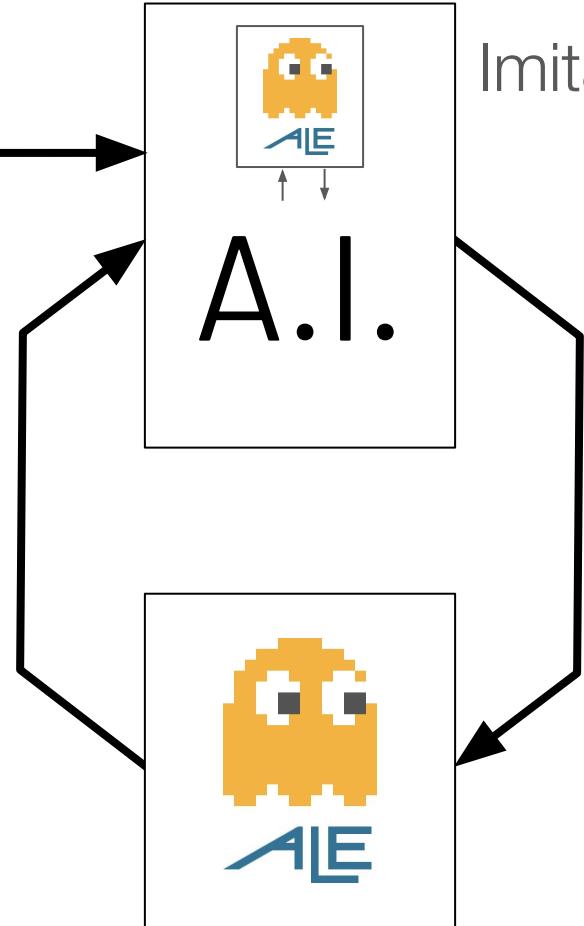
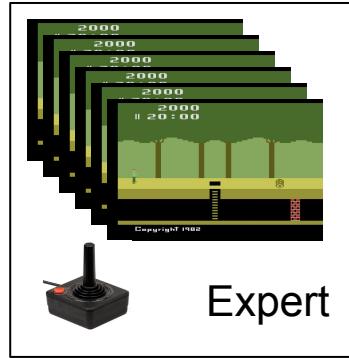
Over 50 Domains in 8 Minutes 23 Seconds

Reinforcement Learning



Model Learning

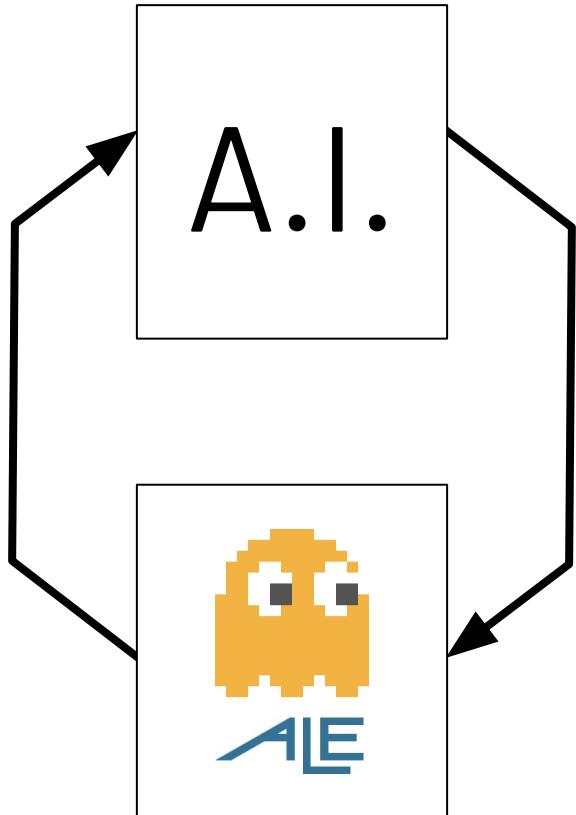




Imitation/Apprenticeship Learning



Exploration

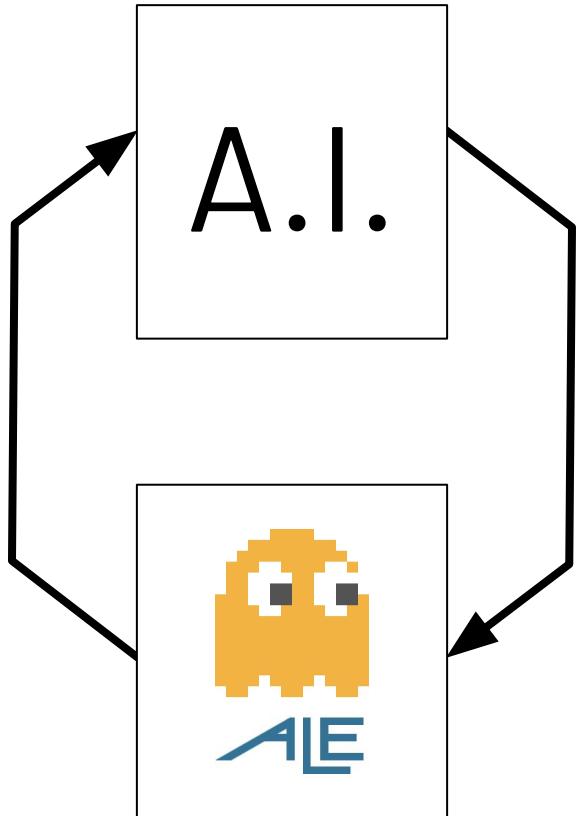


Transfer Learning

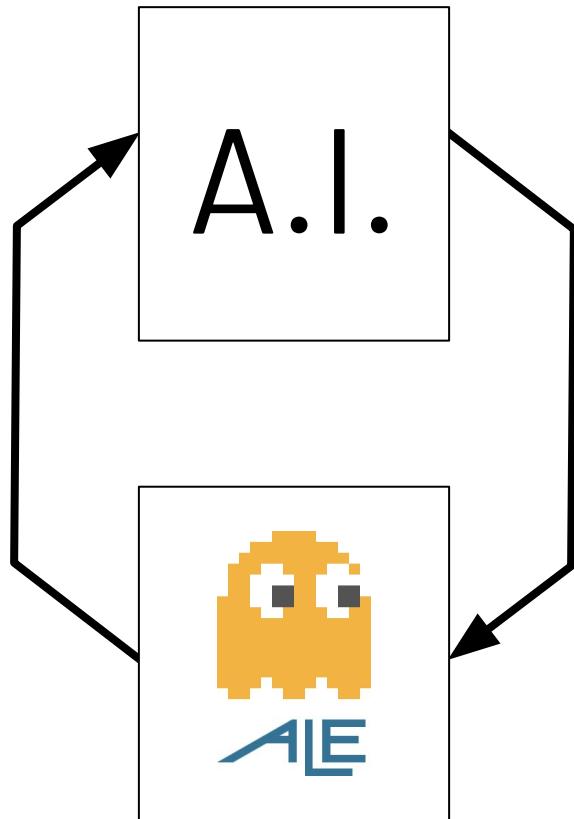
Pitfall!



Pitfall II



Intrinsic Motivation



Papers were written on the topic and many others tried

Sketch-Based Linear Value Function Approximation

Marc G. Bellemare
University of Alberta
mg17@cs.ualberta.ca

Joel Veness
University of Alberta
veness@cs.ualberta.ca

Michael Bowling
University of Alberta
bowling@cs.ualberta.ca

Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Investigating Contingency Awareness Using Atari 2600 Games

Marc G. Bellemare and Joel Veness and Michael Bowling

HyperNEAT-GGP: A HyperNEAT-based Atari General Game Player

Bayesian Learning of Recursively Factored Value Functions

Matthew Hausknecht, Piyush Khandelwal, Risto Miikkulainen, Peter Stone
Department of Computer Science

Marc Bellemare
Joel Veness
Michael Bowling
University of Alberta, Edmonton, Canada, T6G 2E8

IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, VOL. 6, NO. 4, DECEMBER 2014

355

A Neuroevolution Approach to General Atari Game Playing

Matthew Hausknecht, Joel Lehman, Risto Miikkulainen, and Peter Stone

Deep learning and RL were finally combined

[Mnih et al., 2013, 2015]

Dec 2013

Playing Atari with Deep Reinforcement Learning

Volodymyr Mnih Koray Kavukcuoglu David Silver Alex Graves Ioannis Antonoglou
Daan Wierstra Martin Riedmiller

DeepMind Technologies

{vlad, koray, david, alex.graves, ioannis, daan, martin.riedmiller} @ deepmind.com

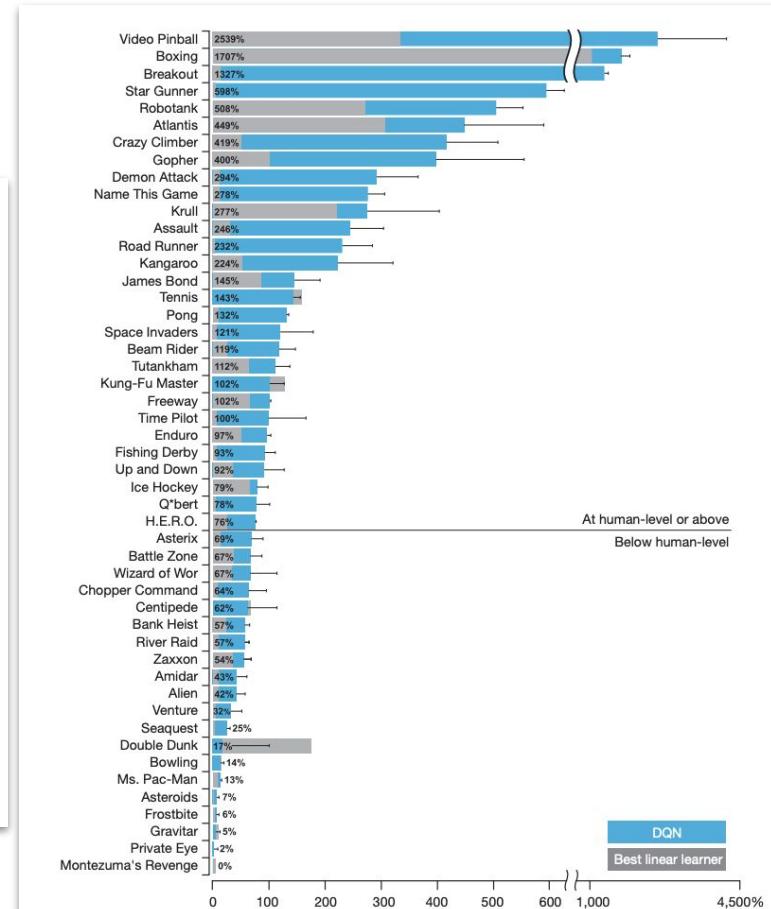
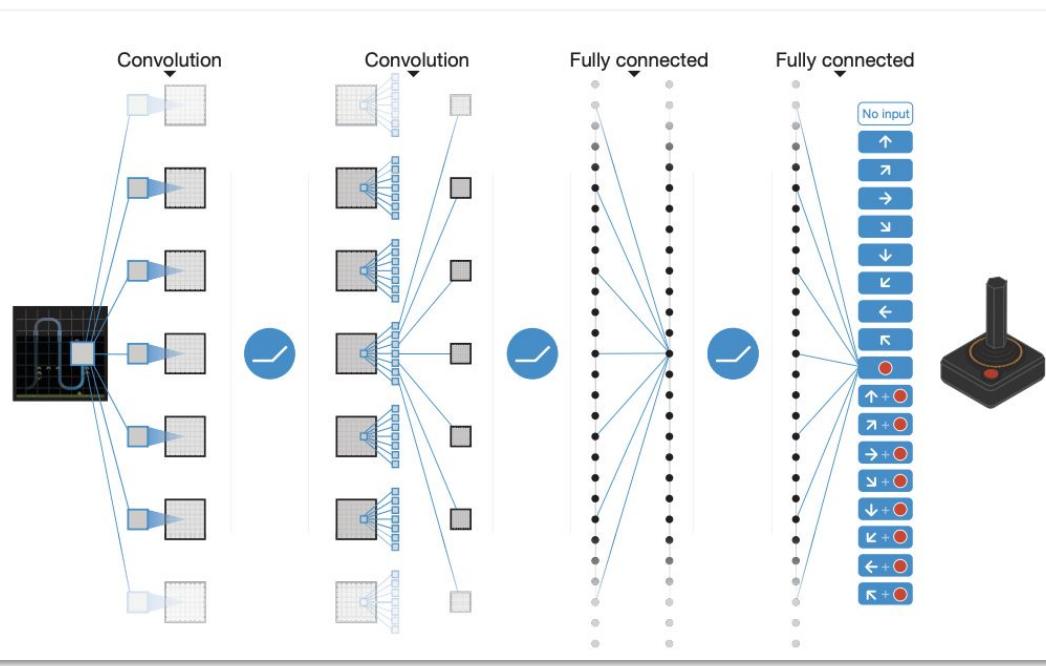
Abstract





Deep Q-Network (and Deep RL)

[Mnih et al., 2013, 2015]



Deep RL caught everyone's attention for its potential



The image shows a screenshot of a TechCrunch news article. The header features the TechCrunch logo (TC) and a 'Join TechCrunch+' button. Below the header, there are links for 'Login' and a search bar with the placeholder 'Search Q'. The main content is a green 'Startups' category tag. The main title of the article is 'Google Acquires Artificial Intelligence Startup DeepMind For More Than \$500M', written in large, bold, black text. Below the title, the author is listed as 'Catherine Shu' with the handle '@catherineshu', and the publication details are '6:20 PM MST • January 26, 2014'. To the right of the author information is a 'Comment' section icon.

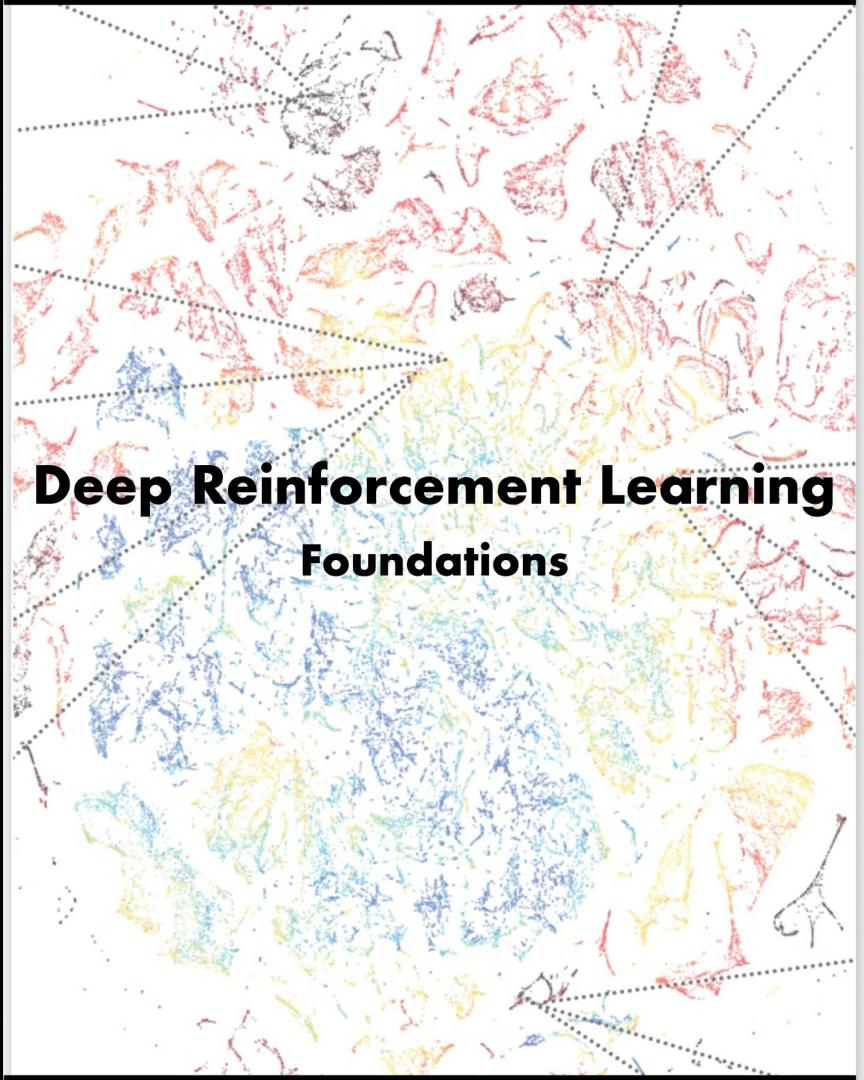
Startups

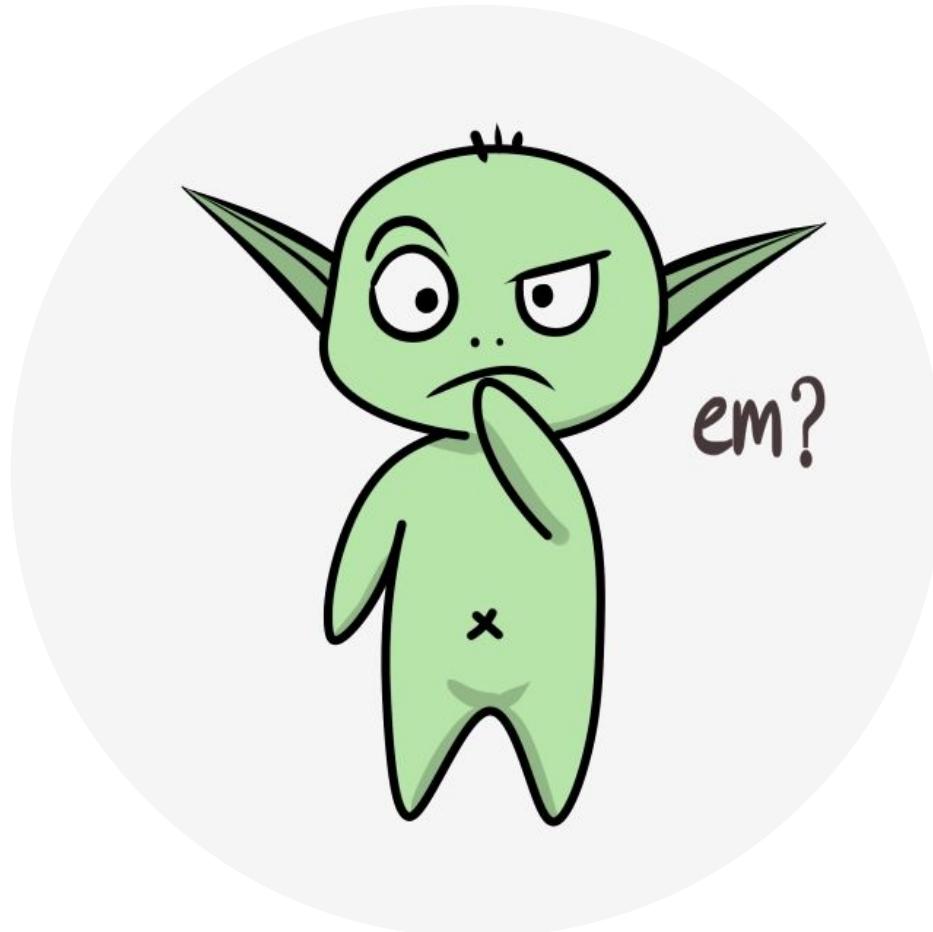
Google Acquires Artificial Intelligence Startup DeepMind For More Than \$500M

Catherine Shu @catherineshu / 6:20 PM MST • January 26, 2014

Comment

And that's what we
are going to study!





There are
some caveats



Deep RL is not RL + NNs

- NNs have been used with reinforcement learning for almost 40 years now.
- Papers were published in 2012 and 2013 on playing Atari with NNs.

But somehow, these don't feel like modern deep RL.

Deep RL is not RL + NNs

- NNs have been used with reinforcement learning for almost 40 years now.
- Papers were published in 2012 and 2013 on playing Atari with NNs.

But somehow, these don't feel like modern deep RL.

- Maybe it is just a matter of clever branding, but there was a paradigm shift.
- This shift affected both solution methods and the problems investigated (and also in the amount of computation people became comfortable using).
- This shift led to major successes and deployment of deep RL algorithms in the real world; so thinking about deep RL as being RL + NNs doesn't seem right.

A definition of deep reinforcement learning

The solution methods in deep RL **involve using neural networks** to parameterize artifacts such as value functions, policies, or models. Importantly, deep RL studies the design of algorithms to **explicitly allow for the use of modern deep learning techniques**. This is in contrast to being agnostic to the function class used for function approximation and simply seeing RL as a consumer of supervised learning techniques (specifically deep learning) in a plug-and-play fashion. Additionally, due to the generality of neural networks, deep RL methods **are often expected to be applicable to a wide range of problems** without necessarily requiring significant changes in the algorithm itself, only hyperparameter tuning. In terms of the problems, deep reinforcement learning has historically studied control (and prediction) **problems with high-dimensional observations**, as those are the problems one would expect a complex function approximator to be needed.

None of the major problems in RL go away with deep RL

- Exploration, credit assignment, and generalization are still a thing.
- In a sense, deep RL adds an extra challenge as many deep RL algorithms can be quite complex and unstable due to using neural networks.
- But the implicit assumption is that embracing the additional complexity introduced by NNs is worth it, mainly because of generalization.
- It is important to acknowledge a tension around appreciating and embracing the RL problem formulation and the fact that deep learning techniques benefit from problem formulations closer to supervised learning.

What are we covering?

- Part I: Course Overview, Background on RL and NNs (~4 classes)
- Part II: Value-based Model-Free Methods (~8 classes)
- Part III: Policy Gradient Methods (~4 classes)
- Part IV: Model-based Methods (~2 classes)
- Part V: Your talks (6 classes)



Teaching deep RL is hard

- Deep RL is primarily an empirical field.
- Deep RL algorithms (and/or the environments they are evaluated in) often have a high computational demand. E.g., it takes ~4 days to get Atari 2600 results.
- Because of that (and more) researchers started reporting results over few runs.
- Also, researchers often apply wrong statistical tests, don't properly tune hyperparameters or explore alternative design choices, and the evaluation protocols vary wildly across different research groups.
- Most claims revolve around “state-of-the-art” performance, not understanding.
- In this context, teaching deep RL is a nightmare. What do we *really* know?



Next class

- What I plan to do:
 - Start an overview of neural networks / deep learning.
 - Make the first assignment available for you, ish.
- What I recommend YOU to do for next class:
 - Brush-up on the basics of deep learning if you don't remember.
Specifically, Goodfellow, Bengio & Courville (2016)'s chapters 6–10.