Intro to Reinforcement Learning

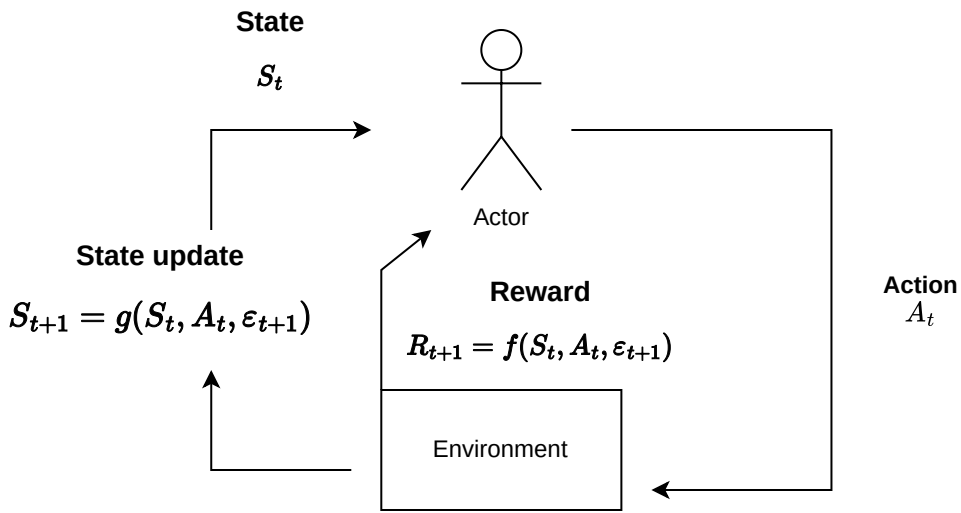
Chase Coleman + John Stachurski

IMF Workshop, 2025

Reinforcement learning

What is reinforcement learning?

Reinforcement learning (RL) is a machine learning approach where an "agent" learns to make a sequence of decisions by trying to maximize a reward it receives for its actions. The agent interacts with an "environment," takes an action, and receives feedback as a reward or penalty.



Reinforcement learnings vs dynamic programming

Economists know a lot about something that looks like reinforcement learning – Dynamic programming!

In fact, as you'll see below, they target essentially the same type of problem

Dynamic Programming

$$V(s_t) = \max_{a_t} u(a_t, s_t) + \beta E\left[V(s_{t+1}, \varepsilon_{t+1})\right]$$

Reinforcement learning

$$V(s_t) = \max_{a_t} E\left[R(a_t, s_t, \varepsilon_{t+1}) + \beta V(s_{t+1}, \varepsilon_{t+1})\right]$$

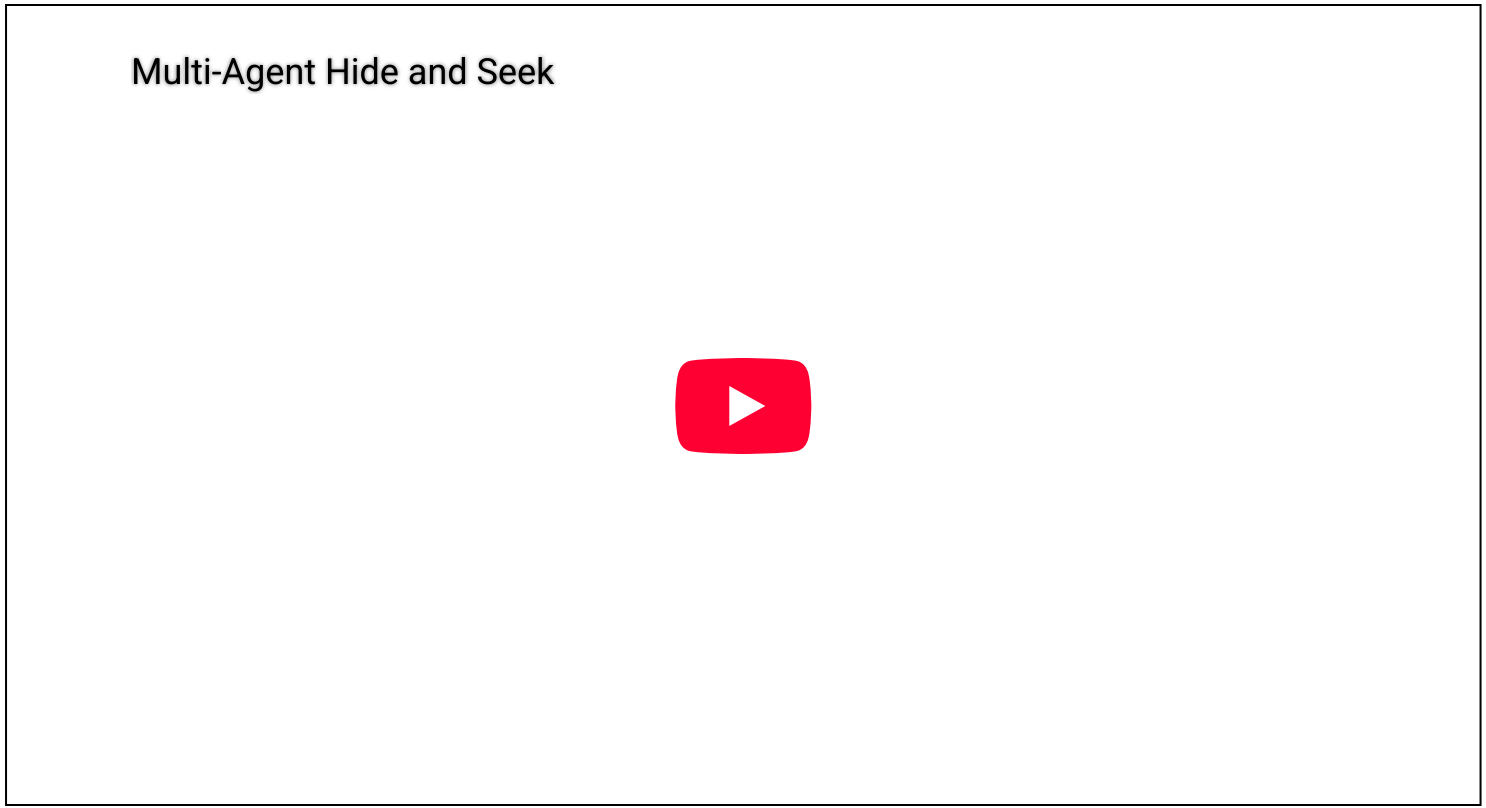
Reinforcement learnings vs dynamic programming

However, there are a few slight differences:

- In dynamic programming the decision maker typically has a deep, and perfect, understanding of how model works and understands the risks and the probability distributions that they face.
- In reinforcement learning, you only endow the agent with information on what they are allowed to do and there is no explicit information about the environment.
- Dynamic programming typically is solved by iterating over *all possible states* while in reinforcement learning, the agent typically just *learns through experience*.

Examples of success

Examples of reinforcement learning: Hide and seek



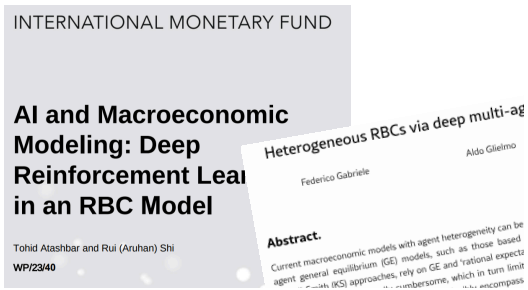
Examples of reinforcement learning: Go

Google DeepMind: Ground-breaking AlphaGo masters the game of Go

Examples of reinforcement learning: LLMs

Reinforcement learning is terrible – Andrej Karpathy

Examples of reinforcement learning: Economics



INTERNATIONAL MONETARY FUND

AI and Macroeconomic Modeling: Deep Reinforcement Lea in an RBC Model

Tohid Atashbar and Rui (Aruhan) Shi

WP/23/40

Heterogeneous RBCs via deep multi-ag

Federico Gabriele

Aldo Gilielmo

Abstract.

Current macroeconomic models with agent heterogeneity can be agent general equilibrium (GE) models, such as those based (KS) approaches, rely on GE and "rational expectu cumbersome, which in turn limit encompass

Open Access Article

Deep Reinforcement Learning in a Search-Matching Model of Labor Market Fluctuations

by Ruxin Chen

Department of Economics, Nagoya University, Aichi 464-8601, Japan

Economies **2025**, *13*(10), 302; https://doi.org/10.3390/economies13100302

Submission received: 4 September 2025 / Revised: 9 October 2025 / Accepted: 14 October 2025 / Published: 20 October 2025

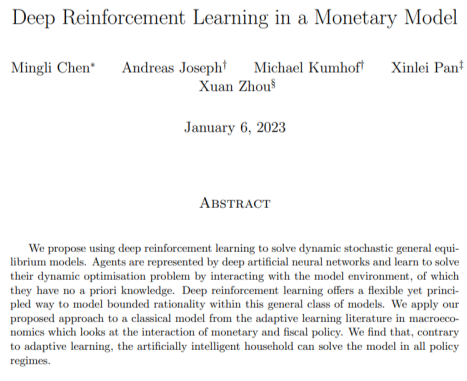
(This article belongs to the Topic Advanced Techniques and Modeling in Business and Economics)

Download Browse Figures Versions Notes

Abstract

Shimer documents that the search-and-matching model driven by productivity shocks explains only a small share of the observed volatility of unemployment and vacancies, which is known as the Shimer puzzle. We revisit this evidence by replacing the representative firm's optimization with a deep reinforcement learning (DRL) agent that learns its vacancy-posting policy through interaction in a Diamond–Mortensen–Pissarides (DMP) model. Comparing the learning economy with a conventional log-linearized DSGE solution under the same parameters, we find that while both frameworks preserve a downward-sloping Beveridge curve, learning-based economy produces much higher volatility in key labor market variables and returns to a steady state more slowly after shocks. These results point to bounded rationality and endogenous learning as mechanisms for labor market fluctuations and suggest that reinforcement learning can serve as a useful complement to standard macroeconomic analysis.

Keywords: search-and-matching model; labor market simulation; macroeconomic modeling; deep reinforcement learning



Deep Reinforcement Learning in a Monetary Model

Mingli Chen* Andreas Joseph† Michael Kumhof† Xinlei Pan‡ Xuan Zhou§

January 6, 2023

ABSTRACT

We propose using deep reinforcement learning to solve dynamic stochastic general equilibrium models. Agents are represented by deep artificial neural networks and learn to solve their dynamic optimisation problem by interacting with the model environment, of which they have no a priori knowledge. Deep reinforcement learning offers a flexible yet principled way to model bounded rationality within this general class of models. We apply our proposed approach to a classical model from the adaptive learning literature in macroeconomics which looks at the interaction of monetary and fiscal policy. We find that, contrary to adaptive learning, the artificially intelligent household can solve the model in all policy regimes.

step towards a synthesis of

Yaolang Zhong*

Abstract

This study delves into the numerical resolution of a critical Markov Equilibrium Games, employing a multi-agent deep re gorithm for strategy optimization. Specifically, it focuses on a duopoly context, resembling a Stackelberg game, where two firms engage in sequential decision-making each period. Characterized as model-free learners, these firms initially lack economic theoretical knowledge and develop optimal decision-making strategies solely through mutual interactions in simulations. Their policy functions are defined using neural networks, and training is executed via the Multi-Agent Deep Deterministic Policy Gradient (MA-DDPG) algorithm, a concept from deep reinforcement learning. The study aims to investigate if, under these conditions, the economy can attain an equilibrium where each firm's behavior is optimal. This exploration is set in a linear-quadratic framework, allowing for analytical derivation of the firms' optimal policy functions. The experimental findings indicate a nuanced outcome: the economy occasionally aligns with the analytical equilibrium, but diverges at times. These outcomes provide valuable insights for economists in refining model formulation and applying model-free numerical solutions in economic analysis.

Penn

orcement Learning

Jesús Fernández-Villaverde[1] and Galo Nuño[2]

October 4, 2025

[1]University of Pennsylvania

[2]Banco de España

The Bitter Lesson

The Bitter Lesson

Rich Sutton

March 13, 2019

The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. The ultimate reason for this is Moore's law, or rather its generalization of continued exponentially falling cost per unit of computation. Most AI research has been conducted as if the computation available to the agent were constant (in which case leveraging human knowledge would be one of the only ways to improve performance) but, over a slightly longer time than a typical research project, massively more computation inevitably becomes available. Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation. These two need not run counter to each other, but in practice they tend to. Time spent on one is time not spent on the other. There are psychological commitments to investment in one approach or the other. And the human-knowledge approach tends to complicate methods in ways that make them less suited to taking advantage of general methods leveraging computation. There were many examples of AI researchers' belated learning of this bitter lesson, and it is instructive to review some of the most prominent.

In computer chess, the methods that defeated the world champion, Kasparov, in 1997, were based on massive, deep search. At the time, this was looked upon with dismay by the majority of computer-chess researchers who had pursued methods that leveraged human understanding of the special structure of chess. When a simpler, search-based approach with special hardware and software proved vastly more effective, these human-knowledge-based chess researchers were not good losers. They said that ``brute force" search may have won this time, but it was not a general strategy, and anyway it was not how people played chess. These researchers wanted methods based on human input to win and were disappointed when they did not.

A similar pattern of research progress was seen in computer Go, only delayed by a further 20 years. Enormous initial efforts went into avoiding search by taking advantage of human knowledge, or of the special features of the game, but all those efforts proved irrelevant, or worse, once search was applied effectively at scale. Also important was the use of learning by self play to learn a value function (as it was in many other games and even in chess, although learning did not play a big role in the 1997 program that first beat a world champion). Learning by self play, and learning in general, is like search in that it enables massive computation to be brought to bear. Search and learning are the two most important classes of techniques for utilizing massive amounts of computation in AI research. In computer Go, as in computer chess, researchers' initial effort was directed towards utilizing human understanding (so that less search was needed) and only much later was much greater success had by embracing search and learning.

Toy environment

We are going to get started by learning about reinforcement learning in famous the cliff-walking example from Sutton Barto.

The rules of the cliff-walking game are as follows:

- You start at grid point (0, 0)
- Your goal is to reach (11, 0)
- You can go up, right, down, or left at each grid point.
- Each step you take costs 1 unit of effort
- There is a cliff that spans (0, 1) to (0, 11) and if you step on one of these squares then it takes 100 units of effort to climb back up the cliff and start again