資料探勘
(Data Mining)

強化學習
(Reinforcement Learning)

1092DM11
MBA, IM, NTPU (M5026) (Spring 2021)
Tue 2, 3, 4 (9:10-12:00) (B8F40)

Min-Yuh Day
戴敏育
Associate Professor
副教授
Institute of Information Management, National Taipei University
國立臺北大學 資訊管理研究所

https://web.ntpu.edu.tw/~myday

2021-06-01

課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)

1 2021/02/23 資料探勘介紹 (Introduction to data mining)

2 2021/03/02 ABC：人工智慧，大數據，雲端運算
(ABC: AI, Big Data, Cloud Computing)

3 2021/03/09 Python資料探勘的基礎
(Foundations of Data Mining in Python)

4 2021/03/16 資料科學與資料探勘：發現，分析，可視化和呈現數據
(Data Science and Data Mining:
Discovering, Analyzing, Visualizing and Presenting Data)

5 2021/03/23 非監督學習：關聯分析，購物籃分析
(Unsupervised Learning: Association Analysis,
Market Basket Analysis)

6 2021/03/30 資料探勘個案研究 I
(Case Study on Data Mining I)

課程大綱 (Syllabus)

週次 (Week)　日期 (Date)　內容 (Subject/Topics)

7 2021/04/06 放假一天 (Day off)

8 2021/04/13 非監督學習：集群分析，行銷市場區隔
(Unsupervised Learning: Cluster Analysis, Market Segmentation)

9 2021/04/20 期中報告 (Midterm Project Report)

10 2021/04/27 監督學習：分類和預測
(Supervised Learning: Classification and Prediction)

11 2021/05/04 機器學習和深度學習
(Machine Learning and Deep Learning)

12 2021/05/11 卷積神經網絡
(Convolutional Neural Networks)

課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)

13 2021/05/18 資料探勘個案研究 II
(Case Study on Data Mining II)

14 2021/05/25 遞歸神經網絡
(Recurrent Neural Networks)

15 2021/06/01 強化學習
(Reinforcement Learning)

16 2021/06/08 社交網絡分析
(Social Network Analysis)

17 2021/06/15 期末報告 I (Final Project Report I)

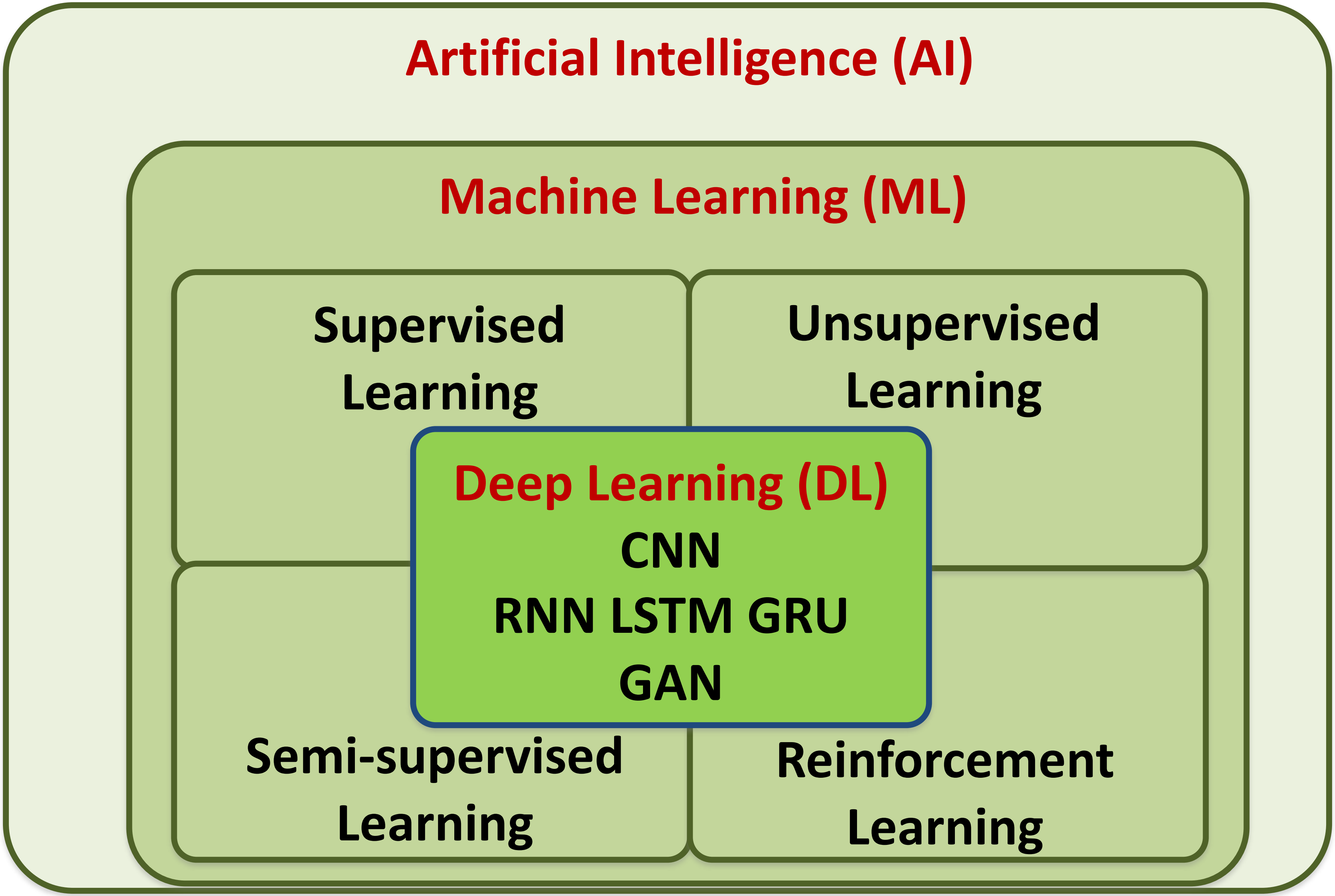
18 2021/06/22 期末報告 II (Final Project Report II)

Reinforcement Learning

Outline

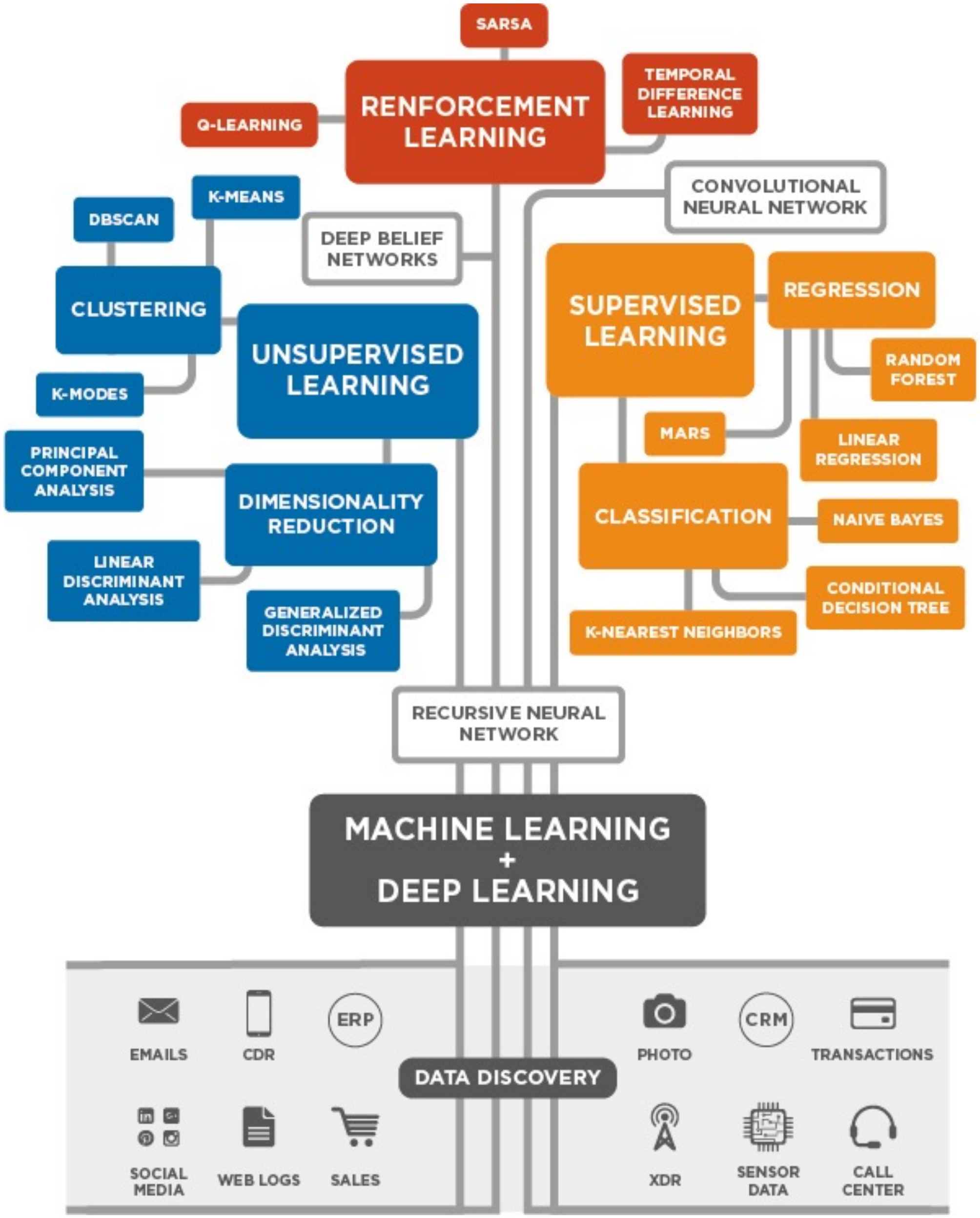
- Reinforcement Learning (RL)
  - Markov Decision Processes (MDP)
- Deep Reinforcement Learning (DRL) Algorithms
  - SARSA
  - Q-Learning
  - DQN
  - A3C
  - Rainbow

AI, ML, DL



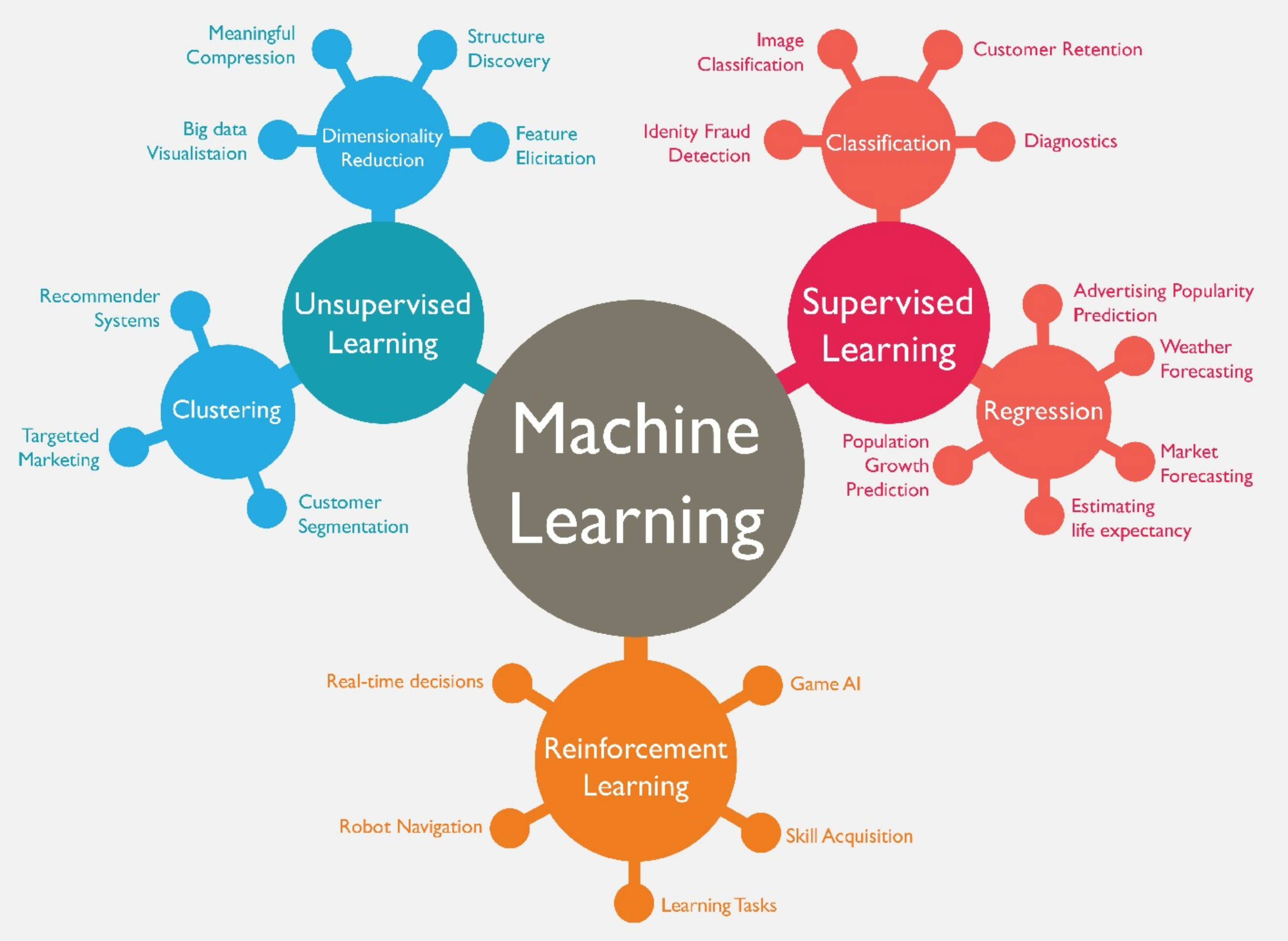
Source: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/deep_learning.html

3 Machine Learning Algorithms



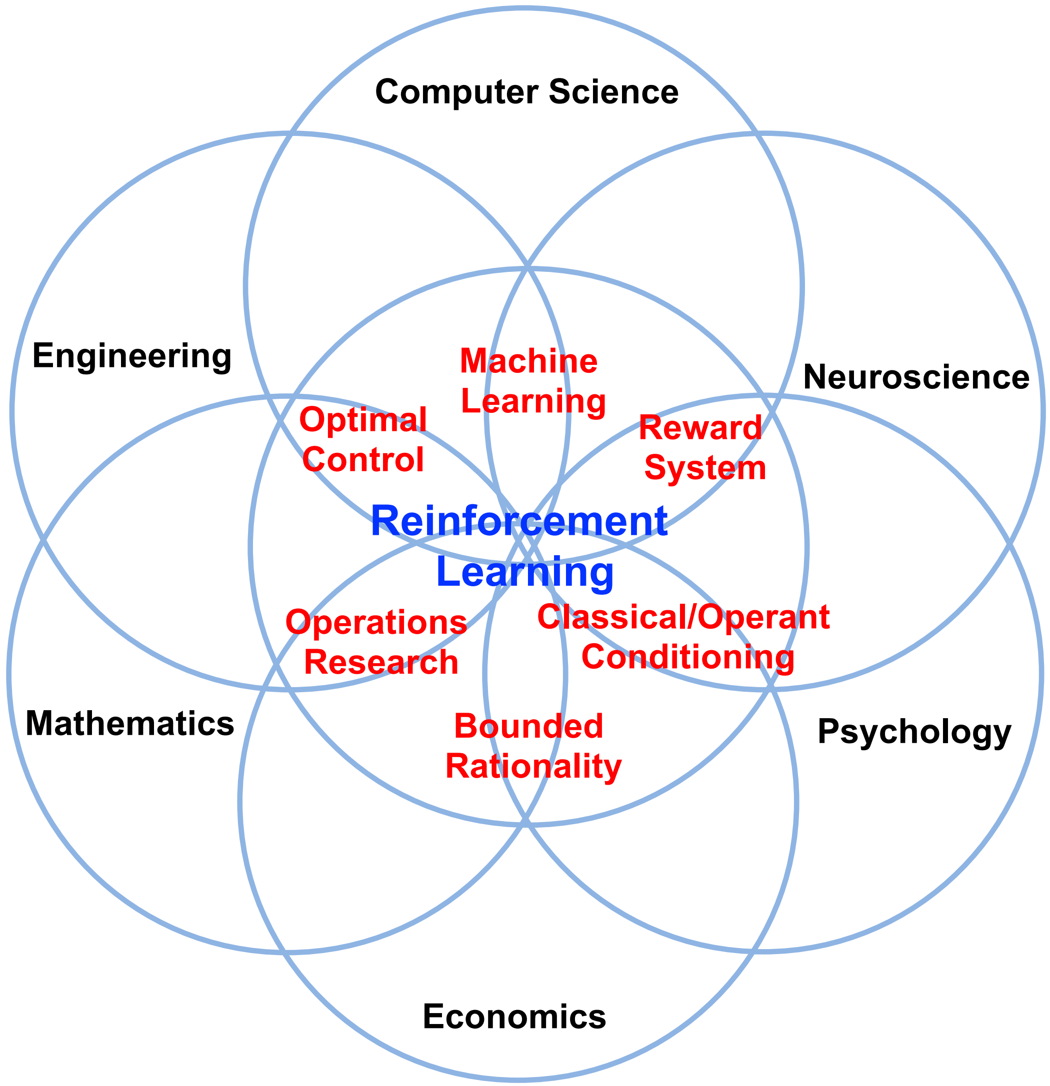
Source: Enrico Galimberti, http://blogs.teradata.com/data-points/tree-machine-learning-algorithms/

Machine Learning (ML)



Source: https://www.mactores.com/services/aws-big-data-machine-learning-cognitive-services/

Reinforcement Learning (RL)

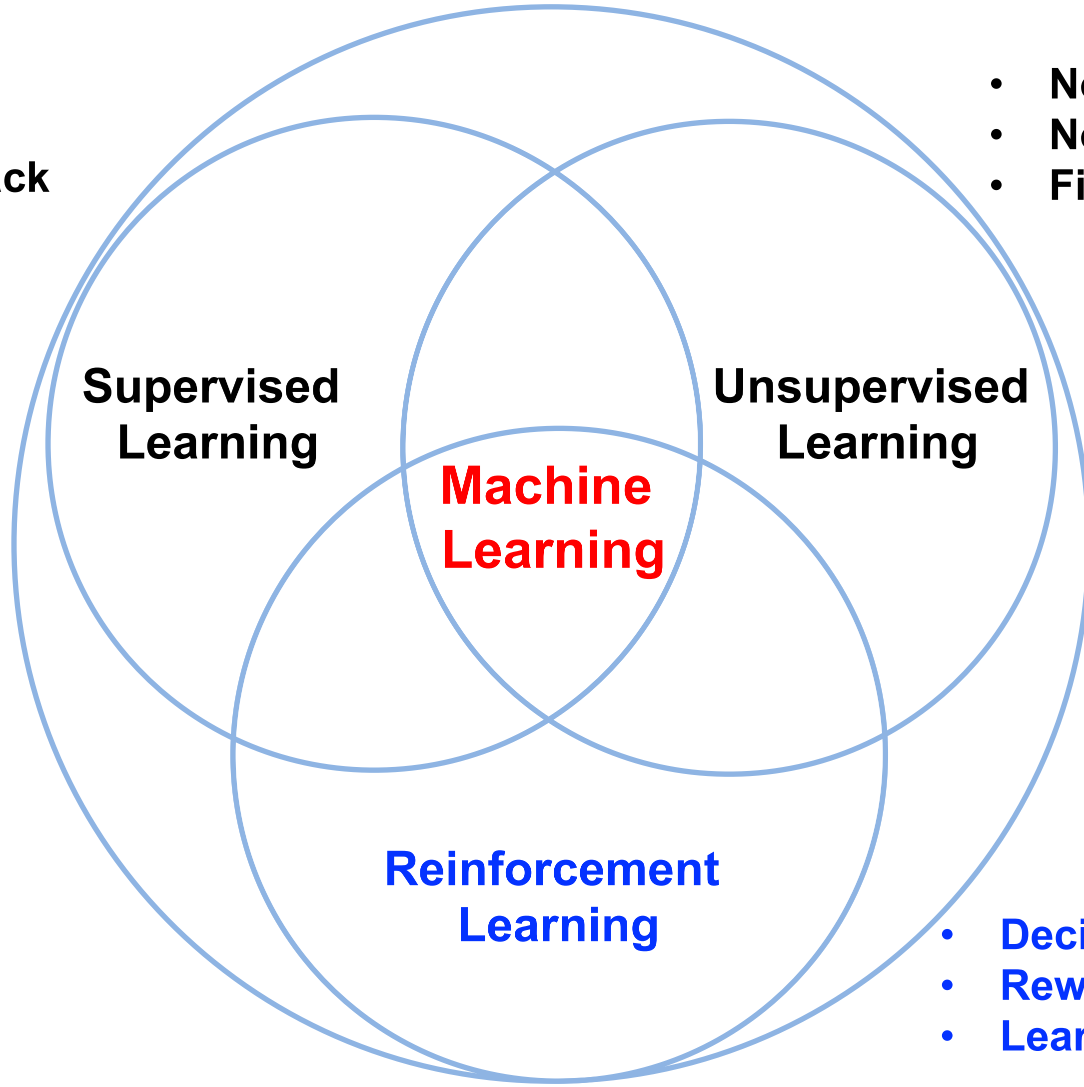


Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Branches of Machine Learning (ML)
Reinforcement Learning (RL)

- **Labeled data**
- **Direct feedback**
- **Predict**

- **No Labels**
- **No feedback**
- **Find hidden structure**



- **Decision process**
- **Reward system**
- **Learn series of actions**

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

David Silver (2015), Introduction to reinforcement learning

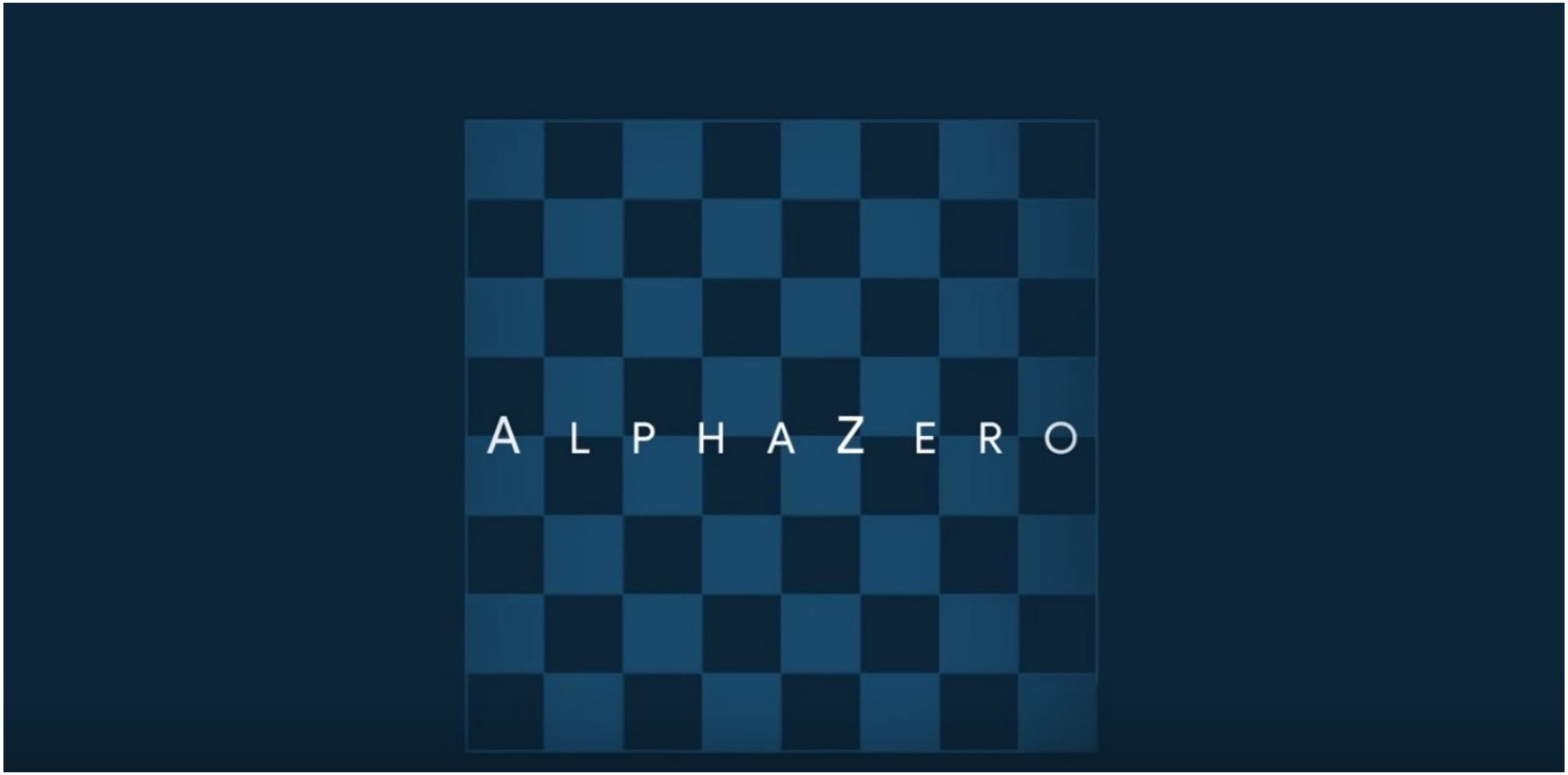
- **Elementary Reinforcement Learning**
  - 1: Introduction to Reinforcement Learning
  - 2: Markov Decision Processes
  - 3: Planning by Dynamic Programming
  - 4: Model-Free Prediction
  - 5: Model-Free Control
- **Reinforcement Learning in Practice**
  - 6: Value Function Approximation
  - 7: Policy Gradient Methods
  - 8: Integrating Learning and Planning
  - 9: Exploration and Exploitation
  - 10: Case Study: RL in Classic Games

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement Learning
AlphaZero (AZ) and AlphaGo Zero (AZ0)

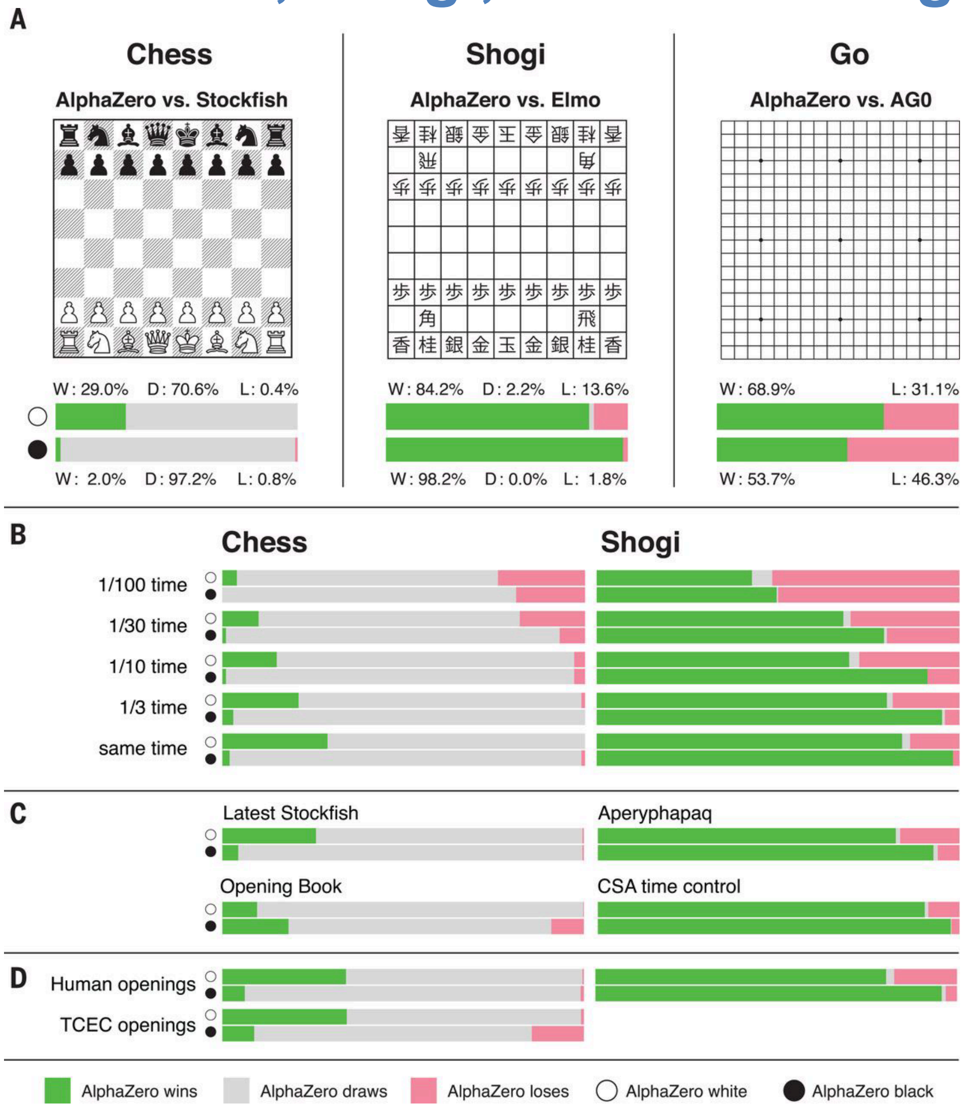
- AlphaZero (Silver et al., 2018)
  - A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. (Science)
- AlphaGo Zero (Silver et al., 2017)
  - Mastering the game of Go without human knowledge (Nature)

AlphaZero: Shedding new light on the grand games of chess, shogi and Go



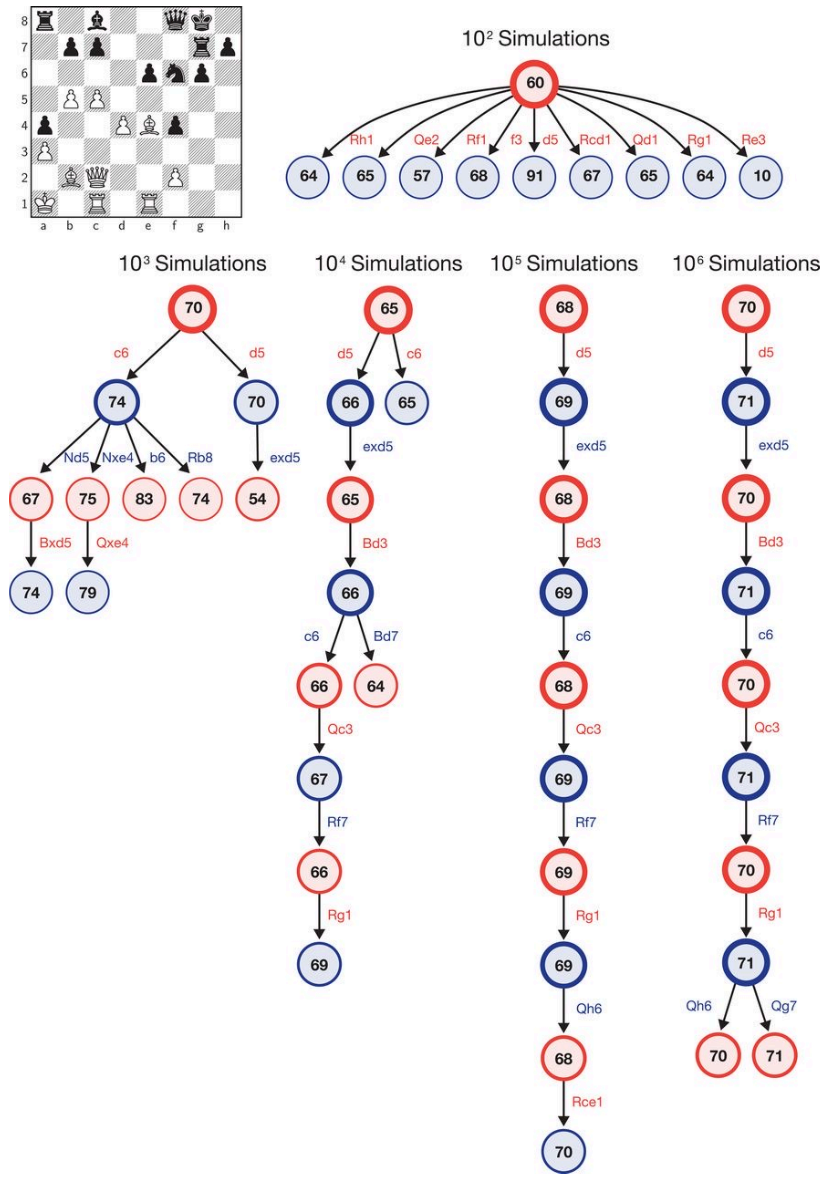
https://www.youtube.com/watch?v=7L2sUGcOgh0

AlphaZero
A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play



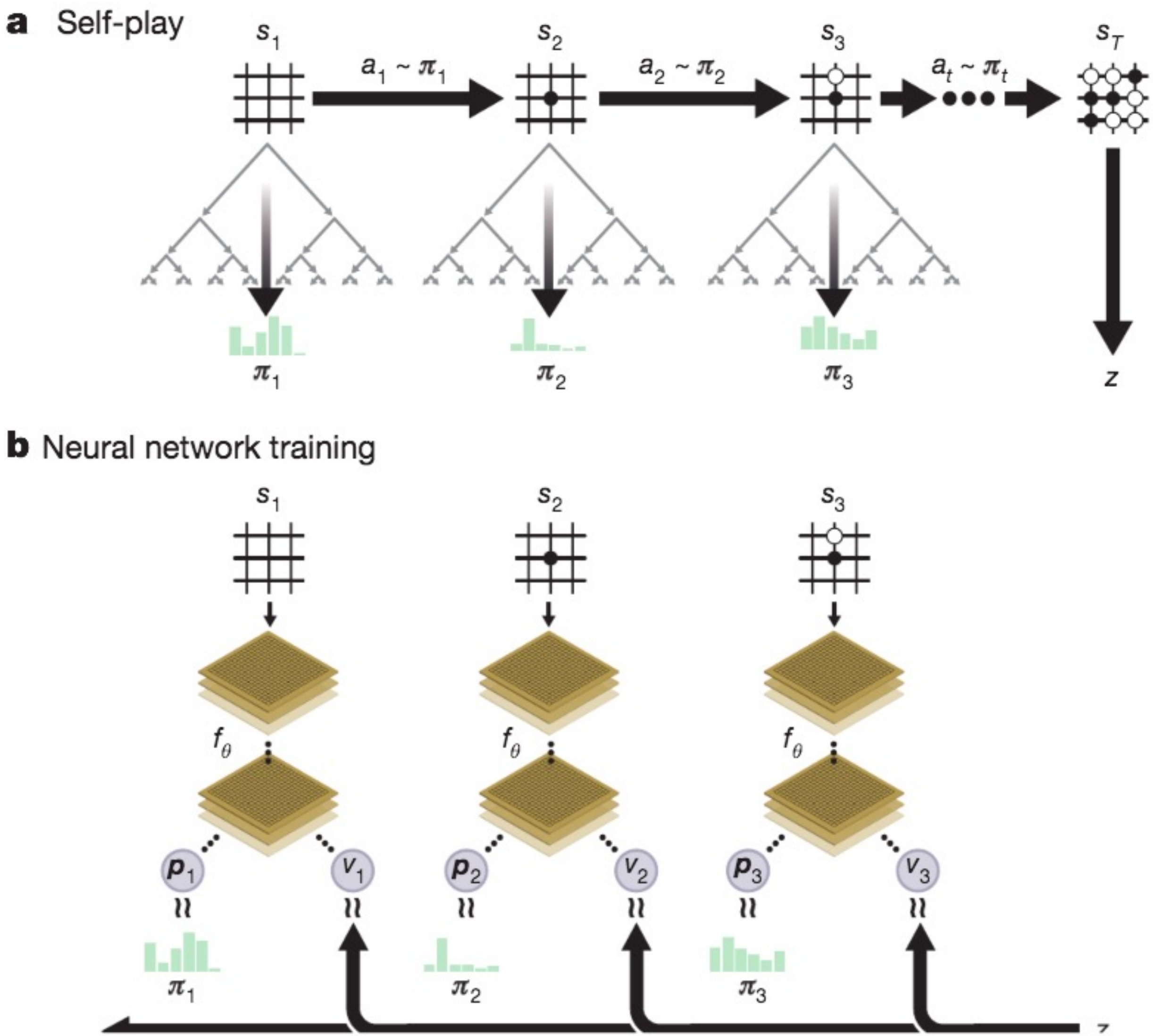
Source: David Silver et al. (2018), "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play." Science 362, no. 6419 (2018): 1140-1144.

AlphaZero’s search procedure



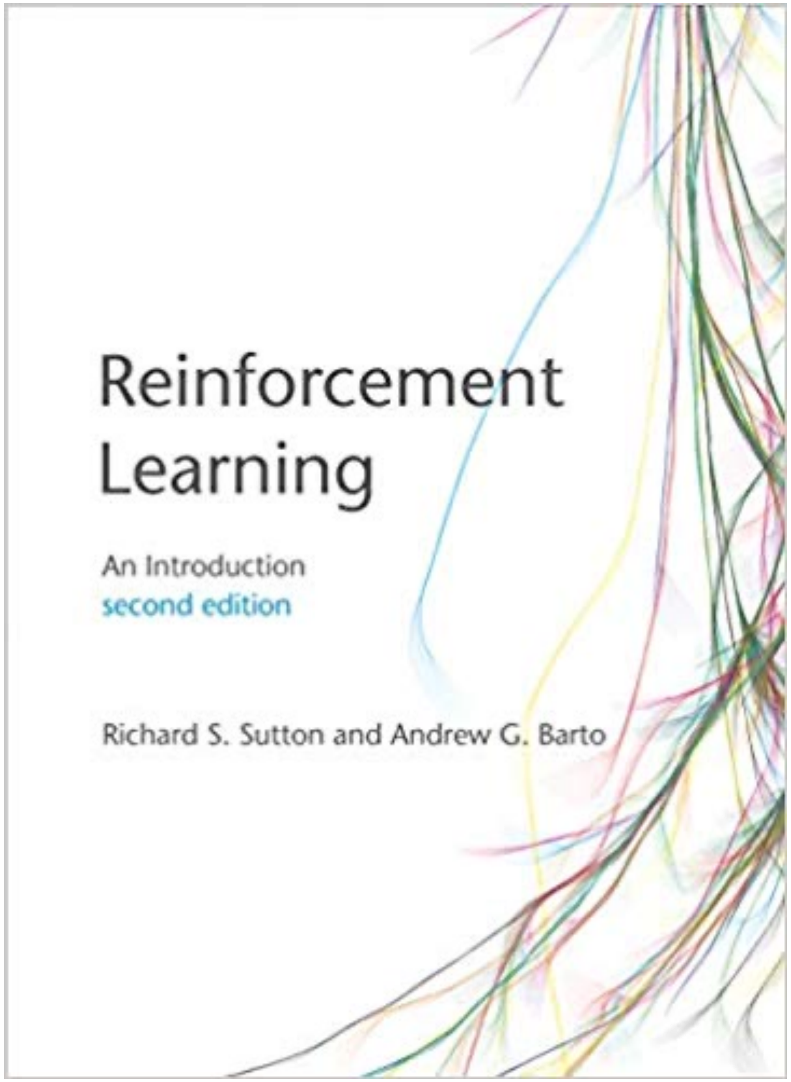
Source: David Silver et al. (2018), "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play." Science 362, no. 6419 (2018): 1140-1144.

Self-play reinforcement learning in AlphaGo Zero



Source: David Silver et al. (2017), "Mastering the game of Go without human knowledge." Nature 550 (2017): 354–359.

Richard S. Sutton & Andrew G. Barto (2018),
Reinforcement Learning: An Introduction,
2nd Edition, A Bradford Book



Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.
https://www.amazon.com/Reinforcement-Learning-Introduction-Adaptive-Computation/dp/0262039249

Reinforcement learning

- Reinforcement learning is
  learning what to do
  —how to map situations to actions
  —so as to maximize a numerical
  reward signal.

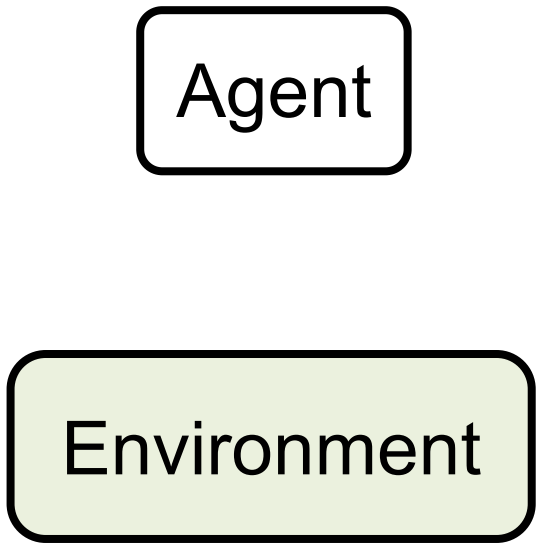
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Two most important distinguishing features of reinforcement learning

- trial-and-error search
- delayed reward

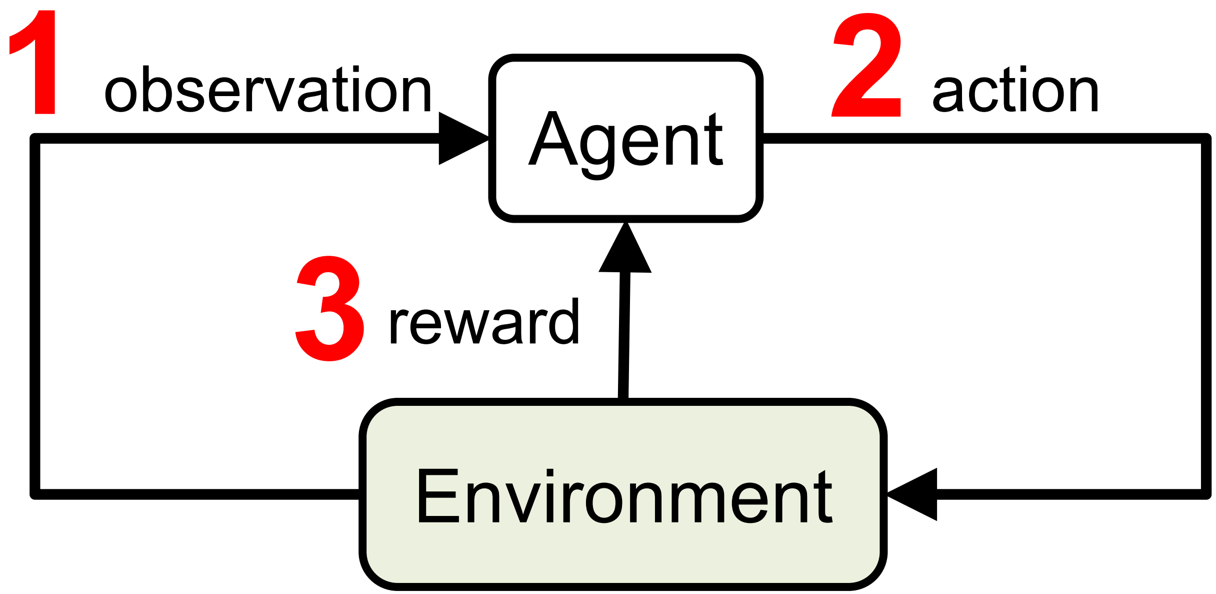
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning (DL)



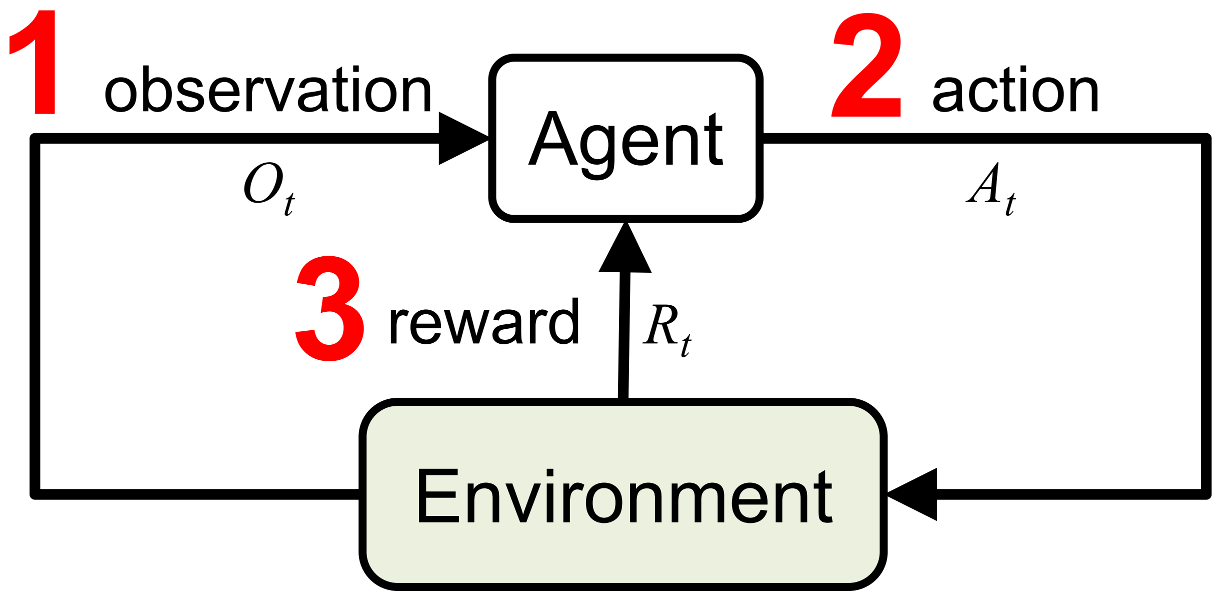
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning (DL)



Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

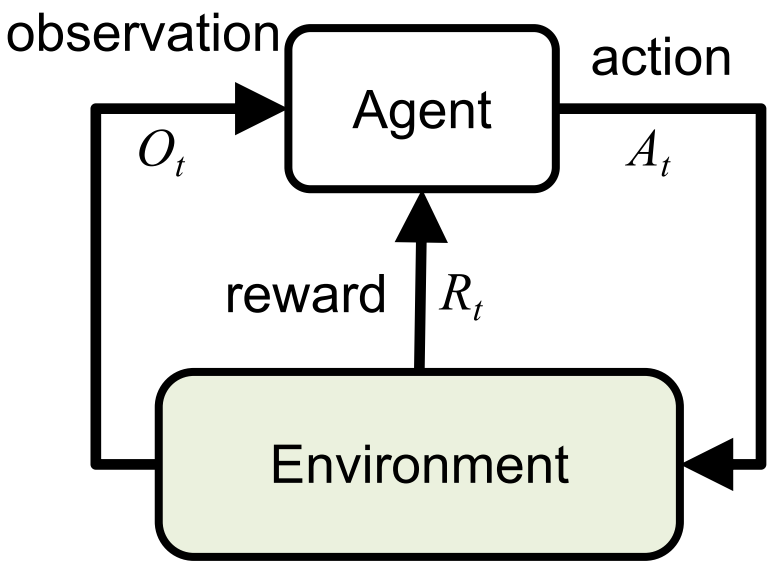
Reinforcement Learning (DL)



Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Agent and Environment

- At each step t the agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at env. step



Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

History and State

- The history is the sequence of observations, actions, rewards

$$H_t = O_1, A_1, R_1, \ldots, A_{t-1}, O_t, R_t$$

- i.e. all observable variables up to time t
- i.e. the sensorimotor stream of a robot or embodied agent
- What happens next depends on the history:
  - The agent selects actions
  - The environment selects observations/rewards
- State is the information used to determine what happens next
- Formally, state is a function of the history:

$$S_t = f(H_t)$$

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Information State

- An information state (a.k.a. Markov state) contains all useful information from the history.
- Definition

  A state S_t is Markov if and only if

  $$P[S_{t+1} \mid S_t] = P[S_{t+1} \mid S_1, \ldots, S_t]$$

- "The future is independent of the past given the present"

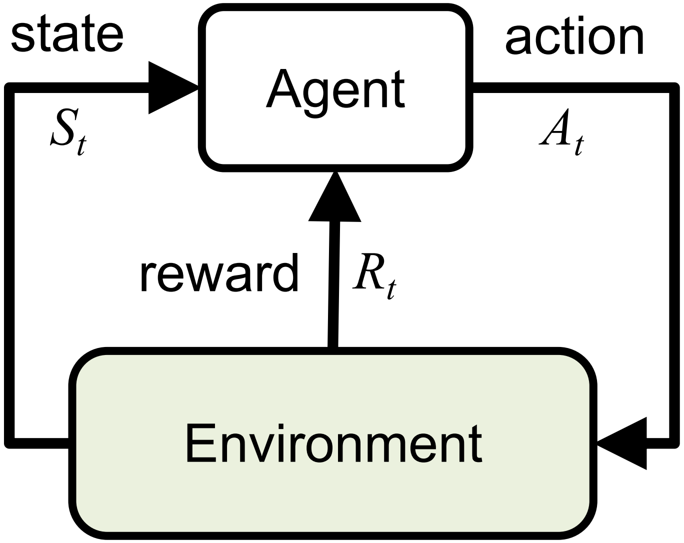
  $$H_{1:t} \rightarrow S_t \rightarrow H_{t+1:\infty}$$

- Once the state is known, the history may be thrown away i.e. The state is a sufficient statistic of the future
- The environment state S_t^e is Markov
- The history H_t is Markov

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Fully Observable Environments

- Full observability:
  - agent directly observes environment state
  - Agent state = environment state = information state
  - Formally, this is a Markov decision process (MDP)



Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

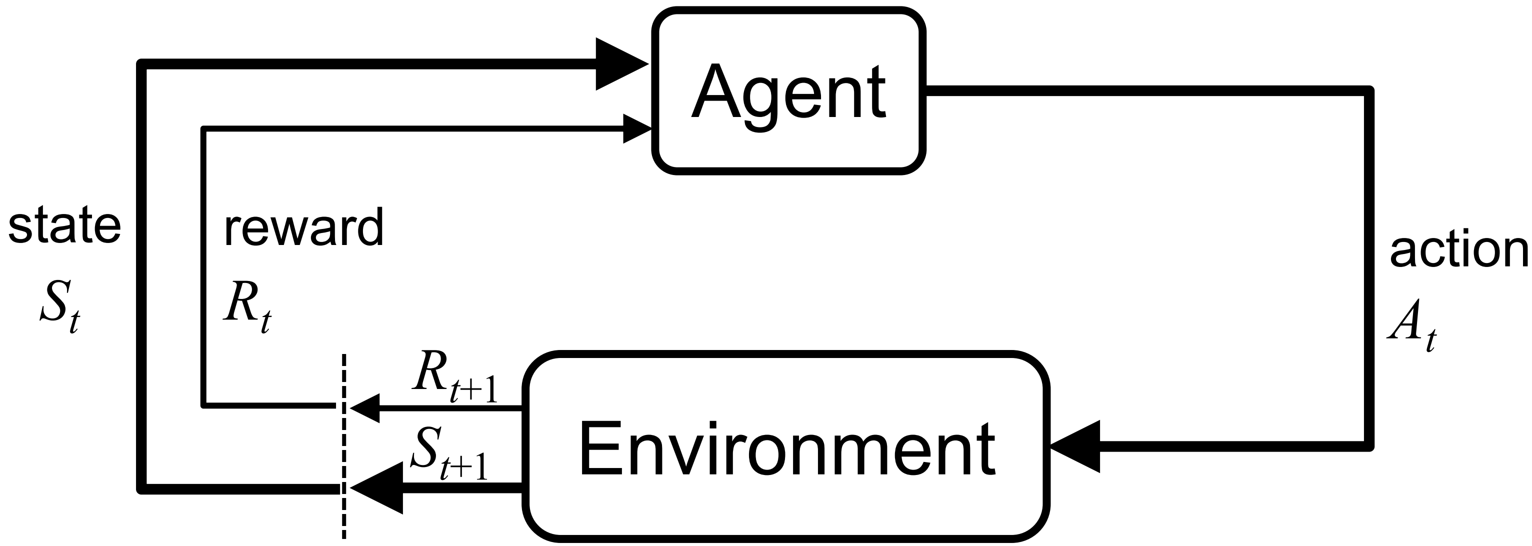
Partially Observable Environments

- Partial observability: agent indirectly observes environment
  - A robot with camera vision isn't told its absolute location
  - A trading agent only observes current prices
  - A poker playing agent only observes public cards
- Now agent state ≠ environment state
- Formally this is a partially observable Markov decision process (POMDP)
- Agent must construct its own state representation S^a_t, e.g.
  - Complete history: $S^a_t = H_t$
  - Beliefs of environment state: $S^a_t = (P[S^e_t = s_1],\ldots,P[S^e_t = s_n])$
  - Recurrent neural network: $S^a_t = \sigma(S^a_{t-1}\, W_s + O_t\, W_o)$

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement Learning (DL)

The Agent-Environment Interaction in a Markov Decision Process (MDP)



Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Characteristics of Reinforcement Learning

- No supervisor, only a reward signal
- Feedback is delayed, not instantaneous
- Time really matters (sequential, non i.i.d data)
- Agent's actions affect the subsequent data it receives

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Examples of Reinforcement Learning

- Make a humanoid robot walk
- Play may different Atari games better than humans
- Manage an investment portfolio

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Examples of Rewards

- Make a humanoid robot walk
  - +ve reward for forward motion
  - -ve reward for falling over
- Play may different Atari games better than humans
  - +/-ve reward for increasing/decreasing score
- Manage an investment portfolio
  - +ve reward for each $ in bank

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Sequential Decision Making

- Goal: select actions to maximize total future reward
- Actions may have long term consequence
- Reward may be delayed
- It may be better to sacrifice immediate reward to gain more long-term reward
- Examples:
  - A financial investment (may take months to mature)
  - Blocking opponent moves (might help winning chances many moves from now)

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Elements of Reinforcement Learning

- Agent
- Environment
- Policy
- Reward signal
- Value function
- Model

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Elements of Reinforcement Learning

- Policy
  - Agent’s behavior
  - It is a map from state to action
- Reward signal
  - The goal of a reinforcement learning problem
- Value function
  - How good is each state and/or action
  - A prediction of future reward
- Model
  - Agent’s representation of the environment

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Major Components of an RL Agent

1. **Policy**: agent's behaviour function
2. **Value** function: how good is each state and/or action
3. **Model**: agent's representation of the environment

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Policy

- A policy is the agent's behaviour
- It is a map from state to action, e.g.
  - Deterministic policy: $a = \pi(s)$
  - Stochastic policy: $\pi(a|s) = P[A_t = a|S_t = s]$

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Value Function

- Value function is a prediction of future reward
- Used to evaluate the goodness/badness of states
- And therefore to select between actions, e.g.

$$v_\pi(s) = E_\pi [R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | S_t = s]$$

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Model

- A model predicts what the environment will do next
- P predicts the next state
- R predicts the next (immediate) reward, e.g.

$$P^a_{ss'} = P[S_{t+1} = s' \mid S_{t+1} = s, A_t = a]$$

$$R^a_s = E[R_{t+1} \mid S_t = s, A_t = a]$$

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement Learning

- Value Based
  - No Policy (Implicit)
  - Value Function
- Policy Based
  - Policy
  - No Value Function
- Actor Critic
  - Policy
  - Value Function

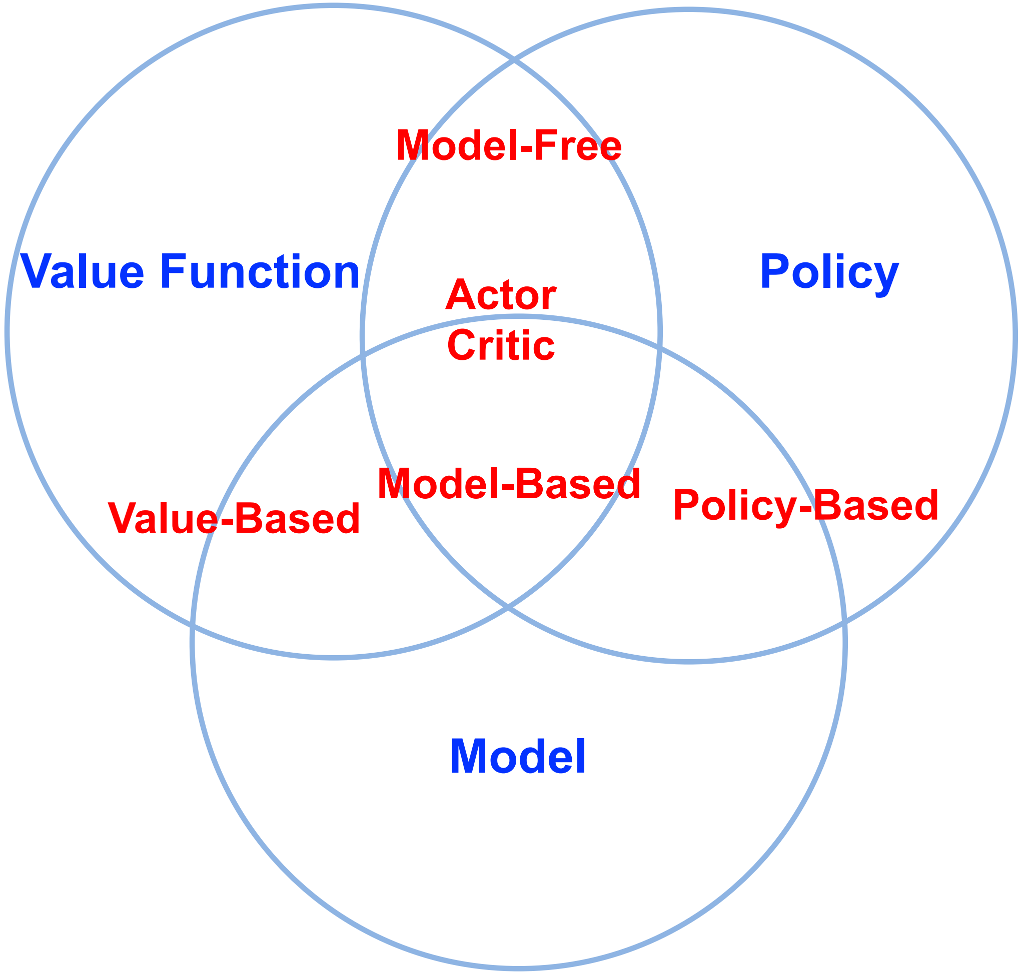
Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement Learning

- Model Free
  - Policy and/or Value Function
  - No Model
- Model Based
  - Policy and/or Value Function
  - Model

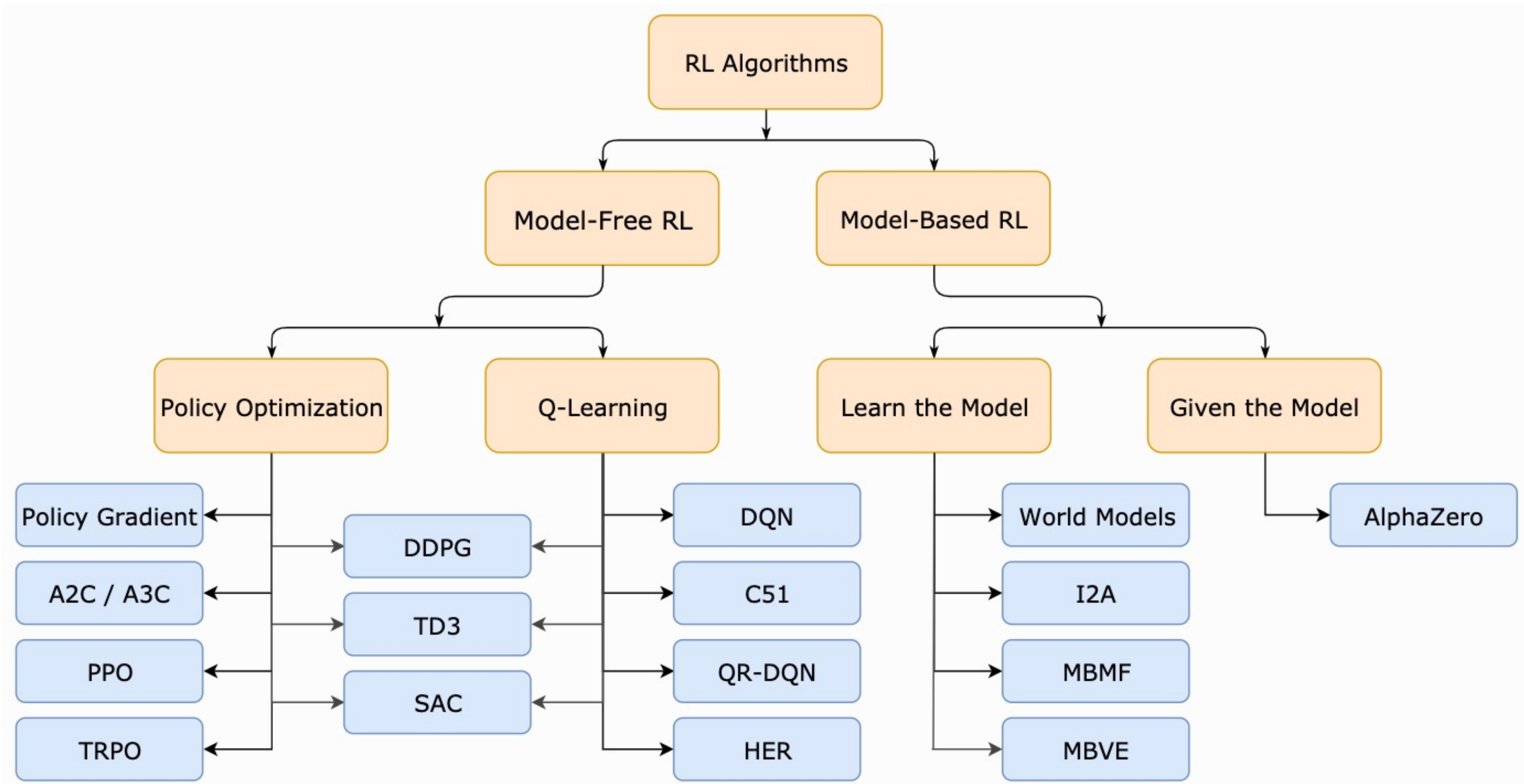
Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement Learning (RL) Taxonomy



Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement Learning (RL)
A Taxonomy of RL Algorithms



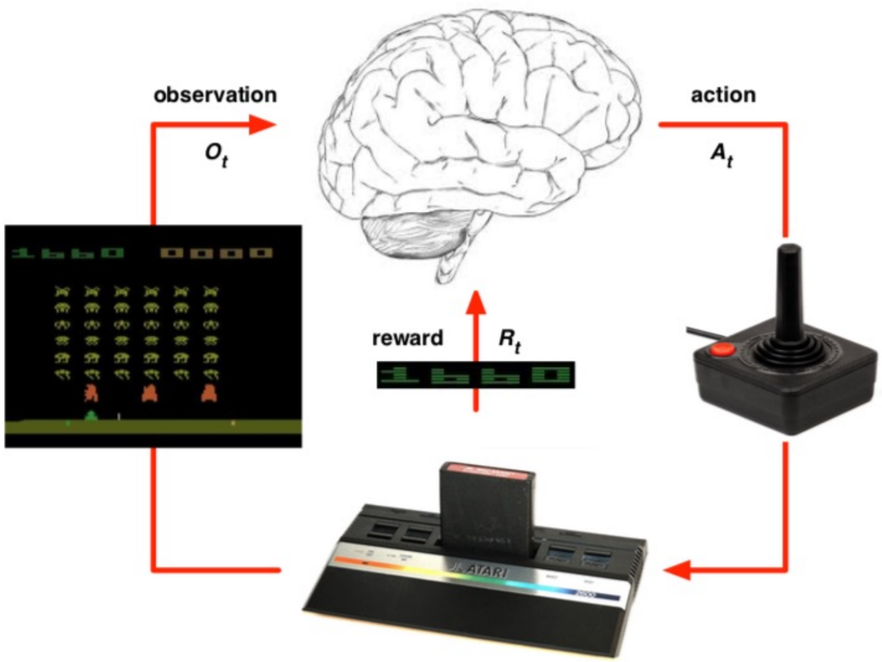
Source: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html

Learning and Planning

- Two fundamental problems in sequential decision making
  - Reinforcement Learning
    - The environment is initially unknown
    - The agent interacts with environment
    - The agent improves its policy
  - Planning
    - A model of the environment is known
    - The agent performs computations with its model (without any external interaction)
    - The agent improves its policy
    - a.k.a deliberation, reasoning, introspection, pondering, thought, search

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Atari Example: Reinforcement Learning

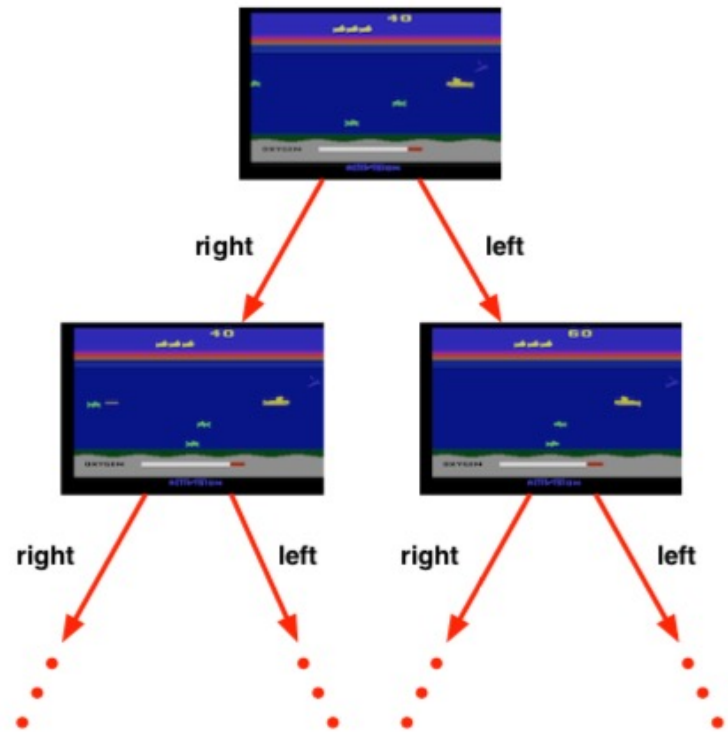


- Rules of the game are unknown
- Learn directly from interactive game-play
- Pick actions on joystick, see pixels and scores

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Atari Example: Planning

- Rules of the game are known
- Can query emulator
  - perfect model inside agent's brain
- If I take action a from state s:
  - what would the next state be?
  - what would the score be?
- Plan ahead to find optimal policy
  - e.g. tree search



Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Exploration and Exploitation

- Reinforcement learning is like trial-and-error learning
- The agent should discover a good policy
- From its experiences of the environment
- Without losing too much reward along the way
- **Exploration** finds more information about the environment
- **Exploitation** exploits known information to maximise reward
- It is usually important to explore as well as exploit

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Exploration and Exploitation Examples

- Restaurant Selection
  - Exploitation: Go to your favorite restaurant
  - Exploration: Try a new restaurant
- Online Banner Advertisements
  - Exploitation: Show the most successful advert
  - Exploration: Show a different advert

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Exploration and Exploitation Examples

- Oil Drilling
  - Exploitation: Drill at the best known location
  - Exploration: Drill at a new location
- Game Playing
  - Exploitation: Play the move you believe is best
  - Exploration: Play an experimental move

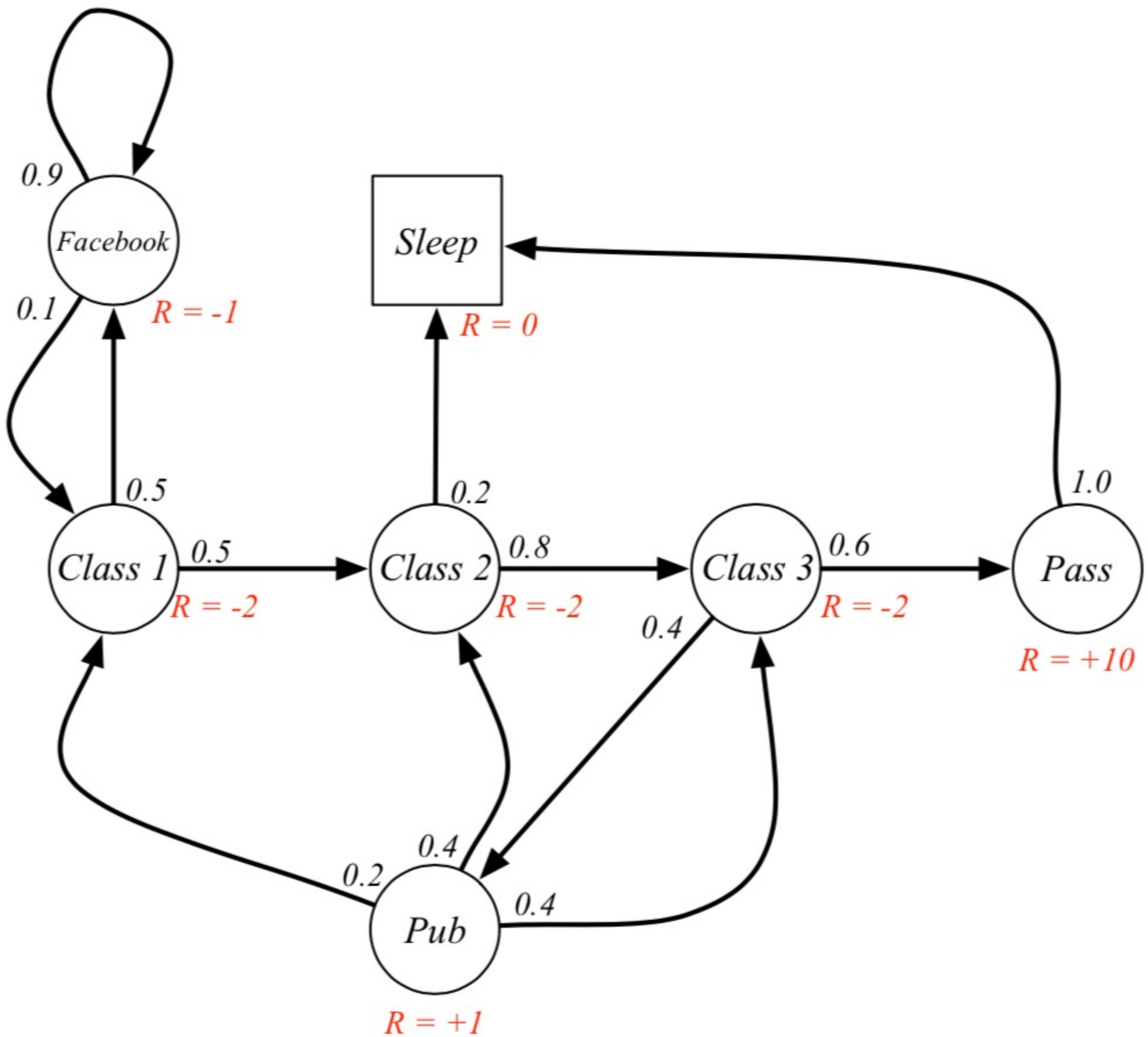
Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Prediction and Control

- Prediction: evaluate the future
  - Given a policy
- Control: optimize the future
  - Find the best policy

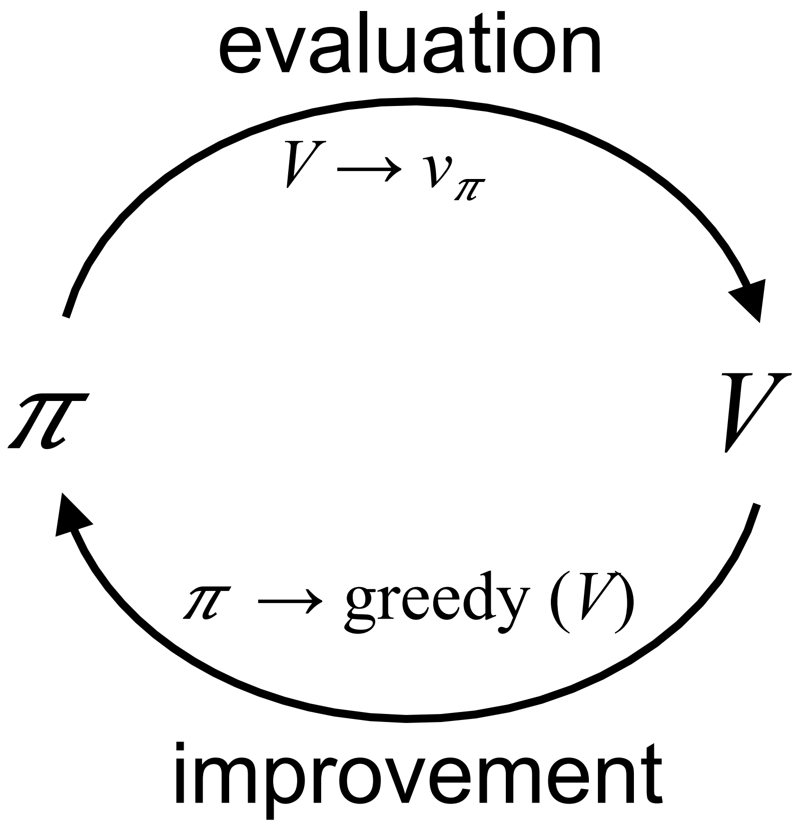
Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

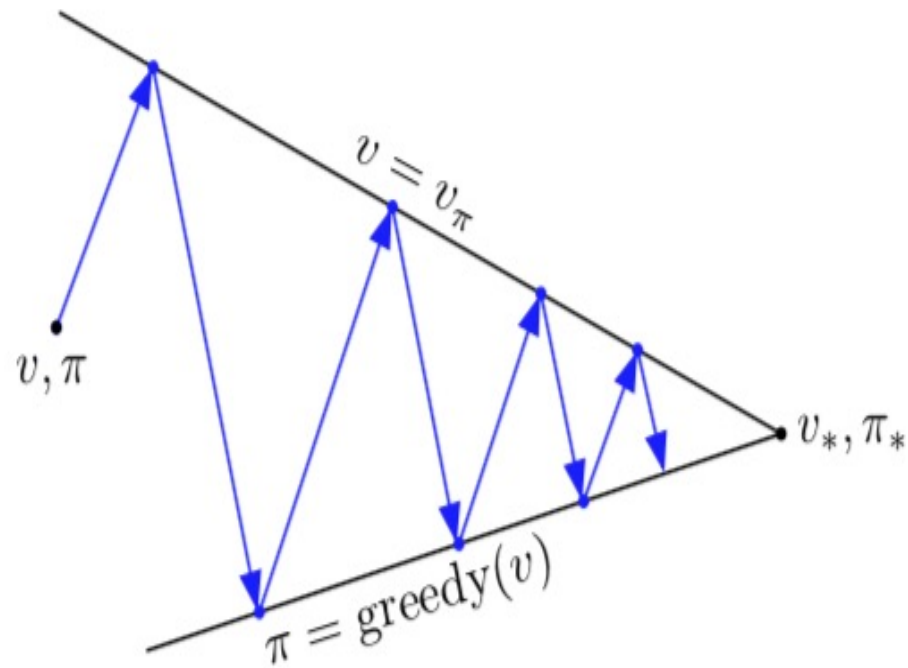
Markov Decision Processes (MDP)
Example: Student MDP



Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Generalized Policy Iteration (GPI)



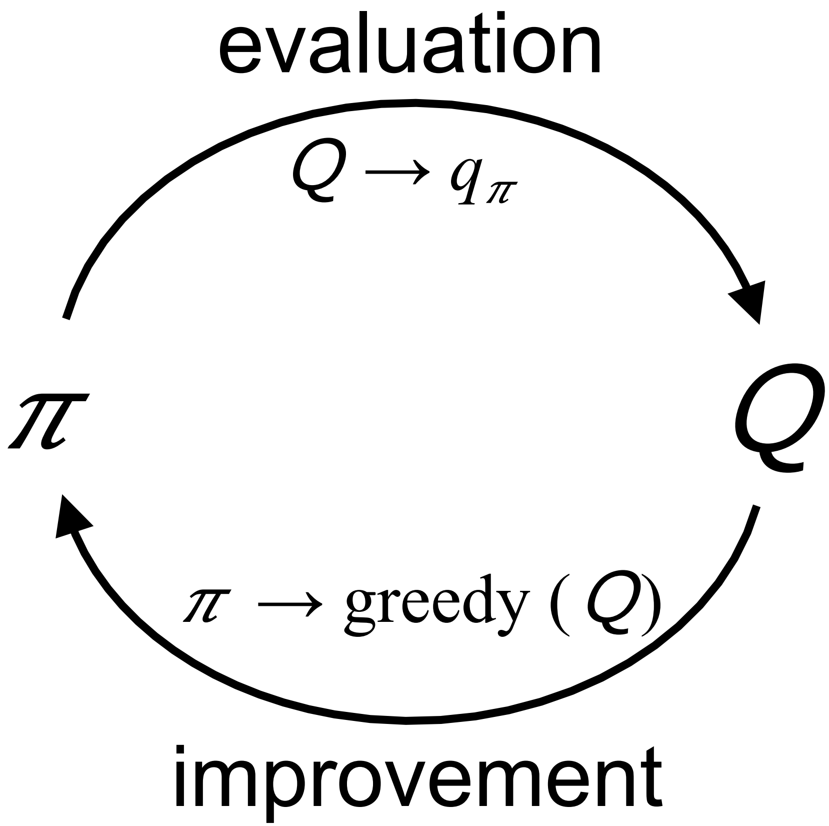


$$\pi_* \rightleftarrows v_*$$

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Generalized Policy Iteration (GPI)

Any iteration of **policy evaluation** and **policy improvement**, independent of their granularity.



Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

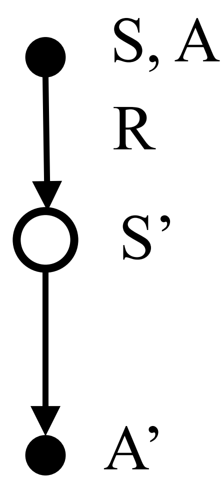
Temporal-Difference (TD) Learning

- Sarsa: On-policy TD Control
- Q-learning: Off-policy TD Control

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

SARSA
(state-action-reward-state-action)
On-policy TD Control

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



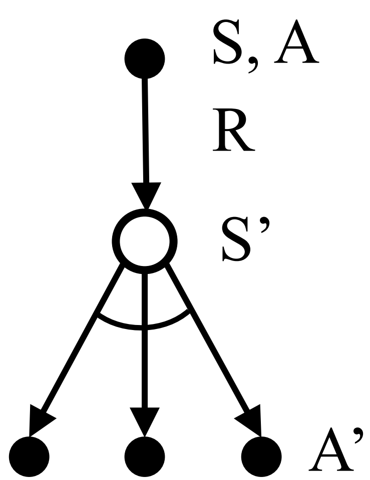
SARSA

Q-learning

(Watkins, 1989)

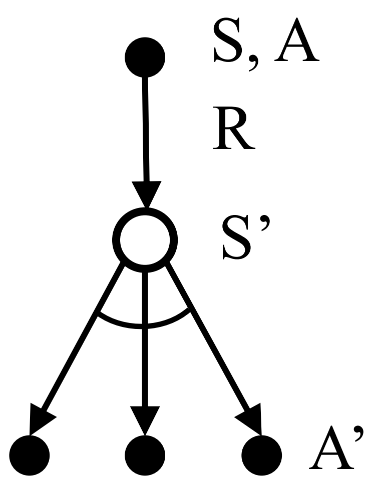
Off-policy TD Control

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

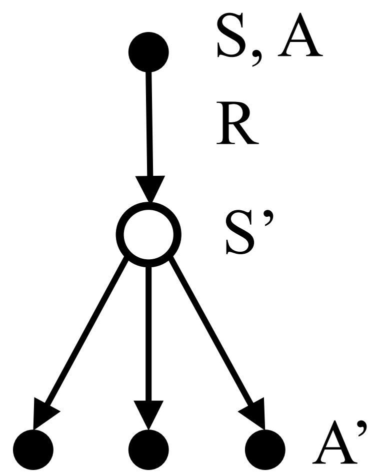


Q-learning

Q-learning and Expected SARSA



Q-learning



Expected SARSA

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Q-learning and Double Q-learning

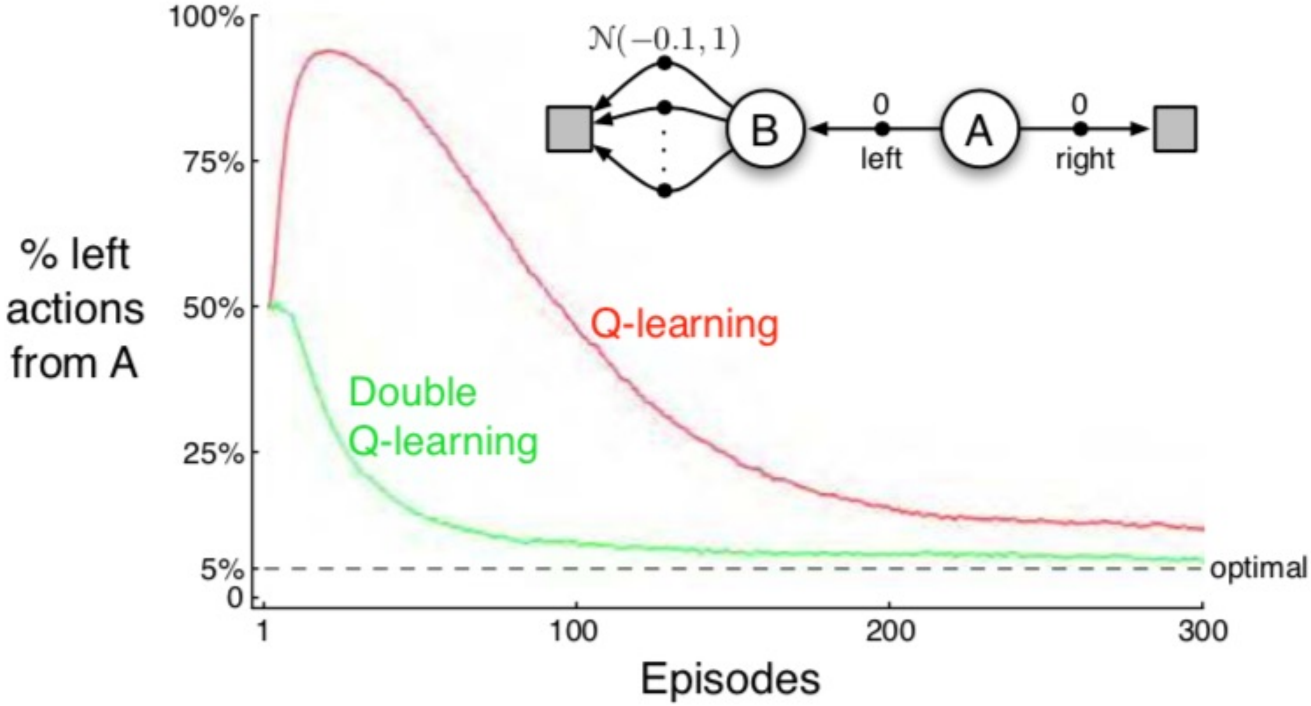


Figure 6.5: Comparison of Q-learning and Double Q-learning on a simple episodic MDP (shown inset). Q-learning initially learns to take the left action much more often than the right action, and always takes it significantly more often than the 5% minimum probability enforced by ε-greedy action selection with $\varepsilon = 0.1$. In contrast, Double Q-learning is essentially unaffected by maximization bias. These data are averaged over 10,000 runs. The initial action-value estimates were zero. Any ties in ε-greedy action selection were broken randomly.

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

n-step methods for sate-action value

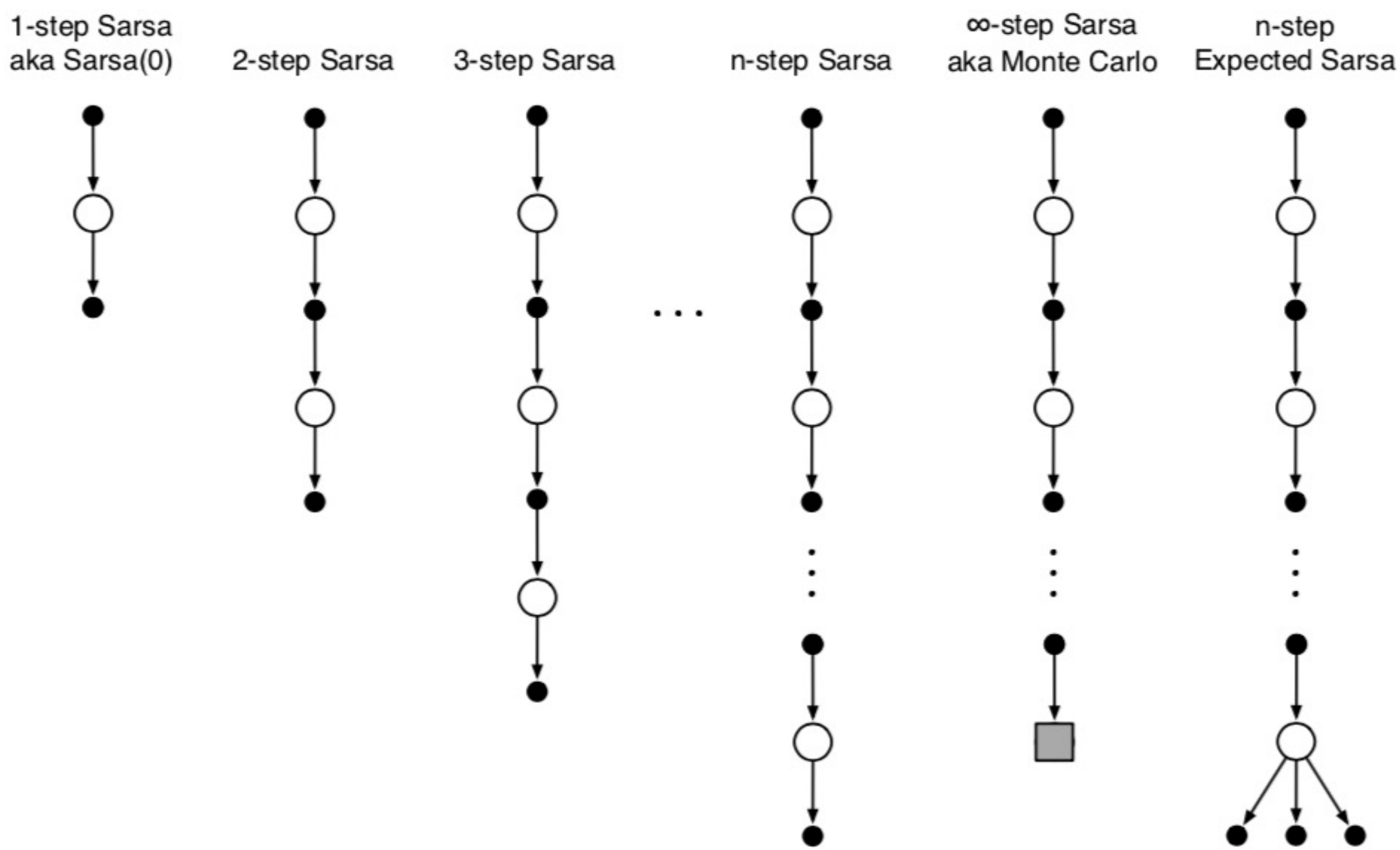
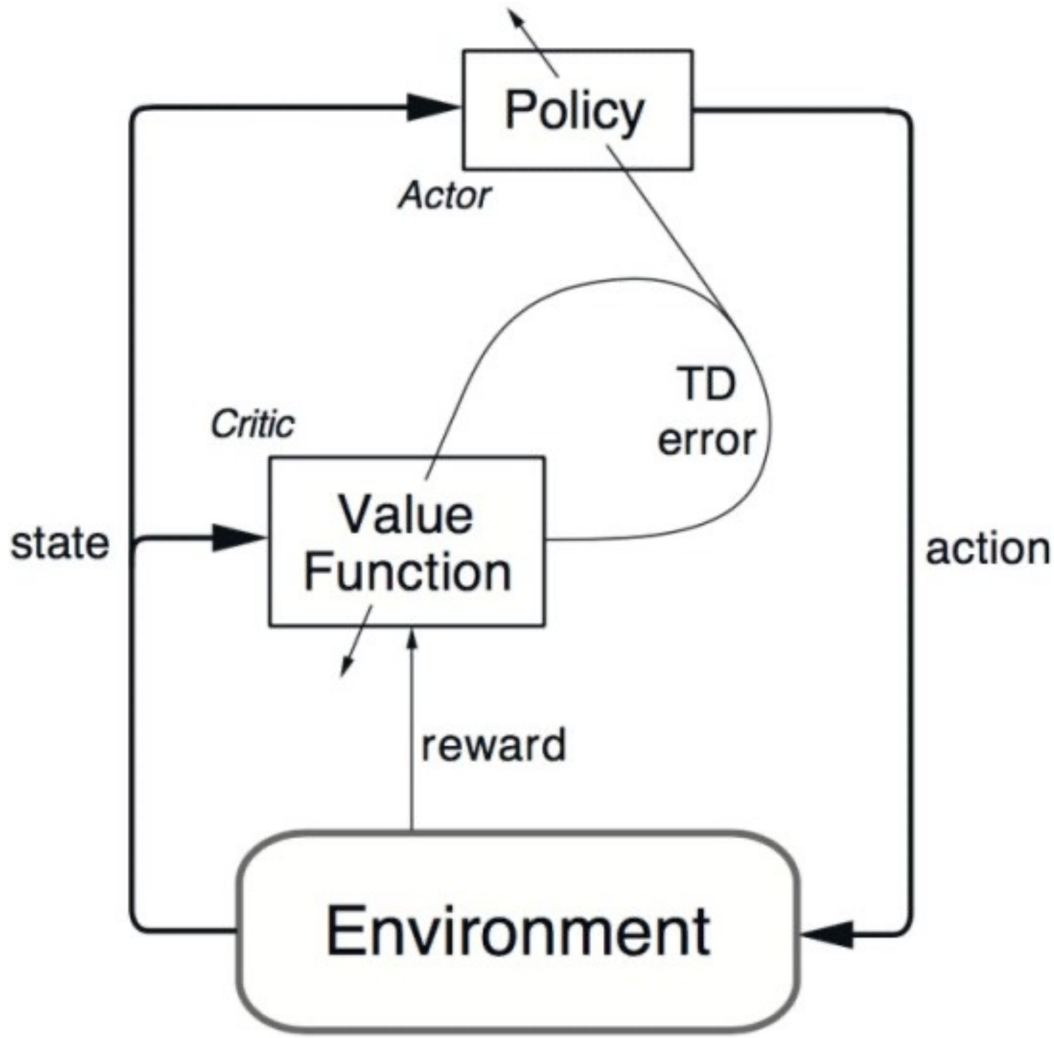


Figure 7.3: The backup diagrams for the spectrum of n-step methods for state–action values. They range from the one-step update of Sarsa(0) to the up-until-termination update of the Monte Carlo method. In between are the n-step updates, based on n steps of real rewards and the estimated value of the nth next state–action pair, all appropriately discounted. On the far right is the backup diagram for n-step Expected Sarsa.

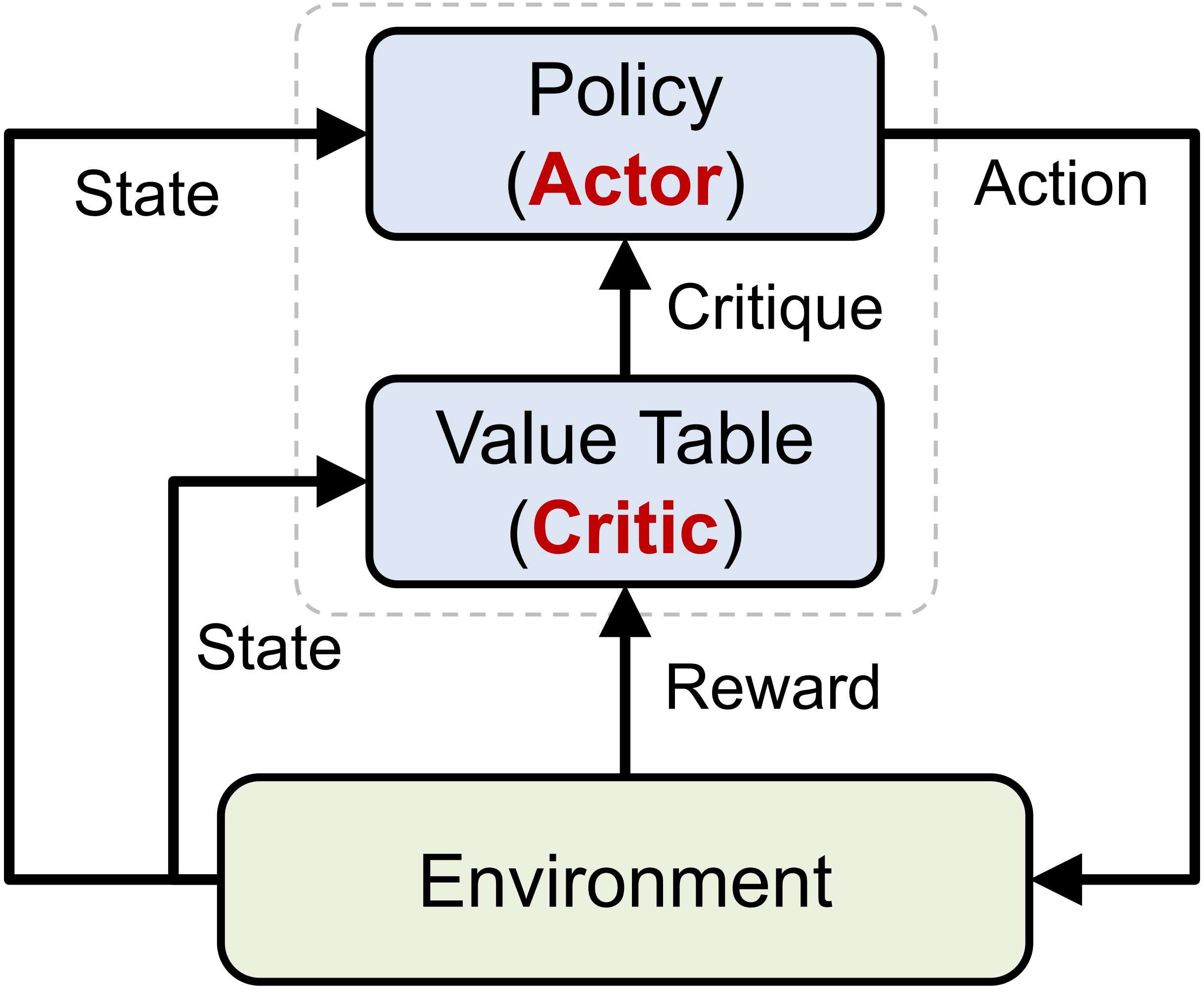
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning
Actor-Critic (AC) Architecture



Source: https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html

Reinforcement Learning
Actor-Critic (AC) Learning Methods



Reinforcement Learning Methods

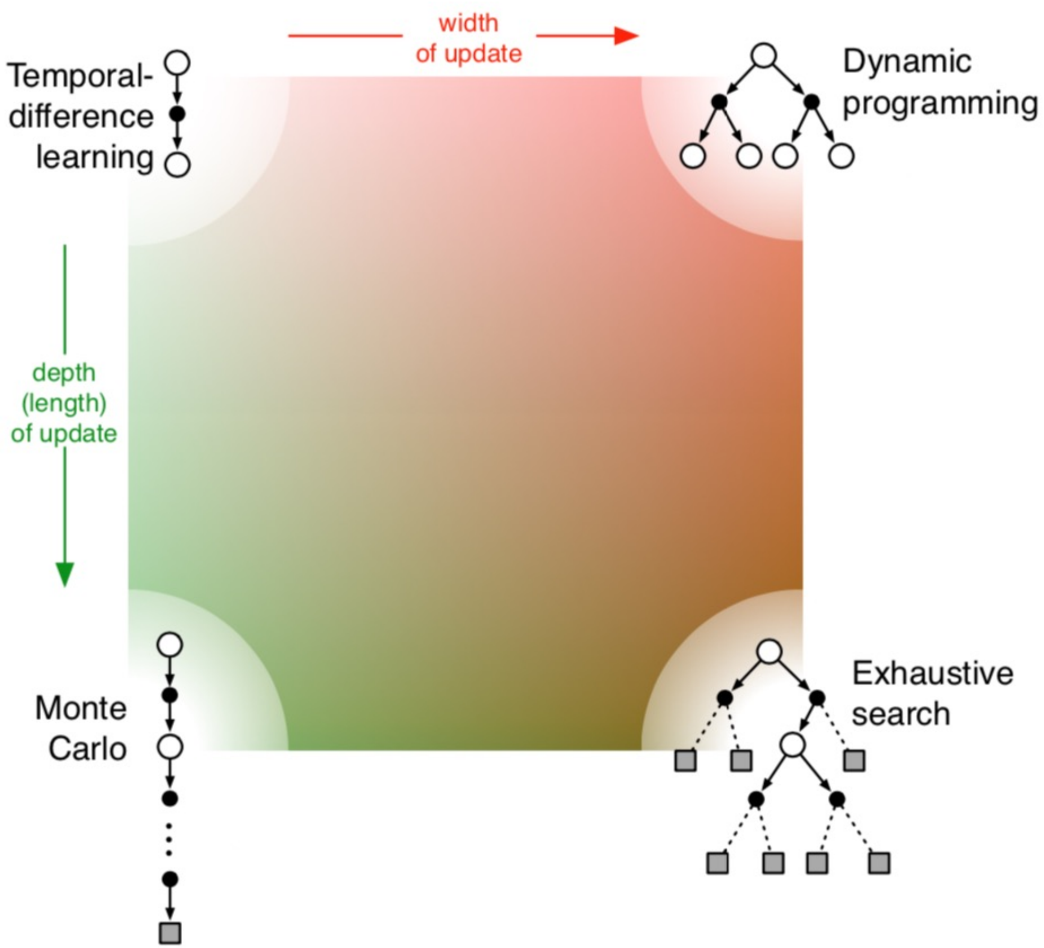


Figure 8.11: A slice through the space of reinforcement learning methods, highlighting the two of the most important dimensions explored in Part I of this book: the depth and width of the updates.

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Monte Carlo Tree Search (MCTS)

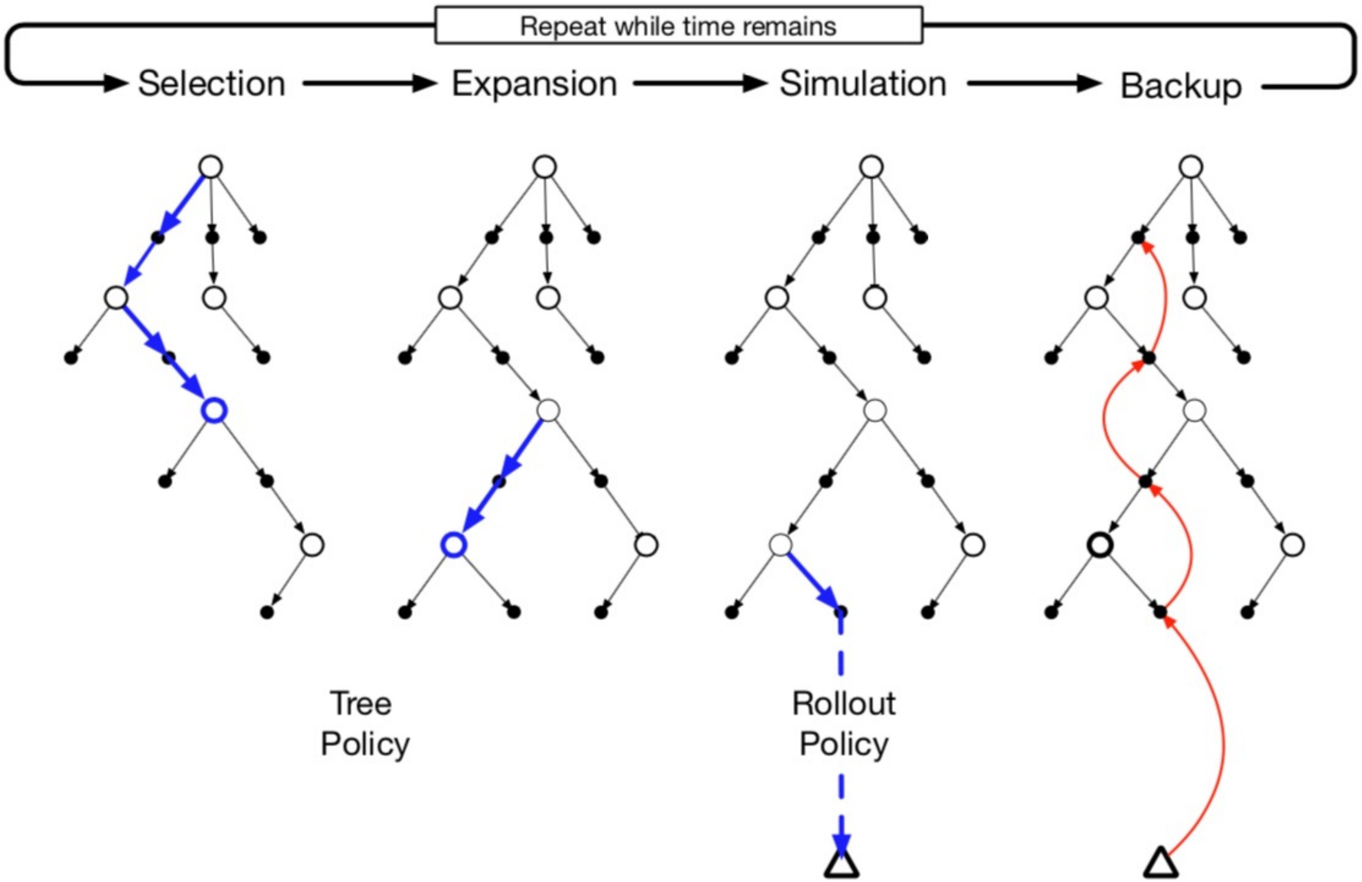
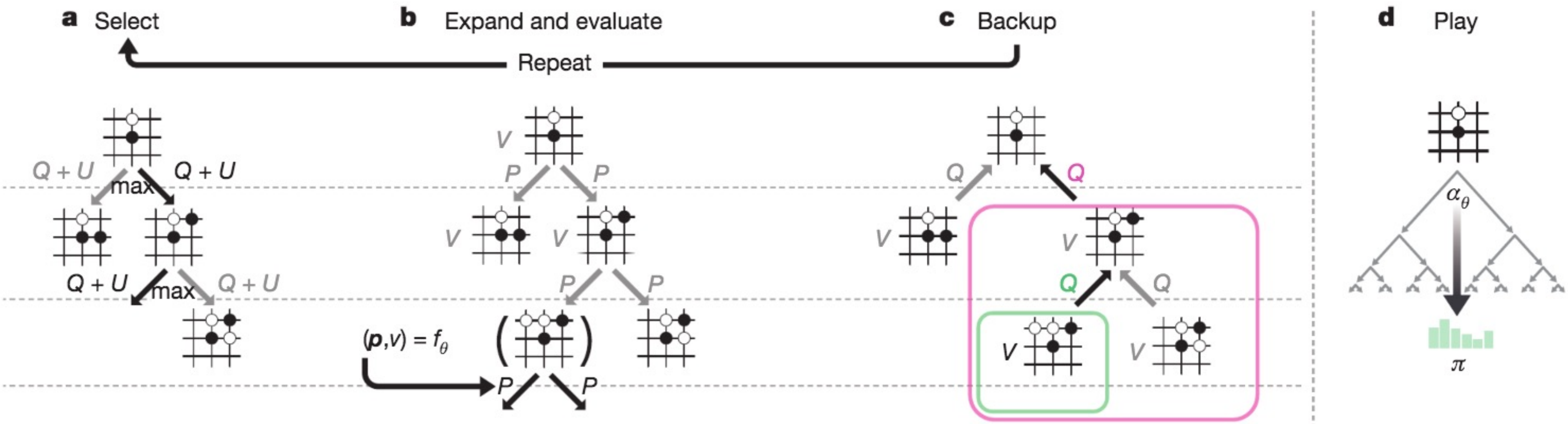


Figure 8.10: Monte Carlo Tree Search. When the environment changes to a new state, MCTS executes as many iterations as possible before an action needs to be selected, incrementally building a tree whose root node represents the current state. Each iteration consists of the four operations **Selection**, **Expansion** (though possibly skipped on some iterations), **Simulation**, and **Backup**, as explained in the text and illustrated by the bold arrows in the trees. Adapted from Chaslot, Bakkes, Szita, and Spronck (2008).

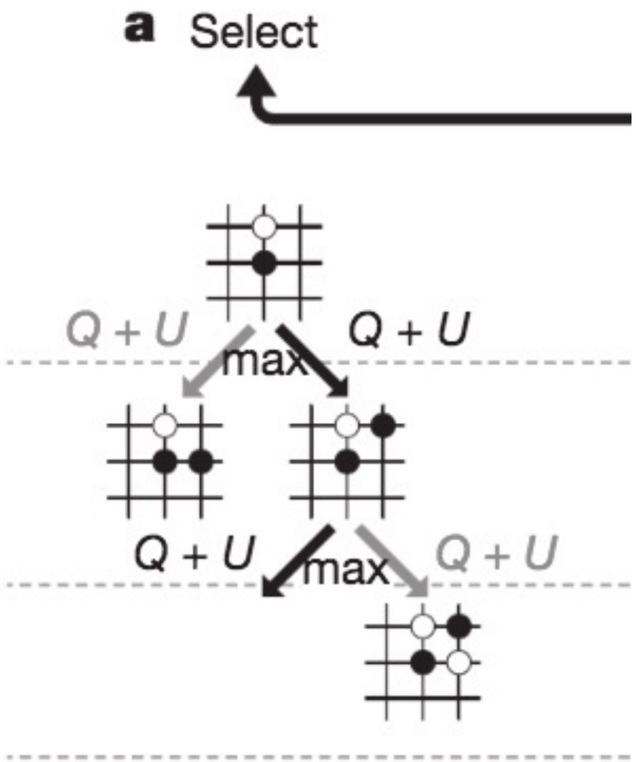
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Monte Carlo Tree Search (MCTS)
MCTS in AlphaGo Zero



Source: David Silver et al. (2017), "Mastering the game of Go without human knowledge." Nature 550 (2017): 354–359.

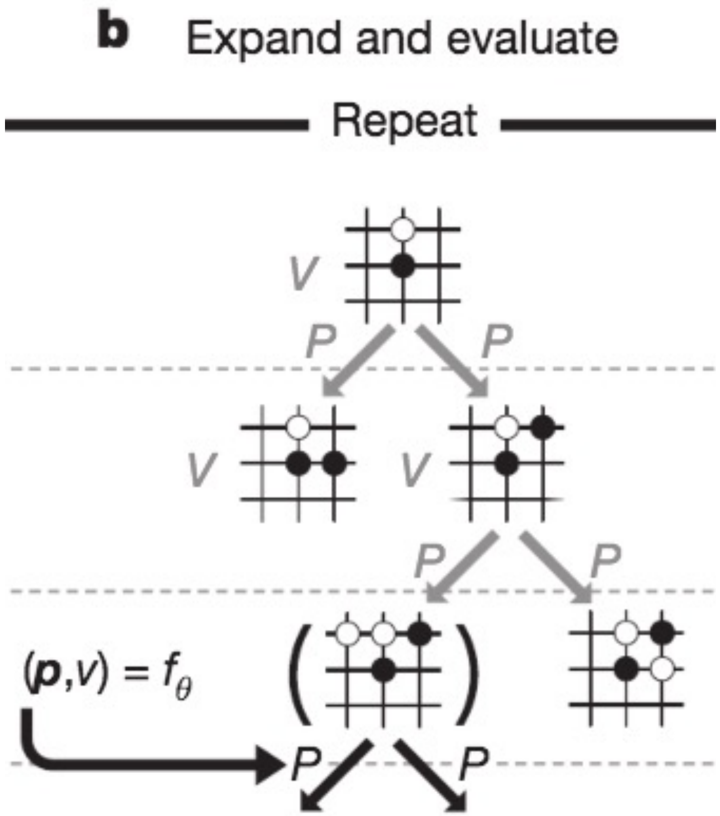
MCTS in AlphaGo Zero



a: Each simulation traverses the tree by selecting the edge with maximum action value Q, plus an upper confidence bound U that depends on a stored prior probability P and visit count N for that edge (which is incremented once traversed).

Source: David Silver et al. (2017), "Mastering the game of Go without human knowledge." Nature 550 (2017): 354–359.

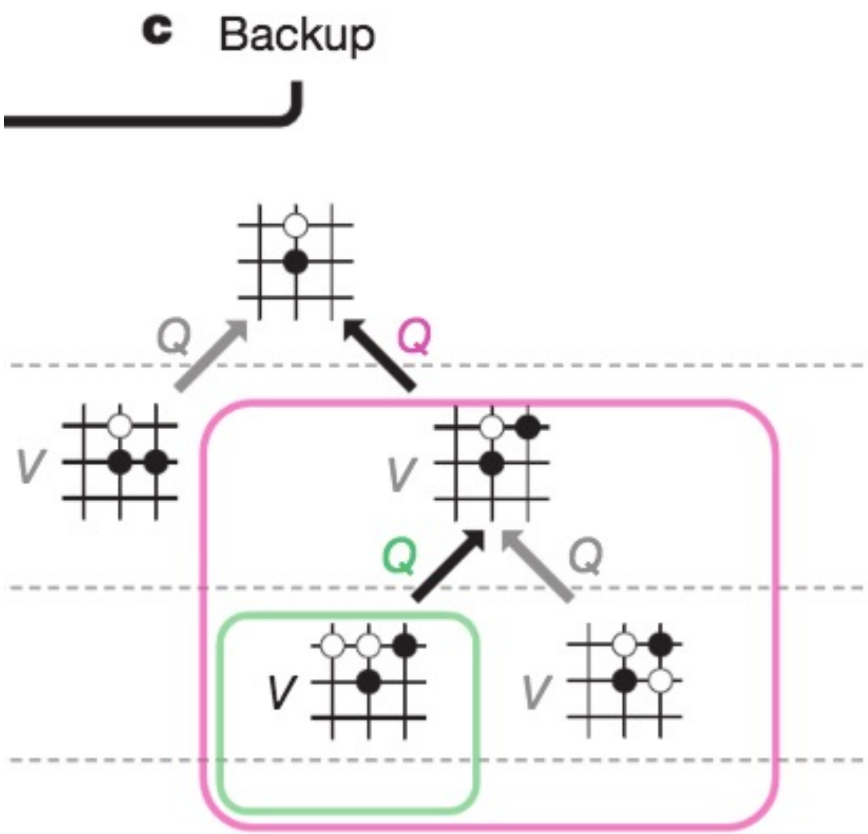
MCTS in AlphaGo Zero



b: The leaf node is expanded and the associated position s is evaluated by the neural network $(P(s, \cdot), V(s)) = f_\theta(s)$; the vector of P values are stored in the outgoing edges from s.

Source: David Silver et al. (2017), "Mastering the game of Go without human knowledge." Nature 550 (2017): 354–359.

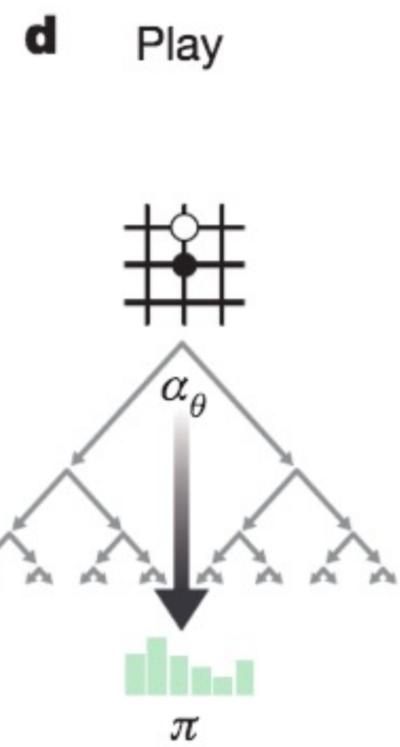
MCTS in AlphaGo Zero



c: Action value Q is updated to track the mean of all evaluations V in the subtree below that action

Source: David Silver et al. (2017), "Mastering the game of Go without human knowledge." Nature 550 (2017): 354–359.

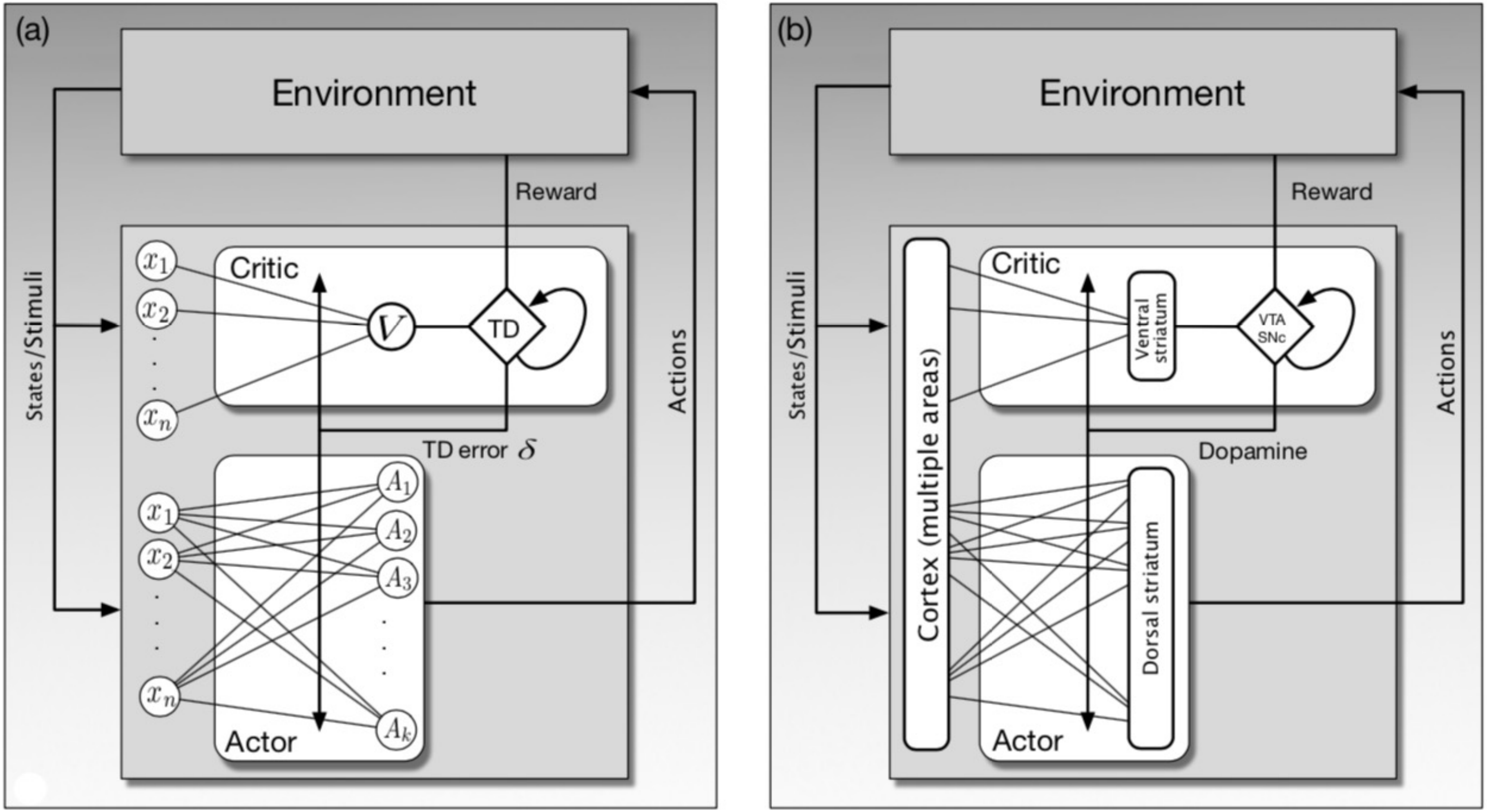
MCTS in AlphaGo Zero



d: Once the search is complete, search probabilities π are returned, proportional to $N^{1/\tau}$, where N is the visit count of each move from the root state and τ is a parameter controlling temperature.

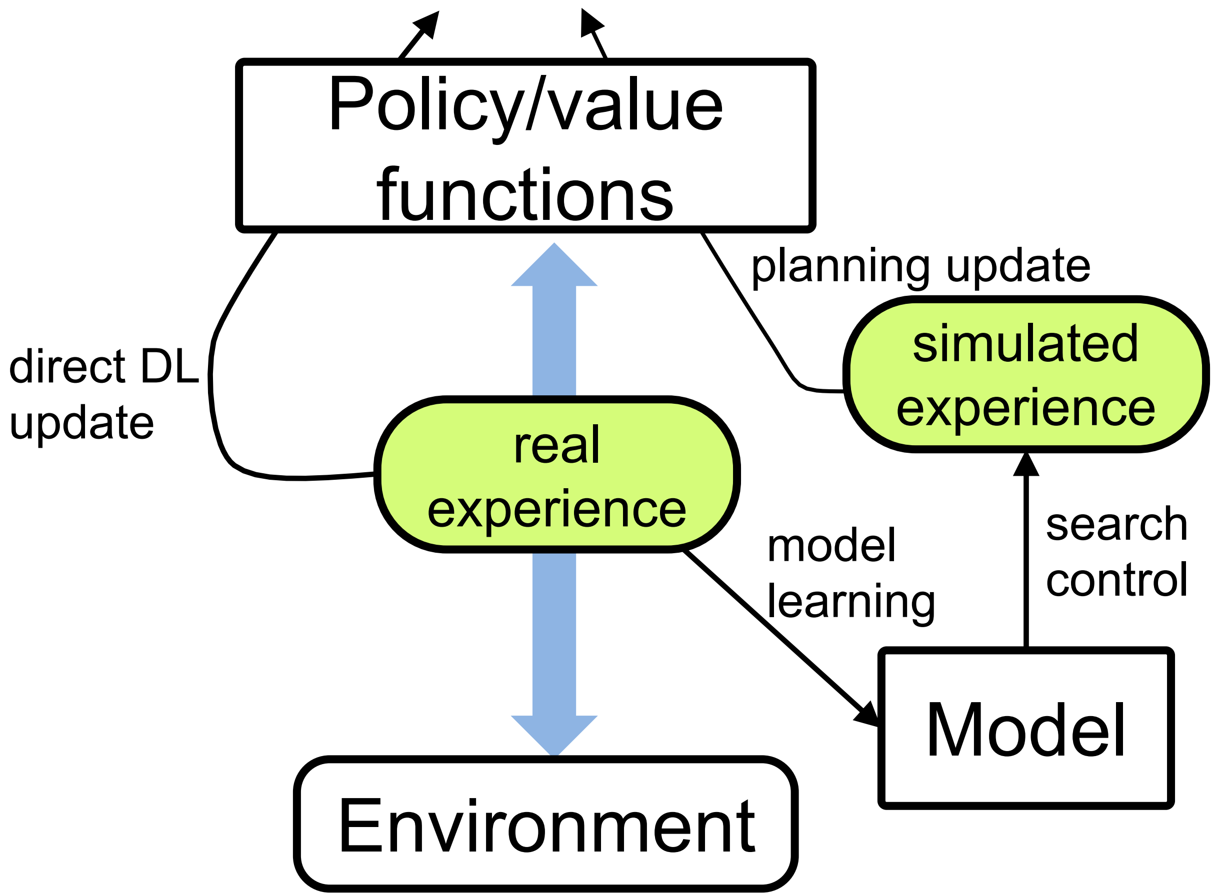
Source: David Silver et al. (2017), "Mastering the game of Go without human knowledge." Nature 550 (2017): 354–359.

Reinforcement Learning
Actor Critic ANN



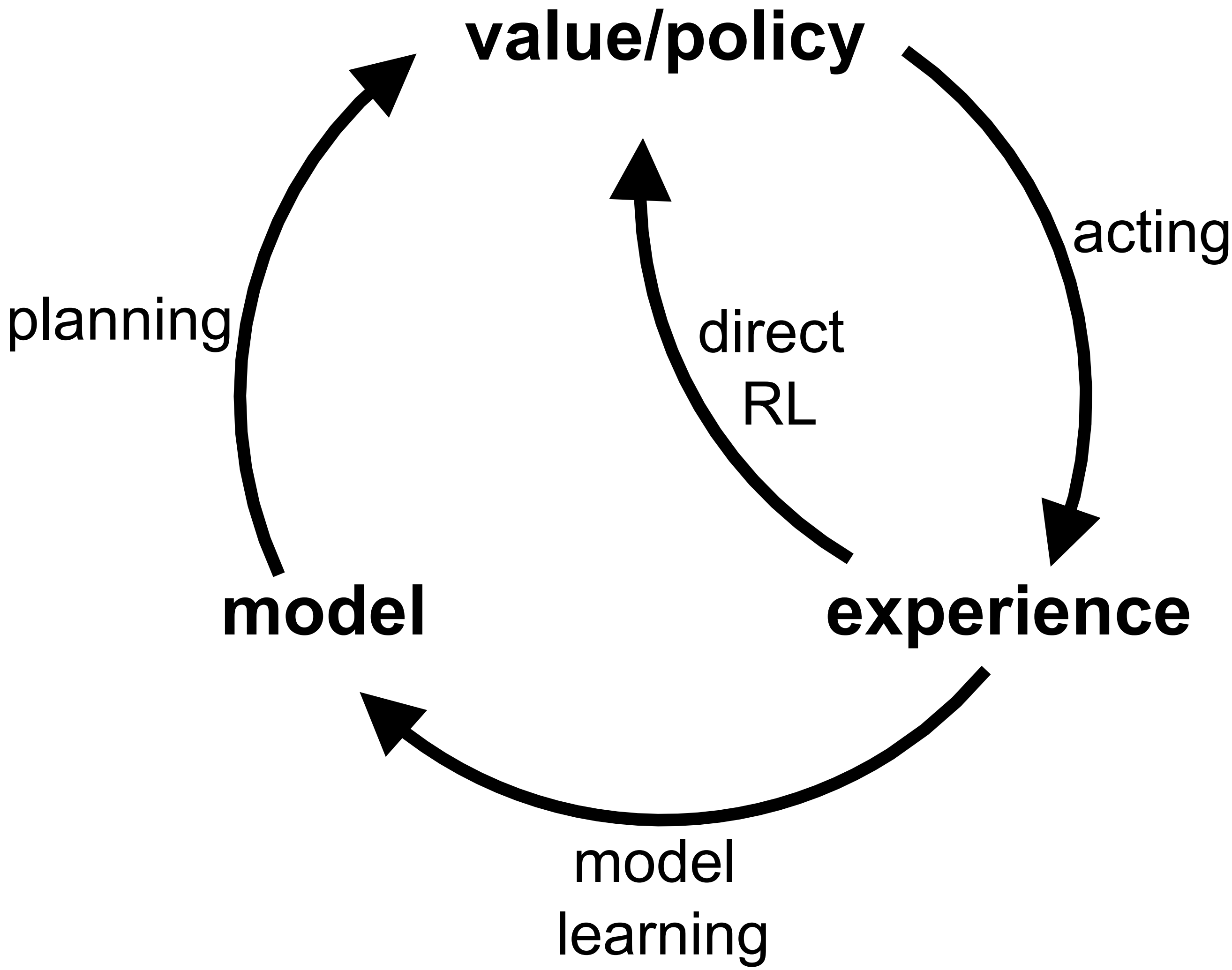
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning
General Dyna Architecture



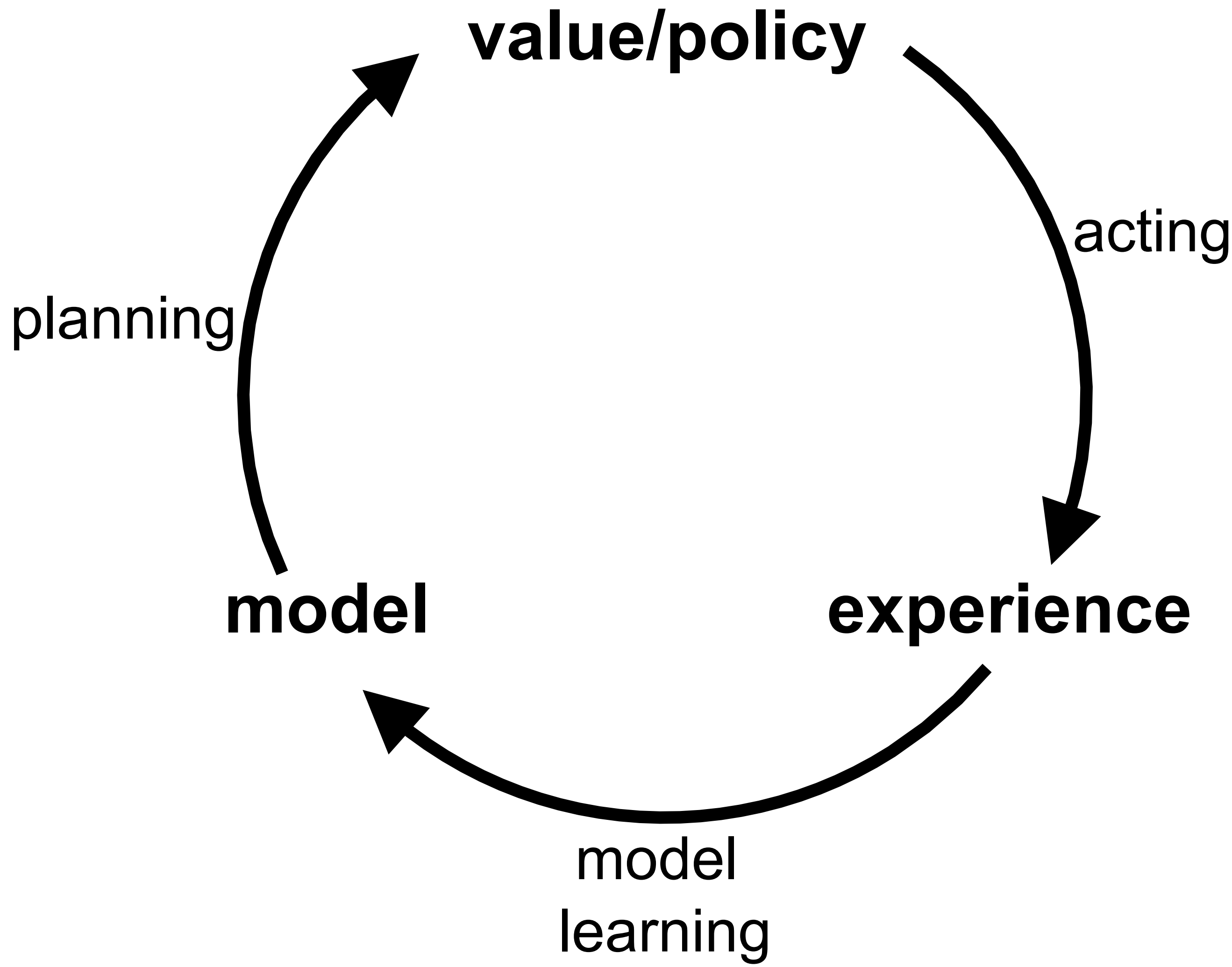
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Dyna:
Integrated Planning, Acting, and Learning



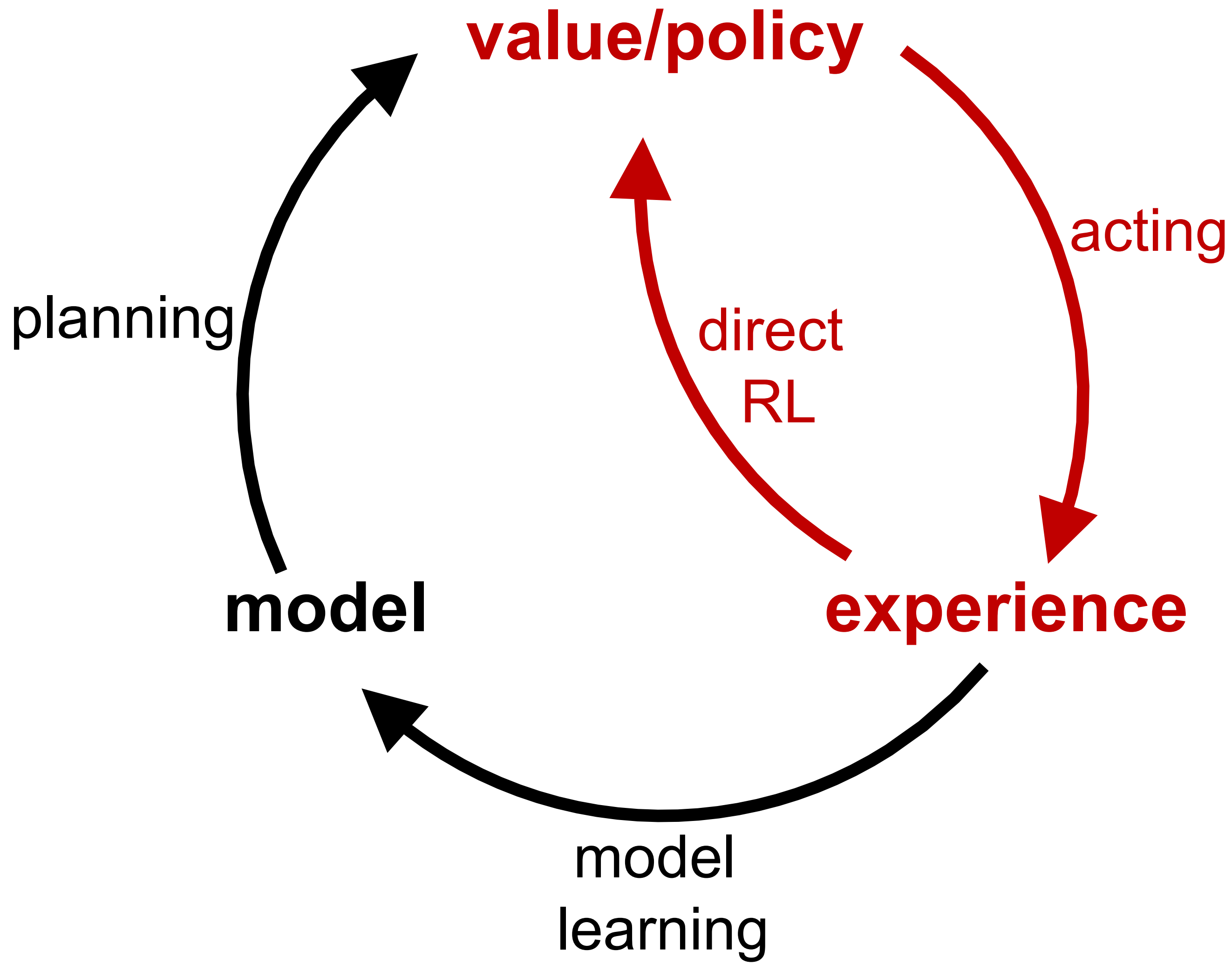
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Model-Based RL



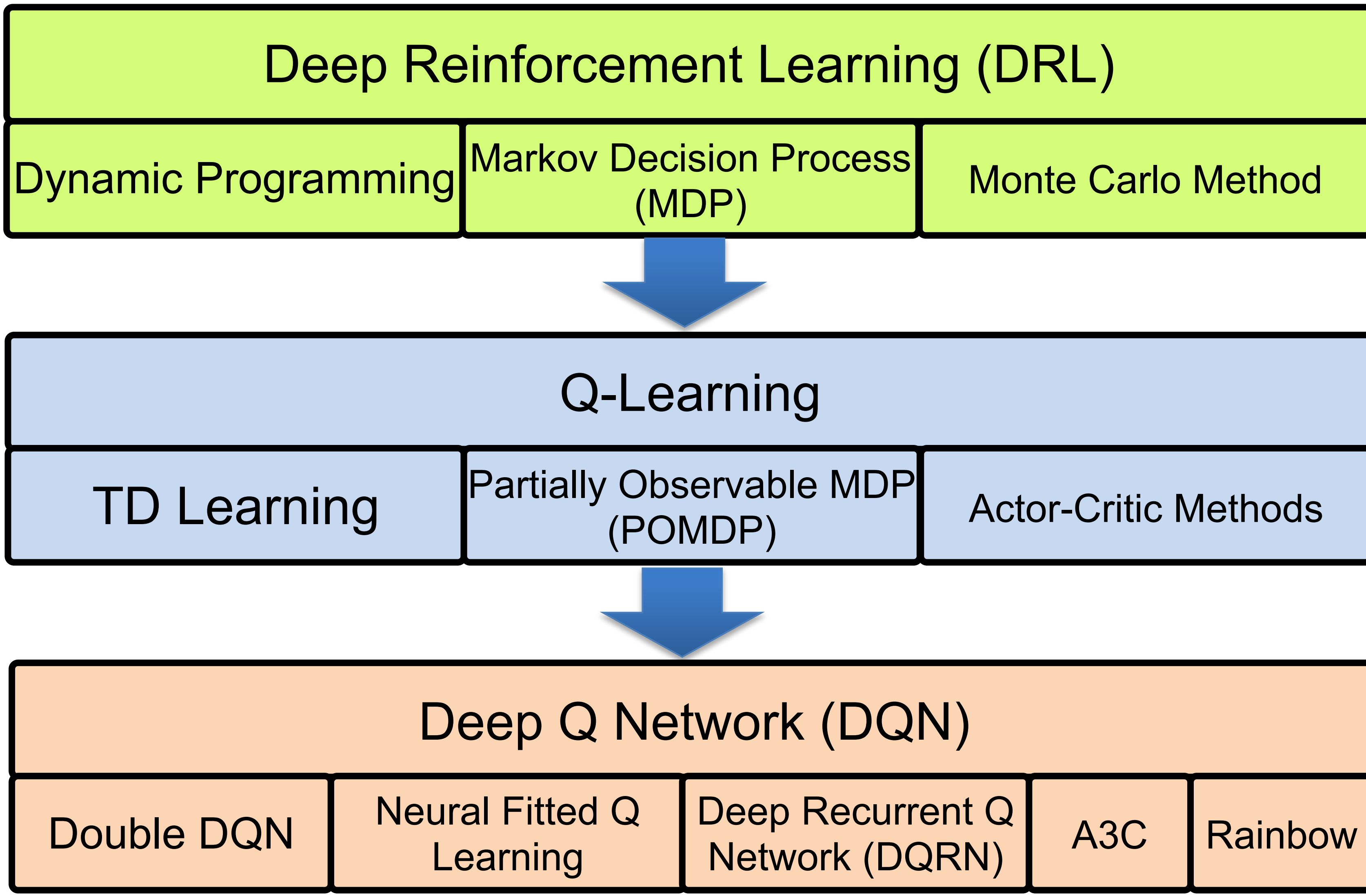
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Model-Free RL
(DQN, A3C)



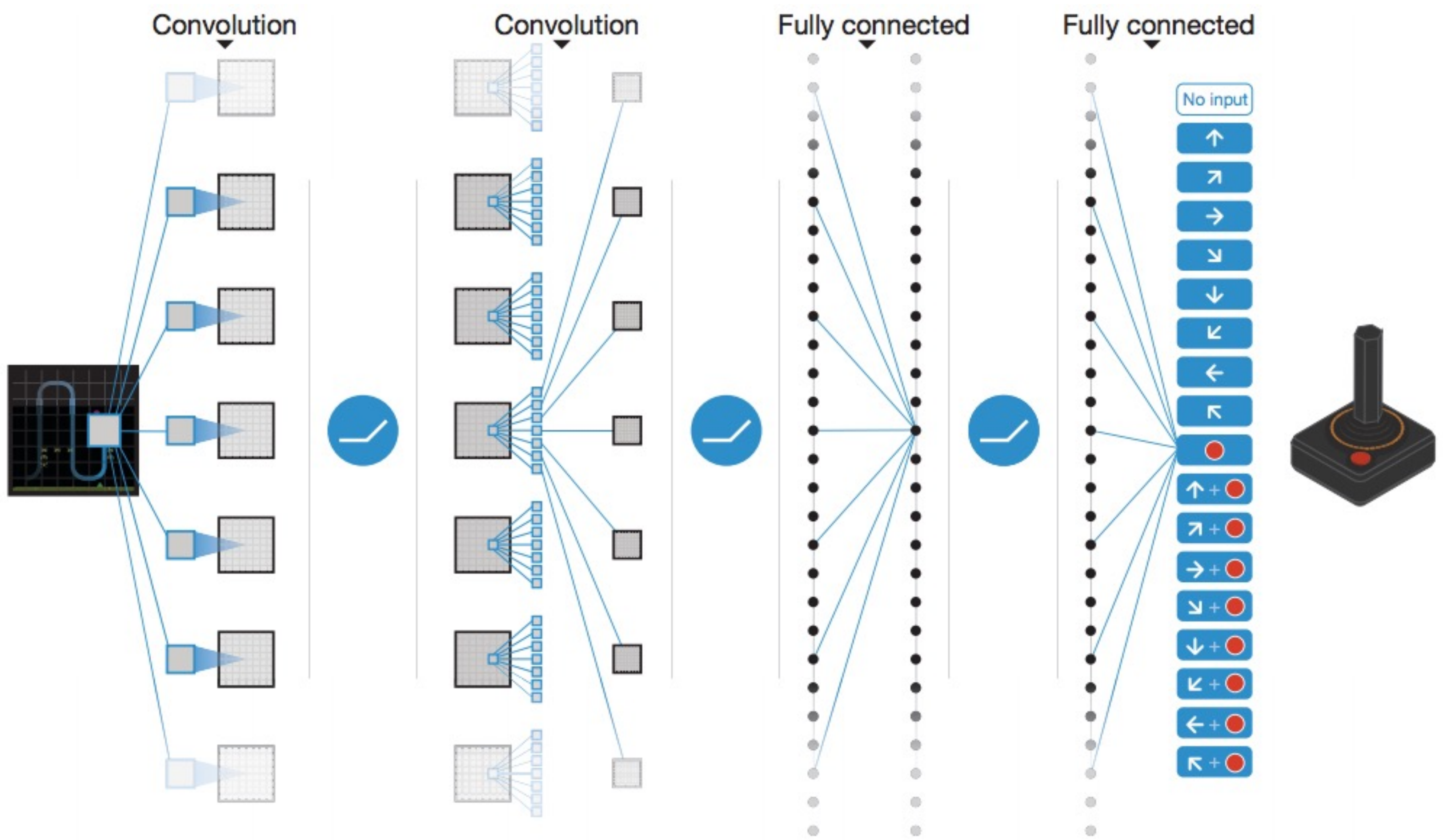
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning Algorithms



Source: https://amitray.com/deep-learning-past-present-and-future-a-review/

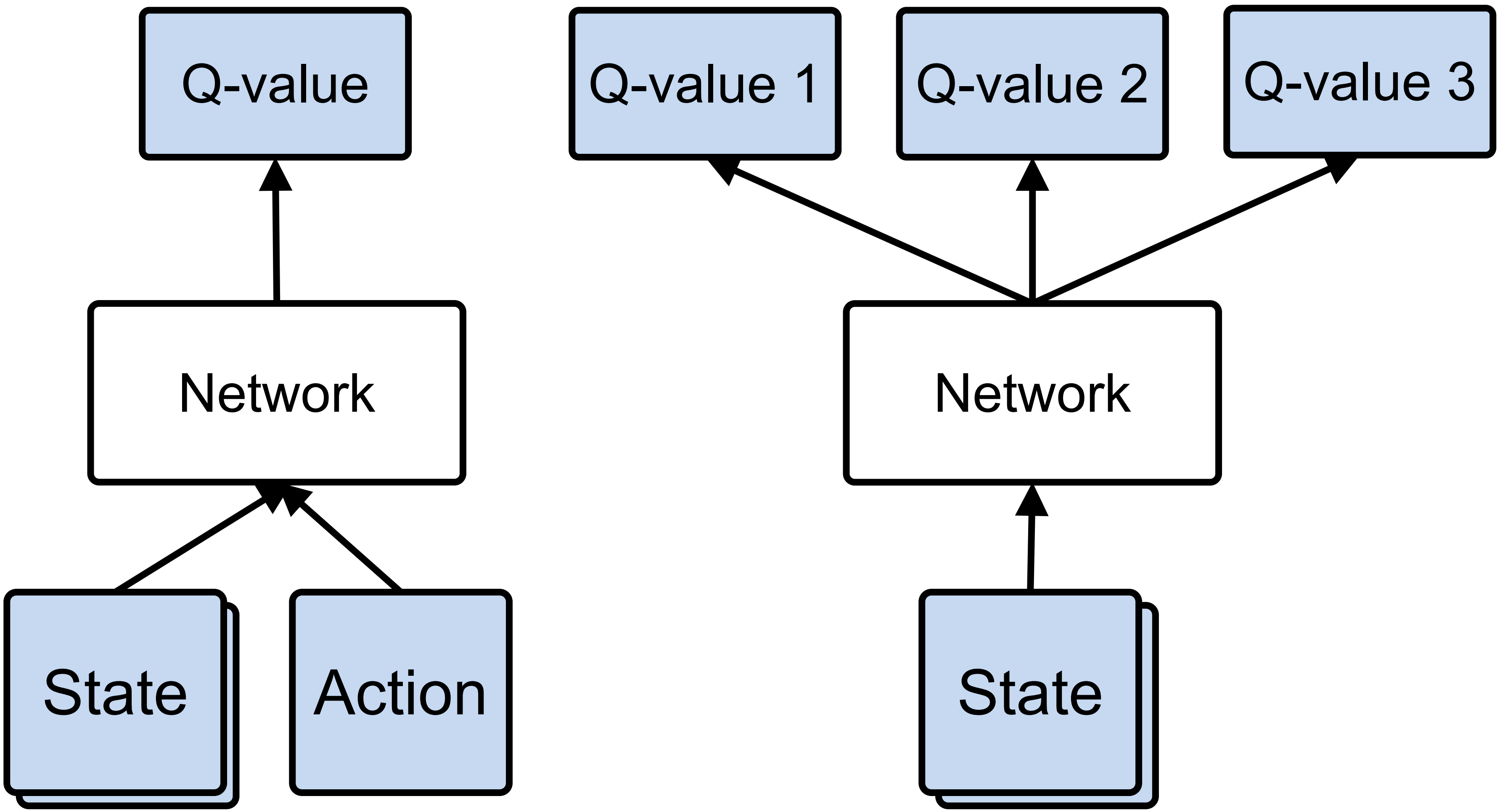
Human-level control through deep reinforcement learning (DQN)



Schematic illustration of the convolutional neural network

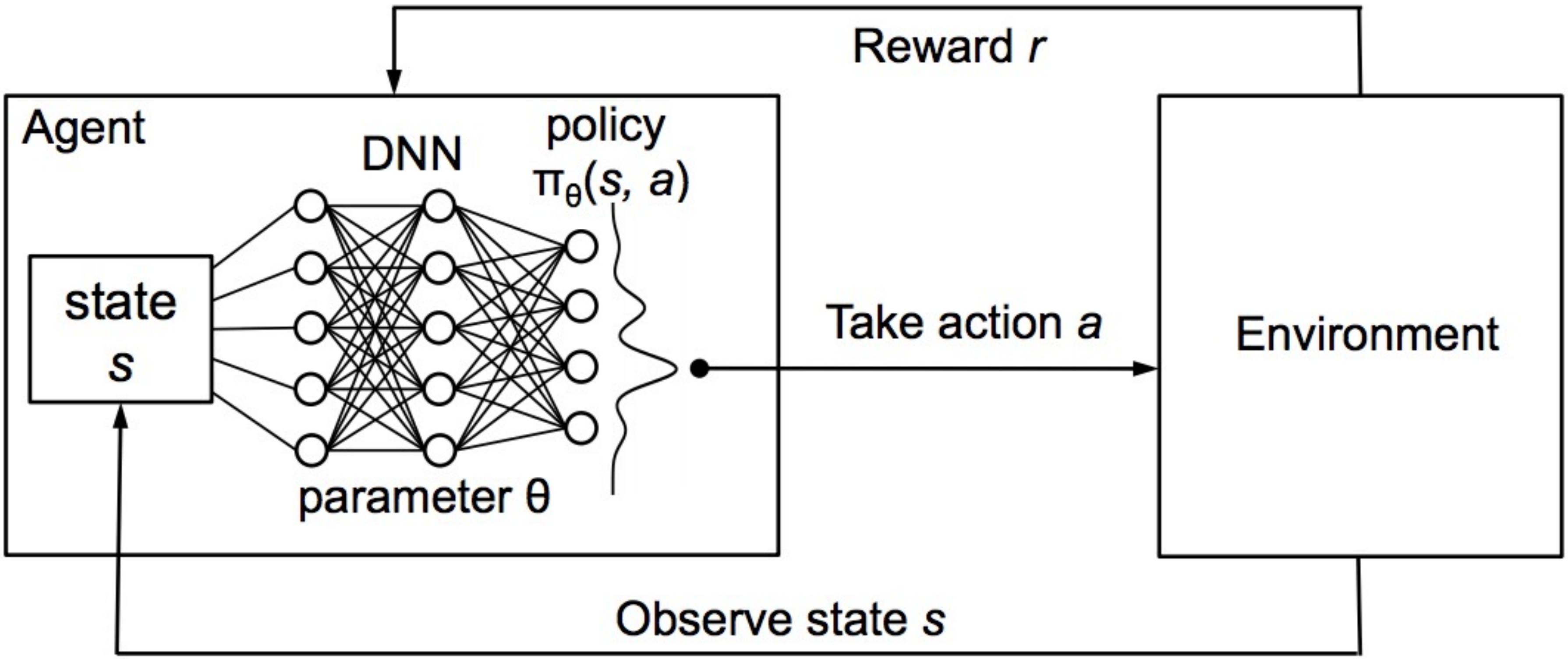
Source: Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves et al. "Human-level control through deep reinforcement learning." Nature 518, no. 7540 (2015): 529.

Deep Q-Network (DQN)



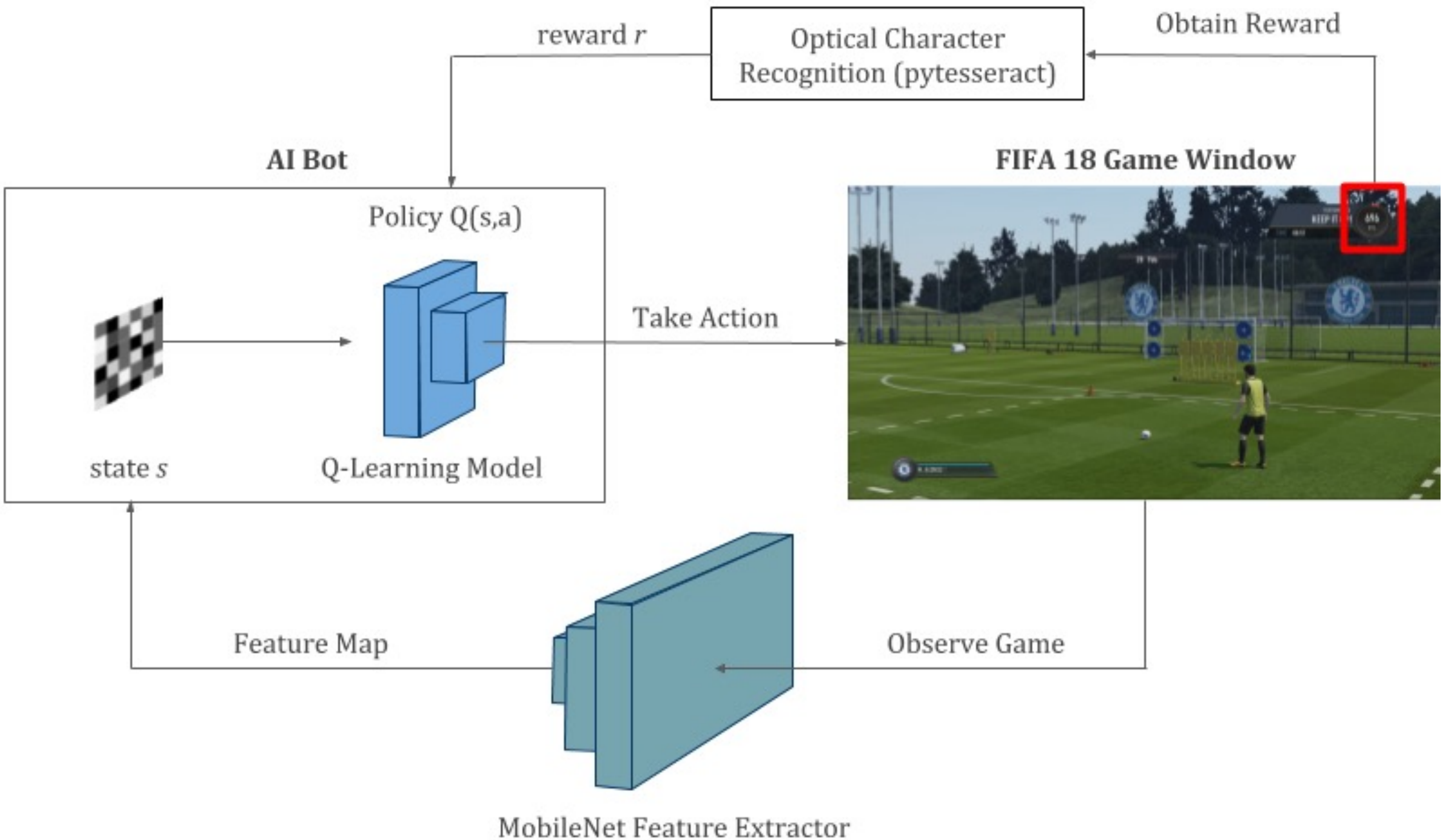
Source: https://neuro.cs.ut.ee/demystifying-deep-reinforcement-learning/

Reinforcement Learning with policy represented via DNN



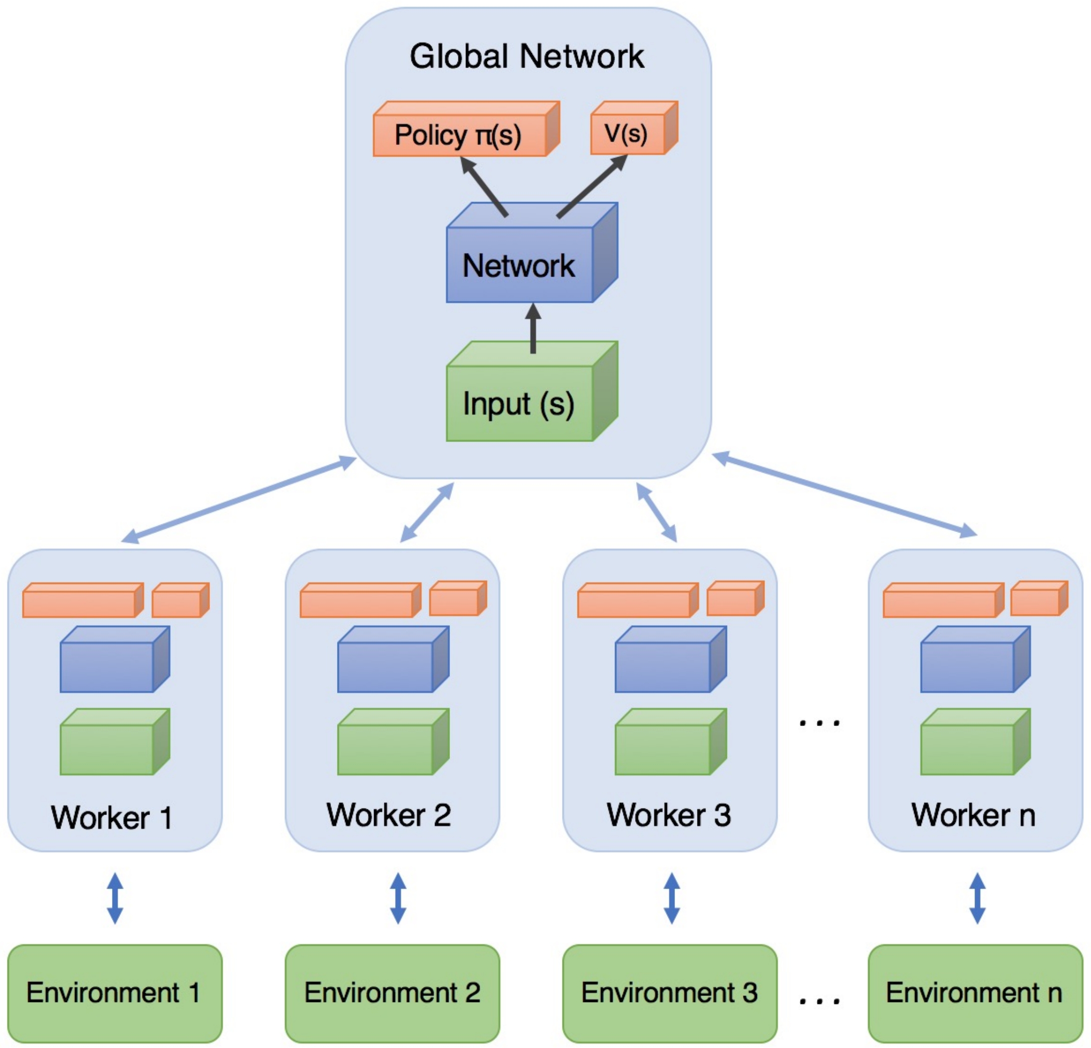
Source: Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. (2016) "Resource management with deep reinforcement learning." In Proceedings of the 15th ACM Workshop on Hot Topics in Networks, pp. 50-56. ACM, 2016.

Reinforcement Learning
Deep Q-Learning in FIFA 18



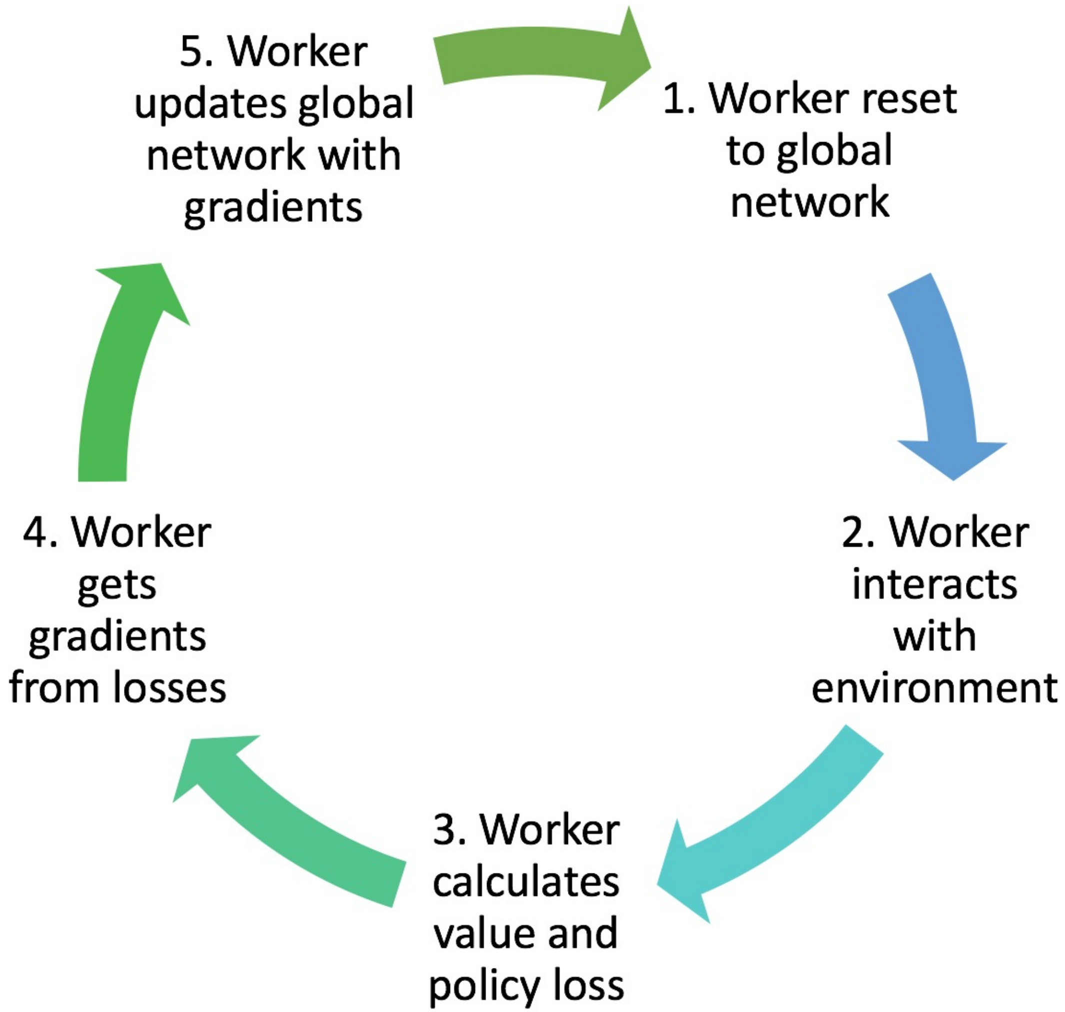
Source: https://towardsdatascience.com/using-deep-q-learning-in-fifa-18-to-perfect-the-art-of-free-kicks-f2e4e979ee66

Asynchronous Advantage Actor-Critic (A3C)



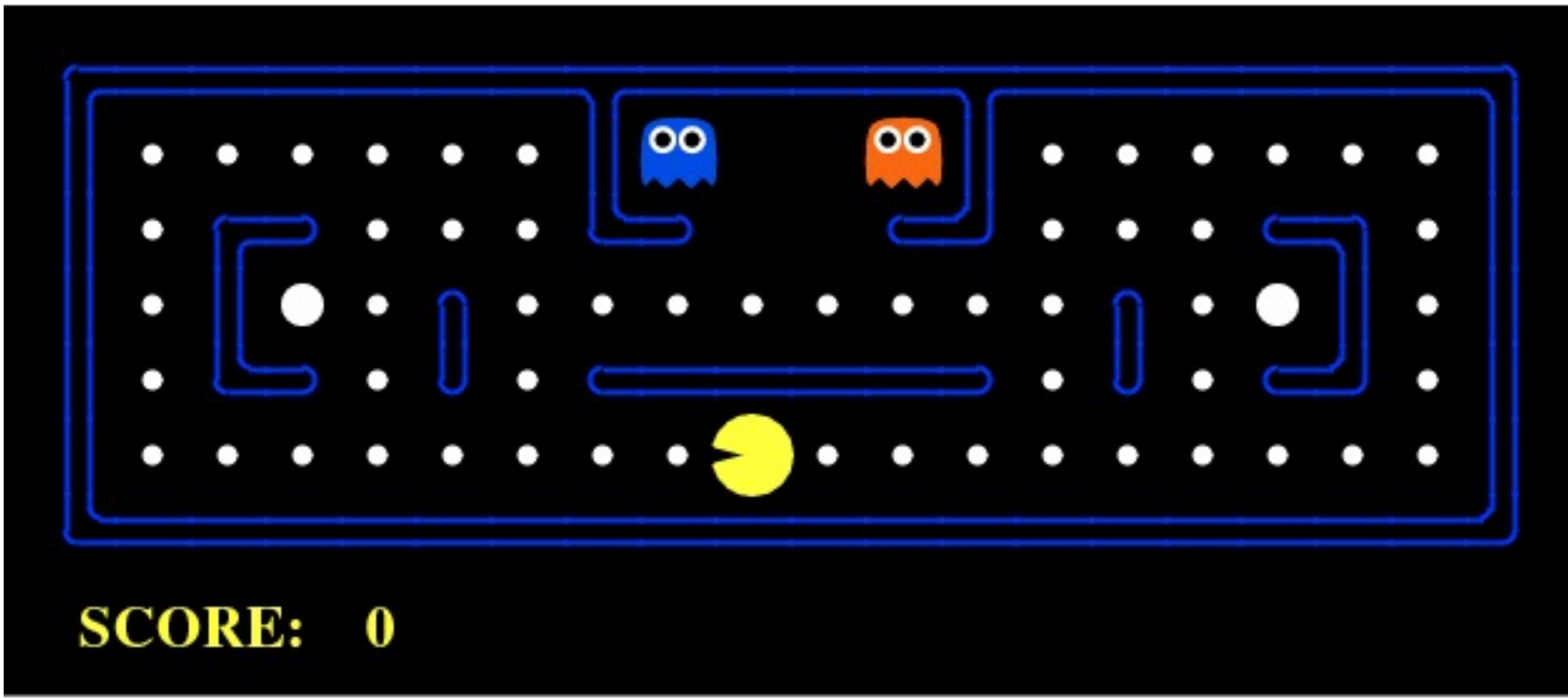
Source: https://medium.com/emergent-future/simple-reinforcement-learning-with-tensorflow-part-8-asynchronous-actor-critic-agents-a3c-c88f72a5e9f2

Training workflow of each worker agent in A3C



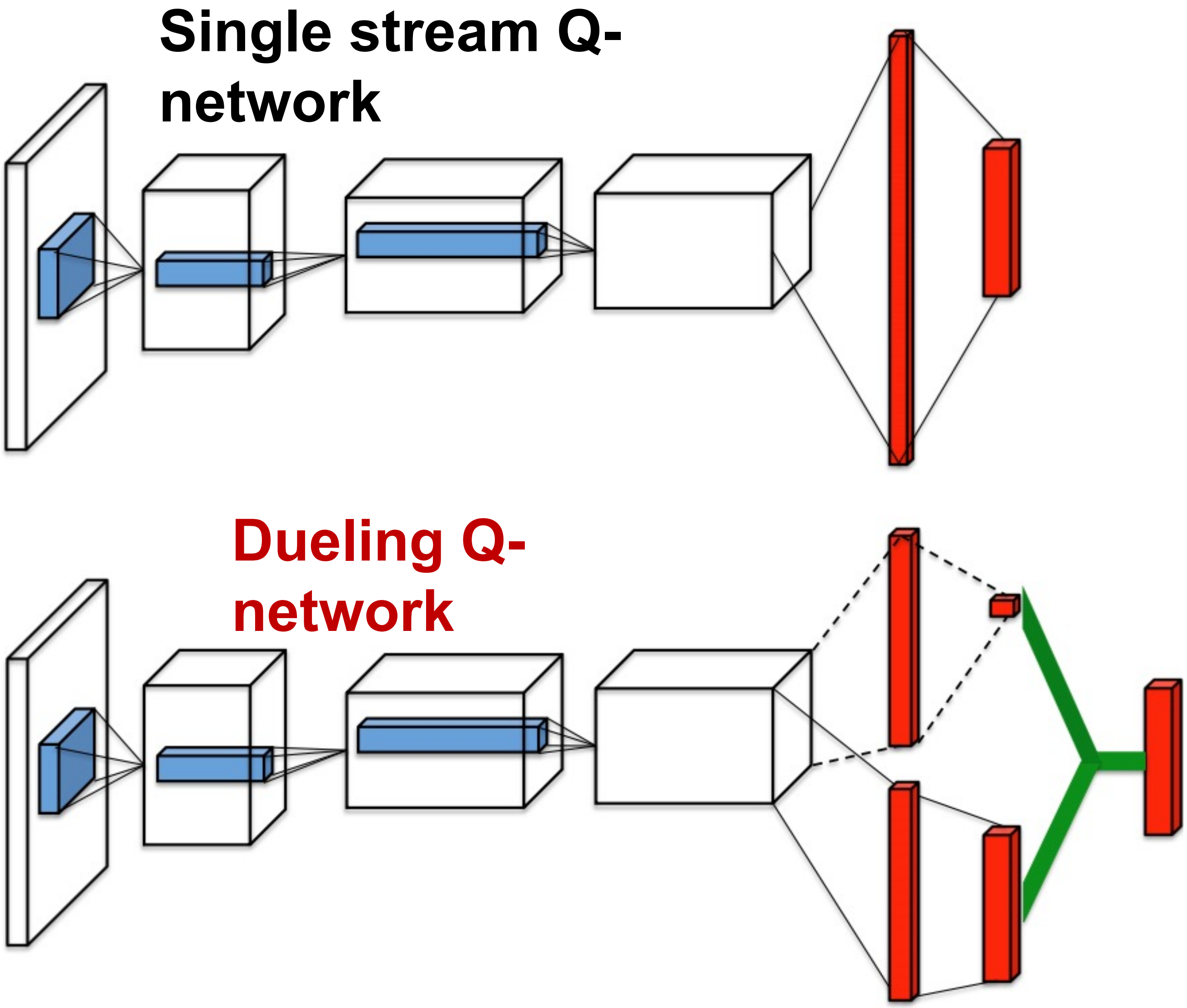
Source: https://medium.com/emergent-future/simple-reinforcement-learning-with-tensorflow-part-8-asynchronous-actor-critic-agents-a3c-c88f72a5e9f2

Reinforcement Learning Example: PCMAN



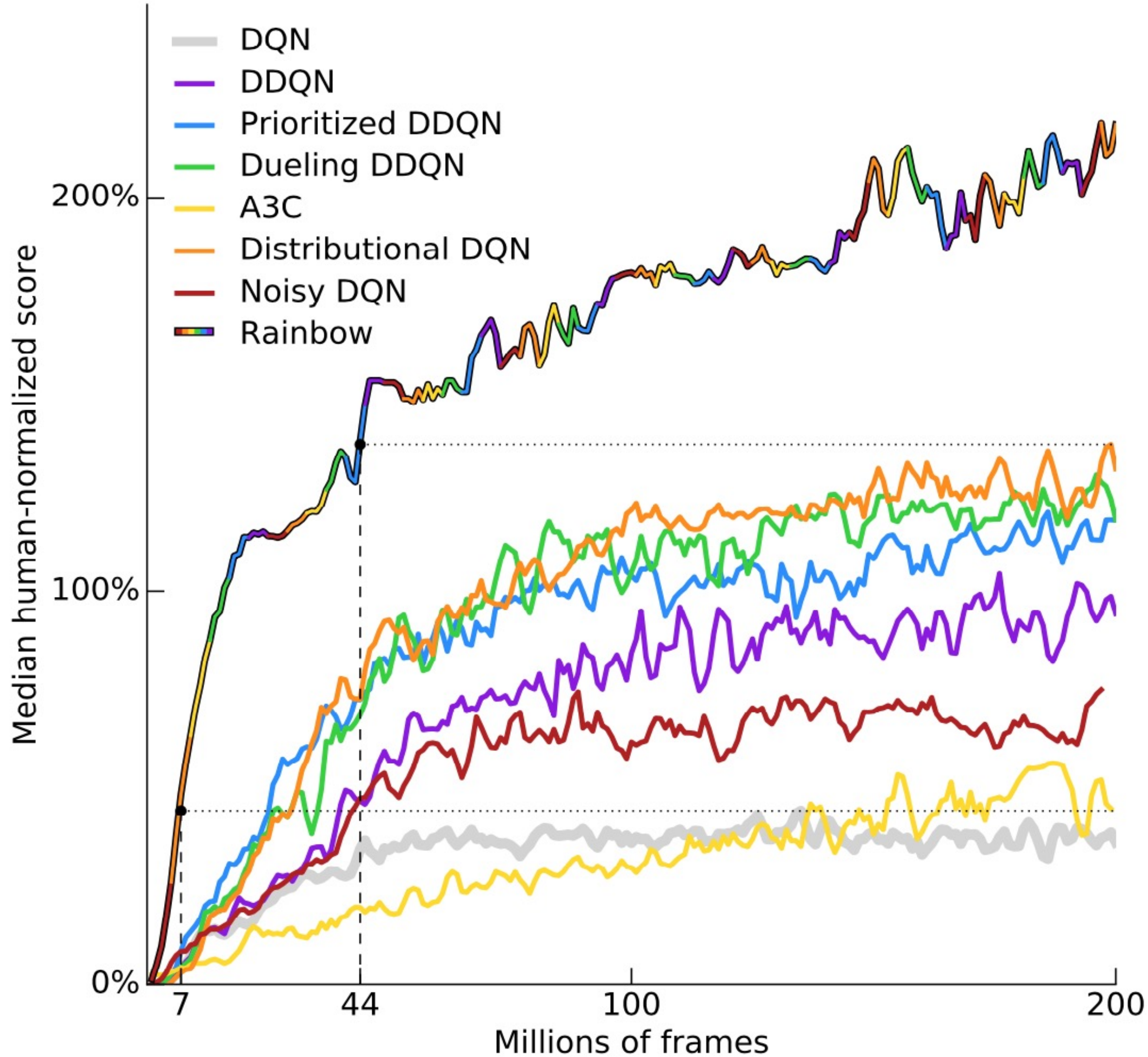
Source: https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html

Dueling Network Architectures for Deep Reinforcement Learning



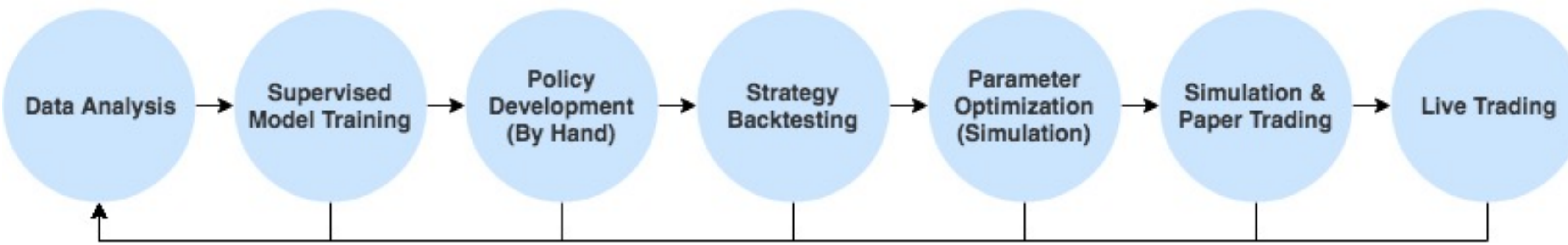
Source: Ziyu Wang et al. (2015). "Dueling network architectures for deep reinforcement learning." arXiv preprint arXiv:1511.06581 (2015)

Rainbow: Combining improvements in deep reinforcement learning



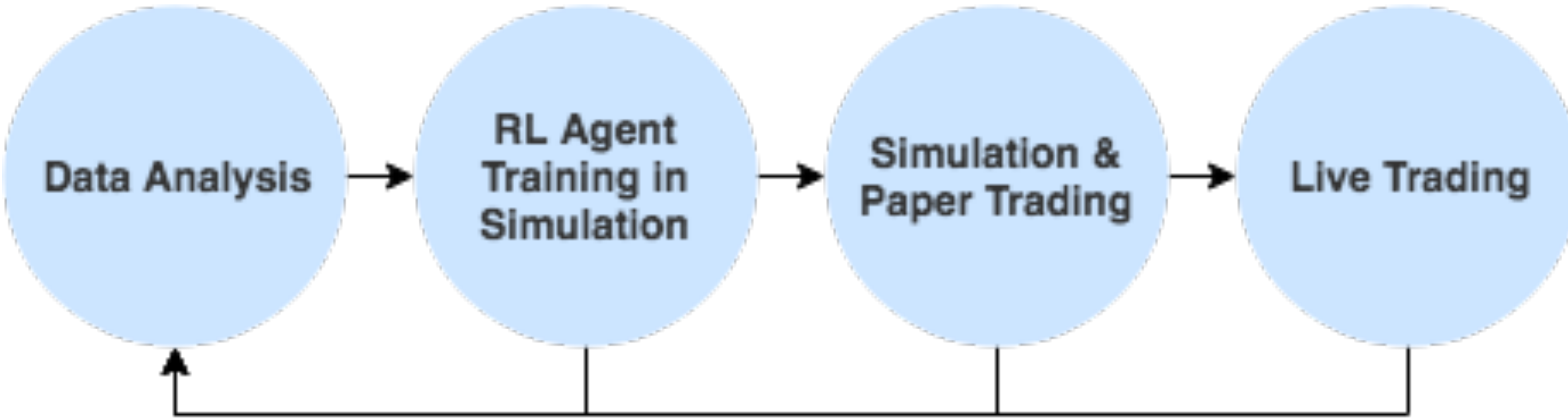
Source: Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver (2017). "Rainbow: Combining improvements in deep reinforcement learning." arXiv preprint arXiv:1710.02298 (2017).

A Typical Strategy Development Workflow



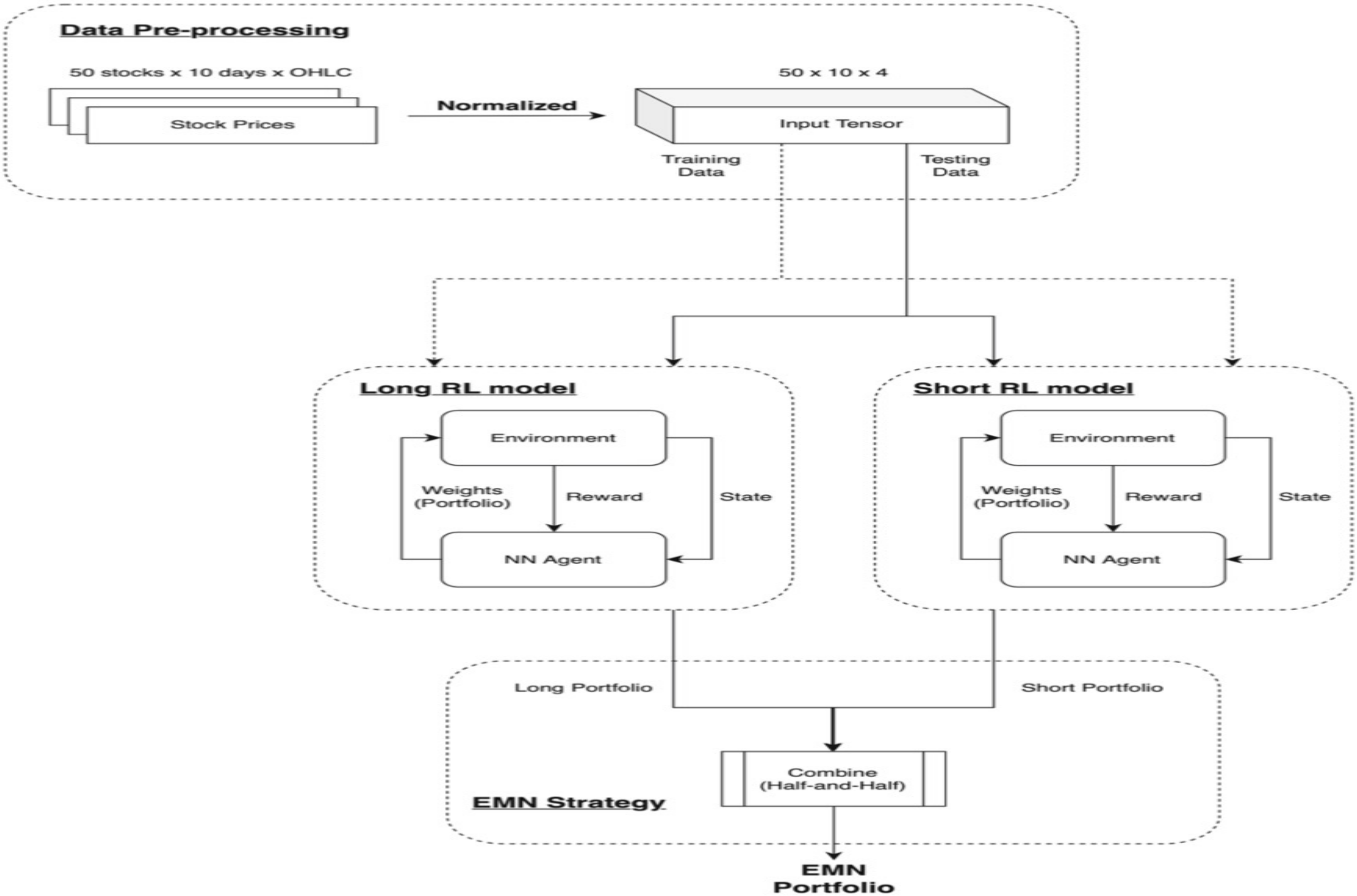
Source: http://www.wildml.com/2018/02/introduction-to-learning-to-trade-with-reinforcement-learning/

Reinforcement Learning (RL) in Trading Strategies



Source: http://www.wildml.com/2018/02/introduction-to-learning-to-trade-with-reinforcement-learning/

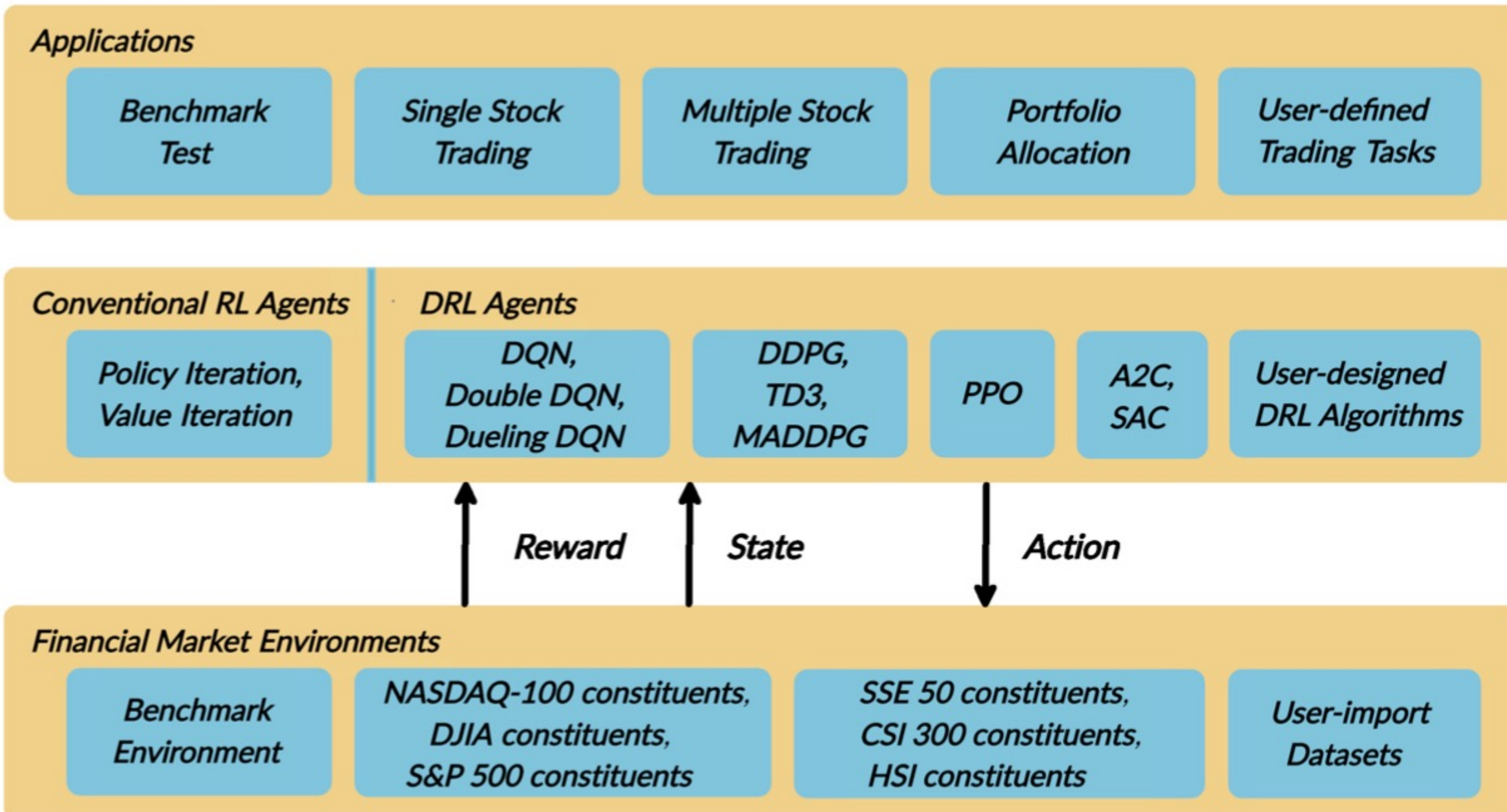
Portfolio management system in equity market neutral using reinforcement learning
(Wu et al., 2021)



Source: Mu-En Wu, Jia-Hao Syu, Jerry Chun-Wei Lin, and Jan-Ming Ho. "Portfolio management system in equity market neutral using reinforcement learning." Applied Intelligence (2021): 1-13.

FinRL:

A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance



Source: Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang (2020). "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance." arXiv preprint arXiv:2011.09607 (2020).

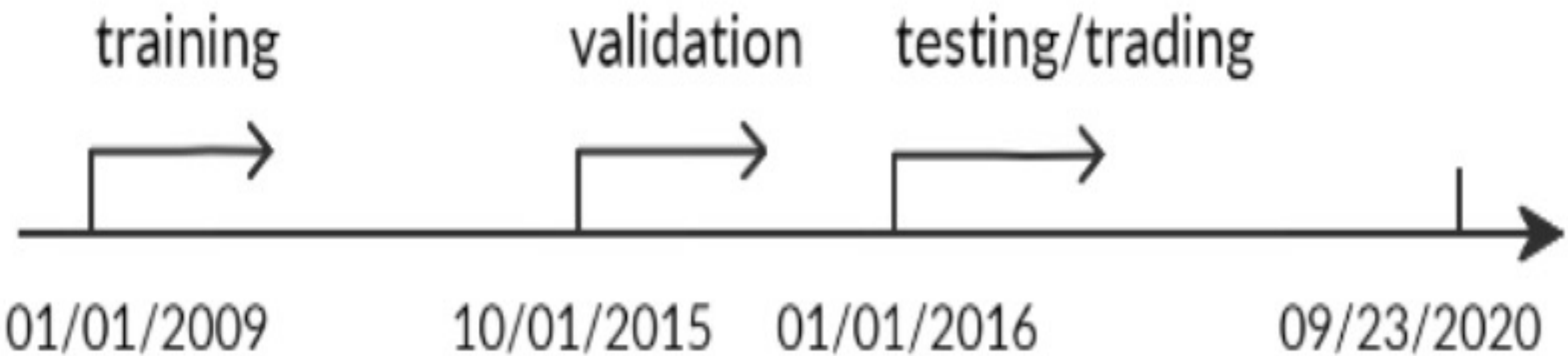
FinRL

Deep Reinforcement Learning Algorithms

Algorithms	Input	Output	Type	State-action spaces support	Finance use cases support	Features and Improvements	Advantages
DQN	States	Q-value	Value based	Discrete only	Single stock trading	Target network, experience replay	Simple and easy to use
Double DQN	States	Q-value	Value based	Discrete only	Single stock trading	Use two identical neural network models to learn	Reduce overestimations
Dueling DQN	States	Q-value	Value based	Discrete only	Single stock trading	Add a specialized dueling Q head	Better differentiate actions, improves the learning
DDPG	State action pair	Q-value	Actor-critic based	Continuous only	Multiple stock trading, portfolio allocation	Being deep Q-learning for continuous action spaces	Better at handling high-dimensional continuous action spaces
A2C	State action pair	Q-value	Actor-critic based	Discrete and continuous	All use cases	Advantage function, parallel gradients updating	Stable, cost-effective, faster and works better with large batch sizes
PPO	State action pair	Q-value	Actor-critic based	Discrete and continuous	All use cases	Clipped surrogate objective function	Improve stability, less variance, simply to implement
SAC	State action pair	Q-value	Actor-critic based	Continuous only	Multiple stock trading, portfolio allocation	Entropy regularization, exploration-exploitation trade-off	Improve stability
TD3	State action pair	Q-value	Actor-critic based	Continuous only	Multiple stock trading, portfolio allocation	Clipped double Q-Learning, delayed policy update, target policy smoothing.	Improve DDPG performance
MADDPG	State action pair	Q-value	Actor-critic based	Continuous only	Multiple stock trading, portfolio allocation	Handle multi-agent RL problem	Improve stability and performance

Source: Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang (2020). "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance." arXiv preprint arXiv:2011.09607 (2020).

FinRL:
A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance
Evaluation of Trading Performance
Training-Validation-Testing Flow



Source: Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang (2020). "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance." arXiv preprint arXiv:2011.09607 (2020).

Reinforcement Learning (RL)
FinRL
Performance of single stock trading
using Proximal policy optimization (PPO) in the FinRL library

Source: Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang (2020). "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance." arXiv preprint arXiv:2011.09607 (2020).

Reinforcement Learning (RL)
FinRL
Performance of multiple stock trading and portfolio allocation using the FinRL library

Source: Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang (2020). "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance." arXiv preprint arXiv:2011.09607 (2020).

Reinforcement Learning (RL)
FinRL
Performance of single stock trading using Proximal policy optimization (PPO) in the FinRL library

2019/01/01-2020/09/23	SPY	QQQ	GOOGL	AMZN	AAPL	MSFT	S&P 500
Initial value	100,000	100,000	100,000	100,000	100,000	100,000	100,000
Final value	127,044	163,647	174,825	192,031	173,063	172,797	133,402
Annualized return	14.89%	32.33%	37.40%	44.94%	36.88%	36.49%	17.81%
Annualized Std	9.63%	27.51%	33.41%	29.62%	25.84%	33.41%	27.00%
Sharpe ratio	1.49	1.16	1.12	1.40	1.35	1.10	0.74
Max drawdown	20.93%	28.26%	27.76%	21.13%	22.47%	28.11%	33.92%

Source: Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang (2020). "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance." arXiv preprint arXiv:2011.09607 (2020).

Reinforcement Learning (RL)
FinRL

Performance of multiple stock trading and portfolio allocation

over the DJIA constituents stocks using the FinRL library

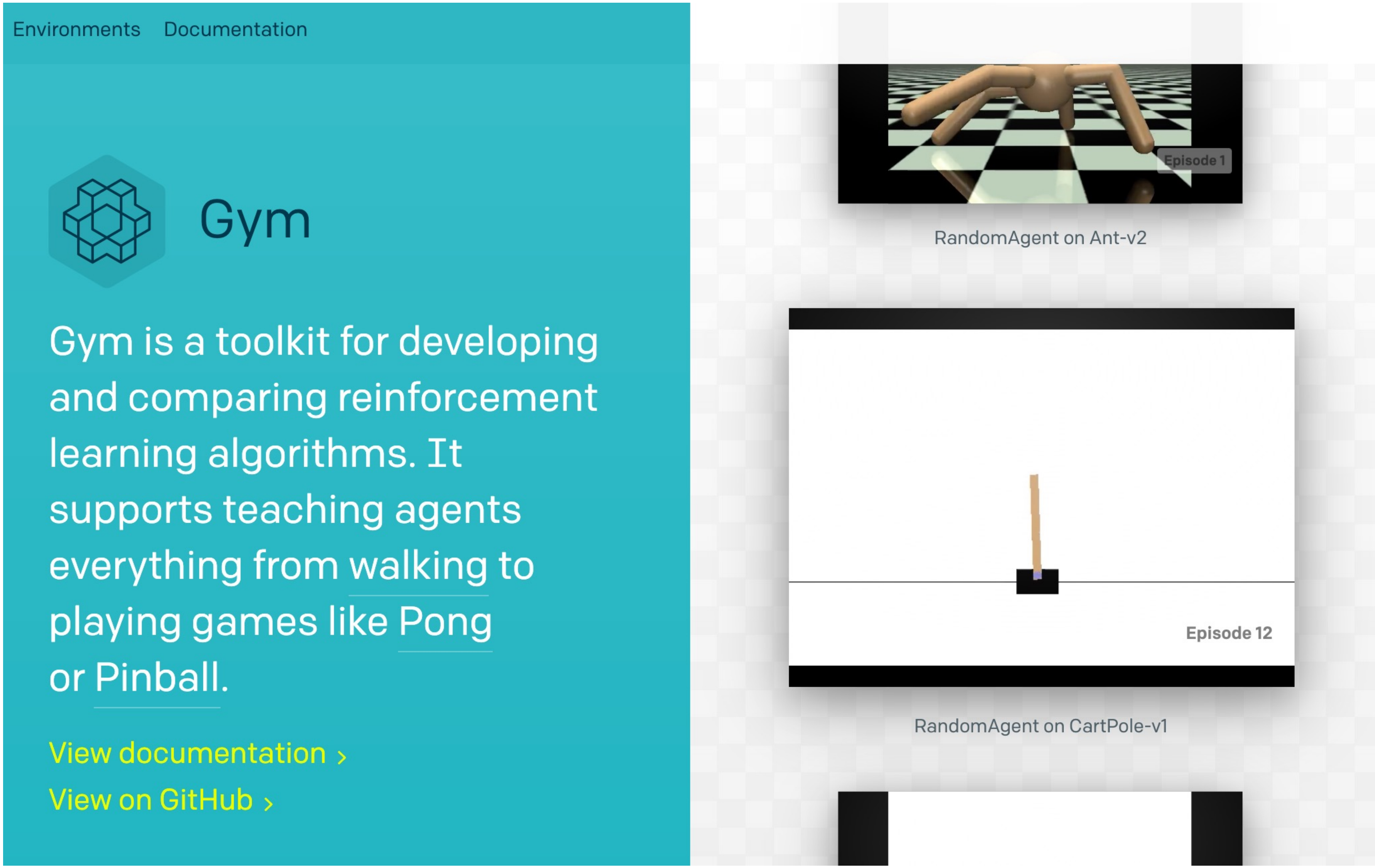
2019/01/01-2020/09/23	TD3	DDPG	Min-Var.	DJIA
Initial value	1,000,000	1,000,000	1,000,000	1,000,000
Final value	1,403,337; 1,381,120	1,396,607; 1,281,120	1,171,120	1,185,260
Annualized return	21.40%; 17.61%	20.34%; 15.81%	8.38%	10.61%
Annualized Std	14.60%; 17.01%	15.89%; 16.60%	26.21%	28.63%
Sharpe ratio	1.38; 1.03	1.28; 0.98	0.44	0.48
Max drawdown	11.52% 12.78%	13.72%; 13.68%	34.34%	37.01%

Source: Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang (2020). "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance." arXiv preprint arXiv:2011.09607 (2020).

Deep Reinforcement Learning Library

- OpenAI Gym
- Google Dopamine
- RLlib
- Horizon
- FinRL

Open AI Gym



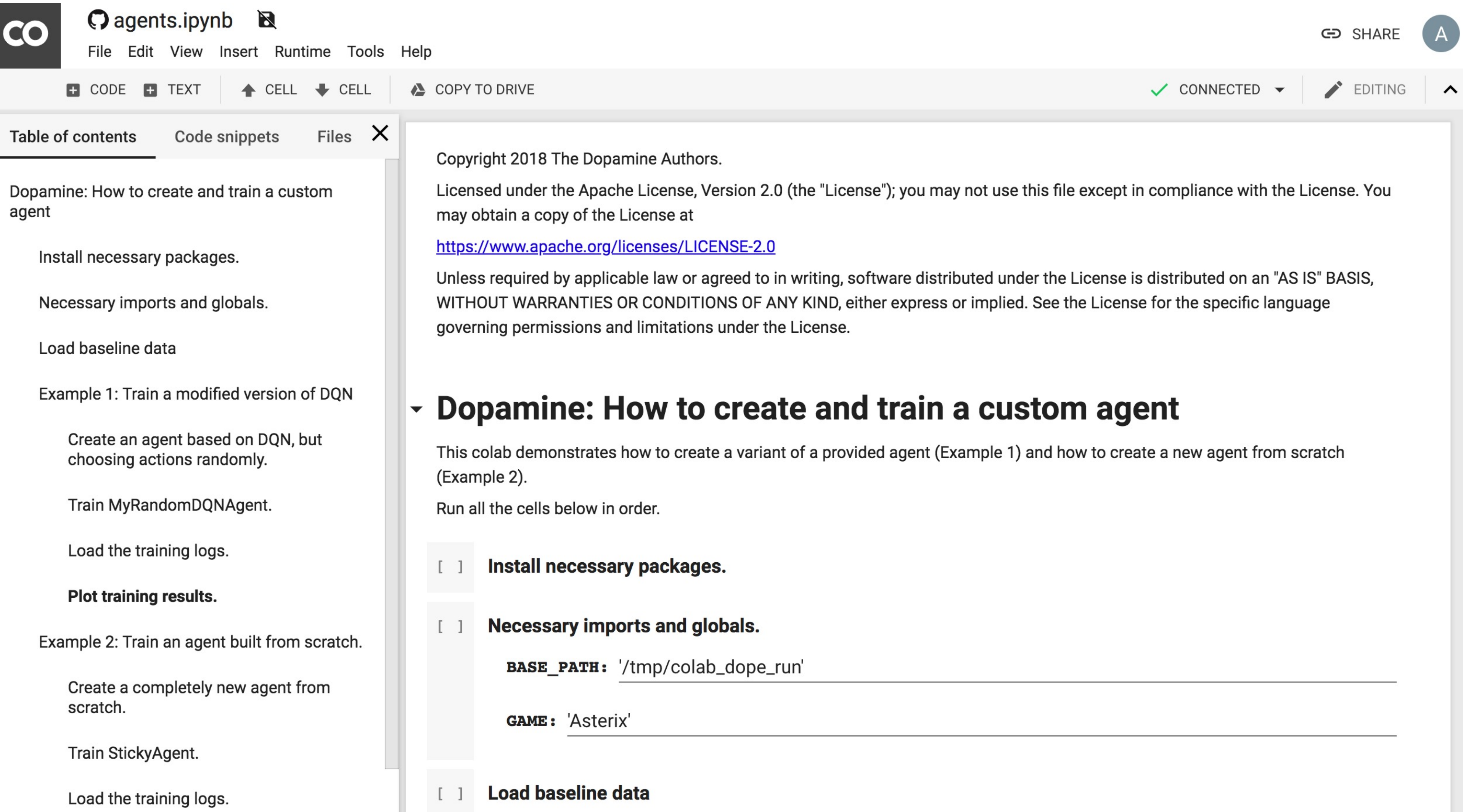
https://gym.openai.com/

Google Dopamine

Dopamine is a research framework for fast prototyping of reinforcement learning algorithms.

https://github.com/google/dopamine

Deep Reinforcement Learning
Dopamine Colab Examples
DQN Rainbow



https://colab.research.google.com/github/google/dopamine/blob/master/dopamine/colab/agents.ipynb

RLlib:
Scalable Reinforcement Learning

Examples

Tune API Reference

Contributing to Tune

RLLIB

RLlib: Scalable Reinforcement Learning

RLlib Table of Contents

RLlib Training APIs

RLlib Environments

RLlib Models, Preprocessors, and Action Distributions

RLlib Algorithms

RLlib Sample Collection and Trajectory Views

RLlib Offline Datasets

RLlib Concepts and Custom Algorithms

RLlib Examples

RLlib Package Reference

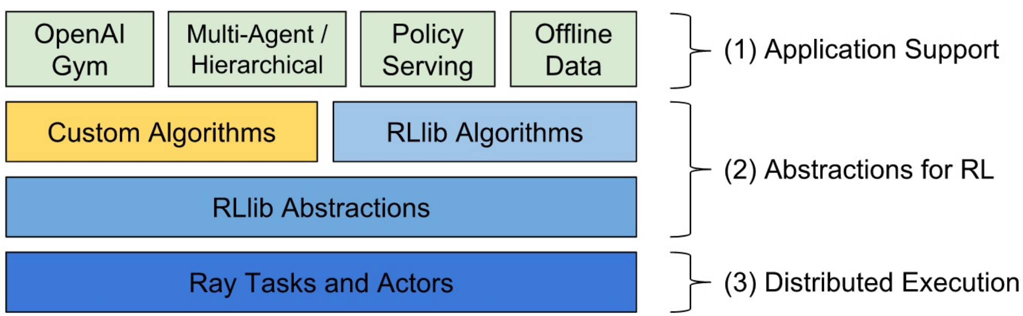
Contributing to RLlib

RAY SGD

RaySGD: Distributed Training

RLlib: Scalable Reinforcement Learning

RLlib is an open-source library for reinforcement learning that offers both high scalability and a unified API for a variety of applications. RLlib natively supports TensorFlow, TensorFlow Eager, and PyTorch, but most of its internals are framework agnostic.



To get started, take a look over the custom env example and the API documentation. If you're looking to develop custom algorithms with RLlib, also check out concepts and custom algorithms.

RLlib in 60 seconds

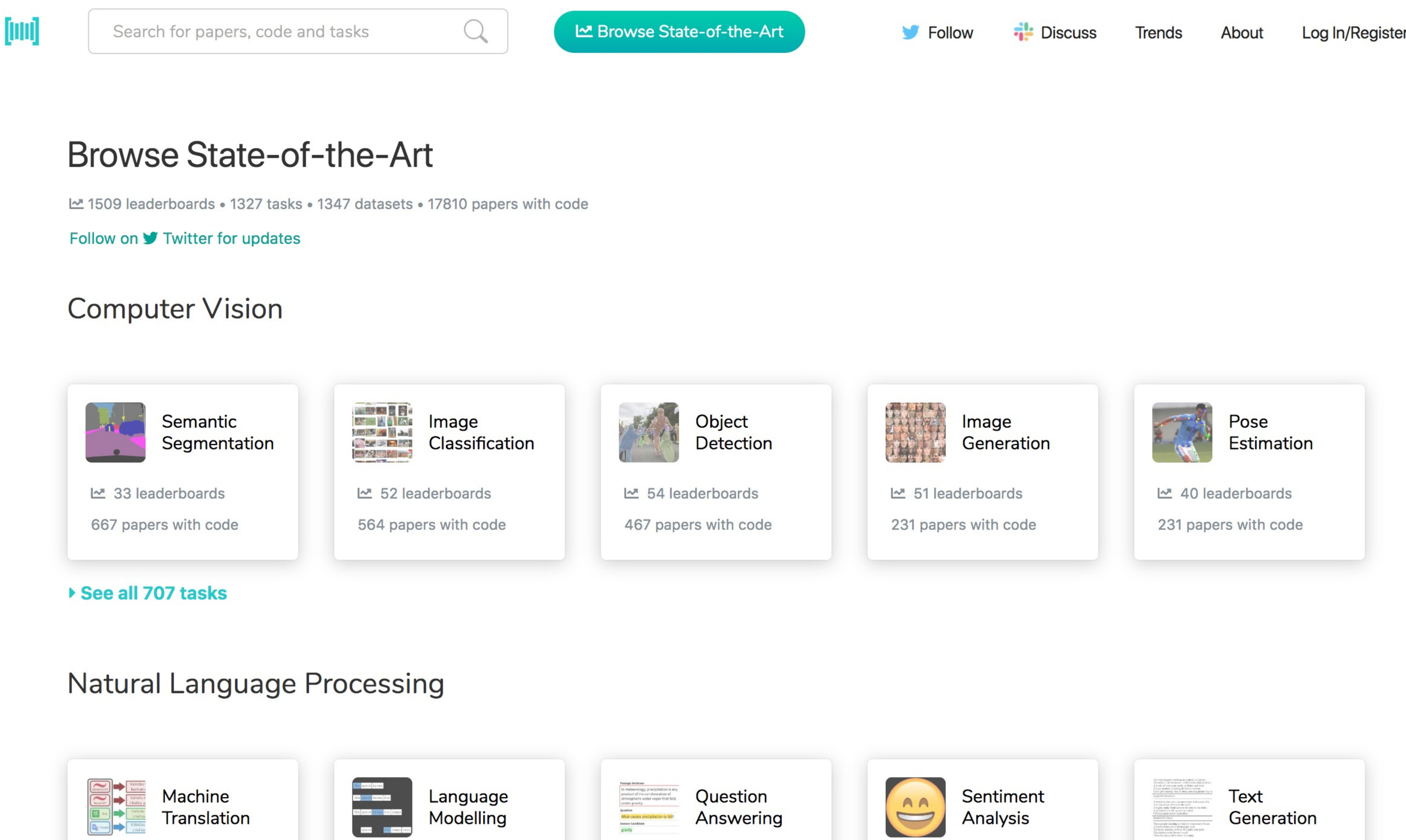
The following is a whirlwind overview of RLlib. For a more in-depth guide, see also the full table of contents and RLlib blog posts. You may also want to skim the list of built-in algorithms. Look out for the and icons to see which algorithms are available for each framework.

Contents

v: master

https://docs.ray.io/en/master/rllib.html

Papers with Code
State-of-the-Art (SOTA)

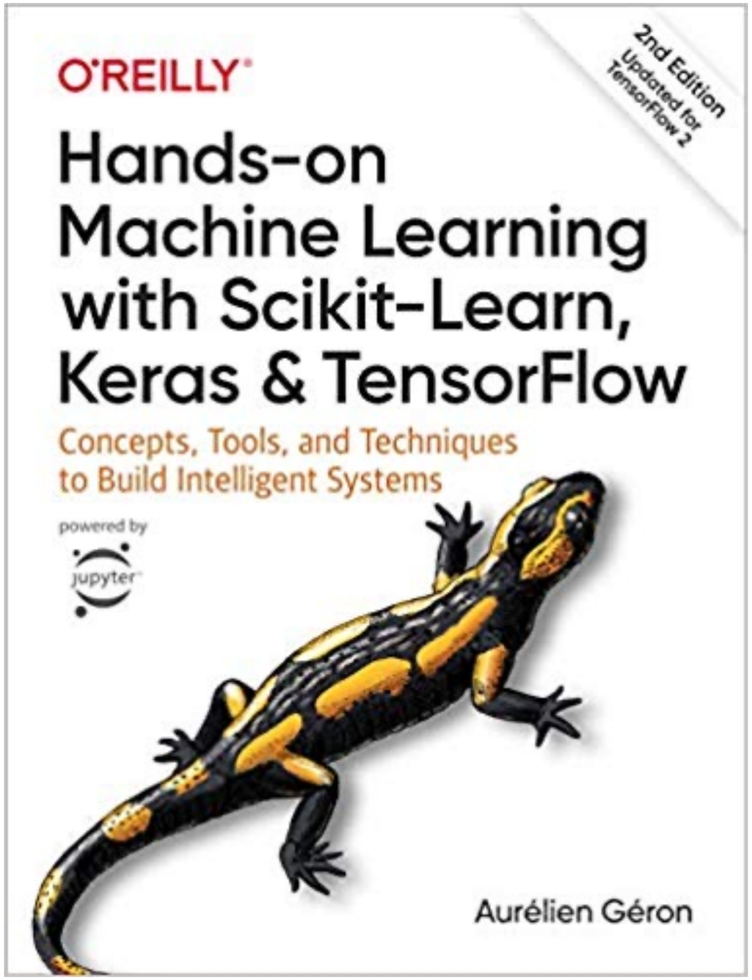


https://paperswithcode.com/sota

Aurélien Géron (2019),

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd Edition

O'Reilly Media, 2019



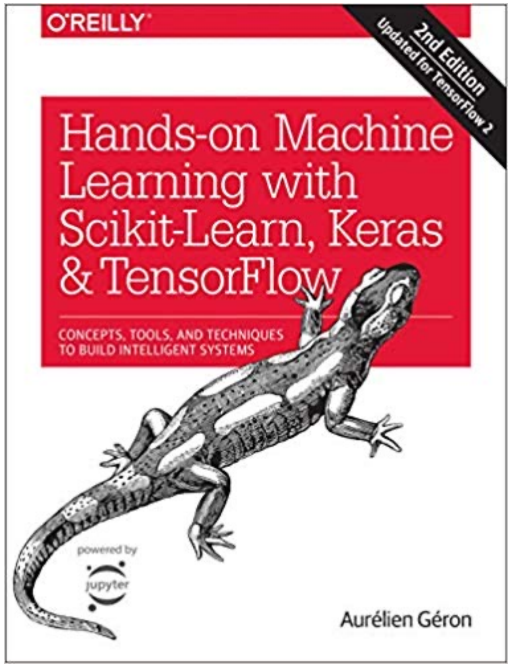
https://github.com/ageron/handson-ml2

Source: https://www.amazon.com/Hands-Machine-Learning-Scikit-Learn-TensorFlow/dp/1492032646/

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

Notebooks

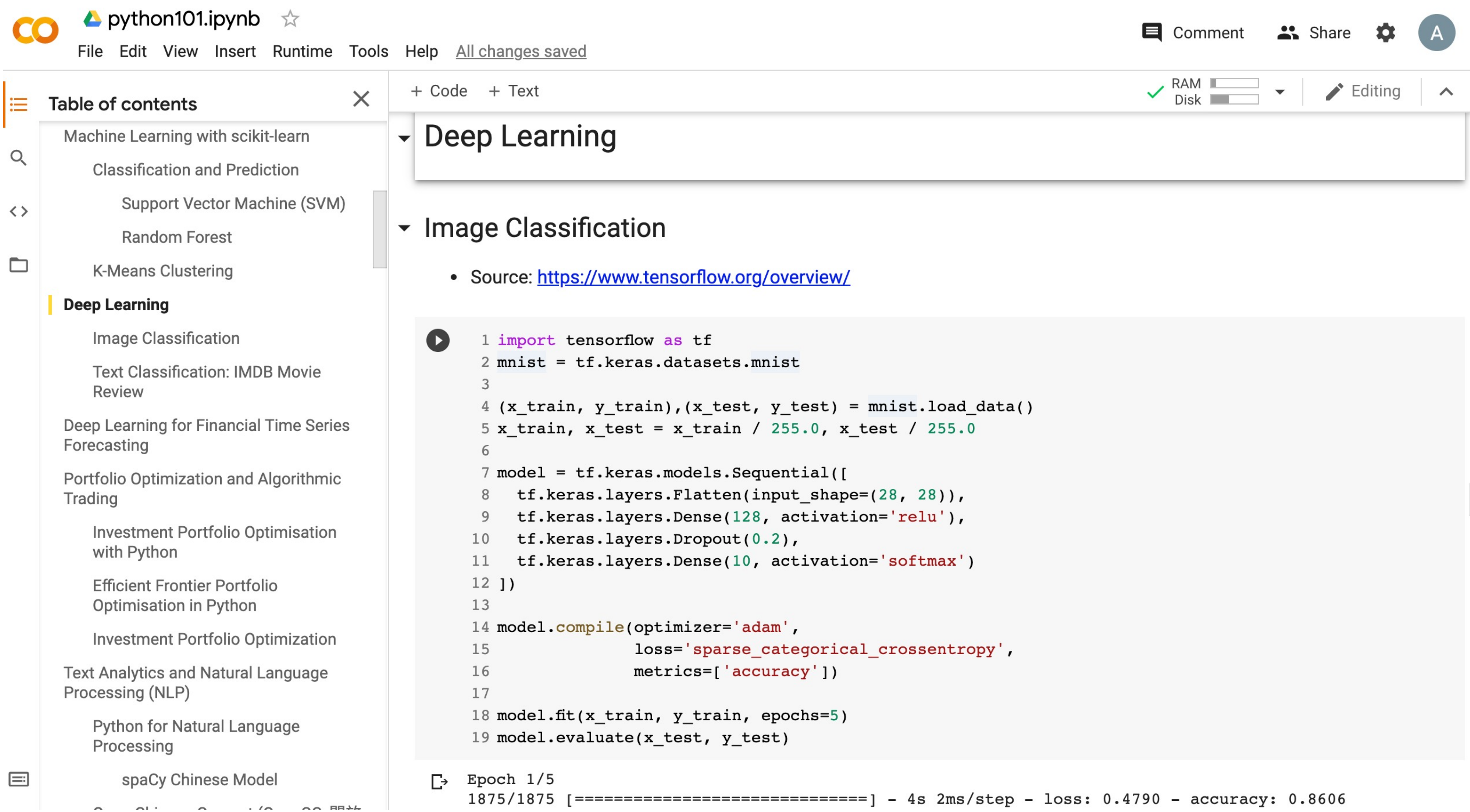
1. The Machine Learning landscape
2. End-to-end Machine Learning project
3. Classification
4. Training Models
5. Support Vector Machines
6. Decision Trees
7. Ensemble Learning and Random Forests
8. Dimensionality Reduction
9. Unsupervised Learning Techniques
10. Artificial Neural Nets with Keras
11. Training Deep Neural Networks
12. Custom Models and Training with TensorFlow
13. Loading and Preprocessing Data
14. Deep Computer Vision Using Convolutional Neural Networks
15. Processing Sequences Using RNNs and CNNs
16. Natural Language Processing with RNNs and Attention
17. Representation Learning Using Autoencoders
18. Reinforcement Learning
19. Training and Deploying TensorFlow Models at Scale



https://github.com/ageron/handson-ml2

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Summary

- Reinforcement Learning (RL)
  - Markov Decision Processes (MDP)
- Deep Reinforcement Learning (DRL) Algorithms
  - SARSA
  - Q-Learning
  - DQN
  - A3C
  - Rainbow

References

- Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson.
- Aurélien Géron (2019), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd Edition, O'Reilly Media.
- Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.
- David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ
- David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, Demis Hassabis (2018), "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play." Science 362, no. 6419 (2018): 1140-1144.
- David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis (2017), "Mastering the game of Go without human knowledge." Nature 550 (2017): 354–359.
- Hado Van Hasselt, Arthur Guez, and David Silver (2016). "Deep Reinforcement Learning with Double Q-Learning." In AAAI, vol. 2, p. 5. 2016.
- Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver (2017). "Rainbow: Combining improvements in deep reinforcement learning." arXiv preprint arXiv:1710.02298 (2017).
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves et al. (2015) "Human-level control through deep reinforcement learning." Nature 518, no. 7540 (2015): 529.
- Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas (2015). "Dueling network architectures for deep reinforcement learning." arXiv preprint arXiv:1511.06581 (2015).
- Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang (2020). "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance." arXiv preprint arXiv:2011.09607 (2020).
- Mu-En Wu, Jia-Hao Syu, Jerry Chun-Wei Lin, and Jan-Ming Ho. "Portfolio management system in equity market neutral using reinforcement learning." Applied Intelligence (2021): 1-13.
- Min-Yuh Day (2021), Python 101, https://tinyurl.com/aintpupython101