Introduction to Machine Learning Methods in Condensed Matter Physics

LECTURE 5, FALL 2021

Yi Zhang (张亿)

International Center for Quantum Materials, School of Physics
Peking University, Beijing, 100871, China

Email: frankzhangyi@pku.edu.cn

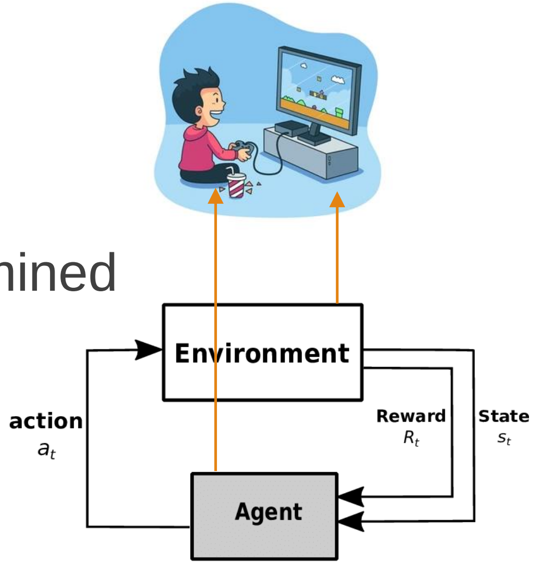
What is and why reinforcement learning?

- An agent (strategy) that interacts with an environment and maximizes reward (minimize) penalty, e.g.:
  - Board games, computer games, puzzles
  - Autopilot for cars and airplanes
  - Robot controls, etc.
  - "Artificial intelligence" of a given task

- General setup: S: current state; A: action upon state; R: reward

$S \xrightarrow{A} S' \xrightarrow{A'} S'' \xrightarrow{A''} \cdots \rightarrow$ end, where the *accumulated reward* is determined

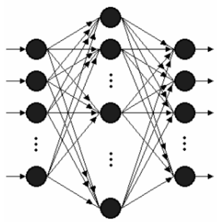
  - Video games: S: screen; A: joystick input; R: score, life, cleared levels …
  - Chess, Goal: S: current board configuration; A: next move; R: win, points …
  - Rubik's cube: S: current colorings; A: next twist; R: - steps taken…



What is and why reinforcement learning?

- Value function: V-learning and Q-learning:
  - $V(S)$: the approximate most accumulated reward from state S. Then, given S, we can choose then optimal action $\mathrm{argmax}[V(S' = S(A)]$
  - $Q(S, A)$: the approximate most accumulated reward from state S after choosing action A. Then, given S, we can choose the optimal action $\mathrm{argmax}[Q(S, A)]$
  - Example: Q-learning table
  - Better to represent $V(S)$ and/or $Q(S, A)$ with ANNs

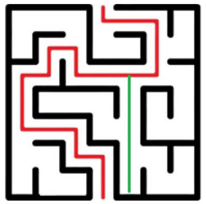
Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	328	-2.30108105	-1.97092096	-2.30357004	-2.20591839	-10.3607344	-8.5583017
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	499	9.96984239	4.02706992	12.96022777	29	3.32877873	3.38230603



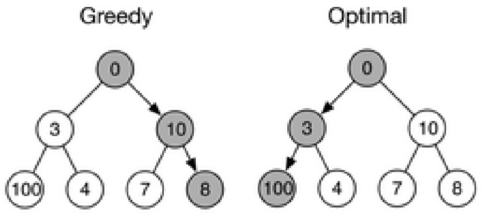
- For supervised machine learning, we update the parameters (with MSE cost function) as:

ANN as a regression: $\Delta w_t = \alpha\,(z - V(s_t))\,\nabla_w V(s_t),$

s_t: input sample, z: target value, V: current output, α: learning rate, w: ANN parameters.



- Why not simply apply supervised machine learning?
  - Accumulated score is only available at the very end, but not at each individual step.
  - Sometimes, a good move locally can be a bad move globally! Example: greedy algorithm



Temporal-difference (TD) learning

- V-learning for a sequence of steps: $$w \leftarrow w + \sum_{t=1}^{m} \Delta w_t$$

  m: number of steps, $V_{m+1} = z$.

- An insight that allows us to treat such multi-step prediction problem incrementally:

$$w \leftarrow w + \sum_{t=1}^{m} \alpha (z - V_t) \nabla_w V_t,$$

$$z - V_t = \sum_{k=t}^{m} (V_{k+1} - V_k)$$

$$w \leftarrow w + \sum_{t=1}^{m} \alpha \sum_{k=t}^{m} (V_{k+1} - V_k) \nabla_w V_t.$$

successive prediction values, hence 'temporal difference'

Learning occurs for previous states: $$w \leftarrow w + \sum_{t=1}^{m} \alpha \left(V_{t+1} - V_t\right) \sum_{k=1}^{t} \nabla_w V_k \Rightarrow \Delta w_t = \alpha \left(V_{t+1} - V_t\right) \sum_{k=1}^{t} \nabla_w V_k.$$

- Generalization: (a decay to discount past gradients, temporal correlation)

$$\Delta w_t = \alpha \left(V_{t+1} - V_t\right) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w V_k.$$

$\lambda = 0$ limit: Markov chain $$\Delta w_t = \alpha \left(V_{t+1} - V_t\right) \nabla_w V_t.$$

The subsequent back propagation is similar to supervised machine learning.

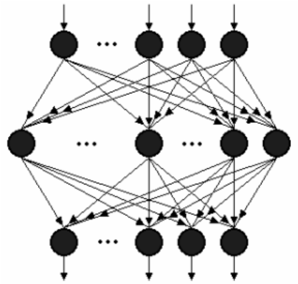
TD-Gammon

- How to gather training samples? Play by itself / against itself!

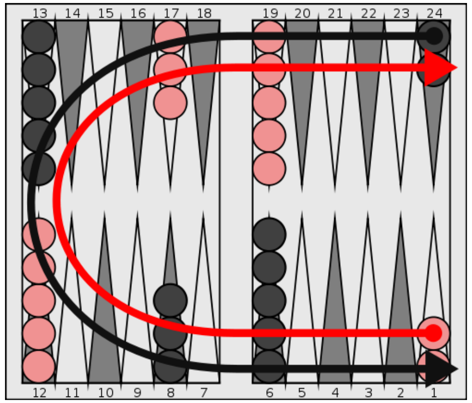
$V(S)$

Using ANN for practices ↓ ↑ Training ANN for better performance

Training sets



Single hidden-layer ANN for backgammon



For competitive strategy (two players playing against each other):

$S \rightarrow S' \rightarrow S'' \rightarrow \cdots \rightarrow$ switch opponent/self after each step

- World-champion level performance, novel prior steps:
- Understanding the learning process:
  - Only fair absolute accuracy but *good enough* relative accuracy
  - A stochastic noise source helps to discover new strategies
  - Learn local, linear concepts first before nonlinear correlated ones

Program	Training Games	Opponents	Results
TDG 1.0	300,000	Robertie, Davis, Magriel	-13 pts/51 games (-0.25 ppg)
TDG 2.0	800,000	Goulding, Woolsey, Snellings, Russell, Sylvester	-7 pts/38 games (-0.18 ppg)
TDG 2.1	1,500,000	Robertie	-1 pt/40 games (-0.02 ppg)

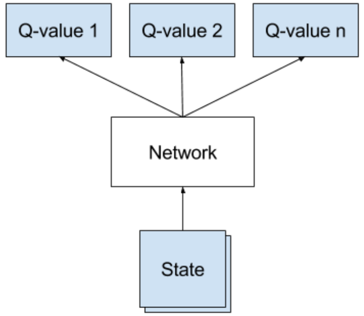
Table 1. Results of testing TD-Gammon in play against world-class human opponents. Version 1.0 used 1-play search for move selection; versions 2.0 and 2.1 used 2-ply search. Version 2.0 had 40 hidden units; versions 1.0 and 2.1 had 80 hidden units.

Gerald Tesauro, Communications of the ACM 38, 3 (1995).

Deep Q-learning

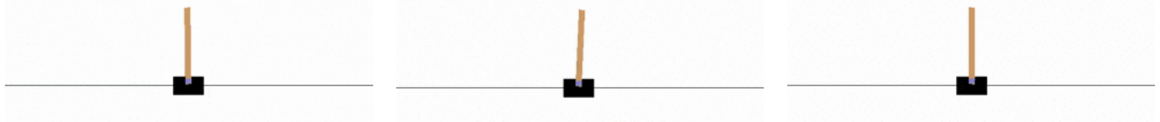
The Bellman equation:

$$Q^{new}(s_t,a_t) \leftarrow \underbrace{Q(s_t,a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\bigg(\underbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1},a)}_{\text{estimate of optimal future value}}}_{\text{new value (temporal difference target)}} - \underbrace{Q(s_t,a_t)}_{\text{old value}}\bigg)}^{\text{temporal difference}}$$



- Two deep ANNs for stability: use the 2nd ANN to update the 1st ANN, which updates the 2nd ANN at intervals
- Memory optimization: experience replay, prioritize the samples with high values
- Nevertheless, convergence is not guaranteed…
- Example: OpenAI gym (https://gym.openai.com/)

The cart-and-pole game: for your game of interest, simply modify the environment (and reward)



Random exploration $\epsilon = 0.1$, discount factor $\gamma = 0.9$, learning rate $\alpha = 0.5$

https://github.com/vmayoral/basic_reinforcement_learning/blob/master/tutorial4/README.md

Reinforcement learning for fast state preparation

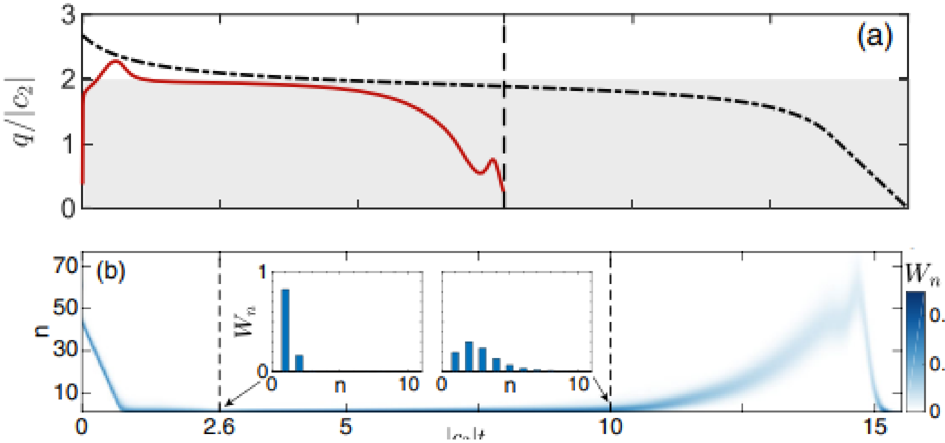
- Conventionally, one can prepare a state through adiabatic evolution:

$$H = \frac{c_2}{2N}\mathbf{L}^2 - q(t)N_0$$ for the Dicke state

The system is kept at the ground state. (black dot-dashed line)

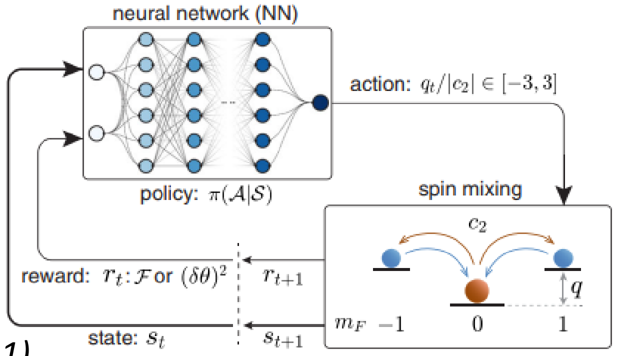
For generic paths, an evaluation (state overlap / fidelity) is only available at the end of path.

- An unconventional route by reinforcement learning for faster preparation (red line):



Shuai-Feng Guo#, Feng Chen#, et al., Phys. Rev. Lett. ***126****, 060401 (2021).*

$$\mathcal{F} = |\langle\psi(t)|\psi^{(0)}_{\text{Dicke}}\rangle|^2$$



for a single state,
more generally:
$H_{tar} \to H \to H' \to \cdots \to H_0$
S: current H; A: ΔH;
R: minimum gap or fidelity on path.
However, *no* good convergence yet.