M A D
Math and Data

NYU COURANT INSTITUTE OF MATHEMATICAL SCIENCES
NYU CENTER FOR DATA SCIENCE

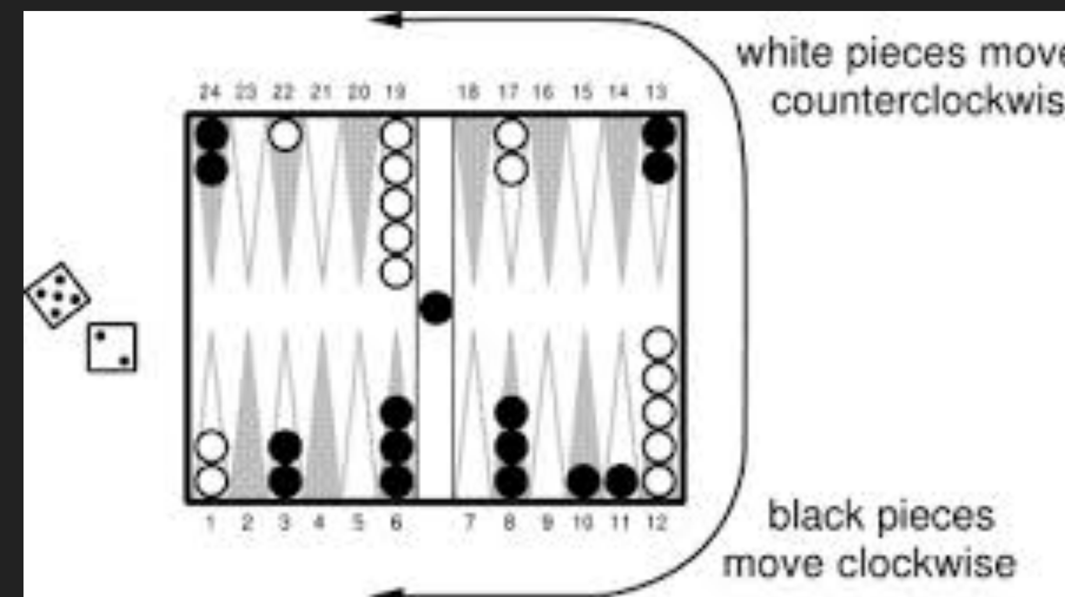
JOAN BRUNA

GEOMETRIC INSIGHTS FOR NONLINEAR TD CONVERGENCE

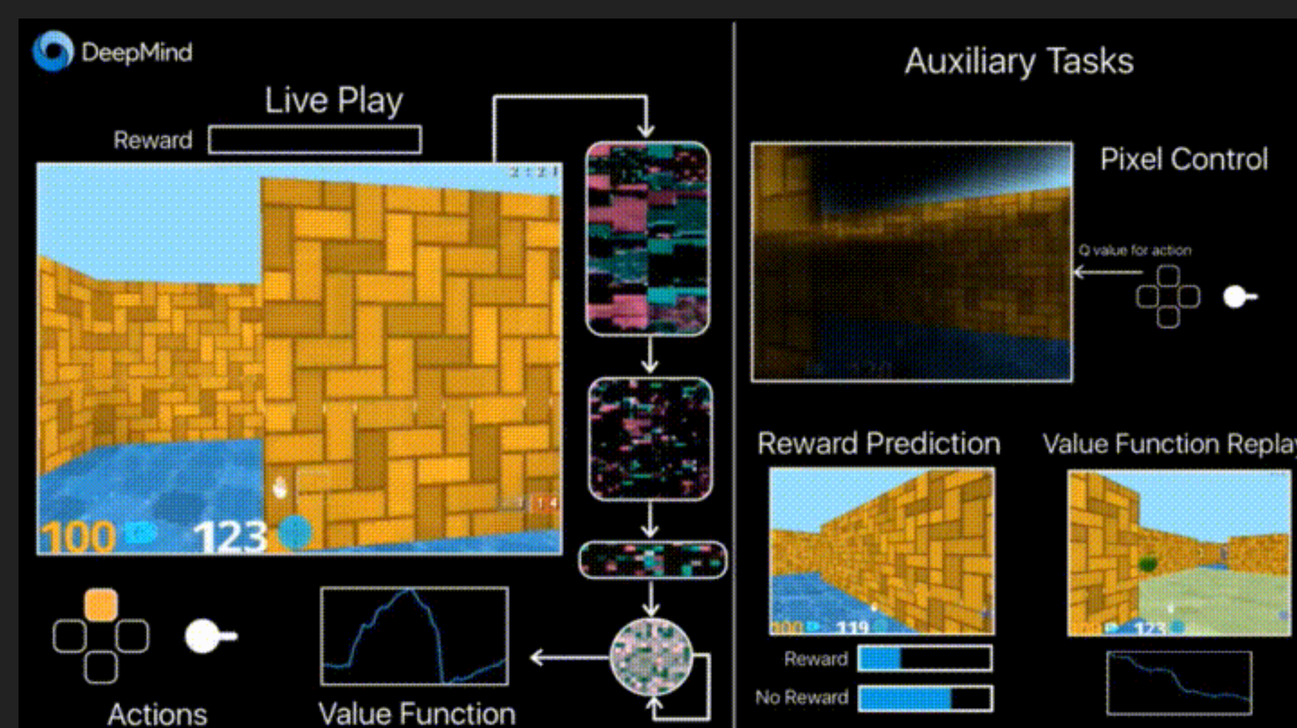
joint work with David Brandfonbrener

- General framework to learn how to interact in complex, high-dimensional environments.

Deepmind'16

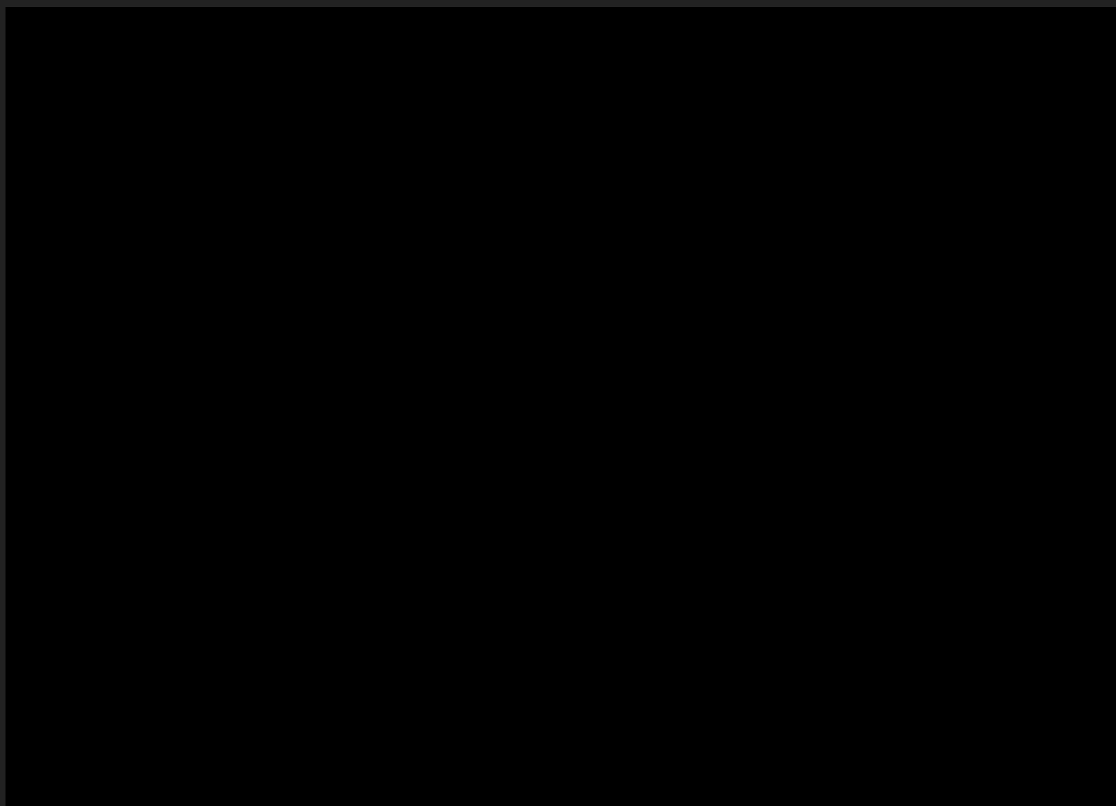


TD-Gammon, Tesauro'92

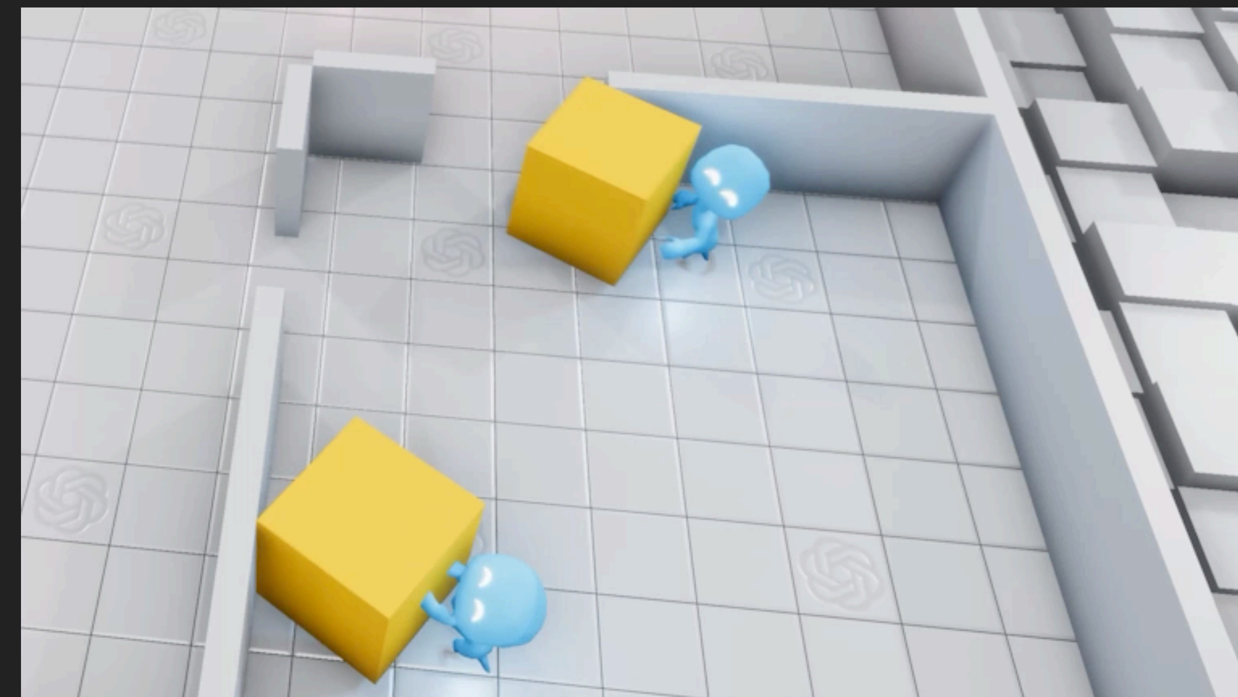


Deepmind'17

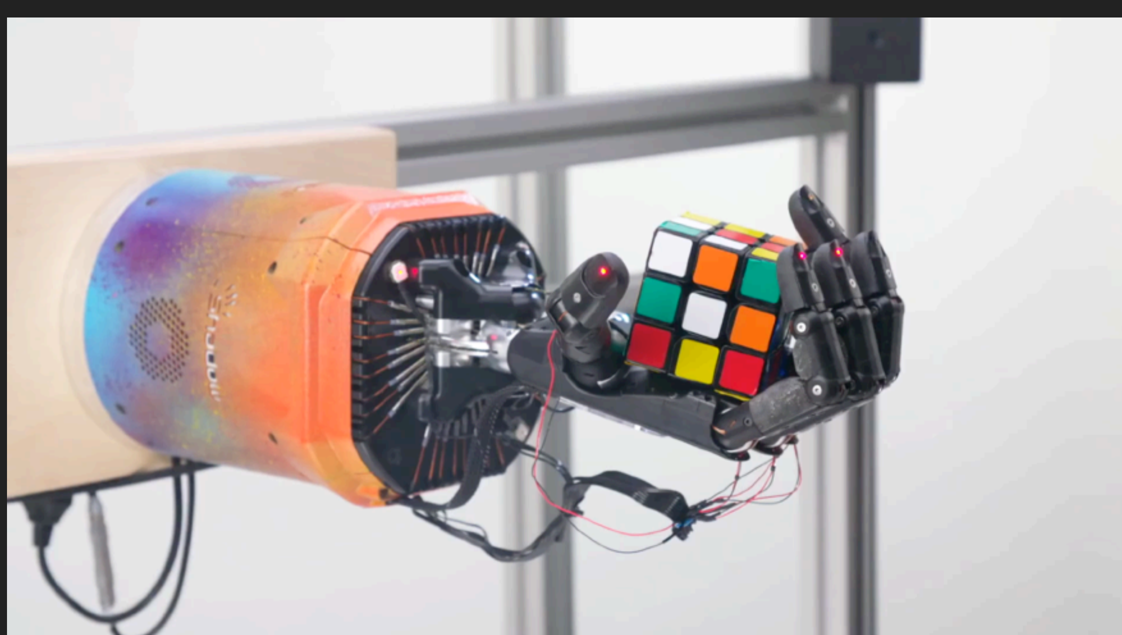
REINFORCEMENT LEARNING



Deepmind'17



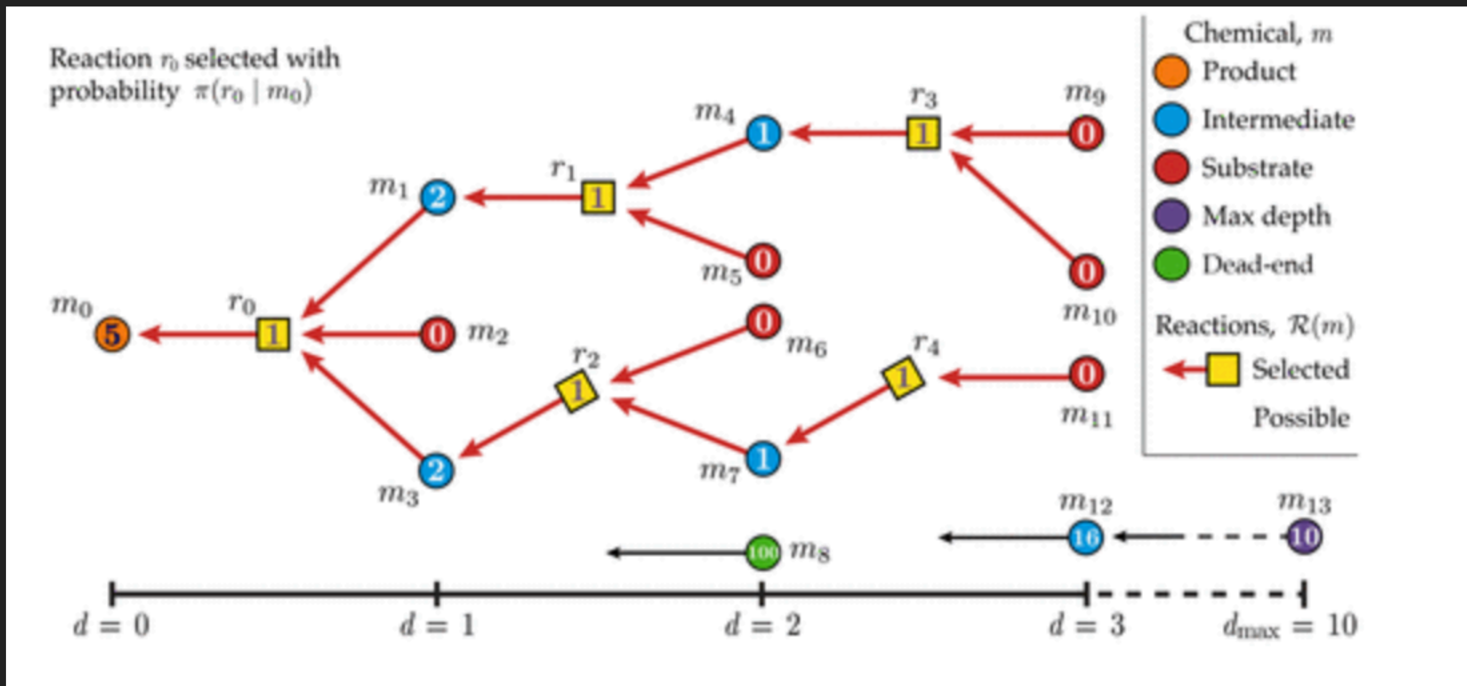
Hide and Seek, OpenAI'19



OpenAI'19

Alphastar, Deepmind'19

REINFORCEMENT LEARNING



Chemical Retrosynthesis, Shreck et al.

Quantum Control

ARTICLE OPEN

Universal quantum control through deep reinforcement learning

Murphy Yuezhen Niu[1,2], Sergio Boixo[2], Vadim N. Smelyanskiy[2] and Hartmut Neven[2]

Reinforcement Learning for Integer Programming: Learning to Cut

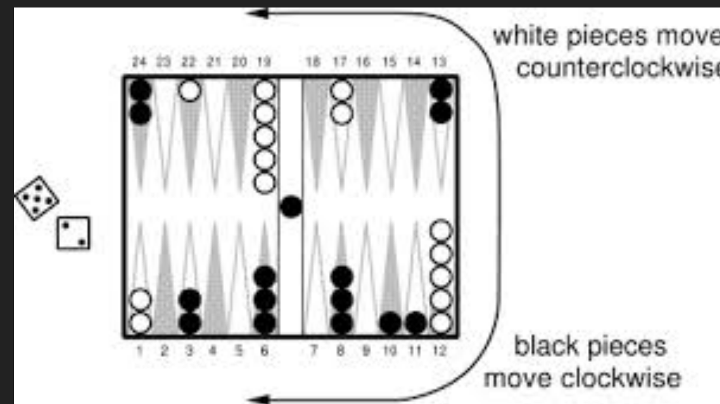
Yunhao Tang Columbia University yt2541@columbia.edu

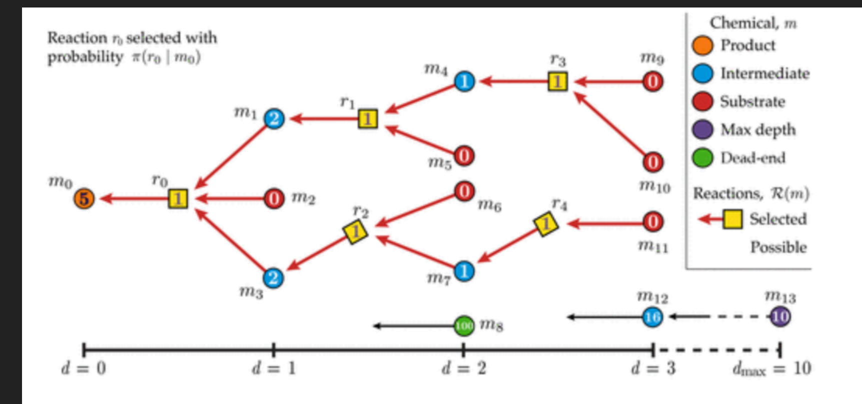
Shipra Agrawal* Columbia University sa3305@columbia.edu

Yuri Faenza Columbia University yf2414@columbia.edu

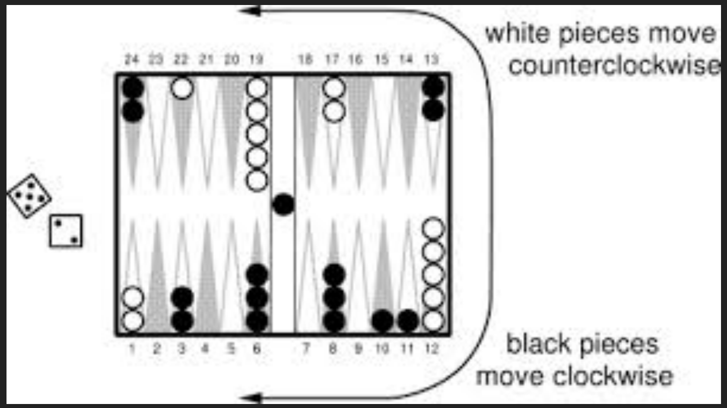
Integer Programming

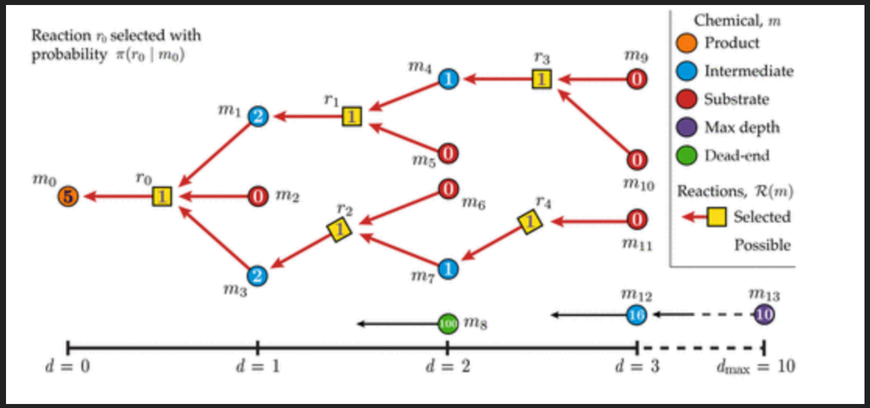
RL TODAY





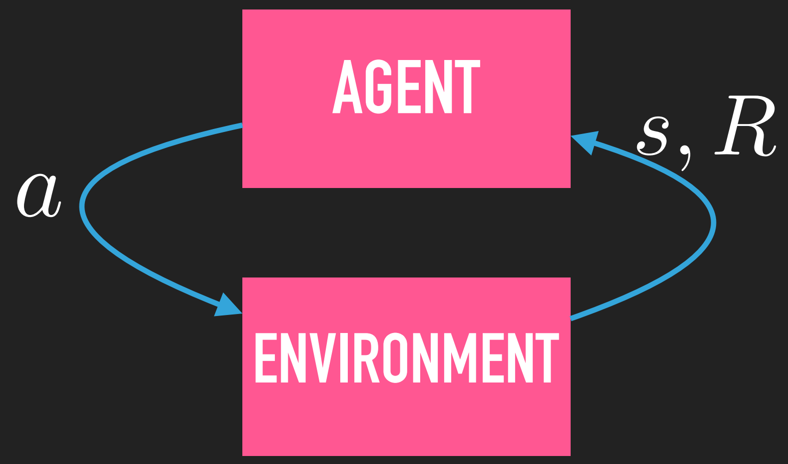
- Remarkable ability to discover useful policies in large environments.
- High-dimensional, noisy, observations.



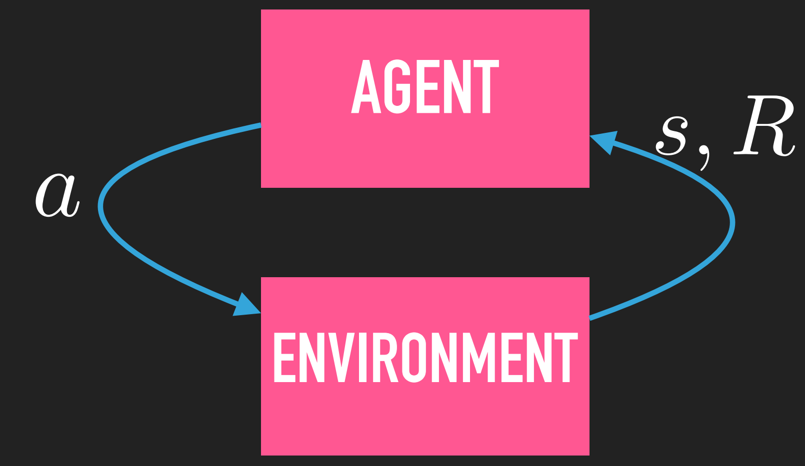


- Remarkable ability to discover useful policies in large environments.
- High-dimensional, noisy, observations.
- Yet, with
  - poor sample efficiency.
  - limited theoretical guarantees.

- Mathematical Setup: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma, \rho)$
  - $\mathcal{S}$: state space (might be discrete or continuous).
  - $\mathcal{A}$: space of actions (assumed the same for all states).
  - $\mathcal{P}(s' \mid s, a)$: Markov transition probability kernel.
  - ρ : initial state distribution.
  - $R(s, a)$: instantaneous reward.
  - γ : discount factor, assume $0 \leq \gamma < 1$.



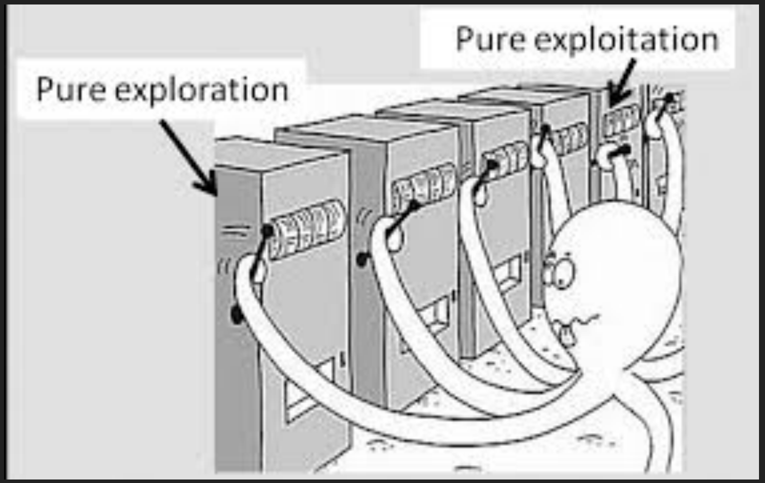
- Mathematical Setup: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma, \rho)$
  - $\mathcal{S}$: state space (might be discrete or continuous).
  - $\mathcal{A}$: space of actions (assumed the same for all states).
  - $\mathcal{P}(s' \mid s, a)$: Markov transition probability kernel.
  - ρ : initial state distribution.
  - $R(s, a)$: instantaneous reward.
  - γ : discount factor, assume $0 \leq \gamma < 1$.



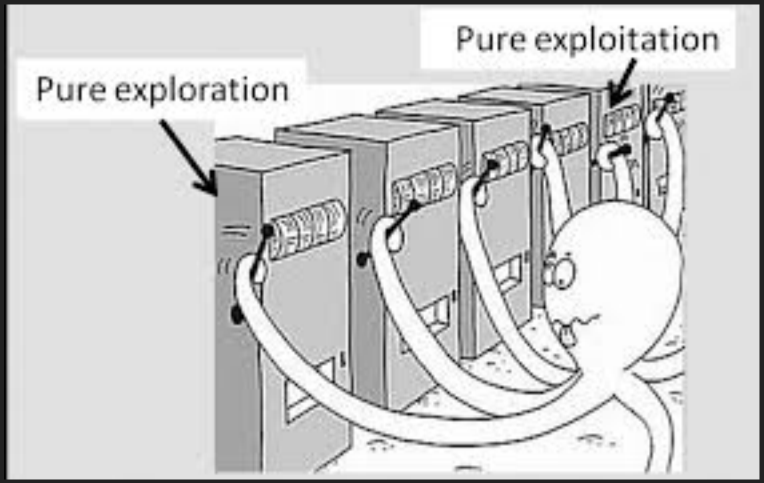
- Goal: Find a *policy* $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes expected sum of discounted rewards:

$$\max_{\pi} \mathbb{E}_{\rho,\mathcal{P}} \sum_k \gamma^k R(s_k, a_k) \text{ subject to } \begin{aligned} & s_{k+1} \sim \mathcal{P}(s'|s_k, a_k) \\ & s_0 \sim \rho \\ & a_k = \pi(s_k) \end{aligned}$$

- Exploration/Exploitation tradeoff:
    - Unknown environment, need to uncover potential rewards while exploiting known good strategies.

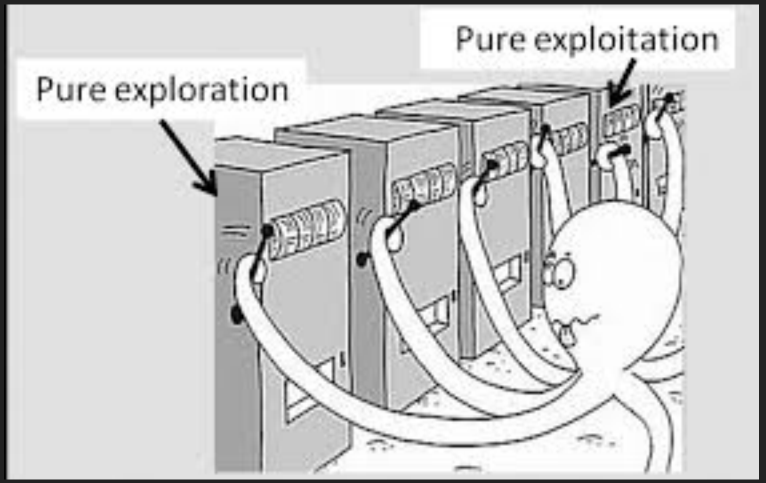


- Exploration/Exploitation tradeoff:
    - Unknown environment, need to uncover potential rewards while exploiting known good strategies.



- Credit Assignment
    - Valid strategies may pay off at later stages.

- Exploration/Exploitation tradeoff:
  - Unknown environment, need to uncover potential rewards while exploiting known good strategies.



- Credit Assignment
  - Valid strategies may pay off at later stages.

- High-dimensional, complex observations.
  - Need to learn good state representations.

- *Model-based:* estimate dynamics $\hat{\mathcal{P}}$ and then solve the resulting optimal control problem (planning and simulation).

- *Model-based:* estimate dynamics $\hat{\mathcal{P}}$ and then solve the resulting optimal control problem (planning and simulation).
- *Approximate Dynamic Programming:* Exploit recurrence structure in optimal policy (Q-learning):
  - Estimation: Given a policy π, compute the *Value* of a state *s:*

$$V^{\pi}(s) := \mathbb{E} \sum_k \gamma^k R(s_k, a_k); s_0 = s, a_k = \pi(s_k).$$

  - Temporal-Difference (TD) learning enforces V^{π} to satisfy rec.
  - Control: Modify π greedily from estimated value functions.

- *Model-based:* estimate dynamics $\hat{\mathcal{P}}$ and then solve the resulting optimal control problem (planning and simulation).
- *Approximate Dynamic Programming:* Exploit recurrence structure in optimal policy (Q-learning):
  - Estimation: Given a policy π, compute the *Value* of a state *s:*

$$V^{\pi}(s) := \mathbb{E}\sum_{k}\gamma^{k}R(s_k,a_k); s_0 = s, a_k = \pi(s_k).$$

  - Temporal-Difference (TD) learning enforces V^{π} to satisfy rec.
  - Control: Modify π greedily from estimated value functions.
- *Policy gradient:* Bypass both model and value, optimize directly over parameters of policy. *Essentially a derivative-free method.*

- State and Action Spaces can be huge (2^{170} for GO) or even infinite and high-dimensional.
- In absence of structural/modeling assumptions, sample complexity will be at least linear with respect to $|\mathcal{S}| \cdot |\mathcal{A}|$.

- State and Action Spaces can be huge (2^{170} for GO) or even infinite and high-dimensional.
- In absence of structural/modeling assumptions, sample complexity will be at least linear with respect to $|\mathcal{S}| \cdot |\mathcal{A}|$.
- Such structure can be incorporated by *function approximation*, ie appropriate parametrisations of value functions, policies, and model dynamics: $\theta \mapsto \{V^{\pi}_{\theta}(s), s \in \mathcal{S}\}$.
  - In generic cases, efficient function approximation will be non-linear.
  - Deep RL: Use neural networks as function approximation.

- State and Action Spaces can be huge (2^{170} for GO) or even infinite and high-dimensional.
- In absence of structural/modeling assumptions, sample complexity will be at least linear with respect to $|\mathcal{S}| \cdot |\mathcal{A}|$.
- Such structure can be incorporated by *function approximation,* ie appropriate parametrisations of value functions, policies, and model dynamics: $\theta \mapsto \{V_\theta^\pi(s), s \in \mathcal{S}\}$.
    - In generic cases, efficient function approximation will be non-linear.
    - Deep RL: Use neural networks as function approximation.
- How to learn with guarantees using nonlinear approx?

- Focus on Value estimation with non-linear function approximation: convergence of non-linear TD learning.
- Interplay between MDP and function approximation geometry: we establish convergence conditions.
- Key geometric properties of function approximation:
  - Homogeneity
  - “Includes” linear functions –> Residual architecture.

- Recall the value function associated to a current policy:

$$V^{\pi}(s) := \mathbb{E} \sum_{k} \gamma^{k} R(s_k, a_k); s_0 = s, a_k = \pi(s_k).$$

- Recall the value function associated to a current policy:

$$V^\pi(s) := \mathbb{E} \sum_k \gamma^k R(s_k, a_k); s_0 = s, a_k = \pi(s_k).$$

- It is the unique solution of the *Bellman equation*:

$$V^\pi(s) = \bar{R}(s) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s)} V^\pi(s') \ \text{, with } \bar{R}(s) = \mathbb{E} R(s, \pi(s)).$$

- Recall the value function associated to a current policy:

$$V^\pi(s) := \mathbb{E}\sum_k \gamma^k R(s_k, a_k); s_0 = s, a_k = \pi(s_k).$$

- It is the unique solution of the *Bellman equation*:

$$V^\pi(s) = \bar{R}(s) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s)} V^\pi(s') \ \ , \text{ with } \bar{R}(s) = \mathbb{E}R(s, \pi(s)).$$

- The most popular algorithm to estimate it is *Temporal-Difference learning* [Sutton, Samuel].
  - Given transition $(s, \bar{R}(s), s')$ and step-size α_k

$$V^{(k+1)}(s) = V^{(k)}(s) + \alpha_k \left(R(s,a) + \gamma V^{(k)}(s') - V^{(k)}(s) \right).$$

- Recall the value function associated to a current policy:

$$V^\pi(s) := \mathbb{E} \sum_k \gamma^k R(s_k, a_k); s_0 = s, a_k = \pi(s_k).$$

- It is the unique solution of the *Bellman equation*:

$$V^\pi(s) = \bar{R}(s) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s)} V^\pi(s') \ \ , \text{with } \bar{R}(s) = \mathbb{E} R(s, \pi(s)).$$

- The most popular algorithm to estimate it is *Temporal-Difference learning* [Sutton, Samuel].
  - Given transition $(s, \bar{R}(s), s')$ and step-size α_k

$$V^{(k+1)}(s) = V^{(k)}(s) + \alpha_k \left(R(s,a) + \gamma V^{(k)}(s') - V^{(k)}(s) \right).$$

  - Under appropriate conditions, we have $V^{(k)} \to V$ as $k \to \infty$.

[Robbins & Munro,50s]

- This algorithm can be seen as taking a stochastic gradient step with respect to the expected squared Bellman error.

- This algorithm can be seen as taking a stochastic gradient step with respect to the expected squared Bellman error.
- Continuous-time interpretation: suppose $\mathcal{P}$ defines an aperiodic, irreducible Markov chain, with stationary distribution μ.

EXPECTED CONTINUOUS-TIME DYNAMICS

- This algorithm can be seen as taking a stochastic gradient step with respect to the expected squared Bellman error.
- Continuous-time interpretation: suppose $\mathcal{P}$ defines an aperiodic, irreducible Markov chain, with stationary distribution μ .
    - as $\alpha_k \to 0$, the *expected* dynamics of TD become

$$\dot{V}(s) = \mu(s)\left(\bar{R}(s) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(.|s)}[V(s')] - V(s)\right)$$

- This algorithm can be seen as taking a stochastic gradient step with respect to the expected squared Bellman error.

- Continuous-time interpretation: suppose $\mathcal{P}$ defines an aperiodic, irreducible Markov chain, with stationary distribution μ.

  - as $\alpha_k \to 0$, the *expected* dynamics of TD become

$$\dot{V}(s) = \mu(s)\left(\bar{R}(s) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(.|s)}[V(s')] - V(s)\right)$$

  - In matrix form, using Bellman equation $V^\pi = \bar{R} + \gamma \mathcal{P} V^\pi$,

$$\dot{V} = D_\mu(\bar{R} + \gamma \mathcal{P} V - V) = -A(V - V^\pi), \text{ with}$$
$$A := D_\mu(I - \gamma \mathcal{P}),\ D_\mu = \mathrm{diag}(\mu).$$

- Fact: A is a "positive-definite", non-symmetric, matrix, ie $x^\top A x > 0 \text{ when } \|x\| > 0.$ [Sutton,'88]

CONSISTENCY OF TD-LEARNING: TABULAR CASE

- Fact: A is a “positive-definite”, non-symmetric, matrix, ie $x^\top Ax > 0 \text{ when } \|x\| > 0.$ [Sutton,’88]

- Consequence: $V(t)$ converges (linearly) to V^π as $t \to \infty$.

- Fact: A is a "positive-definite", non-symmetric, matrix, ie $x^\top A x > 0 \text{ when } \|x\| > 0.$ [Sutton,'88]

- Consequence: $V(t)$ converges (linearly) to V^π as $t \to \infty$.

- However, this algorithm currently computes an independent quantity for each $s \in \mathcal{S}$ (the "tabular" case).

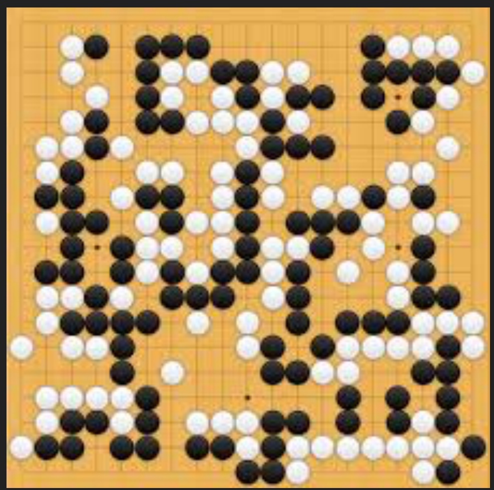
CONSISTENCY OF TD-LEARNING: TABULAR CASE

- Fact: A is a "positive-definite", non-symmetric, matrix, ie $x^\top A x > 0$ when $\|x\| > 0.$ [Sutton,'88]

- Consequence: $V(t)$ converges (linearly) to V^π as $t \to \infty$.

- However, this algorithm currently computes an independent quantity for each $s \in \mathcal{S}$ (the "tabular" case).

- Infeasible in any typical large-scale scenario.



- To overcome such blowup, one considers function approximation. Let $\theta \in \mathbb{R}^d$ and $\theta \mapsto V_\theta \in \mathbb{R}^{|\mathcal{S}|}$ differentiable.

- To overcome such blowup, one considers function approximation. Let $\theta \in \mathbb{R}^d$ and $\theta \mapsto V_\theta \in \mathbb{R}^{|\mathcal{S}|}$ differentiable.
- TD(0) "semi-gradient" algorithm [Sutton]:

$$\theta^{(k+1)} = \theta^{(k)} + \alpha_k \nabla_\theta V_{\theta^{(k)}}(s) \left(\bar{R}(s) + \gamma V_{\theta^{(k)}}(s') - V_{\theta^{(k)}}(s) \right).$$

- To overcome such blowup, one considers function approximation. Let $\theta \in \mathbb{R}^d$ and $\theta \mapsto V_\theta \in \mathbb{R}^{|\mathcal{S}|}$ differentiable.

- TD(0) "semi-gradient" algorithm:

$$\theta^{(k+1)} = \theta^{(k)} + \alpha_k \nabla_\theta V_{\theta^{(k)}}(s) \left(\bar{R}(s) + \gamma V_{\theta^{(k)}}(s') - V_{\theta^{(k)}}(s)\right).$$

- Such update approximates the stochastic gradient of the squared Bellman error $\Delta(\theta) := \|V_\theta - \bar{R} - \gamma \mathcal{P} V_\theta\|^2$

- To overcome such blowup, one considers function approximation. Let $\theta \in \mathbb{R}^d$ and $\theta \mapsto V_\theta \in \mathbb{R}^{|\mathcal{S}|}$ differentiable.

- TD(0) "semi-gradient" algorithm:

$$\theta^{(k+1)} = \theta^{(k)} + \alpha_k \nabla_\theta V_{\theta^{(k)}}(s) \left(\bar{R}(s) + \gamma V_{\theta^{(k)}}(s') - V_{\theta^{(k)}}(s)\right).$$

- Such update approximates the stochastic gradient of the squared Bellman error $\Delta(\theta) := \|V_\theta - \bar{R} - \gamma \mathcal{P} V_\theta\|^2$
  - Problem: an unbiased estimator of $\nabla_\theta \Delta(\theta)$ requires two samples s' from the environment ("*double-sample*" problem):

$$\nabla_\theta \Delta(\theta) := 2(V_\theta - \bar{R} - \gamma \mathcal{P} V_\theta) \cdot (\nabla_\theta V_\theta - \gamma \mathcal{P} \nabla_\theta V_\theta)$$

  - This breaks convergence guarantees of stochastic optimization.

- In continuous time, the corresponding ODE becomes

$$\dot{\theta} = -\nabla V(\theta)^\top A(V(\theta) - V^\pi)$$

- In continuous time, the corresponding ODE becomes

$$\dot{\theta} = -\nabla V(\theta)^\top A(V(\theta) - V^\pi)$$

- Two known regimes where this ODE converges:
    - Linear function approximation [Tsitsiklis & Van Roy'97]:

$$V(\theta) = \Phi\theta \longrightarrow \dot{\theta} = -\Phi^\top A\Phi(\theta - \theta^*), \theta^* = (\Phi^\top A\Phi)^{-1}\Phi^\top AV^\pi.$$

- In continuous time, the corresponding ODE becomes

$$\dot{\theta} = -\nabla V(\theta)^\top A(V(\theta) - V^\pi)$$

- Two known regimes where this ODE converges:
  - Linear function approximation [Tsitsiklis & Van Roy'97]:

$$V(\theta) = \Phi\theta \longrightarrow \dot{\theta} = -\Phi^\top A\Phi(\theta - \theta^*),\, \theta^* = (\Phi^\top A\Phi)^{-1}\Phi^\top A V^\pi.$$

  - Reversible Markov Chain [Ollivier,'18].

$$A = A^\top \longrightarrow \dot{\theta} = -\nabla\|V(\theta) - V^\pi\|_A^2,\, (\langle x, y\rangle_A := x^\top A y).$$

- In continuous time, the corresponding ODE becomes

$$\dot{\theta} = -\nabla V(\theta)^\top A(V(\theta) - V^\pi)$$

- Two known regimes where this ODE converges:
  - Linear function approximation [Tsitsiklis & Van Roy'97]:

$$V(\theta) = \Phi\theta \longrightarrow \dot{\theta} = -\Phi^\top A\Phi(\theta - \theta^*), \theta^* = (\Phi^\top A\Phi)^{-1}\Phi^\top AV^\pi.$$

  - Reversible Markov Chain [Ollivier,'18].

$$A = A^\top \longrightarrow \dot{\theta} = -\nabla\|V(\theta) - V^\pi\|_A^2, (\langle x, y\rangle_A := x^\top Ay).$$

- Alternative Strategies to TD to ensure convergence:
  - "Two-time-scale" algorithms [Dai et al., Borkar et al, Chung et al]

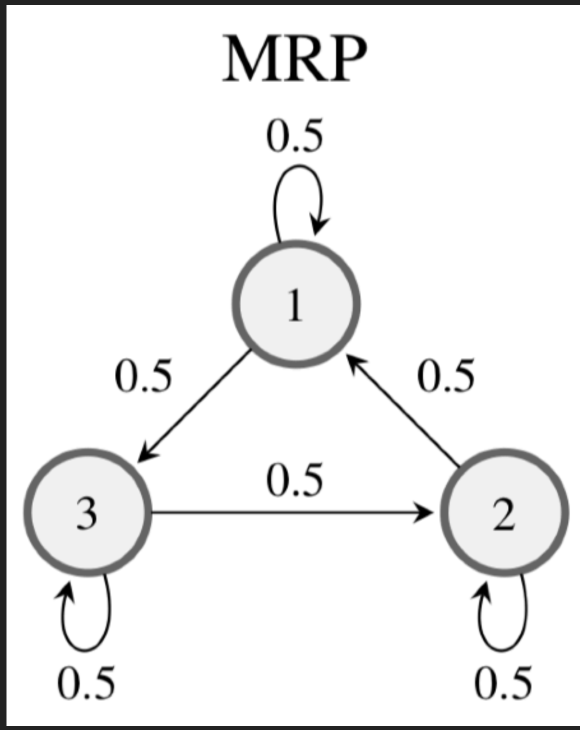
CONSISTENCY IN THE GENERAL CASE?

CONSISTENCY IN THE GENERAL CASE?

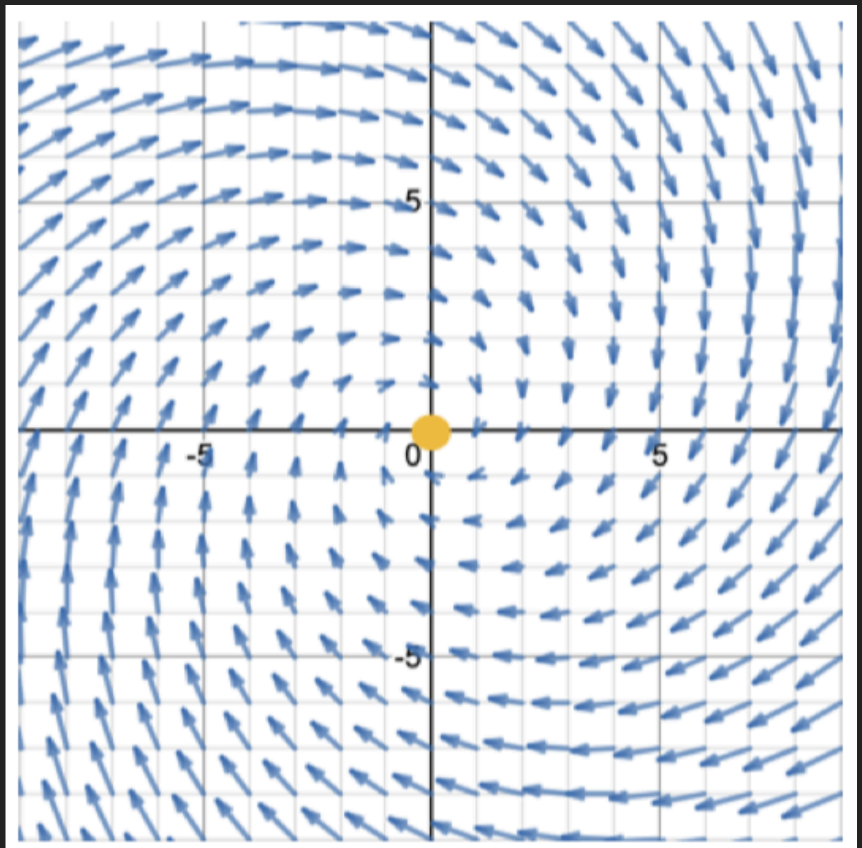
- Convergence is not generic.

CONSISTENCY IN THE GENERAL CASE?

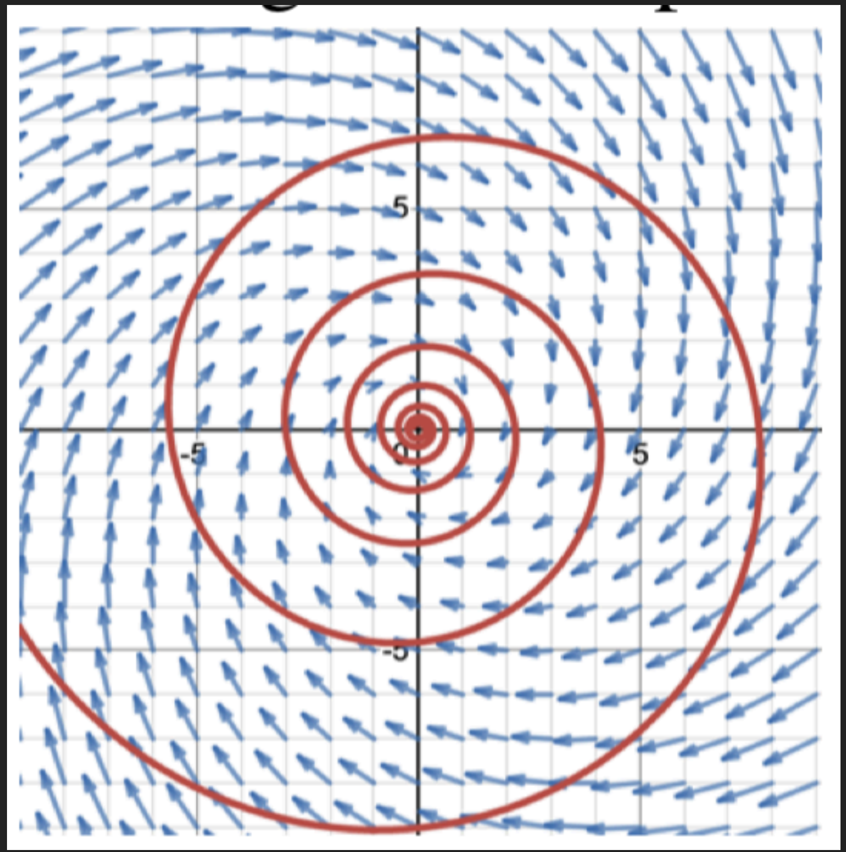
- Convergence is not generic.
- Divergence example from [Tsitsiklis & van Roy]:



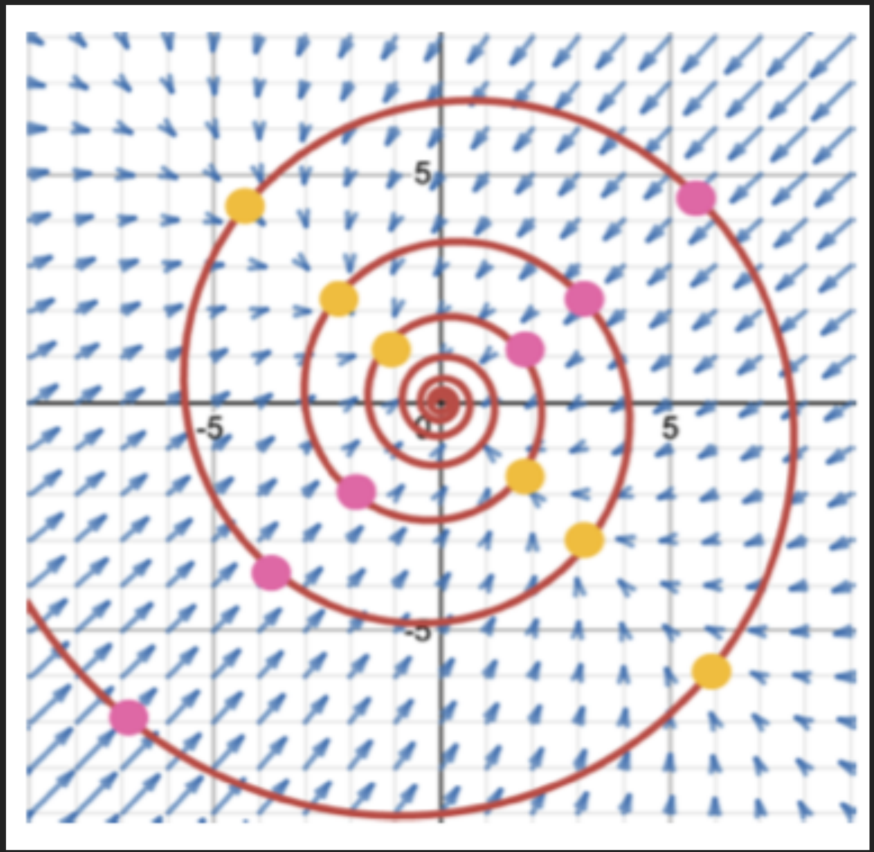
Tabular Case



Nonlinear Function Approximation



Reversible Case



- $f : \mathbb{R}^k \to \mathbb{R}^m$ is *h*-homogenous for $h \in \mathbb{R}$ if

$$\forall x, \forall \alpha > 0\,,\ f(\alpha x) = \alpha^h f(x)$$

- If σ is a homogeneous activation function, then neural networks using σ are also homogeneous (wrt parameters).

- $f : \mathbb{R}^k \to \mathbb{R}^m$ is h-homogenous for $h \in \mathbb{R}$ if

$$\forall x, \forall \alpha > 0\,,\; f(\alpha x) = \alpha^h f(x)$$

- If σ is a homogeneous activation function, then neural networks using σ are also homogeneous (wrt parameters).
- Homogeneous function approximation prevents divergence:

Theorem [BB'19]: Let $\theta \mapsto V(\theta)$ be h-homogeneous and l-Holder. Then for each $\epsilon > 0$ and any initial θ_0, we have

$$\liminf_{t\to\infty} \|V(\theta_t)\|_\mu \leq \frac{\|\bar{R}\|_\mu}{1-\gamma} + \epsilon\,.$$

- $f : \mathbb{R}^k \to \mathbb{R}^m$ is h-homogenous for $h \in \mathbb{R}$ if

$$\forall x, \forall \alpha > 0\,,\; f(\alpha x) = \alpha^h f(x)$$

- If σ is a homogeneous activation function, then neural networks using σ are also homogeneous (wrt parameters).
- Homogeneous function approximation prevents divergence:

Theorem [BB'19]: Let $\theta \mapsto V(\theta)$ be h-homogeneous and l-Holder. Then for each $\epsilon > 0$ and any initial θ_0, we have

$$\liminf_{t \to \infty} \|V(\theta_t)\|_\mu \leq \frac{\|\bar{R}\|_\mu}{1-\gamma} + \epsilon\,.$$

  - In the worst case, homogeneous TD is not worse than using the 0 function baseline: $\|0 - V^*\|_\mu \simeq \frac{\|\bar{R}\|_\mu}{1-\gamma}$
  - Stronger baseline?

- $f : \mathbb{R}^{k_1} \times \mathbb{R}^{k_2} \to \mathbb{R}^m$ is *residual-homogeneous* if $f(x_1, x_2) = \Phi x_1 + g(x_2),\ \text{with } g \text{ homogeneous.}$

- $f : \mathbb{R}^{k_1} \times \mathbb{R}^{k_2} \to \mathbb{R}^m$ is *residual-homogeneous* if $f(x_1, x_2) = \Phi x_1 + g(x_2), \text{ with } g \text{ homogeneous.}$

- With residual-homogeneous, we are provably not worse than using linear models:

Theorem [BB'19]: Let $(\theta_1, \theta_2) \mapsto V(\theta_1, \theta_2)$ be residual-homogeneous and l-Holder. Then for each $\epsilon > 0$ and any initial θ_0, we have

$$\liminf_{t \to \infty} \|V(\theta_t) - V^\pi\|_\mu \le \frac{2\|V^\pi - \Pi_\Phi V^\pi\|_\mu}{1-\gamma} + \epsilon\,.$$

- Similar guarantee as in non-convex optimization using Resnets [Shamir'18].

- $f : \mathbb{R}^{k_1} \times \mathbb{R}^{k_2} \to \mathbb{R}^m$ is *residual-homogeneous* if
  $f(x_1, x_2) = \Phi x_1 + g(x_2)$, with g homogeneous.

- With residual-homogeneous, we are provably not worse than using linear models:

Theorem [BB'19]: Let $(\theta_1, \theta_2) \mapsto V(\theta_1, \theta_2)$ be residual-homogeneous and l-Holder. Then for each $\epsilon > 0$ and any initial θ_0, we have

$$\liminf_{t\to\infty} \|V(\theta_t) - V^\pi\|_\mu \leq \frac{2\|V^\pi - \Pi_\Phi V^\pi\|_\mu}{1-\gamma} + \epsilon\,.$$

  - Similar guarantee as in non-convex optimization using Resnets [Shamir'18].

- Generically, no global convergence. Role of overparametrisation?

- Recall that reversible dynamics result in gradient descent [Ollivier'18]. Can we leverage this property?

- Recall that reversible dynamics result in gradient descent [Ollivier'18]. Can we leverage this property?
- **Definition:** The reversibility coefficient of a Markov Chain $\mathcal{P}$ is

$$\rho(\mathcal{P}) = \inf_{v \in \mathbb{R}^n \setminus \{0\}} \frac{\|S_A v\|^2 + \|Av\|^2}{\|R_A v\|^2}, \text{ with}$$

$$A = D_\mu(I - \gamma\mathcal{P}), S_A = (A + A^\top)/2, R_A = (A - A^\top)/2.$$

- Recall that reversible dynamics result in gradient descent [Ollivier'18]. Can we leverage this property?
- **Definition:** The reversibility coefficient of a Markov Chain $\mathcal{P}$ is

$$\rho(\mathcal{P}) = \inf_{v \in \mathbb{R}^n \setminus \{0\}} \frac{\|S_A v\|^2 + \|Av\|^2}{\|R_A v\|^2}, \text{ with}$$

$$A = D_\mu(I - \gamma\mathcal{P}), S_A = (A + A^\top)/2, R_A = (A - A^\top)/2.$$

- Global convergence with well-conditioned function approximation:

Theorem [BB'19]: Assume that $\kappa(\nabla V(\theta)\nabla V(\theta)^\top) < \rho(\mathcal{P})$ for all θ. Then $V(\theta(t)) \to V^\pi$ as $t \to \infty$.

- Recall that reversible dynamics result in gradient descent [Ollivier'18]. Can we leverage this property?
- **Definition:** The reversibility coefficient of a Markov Chain $\mathcal{P}$ is
$$\rho(\mathcal{P}) = \inf_{v \in \mathbb{R}^n \setminus \{0\}} \frac{\|S_A v\|^2 + \|Av\|^2}{\|R_A v\|^2}, \text{ with}$$
$$A = D_\mu(I - \gamma\mathcal{P}), S_A = (A + A^\top)/2, R_A = (A - A^\top)/2.$$
- Global convergence with well-conditioned function approximation:

Theorem [BB'19]: Assume that $\kappa(\nabla V(\theta) \nabla V(\theta)^\top) < \rho(\mathcal{P})$ for all θ. Then $V(\theta(t)) \to V^\pi$ as $t \to \infty$.

  - Observe that $\kappa(\nabla V(\theta) \nabla V(\theta)^\top) < \infty$ requires $d > |\mathcal{S}|$.
  - Open: underparametrised case with extra smoothness?

BEYOND CONVERGENCE

- Convergence of Value Estimation is the weakest possible guarantee.
- Rate of convergence currently only known for linear models [Bandhari et al.'18]. Non-linear case?

- Convergence of Value Estimation is the weakest possible guarantee.
- Rate of convergence currently only known for linear models [Bandhari et al.'18]. Non-linear case?
- Our current analysis does not measure sample efficiency.

- Convergence of Value Estimation is the weakest possible guarantee.
- Rate of convergence currently only known for linear models [Bandhari et al.'18]. Non-linear case?
- Preliminary analysis: does not measure sample efficiency.
- Sample Complexity of "model-free" RL
  - Tabular Case [Jin et al.'18], [Azar et al.'17], [Brunskill et al.].
  - Linear Function Approximation [Jin et al.'19], [Brandfonbrener et al.'19]
  - Policy Gradients [Argawal et al.'19]

CONCLUSIONS

- Analysis of TD-learning using nonlinear function approximation.
- Interplay between geometry of function approximation (homogeneity, conditioning, linear baseline) and environment (reversibility).
- Learning with guarantees under such conditions.
- Next: Further exploit regularity of reward/environment to reduce overparametrisation.
- Next: From of value estimation to policy update, tighter link between environment and network parametrisation.

Thanks!

Reference:

“Geometric Insights into the convergence of nonlinear TD learning”, D. Brandfonbrener and J. Bruna, *submitted*, https://arxiv.org/abs/1905.12185