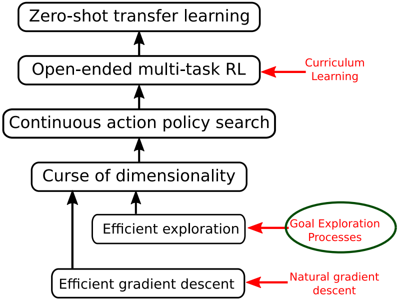
Some work around deep developmental learning

Olivier Sigaud

ISIR, Sorbonne Université
http://people.isir.upmc.fr/sigaud

November 29, 2018

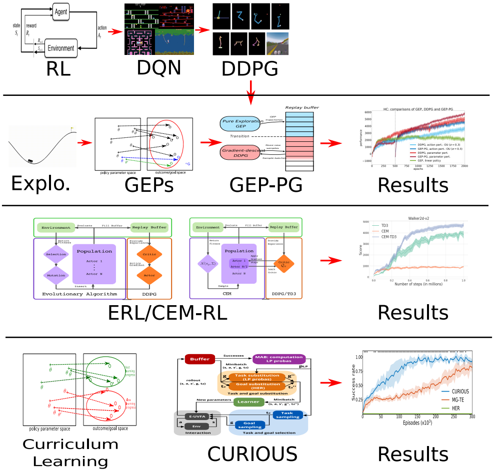
Developmental robotics challenges



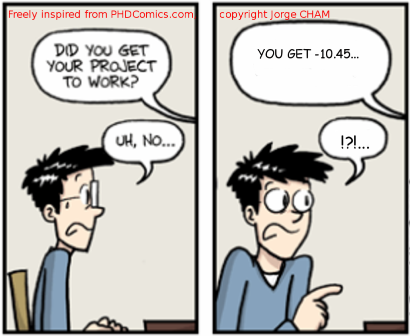
Sigaud, O. & Droniou, A. (2016) Towards deep developmental learning. *IEEE Transactions on Cognitive and Developmental Systems*, 8(2), 99–114

Sigaud, O., Oudeyer P.-Y., et al. *(In preparation)* Intrinsically Motivated Goal Exploration Processes as a central framework for open-ended learning of rich representations.

Outline


RL
DQN
DDPG
Explo.
GEPs
GEP-PG
Results
ERL/CEM-RL
Results
Curriculum Learning
CURIOUS
Results

Reinforcement learning

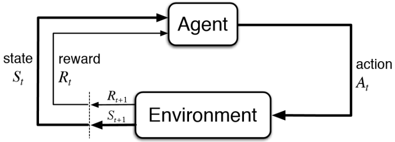


- In SL, the learning signal is the correct answer
- In RL, the learning signal is a scalar
- How good is -10.45?
- Necessity of exploration

The exploration/exploitation trade-off

- Exploring can be (very) harmful
- Shall I exploit what I know or look for a better policy?
- Am I optimal? Shall I keep exploring or stop?
- Decrease the rate of exploration along time

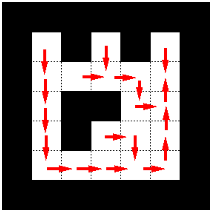
Markov Decision Processes

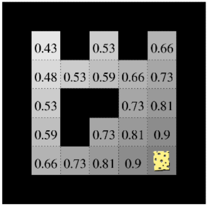


- S: states space
- A: action space
- $T : S \times A \rightarrow \Pi(S)$: transition function
- $r : S \times A \rightarrow \mathbb{R}$: reward function

Sutton, R. S. & Barto, A. G. (1998) *Reinforcement Learning: An Introduction*. MIT Press.

Policy and value functions





state / action	a_0	a_1	a_2	a_3
e_0	0.66	0.88	0.81	0.73
e_1	0.73	0.63	0.9	0.43
e_2	0.73	0.9	0.95	0.73
e_3	0.81	0.9	1.0	0.81
e_4	0.81	1.0	0.81	0.9
e_5	0.9	1.0	0.0	0.9

- Goal: find a policy $\pi : S \to A$ maximizing the agregation of rewards on the long run
- The value function $V^\pi : S \to \mathbb{R}$ records the agregation of reward on the long run for each state (following policy π). It is a vector with one entry per state
- The action value function $Q^\pi : S \times A \to \mathbb{R}$ records the agregation of reward on the long run for doing each action in each state (and then following policy π). It is a matrix with one entry per state and per action

RL Basics

- In dynamic programming, the agent knows the MDP
- In RL it doesn't, it has to explore

- Two approaches:
  - Learn a model of T: model-based (or indirect) reinforcement learning
  - Perform local updates at each step: model-free RL

- Model-free basics:
  - TD error (RPE): $\delta = r_{t+1} + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)$
  - TD(0): $V^{\pi}(s_t) \leftarrow V^{\pi}(s_t) + \alpha[r_{t+1} + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)]$
  - V (or Q) converges when δ converges to 0
  - TD(0) evaluates $V^{\pi}(s)$ for a given policy π, but how shall the agent act?

- Two solutions:
  - Work with $Q^{\pi}(s, a)$ rather than $V^{\pi}(s)$ (SARSA and Q-Learning)
  - Actor-critic methods (simultaneously learn V^{π} and update π)

Q-Learning

- For each observed $(s_t, a_t, r_{t+1}, s_{t+1})$:
  $\delta = r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t)$
- Update rule:
  $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\delta$
- Policy: necessity of exploration (e.g. ϵ-*greedy*)
- Convergence proved given infinite exploration

Watkins, C. J. C. H. (1989). *Learning with Delayed Rewards*. PhD thesis, University of Cambridge, England.

Watkins, C. J. C. H. and Dayan, P. (1992). Q-Learning. *Machine Learning*, 8, 279–292.

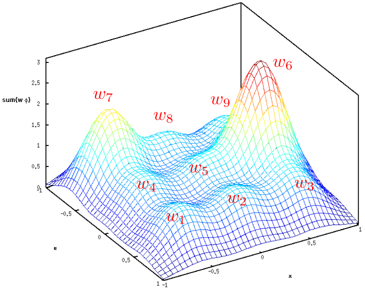
From Q-Learning to Actor-Critic

state / action	a_0	a_1	a_2	a_3
e_0	0.66	0.88*	0.81	0.73
e_1	0.73	0.63	0.9*	0.43
e_2	0.73	0.9	0.95*	0.73
e_3	0.81	0.9	1.0*	0.81
e_4	0.81	1.0*	0.81	0.9
e_5	0.9	1.0*	0.0	0.9

state	chosen action
e_0	a_1
e_1	a_2
e_2	a_2
e_3	a_2
e_4	a_1
e_5	a_1

- In $Q-learning$, given a $Q-Table$, get the max at each step
- Expensive if numerous actions (optimization in continuous action case)
- Storing the max is equivalent to storing the policy
- Update the policy as a function of value updates (only look for the max when decreasing max action)
- Note: looks for local optima, not global ones anymore

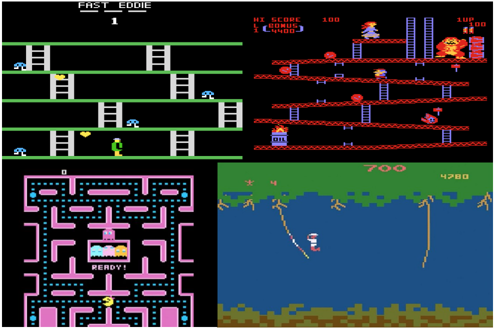
Parametrized representations



- To represent a continuous function, use features and a vector of parameters
- Learning tunes the weights
- Linear architecture: linear combination of features

- A deep neural network is not a linear architectures: deep layer parameters tune the features
- Parametrized representations:
  - In critic-based methods, like DQN: of the critic $Q(s_t, a_t|\theta)$
  - In policy gradient methods: of the policy $\pi(a_t|s_t, \mu)$
  - In actor-critic methods: both

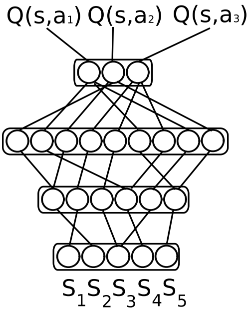
DQN: the breakthrough



- DQN: Atari domain, Nature paper, small discrete actions set
- Learned very different representations with the same tuning

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015) Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.

The Q-network in DQN



- Parametrized representation of the critic $Q(s_t, a_t|\theta)$
- The Q-network is the equivalent of the Q-Table
- Select action by finding the max (as in Q-Learning)
- Limitation: requires one output neuron per action

Learning the Q-function

- Supervised learning: minimize a loss-function, often the squared error w.r.t. the output:

$$L(s,a) = (y^*(s,a) - Q(s,a|\theta))^2 \tag{1}$$

  with backprop on weights θ
- For each sample i, the Q-network should minimize the RPE:

$$\delta_i = r_i + \gamma \max_a Q(s_{i+1}, a|\theta) - Q(s_i, a_i|\theta)$$

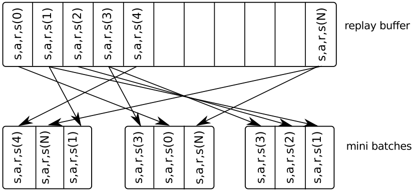
- Thus, given a minibatch of N samples $\{s_i, a_i, r_i, s_{i+1}\}$, compute $y_i = r_i + \gamma \max_a Q(s_{i+1}, a|\theta')$
- And update θ by minimizing the loss function

$$L = 1/N \sum_i (y_i - Q(s_i, a_i|\theta))^2 \tag{2}$$

Trick 1: Stable Target Q-function

- The target $y_i = r_i + \gamma \max_a Q(s_{i+1}, a)|\theta)$ is itself a function of Q
- Thus this is not truly supervised learning, and this is unstable
- Key idea: "periods of supervised learning"
- Compute the loss function from a separate *target network* $Q'(...|\theta')$
- So rather compute $y_i = r_i + \gamma \max_a Q'(s_{i+1}, a|\theta')$
- θ' is updated to θ only each K iterations

Trick 2: Replay buffer shuffling



- In most learning algorithms, samples are assumed independently and identically distributed (iid)
- Obviously, this is not the case of behavioral samples (s_i, a_i, r_i, s_{i+1})
- Idea: put the samples into a buffer, and extract them randomly
- Use training minibatches (make profit of GPU when the input is images)
- The replay buffer management policy is an issue

Lin, L.-J. (1992) Self-Improving Reactive Agents based on Reinforcement Learning, Planning and Teaching. *Machine Learning*, 8(3/4), 293–321

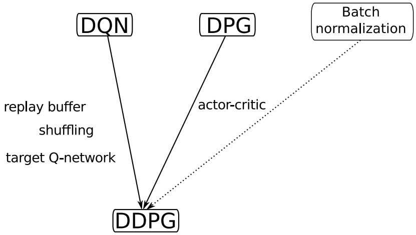
Zhang, S. & Sutton, R. S. (2017) A deeper look at experience replay. *arXiv preprint arXiv:1712.01275*

Deep Deterministic Policy Gradient

- Continuous control with deep reinforcement learning
- Works well on "more than 20" (27-32) domains coded with MuJoCo (Todorov) / TORCS
- End-to-end policies (from pixels to control)

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015) Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971 7/9/15*

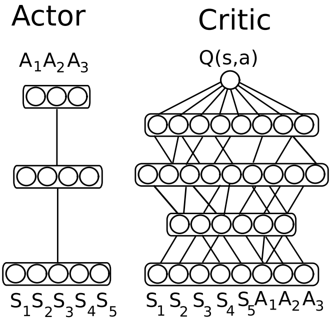
DDPG: ancestors



- Most of the actor-critic theory for continuous problem is for stochastic policies (policy gradient theorem, compatible features, etc.)
- DPG: an efficient gradient computation for deterministic policies, with proof of convergence
- Batch norm: inconclusive studies about importance

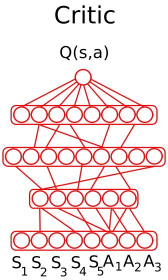
Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014) Deterministic policy gradient algorithms. In *ICML*

General architecture



- Actor parametrized by μ, critic by θ
- All updates based on SGD (as in most deep RL algorithms)

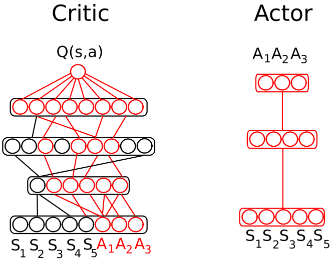
Training the critic



- Same idea as in DQN, but for actor-critic rather than Q-Learning
- Minimize the RPE: $\delta_t = r_t + \gamma Q(s_{t+1}, \pi(s_t)|\theta) - Q(s_t, a_t|\theta)$
- Given a minibatch of N samples $\{s_i, a_i, r_i, s_{i+1}\}$ and a target network Q', compute $y_i = r_i + \gamma Q'(s_{i+1}, \pi(s_{i+1})|\theta')$
- And update θ by minimizing the loss function

$$L = 1/N \sum_i (y_i - Q(s_i, a_i|\theta))^2 \quad (3)$$

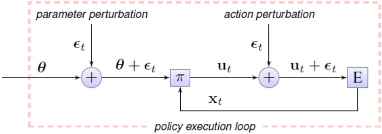
Training the actor



- Deterministic policy gradient theorem: the true policy gradient is
$$\nabla_{\mu}\pi(s,a) = \mathbb{E}_{\rho(s)}[\nabla_a Q(s,a|\theta)\nabla_{\mu}\pi(s|\mu)] \quad (4)$$
- $\nabla_a Q(s,a|\theta)$ is used as error signal to update the actor weights.
- Comes from NFQCA
- $\nabla_a Q(s,a|\theta)$ is a gradient over actions
- $y = f(w.x + b)$ (symmetric roles of weights and inputs)
- Gradient over actions $\sim$ gradient over weights

Hafner, R. & Riedmiller, M. (2011) Reinforcement learning in feedback control. *Machine learning*, 84(1-2), 137–169.

Exploration in DDPG

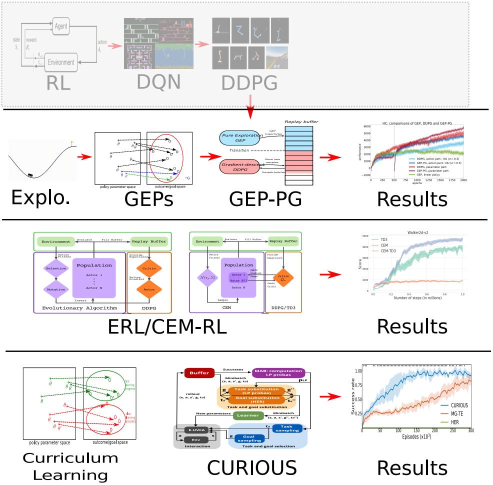


- Action perturbation (versus param. perturbation)
- Adding to the action an Ornstein-Uhlenbenk (correlated) noise process
- Several papers found that using Gaussian noise does not make a difference

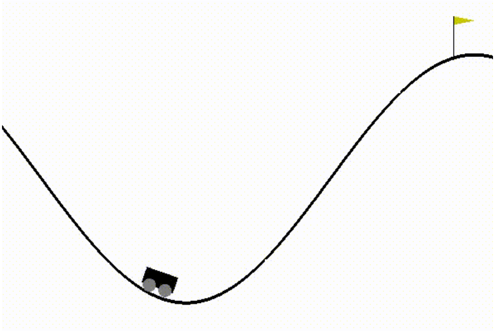
Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., Asfour, T., Abbeel, P., & Andrychowicz, M. (2017) Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., et al. (2017) Noisy networks for exploration. *arXiv preprint arXiv:1706.10295*

Where are we now?

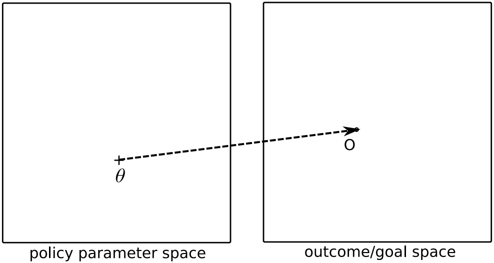


Continuous Mountain Car



- Loss of energy depending on action, reward +100 for reaching the goal
- Deceptive gradient issue: before finding the goal, the agent is driven towards doing nothing
- Spoiler alert: DDPG fails because of poor exploration

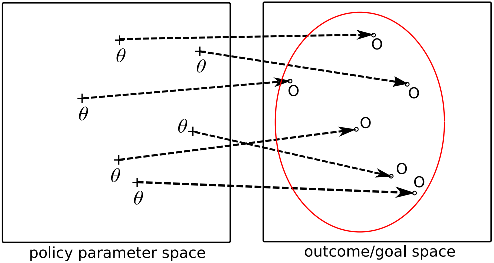
Goal Exploration Processes: algorithm



- Define a relevant outcome space/goal space
- To each policy parameter θ corresponds an outcome O

Pere, A., Forestier, S., Sigaud, O., & Oudeyer, P.-Y. (2018) Unsupervised learning of goal spaces for intrinsically motivated goal exploration. In *International Conference on Learning Representations (ICLR)*, arXiv preprint arXiv:1803.00781

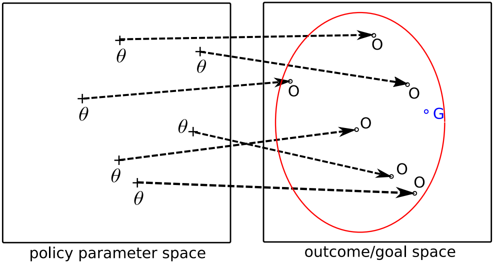
Goal Exploration Processes: algorithm



- Bootstrap phase: draw a few random θ
- Store the resulting (θ, O) pairs into an archive

Pere, A., Forestier, S., Sigaud, O., & Oudeyer, P.-Y. (2018) Unsupervised learning of goal spaces for intrinsically motivated goal exploration. In *International Conference on Learning Representations (ICLR)*, arXiv preprint arXiv:1803.00781

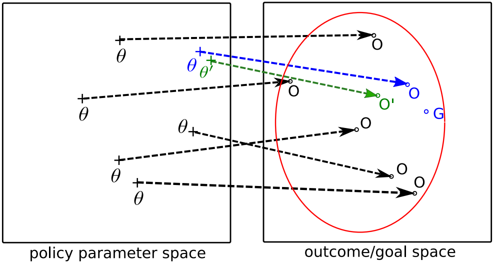
Goal Exploration Processes: algorithm



- Sample a goal at random in the outcome space
- May use the convex hull from bootstrap

Pere, A., Forestier, S., Sigaud, O., & Oudeyer, P.-Y. (2018) Unsupervised learning of goal spaces for intrinsically motivated goal exploration. In *International Conference on Learning Representations (ICLR)*, arXiv preprint arXiv:1803.00781

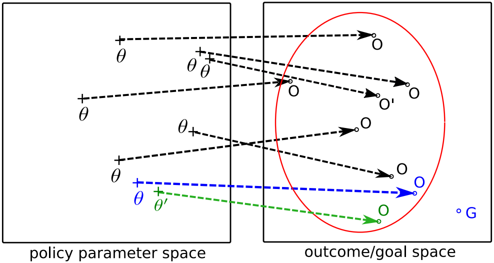
Goal Exploration Processes: algorithm



- Find the nearest neighbor O in archive and select the associated θ
- Perturb the corresponding θ into θ' and get a new outcome O'

Pere, A., Forestier, S., Sigaud, O., & Oudeyer, P.-Y. (2018) Unsupervised learning of goal spaces for intrinsically motivated goal exploration. In *International Conference on Learning Representations (ICLR)*, arXiv preprint arXiv:1803.00781

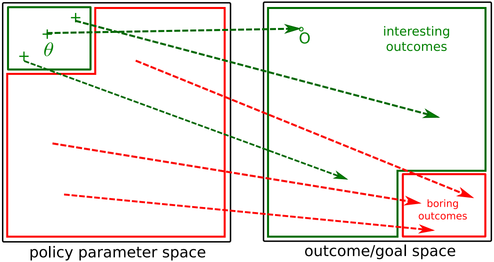
Goal Exploration Processes: algorithm



- One may sample unfeasible goals, favors outcome diversity
- As the archive fills up, performance improves

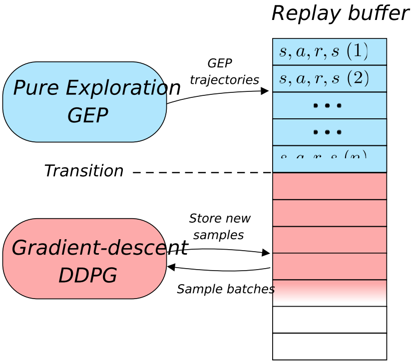
Pere, A., Forestier, S., Sigaud, O., & Oudeyer, P.-Y. (2018) Unsupervised learning of goal spaces for intrinsically motivated goal exploration. In *International Conference on Learning Representations (ICLR)*, arXiv preprint arXiv:1803.00781

Why does GEP work better than random search?



- Very often, few parameter vectors map to interesting outcomes
- The GEP algorithm favors sampling these interesting outcomes
- If the mapping is the identity, similar to random search

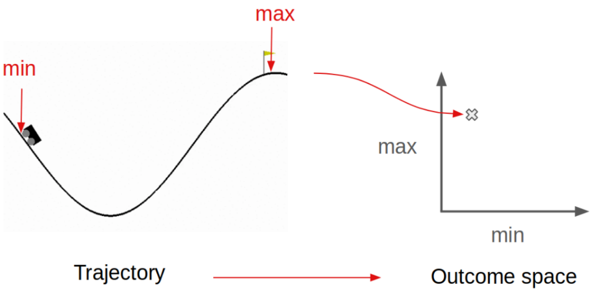
GEP-PG



- Combines GEP for exploration and DDPG for gradient-based search
- Transfer is through the replay buffer
- Strong evaluation methodology (openAI baselines, 20 seeds...)

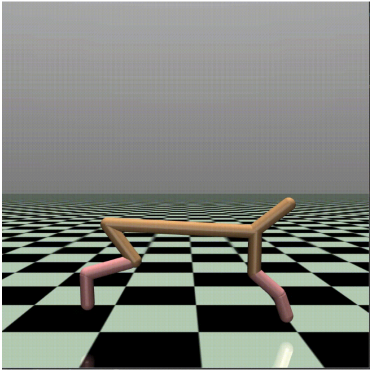
Colas, C., Sigaud, O., & Oudeyer, P.-Y. (2018) GEP-PG: Decoupling exploration and exploitation in deep reinforcement learning algorithms. *arXiv preprint arXiv:1802.05054*

CMC: outcome/goal space



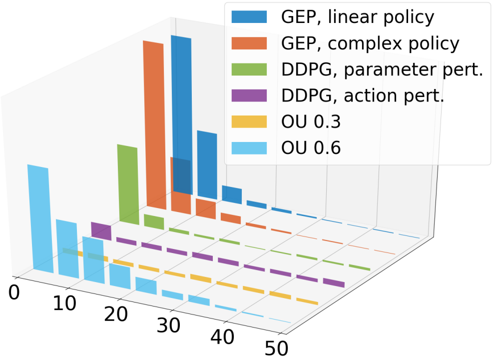
- Defined by hand, informs the search process about relevant dimensions

Half-Cheetah



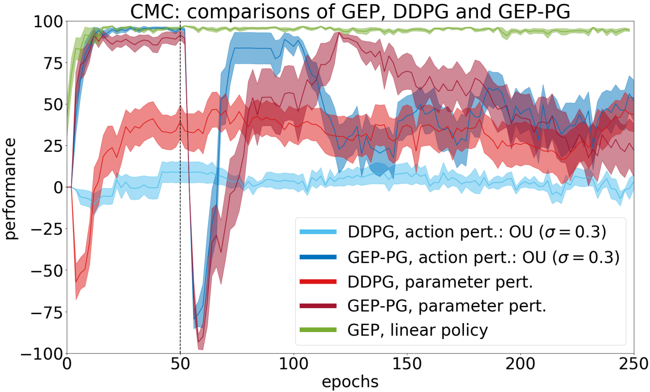
- 17D observation vector, 6D action vector
- Outcome/goal space: average velocity and min height of head

DDPG fails on CMC



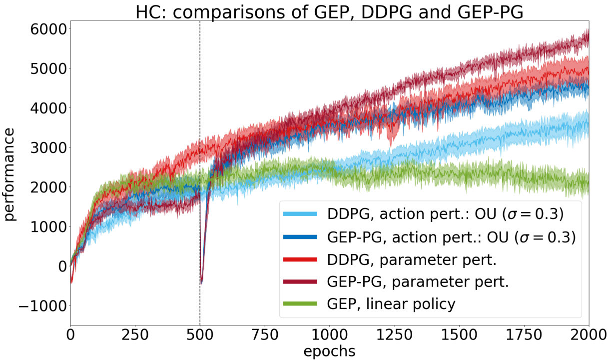
- Key factor: when does it find the reward first?
- DDPG is sensitive to the deceptive gradient issue
- But still better than pure random noise

GEP-PG performs better on CMC



- Efficient exploration solves the deceptive gradient problem
- But isn't the GEP enough?

GEP-PG performs very well on half-cheetah

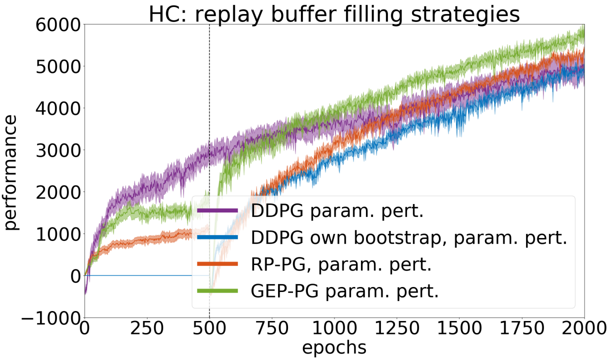


- SOTA results when submitted to ICML (SAC & TD3 do better now)

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018) Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*

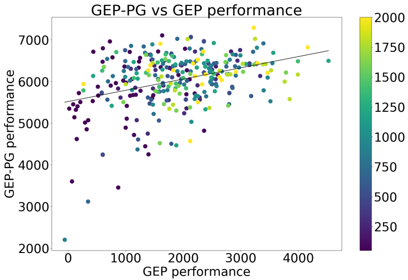
Fujimoto, S., van Hoof, H., & Meger, D. (2018) Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*

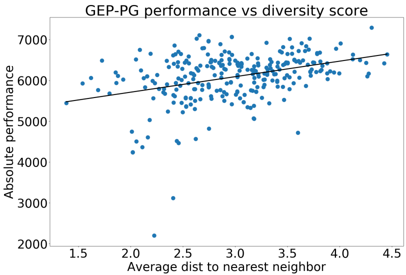
Sanity check



- GEP exploration is better than random exploration
- Random exploration is better than DDPG exploration!

Analyzing GEP-PG performance



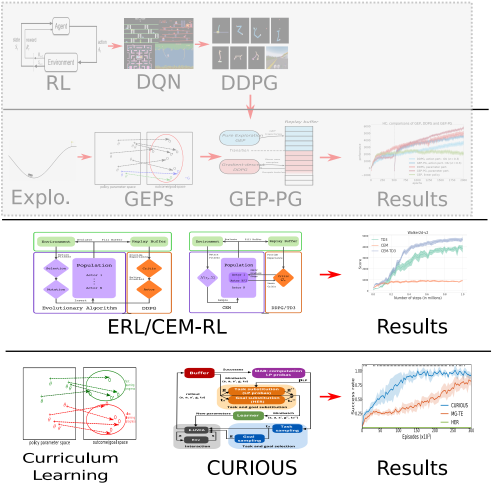


- GEP-PG performance correlates with GEP performance and diversity
- But does not correlate with the size of the GEP buffer
- Thus, the better and the more diverse the replay buffer, the better DDPG

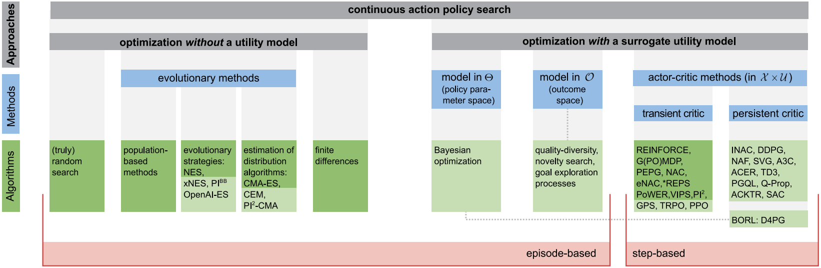
Take home messages

- State-of-the-art deep RL algorithms like DDPG can fail on simple 2D benchmarks like Continuous Mountain Car
- Efficient exploration is needed to improve over deep RL
- GEPs are good at exploring
- They are also more stable: the archive/population does not forget
- Better combinations than GEP-PG can be found (using SAC or TD3, advanced GEPs...)

Where are we now?



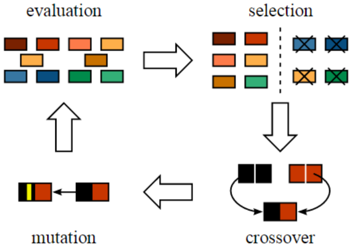
From GEPs to evolutionary methods



- Evo. methods and GEPs are similar (episode-based, population)

Sigaud, O. & Stulp, F. (2018) Policy search in continuous action domains: an overview. *arXiv preprint arXiv:1803.04706*

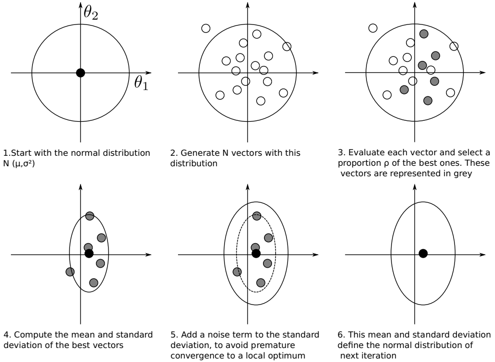
Genetic Algorithms



- Inspired from theory of natural selection
- Many different implementations (here, tournament selection)

Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

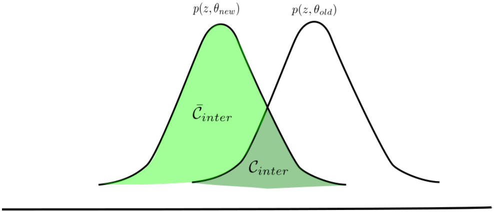
The Cross Entropy Method



- A particular case of evolution strategy

Mannor, S., Rubinstein, R. Y., & Gat, Y. (2003) The cross-entropy method for fast policy search. In *Proceedings of the 20th International Conference on Machine Learning* (pp. 512–519).

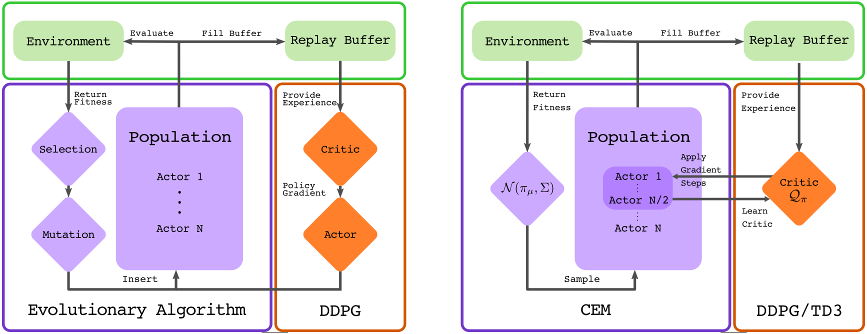
Importance Mixing



- A mechanism to improve sample efficiency

Sun, Y., Wierstra, D., Schaul, T., & Schmidhuber, J. (2009) Efficient natural evolution strategies. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation* (pp. 539–546).: ACM.

Two Combinations

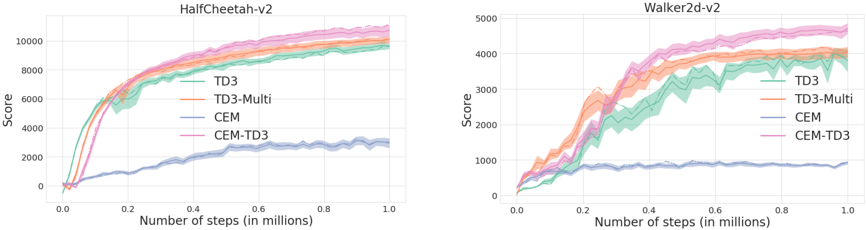


- Combining evolutionary methods and deep RL is an emerging domain

Khadka, S. & Tumer, K. (2018a) Evolution-guided policy gradient in reinforcement learning. In *Neural Information Processing Systems*

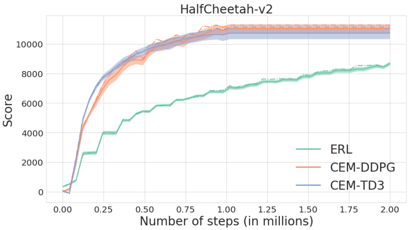
Pourchot, A. & Sigaud, O. (2018) CEM-RL: Combining evolutionary and gradient-based methods for policy search. *arXiv preprint arXiv:1810.01222* (submitted to ICLR)

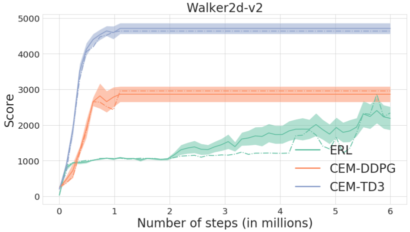
Results (1)



- CEM-TD3 outperforms CEM and TD3

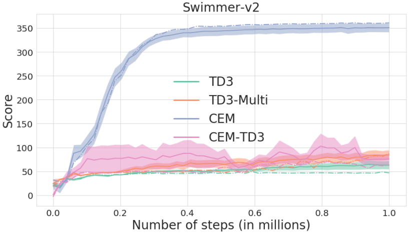
Results (2)

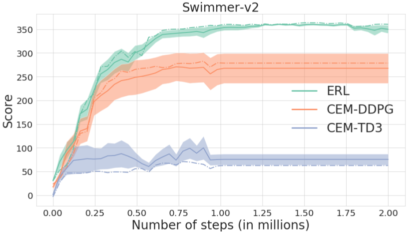




- CEM-TD3 outperforms ERL

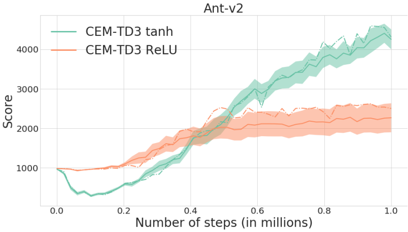
Results (3)

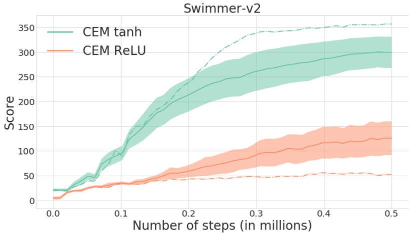




- On swimmer, the best is CEM

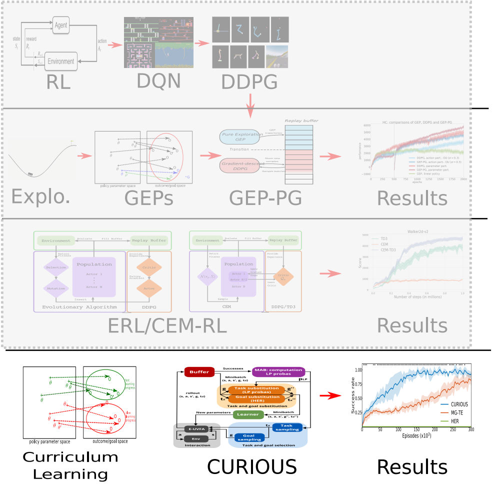
Results (4)



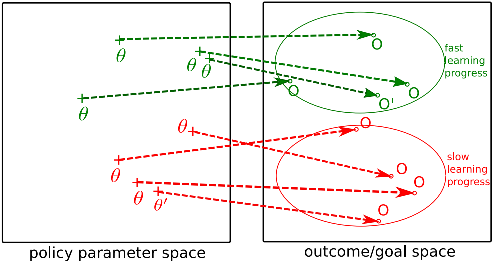


- Changing from ReLu to *tanh* significantly improves performance
- Strong incentive for neural architecture search

Where are we now?



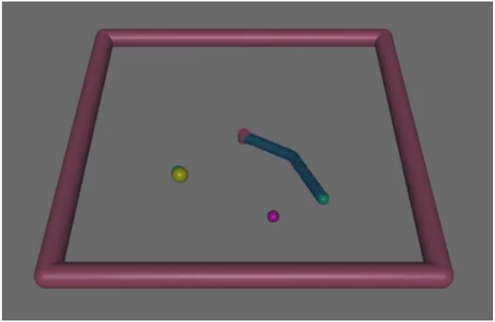
Goal Exploration Processes: curriculum learning



- Sample preferentially regions where learning progress is greater
- Known to improve performance on multitask learning

Baranes, A. & Oudeyer, P.-Y. (2013) Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1), 49–73

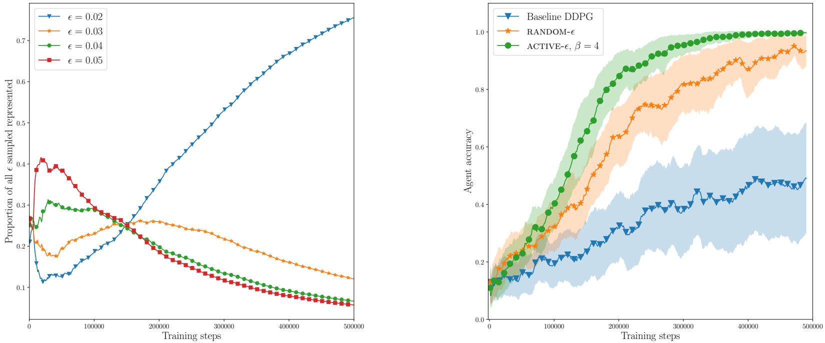
Curriculum based on competence progress



- Experiments with Reacher using various accuracy requirements

Fournier, P., Chetouani, M., Oudeyer, P.-Y., & Sigaud, O. (2018) Accuracy-based curriculum learning in deep reinforcement learning. *arXiv preprint arXiv:1806.09614*

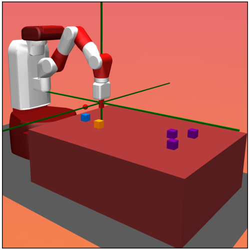
Curriculum performance



- Random sampling of required accuracy is better than always using the strongest requirement
- Sampling based on competence progress is better than random sampling

Fournier, P., Chetouani, M., Oudeyer, P.-Y., & Sigaud, O. (2018) Accuracy-based curriculum learning in deep reinforcement learning. *arXiv preprint arXiv:1806.09614*

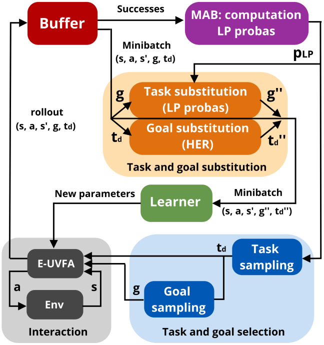
Experimental setup

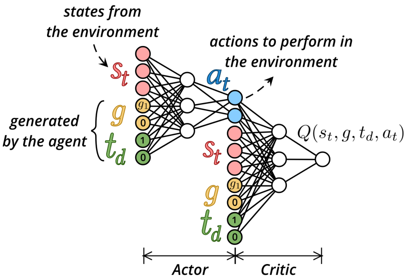


- Move various blocks to various position, stack them etc.
- Combine curriculum learning with Hindsight Experience Replay

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., & Zaremba, W. (2017) Hindsight experience replay. *arXiv preprint arXiv:1707.01495*

A sophisticated architecture

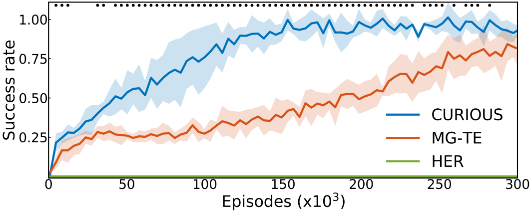


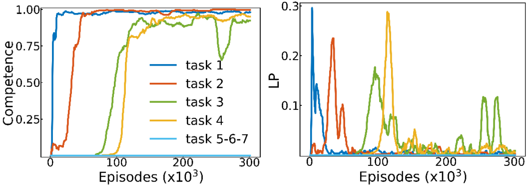


- Dedicated to dealing with tasks and goals

Colas, C., Fournier, P., Sigaud, O., & Oudeyer, P.-Y. (2018) CURIOUS: Intrinsically motivated multi-task, multi-goal reinforcement learning. *arXiv preprint arXiv:1810.06284*

Results





- Generalization over task and goal is better than learning separated tasks

Conclusion

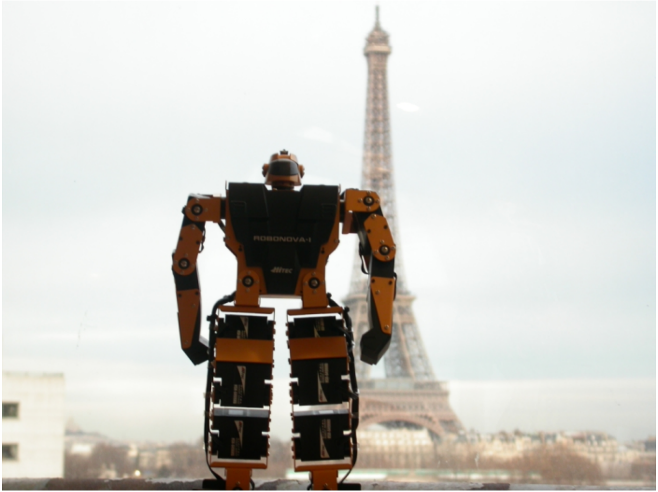
- State-of-the-art deep RL tools still fail on easy benchmarks
- Work needed on exploration, gradient descent, fundamental understanding

- Towards open-ended multi-task learning, zero-shot transfer learning
- Hot topics: curriculum learning, hierarchical RL, model-based RL...

Pierrot, T., Perrin, N., & Sigaud, O. (2018) First-order and second-order variants of the gradient descent: a unified framework. *arXiv preprint arXiv:1810.08102*

ISIR
INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE

Any question?



Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., & Zaremba, W. (2017).
Hindsight experience replay.
arXiv preprint arXiv:1707.01495.

Baranes, A. & Oudeyer, P.-Y. (2013).
Active learning of inverse models with intrinsically motivated goal exploration in robots.
Robotics and Autonomous Systems, 61(1), 49–73.

Colas, C., Fournier, P., Sigaud, O., & Oudeyer, P.-Y. (2018a).
CURIOUS: Intrinsically motivated multi-task, multi-goal reinforcement learning.
arXiv preprint arXiv:1810.06284.

Colas, C., Sigaud, O., & Oudeyer, P.-Y. (2018b).
GEP-PG: Decoupling exploration and exploitation in deep reinforcement learning algorithms.
arXiv preprint arXiv:1802.05054.

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., et al. (2017).
Noisy networks for exploration.
arXiv preprint arXiv:1706.10295.

Fournier, P., Chetouani, M., Oudeyer, P.-Y., & Sigaud, O. (2018).
Accuracy-based curriculum learning in deep reinforcement learning.
arXiv preprint arXiv:1806.09614.

Fujimoto, S., van Hoof, H., & Meger, D. (2018).
Addressing function approximation error in actor-critic methods.
arXiv preprint arXiv:1802.09477.

Goldberg, D. E. (1989).
Genetic Algorithms in Search, Optimization and Machine Learning.
Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018).
Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor.
arXiv preprint arXiv:1801.01290.

Hafner, R. & Riedmiller, M. (2011).
Reinforcement learning in feedback control.
Machine learning, 84(1-2), 137–169.

Khadka, S. & Tumer, K. (2018).
Evolution-guided policy gradient in reinforcement learning.
In *Neural Information Processing Systems.*

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015).
Continuous control with deep reinforcement learning.
arXiv preprint arXiv:1509.02971.

Lin, L.-J. (1992).
Self-Improving Reactive Agents based on Reinforcement Learning, Planning and Teaching.
Machine Learning, 8(3/4), 293–321.

Mannor, S., Rubinstein, R. Y., & Gat, Y. (2003).
The cross-entropy method for fast policy search.
In *Proceedings of the 20th International Conference on Machine Learning* (pp. 512–519).

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015).
Human-level control through deep reinforcement learning.
Nature, 518(7540), 529–533.

Pere, A., Forestier, S., Sigaud, O., & Oudeyer, P.-Y. (2018).
Unsupervised learning of goal spaces for intrinsically motivated goal exploration.
In *International Conference on Learning Representations (ICLR).*
arXiv preprint arXiv:1803.00781.

Pierrot, T., Perrin, N., & Sigaud, O. (2018).
First-order and second-order variants of the gradient descent: a unified framework.
arXiv preprint arXiv:1810.08102.

Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., Asfour, T., Abbeel, P., & Andrychowicz, M. (2017).
Parameter space noise for exploration.
arXiv preprint arXiv:1706.01905.

Pourchot, A. & Sigaud, O. (2018).
Cem-rl: Combining evolutionary and gradient-based methods for policy search.
arXiv preprint arXiv:1810.01222.

Sigaud, O. & Droniou, A. (2016).
Towards deep developmental learning.
IEEE Transactions on Cognitive and Developmental Systems, 8(2), 99–114.

Sigaud, O. & Oudeyer, Pierre-Yves, e. a. (2019).
Intrinsically motivated goal exploration processes as a central framework for open-ended learning of rich representations.
In preparation.

Sigaud, O. & Stulp, F. (2018).
Policy search in continuous action domains: an overview.
arXiv preprint arXiv:1803.04706.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014).
Deterministic policy gradient algorithms.
In *Proceedings of the 30th International Conference in Machine Learning.*

Sun, Y., Wierstra, D., Schaul, T., & Schmidhuber, J. (2009).
Efficient natural evolution strategies.
In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation* (pp. 539–546).: ACM.

Sutton, R. S. & Barto, A. G. (1998).

Reinforcement Learning: An Introduction.
MIT Press.

Watkins, C. J. C. H. (1989).
Learning with Delayed Rewards.
PhD thesis, Psychology Department, University of Cambridge, England.

Watkins, C. J. C. H. & Dayan, P. (1992).
Q-learning.
Machine Learning, 8, 279–292.

Zhang, S. & Sutton, R. S. (2017).
A deeper look at experience replay.
arXiv preprint arXiv:1712.01275.