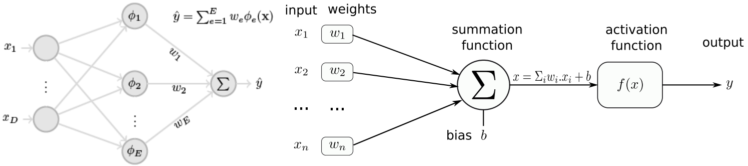
Direct Policy Search vs Reinforcement Learning

Direct Policy Search Methods

Olivier Sigaud

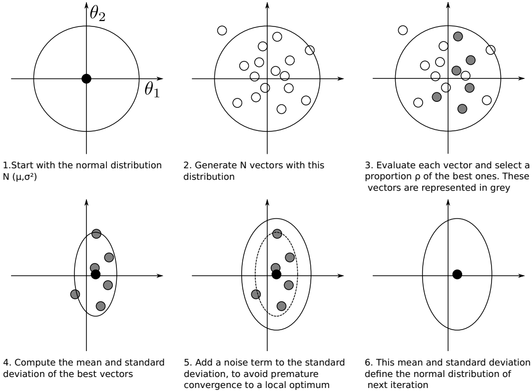
Sorbonne Université
http://people.isir.upmc.fr/sigaud

Direct policy search on neural networks



- The neural network is a controller: input = state, output = action
- The parameters $\boldsymbol{\theta}$ are the weigths and biases of all neurons
- By changing $\boldsymbol{\theta}$, you change the controller $\pi_{\boldsymbol{\theta}}$
- You want to take the best actions in all states to optimize $J(\boldsymbol{\theta})$
- Key feature in the direct policy search problem: $\boldsymbol{\theta}$ is often large
- Two families of approaches:
  - Cross Entropy Method (CEM), CMA-ES...
  - Finite difference methods

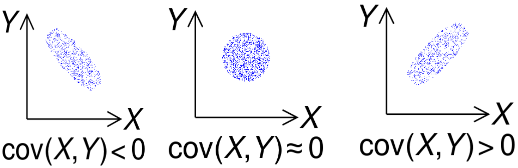
CEM for policy search: overview



- Here, an example where $\boldsymbol{\theta}$ is 2D

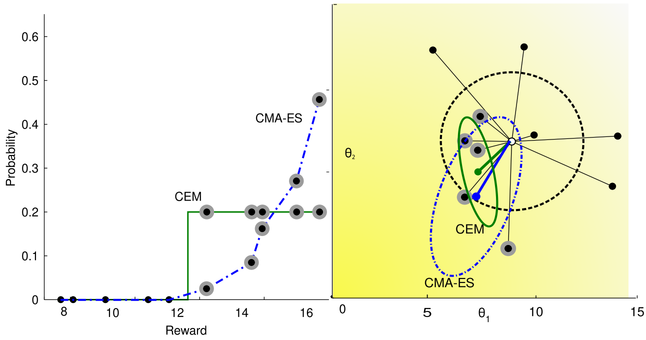
Marin, and Sigaud, O. (2012) Towards fast and adaptive optimal control policies for robots: A direct policy search approach, *Proceedings conference Robotica*, pp. 21-26

The covariance matrix



- The covariance is a measure of the joint variability of two random variables (wikipedia).
- The covariance matrix is a square matrix giving the covariance between each pair of elements of a given random vector (wikipedia).
- The ellipsoid illustrates the range of likely values for the random variables
- In CEM, the random variables are single parameters of vectors $\boldsymbol{\theta}$
- The covariance matrix is in $\boldsymbol{\theta} \times \boldsymbol{\theta}$, too large if $\boldsymbol{\theta}$ is large
- Just use the diagonal

CMA-ES vs CEM



- The stronger the yellow, the higher the return
- CMA-ES uses many additional tricks

Hansen, N. & Auger, A. (2011) CMA-ES: evolution strategies and covariance matrix adaptation. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation* (pp. 991–1010)

Finite difference methods

- We consider we do not know the derivative $\nabla J(\boldsymbol{\theta})$
- Intuition: for a small enough ϵ, $\hat{\nabla} J(\boldsymbol{\theta}) \sim \frac{J(\boldsymbol{\theta}+\epsilon)-J(\boldsymbol{\theta}-\epsilon)}{2\epsilon}$
- Sample ϵ from a weighted Gaussian $\sigma\mathcal{N}(0, \mathbf{I})$
- Use a Monte Carlo approach to estimate $J(\boldsymbol{\theta}+\epsilon), J(\boldsymbol{\theta}-\epsilon)$
- Then use the estimated derivative to perform gradient descent
- More formal account in Choromanski: Gaussian smoothing objective

Choromanski, K., Rowland, M., Sindhwani, V., Turner, R., and Weller, A. Structured evolution with compact architectures for scalable policy optimization. In *International Conference on Machine Learning*, pp.970–978. PMLR, 2018

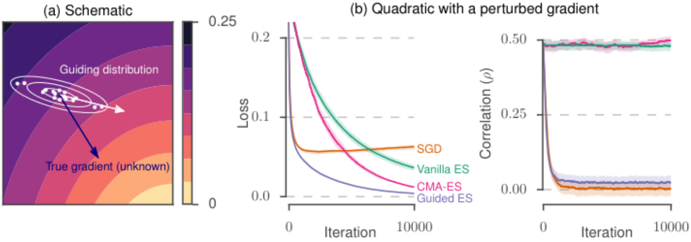
Finite difference variants

- Rather sample ϵ from $\mathcal{N}(0, \mathbf{I})$ and show σ
- Three ES estimators:
  1. Vanilla (P samples): $\hat{\nabla}_N^V J_\sigma(\boldsymbol{\theta}) = \frac{1}{N\sigma} \sum_{i=1}^N J(\boldsymbol{\theta} + \sigma\epsilon_i)\epsilon_i$
  2. Antithetic (2P samples):
  $\hat{\nabla}_N^{AT} J_\sigma(\boldsymbol{\theta}) = \frac{1}{N\sigma} \sum_{i=1}^N (J(\boldsymbol{\theta} + \sigma\epsilon_i) - J(\boldsymbol{\theta} - \sigma\epsilon_i))\epsilon_i$
  3. Forward finite-difference (P+1 samples):
  $\hat{\nabla}_N^{FFD} J_\sigma(\boldsymbol{\theta}) = \frac{1}{N\sigma} \sum_{i=1}^N (J(\boldsymbol{\theta} + \sigma\epsilon_i) - J(\boldsymbol{\theta}))\epsilon_i$
- In OpenAI ES, the gradient is estimated with 1. then applied with Adam
- Augmented Random Search (Mania et al., 2018) compares the variants

Tim Salimans, Jonathan Ho, Xi Chen, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017

Mania, H., Guy, A., and Recht, B. (2018) Simple random search of static linear policies is competitive for reinforcement learning. *Advances in Neural Information Processing Systems*, 31

Improvements over OpenAI ES



- Guided ES: One can improve efficiency by adding extra information about the gradient (Maheswaranathan et al., 2018)
- Suggests combinations with RL
- Trust-Region ES: One can improve exploration, by drawing better-than-Gaussian directions (Liu et al., 2019)

Maheswaranathan, N., Metz, L., Tucker, G., and Sohl-Dickstein, J. Guided evolutionary strategies: escaping the curse of dimensionality in random search. *arXiv preprint arXiv:1806.10230*, 2018

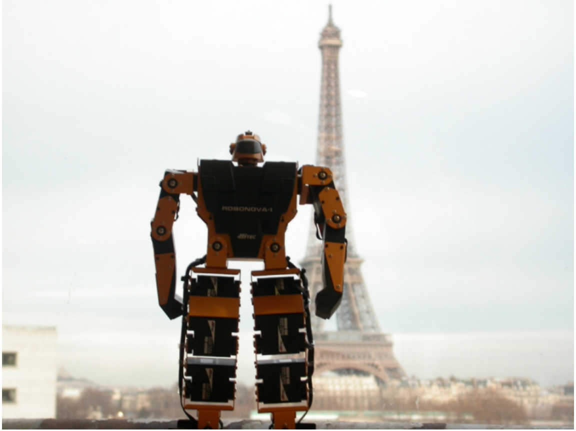
Liu, G., Zhao, L., Yang, F., Bian, J., Qin, T., Yu, N., and Liu, T.-Y. Trust Region Evolution Strategies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4352–4359, 2019

Finite difference methods vs CEM (and CMA-ES)

- Finite difference methods are gradient-based direct policy search methods.
- They are derivative-free, but a backprop step is applied, using an approximate gradient (OpenAI ES uses Adam!)
- CEM and CMA-ES sample policies around the current one.
- They do not compute a *variation* to the current policy nor do they apply a gradient
- The new policy is a weighted barycenter of sampled policies
- In CEM and CMA-ES, directions are not sampled from $\mathcal{N}(0, \mathbf{I})$, but from an updated covariance matrix $\mathcal{N}(\boldsymbol{\theta}, \Sigma)$
- Open questions:
  - do FD methods scale better than CEM-like methods?
  - does Adam optimization compensate for not using the covariance matrix?
- A lot of such questions are still open in direct policy search methods
- Research on advanced derivative-free methods is active

Berahas, A. S., Cao, L., Choromanski, K., and Scheinberg, K. (2022) A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *Foundations of Computational Mathematics*, 22(2):507–560

Any question?



Send mail to: `Olivier.Sigaud@isir.upmc.fr`

Berahas, A. S., Cao, L., Choromanski, K., and Scheinberg, K. (2022).
A theoretical and empirical comparison of gradient approximations in derivative-free optimization.
Foundations of Computational Mathematics, 22(2):507–560.

Choromanski, K., Rowland, M., Sindhwani, V., Turner, R., and Weller, A. (2018).
Structured evolution with compact architectures for scalable policy optimization.
In *International Conference on Machine Learning*, pages 970–978. PMLR.

Hansen, N. and Auger, A. (2011).
Cma-es: evolution strategies and covariance matrix adaptation.
In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, pages 991–1010.

Liu, G., Zhao, L., Yang, F., Bian, J., Qin, T., Yu, N., and Liu, T.-Y. (2019).
Trust region evolution strategies.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4352–4359.

Maheswaranathan, N., Metz, L., Tucker, G., and Sohl-Dickstein, J. (2018).
Guided evolutionary strategies: escaping the curse of dimensionality in random search.
arXiv preprint arXiv:1806.10230.

Mania, H., Guy, A., and Recht, B. (2018).
Simple random search of static linear policies is competitive for reinforcement learning.
Advances in Neural Information Processing Systems, 31.

Marin, D. and Sigaud, O. (2012).
Towards fast and adaptive optimal control policies for robots: A direct policy search approach.
In *Proceedings Robotica*, pages 21–26, Guimaraes, Portugal.

Salimans, T., Ho, J., Chen, X., and Sutskever, I. (2017).
Evolution strategies as a scalable alternative to reinforcement learning.
arXiv preprint arXiv:1703.03864.