Variable-metric Evolution Strategies for Direct Policy Search

Verena Heidrich-Meisner VERENA.HEIDRICH-MEISNER@NEUROINFORMATIK.RUB.DE
Christian Igel CHRISTIAN.IGEL@NEUROINFORMATIK.RUB.DE
Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany

1. Introduction

We promote the covariance matrix evolution strategy (CMA-ES, Hansen et al., 2003; Hansen, 2006; Suttorp et al., 2009) for direct policy search (an approach also referred to as *Neuroevolution Strategies* when applied to neural network policies). The algorithm gives striking results on reinforcement learning (RL) benchmark problems (Gomez et al., 2008; Heidrich-Meisner and Igel 2008a; 2008c; in press), see Table 1 for an example.

2. The CMA-ES for RL

Evolution strategies are random, derivative-free search methods (Beyer, 2007). They iteratively sample a set of candidate solutions from a probability distribution over the search space (i.e., the space of policies), evaluate these potential solutions, and construct a new probability distribution over the search space based on the gathered information. In evolution strategies, this search distribution is parametrized by a set of μ candidate solutions (*parents*) and by parameters of the variation operators that are used to create new policies (*offspring*) from the μ candidate policies.

In each iteration k of the CMA-ES, the lth candidate policy with parameters $\boldsymbol{x}_l^{(k+1)} \in \mathbb{R}^n$ ($l \in \{1, \ldots, \lambda\}$) is generated by multi-variate *Gaussian mutation* and *weighted global intermediate recombination*:

$$\boldsymbol{x}_l^{(k+1)} = \boldsymbol{m}^{(k)} + \sigma^{(k)} \boldsymbol{z}_l^{(k)}$$

The *mutation* $\boldsymbol{z}_l^{(k)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}^{(k)})$ is the realization of a normally distributed random vector with zero mean and covariance matrix $\boldsymbol{C}^{(k)}$. The *recombination* is given by the weighted mean $\boldsymbol{m}^{(k)} = \sum_{l=1}^{\mu} w_l \boldsymbol{x}_{l:\lambda}^{(k)}$, where $\boldsymbol{x}_{l:\lambda}^{(k)}$ denotes the lth best individual among $\boldsymbol{x}_1^{(k)}, \ldots, \boldsymbol{x}_\lambda^{(k)}$. This corresponds to rank-based selection, in which the best μ of the λ offspring form the next parent population. A common choice for the recombination weights is $w_l \propto \ln(\mu+1) - \ln(l)$, $\|\boldsymbol{w}\|_1 = 1$. The quality of an individual $\boldsymbol{x}_l^{(k+1)}$ is determined by evaluating the corresponding policy. This evaluation is based on the Monte Carlo return of one or several episodes using the policy with parameters $\boldsymbol{x}_l^{(k+1)}$.

The CMA-ES is a variable-metric algorithm adapting both the n-dimensional *covariance matrix* $\boldsymbol{C}^{(k)}$ of the normal mutation distribution as well as the *global step size* $\sigma^{(k)} \in \mathbb{R}^+$. The covariance matrix update has two parts, the rank-1 update considering the change of the population mean over time and the rank-μ update considering the successful variations in the last iteration. For example, the rank-1 update is based on a low-pass filtered *evolution path* $\boldsymbol{p}^{(k)}$ of successful steps

$$\boldsymbol{p}_c^{(k+1)} \leftarrow c_1 \boldsymbol{p}_c^{(k)} + c_2 \frac{\boldsymbol{m}^{(k+1)} - \boldsymbol{m}^{(k)}}{\sigma^{(k)}}$$

and aims at changing $\boldsymbol{C}^{(k)}$ to make steps in the promising direction $\boldsymbol{p}^{(k+1)}$ more likely by morphing the covariance towards $\left[\boldsymbol{p}_c^{(k+1)}\right] \left[\boldsymbol{p}_c^{(k+1)}\right]^{\mathrm{T}}$. For details of the CMA-ES (the choice of the constants $c_1, c_2 \in \mathbb{R}^+$, the rank-$\mu$ update, the update of σ, etc.) we refer to the original articles by Hansen et al. (Hansen et al., 2003; Hansen, 2006).

3. Why CMA-ES for RL?

Employing the CMA-ES for RL

1. allows for direct search in policy space and is not restricted to optimizing policies "indirectly" by adapting state-value or state-action-value functions,

2. is straightforward to apply and robust w.r.t. tuning of hyperparameters (e.g., compared to temporal difference learning algorithms or policy gradient methods),

3. is based on ranking policies, which is less susceptible to uncertainty and noise (e.g., due to random rewards and transitions, random initialization, and noisy state observations) compared to estimating a value function or a gradient of a performance measure w.r.t. policy parameters,

method	reward function	
	standard	damping
RWG	415,209	1,232,296
CE	–	(840,000)
SANE	262,700	451,612
CNE	76,906	87,623
ESP	7,374	26,342
NEAT	–	6,929
RPG	(5,649)	–
CoSyNE	1,249	3,416
CMA-ES	860	1,141

Table 1. Mean number of episodes required for different RL algorithms to solve the partially observable double pole balancing problem (i.e., pole and cart velocities are not observed) using the standard performance function and using the damping performance function, respectively (see Gruau et al., 1996). The CMA-ES adapts standard recurrent neural network representing policies. The Neuroevolution Strategy results are taken from the paper by Heidrich-Meisner and Igel (in press) and the other results were compiled by Gomez et al. (2008). The abbreviation RWG stands for Random Weight Guessing, PGRL for Policy Gradient RL, and RPG for Recurrent Policy Gradients. The other methods are evolutionary approaches, CNE stands for Conventional Neuroevolution, ESP for Enforced Sub-Population, NEAT for NeuroEvolution of Augmenting Topologies , and CoSyNE for Cooperative Synapse Neuroevolution (see Gomez et al., 2008; Heidrich-Meisner & Igel, in press, for references).

4. allows for simple uncertainty handling strategies to dynamically adjust the overall number and the distribution of roll-outs for evaluating policies in each iteration in order to learn efficiently in the presence of uncertainty and noise (Heidrich-Meisner and Igel 2008b; 2009),

5. is a variable-metric algorithm learning an appropriate coordinate system for a specific problem (by means of adapting the covariance matrix and thereby considering correlations between parameters),

6. can be applied if the function approximators are non-differentiable, whereas many other methods require a differentiable structure, and

7. extracts a search direction, stored in the evolution path $\boldsymbol{p}_c^{(k)}$, from the scalar reward signals.

Arguably, the main drawback of the CMA-ES for RL in its current form is that the CMA-ES does not exploit intermediate rewards, just final Monte Carlo returns. This currently restricts the applicability of the CMA-ES to episodic tasks and may cause problems for tasks with long episodes. Addressing these issues will be part of out future research.

References

Beyer, H.-G. (2007). Evolution strategies. *Scholarpedia*, *2*, 1965.

Gomez, F., Schmidhuber, J., & Miikkulainen, R. (2008). Accelerated neural evolution through cooperatively coevolved synapses. *Journal of Machine Learning Research*, *9*, 937–965.

Gruau, F., Whitley, D., & Pyeatt, L. (1996). A comparison between cellular encoding and direct encoding for genetic neural networks. *Genetic Programming 1996: Proceedings of the First Annual Conference* (pp. 81–89). MIT Press.

Hansen, N. (2006). The CMA evolution strategy: A comparing review. In *Towards a new evolutionary computation. advances on estimation of distribution algorithms*, 75–102. Springer-Verlag.

Hansen, N., Müller, S. D., & Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, *11*, 1–18.

Heidrich-Meisner, V., & Igel, C. Neuroevolution strategies for episodic reinforcement learning. *Journal of Algorithms*. In press.

Heidrich-Meisner, V., & Igel, C. (2008a). Similarities and differences between policy gradient methods and evolution strategies. *16th European Symposium on Artificial Neural Networks (ESANN)* (pp. 149–154). Evere, Belgium: d-side publications.

Heidrich-Meisner, V., & Igel, C. (2008b). Uncertainty handling in evolutionary direct policy search. In Y. Engel, M. Ghavamzadeh, P. Poupart and S. Mannor (Eds.), *NIPS-08 workshop on model uncertainty and risk in reinforcement learning*.

Heidrich-Meisner, V., & Igel, C. (2008c). Variable metric reinforcement learning methods applied to the noisy mountain car problem. *European Workshop on Reinforcement Learning (EWRL 2008)* (pp. 136–150). Springer-Verlag.

Heidrich-Meisner, V., & Igel, C. (2009). Hoeffding and bernstein races for selecting policies in evolutionary direct policy search. *Proceedings of the 26th International Conference on Machine Learning (ICML 2009)*.

Suttorp, T., Hansen, N., & Igel, C. (2009). Efficient covariance matrix update for variable metric evolution strategies. *Machine Learning*, *75*, 167–197.