How to Evolve Gradient Descent into Evolution Strategies and CMA-ES

Nikolaus Hansen
Inria
CMAP, CNRS, Ecole Polytechnique, Institut Polytechnique de Paris, France

Presented in Delft 2019
Outline

• Preliminaries / Context

• From gradient descent to evolution strategies

• A second order (variable metric) evolution strategy: CMA-ES
Context: Objective

minimize an objective function

\[ f : \mathbb{R}^n \rightarrow \mathbb{R}, \, x \mapsto f(x) \]

• in a black-box / direct search scenario
  ✓ no first order information (i.e. no gradient)
  ✓ unknown structure

• in theory: convergence to the global optimum

• in practice: find a good solution iteratively as quickly as possible
Section Through a 5-Dimensional Rugged Landscape

\[ f : \mathbb{R}^n \rightarrow \mathbb{R}, \ x \mapsto f(x), \ n = 5 \]

How can we modify gradient descent to solve this problem?
The Optimization/Search Algorithm
From Gradient Descent to Evolution Strategies
Basic Approach: Gradient Descent

The gradient is the local direction of the maximal $f$ increase.

- $f(x) = \text{const}$
- tangent space
- current iterate
- $
\nabla f(x)$
- $\mathbb{R}^2$
- $e_1$
- $e_2$
Basic Approach: Gradient Descent

The gradient is the local direction of the maximal $f$ increase

$$f(x) = \text{const}$$

$$-\nabla f(x)$$

$tangent space$

$\mathbb{R}^2$

$e_1$

$e_2$

optimal gradient step length

current iterate

Nikolaus Hansen, Inria, IP Paris
Basic Approach: Gradient Descent

The gradient is the local direction of the maximal $f$ increase

$$\nabla f(x) = -\sum_{i=1}^{n} w_i e_i \quad -w_i = \lim_{\delta \to 0} \frac{f(x + \delta e_i) - f(x)}{\delta}$$

$$x \leftarrow x - \sigma \nabla f(x) = x + \sigma \sum_{i=1}^{n} w_i e_i$$
Basic Approach: Gradient Descent

The \textit{gradient} is the local direction of the maximal $f$ increase

$$\nabla f(x) = -\sum_{i=1}^{n} w_i e_i \quad -w_i = \lim_{\delta \to 0} \frac{f(x + \delta e_i) - f(x)}{\delta}$$

$$x \leftarrow x - \sigma \nabla f(x)$$

$$= x + \sigma \sum_{i=1}^{n} w_i e_i$$
Basic Approach: Gradient Descent

The **gradient** is the local direction of the maximal $f$ increase

$$
\nabla f(x) \approx - \sum_{i=1}^{n} w_i e_i \quad -w_i = \lim_{\delta \to 0} \frac{f(x + \delta e_i) - f(x)}{\delta}
$$

small test step

$$
x \leftarrow x - \sigma \nabla f(x)
$$

$$
\approx x + \sigma \sum_{i=1}^{n} w_i e_i
$$
Now we do very few changes leading to a very different algorithm (with very different behavior)
Basic Approach: Gradient Descent

The *gradient* is the local direction of the maximal $f$ increase.

\[
\nabla f(x) \approx - \sum_{i=1}^{n} w_i e_i \quad -w_i = \lim_{\delta \to 0} \frac{f(x + \delta e_i) - f(x)}{\delta}
\]

\[
x \leftarrow x - \sigma \nabla f(x)
\]

\[
x + \sigma \sum_{i=1}^{n} w_i e_i
\]
Basic Approach: Approximated Gradient Descent

We modify the gradient equation: (1) use $y_i$ instead of $e_i$

\[
\nabla f(x) \approx - \sum_{i=1}^{m} w_i y_i \quad -w_i = \lim_{\delta \to 0} \frac{f(x + \delta y_i) - f(x)}{\delta}
\]

\[
x \leftarrow x - \sigma \nabla f(x)
\]

\[
x + \sigma \sum_{i=1}^{m} w_i y_i
\]
Basic Approach: Approximated Gradient Descent

We modify the gradient equation: (1) use $y_i$ instead of $e_i$

$$y_i \sim \mathcal{N}(0, I)$$

$$-w_i = \lim_{\delta \to 0} \frac{f(x + \delta y_i) - f(x)}{\delta}$$

$$x \leftarrow x - \sigma \nabla f(x)$$

$$x + \sigma \sum_{i=1}^{m} w_i y_i$$
Basic Approach: Approximated Gradient Descent

We modify the gradient equation: (2) make large test steps

$$y_i \sim \mathcal{N}(0,I)$$

$$-w_i = \lim_{\delta \to 0} \frac{f(x + \delta y_i) - f(x)}{\delta}$$

$$x \leftarrow x - \sigma \nabla f(x)$$

$$x + \sigma \sum_{i=1}^{m} w_i y_i$$

Evolutionary Gradient Search (EGS) [Salmon 1998, Arnold & Salomon 2007]
Rank-Based Approximated Gradient Descent

We modify the gradient equation: (3) use ranks instead of $f$-values

$$y_i \sim \mathcal{N}(0, I)$$

$$-w_i \propto \text{rank}_i(f(x + \delta y_i)) - m/2$$

$$x \leftarrow x - \sigma \nabla f(x)$$

$$x + \sigma \sum_{w_i > 0} w_i y_i$$

Rank-Based Approximated Gradient Descent

We modify the gradient equation: (3) use ranks instead of $f$-values

\[ \sum_{w_i>0} w_i \approx 1 \]

\[ -w_i = \frac{\ln(\text{rank}_i(f(x + \delta y_i))) - \ln \frac{m+1}{2}}{m/2} \]

\[ y_i \sim N(0, I) \]

\[ x \leftarrow x - \sigma \nabla f(x) \]

\[ x + \sigma \sum_{w_i>0} w_i y_i \]

Using Rank-Based Weights

- introduces robustness to (erroneously) $f$-value differences
- introduces invariance to
  - scaling of (the gradient of) $f$
  - strictly monotonous $f$-transformations
Invariance from Rank-Based Weights

Three functions belonging to the same equivalence class

A rank-based search algorithm is invariant under the transformation with any order preserving (strictly increasing) $g$.

Invariances make

- observations meaningful as a rigorous notion of generalization
- algorithms predictable and/or "robust"
## From Gradient Descent to Evolution Strategies

<table>
<thead>
<tr>
<th>Test Steps:</th>
<th>Gradient Descent</th>
<th>Evolution Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>unit vectors</td>
<td>(symmetric) random vectors</td>
</tr>
<tr>
<td></td>
<td>very small</td>
<td>(very) large</td>
</tr>
<tr>
<td></td>
<td>dimension (n) or (2n)</td>
<td>any number (&gt; 1)</td>
</tr>
</tbody>
</table>

| Weights:             | Partial derivatives (estimated) | fixed rank-based |

| Realized Step Length:| line search | step-size control (non-trivial) |
Ill-Conditioned Problems

Curvature of level sets

Consider the convex-quadratic function
\[ f(x) = \frac{1}{2} (x - x^*)^T H (x - x^*) = \frac{1}{2} \sum_i h_{i,i} (x_i - x_i^*)^2 + \frac{1}{2} \sum_{i \neq j} h_{i,j} (x_i - x_i^*) (x_j - x_j^*) \]

\( H \) is Hessian matrix of \( f \) and symmetric positive definite

gradient direction \(-f'(x)^T\)

Newton direction \(-H^{-1}f'(x)^T\)

Ill-conditioning means squeezed level sets (high curvature). Condition number equals nine here. Condition numbers up to \( 10^{10} \) are not unusual in real world problems.

If \( H \approx I \) (small condition number of \( H \)) first order information (e.g. the gradient) is sufficient. Otherwise second order information (estimation of \( H^{-1} \)) is necessary.
Rank-Based Approximated Gradient Descent

\[ y_i \sim \mathcal{N}(0, I) \]

\[ -w_i = \frac{\ln(\text{rank}_i(f(x + \delta y_i))) - \ln \frac{m+1}{2}}{m/2} \]

\[ x \leftarrow x - \sigma \nabla f(x) \]

\[ x + \sigma \sum_{w_i > 0} w_i y_i \]

Rank-Based Approximated Gradient Descent: Variable Metric

We estimate the shape of the level sets (without using $f$-values)

\[ y_i \sim \mathcal{N}(0, C) \]
\[ w_i = \frac{\ln(\text{rank}_i(f(x + \delta y_i))) - \ln \frac{m+1}{2}}{m/2} \]

\[ x \leftarrow x - \sigma \nabla f(x) \]
\[ x + \sigma \sum_{w_i > 0} w_i y_i \]

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [Hansen & Ostermeier 2001, Hansen et al 2003]
Let $x \in \mathbb{R}^n$, $\sigma > 0$, $C = I_n$, $y_0 = 0$

$x_k \sim \mathcal{N}(x, \sigma^2 C) = x + \sigma \mathcal{N}(0, C) \in \mathbb{R}^n$, $k = 1 \ldots \lambda$

$$y_k = \frac{x_{\text{permute}_k(k)} - x}{\sigma}$$

sorted by $f$

$y_k \sim \mathcal{N}(0, C)$

$$x \leftarrow x + c_m \sigma \sum_{w_k > 0, k \neq 0} w_k y_k$$

$c_m \approx \sum_{i=1}^{\mu} w_k \approx 1, \mu \approx \lambda / 2$

$$y_0 \leftarrow (1 - c_c) y_0 + \sqrt{c_c (2 - c_c) \mu} \sum_{k=1}^{\mu} w_k y_k$$

$c_c \approx \sqrt{c}\mu$, $\mu = \frac{(\sum_{i=1}^{\mu} w_k)^2}{\sum_{i=1}^{\mu} w_k^2}$

$$C \leftarrow C + c_{\mu} \sum_{k=0}^{\lambda} w_k (y_k y_k^T - C)$$

$c_{\mu} \approx \mu_w / n^2$, $\sum_{k=0}^{\lambda} w_k \approx 0$

$$\sigma \leftarrow \sigma \times \exp(\ldots)$$
Let \( x \in \mathbb{R}^n, \sigma > 0, C = I_n, y_0 = 0 \)

\[
x_k \sim \mathcal{N}(x, \sigma^2 C) = x + \sigma \mathcal{N}(0, C) \in \mathbb{R}^n, \quad k = 1 \ldots \lambda
\]

\[
y_k = \frac{x_{\text{permute}_i(k)} - x}{\sigma}
\]

sorted by \( f \) \( y_k \sim \mathcal{N}(0, C) \)

\[
x \leftarrow x + c_m \sigma \sum_{w_k > 0, k \neq 0} w_k y_k, \quad c_m \approx \sum_{k=1}^{\mu} w_k \approx 1, \mu \approx \lambda/2
\]

\[
y_0 \leftarrow (1 - c_c) y_0 + \sqrt{c_c (2 - c_c) \mu_w} \sum_{k=1}^{\mu} w_k y_k, \quad c_c \approx \sqrt{c_{\mu}}, \quad \mu_w = \frac{(\sum_{i=1}^{\mu} w_k)^2}{\sum_{i=1}^{\mu} w_k^2}
\]

\[
C \leftarrow C + c_\mu \sum_{k=0}^{\lambda} w_k (y_k y_k^\top - C), \quad c_\mu \approx \mu_w/n^2, \sum_{k=0}^{\lambda} w_k \approx 0
\]

\[
\sigma \leftarrow \sigma \times \exp(\ldots)
\]
Summary

• There are many interesting applications for robust black-box optimization

• It takes three modifications to turn gradient descent into an evolution strategy
  • Replace unit vectors with a symmetrical distribution of test step (of any number)
  • Replace small test steps with large test steps (no limit to zero)
  • Replace $f$-value differences with fixed weights for linear combination of test steps

• We can reliably estimate the shape of the level sets (the inverse Hessian) in evolution strategies (CMA-ES) without using $f$-values