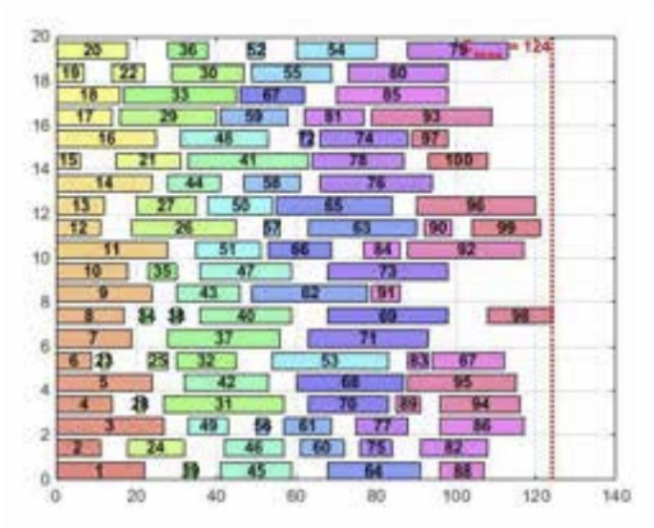
Covariance matrix adaptation evolution strategy (CMA-ES)

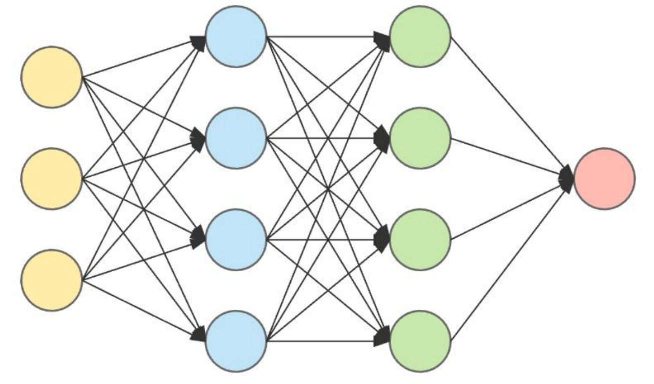
Mikołaj Słupiński

Zastosowania



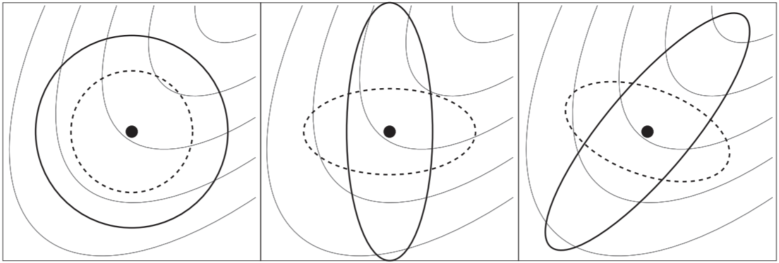
Task Scheduling Algorithm Using Covariance Matrix Adaptation Evolution Strategy (CMA-ES) in Cloud Computing Ghazaal Emadi et al.

Zastosowania



CMA-ES for Hyperparameter Optimization of Deep Neural Networks Ilya Loshchilov et al.

Rozkład normalny



$\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ | $\mathcal{N}(\mathbf{0}, \boldsymbol{D}^2)$ | $\mathcal{N}(\mathbf{0}, \boldsymbol{C})$

Rozkład normalny

$$C = BD^2B^\top$$

Rozkład normalny

$$
\begin{aligned}
C^{-1} &= \left(BD^2B^\top\right)^{-1} \\
&= {B^\top}^{-1} D^{-2} B^{-1} \\
&= BD^{-2}B^\top \\
&= B \operatorname{diag}\left(\frac{1}{d_1^2}, \ldots, \frac{1}{d_n^2}\right) B^\top.
\end{aligned}
$$

Rozkład normalny

$$C^{-\frac{1}{2}} = BD^{-1}B^\top$$
$$= B \operatorname{diag}\left(\frac{1}{d_1}, \ldots, \frac{1}{d_n}\right) B^\top$$

Rozkład normalny

$$
\begin{aligned}
\mathcal{N}(m, C) &\sim m + \mathcal{N}(\mathbf{0}, C) \\
&\sim m + C^{\frac{1}{2}} \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
&\sim m + BD \underbrace{B^\top \mathcal{N}(0, \mathbf{I})}_{\sim \mathcal{N}(0, \mathbf{I})} \\
&\sim m + B \underbrace{D \mathcal{N}(0, \mathbf{I})}_{\sim \mathcal{N}(0, D^2)},
\end{aligned}
$$

Schemat algorytmu

initialization $m, \sigma, C = I, p_\sigma = 0, p_c = 0$;
while *not terminate* **do**
    for $i \in \{1 \ldots \lambda\}$ **do**
        $x_i = \texttt{sample}(m, \sigma^2 C)$;
        $f_i = \texttt{fitness}(x_i)$
    end
    $x_{1 \ldots \lambda} \leftarrow x_{s(1) \ldots s(\lambda)}$;
    $m' = m$;
    $m \leftarrow \texttt{update\_m}(x_1, \ldots, x_\lambda)$;
    $p_\sigma \leftarrow \texttt{update\_ps}\left(p_\sigma, \sigma^{-1} C^{-1/2} \left(m - m'\right)\right)$;
    $p_c \leftarrow \texttt{update\_pc}\left(p_c, \sigma^{-1} \left(m - m'\right), \|p_\sigma\|\right)$;
    $C \leftarrow \texttt{update\_c}(C, p_c, \left(x_1 - m'\right)/\sigma, \ldots, \left(x_\lambda - m'\right)/\sigma)$;
    $\sigma \leftarrow \texttt{update\_sigma}(\sigma, \|p_\sigma\|)$;
end
return x_1

Próbkowanie

$$x_k^{(g+1)} \sim m^{(g)} + \sigma^{(g)} \mathcal{N}\left(0, C^{(g)}\right) \quad \text{for } k = 1, \ldots, \lambda$$

Skąd wziąć $m^{(g)}$, $C^{(g)}$, $\sigma^{(g+1)}$?

Selekcja i rekombinacja

$$\boldsymbol{m}^{(g+1)} = \sum_{i=1}^{\mu} w_i \boldsymbol{x}_{i:\lambda}^{(g+1)}$$
$$\sum_{i=1}^{\mu} w_i = 1, \quad w_1 \geq w_2 \geq \cdots \geq w_\mu > 0$$

$\mathrm{x}_{i:\lambda}$ oznacza x_k o i-tej najmniejszej wartości funkcji celu

$$\mu_{\mathrm{eff}} = \left(\frac{\|\boldsymbol{w}\|_1}{\|\boldsymbol{w}\|_2}\right)^2 = \frac{\|\boldsymbol{w}\|_1^2}{\|\boldsymbol{w}\|_2^2} = \frac{\left(\sum_{i=1}^{\mu} |w_i|\right)^2}{\sum_{i=1}^{\mu} w_i^2} = \frac{1}{\sum_{i=1}^{\mu} w_i^2}$$

Dobre założenia:

$$1 \leq \mu_{\mathrm{eff}} \leq \mu$$

$$\mu_{\mathrm{eff}} \approx \lambda/4$$

$$w_i \propto \mu - i + 1$$

Aktualizacja średniej

$$m^{(g+1)} = m^{(g)} + c_{\mathrm{m}} \sum_{i=1}^{\mu} w_i \left(x_{i:\lambda}^{(g+1)} - m^{(g)} \right)$$

Estymacja macierzy kowariancji

$$C_{\mathrm{emp}}^{(g+1)} = \frac{1}{\lambda - 1} \sum_{i=1}^{\lambda} \left(x_i^{(g+1)} - \frac{1}{\lambda} \sum_{j=1}^{\lambda} x_j^{(g+1)} \right) \left(x_i^{(g+1)} - \frac{1}{\lambda} \sum_{j=1}^{\lambda} x_j^{(g+1)} \right)^{\top}$$

$$C_{\lambda}^{(g+1)} = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \left(x_i^{(g+1)} - m^{(g)} \right) \left(x_i^{(g+1)} - m^{(g)} \right)^{\top}$$

Estymacja macierzy kowariancji

$$C_{\mu}^{(g+1)} = \sum_{i=1}^{\mu} w_i \left(x_{i:\lambda}^{(g+1)} - m^{(g)} \right) \left(x_{i:\lambda}^{(g+1)} - m^{(g)} \right)^{\top}$$

Estymacja macierzy kowariancji

Aby $C_\mu^{(g+1)}$ było wiarygodnym estymatorem, wartość μ_{eff} musi być dostatecznie duża: otrzymanie wskaźnika uwarunkowania niższego niż dziesięć dla $\boldsymbol{C}_\mu^{(g)}$ dla $f_{\text{sphere}}(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$, wymaga $\mu_{\text{eff}} \approx 10n$.

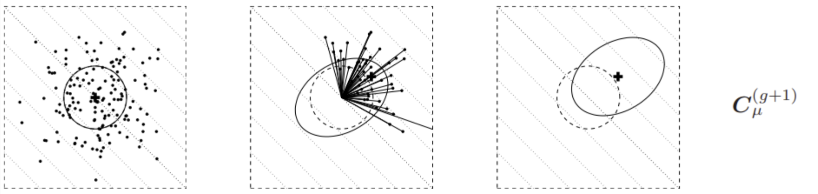
Zauważmy, że żeby przeszukiwanie było szybkie, to λ musi być małe. Powiedzieliśmy wcześniej, że zazwyczaj $\mu_{\text{eff}} \approx \lambda/4$, a zatem również μ_{eff} musi być małe.

Jak zatem z tego wybrnąć?

Rank-μ-Update

$$C^{(g+1)} = \frac{1}{g+1} \sum_{i=0}^{g} \frac{1}{{\sigma^{(i)}}^2} C_\mu^{(i+1)}$$

Rank-μ-Update



The CMA Evolution Strategy: A Tutorial N.Hansen

Rank-μ-Update

$$\begin{aligned} C^{(g+1)} &= (1 - c_\mu)\, C^{(g)} + c_\mu \frac{1}{\sigma^{(g)2}} C_\mu^{(g+1)} \\ &= (1 - c_\mu)\, C^{(g)} + c_\mu \sum_{i=1}^{\mu} w_i y_{i:\lambda}^{(g+1)} {y_{i:\lambda}^{(g+1)}}^\top \end{aligned}$$

$$y_{i:\lambda}^{(g+1)} = \left(x_{i:\lambda}^{(g+1)} - m^{(g)} \right) / \sigma^{(g)}$$

Rank-μ-Update

$$C^{(g+1)} = \left(1 - c_\mu \sum w_i\right) C^{(g)} + c_\mu \sum_{i=1}^{\lambda} w_i y_{i:\lambda}^{(g+1)} {y_{i:\lambda}^{(g+1)}}^\top$$

$$= {C^{(g)}}^{1/2} \left(I + c_\mu \sum_{i=1}^{\lambda} w_i \left(z_{i:\lambda}^{(g+1)} {z_{i:\lambda}^{(g+1)}}^\top - I \right) \right) {C^{(g)}}^{1/2}$$

$$c_\mu \approx \mu_{\mathrm{eff}} / n^2$$

$$z_{i:\lambda}^{(g+1)} = {C^{(g)}}^{-1/2} y_{i:\lambda}^{(g+1)}$$

Rank-One-Update

$$\mathcal{N}(0,1)\boldsymbol{y}_1 + \cdots + \mathcal{N}(0,1)\boldsymbol{y}_{g_0} \sim \mathcal{N}\left(\mathbf{0}, \sum_{i=1}^{g_0} \boldsymbol{y}_i \boldsymbol{y}_i^\top\right)$$

$$\boldsymbol{C}^{(g+1)} = (1 - c_1)\,\boldsymbol{C}^{(g)} + c_1 \boldsymbol{y}_{g+1} \boldsymbol{y}_{g+1}^\top$$

Ścieżka ewolucji

$$\frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} + \frac{m^{(g)} - m^{(g-1)}}{\sigma^{(g-1)}} + \frac{m^{(g-1)} - m^{(g-2)}}{\sigma^{(g-2)}}$$

$$p_{\mathrm{c}}^{(g+1)} = (1 - c_{\mathrm{c}})\, p_{\mathrm{c}}^{(g)} + \sqrt{c_{\mathrm{c}}\,(2 - c_{\mathrm{c}})\,\mu_{\mathrm{eff}}}\frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}}$$

$$p_{\mathrm{c}}^{(g+1)} \sim \mathcal{N}(0, C)$$

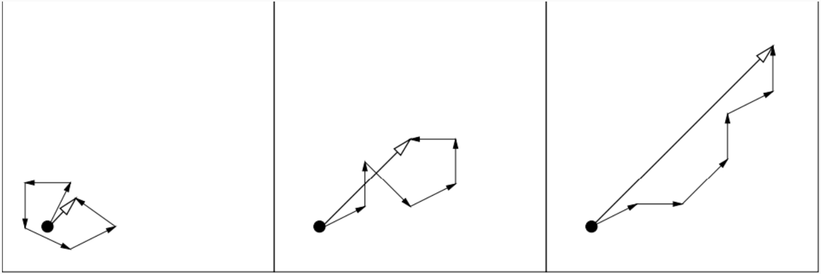
$$C^{(g+1)} = (1 - c_1)\, C^{(g)} + c_1 p_{\mathrm{c}}^{(g+1)} {p_{\mathrm{c}}^{(g+1)}}^{\top}$$

Finalne rownanie macierzy kowariancji

$$C^{(g+1)} = (1 \underbrace{- c_1 - c_\mu \sum w_j}_{\text{can be close or equal to 0}}) C^{(g)} + c_1 \underbrace{p_c^{(g+1)} p_c^{(g+1)^\mathsf{T}}}_{\text{rank-one update}} + c_\mu \underbrace{\sum_{i=1}^{\lambda} w_i \, y_{i:\lambda}^{(g+1)} \left(y_{i:\lambda}^{(g+1)} \right)^\mathsf{T}}_{\text{rank-}\mu\text{ update}} \tag{30}$$

$$c_1 \approx 2/n^2$$
$$c_\mu \approx \min\left(\mu_{\text{eff}}/n^2, 1 - c_1\right)$$
$$y_{i:\lambda}^{(g+1)} = \left(x_{i:\lambda}^{(g+1)} - m^{(g)}\right)/\sigma^{(g)}$$

Kontrola rozmiaru kroku



The CMA Evolution Strategy: A Tutorial N.Hansen

Cumulative step length adaptation (CSA)

$$p_\sigma^{(g+1)} = (1 - c_\sigma)\, p_\sigma^{(g)} + \sqrt{c_\sigma\,(2 - c_\sigma)\,\mu_{\mathrm{eff}}}\, C^{(g)^{-\frac{1}{2}}}\, \frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}}$$

Aktualizacja σ

$$\ln \sigma^{(g+1)} = \ln \sigma^{(g)} + \frac{c_\sigma}{d_\sigma} \left(\frac{\left\| p_\sigma^{(g+1)} \right\|}{\mathrm{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1 \right)$$

$$d_\sigma \approx 1$$

$$\mathrm{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\| = \sqrt{2}\Gamma\left(\frac{n+1}{2}\right) / \Gamma\left(\frac{n}{2}\right)$$

Uzasadnienie doboru parametrów można znaleźć w *Verallgemeinerte individuelle Schrittweitenregelung in der Evolutionsstrategie* Hansen N.

Aktualizacja σ

$$\sigma^{(g+1)} = \sigma^{(g)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\left\|p_\sigma^{(g+1)}\right\|}{\mathrm{E}\|\mathcal{N}(0, \mathrm{I})\|} - 1\right)\right)$$

Finalny algorytm

initialization $m, \sigma, C = I, p_\sigma = 0, p_c = 0$;
while *not terminate* **do**
    for $i \in \{1 \dots \lambda\}$ **do**
        $x_i = \mathtt{sample}(m, \sigma^2 C)$;
        $f_i = \mathtt{fitness}(x_i)$
    end
    $x_{1 \dots \lambda} \leftarrow x_{s(1) \dots s(\lambda)}$;
    $m' = m$;
    $m \leftarrow \mathtt{update\_m}(x_1, \dots, x_\lambda)$;
    $p_\sigma \leftarrow \mathtt{update\_ps}\left(p_\sigma, \sigma^{-1} C^{-1/2} (m - m')\right)$;
    $p_c \leftarrow \mathtt{update\_pc}\left(p_c, \sigma^{-1} (m - m'), \|p_\sigma\|\right)$;
    $C \leftarrow \mathtt{update\_c}(C, p_c, (x_1 - m')/\sigma, \dots, (x_\lambda - m')/\sigma)$;
    $\sigma \leftarrow \mathtt{update\_sigma}(\sigma, \|p_\sigma\|)$;
end
return x_1

Przebieg algorytmu