A consensus-based global optimization method with adaptive momentum estimation

December 16, 2020
Machine learning tasks

Highly nonconvex unconstrained optimization problem

\[ x^* = \arg \min_{x \in \mathbb{R}^d} f(x) \]

with the loss function

\[ f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) = \frac{1}{n} \sum_{i=1}^{n} \| \mathcal{N}_x(\hat{x}) - \hat{y} \| \]

\( x \) is the parameter vector
\( \mathcal{N}_x \) represents a neural network representation
\( (\hat{x}_i, \hat{y}_i)_{i=1}^{n} \) is a set of labeled data
\( \| \cdot \| \) is the \( L^2 \) distance
\( d \gg 1 \)
Outline

1. Optimization methods: Zero-order or first-order?
2. CBO method
3. Adam-CBO Method
4. Linear stability analysis of Adam-CBO
5. Numerical results
   - Rastirgin function
   - Machine learning tasks
First-order methods

- gradient descent method

\[ x^{t+1} = x^t - \alpha \nabla f(x^t) \]

with \( \alpha \) being the learning rate

- stochastic gradient descent (SGD) method

\[ x^{t+1} = x^t - \alpha \nabla f_i(x^t) \]

- SGD method with momentum term\(^1\)

\[ x^{t+1} = x^t - m^t \]
\[ m^t = -\gamma m^{t-1} + \alpha \nabla f_i(x^t) \]

Adaptive momentum method (Adam)\(^2\)

\[
\begin{align*}
x^{t+1} &= x^t - \gamma \frac{\hat{m}^t}{\sqrt{\hat{v}^t} + \epsilon} \\
m^t &= \beta_1 m^{t-1} + (1 - \beta_1) \nabla f(x^t), \quad \hat{m}_t = \frac{m_t}{1 - \beta_1^t} \\
v^t &= \beta_2 v^{t-1} + (1 - \beta_2) \nabla^2 f(x^t), \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}
\end{align*}
\]

where \(0 < \beta_1, \beta_2 < 1\)

First-order methods

- mostly have problems with loss functions containing large noise or non-differentiable units
- gradient tends to explode or vanish as the neural network gets deeper\(^3\)
- are easily influenced by the loss landscape\(^4\)

---


\(^4\) Shengchao Liu, Dimitris Papailiopoulos, and Dimitris Achlioptas. “Bad global minima exist and sgd can reach them”. In: *Advances in Neural Information Processing Systems 33* (2020).
Zero-order methods: Gradient-free

- Nelder-Mead method
- genetic algorithm
- simulated annealing method
- particle swarm optimization
- consensus based optimization (CBO) method

---


Original CBO method

Interacting particles during the dynamic evolution

- tend to their weighted average
- undergo fluctuation due to the random noise

\( N \) particles \( X^i, \ i = 1, \cdots N \)

\[
\dot{X}^i = -\lambda (X^i - \bar{x}^*) + \sigma |X^i - \bar{x}^*| \dot{W}_t^i
\]

weighted average \( \bar{x}^* = \frac{1}{\sum_{i=1}^N e^{-\beta L(X^i)}} \sum_{i=1}^N X^i e^{-\beta L(X^i)} \)

cost (loss) function \( L(x) \) to be optimized

white noise \( \dot{W}_t \)

Discretization of the above system with unit stepsize

\[
X_{t+1}^i = X_t^i - \lambda (X_t^i - \bar{x}^*) + \sigma |X_t^i - \bar{x}^*| dW_t^i
\]
Curse of dimensionality (CoD)

- Exponential convergence rate under dimension-dependent conditions\(^8\)
- The larger the dimension, the smaller the learning rate (CoD)
- Replacement of the isotropic geometric Brownian motion with the component-wise one\(^9\)

\[
X_{t+1}^i = X_t^i - \lambda(X_t^i - \bar{x}^*) + \sigma(X_t^i - \bar{x}^*)dW_t^i
\]

Random mini-batch: \(\mathcal{O}(N) \rightarrow \mathcal{O}(\frac{N}{M})\)

- Convergence to the global minimizer with dimension-independent parameters\(^{10}\)


Some practical issues in CBO

- the initial data need to be well-chosen
- difficult to optimize high dimensional no-convex function (Rastrigin Function over 20 dimension)
- difficult to optimize deep neural networks with many parameters
First-order momentum

The same system without random term but with inertial effect

\[ \sigma \ddot{X}_t^i + \dot{X}_t^i = -(X_t - x^*), \quad i = 1, \cdots, N \]

An equivalent first-order system

\[ \dot{X}_t^i = -M_t^i \]
\[ \sigma \dot{M}_t^i + M_t^i = X_t^i - x^* \]

Discretization

\[ X_{t+1}^i = X_t^i - \delta t M_{t+\frac{1}{2}}^i \]
\[ M_{t+\frac{1}{2}}^i = \frac{\sigma - \delta t}{\sigma + \delta t} M_{t-\frac{1}{2}}^i + \frac{2\delta t}{\sigma + \delta t} (X_t^i - x^*) \]
Relabel $M_{t+\frac{1}{2}}^i$ by $M_{t+1}^i$

\[ X_{t+1}^i = X_t^i - \lambda M_{t+1}^i \]
\[ M_{t+1}^i = \beta_1 M_t^i + (1 - \beta_1)(X_t^i - \bar{x}^*) \]

with $\lambda = \delta t$ and $\beta_1 = \frac{\sigma - \delta t}{\sigma + \delta t} = 1 - \frac{2\delta t}{\sigma + \delta t}$

$\beta_1$ is near 1 (= 0.9 in practice) since $\delta t$ is small

Add the stochastic term

\[ X_{t+1}^i = X_t^i - \lambda M_{t+1}^i + \sigma_t W_t^i \]
\[ M_{t+1}^i = \beta_1 M_t^i + (1 - \beta_1)(X_t^i - \bar{x}^*) \]
A recursive argument of $M_t^i$ yields

$$M_t^i = \beta_1 M_{t-1}^i + (1 - \beta_1)(X_{t-1}^i - x^*)$$

$$= \beta_1(\beta_1 M_{t-2}^i + (1 - \beta_1)(X_{t-1}^i - x^*)) + (1 - \beta_1)(X_{t-2}^i - x^*)$$

$$= (1 - \beta_1) \sum_{k=0}^{t-1} \beta_1^{t-k}(X_k^i - x^*).$$

Stationary assumption of $X_k^i - x^*$ w.r.t. $k$ leads to

$$\mathbb{E}[M_t^i] = (1 - \beta_1)\mathbb{E}\left[\sum_{k=0}^{t} \beta_1^{t-k}(X_k^i - x^*)\right]$$

$$= (1 - \beta_1^t)\mathbb{E}[X_t^i - x^*]$$

Unbiased estimation of first-order moment
Second-order momentum $\mathbb{E}(|X^i_t - x^*|^2)$

- Define $V^i_t = \beta_2 V^i_{t-1} + (1 - \beta_2)|X^i_t - x^*|^2$

  Application of the same argument for $\mathbb{E}[X^i_t]$ yields

  $$\mathbb{E}[V^i_t] = (1 - \beta_2^t)\mathbb{E}[|X^i_t - x^*|^2]$$

  Unbiased estimation of $\mathbb{E}(|X^i_t - x^*|^2)$

  $$\hat{V}^i_t = \frac{V^i_t}{1 - \beta_2^t}$$

- Modify the model

  $$X^i_{t+1} = X^i_t - \frac{\lambda \hat{M}^i_{t+1}}{\sqrt{\hat{V}^i_{t+1} + \epsilon}} + \sigma^t W^i_t$$

  with a small $\epsilon (1e - 8)$ to avoid the vanishing of denominator
Input: $\lambda, N, M, t_N, \beta_1, \beta_2$

1. Initialize $X^i_0, i = 1, \cdots, N$ by the uniform distribution;
2. Initial $M^i_0, V^i_0 = 0$; /* Initialize first order and second order moments. */

for $t = 0$ to $t_N$ do

3. Generate a random permutation of index $\{1, 2, \cdots, N\}$ to form set $P_k$;

4. Generate batch set of particles in order of $P_k$ as $B^1, \cdots, B^N_M$ with each batch having $M$ particles;

5. for $j = 0$ to $\frac{N}{M}$ do

6. Update $x^* = \sum_{k \in B_j} \frac{X^k_t \mu^k_t}{\sum_{i \in B^j} \mu^i_t}$, where $\mu^i_t = \omega_t^\alpha(X^i_t)$;

7. Update $X^i_t$ for $j \in B^j$ as follows

8. $M^i_{t+1} = \beta_1 M^i_t + (1 - \beta_1)(X^i_t - x^*)$ \quad $\hat{M}^i_{t+1} = M^i_{t+1}/(1 - \beta^t_1)$;

9. $V^i_{t+1} = \beta_2 V^i_t + (1 - \beta_2)(X^i_t - x^*)^2$ \quad $\hat{V}^i_{t+1} = V^i_{t+1}/(1 - \beta^t_2)$;

10. $X^i_{t+1} = X^i_t - \lambda \hat{M}^i_t/(\sqrt{\hat{V}^i_t} + \epsilon) + \sigma^t \sum_{k=1}^d \bar{e}_k z_i$.

end

end

Output: $X^i_{t_N}, i = 1 \cdots N$
Continuous formulation without the stochastic term

\[
\dot{m} = (\beta_1 - 1)m + (1 - \beta_1)(x - \bar{x})
\]
\[
\dot{v} = (\beta_2 - 1)v + (1 - \beta_2)(x - \bar{x})^2
\]
\[
\hat{m} = \frac{m}{1 - \beta_1^t} \quad \hat{v} = \frac{v}{1 - \beta_2^t}
\]
\[
\dot{x} = -\lambda \frac{\hat{m}}{\sqrt{\hat{v}} + \epsilon}
\]
Linearization around $m = 0, x = \bar{x}, v = 0$

$$\dot{m} = -(1 - \beta_1)m + (1 - \beta_1)\bar{x}$$
$$\dot{v} = -(1 - \beta_2)v$$

$$\dot{\bar{x}} = -\frac{\lambda}{(1 - \beta_1^t)\epsilon} m \rightarrow -\frac{\lambda}{\epsilon} m = -\mu m \quad (t \rightarrow \infty)$$

with $\bar{x} = x - \bar{x}$ and $\mu = \lambda/\epsilon$, and in a vector form

$$\frac{d}{dt} \begin{pmatrix} m \\ v \\ \bar{x} \end{pmatrix} = \begin{pmatrix} -(1 - \beta_1) & 0 & 1 - \beta_1 \\ 0 & -(1 - \beta_2) & 0 \\ -\mu & 0 & 0 \end{pmatrix} \begin{pmatrix} m \\ v \\ \bar{x} \end{pmatrix}$$
Theorem

The Adam-CBO method generates a sequence that converges to the optimal solution with rates independent of the learning rate $\lambda$.

Proof.

Eigenvalues of the matrix on the right-hand side are $\beta_2 - 1$ and $\frac{1}{2}(\beta_1 - 1 \pm i\sqrt{1 - \beta_1\sqrt{1 - 1 + 4\mu}})$ (typically $1 - \beta_1 \ll 4\mu$), respectively. Thus, $m, v, \tilde{x}$ decay to 0 exponentially with rate $\beta_2 - 1$ when $\beta_1 > 2\beta_2 + 1$ and with rate $\frac{1}{2}(\beta_1 - 1)$ when $\beta_1 < 2\beta_2 + 1$ in an oscillatory way.

- $\beta_1 = 0.9$ and $\beta_2 = 0.99$
- Continuous formulation of CBO without random noise

$$\dot{x} = -\lambda(x - \bar{x})$$

- The decay rate of the CBO method depends exponentially on the learning rate $\lambda$
Rastrigin function

\[ f(x) = \frac{1}{d} \sum_{i=1}^{d} [(x_i - B)^2 - 10 \cos(2\pi(x_i - B)) + 10] + C \]

with \( B = \arg \min f(x) \) and \( C = \min f(x) \)
Massive local minima of Rastrigin function

- Exponential growth of the number of local minima: $5^d$
- Number of minima is $5^{1000} \approx 10^{690}$, when $d = 1000$

<table>
<thead>
<tr>
<th>d</th>
<th>1</th>
<th>2</th>
<th>30</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of local minima</td>
<td>5</td>
<td>$5^2$</td>
<td>$5^{30}$</td>
<td>$5^{100}$</td>
<td>$5^{1000}$</td>
</tr>
</tbody>
</table>

Table: Number of local minima in terms of dimension
Comparison with different random processes

<table>
<thead>
<tr>
<th>$d$</th>
<th>$N$</th>
<th>$M$</th>
<th>CBO</th>
<th>Wiener process</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>50</td>
<td>40</td>
<td>$\mathcal{N}(0, 1)$</td>
<td>$99%$</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>40</td>
<td>$\mathcal{N}(0, 1)$</td>
<td>$2%$</td>
</tr>
<tr>
<td>20</td>
<td>50</td>
<td>40</td>
<td>$\mathcal{N}(0, 1)$</td>
<td>$0%$</td>
</tr>
<tr>
<td>20</td>
<td>50</td>
<td>20</td>
<td>$\mathcal{N}(0, 1)$</td>
<td>$0%$</td>
</tr>
<tr>
<td>30</td>
<td>50</td>
<td>40</td>
<td>$\mathcal{N}(0, 1)$</td>
<td>$0%$</td>
</tr>
<tr>
<td>30</td>
<td>500</td>
<td>5</td>
<td>$\mathcal{N}(0, 1)$</td>
<td>$0%$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$U(-1, 1)$</td>
<td>$99%$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$U(-1, 1)$</td>
<td>$2%$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$U(-1, 1)$</td>
<td>$0%$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$U(-1, 1)$</td>
<td>$0%$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$d$</th>
<th>$N$</th>
<th>$M$</th>
<th>Adam-CBO</th>
<th>Wiener process</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>500</td>
<td>5</td>
<td>$\mathcal{N}(0, 1)$</td>
<td>$0%$</td>
</tr>
<tr>
<td>100</td>
<td>5000</td>
<td>5</td>
<td>$\mathcal{N}(0, 1)$</td>
<td>$0%$</td>
</tr>
<tr>
<td>1000</td>
<td>8000</td>
<td>50</td>
<td>$\mathcal{N}(0, 1)$</td>
<td>$0%$</td>
</tr>
</tbody>
</table>
\( \lambda = 0.1 \), and \( \sigma^t = 0.99^{\frac{t}{5}} \)

<table>
<thead>
<tr>
<th>( d )</th>
<th>( N )</th>
<th>( M )</th>
<th>Adam-CBO ( \mathcal{N}(0, 1) )</th>
<th>Adam-CBO ( \mathcal{U}(−1, 1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>8000</td>
<td>50</td>
<td>92%</td>
<td>20%</td>
</tr>
<tr>
<td>1000</td>
<td>10000</td>
<td>50</td>
<td>100%</td>
<td>28%</td>
</tr>
<tr>
<td>1000</td>
<td>12000</td>
<td>50</td>
<td>100%</td>
<td>28%</td>
</tr>
<tr>
<td>1000</td>
<td>14000</td>
<td>50</td>
<td>100%</td>
<td>32%</td>
</tr>
<tr>
<td>1000</td>
<td>16000</td>
<td>50</td>
<td>100%</td>
<td>32%</td>
</tr>
</tbody>
</table>
Spectrail bias [10]/Frequency principle [11]

\[ u(x) = \sin(2\pi x) + \sin(8\pi x^2) \]

- Network width = 50, depth = 3, and 2701 parameters
- \( \lambda = 0.2 \)
- \( N = 500 \) and \( M = 5 \) in the first 50000 iterations
- Afterwards the random term is ignored and \( M = 10 \)
$u(x) = \begin{cases} 
1 & x < -\frac{7}{8}, x > \frac{7}{8}, -\frac{1}{8} < x < \frac{1}{8} \\
-1 & \frac{3}{8} < x < \frac{5}{8}, -\frac{5}{8} < x < -\frac{3}{8} \\
0 & \text{otherwise}
\end{cases}$

The same setup as in the previous slide
Gradient exploding or vanishing: DNN with fixed width 10 and different depths

\[ u(x) = \sin(k\pi x^k) \]

\( N = 500, \ M = 5 \)

<table>
<thead>
<tr>
<th>depth</th>
<th>Num of parameters</th>
<th>k = 2</th>
<th>k = 3</th>
<th>k = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>141</td>
<td>6.62 e-03</td>
<td>1.32 e-02</td>
<td>1.71 e-01</td>
</tr>
<tr>
<td>7</td>
<td>471</td>
<td>4.78 e-03</td>
<td>1.42 e-02</td>
<td>7.54 e-03</td>
</tr>
<tr>
<td>12</td>
<td>1021</td>
<td>7.44 e-03</td>
<td>1.30 e-02</td>
<td>5.32 e-02</td>
</tr>
<tr>
<td>22</td>
<td>2121</td>
<td>1.00 e-02</td>
<td>1.01 e-02</td>
<td>1.21 e-01</td>
</tr>
</tbody>
</table>

Table: Absolute \( L^2 \) norm in terms of network depth when \( k = 2, 3, 4 \)

SGD or Adam fails to converges well (with final error around 0.3) when the network depth is 4 and 10, respectively
Numerical results

Machine learning tasks

Solving PDEs by DNNs: Deep Ritz method

\[
\begin{align*}
- \nabla \cdot (A(x) \nabla u) &= - \sum_{i=1}^{d} \delta(x_i) \quad x \in \Omega = [-1, 1]^d \\
u(x) &= g(x) \quad x \in \partial \Omega
\end{align*}
\]

with

\[
A(x) = \begin{bmatrix}
(x_1^2)^\frac{1}{4} \\
\vdots \\
(x_d^2)^\frac{1}{4}
\end{bmatrix}.
\]

- Exact solution \( u(x) = \sum_{i=1}^{d} |x_i|^\frac{1}{2} \) is only in \( H^{1/2}(\Omega) \)
- Derivatives have singularities at \( x_i = 0 \)

Loss function in Deep Ritz method

\[ I[u] = \int_{\Omega} \frac{1}{2} (\nabla u)^T A(x) \nabla u(x) \, dx + \sum_{i=1}^{d} \int_{-1}^{1} \delta(x_i) u(x) \, dx_i \]

\[ + \eta \int_{\partial \Omega} (u(x) - g(x))^2 \, dx, \]

<table>
<thead>
<tr>
<th>d</th>
<th>n</th>
<th>m</th>
<th>Activation-Optimizer</th>
<th>(L^2) error</th>
<th>(L^\infty) error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>20</td>
<td>2</td>
<td>ReLu-Adam</td>
<td>1.23 e-02</td>
<td>9.91 e-02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ReQu-Adam</td>
<td>2.22 e-02</td>
<td>4.21 e-01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sigmoid-Adam</td>
<td>2.19 e-02</td>
<td>3.14 e-01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(</td>
<td>x</td>
<td>^{0.5}) - Adam-CBO</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>2</td>
<td>ReLu-Adam</td>
<td>6.72 e-03</td>
<td>3.70 e-01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ReQu-Adam</td>
<td>1.43 e-02</td>
<td>1.10 e-00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sigmoid-Adam</td>
<td>7.90 e-03</td>
<td>7.66 e-02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(</td>
<td>x</td>
<td>^{0.5}) - Adam-CBO</td>
</tr>
</tbody>
</table>

Table: Errors in \(L^2\) and \(L^\infty\) norms by Adam and Adam-CBO methods
Numerical results

Machine learning tasks

Cont’d

Figure: Training process of Adam and Adam-CBO methods when $d = 4$. 

(a) Absolute $L^\infty$ error 

(b) Relative $L^2$ error
Singularities

Figure: One-dimensional solution profiles at the intersection
Conclusion

Adam-CBO is

- able to find the global minimizer in high dimensions
- free of curse of dimensionality
- suitable for machine learning tasks with
  - gradient explosion or vanishing
  - non-different activation functions

Thank you for your attention!