A consensus-based global optimization method with adaptive momentum estimation

December 13, 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

Optimization methods: Zero-order or first-order?

CBO method

Adam-CBO Method

Linear stability analysis of Adam-CBO

Numerical results
- Rastirgin function
- Machine learning tasks

First-order methods

- gradient descent method

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

  with α 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]

$$\begin{aligned} x^{t+1} &= x^t - m^t \\ m^t &= -\gamma m^{t-1} + \alpha \nabla f_i(x^t) \end{aligned}$$

[1] Ning Qian. "On the momentum term in gradient descent learning algorithms". In: *Neural Networks* 12.1 (1999), pp. 145–151. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(98)00116-6.

Cont'd

- Adaptive momentum method (Adam)[2]

$$x^{t+1} = x^t - \gamma \frac{\hat{m}^t}{\sqrt{\hat{v}^t} + \epsilon}$$

$$m^t = \beta_1 m^{t-1} + (1-\beta_1)\nabla f(x^t), \quad \hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$v^t = \beta_2 v^{t-1} + (1-\beta_2)\nabla^2 f(x^t), \quad \hat{v}_t = \frac{v_t}{1-\beta_1^t}$$

where $0 < \beta_1, \beta_2 < 1$

[2] Diederik P. Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

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]

[3] Boris Hanin. "Which neural net architectures give rise to exploding and vanishing gradients?" In: *Advances in neural information processing systems*. 2018, pp. 582–591.

[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[5][6][7]

[5] José A. Carrillo et al. "An analytical framework for consensus-based global optimization method". In: *Mathematical Models and Methods in Applied Sciences* 28.06 (2018), pp. 1037–1066.

[6] Claudia Totzeck et al. "A Numerical Comparison of Consensus-Based Global Optimization to other Particle-based Global Optimization Schemes". In: *PAMM* 18.1 (2018), e201800291.

[7] René Pinnau et al. "A consensus-based model for global optimization and its mean-field limit". In: *Mathematical Models and Methods in Applied Sciences* 27.01 (2017), pp. 183–204.

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

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

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]

$$X^i_{t+1} = X^i_t - \lambda(X^i - \bar{x}^*) + \sigma(X^i - \bar{x}^*)dW^i_t$$

  Random mini-batch: $\mathcal{O}(N) \to \mathcal{O}(\frac{N}{M})$
- Convergence to the global minimizer with dimension-independent parameters[10]

[8] José A. Carrillo et al. "An analytical framework for consensus-based global optimization method". In: *Mathematical Models and Methods in Applied Sciences* 28.06 (2018), pp. 1037–1066.

[9] José A. Carrillo et al. "A consensus-based global optimization method for high dimensional machine learning problems". In: *arXiv preprint arXiv:1909.09249* (2019).

[10] Seung-Yeal Ha, Shi Jin, and Doheon Kim. "Convergence of a first-order consensus-based global optimization algorithm". In: *arXiv preprint arXiv:1910.08239* (2019).

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

$$\begin{aligned} \dot{X}_t^i &= -M_t^i \\ \sigma \dot{M}_t^i + M_t^i &= X_t^i - x^* \end{aligned}$$

Discretization

$$\begin{aligned} X_{t+1}^i &= X_t^i - \delta t M_{t+\frac{1}{2}}^i \\ M_{t+\frac{1}{2}}^i &= \frac{\sigma - \delta t}{\sigma + \delta t} M_{t-\frac{1}{2}} + \frac{2\delta t}{\sigma + \delta t}(X_t^i - x^*) \end{aligned}$$

Cont'd

Relabel $M^i_{t+\frac{1}{2}}$ by M^i_{t+1}

$$X^i_{t+1} = X^i_t - \lambda M^i_{t+1}$$
$$M^i_{t+1} = \beta_1 M^i_t + (1-\beta_1)(X^i_t - \bar{x}^*)$$

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

β_1 is near 1 ($= 0.9$ in practice) since δt is small

Add the stochastic term

$$X^i_{t+1} = X^i_t - \lambda M^i_{t+1} + \sigma_t W^i_t$$
$$M^i_{t+1} = \beta_1 M^i_t + (1-\beta_1)(X^i_t - \bar{x}^*)$$

Expectation

A recursive argument of M_t^i yields

$$\begin{aligned} M_t^i &= \beta_1 M_{t-1}^i + (1-\beta_1)(X_{t-1}^i - x^*) \\ &= \beta_1(\beta_1 M_{t-2}^i + (1-\beta_1)(X_{t-1}^i - x^*)) + (1-\beta_1)(X_{t-2}^i - x^*) \\ &= (1-\beta_1)\sum_{k=0}^{t-1} \beta_1^{t-k}(X_k^i - x^*). \end{aligned}$$

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

$$\begin{aligned} \mathbb{E}[M_t^i] &= (1-\beta_1)\mathbb{E}[\sum_{k=0}^{t} \beta_1^{t-k}(X_k^i - x^*)] \\ &= (1-\beta_1^t)\mathbb{E}[X_t^i - x^*] \end{aligned}$$

Unbiased estimation of first-order moment

$$\hat{M}_{t+1}^i = \frac{M_{t+1}^i}{(1-\beta_1^t)}$$

Second-order momentum $\mathbb{E}(|X_t^i - x^*|^2)$

- Define $V_t^i = \beta_2 V_{t-1}^i + (1-\beta_2)|X_t^i - x^*|^2$
  Application of the same argument for $\mathbb{E}[X_t^i]$ yields

$$\mathbb{E}[V_t^i] = (1-\beta_2^t)\mathbb{E}[|X_t^i - x^*|^2]$$

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

$$\hat{V}_t^i = \frac{V_t^i}{1-\beta_2^t}$$

- Modify the model

$$X_{t+1}^i = X_t^i - \frac{\lambda \hat{M}_{t+1}^i}{\sqrt{\hat{V}_{t+1}^i} + \epsilon} + \sigma^t W_t^i$$

  with a small ϵ $(1e-8)$ to avoid the vanishing of denominator

Input: λ, N, M, t_N, β_1, β_2

Initialize X_0^i, $i = 1, \cdots N$ by the uniform distribution;

Initial M_0^i, $V_0^i = 0$; `/* Initialize first and second order moments. */`

for $t = 0$ to t_N **do**

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

    Generate batch set of particles in order of P_k as $B^1, \cdots B^{\frac{N}{M}}$ with each batch having M particles;

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

        Update $x^* = \sum\limits_{k \in B^j} \frac{X_t^k \mu_t^k}{\sum\limits_{i \in B^j} \mu_t^i}$, where $\mu_t^i = \omega_f^\alpha(X_t^i)$;

        Update X_t^i for $j \in B^j$ as follows

        $M_{t+1}^i = \beta_1 M_t^i + (1-\beta_1)(X_t^i - x^*) \qquad \hat{M}_{t+1}^i = M_{t+1}^i/(1-\beta_1^t)$;

        $V_{t+1}^i = \beta_2 V_t^i + (1-\beta_2)(X_t^i - x^*)^2 \qquad \hat{V}_{t+1}^i = V_{t+1}^i/(1-\beta_2^t)$;

        $X_{t+1}^i = X_t^i - \lambda \hat{M}_t^i / (\sqrt{\hat{V}_t^i} + \epsilon) + \sigma^t \sum_{k=1}^d \vec{e}_k z_i$.

    end

end

Output: $X_{t_N}^i, \quad i = 1 \cdots N$

Linear stability analysis of Adam-CBO

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 $m = 0, x = \bar{x}, v = 0$

$$
\begin{aligned}
\dot{m} &= -(1-\beta_1)m + (1-\beta_1)\tilde{x} \\
\dot{v} &= -(1-\beta_2)v \\
\dot{\tilde{x}} &= -\frac{\lambda}{(1-\beta_1^t)\epsilon} m \to -\frac{\lambda}{\epsilon} m = -\mu m \quad (t \to \infty)
\end{aligned}
$$

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

$$
\mathrm{d}_t \begin{pmatrix} m \\ v \\ \tilde{x} \end{pmatrix} = \begin{pmatrix} -(1-\beta_1) & 0 & 1-\beta_1 \\ 0 & -(1-\beta_2) & 0 \\ -\mu & 0 & 0 \end{pmatrix} \begin{pmatrix} m \\ v \\ \tilde{x} \end{pmatrix}
$$

Theorem

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

Proof.

Eigenvalues of the matrix on the right-hand side are $\beta_2 - 1$ and $\frac{1}{2}(\beta_1 - 1 \pm i\sqrt{1-\beta_1}\sqrt{\beta_1 - 1 + 4\mu})$ (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 formuation of CBO without random noise

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

- The decay rate of the CBO method depends exponentially on the learning rate λ

Rastrigin function

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

with $B = \arg\min f(x)$ and $C = \min f(x)$

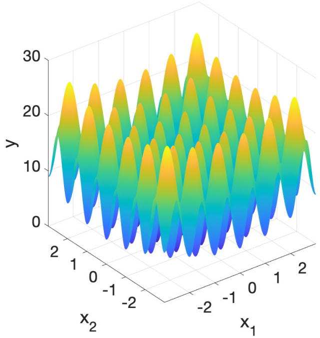


Figure: $d = 2$ and $B = C = 0$

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 $d = 1000$

d	1	2	30	100	1000
Number of local minima	5	5^2	5^{30}	5^{100}	5^{1000}

Table: Number of local minima in terms of dimension

Comparison with different random processes

d	N	M	CBO		
			$\mathcal{N}(0,1)$	$\mathcal{U}(-1,1)$	Wiener process
2	50	40	100%	100%	99%
10	50	40	100%	100%	2%
20	50	40	98%	22%	0%
20	50	20	66%	2%	0%
30	50	40	26%	0%	0%
30	500	5	0%	0%	0%
d	N	M	Adam-CBO		
			$\mathcal{N}(0,1)$	$\mathcal{U}(-1,1)$	Wiener process
30	500	5	99%	100%	0%
100	5000	5	100%	100%	0%
1000	8000	50	92%	20%	0%

$\lambda = 0.1$, and $\sigma^t = 0.99^{\frac{t}{20}}$

d	N	M	Adam-CBO	
			$\mathcal{N}(0,1)$	$\mathcal{U}(-1,1)$
1000	8000	50	92%	20%
1000	10000	50	100%	28%
1000	12000	50	100%	28%
1000	14000	50	100%	32%
1000	16000	50	100%	32%

Table: Different numbers of particles when the dimension is 1000

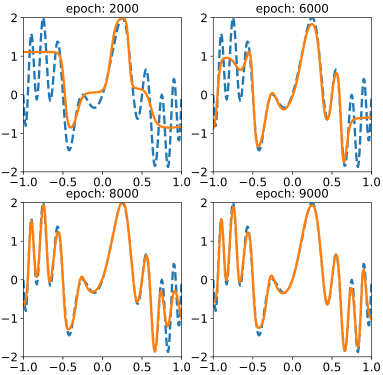
d	N	M	Adam-CBO	
			$\mathcal{N}(0,1)$	$\mathcal{U}(-1,1)$
30	500	5	94%	100%
100	5000	5	100%	94%
1000	10000	50	100%	11%

Table: Different dimensions when X_t^i is initialized by 0 ($X_0^i = 0$)

Spectrail bias[11]/Frequency principle[12]

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

- 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 $M = 10$

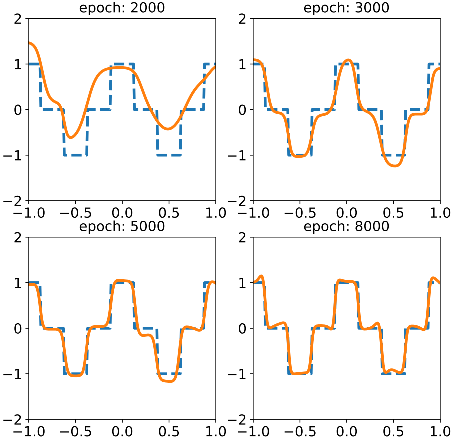


[11] Nasim Rahaman et al. "On the Spectral Bias of Deep Neural Networks". In: *arXiv preprint arXiv:1806.08734* (2018).

[12] Zhi-Qin John Xu et al. "Frequency principle: Fourier analysis sheds light on deep neural networks". In: *arXiv preprint arXiv:1901.06523* (2019).

$$u(x) = \begin{cases} 1 & x < -\frac{7}{8}, x > \frac{7}{8}, -\frac{1}{8} < x < \frac{1}{8} \\ -1 & \frac{3}{8} < x < \frac{5}{8}, -\frac{5}{8} < x < -\frac{3}{8} \\ 0 & \text{otherwise} \end{cases}$$

The same setup as in the previous slide



Gradient exploding or vanishing: DNN with fixed width 10 and different depths

$$u(x) = \sin(k\pi x^k)$$

$N = 500$, $M = 5$

depth	Num of parameters	k = 2	k = 3	k = 4
4	141	6.62 e-03	1.32 e-02	1.71 e-01
7	471	4.78 e-03	1.42 e-02	7.54 e-03
12	1021	7.44 e-03	1.30 e-02	5.32 e-02
22	2121	1.00 e-02	1.01 e-02	1.21 e-01

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

Solving PDEs by DNNs: Deep Ritz method[13]

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

with

$$A(x) = \begin{bmatrix} (x_1^2)^{\frac{1}{4}} & & \\ & \ddots & \\ & & (x_d^2)^{\frac{1}{4}} \end{bmatrix}.$$

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

[13] E Weinan and Bing Yu. "The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems". In: *Communications in Mathematics and Statistics* 6.1 (2018), pp. 1–12.

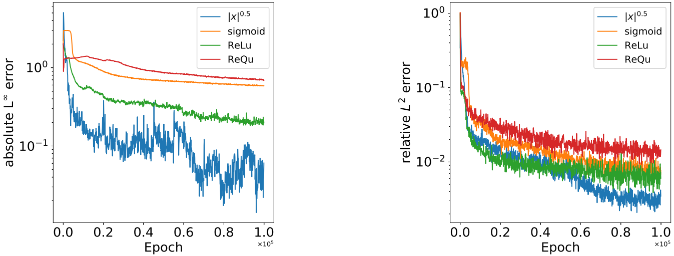
Loss function in Deep Ritz method

$$
\begin{aligned}
I[u] = &\int_{\Omega} \frac{1}{2}(\nabla u)^T A(x) \nabla u(x) \mathrm{d}x + \sum_{i=1}^{d} \int_{-1}^{1} \delta(x_i) u(x) \mathrm{d}x_i \\
&+ \eta \int_{\partial\Omega} (u(x) - g(x))^2 \mathrm{d}x,
\end{aligned}
$$

d	n	m	Activation-Optimizer	L^2 error	L^∞ error
2	20	2	ReLu-Adam	1.23 e-02	9.91 e-02
			ReQu-Adam	2.22 e-02	4.21 e-01
			sigmoid-Adam	2.19 e-02	3.14 e-01
			$\lvert x\rvert^{0.5}$ - Adam-CBO	3.96 e-03	2.09 e-02
4	40	2	ReLu-Adam	6.72 e-03	3.70 e-01
			ReQu-Adam	1.43 e-02	1.10 e -00
			sigmoid-Adam	7.90 e-03	7.66 e -02
			$\lvert x\rvert^{0.5}$ -Adam-CBO	3.13 e-03	9.52 e -02

Table: Errors in L^2 and L^∞ norms by Adam and Adam-CBO methods

Cont'd

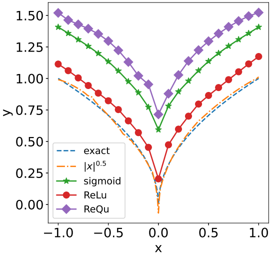


(a) L^{∞} error

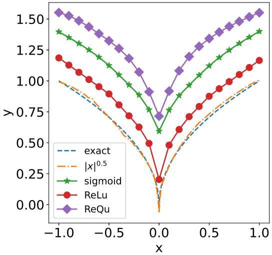
(b) L^{2} error

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

Singularities



(a) $x_2 = x_3 = x_4 = 0$



(b) $x_1 = x_3 = x_4 = 0$

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!