A CONSENSUS-BASED GLOBAL OPTIMIZATION METHOD WITH ADAPTIVE MOMENTUM ESTIMATION

December 11, 2020

Outline

Why Zero Order Method

CBO Method

Adam CBO Method

Numerical Example

- Rastirgin Function
- DNN Approximate function
- DNN Solving PDE

Why Zero Order Method

First Order Method (gradient based) is widely used in Machine learning problems, including SGD, SGD momentum, AdaGrad, and Adam and so forth. The problems of Gradient based method

- Most gradient-based methods have problems dealing with functions that have large noise or non-differentiable functions.
- It has been proved that as the deep neural network gets deeper, the gradient tends to explode or vanish.[1]
- It will be easily influenced by the geometry of the landscape[2]

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

[2] 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).

Gradient-free methods

There are also gradient-free methods such as

- Nelder-Mead (NM)
- genetic algorithm (GA)
- simulated annealing (SA)
- particle swarm optimization (PSO)

The original CBO method

CBO[3][4][5] is based on interacting particle system, along the line of consensus based models. During the dynamic evolution, the particle system tends to their weighted average, and meanwhile undergoes some fluctuation due to the random noise, such as the isotropic geometric Brownian motion

[3] 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.

[4] 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.

[5] 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.

Consider N particles, labeled as X^i, $i = 1, \cdots N$,

$$\dot{X}^i = -\lambda(X^i - \bar{x}^*) + \sigma|X^i - \bar{x}^*|\dot{W}^i \tag{1}$$

$\bar{x}^* = \frac{1}{\sum_{i=1}^N e^{-\beta L(X^i)}} \sum_{i=1}^N X^i e^{-\beta L(X^i)}$: weighted average of the position of the particle
L: cost function
$\dot{W}$: white noise.
Discretize the above system as follows

$$X_{t+1}^i = X_t^i - \lambda(X^i - \bar{x}^*) + \sigma|X^i - \bar{x}^*|W^i \tag{2}$$

Curse of Dimension of original CBO method

The convergence was proved in with exponential rate in time under dimension-dependent conditions, i.e., the learning rate depends on the dimension.
Therefore, the CBO method may suffer from the curse of dimensionality.
To overcome this issue,[6] proposed to replace the isotropic geometric Brownian motion with the component-wise one.
Such a modification leads to the convergence to the global minimizer with dimension-independent parameters.

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

Problem of CBO

- Dependent on the initial data
- Hard to optimize high dimensional no-convex function, like Rastrigin Function over 20 dimension.
- Hard to optimize deep neural network, for too much parameters.

First Order Momentum

We consider model

$$\sigma \ddot{X}_t^i + \dot{X}_t^i = -(X_t - x^*), \quad i = 1, \cdots, N, \tag{3}$$

It can be written into a first order PDE system

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

can be further simplified as

$$X_{t+1}^i = X_t^i - \delta t M_{t+\frac{1}{2}}^i, \tag{4}$$

$$M_{t+\frac{1}{2}}^i = \frac{\sigma - \delta t}{\sigma + \delta t} M_{t-\frac{1}{2}} + \frac{2\delta t}{\sigma + \delta t}(X_t^i - x^*). \tag{5}$$

It's trivial that we can find parameters that satisfy the $\lambda = \delta t$ and $\beta_1 = \frac{\sigma - \delta t}{\sigma + \delta t} = 1 - \frac{2\delta t}{\sigma + \delta t}$.

$$\begin{aligned} X_{t+1}^i &= X_t^i - \lambda M_{t+1}^i \\ M_{t+1}^i &= \beta_1 M_t^i + (1-\beta_1)(X_t^i - \bar{x}^*). \end{aligned} \tag{6}$$

Since δt is a small number, we have β_1 is near 1 ($= 0.9$ in practice). Similar with the CBO method, we add the stochastic terms in the model,

$$\begin{aligned} X_{t+1}^i &= X_t^i - \lambda M_{t+1}^i + \sigma_t W_t^i \\ M_{t+1}^i &= \beta_1 M_t^i + (1-\beta_1)(X_t^i - \bar{x}^*) + \gamma_t \xi_t^i. \end{aligned} \tag{7}$$

Notice that the ξ_t don't need to add artifically, since the calculation of $\bar{x}^*$ in optimization with random sampling, already introduce stochastic term in momentum equation.

The connection between CBO momentum method and original one

In Exception means,

$$\begin{aligned} X_{t+1}^i &= X_t^i - \lambda M_{t+1}^i + \sigma_t W_t^i \\ &\downarrow \\ \dot{X}_{t+1}^i &= X_t^i - \lambda(X^i - \bar{x}^*) + \sigma|X^i - \bar{x}^*|W^i \end{aligned} \tag{8}$$

The M_t^i can is the moving average $X_t^i - x^*$

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

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

To get an unbiased estimation of $(X_k^i - x^*)$ for small t as well, we rescale M_t^i by $(1-\beta_1^t)$ and denote by $\hat{M}_t^i$. This shows the our method and original one is the same in Exceptation means.

Second Order Momentum

For the second order moment $\mathbb{E}(|X_t^i - x^*|^2)$[7], we define

$$V_t^i = \beta_2 V_{t-1}^i + (1-\beta_2)|X_t^i - x^*|^2. \tag{9}$$

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

$$\mathbb{E}[V_t^i] = (1-\beta_2^t)\mathbb{E}[|X_t^i - x^*|^2], \tag{10}$$

and $\hat{V}_t^i = \frac{V_t^i}{1-\beta_2^t}$ is an unbiased estimation of $\mathbb{E}[|X_t^i - x^*|^2]$. Therefore, we modify the model into

$$X_{t+1}^i = X_t^i - \frac{\lambda \hat{M}_{t+1}^i}{\sqrt{\hat{V}_{t+1}^i} + \epsilon} + \sigma^t W_t^i, \tag{11}$$

where ϵ is a small number and typically takes the value $1e-8$ to avoid the vanishing of the denominator.

[7] The square here is defined in the element-wise sense.

```
Input: $\lambda$, $N$, $M$, $t_N$, $\beta_1$, $\beta_2$
Initialize $X_0^i$, $i = 1, \cdots N$ by the uniform distribution;
Initial $M_0^i, V_0^i = 0$; /* Initialize first order 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
        $M_{t+1}^i = \beta_1 M_t^i + (1 - \beta_1)(X_t^i - x^*)$      $\hat{M}_{t+1}^i = M_{t+1}^i/(1 - \beta_1^t)$;
        $V_{t+1}^i = \beta_2 V_t^i + (1 - \beta_2)(X_t^i - x^*)^2$      $\hat{V}_{t+1}^i = V_{t+1}^i/(1 - \beta_2^t)$;
        $X_{t+1}^i = X_t^i - \lambda \hat{M}_t^i/(\sqrt{\hat{V}_t^i} + \epsilon) +$
          $\sigma^t \sum_{k=1}^{d} \vec{e}_k z_i$   $z_i$  is a random variable.
    end
end
Output: $X_{t_N}^i$,   $i = 1 \cdots N$
```

A linear stability analysis of the Adam-CBO method

We first rewrite Algorithm into a continuous form and ignore the stochastic term

$$\dot{m} = (\beta_1 - 1)m + (1 - \beta_1)(x - \bar{x}), \tag{12}$$

$$\dot{v} = (\beta_2 - 1)v + (1 - \beta_2)(x - \bar{x})^2, \tag{13}$$

$$\hat{m} = \frac{m}{1 - \beta_1^t} \quad \hat{v} = \frac{v}{1 - \beta_2^t}, \tag{14}$$

$$\dot{x} = -\lambda \frac{\hat{m}}{\sqrt{\hat{v}} + \epsilon}, \tag{15}$$

Denote $\tilde{x} = x - \bar{x}$. Linearizing the system around $m = 0, x = \bar{x}, v = 0$, we have

$$\dot{m} = -(1-\beta_1)m + (1-\beta_1)\tilde{x}, \tag{16}$$

$$\dot{v} = -(1-\beta_2)v, \tag{17}$$

$$\dot{\tilde{x}} = -\frac{\lambda}{(1-\beta_1^t)\epsilon} m \to -\frac{\lambda}{\epsilon} m = -\mu m \quad (t \to \infty) \tag{18}$$

with $\mu = \lambda/\epsilon$, and in a vector form,

$$\partial_t \begin{pmatrix} m \\ v \\ \tilde{x} \end{pmatrix} = \begin{pmatrix} -(1-\beta_1) & 0 & 1-\beta_1 \\ 0 & -(1-\beta_2) & 0 \\ -\mu & 0 & 0 \end{pmatrix} \begin{pmatrix} m \\ v \\ \tilde{x} \end{pmatrix}. \tag{19}$$

Theorem

Algorithm *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.* □

The CBO method without random noise can be written into a continuous form

$$\dot{x} = -\lambda(x - \bar{x}). \tag{20}$$

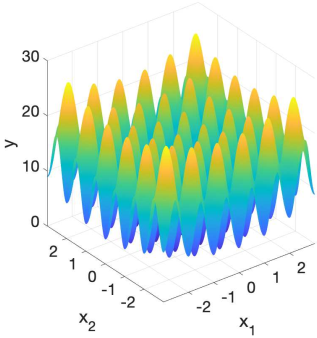
The ODE can be solved analytically with a decay rate $e^{-\lambda t}$ towards the stationary point. Therefore, the decay rate of the CBO method depends exponentially on the learning rate λ.

Rastrigin Function

Finding the global minimizer of the Rastrigin function

$$f(x) = \frac{1}{d}\sum_{i=1}^{d}\left[(x_i - B)^2 - 10\cos(2\pi(x_i - B)) + 10\right] + C \qquad (21)$$

with $B = \arg\min f(x)$ and $C = \min f(x)$. Figure is a visualization of (21) when $d = 2$ and $B = C = 0$.



The number of local minima is 5^d, which grows exponentially fast in term of the dimensionality. When $d = 1000$, the number of minima is 5^{1000}, approximately 10^{690}.

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

Table: Number of local minima for the Rastrigin function in terms of dimension.

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%

Table: Comparison of CBO and Adam-CBO methods with different random processes.

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: Comparison of success rates for different dimensions when X_t^i is initialized by 0 ($X_0^i = 0$), $\lambda = 0.1$, and $\sigma^t = 0.99^{\frac{t}{20}}$.

Frequency Principle

Consider two functions

$$u(x) = \sin(2\pi x) + \sin(8\pi x^2), \tag{22}$$

$$u(x) = \begin{cases} 1 & x < -\frac{7}{8}, x > \frac{7}{8}, -\frac{1}{8} < x < \frac{1}{8} \\ -1 & \frac{3}{8} < x < \frac{5}{8}, -\frac{5}{8} < x < -\frac{3}{8} \\ 0 & \text{otherwise} \end{cases}, \tag{23}$$

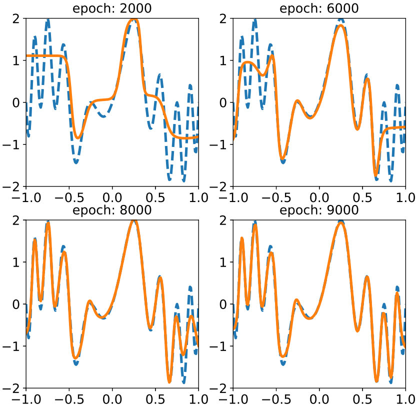


Figure: Approximating function (22) using a network with width $= 50$, depth $= 3$, and 2701 parameters in total. The learning rate is $\lambda = 0.2$. $N = 500$ particles and $M = 5$ particles for each batch are used in the first 50000 iterations. After that, the random term is ignored and $M = 10$ is used for faster convergence to the optimal solution.

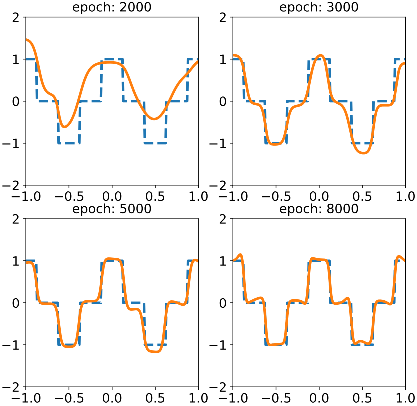


Figure: Approximating function (23) using a network with $n = 50$, $m = 3$, and 2701 parameters in total. The learning rate is $\lambda = 0.2$. $N = 500$ particles and $M = 5$ particles for each batch are used in the first 50000 iterations. After that, the random term is ignored and $M = 10$ is used for faster convergence to the optimal solution.

We use DNNs with a fixed width 10 and different depths to approximate the function

$$u(x) = \sin(k\pi x^k). \tag{24}$$

Set $N = 500$ particles, $M = 5$ particles for each batch.

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: Dependence of approximation error measured in absolute L^2 norm in terms of network depth for (24) when $k = 2, 3, 4$.

SGD or Adam will failed to converges well (with final error around 0.3 in absolute L^2 norm) when the network depth is 4 and 10, respectively.

DNN Solving PDE

We adopt the Deep Ritz method (DRM), which is based on the variational formulation associated to the PDE. Consider an elliptic PDE

$$\begin{cases} -\nabla \cdot (A(x)\nabla u) = -\sum_{i=1}^{d} \delta(x_i) & x \in \Omega = [-1,1]^d \\ u(x) = g(x) & x \in \partial\Omega \end{cases} \tag{25}$$

with

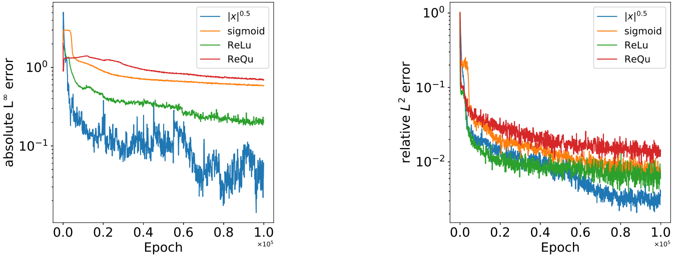
$$A(x) = \begin{bmatrix} (x_1^2)^{\frac{1}{4}} & & \\ & \ddots & \\ & & (x_d^2)^{\frac{1}{4}} \end{bmatrix}. \tag{26}$$

The exact solution $u(x) = \sum_{i=1}^{d} |x_i|^{\frac{1}{2}}$. One can see that the solution is only in $H^{1/2}(\Omega)$ and has singularities when evaluating its derivative at $x_i = 0$. The loss function in DRM reads as

$$\begin{aligned} I[u] = & \int_{\Omega} \frac{1}{2}(\nabla u)^T A(x) \nabla u(x) \mathrm{d}x + \sum_{i=1}^{d} \int_{-1}^{1} \delta(x_i) u(x) \mathrm{d}x_i \\ & + \eta \int_{\partial\Omega} (u(x) - g(x))^2 \mathrm{d}x, \end{aligned} \tag{27}$$

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
			$\|x\|^{0.5}$ - 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
			$\|x\|^{0.5}$ -Adam-CBO	3.13 e-03	9.52 e -02

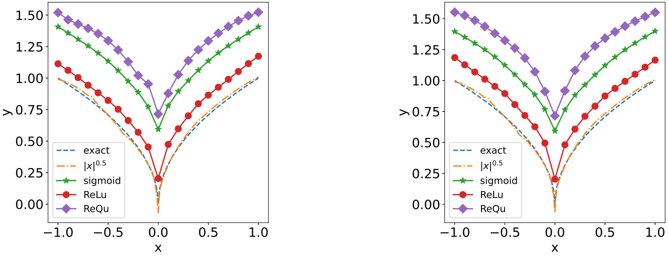
Table: Errors measured in L^2 and L^∞ norms for (25) by Adam and Adam-CBO methods.



(a) L^∞ error

(b) L^2 error

Figure: Training process of Adam and Adam-CBO methods for (25) when the dimension is 4. (a) L^∞ error; (b) L^2 error.



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

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

Figure: One-dimensional solution profiles at the intersection. (a) $x_2 = x_3 = x_4 = 0$; (b) $x_1 = x_3 = x_4 = 0$;