SGD Momentum based on Inter-gradient Collision

Journal:	*Transactions on Pattern Analysis and Machine Intelligence*
Manuscript ID	TPAMI-2022-02-0244
Manuscript Type:	Short
Keywords:	Deep Neural Networks, Optimization Algorithm, SGD, Adam

SGD Momentum based on Inter-gradient Collision

Weidong Zou*, Yuanqing Xia , Weipeng Cao, Gao Huang, Yizeng Han, and Xinwang Liu.

Abstract—Deep neural networks (DNNs) are widely used in every field, such as computer vision and natural language processing. And the optimizer is the main part of DNNs training. SGD-Momentum performs well in many DNNs methods (ResNet and DenseNet) because of its simpleness and currency, which is the most general optimizer at present. Even so, the slow convergent rate of SGD-Momentum has extremely restricted its application. Inter-gradient collision is integrated into SGD-Momentum to improve convergent rate, which is inspired by the elastic collision model in physics. And we term it SGD-Momentum based on inter-gradient collision (ICSGD-Momentum). We also give a theoretical proof of convergence and a regret bound on the ICSGD-Momentum method. Extensive experiments on function optimization, CIFAR-100, ImageNet, Penn Treebank, COCO, and YCB-Video show that ICSGD-Momentum can accelerate the training process and improve the generalization performance of DNNs compared to SGD-Momentum, Adam, RAdam, Adabound, and AdaBelief.

Index Terms—Deep Neural Networks, SGD, Adam, optimization algorithm.

1 INTRODUCTION

OWING to the availability of massive machine learning data-sets such as TEyeD [1], ImageNet [2], deep neural networks (DNNs) based on complicated neural network structure have made considerable progress such as ResNet [3], and DenseNet [4]. Despite the success of the experiment and application for DNNs, massive public data-sets, powerful computing resources and advanced optimization algorithms are used to train DNNs which is time-consuming. Therefore, how to accelerate the training of DNNs is becoming a new research topic.

The core of the training of DNNs is optimizer. At present, optimizer of DNNs is categorized into three types by optimization strategy, which are acceleration strategy (SGD, SGD-Momentum, PID), adaptation strategy (AdaGrad, RMSProp), acceleration and adaptation strategy (Adam, Adabound) as show in Table 1, which $\mathbf{g}_t$ denotes gradients of stochastic objective for DNNs at time-step t, $\boldsymbol{\phi}_1(t) = \sum_{i=1}^{t} \mu_1^{t-i}\mathbf{g}_i$, $\boldsymbol{\phi}_2(t) = \sum_{i=1}^{t} \mu_1^{t-i}(\mathbf{g}_i - \mathbf{g}_{i-1})$, $\boldsymbol{\phi}_3(t) = \sum_{i=1}^{t} \mathbf{g}_i \odot \mathbf{g}_i$, $\boldsymbol{\phi}_4(t) = \sum_{i=1}^{t} \mu_2^{t-i}\mathbf{g}_i \odot \mathbf{g}_i$, $\varphi_1(t) = \sum_{i=1}^{t} \mu_1^{t-i}$, $\varphi_2(t) = \sum_{i=1}^{t} \mu_2^{t-i}$, $\varphi_3(t) = \frac{(1-\mu_2)t}{(1-\mu_2)t+1}$, $\boldsymbol{\phi}_5(t) = Clip(\frac{1}{\sqrt{(1-\mu_2)\boldsymbol{\phi}_4(t)}}, \frac{\varphi_3(t)}{10}, \frac{1}{10\varphi_3(t)})$, $\mu_1, \mu_2 \in [0, 1)$ are the hyper-parameters of optimization methods, which adjusts the exponential decay rates of these moving averages. δ is a considerably small constant ($\delta = 10^{-8}$), and K_d is hyper-parameter of PID based optimization method, which can be tuned by using the theory of Laplace Transform with Ziegler-Nichols.

This work was supported by National Natural Science Foundation of China (61906015, 61836001, 62106150).
Weidong Zou and Yuanqing Xia are with School of Automation, Beijing Institute of Technology, Beijing 100081, China e-mail: zouweidong1985@163.com, xia_yuanqing@bit.edu.cn.
Weipeng Cao is with College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China e-mail: caoweipeng123@gmail.com.
Gao Huang and Yizeng Han are with Department of Automation, Tsinghua University, Beijing 100084, China e-mail:gaohuang@mail.tsinghua.edu.cn, hanyz18@mails.tsinghua.edu.cn.
Xinwang Liu is with School of Computer, National University of Defense Technology, Changsha, Hunan, China. e-mail: xinwangliu@nudt.edu.cn.
**Corresponding author: Weidong Zou.*

The series of adaptive methods (adaptation strategy, acceleration and adaptation strategy) adjust the learning rate according to the gradient value of the independent variable in each dimension, thereby avoiding the problem that the unified learning rate is difficult to adapt to all dimensions. Many cases show that adaptive methods can accelerate the training speed of DNNs.

In spite of their popularity, with the in-depth study of the related optimization algorithms, researchers find that these adaptive optimization algorithms are easy to generate extreme learning rates during the training process, which will seriously affect the stability of the model, resulting in the model performance is not as good as that using SGD in some scenarios [10], [11].

As shown in Table 1, SGD-Momentum introduces first-order momentum on the basis of SGD for suppressing the oscillation of SGD. SGD and SGD-Momentum are simple and easy to implement, and they have been applied to many scenarios. According to [6], the strategy of PID is to consider present, past and changing information of gradients to optimize the parameters of DNNs. However, SGD-Momentum and PID suffer from the overshoot problem [6] that the value of parameters exceeds the value of target and can not change along the gradient direction.

One commonly quadratic function is used to test overshoot for SGD-Momentum and PID. The function can be defined as $f(x) = (x-1)^2 + 2$ which as shown in Fig. 1, and the search domain of quadratic function is $-5 \leq x \leq 7$. There is a global minimum of quadratic function: $\tilde{x} = 1$, $f(\tilde{x}) = 2$. The learning rate and number of iterations for SGD, SGD-Momentum and PID are set to 0.05 and 100 respectively. μ_1 of SGD-Momentum is 0.9, K_d of PID is 10.

As shown in Fig. 2, compared with SGD, the change trend of the evolution of value for SGD-Momentum and PID fluctuates greatly, and the convergence speed of PID is better than SGD-Momentum. But the results prove that SGD-Momentum and PID have obvious overshoot problem.

In order to solve overshoot problem of SGD-Momentum, we propose SGD-Momentum based on inter-gradient collision which is inspired by [12]. Compared with existing SGD-Momentum

TABLE 1: The overview of SGD, SGD-Momentum, PID, RMSProp, AdaGrad, Adam and Adabound

Strategy	Optimization Methods	Key Step
Accelerated	SGD	$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \lambda \mathbf{g}_t$
	SGD-Momentum [5]	$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \lambda(\mathbf{g}_t + \boldsymbol{\phi}_1(t))$
	PID [6]	$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \lambda(\boldsymbol{\phi}_1(t) + K_d(1-\mu_1)\boldsymbol{\phi}_2(t))$
Adaptive	AdaGrad [7]	$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \frac{\lambda}{\sqrt{\boldsymbol{\phi}_3(t)+\delta}} \odot \mathbf{g}_t$
	RMSProp [8]	$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \frac{\lambda}{\sqrt{(1-\mu_2)\boldsymbol{\phi}_4(t)+\delta}} \odot \mathbf{g}_t$
Accelerated and Adaptive	Adam [9]	$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \frac{\lambda}{\varphi_1(t)(\sqrt{\frac{\boldsymbol{\phi}_4(t)}{\varphi_2(t)}}+\delta)} \odot \boldsymbol{\phi}_1(t)$
	Adabound [10]	$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \lambda(1-\mu_1)\boldsymbol{\phi}_1(t) \odot \boldsymbol{\phi}_5(t)$

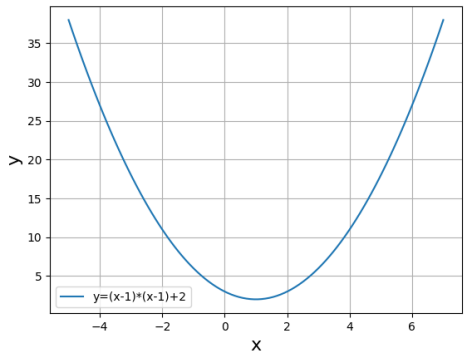


Fig. 1: Quadratic function

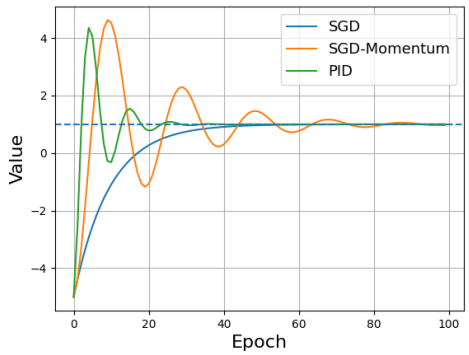


Fig. 2: Overshoot problem of SGD-Momentum and PID for quadratic function

method, ICSGD-Momentum introduces elastic collision factor α ($1 < \alpha < 2$) as its past and current gradients as $\alpha\mathbf{g}_t + (\alpha-1)\boldsymbol{\phi}_1(t)$ ($\mathbf{g}_t$ denotes gradients of stochastic objective for DNNs at time-step t, $\boldsymbol{\phi}_1(t) = \sum_{i=1}^{t} \mu_1^{t-i}\mathbf{g}_i$, $\mu_1 \in [0,1)$ are the hyper-parameters of optimization methods). Then we have proved the advanced nature of this improvement theoretically and experimentally. The contributions of this paper can be summarized as follow:

1) We propose a novel optimization method for DNNs, which is called as ICSGD-Momentum. Inter-gradient collision model will be constructed as $\alpha\mathbf{g}_t + (\alpha-1)\boldsymbol{\phi}_1(t)$ based on past and current gradients of loss function for DNNs.

2) The effectiveness of ICSGD-Momentum is verified on some classical data-sets (i.e., CIFAR-100, ImageNet, Penn Treebank, COCO and YCB-Video) and extensive experimental results show that ICSGD-Momentum can achieve state-of-the-art performance.

2 The Details of the Proposed ICSGD-Momentum Optimization Algorithm

2.1 Convergence Analysis for ICSGD-Momentum

ICSGD-Momentum is an improved SGD-Momentum whose basic idea is to build inter-gradient collision model as follows by introducing elastic collision factor α.

$$\mathbf{u}_t = \alpha\mathbf{g}_t + (\alpha-1)\sum_{i=1}^{t}\mu_1^{t-i}\mathbf{g}_i. \tag{1}$$

then we prove that ICSGD-Momentum has regret bound using the following theorems.

Theorem 2.1: Given the cost function $\mathbf{f}_t(\boldsymbol{\theta})$ of DNNs has bounded gradients, which can be represented as $\mathbf{g}_t = \bigtriangledown_{\boldsymbol{\theta}}\mathbf{f}_t(\boldsymbol{\theta}_t)$ and $\|\mathbf{g}_t\|^2 < G_1$, G_1 is a constant. Let the distance between any $\boldsymbol{\theta}_t$ generated by ICSGD-Momentum is bound, $\|\boldsymbol{\theta}_q - \boldsymbol{\theta}_p\|_2 < G_2$, where G_2 is a constant. For any $p, q \in [1, ..., T]$, $0 < \mu_1 < 1 < \alpha < 2$ and $\mu_1 + \alpha \leq 2$, we set $\tau_t = \frac{\tau}{\sqrt{t}}$, then ICSGD-Momentum brings the following guarantee (for all $T \geq 1$):

$$\begin{aligned}\mathbf{R}_T &= \sum_{t=1}^{T}[\mathbf{f}_t(\boldsymbol{\theta}_t) - \mathbf{f}_t(\boldsymbol{\theta}^*)] \\ &< G_2^2(\frac{\sqrt{T}}{\tau}+2) + \frac{\tau}{\sqrt{T}}(3+\alpha^2)\sum_{t=1}^{T}\|\mathbf{g}_t\|_2^2. \end{aligned}\tag{2}$$

where $\mathbf{f}_t(\boldsymbol{\theta}_t)$ is convex cost function at each time t, $\mathbf{f}_t(\boldsymbol{\theta}^*)$ is the best fixed point parameter from a feasible set $\boldsymbol{\chi}$ when $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\chi}}\sum_{t=1}^{T}\mathbf{f}_t(\boldsymbol{\theta})$ [9].

Proof. According to algorithm 1, we have

$$\begin{aligned}\|\mathbf{u}_t\|_2^2 &= \|\alpha\mathbf{g}_t + (\alpha-1)\sum_{i=1}^{t}\mu_1^{t-i}\mathbf{g}_i\|_2^2 \\ &< \alpha^2\|\mathbf{g}_t\|_2^2 + (\alpha-1)^2\sum_{i=1}^{t}\mu_1^{t-i}\|\mathbf{g}_i\|_2^2 \\ &< \alpha^2\|\mathbf{g}_t\|_2^2 + (1-\mu_1)\sum_{i=1}^{t}\mu_1^{t-i}\|\mathbf{g}_i\|_2^2. \end{aligned}\tag{3}$$

According to Lemma 2 of [13], we get

$$\sum_{t=1}^{T}\|\mathbf{u}_t\|_2^2 < \alpha^2\sum_{t=1}^{T}\|\mathbf{g}_t\|_2^2 + (1-\mu_1)\sum_{t=1}^{T}\sum_{i=1}^{t}\mu_1^{t-i}\|\mathbf{g}_i\|_2^2$$

$$= \alpha^2 \sum_{t=1}^{T} \|\mathbf{g}_t\|_2^2 + (1-\mu_1) \sum_{t=1}^{T} \sum_{t=i}^{T} \mu_1^{t-i} \|\mathbf{g}_i\|_2^2$$
$$< (1+\alpha^2) \sum_{t=1}^{T} \|\mathbf{g}_t\|_2^2. \quad (4)$$

According to Lemma 10.2 of [9], we have

$$\mathbf{f}_t(\boldsymbol{\theta_t}) - \mathbf{f}_t(\boldsymbol{\theta}^*) < \langle \mathbf{g}_t, \boldsymbol{\theta_t} - \boldsymbol{\theta}^* \rangle. \quad (5)$$

Furthermore,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \lambda(\alpha \mathbf{g}_t + (\alpha-1) \sum_{i=1}^{t} \mu_1^{t-i} \mathbf{g}_i). \quad (6)$$

Subtract $\boldsymbol{\theta}^*$ and square both side of Equ. (6), we get

$$\begin{aligned} \|\boldsymbol{\theta_{t+1}} - \boldsymbol{\theta}^*\|_2^2 &= \|\boldsymbol{\theta_t} - \boldsymbol{\theta}^* - \tau_t \mathbf{u}_t\|_2^2 \\ &= \|\boldsymbol{\theta_t} - \boldsymbol{\theta}^*\|_2^2 + \tau_t^2 \|\mathbf{u}_t\|_2^2 - 2\tau_t \langle \mathbf{u}_t, \boldsymbol{\theta_t} - \boldsymbol{\theta}^* \rangle \\ &= \|\boldsymbol{\theta_t} - \boldsymbol{\theta}^*\|_2^2 + \tau_t^2 \|\mathbf{u}_t\|_2^2 \\ &- 2\tau_t \alpha \langle \mathbf{g}_t, \boldsymbol{\theta_t} - \boldsymbol{\theta}^* \rangle \\ &- 2\tau_t(\alpha-1) \langle \sum_{i=1}^{t} \mu_1^{t-i} \mathbf{g}_i, \boldsymbol{\theta_t} - \boldsymbol{\theta}^* \rangle, \end{aligned} \quad (7)$$

then we have

$$\begin{aligned} \langle \mathbf{g}_t, \boldsymbol{\theta_t} - \boldsymbol{\theta}^* \rangle &= \frac{\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2^2 - \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|_2^2}{2\tau_t \alpha} \\ &+ \frac{\tau_t \|\mathbf{u}_t\|_2^2 - 2(\alpha-1) \langle \sum_{i=1}^{t} \mu_1^{t-i} \mathbf{g}_i, \boldsymbol{\theta_t} - \boldsymbol{\theta}^* \rangle}{2\alpha} \\ &< \frac{\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2^2 - \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|_2^2}{2\tau_t \alpha} \\ &+ \frac{\tau_t \|\mathbf{u}_t\|_2^2}{2\alpha} \\ &+ \frac{(\alpha-1)(\sum_{i=1}^{t} \mu_1^{t-i} \|\mathbf{g}_i\|_2^2 + \|\boldsymbol{\theta_t} - \boldsymbol{\theta}^*\|_2^2)}{\alpha} \\ &< \frac{\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2^2 - \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|_2^2}{2\tau_t \alpha} + \frac{\tau_t \|\mathbf{u}_t\|_2^2}{2\alpha} \\ &+ \frac{(1-\mu_1)(\sum_{i=1}^{t} \mu_1^{t-i} \|\mathbf{g}_i\|_2^2 + \|\boldsymbol{\theta_t} - \boldsymbol{\theta}^*\|_2^2)}{\alpha} \end{aligned} \quad (8)$$

Therefore, we have the following regret bound:

$$\begin{aligned} \mathbf{R}(T) &= \sum_{t=1}^{T} [\mathbf{f}_t(\boldsymbol{\theta_t}) - \mathbf{f}_t(\boldsymbol{\theta}^*)] \\ &\leq \sum_{t=1}^{T} \langle \mathbf{g}_t, \boldsymbol{\theta_t} - \boldsymbol{\theta}^* \rangle \\ &< \sum_{t=1}^{T} [\frac{\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2^2 - \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|_2^2}{2\tau_t \alpha} \\ &+ \frac{\tau_t \|\mathbf{u}_t\|_2^2 + 2(1-\mu_1)(\sum_{i=1}^{t} \mu_1^{t-i} \|\mathbf{g}_i\|_2^2 + \|\boldsymbol{\theta_t} - \boldsymbol{\theta}^*\|_2^2)}{2\alpha}] \\ &< \sum_{t=1}^{T} [\frac{\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2^2}{2\alpha} (\frac{1}{\tau_t} + 2(1-\mu_1)) \\ &+ \frac{\tau_t \|\mathbf{u}_t\|_2^2 + 2(1-\mu_1) \sum_{i=1}^{t} \mu_1^{t-i} \|\mathbf{g}_i\|_2^2}{2\alpha}] \\ &< \sum_{t=1}^{T} (G^2 (\frac{1}{\tau_t} + 2) + \tau_t \|\mathbf{u}_t\|_2^2 + 2(1-\mu_1) \sum_{i=1}^{t} \mu_1^{t-i} \|\mathbf{g}_i\|_2^2) \\ &< G^2 (\frac{\sqrt{T}}{\tau} + 2) + \frac{\tau}{\sqrt{T}} \sum_{t=1}^{T} \|\mathbf{u}_t\|_2^2 \\ &+ 2(1-\mu_1) \sum_{i=1}^{T} \sum_{t=i}^{T} \mu_1^{t-i} \|\mathbf{g}_i\|_2^2 \\ &< G_2^2 (\frac{\sqrt{T}}{\tau} + 2) + \frac{\tau}{\sqrt{T}} (3+\alpha^2) \sum_{t=1}^{T} \|\mathbf{g}_t\|_2^2. \end{aligned} \quad (9)$$

2.2 Pseudo-code for ICSGD-Momentum

The proposed ICSGD-Momentum algorithm can be summarized as follows:

Algorithm 1 ICSGD-Momentum

Input: Initial parameter vector $\boldsymbol{\theta}_0$, $\lambda = 0.1$, $\alpha = 1.1$ and $\mu_1 = 0.9$.
Output: The parameters $\boldsymbol{\theta}_T$ of the model.
for $q = 1$; $q \leq T$ **do**
  $\mathbf{g}_q = \bigtriangledown_{\boldsymbol{\theta}} \mathbf{f}_q(\boldsymbol{\theta}_{q-1})$;
  $\boldsymbol{\theta}_q = \boldsymbol{\theta}_{q-1} - \lambda(\alpha \mathbf{g}_q + (\alpha-1) \sum_{i=1}^{q} \mu_1^{q-i} \mathbf{g}_i)$.
end for

3 Simulation Experiments and Discussions

3.1 Setup of Experiments

In this section, we adopt function optimization, image classification, image recognition, language modeling with LSTM and object detection to compare of performance of ICSGD-Momentum with multiple optimization algorithms (e.g., SGD-Momentum, AdaBound, AdaBelief [14], RAdam [15]). All experiments are conducted under Pytorch 1.7 framework with NVIDIA TITAN RTX GPU.

3.2 Function Optimization

In order to simply verify the effectiveness of ICSGD-Momentum ($\alpha = 1.1$), based on Fig. 1 and 2, we add ICSGD-Momentum for quadratic function. The results of the experiment is shown in Fig. 3. As shown in Fig. 3, the overshoot value of ICSGD-Momentum is less than SGD-Momentum and PID, and the change trend of the evolution of value for ICSGD-Momentum fluctuates more smoothly than SGD-Momentum and PID.

3.3 Experiments on CIFAR-100

For adaptive optimization algorithms, we set $\mu_1 = 0.9$, $\mu_2 = 0.999$ and the initial learning rate $\lambda = 0.001$. The initial learning rate λ is set to 0.1 for SGD-Momentum and ICSGD-Momentum($\alpha = 1.1$). In our experiments, we employ the learning rate decay scheme at epoch 150 by multiplying 0.1. The number of epochs is 200. The cross-entropy is chosen as the loss function and the weight decay technique is applied on the parameters to prevent over-fitting. The mini-batch size is set to 128.

The results of running DenseNet-121 on CIFAR-100 are shown in Fig. 4 and Fig. 5, respectively. From Fig. 4 and Fig. 5, it can observe the following phenomena:

- RAdam can achieve faster convergence rate and better prediction performance than others in the early training phase of the model (i.e., before the learning rate decay

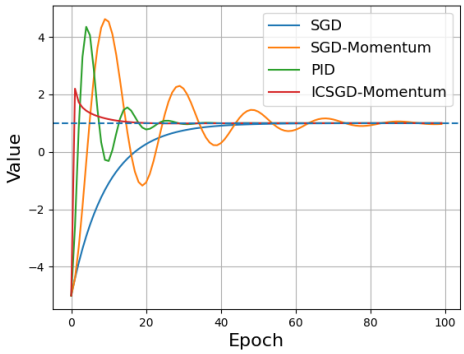


Fig. 3: Overshoot problem of SGD-Momentum, PID and ICSGD-Momentum for quadratic function

point), but the test loss of RAdam is worse than other optimization methods.

- When the learning rate is decayed at epoch 150, the upward trend goes into reverse and the testing accuracy of SGD-Momentum started to creep down again, and the testing accuracy of ICSGD-Momentum gradually exceeds that of other optimization methods.

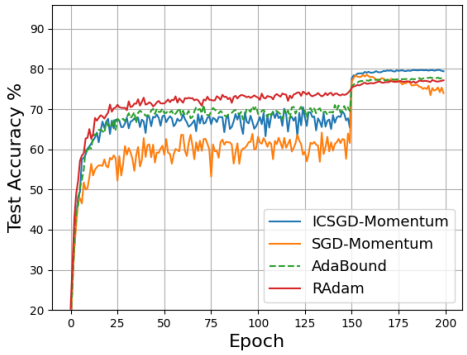


Fig. 4: Testing accuracy curves of DenseNet-121 with different optimizers on CIFAR-100

3.4 Experiments on ImageNet

In our experiments, we also evaluate the performance of ResNet-18 with ICSGD-Momentum on ImageNet (ILSVRC2012 [1]). Following the common practice in [4], we do the standard data augmentation and crop the images to a unified size of 224×224. Then we compare the performance of ICSGD-Momentum with four optimizers, i.e. SGD-Momentum, RAdam and AdaBound. The experiments are conducted on TITAN RTX GPU for 90 epochs with a batch size of 256. The initial learning rate for SGD-Momentum and ICSGD-Momentum is set to 0.1. Following the

1. http://image-net.org/challenges/LSVRC/2012/

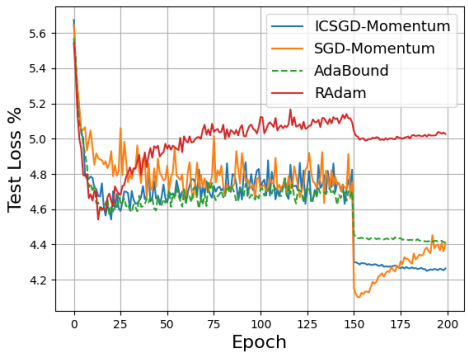


Fig. 5: Log testing loss curves of DenseNet-121 with different optimizers on CIFAR-100

settings in [13] the learning rate is set to 0.001 initially for RAdam and AdaBound. For all the optimizers, the learning rate is lowered by a factor of 10 after epoch 60.

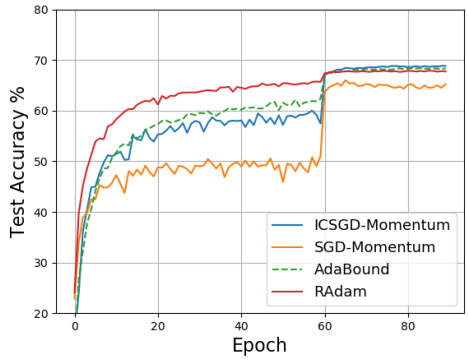


Fig. 6: Testing accuracy curves of ResNet-18 with different optimizers on ImageNet

As shown in Fig. 6 and Fig. 7, the convergence and prediction ability of RAdam are the best before the first learning rate decay point. But when the learning rates are decayed at epoch 60, the testing accuracy of ICSGD-Momentum gradually exceeds that of other optimization methods. ICSGD-Momentum obtains the highest prediction accuracy after the first learning rate decay point.

3.5 Experiments on Penn Treebank

We evaluate the performance of 1-layer LSTM, 2-layer LSTM and 3-layer LSTM with ICSGD-Momentum on the Penn TreeBank. Following the settings in [14], we represent the perplexity (lower is better) on the test set in Fig. 8, Fig. 9 and Fig. 10. For 1-layer, 2-layer and 3-layer LSTM models, ICSGD-Momentum achieves the lowest perplexity, validating its fast convergence as in acceleration methods and good accuracy.

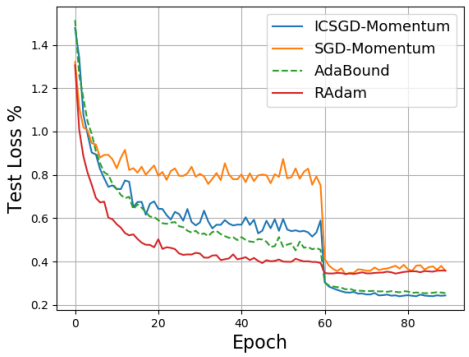


Fig. 7: Log testing loss curves of ResNet-18 with different optimizers on ImageNet

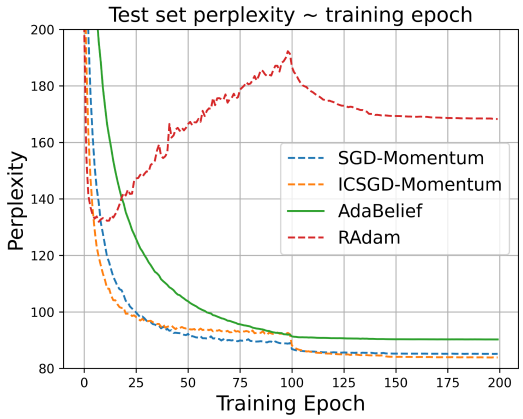


Fig. 8: Test set perplexity on Penn Treebank for 1-layer LSTM

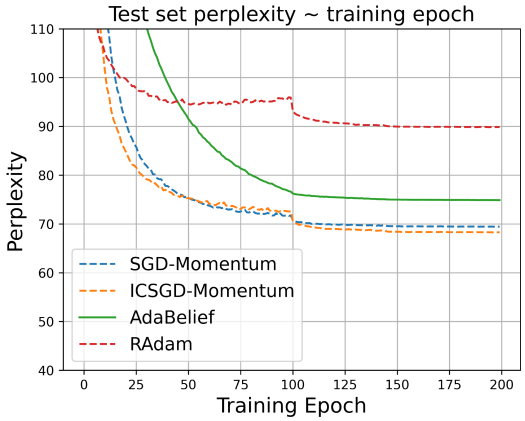


Fig. 9: Test set perplexity on Penn Treebank for 2-layer LSTM

3.6 Experiments on COCO

Yolov5, as an end-to-end primary target detection algorithm, can determine the target category and locate the target at one time. The whole network structure is only composed of convolution layer and input image. It has reached the advanced level of speed and accuracy.

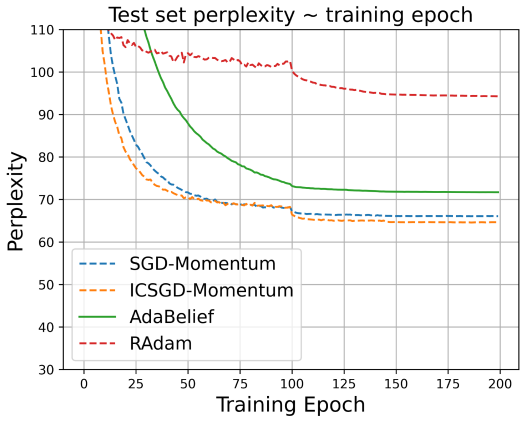


Fig. 10: Test set perplexity on Penn Treebank for 3-layer LSTM

We use SGD-Momentum ($\lambda = 0.01$, $\mu_1 = 0.937$), Adam ($\lambda = 0.001$, $\mu_1 = 0.9$, $\mu_2 = 0.999$) and ICSGD-Momentum ($\lambda = 0.01$, $\mu_1 = 0.937$, $\alpha = 1.1$) to optimize Yolov5, and do comparative experiments on COCO data-sets. mAP is used as the metric to evaluate the performance, where the threshold of IOU is 0.5, and the experimental results for different optimizers are reported in Fig. 11.

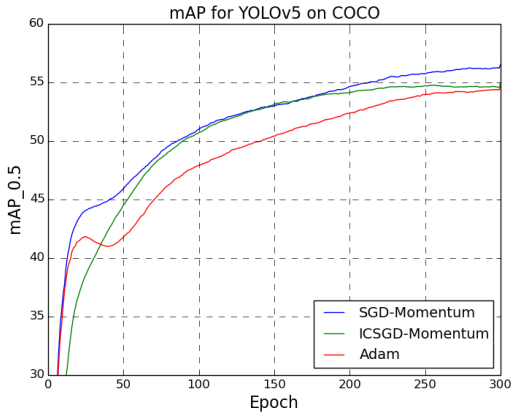


Fig. 11: mAP curves of YOLOv5 on COCO

From the results obtained so far, it seem that ICSGD-Momentum is superior to Adam and is not as good as SGD-Momentum on the mAP of Yolov5 on COCO data-sets. Compared with SGD-Momentum and Adam, mAP curves of ICSGD-Momentum is the most stable.

3.7 Experiments on YCB-Video

The YCB-Video data-sets contains 21 object selected from the YCB data-sets, and each object is different from each other from the class or shape. The data-sets contains 92 videos, and each video contain a series of RGB-D images, which are labeled with 6D pose, instance semantic mask and the object bounding box. Each video is varying in the environment lighting, and the number of the objects.

In this part, we evaluate the ICSGD-Momentum, SGD-Momentum, Adam, AdaBelief and AdaBound on the YCB-Video data-sets, the method used in this section is the PVN3D. The YCB-Video data-sets is split as the training set and the testing set, 80 videos are used for training and 12 videos are used for testing, the operation is the same as the PVN3D. We follow the evaluation metrics proposed in the PoseCNN [16]. The evaluation metrics used in this part is the ADD(S), which means that if the object is symmetric, the ADD-S is used, else the ADD is used.

In this experiment, if the translation and rotation errors are below 5 cm and 5 degree respectively, the predicted pose is correct. The loss function is the same as function proposed in PVN3D, and the learning rate is 0.1. The PVN3D is optimizer by the ICSGD-Momentum, SGD-Momentum, Adam, AdaBelief and AdaBound respectively, and the validating accuracy is shown as Fig. 12.

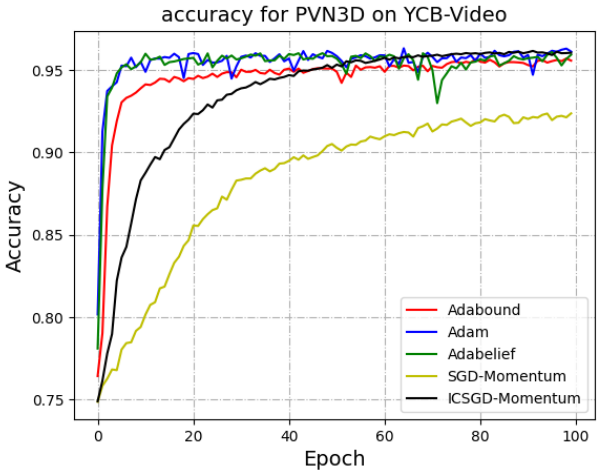


Fig. 12: Accuracy curves for PVN3D on YCB-Video

As shown in Fig. 12, the validating accuracy of SGD-Momentum is lowest, the adaptive optimization algorithms (Adam, AdaBelief and AdaBound) can achieve faster convergence and higher validating accuracy in the whole training process of PoseCNN. When the validating accuracy exceed 0.95, curves exhibit oscillations apparently for the adaptive optimization algorithms. The curve of ICSGD-Momentum is relatively stable, and the validating accuracy is highest when the epoch is 70.

4 Conclusions

Inspired by SGD-Momentum and inter-gradient collision, we proposed a novel ICSGD-Momentum for DNNs in this paper, which utilizes elastic collision factor α as its past and current gradients as $\alpha \mathbf{g}_t + (\alpha - 1)\phi_1(t)$ for improving overshooting problem. At last, some famous data-sets are taken as typical examples to demonstrate the effectiveness of this method.

Acknowledgments

The authors would like to thank the editors and anonymous reviewers for their review and suggestions.

References

[1] W. Fuhl, G. Kasneci, and E. Kasneci, "Teyed: Over 20 million real-world eye images with pupil, eyelid, and iris 2d and 3d segmentations, 2d and 3d landmarks, 3d eyeball, gaze vector, and eye movement types," *arXiv preprint arXiv:2102.02115*, 2021.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[5] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*. PMLR, 2013, pp. 1139–1147.

[6] W. An, H. Wang, Q. Sun, J. Xu, Q. Dai, and L. Zhang, "A pid controller approach for stochastic optimization of deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8522–8531.

[7] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research*, vol. 12, no. 7, 2011.

[8] M. C. Mukkamala and M. Hein, "Variants of rmsprop and adagrad with logarithmic regret bounds," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2545–2553.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[10] L. Luo, W. Huang, Q. Zeng, Z. Nie, and X. Sun, "Learning personalized end-to-end goal-oriented dialog," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6794–6801.

[11] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[12] J. An, F. Liu, J. Zhao, and F. Shen, "Ic networks: Remodeling the basic unit for convolutional neural networks," *arXiv preprint arXiv:2102.03495*, 2021.

[13] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," *arXiv preprint arXiv:1902.09843*, 2019.

[14] J. Zhuang, T. Tang, Y. Ding, S. Tatikonda, N. Dvornek, X. Papademetris, and J. S. Duncan, "Adabelief optimizer: Adapting stepsizes by the belief in observed gradients," *arXiv preprint arXiv:2010.07468*, 2020.

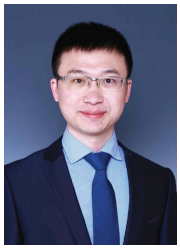
[15] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.

[16] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

Dr. Weidong Zou received the Ph.D. degree in control science and engineering from the School of Automation, Beijing Institute of Technology, Beijing, China, in 2017. He is currently a research fellow with the School of Automation, Beijing Institute of Technology. His current research interests include neural network and modeling and simulation of traffic flow.

Dr. Yuanqing Xia received the Ph.D. degree in control theory and control engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001. He is currently the Dean of the School of Automation, Beijing Institute of Technology. He has published eight monographs in Springer, Wiley, and CRC, and over 100 papers in international scientific journals. His current research interests include networked control systems, robust control and signal processing, active disturbance rejection control and flight control. Dr. Xia was a recipient of the National Science Foundation for Distinguished Young Scholars of China in 2012, the Second Award of the Beijing Municipal Science and Technology (No. 1) in 2010 and 2015, the Second National Award for Science and Technology (No. 2) in 2011, and the Second Natural Science Award of the Ministry of Education (No. 1) in 2012. He is a Deputy Editor of the Journal of Beijing Institute of Technology, an Associate Editor of Acta Automatica Sinica, Control Theory and Applications, the International Journal of Innovative Computing, Information and Control, and the International Journal of Automation and Computing. In 2016, he was honored as the Yangtze River Scholar Distinguished Professor and was supported by National High Level Talents Special Support Plan (Million People Plan) by the Organization Department of the CPC Central Committee.

Dr. Weipeng Cao is currently an associate researcher at College of Computer Science and Software Engineering, Shenzhen University. His research interests include artificial intelligence, machine learning, especially neural networks with random weights. He has published more than 20 high-quality papers in the reputable journals and conferences, including one highly cited paper (ESI top 1%) and one hottest paper (ESI top 0.1%).



Dr. Gao Huang received the B.S. degree from the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, in 2009, and the Ph.D. degree from the Department of Automation, Tsinghua University, Beijing, in 2015. He was a Visiting Research Scholar with the Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO, USA, in 2013 and was a Post-Doctoral Researcher with Department of Computer Science, Cornell University, Ithaca, USA from 2015 to 2018. He is currently an assistant professor at the Department of Automation, Tsinghua University. His research interests include machine learning and computer vision.

Dr. Yizeng Han received the B.S. degree from the Department of Automation, Tsinghua University, Beijing, China, in 2018. And he is currently pursuing the Ph.D. degree in control science and engineering with the Department of Automation, Institute of System Integration in Tsinghua University. His current research interests include computer vision and deep learning, especially in dynamic neural networks.

Dr. Xinwang Liu received his PhD degree from National University of Defense Technology (NUDT), China. He is now Professor at School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. Dr. Liu has published 70+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, IEEE T-NNLS, IEEE T-MM, IEEE T-IFS, ICML, NeurIPS, CVPR, ICCV, AAAI, IJCAI, etc. More information can be found at https://xinwangliu.github.io/.