An SGD-based meta-learner with “growing” descent

I. Kulikovskikh[1,2,3]**, S. Prokhorov**[1]**, T. Legović**[3] **and T. Šmuc**[3]

[1]Samara National Research University, Moskovskoe Shosse 34, Samara, Russia, 443086
[2]Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, Zagreb, Croatia, 10000
[3]Ruđer Bošković Institute, Bijenička cesta 54, Zagreb, Croatia, 10000

E-mail: `ilona@irb.hr, sp.prokhorov@gmail.com, legovic@irb.hr, smuc@irb.hr`

Abstract. The paper considers the problem of accelerating the convergence of stochastic gradient descent (SGD) in an automatic way. Previous research puts forward such algorithms as Adagrad, Adadelta, RMSprop, Adam and etc. to adapt both the updates and learning rates to the slope of a loss function. However, these adaptive methods do not share the same regret bound as the gradient descent method. Adagrad provably achieves the optimal regret bound on the assumption of convexity but accumulates the squared gradients in the denominator that dramatically shrinks the learning rate. This research is aimed at introducing a generalized logistic map directly into the SGD method in order to automatically set its parameters to the slope of the logistic loss function. The optimizer based on the population may be considered as a meta-learner that learns how to tune both the learning rate and gradient updates with respect to the rate of population growth. The present study yields the “growing” descent method and a series of computational experiments to point out the benefits of the proposed meta-learner.

The ability to rapidly adapt from small pieces of data to current tasks is essential to effective learning. However, deep learning algorithms traditionally require big datasets to learn tasks by fitting a deep neural network [1, 19] over them through extensive incremental updates of SGD. This approach seems time-consuming and even more challenging if fast adaptation is crucial.

Meta-learning has opened a door for learning optimizers to exploit problems structure in an automatic way [2–19]. This suggests that optimizers which are used to be hand-designed can serve as meta-learners by moving the learning level up from data to tasks.

While SGD is usually considered as a meta-learner, the algorithm itself still needs improvements. For example, it is advisable to adapt larger learning rates for smaller gradients and smaller learning rates for larger gradients to balance their respective influences. A number of adaptive methods were proposed to overcome this weakness of SGD such as Adagrad, Adadelta, RMSprop, Adam and etc. [20, 21]. However, even if these methods do speed up the convergence rate, they nevertheless result in worse generalization error compared to the SGD with a single learning rate [22].

The aim of this study is to propose a meta-learner based on the SGD method that could adapt the learning rate as well as gradient updates to the slope of the logistic loss function with better generalization error than the plain SGD. For this purpose, we suggest to “grow” descent of stochastic gradient by embedding the generalized logistic map directly in the SGD method. This allows us to:

(i) guarantee the same regret bound as the gradient descent method;

(ii) introduce deterministic chaos that may greatly assist in improving the convergence rate of gradient methods [30–33];

(iii) learn how to tune both the learning rate and gradient updates with respect to the rate of population growth automatically.

In addition, the parameters of logistic map have a clear interpretation in biological and ecological systems that may bring the potential advantage to modelling the nature-inspired framework of meta-learning.

1. Problem statement

Consider a dataset $\{\mathrm{x}_i, y_i\}_{i=1}^m$ with $\mathrm{x}_i \in \mathbb{R}^n$ and $y_i \in \{0, 1\}$. Let us state the problem [20]

$$\mathscr{L}(\boldsymbol{\theta}) \underset{\boldsymbol{\theta}}{\rightarrow} \min, \tag{1}$$

where a loss function with the weight vector $\boldsymbol{\theta} \in \mathbb{R}^n$ is defined as

$$\mathscr{L}(\boldsymbol{\theta}) = \sum_{i=1}^{m} \ell(y_i \boldsymbol{\theta}^T \mathrm{x}_i). \tag{2}$$

In equation (2) it is assumed that all the labels are positive: $\forall i : y_i = 1, \|x_i\| < 1$, the dataset is linearly separable: $\exists\ \boldsymbol{\theta}^*$ such that $\forall i : \boldsymbol{\theta}^{*T}\mathrm{x}_i > 0$ and $\forall t$: $\ell(t)$ is differentiable and monotonically decreasing to zero

$$\ell(t) > 0, \ell'(t) < 0, \lim_{t\to\infty} \ell(t) = \lim_{t\to\infty} \ell'(t) = 0,$$

and its derivative is β-Lipshitz: $\ell(t') \leq \ell(t) + \langle \nabla \ell(t), t' - t\rangle + \frac{\beta}{2}\|t' - t\|^2, \quad \beta > 0$.

1.1. Stochastic gradient descent

The solution to the problem (1) can be found as the iterates of gradient descent with a full batch [20, 21]:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla \mathscr{L}(\boldsymbol{\theta}_t) = \boldsymbol{\theta}_t - \eta \sum_{i=1}^{m} \ell'(\boldsymbol{\theta}_t^{\mathrm{T}} \mathrm{x}_i)\mathrm{x}_i,$$

where η is the learning rate. In the stochastic setting, gradient descent updates the weight vector for each i^{th} mini-batch dataset such as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla \mathscr{L}(\boldsymbol{\theta}_t; \mathrm{x}_i) = \boldsymbol{\theta}_t - \eta \sum_{l=1}^{i} \ell'(\boldsymbol{\theta}_t^{\mathrm{T}} \mathrm{x}_l)\mathrm{x}_l. \tag{3}$$

The equation (3) implies that only the partial information of the loss function for i^{th} batch data guides the gradient direction for each iteration.

The regret bound of gradient descent algorithm with a full batch is bounded by a constant that entails the loss function at a certain step that is bounded by the inverse of the number of iterations. The SGD method shares the same regret bound as it modifies the basic structure of the gradient descent.

In the following section we define the logistic map for insertion into the SGD method.

1.2. The generalized logistic map

The model that describes the population growth $P(t)$ in an infinite environment can be presented as [23]:

$$\frac{\mathrm{d}P(t)}{\mathrm{d}t} = rP(t),$$

where $r \equiv b - d$ is the *per capita* rate of population growth, b and d are respective *per capita* rates of birth and death.

Let us introduce the generalized logistic equation and its solution [23]:

$$\frac{dP(t)}{dt} = -r\left(P(t) - A\right)\ln^{[q]}\left(\frac{P(t) - A}{K - A}\right), \tag{4}$$

where

$$\ln^{[q]}(x) = \int_1^x \frac{dt}{t^{1-q}} = \begin{cases} \frac{x^q - 1}{q}, & q \neq 0; \\ \ln(x), & q \to 0, \end{cases} \tag{5}$$

$$P(t) = A + \frac{K - A}{\left(1 - \left(1 - \left(\frac{K-A}{P_0+A}\right)^q\right)\exp(-rt)\right)^{\frac{1}{q}}}, \tag{6}$$

where q is the generalization parameter, $0 \leq A < K$, K is the upper asymptote or the carrying capacity of population; A is the lower asymptote that indicates critical population thresholds below which a population crashes to extinction. The asymptote A serves as a substitute for the Allee effects [24] which are broadly defined as a decline in individual fitness at low population size or density. Even if modelling these effects is beyond the scope of this paper, it draws promising directions for further research. Given that [23] :

(i) $q = 1$ and $K \to \infty$: the generalized model reduces to the Malthus model that describes the exponential growth of a population [26];

(ii) $q = 1$: the generalized model reduces to the Verhulst model that describes the logistic growth of a population [27];

(iii) $q \to 0$: the generalized model reduces to the Gompertz model that also describes the logistic growth of a population, but it is more flexible in the way of approaching these asymptotes [28, 29].

A discrete-time population model (the logistic map) at the time k is given by [25]:

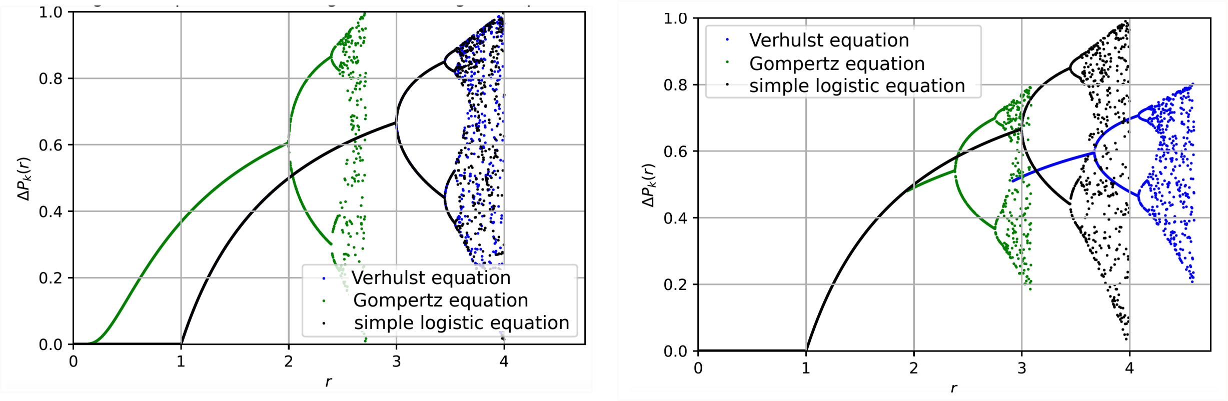
$$\Delta P_k = rP_k(1 - P_k),$$

where $P_k \in [0, 1]$, $k \in \mathbb{N}$ and $r \in (0, 4]$. Fig. 1 illustrates the logistic maps based on the logistic growth models with regard to different values A and K where the simple logistic equation implies $A = 0$ and $K = 1$.

Let $\Delta t = 1$, $\ell_r'(P) = rP(1 - P)$. Then $P_{k+1} = \ell_r'(P_k)$ and the composition of k functions $\ell_r(P)$ can be represented by:

$$\mathcal{L}_r^{k'}(P) = \begin{cases} \ell_r'(P), & k = 1; \\ \left(\ell_r' \circ \mathcal{L}_r^{[k-1]'}\right)(P), & k > 1. \end{cases} \tag{7}$$

The equation (7) describes the dynamics of a population. If the growth rate $0 < r \leq 1$, the population dies out and goes extinct. Increasing the rate of growth allows the population to settle at the stable value or fluctuate across booms and busts. Finally, at a relatively high values of growth rate, the logistic equation produces chaos [25].



a) $A = 0,\ K = 1$

b) $A = 0.1,\ K = 0.9$

Figure 1. The logistic maps for different values A and K

2. Growing descent of stochastic gradient

Let us extend the definition $\ell(t)$ taking into account the generalized logistic equation (4) and its solution (6):

$$\ell_r'(t;a,b,q) = r(\ell_r(t;a,b,c,q) - a)\ln^q\left(\frac{\ell_r(t;a,b,c(a,b),q) - a}{b-a}\right); \tag{8}$$

$$\ell_r(t;a,b,q) = a + \frac{b-a}{(1-(1-(\exp(-c(a,b)))^q)\exp(-rt))^{\frac{1}{q}}}, \tag{9}$$

where $a \equiv A$, $b \equiv K$, $0 \le a < b \le 1$; $c(a,b) \equiv \ln\left(\frac{P_0+A}{K-A}\right)$, $P_0 < K - 2A$, $c(a,b) < 0$. Here $u \equiv t$, $k = 1$, $r \in (0,4]$.

The logistic loss and its derivative with regard to (9) and (8) can be given as follows:

$$\ln\mathscr{L}(\boldsymbol{\theta};\mathrm{x}_i,\mathcal{L}_r'^k(a,b,q)) = -\sum_{l=1}^{i}\ln\ell(\boldsymbol{\theta}^T\mathrm{x}_l;a,b,q), \tag{10}$$

$$-\nabla\ln\mathscr{L}(\boldsymbol{\theta};\mathrm{x}_i,\mathcal{L}_r'^k(a,b,q)) = -\sum_{l=1}^{i}\left(\left(1-\ell(\boldsymbol{\theta}^T\mathrm{x}_l;a,b,q)\right)\frac{\mathcal{L}_r'^k(\boldsymbol{\theta}^{\mathrm{T}}\mathrm{x}_l;a,b,q)}{\ell_r'(\boldsymbol{\theta}^{\mathrm{T}}\mathrm{x}_l)}\mathrm{x}_l\right), \tag{11}$$

where $\mathcal{L}_r'^k(\boldsymbol{\theta}^{\mathrm{T}}\mathrm{x}_l;a,b,q)$ defines the generalized logistic map according to (7), $\ell_r'(\boldsymbol{\theta}^{\mathrm{T}}\mathrm{x}_l) = \ell(\boldsymbol{\theta}^T\mathrm{x}_l)\big(1-\ell(\boldsymbol{\theta}^T\mathrm{x}_l)\big)$ gives the derivative of the sigmoid function $\ell(\boldsymbol{\theta}^T\mathrm{x}_l)$ with $a = 0$ and $b = 1$.

Then, a meta-learner with “growing” descent based on the generalized logistic map with reference to (10) and (11) can be defined as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_{r_j^k}\nabla\ln\mathscr{L}(\boldsymbol{\theta}_t;\mathrm{x}_i,\mathcal{L}_r'^k(a_j,b_j,q)),$$

$$(a,b,r^k)_{j+1} \leftarrow (a,b,r^k)_j - \eta\nabla\mathscr{L}(\boldsymbol{\theta}_{t+1};\mathrm{x}_i,\mathcal{L}_r'^k(a_j,b_j,q)),$$

where the adaptive step size $\eta_{r_j^k} = (b-a)c(a_j,b_j)r_j^k\eta$.

Algorithm 1 presents a step-by-step procedure for the meta-learner implementation.

In the following section we present the experimental evidence on the improved convergence rate of proposed growing descent method.

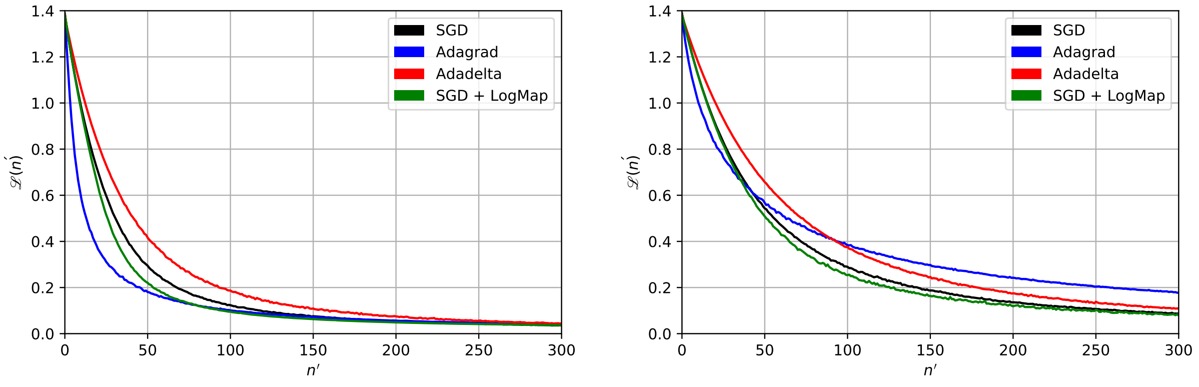
Algorithm 1 SGD-based meta-learner with “growing” descent

```
procedure GrowingSGD($\mathrm{x}, \eta, r$)
    Initialize $\boldsymbol{\theta}_0$;
    $a_0 \leftarrow 0,\ b_0 \leftarrow 1,\ r_0^k \leftarrow 1$;
    Split $\mathrm{x}_i$ into train $\mathrm{x}_i^T$ and cross-validation $\mathrm{x}_i^{CV}$ subsets;
    $j \leftarrow 0$;
    repeat
        $t \leftarrow 0$;
        repeat
            $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta_{r_j^k} \nabla_{(\boldsymbol{\theta})} \mathscr{L}(\boldsymbol{\theta}_t; \mathrm{x}_i^{\mathrm{T}}, \mathcal{L}_r'^k(a_j, b_j, q))$;
            $t \leftarrow t + 1$;
        until converge
        $(a, b, r^k)_{j+1} \leftarrow (a, b, r^k)_j - \eta \nabla_{(a,b,r^k)} \mathscr{L}(\boldsymbol{\theta}_{t+1}; \mathrm{x}_i^{\mathrm{CV}}, \mathcal{L}_r'^k(a_j, b_j, q))$
        $j \leftarrow j + 1$;
    until converge
    return $\boldsymbol{\theta}_{t+1}, (a, b, r^k)_{j+1}$
```

3. Results

3.1. A synthetic setting

We conducted a series of computational experiments on a linear separable dataset modelled with $\mathcal{N}(0,1)$ subject to $m = 5000$, $n = 10$, $n_{\text{epochs}} = 100$, the mini-batch size $m_{\text{batch}} = 100$ and the growth parameters $a = 0$, $b = 1$, $q = 1$. We took the default values for the learning rate η and the parameters of adaptive methods. A simple model of neural network consists of one hidden layer with different number of neurons $n_{\text{neuron}} = 4$ (see Fig. 2 a)) and $n_{\text{neuron}} = 1$ (see Fig. 2 b)).



a) $n_{\text{neuron}} = 4$, $r = 2.8$

b) $n_{\text{neuron}} = 1$, $r = 3.8$

Figure 2. The results of computational experiments

Fig. 2 a) shows the advantage of Adagrad with the adaptive learning rate over other methods. Fig. 2 b), in turn, reveals the Adagrad’s main weakness: the result of the accumulation of the squared gradients in the denominator. This practically leads to a dramatic decrease in the learning rate for which the algorithm is no longer able to gain additional knowledge. Adadelta is an extension of Adagrad that is aimed at reducing its aggressive, steadily decreasing learning rate. However, it gives worse results than the proposed SGD+LogMap (see Fig. 2). We can also see that the growing descent algorithm SGD+LogMap improves the SGD convergence rate:

it requires smaller number of iterations to achieve the same regret bound for both $n_{\text{neuron}} = 4$ if $n' > 10$ (see Fig. 2 a)) and $n_{\text{neuron}} = 1$ if $n' > 25$ (see Fig. 2 b)).

3.2. An experimental setting

We implemented a neural network with two hidden layers, each of them containing 10 neurons, based on the proposed SGD-based meta-learner. We took the default value for the learning rate η and set the number of functions in (7) equal to 1. We trained the model on the freely available MNIST and fMNIST dataset. The training subset was divided into an actual training set and a validation subset for selecting the hyperparameters (a, b, r^k) using 5-fold cross-validation. The number of epochs for training was $n_{\text{epoch}} = 1500$. The batch size was $n_{\text{batch}} = 25$. Fig. 3 and 4 depict the model performance for the plain SGD optimizer and the proposed SGD-based meta-learner. As we can see, the proposed SGD-based meta-learner with "growing" descent demonstrates faster convergence compared to the plain SGD.

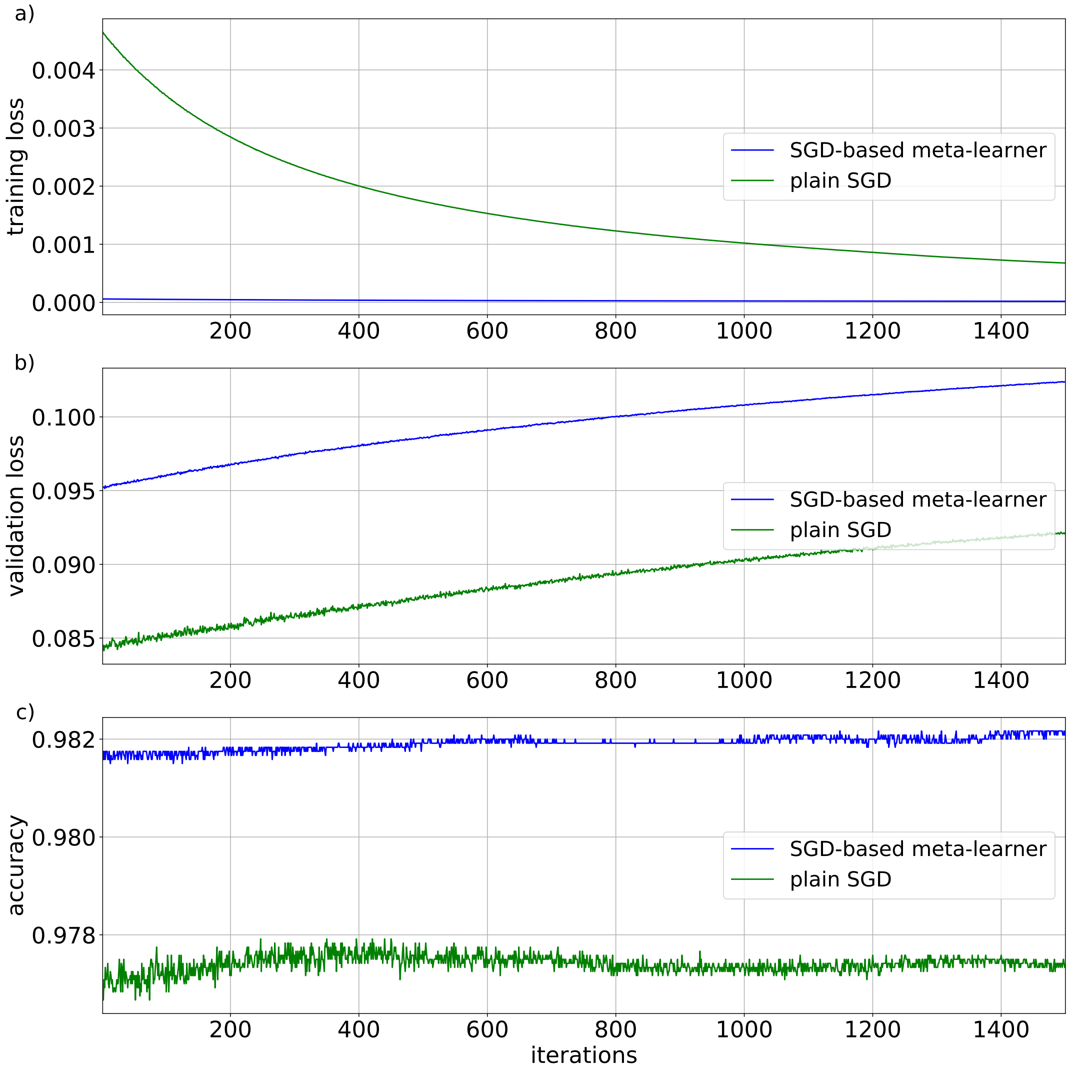


Figure 3. The model performance on the MNIST dataset for the simple and generalized logistic loss: a) training loss; b) validation loss; c) accuracy

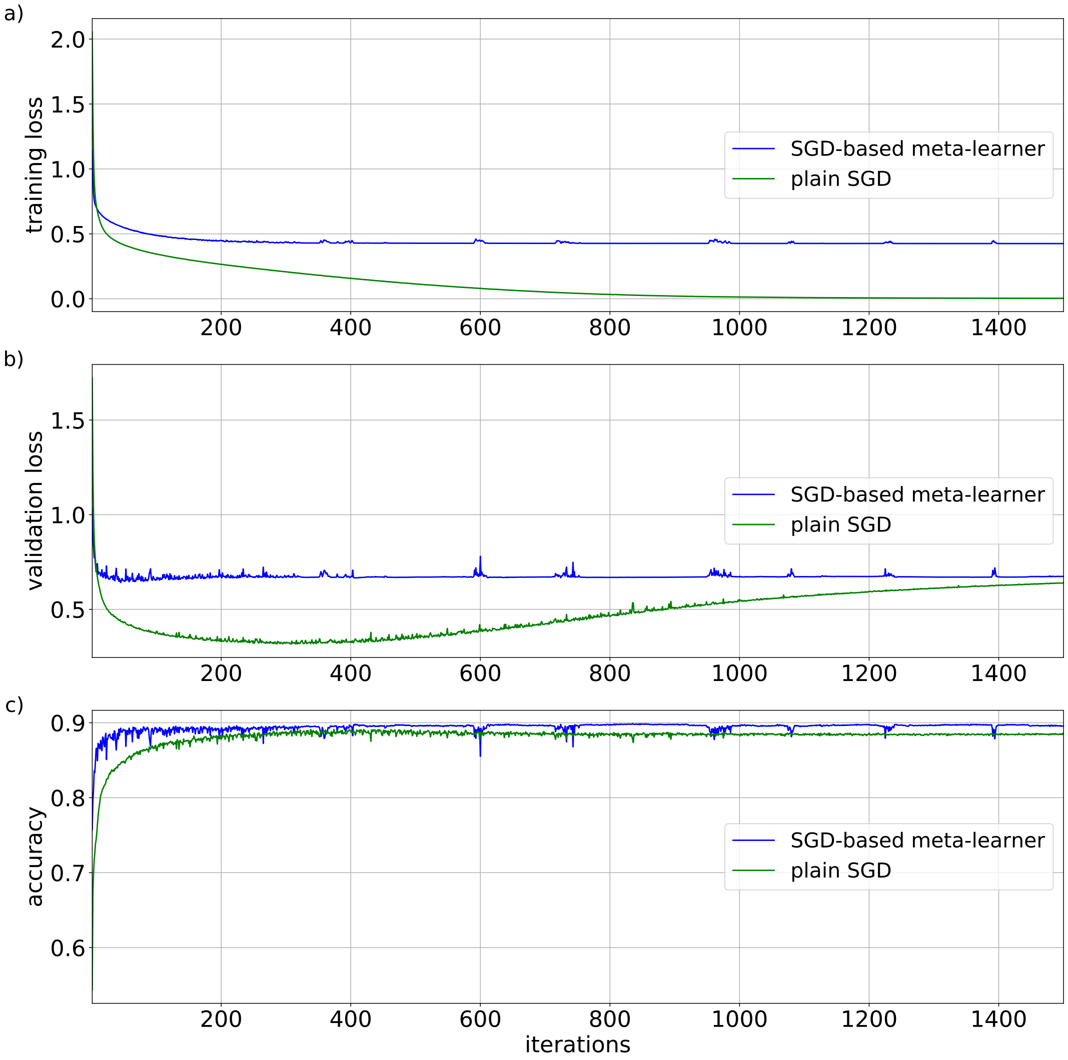


Figure 4. The model performance on the fMNIST dataset for the simple and generalized logistic loss: a) training loss; b) validation loss; c) accuracy

4. Conclusions

The proposed meta-learner is based on the plain SGD method and the generalized logistic map. It shares the same regret bound as the SGD and learns how to adapt both the learning rate and the gradient updates with the growth rate in an automatic way. In addition, the proposed approach to growing descent of stochastic gradient allows one to introduce deterministic chaos that improves the convergence rate. In a series of computational experiments the validity of the proposed approach has been confirmed.

Acknowledgments

This work was supported by the Russian Federation President grant No. MK-6218.2018.9 and the Ministry of Education and Science of the Russian Federation grant No. 074-U01. The research in Section 3 was partly supported by RFBR (project No. 18-37-00219). The authors acknowledge the support by the Centre of Excellence project “DATACROSS”, co-financed by the Croatian Government and the European Union through the European Regional Development Fund - the Competitiveness and Cohesion Operational Programme (KK.01.1.1.01.0009).

References

[1] Savchenko AV 2017 *Computer Optics* **41(3)** 422-430.

[2] Wu X, Ward R, Bottou L 2018 WNGrad: Learn the Learning Rate in Gradient Descent arXiv:1803.02865

[3] Andrychowicz M, Denil M, Gomez Colmenarejo S, Hoffman M W, Pfau D, Schaul T, Shillingford B, de Freitas N 2016 Learning to Learn by Gradient Descent by Gradient Descent arXiv:1606.04474

[4] Ren M, Zeng W, Yang B, Urtasun R 2018 Learning to Reweight Examples for Robust Deep Learning arXiv:1803.09050

[5] Li Q, Tai C, E W 2015 Stochastic Modified Equations and Adaptive Stochastic Gradient Algorithms arXiv:1511.06251

[6] Wichrowska O, Maheswaranathan N, Hoffman M W, Gomez Colmenarejo S, Denil M, de Freitas N, Sohl-Dickstein J 2017 Learned Optimizers that Scale and Generalize arXiv:1703.04813

[7] Li Z, Zhou F, Chen F, Li H 2017 Meta-SGD: Learning to Learn Quickly for Few-Shot Learning arXiv:1707.09835

[8] Li K, Malik J 2016 Learning to Optimize arXiv:1606.01885

[9] Li K, Malik J 2017 Learning to Optimize Neural Nets arXiv:1703.00441

[10] Al-Shedivat M, Bansal T, Burda Y, Sutskever I, Mordatch I, Abbeel P, Li K, Malik JK 2017 Continuous Adaptation via Meta-learning in Nonstationary and Competitve Environments arXiv:1710.03641

[11] Aljundi R, Babiloni F, Elhoseiny M, Rohrbach M, Tuytelaars T 2017 Memory Aware Synapses: Learning What (not) to Forget arXiv:1711.09601

[12] Li D, Yang Y, Song Y Z, Hospedales T M 2017 Learning to Generalize: Meta-Learning for Domain Generalization arXiv:1710.03463

[13] Wang Y X, Ramanan D, Hebert M 2017 Learning to Model the Tail http://papers.nips.cc/paper/7278-learning-to-model-the-tail.pdf

[14] Lv K, Jiang S, Li J 2017 Learning Gradient Descent: Better Generalization and Longer Horizons arXiv:1703.03633

[15] Munkhdalai T, Yu H 2017 Meta Networks arXiv:1703.00837

[16] Mishra N, Rohaninejad M, Chen X, Abbeel P 2017 A Simple Neural Attentive Meta-learner arXiv:1707.03141

[17] Finn C, Abbeel P, Levine S 2017 Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks arXiv:1703.03400

[18] Duan Y, Andrychowicz M, Stadie B, Ho J, Schneider J, Sutskever I, Abbeel P, Zaremba W 2017 One-Shot Imitation Learning http://papers.nips.cc/paper/6709-one-shot-imitation-learning.pdf

[19] Ha D, Dai A, L Q V 2016 Hypernetworks arXiv:1609.09106

[20] Kim H S, Kang J H, Park W M, Ko S H, Cho Y H, Yu D S, Song Y S, Choi J W 2017 arXiv:1707.01647

[21] Ruder S 2016 arXiv:1609.04747

[22] Wilson A, Roelofs R, Stern M, Srebro N, Recht B 2017 *NIPS Proceedings* 4151–4161

[23] Ribeiro F L 2017 *Revista Brasileira de Ensino de Fisica* **39(1)** e1311

[24] Allee W C 1927 *The Quarterly Review of Biology* **2(3)** 367–398

[25] May R M 1976 *Nature* **261(5560)** 459–467

[26] Malthus T R 1798 *An Essay on the Principle of Population, as it Affects the Future Improvement of Society. With Remarks on the Speculations of Mr. Godwin, M. Condorcet and Other Writers* (London: J. Johnson)

[27] Verhulst P F 1838 *Correspondance mathamatique et physique* **10** 113–121

[28] Gompertz B 1825 *Philosophical Transactions of the Royal Society of London B: Biological Sciences* **182** 513–585.

[29] Winsor C P 1932 *Proc. Nat. Acad. Sci* **18(1)** 1–8

[30] Doel K, Ascher U The Chaotic Nature of Faster Gradient Descent www.cs.ubc.ca/ascher/papers/doas1.pdf

[31] Mpitsos G J, Burton R M Jr 1992 *Neural Networks* **5** 605-625

[32] Verschure P F M J 1991 Chaos-based Learning *Complex Systems* **5** 359-370

[33] Zhang H, Zhang Y, Xu D, Liu X 2015 *Cognitive Neurodynamics* **9(3)** 331-340