Weierstraß-Institut
für Angewandte Analysis und Stochastik

Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

Adaptive gradient descent for convex and non-convex stochastic optimization

Aleksandr Ogaltsov[1], Darina Dvinskikh[2],

Pavel Dvurechensky[2], Alexander Gasnikov[1,3], Vladimir Spokoiny[2]

submitted: December 10, 2019

[1] Higher School of Economics
Moscow
Russian Federation
E-Mail: aogalcov@hse.ru
gasnikov@yandex.ru

[2] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: darina.dvinskikh@wias-berlin.de
pavel.dvurechensky@wias-berlin.de
vladimir.spokoiny@wias-berlin.de

[3] Moscow Institute of Physics and Technology
Institutskiy Pereulok, 9
Dolgoprudny, Moscow Region
141701 Russian Federation
and
Institute for Information Transmission Problems RAS
Bolshoy Karetny per. 19, build.1
Moscow 127051
Russian Federation
E-Mail: gasnikov@yandex.ru

No. 2655
Berlin 2019

2010 *Mathematics Subject Classification.* 90C25, 90C30, 90C06, 90C90.

Key words and phrases. Convex and non-convex optimization, stochastic optimization, first-order method, adaptive method, gradient descent, complexity bounds, mini-batch.

The authors are grateful to Yu. Nesterov and A. Tyurin for fruitful discussions.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax:	+49 30 20372-303
E-Mail:	preprint@wias-berlin.de
World Wide Web:	http://www.wias-berlin.de/

Adaptive gradient descent for convex and non-convex stochastic optimization

Aleksandr Ogaltsov, Darina Dvinskikh,

Pavel Dvurechensky, Alexander Gasnikov, Vladimir Spokoiny

Abstract

In this paper we propose several adaptive gradient methods for stochastic optimization. Our methods are based on Armijo-type line search and they simultaneously adapt to the unknown Lipschitz constant of the gradient and variance of the stochastic approximation for the gradient. We consider an accelerated gradient descent for convex problems and gradient descent for non-convex problems. In the experiments we demonstrate superiority of our methods to existing adaptive methods, e.g. AdaGrad and Adam.

1 Introduction

In this paper we consider unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

where $f(x)$ is a smooth, possibly non-convex function with L-Lipschitz continuous gradient. We say that a function $f : E \to \mathbb{R}$ has a L-Lipschitz continuous gradient if it is continuously differentiable and its gradient satisfies

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|x-y\|_2^2, \quad \forall\, x, y \in E.$$

We assume that the access to the objective f is given through stochastic oracle $\nabla f(x, \xi)$, where ξ is a random variable. The main assumptions on the stochastic oracle are standard for stochastic approximation literature [30]

$$\mathbb{E}\nabla f(x,\xi) = \nabla f(x), \;\; \mathbb{E}\left(\|\nabla f(x,\xi) - \nabla f(x)\|_2^2\right) \leq D. \tag{2}$$

One of the cornerstone questions for optimization methods is the choice of the stepsize, which has a dramatic impact on the convergence of the algorithm and the quality of the output, as, e.g., in deep learning, where it is called learning rate. Standard choice of the stepsize for the gradient descent in deterministic optimization is $1/L$ [33] and it is possible to use the (accelerated) gradient descent without knowing this constant [31, 28, 11] in convex case and gradient method [3, 18] in non-convex case using an Armijo-type line search and checking whether the quadratic upper bound based on the L-smoothness is correct. Another option to adapt to the unknown smoothness is to use small-dimensional relaxation [29, 34, 22]. So far there is only partial understanding of how to generalize these ideas for stochastic optimization. Heuristic adaptive algorithms for smooth strongly convex optimization is proposed in [4, 13] (see also review [35]). Theoretical analysis in these papers is made for the idealised versions of their algorithms which either are not practical or not adaptive. Heuristic adaptive algorithms for smooth convex optimization were proposed in [14, 36]. [23] propose and theoretically analyse a method for stochastic monotone variational inequalities.

Another way to construct an adaptive stepsize comes from non-smooth optimization [37], where it is suggested to take it as $1/\|\nabla f(x)\|_2$. This idea turned out to be very productive and allowed to introduce stochastic adaptive methods [9, 24, 6, 2], among which usually the Adam algorithm is a method of choice [38]. Recently this idea was generalized to propose adaptive methods for non-smooth and smooth stochastic convex optimization, yet with acceleration only for non-stochastic optimization in [27]; for non-smooth and smooth stochastic monotone variational inequalities in [1]; for non-convex stochastic optimization [41]. One of the main drawbacks in these methods is that they are not flexible to mini-batching approach, which is widespread in machine learning. The problem is that to choose the optimal mini-batch size (see also [19]) for all these methods, one needs to know all the parameters like D, L and the adaptivity vanishes. Moreover, these methods usually either need some additional information about the problem, e.g. distance to the solution of the problem, which may not be known for a particular problem, or have a set of hyperparameters, the best values for which are not readily available even in the non-adaptive setting. Finally, the stepsize in these methods is decreasing and in the best case asymptotically converges to a constant stepsize of the order $1/L$. This means that the methods could not adapt to the local curvature of the objective function and converge faster in the areas where the function is smoother. We summarize available literature in the Table below.

In this paper we follow an alternative line, trying to extend the idea of Armijo-type line search for the adaptive methods for convex and non-convex stochastic optimization. Surprisingly, the adaptation is needed not to each parameter separately, but to the ratio D/L, which can be considered as signal to noise ratio or an effective Lipschitz constant of the gradient in this case. We propose an accelerated and non-accelerated gradient descent for stochastic convex optimization and a gradient method for stochastic non-convex optimization. Our methods are flexible enough to use optimal choice of mini-batch size without additional information on the problem. Moreover, our procedure allows an increase of the stepsize, which accelerated the methods in the areas where the Lipschitz constant is small. Also, as opposed to the existing methods, our algorithms do not need to know neither the distance to the solution, nor a set of complicated hyperparameters, which are usually fine-tuned by multiple repetition of minimization process. Moreover, since our methods are based on inexact oracle model (see e.g. [7, 16]), they are adaptive not only for a stochastic error, but also for deterministic, e.g. non-smoothness of the problem. This means that our methods are universal for smooth and non-smooth optimization [32, 43, 10]. Finally, We demonstrate in the experiments that our methods work faster than state-of-the-art methods [9, 24].

The paper is structured as follows. In Sect. 2 we present two stochastic algorithms based on stochastic gradient method to solve optimization problem of type (1) with convex objective function. The first algorithm is accompanied by the complexity bounds on total number of iterations and oracle calls for the approximated stochastic gradients. The second algorithm is fully-adaptive and does not require the knowledge of Lipschitz constant for the gradient of the objective and the variance of its stochastic approximation. Sect. 2 renews Sect. 2 for non-convex objective function. Finally, in Sect. 4 we show numerical experiments supporting the theory in above sections.

[1] N-C stands for availability of an algorithm for non-convex optimization, N-Ac for a non-accelerated algorithm for convex optimization, Ac for accelerated algorithm for convex optimization, Prf for proof of the convergence rate, Btch for possibility to adaptively choose batch size without knowing other parameters, Par for non-necessity to know or tune hyperparameters like distance to the solution for choosing the stepsize.

Paper	N-C[1]	N-Ac	Ac	Prf	Btch	Par
Duchi et al.'11	×	√	×	√	×	×
Byrd et al.'12	×	√	×	×	×	×
Friedlander et al.'12	×	√	×	×	×	×
Kingma & Ba'15	×	√	√	×	×	×
Iusem et al.'19	×	√	×	√	×	√
Gasnikov'17	×	√	×	×	√	√
Levy et al.'18	×	√	×	√	×	×
Deng et al.'18	×	×	√	×	×	×
Ogaltsov et al.'19	×	×	√	×	√	√
Ward et al.'19	√	×	×	√	×	√
Bach & Levy'19	×	√	×	√	×	×
This paper	√	√	√	×	√	√

Table 1

2 Stochastic convex optimization

In this section we solve problem (1) for convex objective. Assuming the Lipschitz constant for the continuity of the objective gradient to be known we prove the complexity bounds for the proposed algorithm. Then we refuse this assumption and provide complexity bounds for the adaptive version of the algorithm which does not need the information about Lipschitz constant.

2.1 Non-adaptive algorithm

We start with stochastic gradient descent with general stepsize h

$$x^{k+1} = x^k - h\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r). \tag{3}$$

where $\nabla^r f(x, \{\xi_l\}_{l=1}^r)$ is a stochastic approximation for the gradient $\nabla f(x)$ with mini-batch of size r

$$\nabla^r f(x, \{\xi_l\}_{l=1}^r) = \frac{1}{r}\sum_{l=1}^{r} \nabla f(x, \xi_l),$$

where each stochastic gradient $\nabla f(x, \xi_l)$ satisfies (2).

We start with a Lemma characterizing the decrease of the objective on one step of the algorithm.

Lemma 1 *With step size $h = \frac{1}{2L}$ in stochastic gradient descent* (3) *the following holds*

$$f(x^{k+1}) - f(x^k) \le -\frac{1}{4L}\|\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)\|_2^2 + \frac{1}{2L}\delta_{k+1}^2,$$

where $\delta_{k+1}^2 = \|\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r) - \nabla f(x^k)\|_2^2.$

Proof. From Lipschitz continuity of $\nabla f(x)$ we have

$$f(x^{k+1}) \le f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L}{2}\|x^{k+1} - x^k\|_2^2. \tag{4}$$

By Cauchy–Schwarz inequality and since $ab \leq a^2/2 + b^2/2$ for any a, b, we have

$$\langle \nabla f(x^k) - \nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r), x^{k+1} - x^k \rangle \leq \\ \frac{1}{2L}\|\nabla f(x^k) - \nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)\|_2^2 + \frac{L}{2}\|x^{k+1} - x^k\|_2^2.$$

Using this inequality and adding and subtracting

$\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)$ in (4) we get

$$f(x^{k+1}) \leq f(x^k) + \langle \nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r), x^{k+1} - x^k \rangle + \\ \frac{2L}{2}\|x^{k+1} - x^k\|_2^2 + \frac{1}{2L}\delta_{k+1}^2, \tag{5}$$

where $\delta_{k+1}^2 = \|\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r) - \nabla f(x^k)\|_2^2$.
From (5) and (3) we have

$$f(x^{k+1}) \leq \; f(x^k) - h\|\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)\|_2^2 \\ + L\|\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)\|_2^2 + \frac{1}{2L}\delta_{k+1}^2 = \\ f(x^k) - h(1 - Lh)\|\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)\|_2^2 + \frac{1}{2L}\delta_{k+1}^2.$$

Thus, the step size h is chosen as follows

$$h = \arg\max_{\alpha \geq 0} \alpha(1 - L\alpha) = \frac{1}{2L}.$$

Substituting this h to (5) we finalize the proof

$$f(x^{k+1}) - f(x^k) \leq -\frac{1}{4L}\|\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)\|_2^2 + \frac{1}{2L}\delta_{k+1}^2.$$

Algorithm 1 Stochastic Gradient Descent

```
Require: Number of iterations $N$, variance $D$, Lipschitz constant $L$, accuracy $\varepsilon$.
1: Calculate batch size
       $r = \max\{D/(L\varepsilon),\ 1\}$
2: for $k = 0, \dots, N-1$ do
3:
       $x^{k+1} = x^k - \frac{1}{2L}\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)$
4: end for
Ensure: $\bar{x}^N = \frac{1}{N}\sum_{k=1}^N x^k$.
```

Theorem 1 *Algorithm 1 with stochastic gradient oracle calls* $T = O\left(\frac{DR^2}{\varepsilon^2}\right)$*, batch size* $r = \max\{\frac{D}{L\varepsilon}, 1\}$ *and the number of iterations* $N = O\left(\frac{LR^2}{\varepsilon}\right)$ *outputs a point* $\bar{x}^N$ *satisfying*

$$\mathbb{E} f(\bar{x}^N) - f(x^*) \leq \varepsilon. \tag{6}$$

Proof.

Consider

$$\begin{aligned} &\|x^{k+1}-x\|_2^2 = \|x^k - x - h\nabla^{r_k} f(x, \{\xi_l^{k+1}\}_{l=1}^r)\|_2^2 \\ &= \|x^k - x\|_2^2 + h^2\|\nabla^{r_k} f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)\|_2^2 \\ &- 2h\langle x^k - x, \nabla^{r_k} f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)\rangle \end{aligned} \tag{7}$$

From (7) and Lemma 1 we get

$$\begin{aligned} &\langle \nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r), x^k - x\rangle \leq L\|x^k - x\|_2^2 \\ &- L\|x^{k+1}-x\|_2^2 + f(x^k) - f(x^{k+1}) + \frac{1}{2L}\delta_{k+1}^2. \end{aligned} \tag{8}$$

Taking the conditional expectation $\mathbb{E}_{x^{k+1}}[\,\cdot\,|x^1, \ldots, x^k]$ from both sides. With convexity condition we get

$$\begin{aligned} &f(x^k) - f(x) \leq \langle \nabla f(x), x^k - x\rangle \leq \\ &f(x^k) - \mathbb{E}_{x^{k+1}}[f(x^{k+1})|x^1, ..., x^k] + \frac{1}{2L}\mathbb{E}_{x^{k+1}}[\delta_{k+1}^2|x^1, ..., x^k] \\ &+ L\|x - x^k\|_2 - \mathbb{E}_{x^{k+1}}[L\|x - x^{k+1}\|_2^2 \mid x^1, ..., x^k]. \end{aligned}$$

Then we summarize this and take the total expectation

$$\mathbb{E}f(\bar{x}^N) - f(x^*) \leq \frac{LR^2}{2N} + \frac{1}{2L}\mathbb{E}\delta^2, \tag{9}$$

where we used $x = x^*$ and introduced upper bound $\delta \geq \delta_k$ for any k. We choose batch size r in respect with $\mathbb{E}\delta = \varepsilon L$. Since

$$\mathbb{E}\|\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r) - \nabla f(x^k)\|_2^2 \leq D/r$$

we get

$$r = \max\left\{\frac{D}{L\varepsilon}, 1\right\}. \tag{10}$$

We define total number of iterations N from (9) such that (6) holds. Summing r over all iterations we get the total number of oracle calls T.

2.2 Adaptive algorithm

Now we assume that the constant L may be unknown, moreover, if the true variance D is unavailable we use its upper bound $D_0 \geq D$. We provide an adaptive algorithm which iteratively tunes the Lipschitz constant. Importantly, the approximation of the Lipschitz constant used by the algorithm may decrease as iteration go, leading to larger steps and faster convergence.

Sinse Lipschitz constant L varies from iteration to iteration we need to define different batch size at each iteration. Using (10) we choose batch size as follows $r_{k+1} = \max\left\{\frac{D_0}{L_{k+1}\varepsilon}, 1\right\}$. Using (11) we similarly to (8) get the following

$$\begin{aligned} &\langle \nabla^{r_{k+1}} f(x, \{\xi_l^{k+1}\}_{l=1}^{r_{k+1}}), x^k - x\rangle \leq L_{k+1}\|x^k - x\|_2^2 \\ &- L_{k+1}\|x^{k+1} - x\|_2^2 + f(x^k) - f(x^{k+1}) + \varepsilon/2. \end{aligned} \tag{12}$$

Algorithm 2 Adaptive Stochastic Gradient Descent

```
Require: Number of iterations $N$, accuracy $\varepsilon$, $D_0$, initial guess $L_0$.
for $k = 0, \dots, N-1$ do
    $L_{k+1} := L_k/4$
    repeat
        $L_{k+1} := 2L_{k+1}$
        $$r_{k+1} = \max\{D_0/(L_{k+1}\varepsilon),\ 1\}$$
        $$x^{k+1} = x^k - \frac{1}{2L_{k+1}} \nabla^{r_{k+1}} f(x^k, \{\xi_l^{k+1}\}_{l=1}^{r_{k+1}})$$
    until
        $$f(x^{k+1}) \leq f(x^k) + \langle \nabla^{r_{k+1}} f(x, \{\xi_l^{k+1}\}_{l=1}^{r_{k+1}}), x^{k+1} - x^k \rangle + L_{k+1}\|x^{k+1} - x^k\|_2^2 + \varepsilon/2 \tag{11}$$
end for
Ensure: $\bar{x}_N = \frac{1}{N}\sum_{k=1}^{N} x^k$.
```

Since L_{k+1} is random now, r_{k+1} will be random as well and, consequently, the total number of oracle calls T is not precisely determined. Let us choose it similarly to its counterpart in Theorem 1 which ensures (6)

$$T = \sum_{k=1}^{N-1} r_{k+1} = O\left(\frac{D_0 R^2}{\varepsilon^2}\right). \tag{13}$$

This number of oracle calls (13) can be provided by choosing the last batch size r_N as a residual of the sum (13) and calculate the last Lipschitz constant $L_N = \frac{D_0}{r_N \varepsilon}$. In practice, we do not need to limit ourselves by fixing the number of oracle calls T.

From the convexity of f we have

$$f(x^k) - f(x) \leq \langle \nabla f(x), x^k - x \rangle. \tag{14}$$

Then from (14) we get

$$\begin{aligned} &\langle \nabla^{r_k} f(x, \{\xi_l^{k+1}\}_{l=1}^{r}), x^k - x \rangle \geq (f(x^k) - f(x)) \\ &+ \langle \nabla^{r_k} f(x, \{\xi_l^{k+1}\}_{l=1}^{r}) - \nabla f(x^k), x^k - x \rangle. \end{aligned}$$

From this and (12) we have

$$\begin{aligned} &\frac{1}{L_{k+1}}(f(x^k) - f(x)) \\ &+ \frac{1}{L_{k+1}} \langle \nabla^{r_{k+1}} f(x, \{\xi_l^{k+1}\}_{l=1}^{r_{k+1}}) - \nabla f(x^k), x^k - x \rangle \\ &\leq \|x^k - x\|_2^2 - \|x^{k+1} - x\|_2^2 \\ &+ \frac{1}{L_{k+1}}\left(f(x^k) - f(x^{k+1})\right) + \frac{\varepsilon}{2L_{k+1}}. \end{aligned}$$

The same proof steps with taking conditional expectation, summing progress over iterations and taking the full expectation as for the non-adaptive case fails here. This happens due to this fact: the following sum $\sum_{k=0}^{N-1}\frac{1}{L_{k+1}}\langle\nabla^{r_{k+1}}f(x^k,\{\xi_l^{r_{k+1}}\}_{l=1}^{r_{k+1}}) - \nabla f(x^k), x^k - x\rangle = \frac{\epsilon}{D_0}\sum_{i=0}^{T-1}\langle\nabla f(x^{k(i)},\xi_i) - \nabla f(x^{k(i)}), x^{k(i)} - x\rangle$ is not the sum of martingale-differences and therefore the total expectation is not zero, since r_k is random. Thus, unfortunately, we cannot guarantee that Algorithm 2 converges in $O\left(\frac{LR^2}{\varepsilon}\right)$ iterations. However, numerical experiments are in a good agreement with the provided complexity bound.

2.3 Accelerated adaptive algorithm

To compare our complexity bounds for adaptive stochastic gradient descent with the bounds for accelerated variant of our algorithm we refer to [36]. For the reader convenience we provide that accelerated algorithm in a simpler form and complexity bounds presented there without proof.

Algorithm 3 Adaptive Stochastic Accelerated Gradient Method

```
Require: Number of iterations $N$, $D_0$ accuracy $\varepsilon$, $\Omega \geq 1$, $A_0 = 0$, initial guess $L_0$.
for $k = 0, \dots, N-1$ do
    $L_{k+1} := L_k/4$
    repeat
        $L_{k+1} := 2L_{k+1}$
        $\alpha_{k+1} = (1 + \sqrt{1 + 8A_kL_{k+1}})/(4L_{k+1})\ ;\ A_{k+1} = A_k + \alpha_{k+1}$
        $r_{k+1} = \max\{\Omega\alpha_{k+1}D_0/\varepsilon,\ 1\}$
        $y^{k+1} = (\alpha_{k+1}u^k + A_kx^k)/A_{k+1}$
        $u^{k+1} = u^k - \alpha_{k+1}\nabla^{r_{k+1}}f(y^{k+1}, \{\xi_l^{k+1}\}_{l=1}^{r_{k+1}})$
        $x^{k+1} = (\alpha_{k+1}u^{k+1} + A_kx^k)/A_{k+1}$
    until
        $f(x^{k+1}) \leq f(y^{k+1}) + \langle\nabla^{r_{k+1}}f(y^{k+1}, \{\xi_l^{k+1}\}_{l=1}^{r_{k+1}}), x^{k+1} - y^{k+1}\rangle + L_{k+1}\|x^{k+1} - y^{k+1}\|_2^2 + \Omega D_0/(L_{k+1}r_{k+1})$   (15)
end for
Ensure: $x^N$
```

For Algorithm 3 the number of oracle calls T will be the same as for the non-accelerated version of the algorithm

$$T = \sum_{k=1}^{N-1} r_{k+1} = O\left(\frac{D_0 R^2}{\varepsilon^2}\right)$$

while the number of iterations will be smaller $N = O\left(\sqrt{LR^2/\varepsilon}\right)$. Both these bounds are optimal [42].

Unfortunately, to prove these bounds we also met the problem of martingale-differences mentioned above. We expect that original technique from the paper [23] sheds light on how one can try to resolve it and we defer the complete proof to a future version of this paper.

2.4 Practical implementation of adaptive algorithms

Next we comment on applicability of Algorithm 2 and Algorithm 3 in real problems. Generally, in case when the exact gradients of function $f(x^k)$ is unavailable, function values itself of $f(x^k)$ are also unavailable. It holds, e.g. in stochastic optimization problem, where the objective is presented by its expectation

$$f(x) = \mathbb{E} f(x, \xi). \tag{16}$$

In this case we estimate the function as a sample average

$$f(x, \{\xi_l\}_{l=1}^r) = \frac{1}{r}\sum_{l=1}^r f(x, \xi_l)$$

and use it in adaptive procedures. In this case we interpret L_k as the worst constant among all Lipschitz constants for $f(x, \xi)$ with different realization of ξ. Indeed, if L_{k+1} satisfies the following

$$f(x^{k+1}, \xi^{k+1}) \le f(x^k, \xi^{k+1}) + \langle \nabla f(x^k, \xi), x^{k+1} - x^k \rangle \\ + L_{k+1}\|x^{k+1} - x^k\|_2^2 + \varepsilon/2.$$

Then it satisfies

$$f(x^{k+1}, \{\xi_l^{k+1}\}_{l=1}^{r_{k+1}}) \le f(x^k, \{\xi_l^{k+1}\}_{l=1}^{r_{k+1}}) + \\ \langle \nabla^{r_{k+1}} f(x, \{\xi_l^{k+1}\}_{l=1}^{r_{k+1}}), x^{k+1} - x^k \rangle \\ + L_{k+1}\|x^{k+1} - x^k\|_2^2 + \varepsilon/2. \tag{17}$$

If, e.g, (16) holds we replace adaptive procedure in the algorithms by (17).

We also comment on batch size. If the batch size r_k decreases during the process of L_k selection, we preserve r_k from the previous iteration in order not to recalculate stochastic approximation $\nabla^{r_{k+1}} f(x, \{\xi_l^{k+1}\}_{l=1}^{r_{k+1}})$.

All these remarks remain true also in non-convex case.

3 Stochastic non-convex optimization

In this section we assume that the objective f may be non-convex. As in the previous section we consider two cases: known and unknown Lipschitz constant L.

3.1 Non-adaptive algorithm

Algorithm 4 Non-convex Stochastic Gradient Descent

```
Require: Number of iterations $N$, variance $D$, Lipschitz constant $L$, accuracy $\varepsilon$
1: Calculate
       $r = \max\{12D/(\varepsilon^2),\ 1\}$
2: for $k = 0, \dots, N-1$ do
3:     $x^{k+1} = x^k - \frac{1}{2L}\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)$
4: end for
Ensure: $\hat{x} = \arg\min_{k=1,..N} \|\nabla f(x^k)\|_2$.
```

The next Lemma provides general quite simple inequality which is necessary to prove complexity bounds.

Lemma 2 *For any $a, b \in \mathbb{R}^n$ the following holds*

$$\|a\|^2 \le 2\|b\|^2 + 2\|a-b\|^2.$$

Proof. From triangle inequality

$$\|a\| = \|b + a - b\| \le \|b\| + \|a-b\|.$$

Thus,

$$\|a\|^2 \le (\|b\| + \|a-b\|)^2. \tag{18}$$

We use the following inequality

$$\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$$

for (18) and finish the proof

$$\|a\|^2 \le (\|b\| + \|a-b\|)^2 \le 2\|b\|^2 + 2\|a-b\|^2.$$

Theorem 2 *Algorithm 4 with the total number of stochastic gradient oracle calls*[2] $T = O\left(\frac{DL(f(x^0)-f(x^*))}{\varepsilon^4}\right)$ *and number of iterations* $N = O\left(\frac{L(f(x^0)-f(x^*))}{\varepsilon^2}\right)$ *outputs a point*[3] $\hat{x}^N$ *which satisfies*

$$\|\nabla f(\hat{x}^N)\|_2 \le \varepsilon. \tag{19}$$

Proof. We use Lemma 2 to get the following inequality

$$\begin{aligned}\|\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)\|_2^2 &\ge \frac{1}{2}\|\nabla f(x^k)\|_2^2 \\ &- \|\nabla f(x^k) - \nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)\|_2^2.\end{aligned} \tag{20}$$

[2] According to recent works [5, 8], T and N corresponds to lower bounds.

[3] This $\hat{x}$ is is difficult to calculate in practice. Therefore, we refer to the paper [20], in which this problem is partially solved.

In non-convex case Lemma 1 remains true. Using it and (20) we get (see also [15])

$$f(x^{k+1}) - f(x^k) \leq -\frac{1}{8L}\|\nabla f(x^k)\|_2^2 + \frac{3}{4L}\delta_{k+1}^2.$$

If $\mathbb{E}\delta_k \leq \frac{\varepsilon^2}{12}$ for any k, then to achieve convergence in the norm of the gradient (19), we need to do $N = 16L(f(x^0) - f(x^*))/\varepsilon^2$ iterations.
Then we can define batch size from

$$\mathbb{E}\delta_{k+1}^2 = \mathbb{E}\|\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r) - \nabla f(x^k)\|_2^2 = \frac{D}{r} \leq \frac{\varepsilon^2}{12}$$

Consequently, $r = \frac{12D}{\varepsilon^2}$. Summing r over all N iterations we get the total number of oracle calls T.

3.2 Adaptive algorithm

Algorithm 5 Adaptive Non-convex Stochastic Gradient Descent

```
Require: Number of iterations $N$, $D_0$, accuracy $\varepsilon$, initial guess $L_0$
Calculate
    $r = \max\{8D_0/(\varepsilon^2),\ 1\}$
for $k = 0, \dots, N-1$ do
    $L_{k+1} := L_k/4.$
    repeat
        $L_{k+1} := 2L_{k+1}$
        $x^{k+1} = x^k - \frac{1}{2L_{k+1}}\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)$
    until
        $f(x^{k+1}) \leq f(x^k) + \langle \nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r), x^{k+1} - x^k \rangle + L_{k+1}\|x^{k+1} - x^k\|_2^2 + \frac{\varepsilon^2}{32L_{k+1}}$
end for
Ensure: $\hat{x} = \arg\min_{k=1,..N} \|\nabla f(x^k)\|_2.$
```

Theorem 3 *Algorithm 5 with expected number of stochastic gradient oracle calls* $\tilde{T} = O\left(\frac{D_0 L(f(x^0) - f(x^N))}{\varepsilon^4}\right)$ *and expected number of iterations* $\tilde{N} = O\left(\frac{L(f(x^0) - f(x^N))}{\varepsilon^2}\right)$ *outputs a point* $\hat{x}$ *satisfying*

$$\|\nabla f(\hat{x})\|_2 \leq \varepsilon.$$

Proof. [*Sketch of the Proof*] From (15) using Lemma 1 we get

$$f(x^{k+1}) - f(x^k) \leq -\frac{1}{4L_{k+1}}\|\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r)\|_2^2 + \frac{\varepsilon^2}{32L_{k+1}}. \tag{21}$$

From (20) and (21) and we have

$$f(x^{k+1}) - f(x^k) \leq -\frac{1}{8L_{k+1}}\|\nabla f(x^k)\|_2^2 + \frac{1}{4L_{k+1}}\delta_{k+1}^2 + \frac{\varepsilon^2}{32L_{k+1}}.$$

If $\|\nabla f(x^k)\|_2^2 \geq \varepsilon^2$. Then

$$f(x^{k+1}) - f(x^k) \leq -\frac{3\varepsilon^2 - 8\delta_{k+1}^2}{32L_{k+1}}. \tag{22}$$

Based on iterated procedure (15) we may expect that $L_{k+1} \leq 2L$. The exact proof of this fact in probability of large deviations terminology was provided in [36] (numerical coefficient needs to be corrected). In our work, we limit ourselves by assuming this inequality holds ïn average".

If $3\varepsilon^2 - 8\delta_{k+1}^2 \geq 0$ we may replace L_{k+1} by $2L$. Therefore, we rewrite (22) with minor changing and after taking the expectation we get

$$\mathbb{E}f(x^{k+1}) - \mathbb{E}f(x^k) \leq -\frac{2\varepsilon^2 - 8\mathbb{E}\delta_{k+1}^2}{64L}.$$

Ensuring $\mathbb{E}\delta_{k+1}^2 \geq \frac{\varepsilon^2}{8}$ we obtain

$$\mathbb{E}f(x^{k+1}) - \mathbb{E}f(x^k) \leq -\frac{\varepsilon^2}{64L}.$$

Summing this over expected number of iteration we get

$$\tilde{N} = 64L(f(x^0) - f(x^*))/\varepsilon^2. \tag{23}$$

This $\tilde{N}$ ensures that for some k we get $\|\nabla f(x^k)\|_2^2 \leq \varepsilon^2$.

We choose the batch size according to

$$\mathbb{E}\delta_{k+1}^2 = \mathbb{E}\|\nabla^r f(x^k, \{\xi_l^{k+1}\}_{l=1}^r) - \nabla f(x^k)\|_2^2 = \frac{\varepsilon^2}{8} \leq \frac{D_0}{r}.$$

Consequently, $r = \frac{8D_0}{\varepsilon^2}$. Using the expected number of algorithm iterations (23) we get expected number of oracle calls

$$\tilde{T} = \tilde{N}r = 512D_0L(f(x^0) - f(x^N))/\varepsilon^4.$$

More accurate proof of Theorem 3 can be done using large deviations technique and sub-Gaussian variance similarly to [36].

4 Experiments

We perform experiments using proposed methods with and without acceleration on convex and non-convex problems and compare results with commonly used methods — Adam, [24] and Adagrad, [9]. Experiments consist of three problems:

1 Training logistic regression on MNIST dataset [26] (convex problem). Number of optimized parameters is 7850.

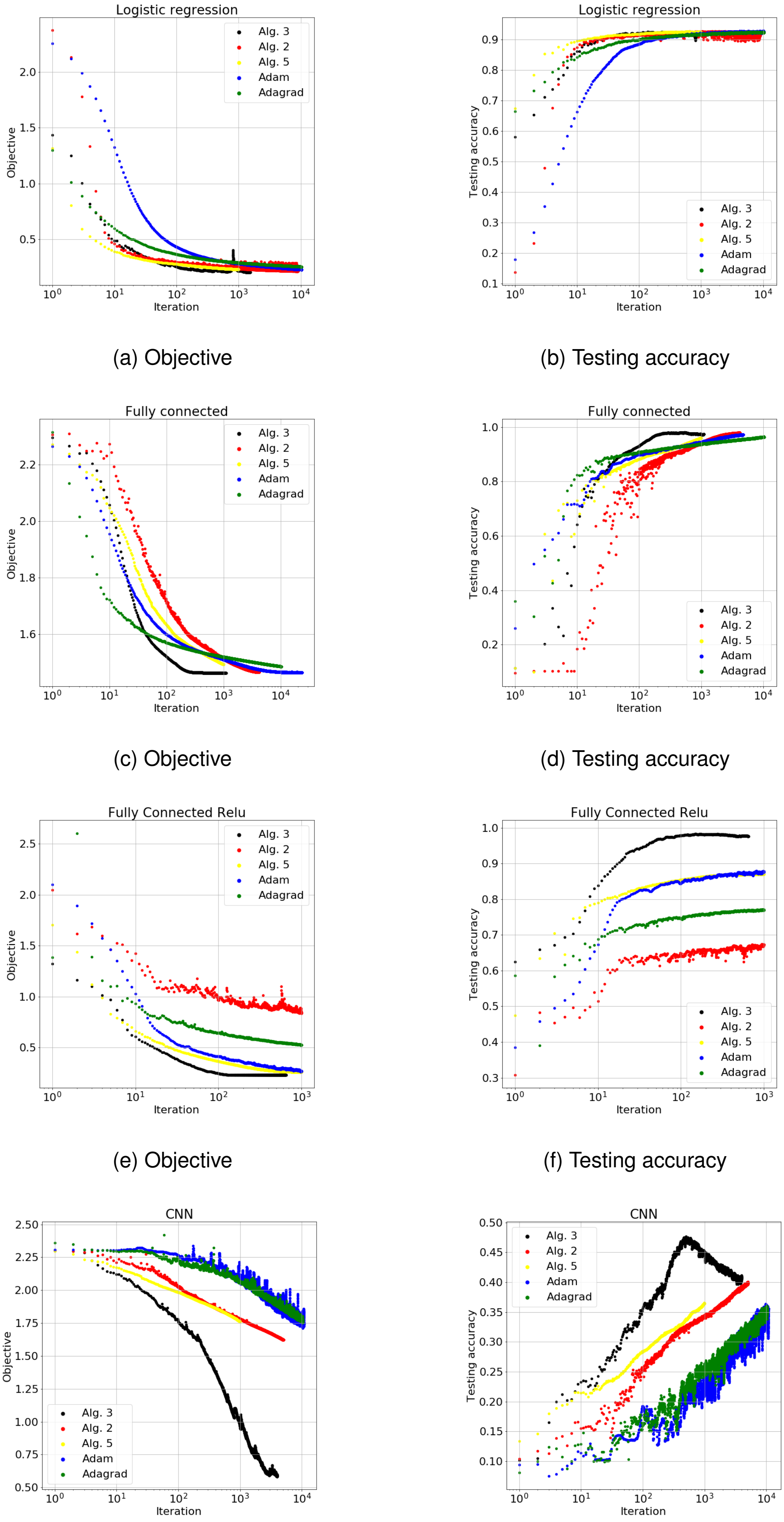
2 Training fully-connected sigmoid-activated neural network with two hidden layers of size 1000 on MNIST dataset (non-convex problem). Number of optimized parameters is 795010.

3 Training fully-connected *relu-activated* neural network with two hidden layers of size 1000 also on MNIST dataset (*non-differentiable* and non-convex problem). Number of optimized parameters is 795010.

4 Training small convolutional neural network with three filters and three fully-connected layers on CIFAR10 dataset [25] (non-convex problem). Number of optimized parameters is 62000.

Objective for all the problems is cross-entropy function between predicted class distribution and ground-truth class. Hyperparameters for proposed methods were the same for every problem, i.e. $D_0 = 0.1, L_0 = 1, \varepsilon = 0.002$. This hyperparameter set is chosen experimentally to obtain universal hyperparameters for broad range of settings. Adam and Adagrad had batch size equal to 128, learning rate $= 0.001$ and $\beta_1 = 0.9, \beta_2 = 0.999$ — these parameters are frequently used in various machine learning tasks and are used in [24]. Since our problems come from machine learning domain we measure not only the progress in the objective function values, but also the accuracy on the test set that we choose in advance. Test set size for all problems is 10K samples. Training set is 60K samples for MNIST dataset and 50K samples for CIFAR10 dataset. Dynamics of objective function value on training set and testing accuracy for every task are depicted on Fig 1. The code for all algorithms is available, visit[4].

[4] https://github.com/alexo256/Adaptive-Gradient-Descent-for-Convex-and-Non-Convex-Stochastic-Optimization



(a) Objective

(b) Testing accuracy

(c) Objective

(d) Testing accuracy

(e) Objective

(f) Testing accuracy

(g) Objective

(h) Testing accuracy

Figure 1: Experiments

5 Conclusion

In this paper we focus on adaptive methods for stochastic convex and non-convex optimization. It would be interesting to combine these ideas with notion of inexact model of the objective function and inexact model [40] of the operator in variational inequalities [12, 17, 39] to obtain adaptive and universal methods using stochastic inexact model. We leave this for future work.

Note also, that if we replace line 2 in Algorithms 2 and 3 to $L_{k+1} := L_k/2$, take $r_{k+1} \equiv \max\{\frac{2D_0}{L_k\varepsilon}, 1\}$ and forbid L_{k+1} to be outside the range $[L_d, L_u]$, where $L_d \equiv L_0 \equiv L_u$ mod 2, $L_0 \in [L_d, L_u]$, then based on union bound inequality and theory of empirical process [21] one can prove the desired estimates up to a logarithmic factors.

References

[1] F. Bach and K. Y. Levy. A universal algorithm for variational inequalities adaptive to smoothness and noise. In A. Beygelzimer and D. Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 164–194, Phoenix, USA, 25–28 Jun 2019. PMLR. arXiv:1902.01637.

[2] A. Bayandina, P. Dvurechensky, A. Gasnikov, F. Stonyakin, and A. Titov. Mirror descent and convex optimization problems with non-smooth inequality constraints. In P. Giselsson and A. Rantzer, editors, *Large-Scale and Distributed Optimization*, chapter 8, pages 181–215. Springer International Publishing, 2018. arXiv:1710.06612.

[3] L. Bogolubsky, P. Dvurechensky, A. Gasnikov, G. Gusev, Y. Nesterov, A. M. Raigorodskii, A. Tikhonov, and M. Zhukovskii. Learning supervised pagerank with gradient-based and gradient-free optimization methods. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4914–4922. Curran Associates, Inc., 2016. arXiv:1603.00717.

[4] R. H. Byrd, G. M. Chin, J. Nocedal, and Y. Wu. Sample size selection in optimization methods for machine learning. *Mathematical Programming*, 134(1):127–155, 2012.

[5] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Lower bounds for finding stationary points ii: First-order methods. *arXiv preprint arXiv:1711.00841*, 2017.

[6] Q. Deng, Y. Cheng, and G. Lan. Optimal adaptive and accelerated stochastic gradient descent. *arXiv:1810.00553*, 2018.

[7] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1):37–75, 2014.

[8] Y. Drori and O. Shamir. The complexity of finding stationary points with stochastic gradient descent. *arXiv preprint arXiv:1910.01845*, 2019.

[9] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul.):2121–2159, 2011.

[10] P. Dvurechensky. Gradient method with inexact oracle for composite non-convex optimization. *arXiv:1703.09180*, 2017.

[11] P. Dvurechensky, A. Gasnikov, and A. Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1367–1376, 2018. arXiv:1802.04367.

[12] P. Dvurechensky, A. Gasnikov, F. Stonyakin, and A. Titov. Generalized Mirror Prox: Solving variational inequalities with monotone operator, inexact oracle, and unknown Hölder parameters. *arXiv:1806.05140*, 2018.

[13] M. P. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, 2012.

[14] A. Gasnikov. Universal gradient descent. *arXiv preprint arXiv:1711.00394*, 2017.

[15] A. V. Gasnikov, P. Dvurechenskii, M. E. Zhukovskii, S. V. Kim, S. S. Plaunov, D. A. Smirnov, and F. A. Noskov. About the power law of the pagerank vector distribution. part 2. backley–osthus model, power law verification for this model and setup of real search engines. *Sibirskii Zhurnal Vychislitel'noi Matematiki*, 21(1):23–45, 2018.

[16] A. V. Gasnikov and P. E. Dvurechensky. Stochastic intermediate gradient method for convex optimization problems. *Doklady Mathematics*, 93(2):148–151, Mar 2016.

[17] A. V. Gasnikov, P. E. Dvurechensky, F. S. Stonyakin, and A. A. Titov. An adaptive proximal method for variational inequalities. *Computational Mathematics and Mathematical Physics*, 59(5):836–841, May 2019.

[18] A. V. Gasnikov, P. E. Dvurechensky, M. E. Zhukovskii, S. V. Kim, S. S. Plaunov, D. A. Smirnov, and F. A. Noskov. About the power law of the pagerank vector component distribution. Part 2. The Buckley–Osthus model, verification of the power law for this model, and setup of real search engines. *Numerical Analysis and Applications*, 11(1):16–32, 2018.

[19] N. Gazagnadou, R. M. Gower, and J. Salmon. Optimal mini-batch and step sizes for saga. *arXiv preprint arXiv:1902.00071*, 2019.

[20] S. Ghadimi and G. Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[21] E. Giné and R. Nickl. *Mathematical foundations of infinite-dimensional statistical models*, volume 40. Cambridge University Press, 2016.

[22] S. V. Guminov, Y. E. Nesterov, P. E. Dvurechensky, and A. V. Gasnikov. Accelerated primal-dual gradient descent with linesearch for convex, nonconvex, and nonsmooth optimization problems. *Doklady Mathematics*, 99(2):125–128, Mar 2019.

[23] A. N. Iusem, A. Jofré, R. I. Oliveira, and P. Thompson. Variance-based extragradient methods with line search for stochastic variational inequalities. *SIAM Journal on Optimization*, 29(1):175–206, 2019. arXiv:1703.00262.

[24] D. Kingma and J. Ba. Adam: a method for stochastic optimization. *ICLR*, 2015.

[25] A. Krizhevsky. Learning multiple layers of features from tiny images. phd thesis. Technical report, University of Toronto, 2009.

[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[27] K. Y. Levy, A. Yurtsever, and V. Cevher. Online adaptive methods, universality and acceleration. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6500–6509. Curran Associates, Inc., 2018. arXiv:1809.02864.

[28] Y. Malitsky and T. Pock. A first-order primal-dual algorithm with linesearch. *SIAM Journal on Optimization*, 28(1):411–432, 2018.

[29] A. Nemirovski. Orth-method for smooth convex optimization. *Izvestia AN SSSR, Transl.: Eng. Cybern. Soviet J. Comput. Syst. Sci*, 2:937–947, 1982.

[30] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[31] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013. First appeared in 2007 as CORE discussion paper 2007/76.

[32] Y. Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152(1):381–404, 2015.

[33] Y. Nesterov. *Lectures on convex optimization*, volume 137. Springer International Publishing, 2018.

[34] Y. Nesterov, A. Gasnikov, S. Guminov, and P. Dvurechensky. Primal-dual accelerated gradient methods with small-dimensional relaxation oracle. *arXiv:1809.05895*, 2018.

[35] D. Newton, F. Yousefian, and R. Pasupathy. *Stochastic Gradient Descent: Recent Trends*, chapter 9, pages 193–220. INFORMS, 2018.

[36] A. Ogaltsov and A. Tyurin. Heuristic adaptive fast gradient method in stochastic optimization tasks. *arXiv:1910.04825*, 2019.

[37] B. Polyak. *Introduction to Optimization*. New York, Optimization Software, 1987.

[38] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv:1609.04747*, 2016.

[39] F. Stonyakin, A. Gasnikov, A. Tyurin, D. Pasechnyuk, A. Agafonov, P. Dvurechensky, D. Dvinskikh, A. Kroshnin, and V. Piskunova. Inexact model: A framework for optimization and variational inequalities. *arXiv:1902.00990*, 2019.

[40] F. S. Stonyakin, D. Dvinskikh, P. Dvurechensky, A. Kroshnin, O. Kuznetsova, A. Agafonov, A. Gasnikov, A. Tyurin, C. A. Uribe, D. Pasechnyuk, and S. Artamonov. Gradient methods for problems with inexact model of the objective. In M. Khachay, Y. Kochetov, and P. Pardalos, editors, *Mathematical Optimization Theory and Operations Research*, pages 97–114, Cham, 2019. Springer International Publishing. arXiv:1902.09001.

[41] R. Ward, X. Wu, and L. Bottou. AdaGrad stepsizes: Sharp convergence over nonconvex landscapes. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6677–6686, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[42] B. E. Woodworth, J. Wang, A. Smith, B. McMahan, and N. Srebro. Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. In *Advances in neural information processing systems*, pages 8496–8506, 2018.

[43] A. Yurtsever, Q. Tran-Dinh, and V. Cevher. A universal primal-dual convex optimization framework. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 3150–3158, Cambridge, MA, USA, 2015. MIT Press.