Hyperfast Second-Order Local Solvers for Efficient Statistically Preconditioned Distributed Optimization

Kamzolov Dmitry
Moscow Institute of Physics and Technology

https://arxiv.org/pdf/2102.08246
Joint work with: P. Dvurechensky, A. Lukashevich,
S. Lee, E. Ordentlich, C. Uribe, A. Gasnikov
Supported by RFBR projects no. 19-31-27001
the Ministry of Science and Higher Education of the Russian Federation (Goszadaniye) No. 075-00337-20-03, project No. 0714-2020-0005

June 2, 2021

Plan

- High-order methods
- Hyperfast Second-Order Method
- Statistically Preconditioned Distributed Method

Problem formulation

High-order convex problem

$$\min f(x),$$

$f(x)$ is convex function with Lipschitz p-th derivative with constant L_p

Problem formulation

Lipschitz derivative

$$\|D^p f(x) - D^p f(y)\| \leq L_p \|x - y\|$$

Problem formulation

Lipschitz derivative

$$\|D^p f(x) - D^p f(y)\| \leq L_p \|x - y\|$$

Taylor approximation

$$\Omega_p(f, x; y) = f(x) + \sum_{k=1}^{p} \frac{1}{k!} D^k f(x) [y - x]^k, y \in E$$

Problem formulation

Lipschitz derivative

$$\|D^p f(x) - D^p f(y)\| \leq L_p \|x - y\|$$

Taylor approximation

$$\Omega_p(f, x; y) = f(x) + \sum_{k=1}^{p} \frac{1}{k!} D^k f(x) \left[y - x\right]^k, y \in E$$

Corrolary

$$|f(y) - \Omega_p(f, x; y)| \leq \frac{L_p}{(p+1)!} \|y - x\|^{p+1}$$

Basic Method

Basic step [Nesterov, 2018]

$$T_{H_p}(x) = \operatorname*{argmin}_y \left\{ \tilde{\Omega}_{p,H_p}(f, x; y) \right\},$$

where

$$\tilde{\Omega}_{p,H_p}(f, x; y) = \Omega_p(f, x; y) + \frac{H_p}{p!} \|y - x\|^{p+1}.$$

For $H_p \geq L_p$ this subproblem is convex and hence implementable.

Basic Method

Basic step [Nesterov, 2018]

$$T_{H_p}(x) = \operatorname*{argmin}_y \left\{ \tilde{\Omega}_{p,H_p}(f, x; y) \right\},$$

where

$$\tilde{\Omega}_{p,H_p}(f, x; y) = \Omega_p(f, x; y) + \frac{H_p}{p!}\|y - x\|^{p+1}.$$

For $H_p \geq L_p$ this subproblem is convex and hence implementable.

Convergence

$$f(x_N) - f(x_*) \leq O\left(\frac{H_p R^{p+1}}{N^p}\right)$$

Accelerated Method [Nesterov, 2018]

Accelerated Tensor Method

Initialization: Choose $x_0 \in \mathbb{E}$ and $M > L_p$. Compute $x_1 = T_{p,M}(x_0)$.

Define $C = \frac{p}{2}\sqrt{\frac{(p+1)}{(p-1)}(M^2 - L_p^2)}$ and $\psi_1(x) = f(x_1) + \frac{C}{p!} d_{p+1}(x - x_0)$.

Iteration k**, (**$k \geq 1$**):**

1. Compute $v_k = \arg\min\limits_{x \in \mathbb{E}} \psi_k(x)$ and choose $y_k = \frac{A_k}{A_{k+1}} x_k + \frac{a_k}{A_{k+1}} v_k$.

2. Compute $x_{k+1} = T_{p,M}(y_k)$ and update

$$\psi_{k+1}(x) = \psi_k(x) + a_k[f(x_{k+1}) + \langle \nabla f(x_{k+1}), x - x_{k+1} \rangle].$$

Accelerated Method

Accelerated Convergence

$$F(x_N) - F(x^*) \leq O\left(\frac{H_p R^{p+1}}{N^{p+1}}\right)$$

Near-optimal tensor method [Gasnikov, et. al. 2019]

Algorithm 1 Accelerated Taylor Descent

Input: convex function $f : \mathbb{R}^d \to \mathbb{R}$ such that $\nabla^p f$ is L_p-Lipschitz.
Set $A_0 = 0, x_0 = y_0 = 0$
for $k = 0$ **to** $k = K - 1$ **do**
  Compute a pair $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^d$ such that

$$\frac{1}{2} \leq \lambda_{k+1} \frac{L_p \cdot \|y_{k+1} - \widetilde{x}_k\|^{p-1}}{(p-1)!} \leq \frac{p}{p+1},$$

  where

$$y_{k+1} = \arg\min_y \left\{ f_p(y; \widetilde{x}_k) + \frac{L_p}{p!} \|y - \widetilde{x}_k\|^{p+1} \right\},$$

  and

$$a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1} A_k}}{2}, \; A_{k+1} = A_k + a_{k+1}, \text{ and } \widetilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k .$$

  Update $x_{k+1} := x_k - a_{k+1} \nabla f(y_{k+1})$
end for
return y_K

Near-optimal tensor method [Gasnikov, et. al. 2019]

Optimal convergence

$$F(x_N) - F(x^*) \leq \tilde{O}\left(\frac{H_p R^{p+1}}{N^{\frac{3p+1}{2}}}\right)$$

Inexact subproblem solution

Inexact solution

Firstly, we introduce the definition of the inexact subproblem solution. Any point from the set

$$\mathcal{N}^{\gamma}_{p,H_p}(x) = \left\{ T \in \mathbb{R}^n \,:\, \|\nabla \tilde{\Omega}_{p,H_p}(f, x; T)\| \leq \gamma \|\nabla f(T)\| \right\}$$

is the inexact subproblem solution, where $\gamma \in [0; 1]$ is an accuracy parameter. $\mathcal{N}^{0}_{p,H_p}$ is the exact solution of the subproblem.

Inexact subproblem solution

Algorithm 1 Inexact pth-Order Near-optimal Accelerated Tensor Method (NATMI)

1: **Input:** convex function $f : \mathbb{R}^n \to \mathbb{R}$ such that $\nabla^p f$ is L_p-Lipschitz, $H_p = \xi L_p$ where ξ is a scaling parameter, γ is a desired accuracy of the subproblem solution.

2: Set $A_0 = 0, x_0 = y_0$
3: **for** $k = 0$ **to** $k = K - 1$ **do**
4: Compute a pair $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^n$ such that

$$\frac{1}{2} \leq \lambda_{k+1} \frac{H_p \cdot \|y_{k+1} - \tilde{x}_k\|^{p-1}}{(p-1)!} \leq \frac{p}{p+1},$$

where

$$y_{k+1} \in \mathcal{N}^{\gamma}_{p,H_p}(\tilde{x}_k)$$

and

$$a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1} A_k}}{2}, \; A_{k+1} = A_k + a_{k+1}$$

$$\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k .$$

5: Update $x_{k+1} := x_k - a_{k+1} \nabla f(y_{k+1})$
6: **return** y_K

Algorithm 3 Bregman-Distance Gradient Method

1: Set $z_0 = \tilde{x}_k$ and $\tau = \frac{3\delta}{8(2+\sqrt{2})\|\nabla f(\tilde{x}_k)\|}$

2: Set objective function

$$\varphi_k(z) = \langle \nabla f(\tilde{x}_k), z - \tilde{x}_k \rangle + \frac{1}{2}\nabla^2 f(\tilde{x}_k)[z-\tilde{x}_k]^2 + \frac{1}{6}D^3 f(\tilde{x}_k)[z-\tilde{x}_k]^3 + \frac{L_3}{4}\|z-\tilde{x}_k\|^4$$

3: Set feasible set

$$S_k = \left\{ z : \|z - \tilde{x}_k\| \leq 2\left(\frac{2+\sqrt{2}}{L_3}\|\nabla f(\tilde{x}_k)\|\right)^{\frac{1}{3}} \right\}$$

4: Set scaling function

$$\rho_k(z) = \frac{1}{2}\left\langle \nabla^2 f(\tilde{x}_k)(z-\tilde{x}_k), z - \tilde{x}_k \right\rangle + \frac{L_3}{4}\|z-\tilde{x}_k\|^4$$

5: **for** $k \geq 0$ **do**

6: Compute the approximate gradient $g_{\varphi_k,\tau}(z_i)$

7: **IF** $\|g_{\varphi_k,\tau}(z_i)\| \leq \frac{1}{6}\|\nabla f(z_i)\| - \delta$, then **STOP**

8: **ELSE** $z_{i+1} = \underset{z \in S_k}{\operatorname{argmin}} \left\{ \langle g_{\varphi_k,\tau}(z_i), z - z_i \rangle + 2\left(1 + \frac{1}{\sqrt{2}}\right)\beta_{\rho_k}(z_i, z) \right\}$,

9: **return** z_i

BDGM [Nesterov 2020]

Bregman distance

$\beta_{\rho_k}(z_i, z)$ is a Bregman distance generated by $\rho_k(z)$

$$\beta_{\rho_k}(z_i, z) = \rho_k(z) - \rho_k(z_i) - \langle \nabla \rho_k(z_i), z - z_i \rangle .$$

Inexact third-order derivative

$$g_{\varphi_k,\tau}(z) = \nabla f(\tilde{x}_k) + \nabla^2 f(\tilde{x}_k)[z - \tilde{x}_k] + g^{\tau}_{\tilde{x}_k}(z)/2 + L_3 \|z - \tilde{x}_k\|^2 (z - \tilde{x}_k)$$

$$g^{\tau}_{\tilde{x}_k}(z) = \frac{1}{\tau^2} \left(\nabla f(\tilde{x}_k + \tau(z - \tilde{x}_k)) + \nabla f(\tilde{x}_k - \tau(z - \tilde{x}_k)) - 2\nabla f(\tilde{x}_k) \right) .$$

Hyperfast Method

Algorithm 2 Hyperfast Second-Order Method

1: **Input:** convex function $f : \mathbb{R}^n \to \mathbb{R}$ with L_3-Lipschitz 3rd-order derivative.
2: Set $A_0 = 0, x_0 = y_0$
3: **for** $k = 0$ **to** $k = K - 1$ **do**
4: Compute a pair $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^n$ such that

$$\frac{1}{2} \le \lambda_{k+1} \frac{3L_3 \cdot \|y_{k+1} - \tilde{x}_k\|^2}{4} \le \frac{3}{4},$$

where $y_{k+1} \in \mathcal{N}^{1/6}_{3,3L_3/2}(\tilde{x}_k)$ solved by Algorithm 3 and

$$a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1} A_k}}{2}, \quad A_{k+1} = A_k + a_{k+1}$$

$$\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k .$$

5: Update $x_{k+1} := x_k - a_{k+1} \nabla f(y_{k+1})$
6: **return** y_K

Theorem

Theorem

Let f be a convex function whose third derivative is L_3-Lipschitz and x_* denote a minimizer of f. Then to reach accuracy ε Algorithm 2 with Algorithm 3 for solving subproblem computes

$$N_1 = \tilde{O}\left(\left(\frac{L_3 R^4}{\varepsilon}\right)^{\frac{1}{5}}\right)$$

Hessians and

$$N_2 = \tilde{O}\left(\left(\frac{L_3 R^4}{\varepsilon}\right)^{\frac{1}{5}} \log\left(\frac{G+H}{\varepsilon}\right)\right)$$

gradients, where G and H are the uniform upper bounds for the norms of the gradients and Hessians computed at the points generated by the main algorithm.

Problem formulation

Empirical Risk Minimization(ERM)

$$\min_{x \in \mathbb{R}^d} F(x) \triangleq \frac{1}{N} \sum_{i=1}^{N} \ell(x; \xi_i, \eta_i), \tag{1}$$

where $\{\zeta_i = (\xi_i, \eta_i)\}_{i=1}^N$ are training samples, and ℓ is a convex loss function with respect to x.

Problem formulation

Empirical Risk Minimization(ERM)

$$\min_{x \in \mathbb{R}^d} F(x) \triangleq \frac{1}{N} \sum_{i=1}^{N} \ell(x; \xi_i, \eta_i), \tag{1}$$

where $\{\zeta_i = (\xi_i, \eta_i)\}_{i=1}^N$ are training samples, and ℓ is a convex loss function with respect to x.

Convexity

Furthermore, we assume F is L_F-smooth and σ_F-strongly convex, i.e.,

$$\sigma_F I_d \preceq \nabla^2 F(x) \preceq L_F I_d, \tag{2}$$

where I_d is the d-dimensional identity matrix. The condition number of F is denoted as $\kappa_F = L_F / \sigma_F$, and the solution to (1) as x_*.

Distributed setup

Distributed configuration

Data is distributed uniformly among m computing units/nodes/agents such that $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$ and $N = mn$. That is, each machine $j \in \{1, \ldots, m\}$ locally stores n samples $\mathcal{D}_j = \{\xi_i^{(j)}, \eta_i^{(j)}\}_{i=1}^n$. Moreover, there is a central node, that is able to communicate with all the worker nodes.

Distributed ERM

Each agent j has a local empirical risk, denoted as $F_j(x) \triangleq (1/n) \sum_{i=1}^n \ell(x; \xi_i^{(j)}, \eta_i^{(j)})$, where $F_1(x)$ is a central node. Thus,

$$F(x) = \frac{1}{N} \sum_{j=1}^m F_j(x) = \frac{1}{nm} \sum_{j=1}^m \sum_{i=1}^n \ell(x; \xi_i^{(j)}, \eta_i^{(j)}). \tag{3}$$

Statistically preconditioning

Reference function

Following the same algorithmic structure as DANE and SPAG, we define a reference function

$$\phi(x) = \frac{1}{n}\sum_{k=1}^{n} \ell(x, \zeta_k) + \frac{\mu}{2}\|x\|_2^2, \tag{4}$$

where the examples ζ_k are taken from the node which is chosen to be central. It follows that $\phi(x)$ is L_ϕ-smooth, and σ_ϕ-strongly convex.

Statistical similarity

The value of the parameter μ is set to be a upper bound that quantifies how statistically similar the function F_1 is from F, i.e., we assume that with high probability, it holds that

$$\|\nabla^2 F(x) - \nabla^2 F_1(x)\|_2 \leq \mu. \tag{5}$$

Relative smoothness

Relative condition number

$F(x)$ is $L_{F/\phi}$-relative smooth and $\sigma_{F/\phi}$-relative strongly convex with respect to $\phi(x)$, i.e.,

$$\sigma_{F/\phi}\nabla^2\phi(x) \preceq \nabla^2 F(x) \preceq L_{F/\phi}\nabla^2\phi(x), \tag{6}$$

with $L_{F/\phi} = 1$, $\sigma_{F/\phi} = \sigma_F/(\sigma_F + 2\mu)$, and $\kappa_{F/\phi} = L_{F/\phi}/\sigma_{F/\phi}$

Bregman divergence

The Bregman divergence is defined as

$$D_\phi(x, y) \triangleq \phi(x) - \phi(y) - \nabla\phi(y)^\top(x - y). \tag{7}$$

Bregman proximal method

$$\operatorname*{argmin}_{x\in\mathbb{R}^d} \left\{\langle\nabla F(z), x-z\rangle + L_{F/\phi} D_\phi(x, z)\right\}. \tag{8}$$

Algorithm 1 InSPAG $(L_{F/\phi}, \sigma_{F/\phi}, x_0, D)$

1: **Input:** D s.t. $x_* \in B_2(0, D)$, $\hat{D}_\phi^2 = 2L_\phi D^2$, $L_{F/\phi}$, $\sigma_{F/\phi}$, G_0.
2: Set $y_0 = u_0 = x_0 \in B_2(0, D)$, $\alpha_0 \triangleq 0$, $A_0 \triangleq \alpha_0$.
3: **for** $t \geq 0$ **do**
4: **At the central node**
5: Find the smallest integer $i_t \geq 0$ such that

$$D_\phi(x_{t+1}, y_{t+1}) \leq \frac{G_{t+1}\alpha_{t+1}^2}{A_{t+1}^2} D_\phi(u_{t+1}, u_t), \quad (9)$$

where $G_{t+1} = 2^{i_t - 1} G_t$, $A_{t+1} \triangleq A_t + \alpha_{t+1}$, and α_{t+1} is the largest root of

$$A_{t+1}(1 + A_t \sigma_{F/\phi}) = L_{F/\phi} G_{t+1} \alpha_{t+1}^2. \quad (10)$$

6: Send $y_{t+1} \triangleq (\alpha_{t+1} u_t + A_t x_t)/A_{t+1}$ to workers.

Inexact SPAG

At every worker node

Compute $\frac{1}{n}\sum_{i=1}^{n}\nabla\ell\big(y_{t+1},\zeta_i^{(j)}\big)$ and send it to the central node.

At the central node

Compute

$$\nabla F(y_{t+1}) = \frac{1}{nm}\sum_{j=1}^{m}\sum_{i=1}^{n}\nabla\ell\big(y_{t+1},\zeta_i^{(j)}\big).$$

$$\text{Solve } \ u_{t+1} \triangleq \operatorname*{arg\,min}_{x\in B_2(0,D)}{}^{\hat{D}_\phi^2/t} V_t(x), \tag{11}$$

$$\begin{aligned}\text{where } \ V_t(x) \triangleq\ & \alpha_{t+1}\langle\nabla F(y_{t+1}), x - y_{t+1}\rangle + \\ & + (1 + A_t\sigma_{F/\phi})D_\phi(x,u_t) + \\ & + \alpha_{t+1}\sigma_{F/\phi}D_\phi(x,y_{t+1}),\end{aligned} \tag{12}$$

$$\text{Set } \ x_{t+1} \triangleq \frac{\alpha_{t+1}u_{t+1} + A_t x_t}{A_{t+1}}. \tag{13}$$

end for

Convergence Theorem

Convergence Theorem

Assume the function F is σ_F-strongly convex and L_F-smooth, and $\sigma_{F/\phi}$-strongly convex and $L_{F/\phi}$-smooth with respect to a function ϕ, where ϕ is σ_ϕ-strongly convex and L_ϕ-smooth. Moreover, let x_t, $t \geq 0$ be the sequence generated by Algorithm 1. Then, after T iterations it holds that

$$F(x_T) - F(x_*) \leq \frac{\hat{D}_\phi^2}{A_T}(3/2 + \log T), \tag{9}$$

Moreover, the value A_T grows as follows:

$$A_T \geq \max\left\{\frac{T^2}{2L_{F/\phi}\widetilde{G}_T}, \frac{1}{L_{F/\phi}G_1}\exp\left(2T\sqrt{\frac{\sigma_{F/\phi}}{2L_{F/\phi}\widetilde{G}_T}}\right)\right\},$$

where $\widetilde{G}_T^{-1/2} = \frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{\sqrt{G_{t+1}}}$.

Complexity

Total complexity

To obtain an ε accurate minimizer of $F(x)$ total number of subiteration is

$$\tilde{O}\left(\frac{\sqrt{\kappa_F D}}{n^{1/4}}\left(\frac{\|A^\top A\|_2^2 D^2}{\min\{\lambda_1, \lambda_2\} + \mu}\right)^{\frac{1}{5}}\right). \tag{10}$$

Communication complexity

If ϕ is a quadratic function, then $G_t = 1$, and the communication complexity will be $O(\sqrt{\kappa_{F/\phi}})$. In the general case, where ϕ is not quadratic, $G_t \to 1$ linearly with rate $\tilde{O}(\sqrt{\kappa_F})$.

Experiments

Communication complexity

Statistics of the datasets. N is the number of samples, d is the number of features, Feat. is the average number of non-zero features, and Size is the data size in MB.

Dataset	N	d	Feat.	Size
RCV1	20k	47k	74.05	13.7
In-house	710M	3,246k	109.86	650.8k

Experiments

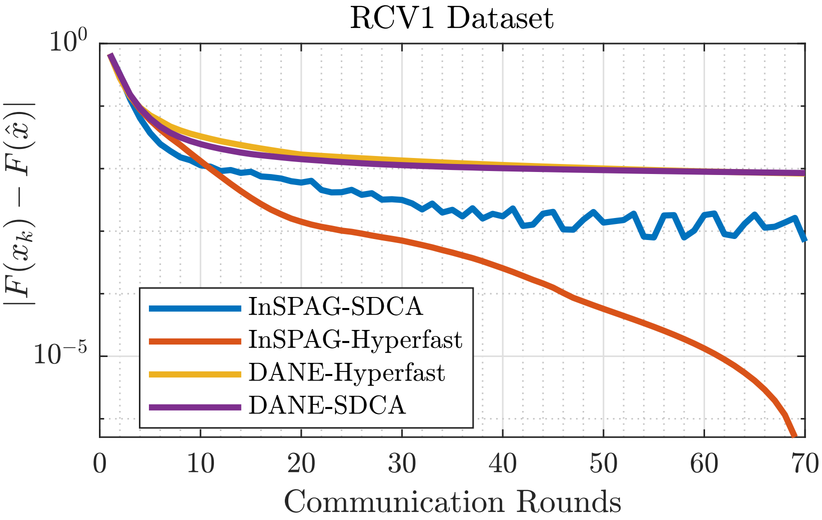

Figure 1: Comparison of the communication rounds for the data set RCV1.

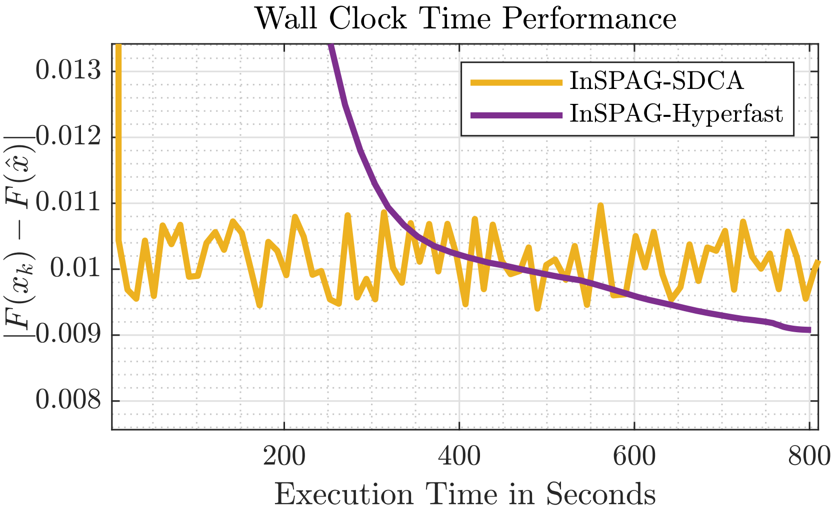

Figure 2: Wall clock time performance of the InSPAG method for the data set RCV1.

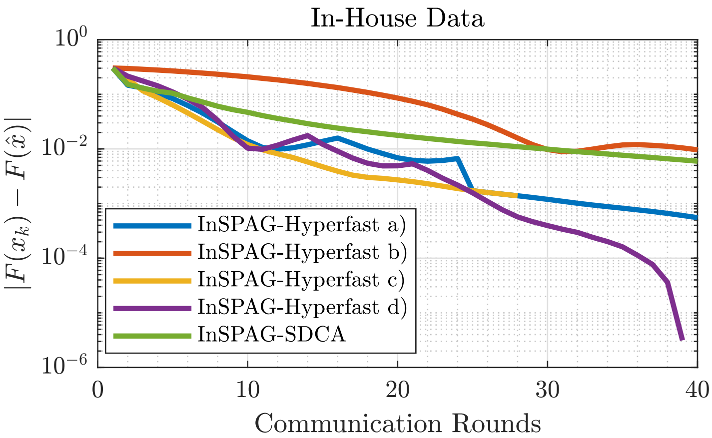

Figure 3: Comparison of the communication rounds for the in house dataset. a) $L_3 = 10$, ADAM learning rate 0.01, $n = 10000$; b) $L_3 = 100$, ADAM learning rate 0.1, $n = 10000$; c) $L_3 = 10$, ADAM learning rate 0.1, $n = 10000$; d) $L_3 = 15$, ADAM learning rate 0.01, $n = 1000$.

Thank you for your attention!