STRANG OPTIMIZATION SPLITTING

Daniil Merkulov & Ivan Oseledets
Skolkovo Institute of Science and Technology
Center for Computational and Data-Intensive Science and Engineering
Bolshoy Boulevard 30, bld. 1, Moscow, Russia 121205
{daniil.merkulov, i.oseledets}@skoltech.ru

ABSTRACT

We present different view on stochastic optimization

1 INTRODUCTION

A lot of practical problems arising in machine learning require minimization of a finite sample average which can be written in the form

$$f(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta) \to \min_{\theta \in \mathbb{R}^p}, \tag{1}$$

where the sum goes over the *minibatches* of the original dataset. Vanilla stochastic gradient descent (SGD) method (Robbins & Monro (1951))has consists in sequential step in the direction of the gradient of $f_i(\theta)$, where i is to be chosen randomly from 1 to n without replacement.

$$\theta_{k+1} = \theta_k - h_k \nabla f_i.$$

Gradient descent method can be considered as an Euler discretization of the ordinary differential equation (ODE) of the form of the gradient flow

$$\frac{d\theta}{dt} = -\nabla f(\theta). \tag{2}$$

In continuous time, SGD if often analyzed by introducing a noise into the right-hand side of equation 2. However, for real dataset the distribution of the noise obtained by replacing the full gradient by its minibatch variant is not known and can be different for different problems. Instead, we propose a new view on the SGD as a *first-order splitting scheme* for equation 2, thus shedding a new light on SGD-type algorithms. Moreover, we interpret stochastic average gradient (SAG)Schmidt et al. (2017) approach as the *splitting scheme with rebalancing*. This representation allows to use more efficient splitting schemes for the approximation of the full gradient flow. We show, that second-order Marchuk/Strang splitting scheme (Marchuk (1968), Strang (1968)) provides faster convergence of the SAG method, which we call *SAG2* method. In implementation, the second-order scheme consists in a sequential processing of minibatches first from 1 to m, and then from m to 1, i.e. a single iteration of SAG2 has the same cost as two iterations of SAG.

The convergence is faster, but the number of iterations is also greater. That's not the point, however, we need to remember about SAG method, so, let's leave as is for now.

Contributions

- We show, that vanilla SGD could be considered as a splitting scheme for a full gradient flow.
- We demonstrate the connection between rebalancing splitting and stochastic average gradient method.
- We propose new optimization method, SAG2 based on second order splitting scheme and show that it gives better convergence, than the standard SAG method.

2 Related work

Lets write it – still seems it helps

3 SGD as a splitting scheme

We want to establish the connection between splitting scheme for ODE and stochastic optimization. In this section we firstly consider simple ODE, where we can apply splitting idea and corresponding minimization problem.

3.1 Splitting schemes for ODEs

The best example to start from is simple ODE with right-hand-side, consisting of two summands:

$$\frac{d\theta}{dt} = -\frac{1}{2}\left(g_1(\theta) + g_2(\theta)\right) \tag{3}$$

Suppose, we want to find the solution $\theta(h)$ of equation 3 via integrating it on the small timestep h. The first order splitting scheme defined by solving first:

$$\frac{d\theta}{dt} = -\frac{1}{2}g_1(\theta)$$

with exact solution $\theta_1(h)$ at the moment h , followed by

$$\frac{d\theta}{dt} = -\frac{1}{2}g_2(\theta)$$

with exact solution $\theta_2(h)$ at the moment h. Thus, the first order approximation could be written as a combinations of both solutions:

$$\theta^I(h) = \theta_2(h) \circ \theta_1(h) \circ \theta_0,$$

while the second order scheme takes 3 substeps:

$$\theta^{II}(h) = \theta_1\left(\frac{h}{2}\right) \circ \theta_2(h) \circ \theta_1\left(\frac{h}{2}\right) \circ \theta_0$$

Order of scheme defines the degree of polynomial of h, up to which the true solution and approximation are coincide. The local error of both schemes could be obtained by Baker - Campbell - Hausdorff formula (Baker (1901), Campbell (1896), Hausdorff (1906))

$$\theta^I(h) - \theta(h) = \frac{h^2}{2}\left[g_1, g_2\right]\theta_0 + o(h^3), \tag{4}$$

$$\theta^{II}(h) - \theta(h) = h^3\left(\frac{1}{12}[g_2,[g_2,g_1]] - \frac{1}{24}[g_1,[g_1,g_2]]\right)\theta_0 + O\left(h^4\right) \tag{5}$$

where $[g_1, g_2] = \dfrac{dg_1}{d\theta}g_2 - \dfrac{dg_2}{d\theta}g_1$ stands for commutator of the vector fields g_1 and g_2. The $\theta_0 = \theta(0)$ for initial condition of original ODE.

Note, that the basic idea of splitting could be also applied, when the number of terms in the right-hand side of ODE is greater, than two. In this case splitting scheme will take the following form:

$$\theta^I(h) = \theta_m(h) \circ \theta_{m-1} \circ \ldots \circ \theta_2(h) \circ \theta_1(h) \circ \theta_0 \tag{6}$$

$$\theta^{II}(h) = \theta_1\left(\frac{h}{2}\right) \circ \theta_2\left(\frac{h}{2}\right) \circ \ldots \theta_m(h) \ldots \theta_2\left(\frac{h}{2}\right) \circ \theta_1\left(\frac{h}{2}\right) \circ \theta_0 \tag{7}$$

3.2 SGD as approximation for the Gradient Flow equation

Now we consider classical SGD method as a splitting scheme for the full gradient descent (Cauchy (1847)). Suppose, we have the simple finite sum minimization problem:

$$\min_{\theta \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} f_i(\theta) = \min_{\theta \in \mathbb{R}^n} \frac{1}{2} \left(f_1(\theta) + f_2(\theta) \right)$$

Let us denote by $g_i^k = \nabla f_i(\theta_k)$, than, the vanilla gradient descent will be written as

$$\theta_{k+1} = \theta_k - h \cdot \frac{1}{2} \left(g_1^k(\theta) + g_2^k(\theta) \right),$$

while SGD version will take steps iteratively over minibatch gradient directions:

$$\begin{aligned} \theta_{k+1} &= \theta_k - h \cdot g_1^k(\theta) \\ \theta_{k+2} &= \theta_{k+1} - h \cdot g_2^{k+1}(\theta) \end{aligned}$$

These two iterations forms an epoch in SGD approach. Each of the substeps can be considered as a forward Euler method for the discretization of the ODE for a timestep h

$$\begin{aligned} \frac{d\theta^I}{dt} &= -g_1(\theta), \quad \theta^I(0) = \theta_k, \\ \frac{d\theta^{II}}{dt} &= -g_2(\theta), \quad \theta^{II}(0) = \theta^I(h), \end{aligned}$$

therefore the final result for a sufficiently small h approximates the gradient flow for at time $t + h$. The vanilla gradient descent, however, is only approximating the gradient flow at time $t + h/2$. For larger number of minibatches, rather then the GD flow. One can notice, that in SGD we use a very simple time integration inside the substep. In some cases, we can integrate the subproblem exactly, without using forward Euler scheme. Generalized linear models are among such problems, but we will first study the linear least squares case in more details, since in this case we can obtain non-trivial error bounds.

3.3 Splitting approximation for the Gradient Flow equation

It is interesting to look how the pure splitting scheme corresponds to the SGD approach. For this purpose we consider illustrative example of Gradient Flow equation 8, where the right-hand side of ODE is just the sum of operators acting on θ, which allows us to apply splitting scheme approximation directly.

$$\frac{d\theta}{dt} = -\frac{1}{2} \sum_{i=1}^{2} \nabla f_i(\theta) = -\frac{1}{2} \nabla f_1(\theta) - \frac{1}{2} \nabla f_2(\theta) \tag{8}$$

In order to establish the connection between splitting scheme and SGD we use the Euler discretization below:

First splitting step: $\frac{d\theta}{dt} = -\frac{1}{2}\nabla f_1(\theta) \rightarrow$ Euler discretization $\rightarrow \quad \tilde{\theta}_I = \theta_0 - \frac{h}{2}\nabla f_1(\theta_0)$

Second splitting step: $\frac{d\theta}{dt} = -\frac{1}{2}\nabla f_2(\theta) \rightarrow$ Euler discretization $\rightarrow \quad \theta_I = \tilde{\theta}_I - \frac{h}{2}\nabla f_2(\tilde{\theta}_I)$

SGD epoch	First order splitting
$\tilde{\theta}_{SGD} = \theta_0 - h\nabla f_1(\theta_0)$	$\tilde{\theta}_I = \theta_0 - \frac{h}{2}\nabla f_1(\theta_0)$
$\theta_{SGD} = \tilde{\theta}_{SGD} - h\nabla f_2(\tilde{\theta}_{SGD})$	$\theta_I = \tilde{\theta}_I - \frac{h}{2}\nabla f_2(\tilde{\theta}_I)$

Moreover, we can conclude, that *one epoch of SGD is just the splitting scheme for the discretized Gradient Flow ODE with $2 \cdot h$ step size ($m \cdot h$ in case of m batches)*

This idea gives additional intuition on the method. Both approaches are solving local problems through the Euler discretization. Given an information about the Euler scheme limitation, why not solve these local problems more accurate?

4 Linear least squares

4.1 Problem

Let $f_i(\theta) = \|x_i^\top \theta - y_i\|^2$, then problem equation 1 is the linear least squares problem, which can be written as

$$f(\theta) = \frac{1}{n}\|X\theta - y\|_2^2 = \frac{1}{n}\sum_{i=1}^{s}\|X_i\theta - y_i\|_2^2 \to \min_{\theta \in \mathbb{R}^p}, \tag{9}$$

where X in an $n \times p$ matrix, and y is a vector of length p and the second part of the equation stands for s mini-batches with size b regrouping ($b \cdot s = n$): $X_i \in \mathbb{R}^{b \times p}, y_i \in \mathbb{R}^b$

$$\nabla_\theta f(\theta) = \nabla f(\theta) = \frac{1}{n}\sum_{i=1}^{s} X_i^\top (X_i\theta - y_i) \tag{10}$$

The gradient flow equation will be written as follows:

$$\frac{d\theta}{dt} = -\frac{1}{n}\sum_{i=1}^{s} X_i^\top (X_i\theta - y_i) \tag{11}$$

4.2 Exact solution of the local problem

On each splitting approximation step we need to solve the local problem:

$$\frac{d\theta}{dt} = -\frac{1}{n} X_i^\top (X_i\theta - y_i) \tag{12}$$

Theorem 1. *For any matrix $X_i \in \mathbb{R}^{b \times p}$, any vector of right-hand side $y_i \in \mathbb{R}^b$ and initial vector of parameters θ_0, there is a solution of the ODE in 12, given by formula:*

$$\theta(h) = Q_i e^{-\frac{1}{n}R_i R_i^\top h}\left(Q_i^\top \theta_0 - R_i^{-\top} y_i\right) + Q_i R_i^{-\top} y_i + (I - Q_i Q_i^\top)\theta_0,$$

where $Q_i \in \mathbb{R}^{p \times b}$ and $R_i \in \mathbb{R}^{b \times b}$ stands for the QR decomposition of the matrix $X_i^\top$, $X_i^\top = Q_i R_i$.

Proof. Given $X_i^\top = Q_i R_i$, we have $(I - Q_i Q_i^\top)X_i^\top = 0$. Note, that Q_i is left unitary matrix, i.e. $Q_i^\top Q_i = I$.

$$\begin{aligned}
\frac{d\theta}{dt} &= -\frac{1}{n} X_i^\top (X_i\theta - y_i) \\
(I - Q_i Q_i^\top)\frac{d\theta}{dt} &= 0 \\
\frac{d\theta}{dt} &= Q_i \frac{d(Q_i^\top \theta)}{dt} \qquad Q_i^\top \theta = \eta_i \\
\frac{d\theta}{dt} &= Q_i \frac{d\eta_i}{dt} \quad \text{integrate from 0 to } h \\
\theta(h) &= Q_i\left(\eta_i(h) - \eta_i(0)\right) + \theta_0
\end{aligned} \tag{13}$$

On the other hand:

$$\begin{aligned}
\frac{d\eta_i}{dt} &= Q_i^\top \frac{d\theta}{dt} = -\frac{1}{n} Q_i^\top X_i^\top (X_i\theta - y_i) = -\frac{1}{n} Q_i^\top Q_i R_i (R_i^\top Q_i^\top \theta - y_i) = \\
&= -\frac{1}{n}\left(R_i R_i^\top \eta_i - R_i y_i\right)
\end{aligned} \tag{14}$$

Consider the moment of time $t = \infty$. $\dfrac{d\eta}{dt} = 0$, since $\exists \theta^*, Q_i^\top \theta^* = \eta_i^*$. Also consider 14:

Need to clarify this assumption

$$\frac{d\eta_i}{dt} = 0 = -\frac{1}{n}\left(R_i R_i^\top \eta_i^* - R_i y_i\right) \quad \rightarrow \quad R_i y_i = R_i R_i^\top \eta_i^* \tag{15}$$

Now we look at the 14 with the replacement, given in 15:

$$\begin{aligned} \frac{d\eta_i}{dt} &= -\frac{1}{n}\left(R_i R_i^\top \eta_i - R_i R_i^\top \eta_i^*\right) \\ \frac{d\eta_i}{dt} &= -\frac{1}{n} R_i R_i^\top \left(\eta_i - \eta_i^*\right) \qquad \text{integrate from 0 to } h \\ \eta_i(h) - \eta_i^* &= e^{-\frac{1}{n} R_i R_i^\top h}(\eta_i(0) - \eta_i^*) \qquad \eta_i^* = R_i^{-\top} y_i, \eta_i(0) = Q_i^\top \theta_0 \\ \eta_i(h) &= e^{-\frac{1}{n} R_i R_i^\top h}(Q_i^\top \theta_0 - R_i^{-\top} y_i) + R_i^{-\top} y_i \end{aligned}$$

Using 13 we obtain the target formula

$$\theta(h) = Q_i \left(e^{-\frac{1}{n} R_i R_i^\top h}(Q_i^\top \theta_0 - R_i^{-\top} y_i) + R_i^{-\top} y_i - Q_i^\top \theta_0 \right) + \theta_0$$

□

In case of the linear right-hand side of an ODE it is easy to solve it analytically. Now let see how the splitting approximation itself depends on the step size h.

$$\theta^{GD}(h) = e^{-Ah}\theta,$$

and splitting gives

$$\theta^{SGD}(h) = e^{-A_1 h} e^{-A_2 h}\theta.$$

The error is bounded as

$$\|\theta^{GD}(h) - \theta^{SGD}(h)\| \leq \|E_1(h)\| \|\theta\|,$$

where

$$E_1(t) = e^{At} - e^{A_1 t} e^{A_2 t}. \tag{16}$$

We need to bound the norm of the matrix $E_1(t)$ for all h, not only for small ones, i.e. we need global estimates. Such kind of estimates were obtained in Sheng (1994) and have the form

$$\|E(t)\| \leq \frac{t^2}{2} \|[A_1, A_2]\| \max\{e^{t\mu(A_1+A_2}), e^{t(\mu(A_1)+\mu(A_2))}\}, \tag{17}$$

where $\mu(Z)$ is the largest eigenvalue of the matrix $\frac{Z+Z^*}{2}$, but in our case all matrices are symmetric, thus these are largest eigenvalues of the matrix. The estimate equation 17 and its generalization to a larger number of summands is not very useful for us, since we will have matrices X_i that have fewer rows, than column, i.e. matrices A_i will have zero eigenvalues, thus the the maximum term will be equal to 1, and the upper bound will grow quadratically with h. In reality, the behaviour is very different, see Figure 1. In this example, we took $N = p = 2$, batch size 1. It can be seen, that the true error reaches a plateu, whereas the upper bound is growing quadratically with t. We will now prove a better upper bound, that takes into account possible zero eigenvalues of the matrices A_1 and A_2.

4.3 Upper bound on the global splitting error

This section is TBD!

Suppose, that we have only two batches, and the problem equation **??** is consistent, i.e. there exists an exact solution θ_* such as $X\theta_* = y$. The GD flow has the form

$$\frac{d\theta}{dt} = -X^\top(X\theta - y) = -X^\top X(\theta - \theta_*) = -(X_1^\top X_1 + X_2^\top X_2)(\theta - \theta_*), \tag{18}$$

i.e. the splitting scheme corresponds to a linear operator splitting

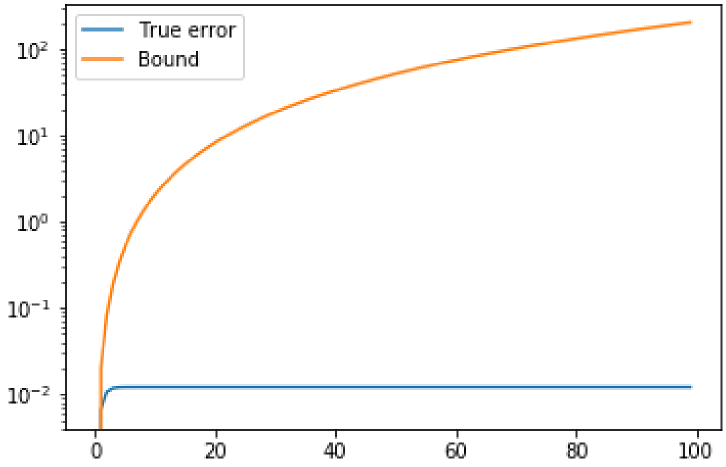


Figure 1: The estimate from equation 17 and the true error for a model example

$$A = A_1 + A_2, \quad A = -X^\top X, \quad A_i = -X_i^\top X_i, \quad i = 1, 2.$$

Both A_1 and A_2 are symmetric non-negative definite matrices. Without loss of generality, we can assume that $\theta_* = 0$,

Suppose that the rank of A is r_1 and the rank of A_2 is r_2. Then, we can write them as

$$A_i = Q_i B_i Q_i^*,$$

where Q_i is an $N \times r_i$ matrix with orthonormal columns. The following Lemma gives the representation of the matrix exponents of such matrices.

Lemma 1. *Let $A = QBQ^*$, where Q is an $N \times r$ matrix with orthonormal columns, and B is an $r \times r$ matrix. Then,*

$$e^{tA} = (I - QQ^*) + Qe^{tB}Q^*. \tag{19}$$

To prove equation 19 we note that

$$e^{tA} = \sum_{k=0}^{\infty} \frac{t^k A^k}{k!} = \sum_{k=0}^{\infty} \frac{t^k QB^kQ^*}{k!} = I - QQ^* + QQ^* + Q\sum_{k=1}^{\infty} \frac{t^k B^k}{k!} Q^* = (I - QQ^*) + Qe^{tB}Q^*.$$

To derive the upper bound, we will follow the ideas of Sheng (1994). The error equation 16 satisfies the differential equation

$$E'(t) = (A_1 + A_2)E_1(t) + [A_2, e^{A_1 t}]e^{tA_2},$$

with initial condition $E(0) = 0$. Thus,

$$E(t) = \int_0^t e^{(t-\tau)(A_1+A_2)}[A_2, e^{\tau A_1}]e^{\tau A_2} d\tau. \tag{20}$$

The commutator

$$S(\tau) = [A_2, e^{\tau A_1}]$$

satisfies the differential equation

$$S'(\tau) = A_1 S(\tau) + [A_2, A_1]e^{\tau A_1},$$

with initial condition $S(0) = 0$, thus

$$S(\tau) = \int_0^\tau e^{(\tau-s)A_1}[A_2, A_1]e^{sA_1} ds. \tag{21}$$

Putting equation 20 and equation 21 together, we get

$$E(t) = \int_0^t d\tau \int_0^\tau e^{(t-\tau)(A_1+A_2)} e^{(\tau-s)A_1} [A_2, A_1] e^{sA_1} e^{\tau A_2} ds. \tag{22}$$

Now we need to work on the inner term. Using Lemma 1 we have

$$S(\tau) = (I - Q_1 Q_1^* + Q_1 e^{(\tau-s)B_1} Q_1^*)[A_2, A_1] \left(I - Q_1 Q_1^* + Q_1 e^{sB_1} Q_1^*\right). \tag{23}$$

Some simplifications are possible. Indeed,

$$(I - Q_1 Q_1^*)[A_2, A_1](I - Q_1 Q_1^*) = 0,$$

since $(I - Q_1 Q_1^*)A_1 = 0$ and $A_1(I - Q_1 Q_1^*) = 0$, and there will be 3 terms in the expansion, which are estimated in the same way.

We have

$$\|E(t)\| \leq E_1 + E_2 + E_3,$$

We will need to compute the integral

$$\int_0^t d\tau \int_0^\tau e^{\beta\tau} e^{\gamma s} ds = \frac{\frac{1-e^{\beta t}}{\beta} + \frac{e^{(\beta+\gamma)t}-1}{\beta+\gamma}}{\gamma}$$

For the terms in the estimate of $E(t)$ we have the following. For E_1, we have

$$\beta = -\mu(A_1 + A_2) + \mu(A_2), \quad \gamma = \mu(B_1),$$

for E_2 we have

$$\beta = -\mu(A_1 + A_2) + \mu(A_2) + \mu(B_1), \gamma = -\mu(B_1).$$

After simplifications and assuming $\mu(A_2) = 0$, we get

$$\|E_1 + E_2\| \leq \frac{\|[A_1, A_2]\|}{\mu(A_1 + A_2)\mu(B_1)}.$$

The estimate for E_3 is the same as for the classical approach, but with $\mu(A_1)$ replaced by $\mu(B_1)$.

5 Binary logistic regression

5.1 Problem

In this classification task then problem equation 1 takes the following form:

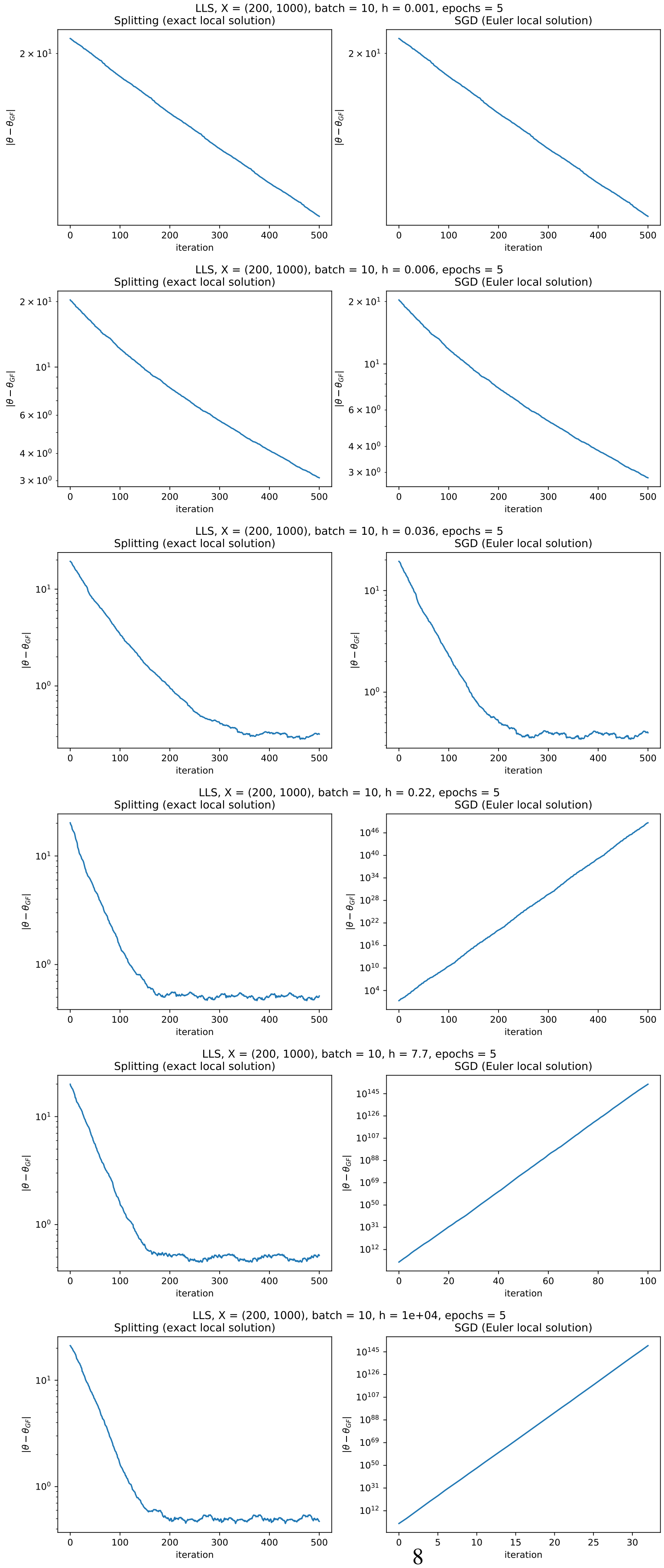
$$f(\theta) = -\frac{1}{n}\left(y_i \ln h_\theta(x_i) + (1 - y_i)\ln(1 - h_\theta(x_i))\right) \to \min_{\theta \in \mathbb{R}^p}, \tag{24}$$

where $h_\theta(x_i) = \dfrac{1}{1 + e^{-\theta^\top x_i}}$ is the hypothesis function with given parameter θ from the object x_i, $y_i \in \{0, 1\}$ stands for the label of the object class.

$$\nabla_\theta f(\theta) = \nabla f(\theta) = \frac{1}{n}\sum_{i=1}^n x_i (h_\theta(x_i) - y_i) \tag{25}$$

The gradient flow equation will be written as follows:

$$\frac{d\theta}{dt} = -\frac{1}{n}\sum_{i=1}^n x_i (h_\theta(x_i) - y_i) \tag{26}$$



LLS, X = (200, 1000), batch = 10, h = 0.001, epochs = 5

Splitting (exact local solution)

SGD (Euler local solution)

LLS, X = (200, 1000), batch = 10, h = 0.006, epochs = 5

Splitting (exact local solution)

SGD (Euler local solution)

LLS, X = (200, 1000), batch = 10, h = 0.036, epochs = 5

Splitting (exact local solution)

SGD (Euler local solution)

LLS, X = (200, 1000), batch = 10, h = 0.22, epochs = 5

Splitting (exact local solution)

SGD (Euler local solution)

LLS, X = (200, 1000), batch = 10, h = 7.7, epochs = 5

Splitting (exact local solution)

SGD (Euler local solution)

LLS, X = (200, 1000), batch = 10, h = 1e+04, epochs = 5

Splitting (exact local solution)

SGD (Euler local solution)

Figure 2: Linear Least Squares, $X \in \mathbb{R}^{200\times 1000}, b = 10$

6 Results

6.1 Precise splitting approximation is way more robust to the stepsize, than SGD

6.1.1 Linear Least Squares

6.1.2 Binary Logistic Regression

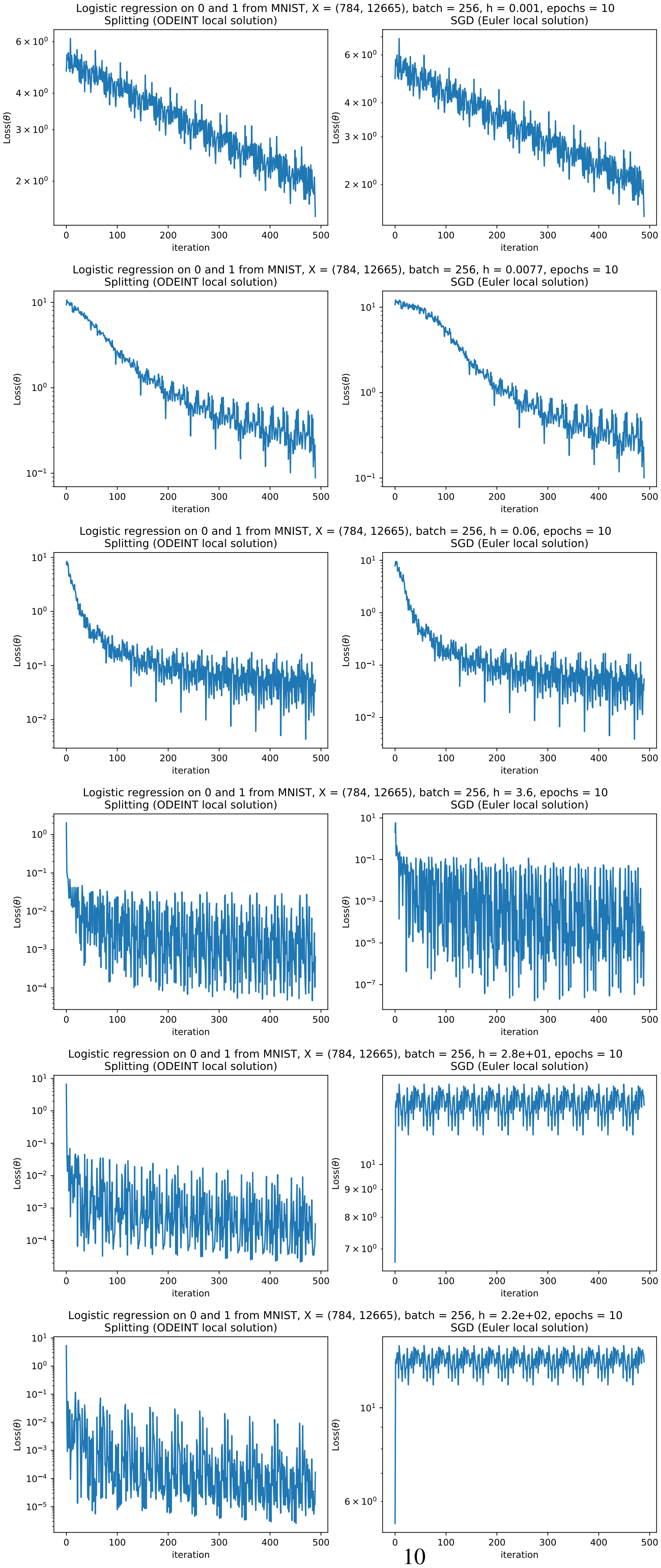


Figure 3: Binary logistic regression on 0 and 1 from MNIST dataset

REFERENCES

Henry Frederick Baker. Further applications of metrix notation to integration problems. *Proceedings of the London Mathematical Society*, 1(1):347–360, 1901.

JE Campbell. On a law of combination of operators bearing on the theory of continuous transformation groups. *Proceedings of the London Mathematical Society*, 1(1):381–390, 1896.

Augustin Cauchy. M'ethode g'en'erale pour la r'esolution des systemes d''equations simultan'ees. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

Felix Hausdorff. Die symbolische exponentialformel in der gruppentheorie. *Ber. Verh. Kgl. SÃ chs. Ges. Wiss. Leipzig., Math.-phys. Kl.*, 58:19–48, 1906.

Gurij Ivanovich Marchuk. Some application of splitting-up methods to the solution of mathematical physics problems. *Aplikace matematiky*, 13(2):103–132, 1968.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Qin Sheng. Global error estimates for exponential splitting. *IMA Journal of Numerical Analysis*, 14 (1):27–56, 1994.

Gilbert Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968.