Stochastic Gradient Descent with Momentum and Line Searches

David Nardi
MSc student in Artificial Intelligence, Univeristy of Florence

6th February 2024

Abstract

In recent years, tailored line search approaches have proposed to define the step-size, or learning rate, in SGD-type algorithms for finite-sum problems. In particular, a stochastic extension of standard Armijo line search has been proposed in `bib1`. The development of this kind of techniques is relevant, because it shall allow to enforce a stronger converging behaviour (due to the Armijo condition), similar to that of standard GD, within SGD methods that are commonly employed with large scale training problems.

However, the stochastic line search is not immediately employable when the momentum term is part of the update equation, as the search direction might not be a descent direction (which is a necessary condition for the Armijo condition). This problem is addressed in `bib2`, where a strategy is proposed to guarantee the descent property with momentum.

Contents

1 Introduction

This report summarizes the analysis performed in order to investigate the behaviour of the algorithms retrieved from the scientific literature. The optimization problem that we aim to solve is that of the Logistic Regression with ℓ_2-regularization term added.

The implemented algorithms are

- Mini-batch Gradient Descent with fixed and decreasing step-size, algorithm 1 on page 6;
- Mini-batch Gradient Descent with Armijo-type line search, algorithm 2 on page 6;
- Mini-batch Gradient Descent with fixed step-size and momentum, algorithm 3 on page 7;
- Mini-batch Gradient Descent with with Armijo-type line search and momentum correction and restart, algorithms 4 and 5 on page 8.

After that, the efficiency of the algorithms is tested on different datasets.

1.1 Classification task

1.2 Optimization problem

Given the following dataset

$$\mathcal{D} = \{(x^{(i)}, y^{(i)}) \mid x^{(i)} \in \mathcal{X}, y^{(i)} \in \mathcal{Y}, i = 1, 2, \dots, N\}$$

the general Machine Learning optimization problem for the supervised learning subset is formulated as follows

$$\min_{w \in \mathbb{R}^p} f(w) = L(w) + \lambda\Omega(w) = \begin{cases} L(w) = \frac{1}{N}\sum_{i=1}^{N} \ell_i(w) \\ \Omega_{\ell_2} = \frac{1}{2}\|w\|_2^2 \end{cases}$$

where $L(w)$ is called *loss function*, and it is a finite-sum, and $\Omega(w)$ it's the *regularization term* with its coefficient λ. There are three regularization possible choices, the ℓ_2 regularization was chosen for this problem.

The task performed is the binary classification, using the Logistic Regression model. Every machine learning model has its own loss function, the logistic regression uses the *log-loss* (figure 1a on page 4), for one single dataset sample $\ell_i = \log\bigl(1 + \exp(-y^{(i)} w^T x^{(i)})\bigr)$, follows the resulting optimization problem

$$\min f(w) = \frac{1}{N}\sum_{i=1}^{N} \log\bigl(1 + \exp(-y^{(i)} w^T x^{(i)})\bigr) + \lambda\frac{1}{2}\|w\|^2 \tag{1}$$

where $i = 1, \dots, N$ are the dataset samples, $\mathcal{Y} = \{-1, 1\}$ is the set of the possible values for the response variable, corresponding to the negative or positive class. $\mathcal{X} \subseteq \mathbb{R}^{(p+1)}$ are dataset examples, $p + 1$ means that there are p features and the intercept. The $1/N$ term isn't always used, we choose to use that term for scaling issues.

We defined the matrix associated to the dataset and the model weights as follow

$$X^T = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \dots & x_p^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \dots & x_p^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(N)} & x_2^{(N)} & \dots & x_p^{(N)} \end{pmatrix} \in \mathbb{R}^{N \times (p+1)} \qquad x^{(i)} = \begin{pmatrix} 1 \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_p^{(i)} \end{pmatrix} \qquad w = \begin{pmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_p \end{pmatrix}$$

the constant column is added for the intercept. We can see that every $x^{(i)}$ is a column vector

$$\nabla f(w) = \frac{1}{N} X^T r + \lambda w, \quad r_i = -y^{(i)} \sigma(-y^{(i)} w^T x^{(i)})$$

$$\nabla^2 f(w) = \frac{1}{N} X D X^T + \lambda I, \quad d_{ii} = \sigma(y^{(i)} w^T x^{(i)}) \sigma(-y^{(i)} w^T x^{(i)})$$

$r \in \mathbb{R}^N$, $D \in \mathbb{R}^{N \times N}$

Proposition 1. *Problem* (1) *admits a unique optimal solution.*

Proof. The objective function $f(w)\colon \mathbb{R}^{(p+1)} \to \mathbb{R}$, $f(w) \in C^2(\mathbb{R}^{(p+1)})$

First thing we prove the existence of an optimal solution, that is the global minimum. The derivatives

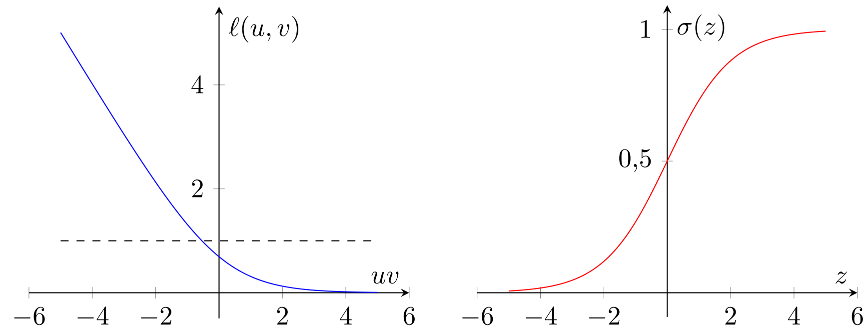
$$\nabla f(w) = \tfrac{1}{N} X^T r + \lambda w, \quad r_i = -y^{(i)} \sigma(-y^{(i)} w^T x^{(i)}),\ r \in \mathbb{R}^N$$
$$\nabla f_i(w) = \tfrac{1}{N} x^{(i)} r_i + \lambda w$$
$$\nabla^2 f(w) = \tfrac{1}{N} X D X^T + \lambda I_{(p+1)}, \quad d_{ii} = \sigma(y^{(i)} w^T x^{(i)}) \sigma(-y^{(i)} w^T x^{(i)}), D \in \mathbb{R}^{N \times N}$$

where $\sigma(z)$ is the sigmoid function, see figure 1b on the next page the second order derivative is positive-defined, this because of the square of the euclidean norm, this implies that the objective function is coercive. By a corollary of the Weirstrass theorem we prove that the function admits global minimum.

$\square$

$$y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{pmatrix} \in \{-1, 1\}$$

$$f(w) = \textstyle\sum_{i=1}^N \log\bigl(1 + \exp(-y^{(i)} w^T x^{(i)})\bigr) \quad f_i(w) = \log\bigl(1 + \exp(-y^{(i)} w^T x^{(i)})\bigr)$$



(a) *Log-loss. If $uv \gg 0$ then the example is labelled correctly; if $uv \ll 0$ then the label is the wrong one.*

(b) *Sigmoid function*

Figure 1: Log-loss and sigmoid function plots

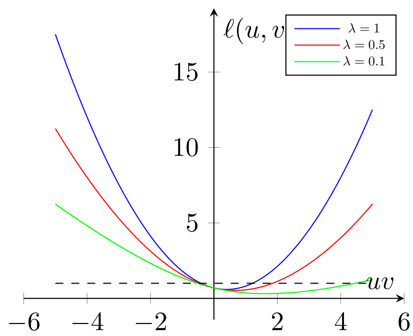


Figure 2

- $uv \gg 0$: the example is labelled correctly
- $uv \ll 0$: the class assigned to the example is the wrong one
- the hessian matrix is positive defined $\forall w$, this means that the objective function, which is quadratic, is coercive and for the continuity that function admits global minimum, so $f(w)$ has finite inferior limit
- the hessian matrix being positive defined implies also that the objective function is strictly convex (on the other hand the loss function is just convex, due to its hessian matrix being positive semi-defined), this implies that if the global minimum exists, that solution is unique
- a global minimum is a point that satisfy $\nabla f(w^*) = 0$, which is a sufficient condition implied by the convexity of the problem, see figure 1a on the preceding page
- the ℓ_2 regularization implies that the objective function is strongly convex, this speeds up the convergence
- further more we can assume that $\nabla f(w)$ is Lipschitz-continuous with constant L

2 Mini-batch gradient descent variants

2.1 Fixed step-size

Algorithm 1: Mini-batch Gradient Descent with fixed or decreasing step-size

```
given $w^0 \in \mathbb{R}^n$, $k = 0$ e $\{\alpha_k\} \mid \alpha_k = \alpha \vee \alpha_k = \frac{\alpha_0}{k+1}$
while ($\|\nabla f(w^k)\| > \varepsilon(1 + |f(w)|)$)
 shuffle $\{1, \dots, N\}$ and split $B_1, \dots, B_{N/M}$ such that $1 < |B_t| = M \ll N$
 set $y_0 = w^k$
 for $t = 1, \dots, N/M$
  get mini-batch indices from $B_t$
  approximate true gradient $\nabla f_{i_t}(w) = \frac{1}{M} \sum_{j \in B_t} \nabla f_j(y_{t-1})$
  compute direction $d_t = -\nabla f_{i_t}(y_{t-1})$
  make a (internal) step $y_t = y_{t-1} + \alpha_k d_t$
 end for
 update weights $w^{k+1} = y_{N/M}$
 epoch ends $k = k + 1$
end while
```

2.2 Stochastic line search

Algorithm 2: Mini-batch Gradient Descent with Armijo line search

```
given $w^0 \in \mathbb{R}^{p+1}$, $\gamma \in (0,1)$, $\delta \in (0,1)$, $\alpha_0 \in \mathbb{R}^+$
$k = 0$
while ($\|\nabla f(w^k)\| > \varepsilon(1 + |f(w)|)$)
 shuffle $\{1, \dots, N\}$ and split $B_1, \dots, B_{N/M}$ such that $1 < |B_t| = M \ll N$
 set $y_0 = w^k$
 for $t = 1, \dots, N/M$
  get mini-batch indices $i_t$ from $B_t$
  approximate true gradient $\nabla f_{i_t}(w) = \frac{1}{M} \sum_{j \in B_t} \nabla f_j(y_{t-1})$
  compute direction $d_t = -\nabla f_{i_t}(y_{t-1})$
  $\alpha = $ reset(), $q = 0$
  compute potential next step $y_t = y_{t-1} + \alpha d_t$
  while ($f_{i_t}(y_t) > f_{i_t}(y_{t-1}) + \gamma\alpha\nabla f_{i_t}(y_{t-1})^T d_t$)
   reduce step-size $\alpha = \delta\alpha$
   rejections counter $q = q + 1$
  end while
  set optimal mini-batch step-size $\alpha_t = \alpha$
  make a (internal) step $y_t = y_{t-1} + \alpha_t d_t$
 end for
 update weights $w^{k+1} = y_{N/M}$
 epoch ends $k = k + 1$
end while
```

2.3 Fixed momentum term

Algorithm 3: Mini-batch Gradient Descent with fixed Momentum term and fixed step-size

```
given $w^0 \in \mathbb{R}^{p+1}$, $\{\alpha_k\} = \alpha$, $\{\beta_k\} = \beta \in (0,1)$
$k = 0$
while ($\|\nabla f(w^k)\| > \varepsilon(1 + |f(w)|)$)
 shuffle $\{1, \ldots, N\}$ and split $B_1, \ldots, B_{N/M}$ such that $1 < |B_t| = M \ll N$
 set $y_0 = w^k$, $d_0 = 0$
 for $t = 1, \ldots, N/M$
  get mini-batch indices $i_t$ from $B_t$
  approximate true gradient $\nabla f_{i_t}(w) = \frac{1}{M}\sum_{j \in B_t} \nabla f_j(y_{t-1})$
  compute direction $d_t = -\big((1-\beta)\nabla f_{i_t}(y_{t-1}) + \beta d_{t-1}\big)$
  make a (internal) step $y_t = y_{t-1} + \alpha_k d_t$
 end for
 update weights $w^{k+1} = y_{N/M}$
 epoch ends $k = k + 1$
end while
```

Algorithm 4: Mini-batch Gradient Descent with Armijo line search for step-size and Momentum correction

```
given $w^0 \in \mathbb{R}^{p+1}$, $\gamma \in (0,1)$, $\delta_a \in (0,1)$, $\alpha_0 \in \mathbb{R}^+$, $\delta_m \in (0,1)$, $\beta_0 \in (0,1)$
$k = 0$
while ($\|\nabla f(w^k)\| > \varepsilon(1 + |f(w)|)$)
 shuffle $\{1, \ldots, N\}$ and split $B_1, \ldots, B_{N/M}$ such that $1 < |B_t| = M \ll N$
 set $y_0 = w^k$, $d_0 = 0$
 for $t = 1, \ldots, N/M$
  get mini-batch indices $i_t$ from $B_t$
  approximate true gradient $\nabla f_{i_t}(w) = \frac{1}{M}\sum_{j \in B_t} \nabla f_j(y_{t-1})$
  compute potential next direction $d_t = -\big((1-\beta)\nabla f_{i_t}(y_{t-1}) + \beta d_{t-1}\big)$
  $q_m = 0$
  while ($(\nabla f_{i_t}(y_{t-1}))^T d_t \geq 0$)
   reduce momentum term $\beta = \delta_m \beta$
   rejections counter $q_m = q_m + 1$
  end while
  set optimal mini-batch momentum term $\beta_t = \beta$
  $\alpha = \mathrm{reset}()$, $q_a = 0$
  compute potential next step $y_t = y_{t-1} + \alpha d_{t-1}$
  while ($f_{i_t}(y_t) > f_{i_t}(y_{t-1}) + \gamma\alpha\nabla f_{i_t}(y_{t-1})^T d_t$)
   reduce step-size $\alpha = \delta_a \alpha$
   rejections counter $q_a = q_a + 1$
  end while
  set optimal mini-batch step-size $\alpha_t = \alpha$
  make a (internal) step $y_t = y_{t-1} + \alpha_t d_t$
 end for
update weights $w^{k+1} = y_{N/M}$
epoch ends $k = k + 1$
end while
```

Algorithm 5: Mini-batch Gradient Descent with Armijo line search for step-size and Momentum restart

```
given $w^0 \in \mathbb{R}^{p+1}$, $\gamma \in (0,1)$, $\delta_a \in (0,1)$, $\alpha_0 \in \mathbb{R}^+$, $\delta_m \in (0,1)$, $\{\beta_k\} = \beta \in (0,1)$
$k = 0$
while ($\|\nabla f(w^k)\| > \varepsilon(1 + |f(w)|)$)
 shuffle $\{1,\ldots,N\}$ and split $B_1,\ldots,B_{N/M}$ such that $1 < |B_t| = M \ll N$
 set $y_0 = w^k$, $d_0 = 0$
 for $t = 1,\ldots,N/M$
  get mini-batch indices $i_t$ from $B_t$
  approximate true gradient $\nabla f_{i_t}(w) = \frac{1}{M}\sum_{j\in B_t}\nabla f_j(y_{t-1})$
  compute potential next direction $d_t = -\big((1-\beta)\nabla f_{i_t}(y_{t-1}) + \beta d_{t-1}\big)$
  if ($(\nabla f_{i_t}(y_{t-1}))^T d_t \geq 0$)
   restart direction $d_t = d_0$
  end if
  $\alpha = $ reset(), $q_a = 0$
  compute potential next step $y_t = y_{t-1} + \alpha d_{t-1}$
  while ($f_{i_t}(y_t) > f_{i_t}(y_{t-1}) + \gamma\alpha\nabla f_{i_t}(y_{t-1})^T d_t$)
   reduce step-size $\alpha = \delta_a \alpha$
   rejections counter $q_a = q_a + 1$
  end while
  set optimal mini-batch step-size $\alpha_t = \alpha$
  make a (internal) step $y_t = y_{t-1} + \alpha_t d_t$
 end for
 update weights $w^{k+1} = y_{N/M}$
 epoch ends $k = k + 1$
end while
```

3 Experiments

4 Mathematical background

Theorem 1 (Weirstrass theorem)**.** *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a continuous function and $S \subseteq \mathbb{R}^n$ a compact set. Then function f admits global minimum in S.*

Corollary 2 (Sufficient condition)**.** *If function $f\colon \mathbb{R}^n \to \mathbb{R}$ is a continuous and coercive function, then f admits global minimum in $\mathbb{R}^n$.*

Proposition 2 (Coercivity of a quadratic function)**.** *A quadratic function $f(x) = \frac{1}{2}x^T Qx - c^T x$ is said to be coercive if and only if the symmetric matrix $Q \in \mathbb{R}^{n\times n}$ is positive-defined.*

Proposition 3 (Unique global minimum)**.** *Let $S \subseteq \mathbb{R}^n$ be a convex set, let $f\colon S \to \mathbb{R}$ be a strictly convex function. Then the global minimum, if exists, is unique.*

Definition 1 (Convex function)**.** Let $S \subseteq \mathbb{R}^n$ be a convex set, a function $f\colon S \to \mathbb{R}$ is said to be convex if the hessian matrix is semi-positive-defined. If the hessian matrix is positive-defined, then the function is strictly convex.