Stochastic Gradient Descent with Momentum and Line Searches

David Nardi
MSc student in Artificial Intelligence, Univeristy of Florence

29th January 2024

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

1 Introduction **2**
1.1 Optimization problem 2

2 Mini-batch gradient descent variants **5**
2.1 Fixed step-size 5
2.2 Stochastic line search 5

1 Introduction

1.1 Optimization problem

$$\min_{w \in \mathbb{R}^p} f(w) = L(w) + \lambda \Omega(w)$$

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

where $i = \ldots, N$ are the dataset indices, $y^i \in \{-1, 1\}$ is the response variable corresponding to the negative or positive class, $x^i \in \mathbb{R}^p$ are dataset examples.

$$\nabla f(w) = X^T r + 2\lambda w, \quad r = -y^{(i)} \sigma(-y^{(i)} w^T x^{(i)})$$

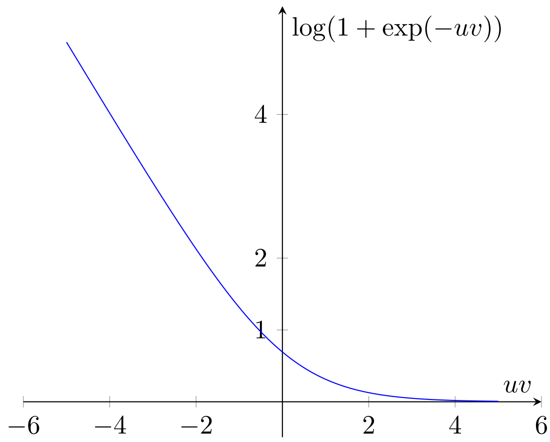
$$\nabla^2 f(w) = X^T D X + 2\lambda I, \quad d_{ii} = \sigma(y^{(i)} w^T x^{(i)}) \sigma(-y^{(i)} w^T x^{(i)})$$

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

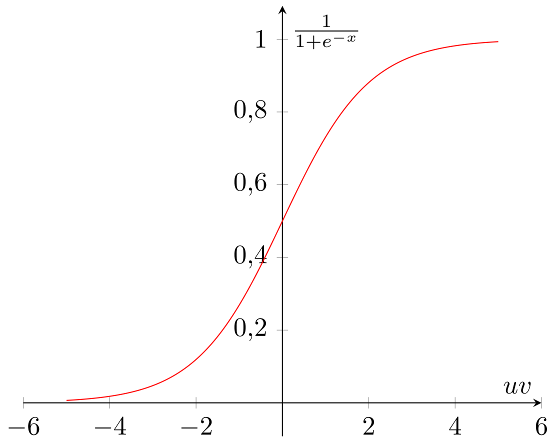
$$X^T = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \ldots & x_p^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \ldots & x_p^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(N)} & x_2^{(N)} & \ldots & x_p^{(N)} \end{pmatrix} \in \mathbb{R}^{N \times (p+1)}$$

$$w = \begin{pmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_p \end{pmatrix} \in \mathbb{R}^{p+1}$$

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



(a) *Log-loss*



(b) *Sigmoid function*

Figure 1: Log-loss and sigmoid function plots

- $uv \gg 0$: 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

Mini-batch Gradient Descent with fixed or decreasing step-size

dati $w^0 \in \mathbb{R}^n$, $f(w) = \sum_{i=1}^{N} f_i(w)$, $k = 0$ e $\{\alpha_k\} \mid \alpha_k = \alpha \vee \alpha_k = \frac{\alpha_0}{k+1}$
while ($\|\nabla f(w^k)\| > \varepsilon$)
 shuffle $\{1, \ldots, N\}$ in N/M blocchi $B_1, \ldots, B_{N/M}$ di dimensione $1 < |B_t| = M \ll N$
 $y_0 = w^k$
 for $t = 1, \ldots, N/M$
  get mini-batch indices from B_t
  $y_t = y_{t-1} - \alpha_k \frac{1}{M} \sum_{j \in B_t} \nabla f_j(y_{t-1})$
 end for
 $w^{k+1} = y_{N/M}$
 $k = k + 1$ fine epoca
end while

2.2 Stochastic line search

Mini-batch Gradient Descent with Armijo line search

dati $w^0 \in \mathbb{R}^n$, $f(w) = \sum_{i=1}^{N} f_i(w)$, $k = 0$, $\gamma \in (0,1), \delta \in (0,1)$
while ($\|\nabla f(w^k)\| > \varepsilon$)
 shuffle $\{1, \ldots, N\}$ and split $B_1, \ldots, B_{N/M}$ such that $1 < |B_t| = M \ll N$
 $y_0 = w^k$
 for $t = 1, \ldots, N/M$
  get mini-batch indices i_t from B_t
  approximate true gradient $\nabla f_{i_t}(w) = \frac{1}{M} \sum_{j \in B_t} \nabla f_j(y_{t-1})$
  $\alpha = \mathrm{reset}()$, $q = 0$
  while ($f_{i_t}\big(y_{t-1} - \alpha \nabla f_{i_t}(w)\big) > f_{i_t}(y_{t-1}) - \gamma \alpha \|\nabla f_{i_t}(y_{t-1})\|^2$)
   $\alpha = \delta \alpha$
   $q = q + 1$
  end while
  $\alpha_t = \alpha$
  $y_t = y_{t-1} - \alpha_t \nabla f_{i_t}(y_{t-1})$
 end for
 $w^{k+1} = y_{N/M}$
 $k = k + 1$
end while