STOCHASTIC GRADIENT METHODS

- The following optimization problem, which minimizes the sum of cost functions over samples from a finite training set, appears frequently in machine learning:

\[
\min F(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x),
\]

where \( n \) is the number of samples, and each \( f_i : \mathbb{R}^d \rightarrow \mathbb{R} \) is the cost function corresponding to a training set element.

- When \( n \) is large, computing \( F(x) \) and \( \nabla F(x) \) is prohibited.
- Stochastic Gradient (SG) method and its variants have been the main approaches for solving (1);
- in the \( t \)-th iteration of SG, a random index of a training sample \( i_t \) is chosen from \( \{1, 2, \ldots, n\} \) and the iterate \( x_t \) is updated by

\[
x_{t+1} = x_t - \eta_t \nabla f_i(x_t)
\]

where \( \nabla f_i(x_t) \) denotes the gradient of the \( i_t \)-th component function at \( x_t \), and \( \eta_t > 0 \) is the step length or learning rate [1].

ADAPTIVE STEPLENGTH SELECTION IN THE STOCHASTIC FRAMEWORK

- The deterministic framework: Selections based on the Ritz-like values [2]
  Choose the steplengths for \( m_R \) next iterations as

\[
\eta_{t+i} = \frac{1}{\beta_i}, \quad i = 1, \ldots, m_R \quad (m_R = 3, 4, 5)
\]

where \( \beta_i \) are the eigenvalues of an \( m_R \times m_R \) symmetric tridiagonal matrix \( T \) derived from the last \( m_R \) gradients

\[
[\nabla F(x_{t-m_R}), \ldots, \nabla F(x_{t-1})]
\]

by generalizing the Lanczos process for approximating the eigenvalues of a symmetric matrix.

In case of quadratic objective function \( F(x) = \frac{1}{2} x^T A x - b^T x \), the values \( \beta_i \) (called Ritz values) are approximations of \( m_R \) eigenvalues of the symmetric positive definite matrix \( A \).

In the general non-quadratic case, the values \( \beta_i \) tend to approximate \( m_R \) eigenvalues of the Hessian matrix at the solution [4].

Compute the symmetric tridiagonal matrix \( T \)

- Let \( G = [\nabla F(x_{t-m_R}), \ldots, \nabla F(x_{t-1})] \quad (m_R = 3, 4, 5) \)

- Compute the Cholesky decomposition \( G^T R = R^T R \)

- Compute

\[
J = \begin{pmatrix}
-\eta_{1,m_R} & \cdots & \eta_{2,m_R} \\
-\eta_{2,m_R} & \ddots & \cdots \\
\vdots & \ddots & -\eta_{m_R,m_R}
\end{pmatrix}
\]

- Compute \( \tilde{T} = [R \ v] J R^T \)

- \( T = \text{tril}(T) + \text{tril}(T, -1)' \)

- The stochastic framework: Selection based on Ritz-like values in SG
  - \( \text{Exploit } \tilde{G} = \nabla f_i(x_{t-m_R}), \ldots, \nabla f_i(x_{t-1}) \)
  - in computing the Ritz-like values \( \beta_i \) for the next \( m_R \) iterations and set in SGD

\[
\eta_t = \max \left\{ 10^{\eta_0 - \frac{1}{m_R}}, 10^{-1} \eta_0 \right\}
\]

THE TEST PROBLEM

- Logistic regression with l2 norm regularization:

\[
\min_x F(x) = \frac{1}{n} \sum_{i=1}^{n} \left[ \log (1 + \exp(-h_i^T x)) + \frac{\lambda}{2} ||x||^2 \right]
\]

where \( h_i \in \mathbb{R}^d \) and \( b_i \in \{0, 1\} \) are the feature vectors and class labels of the \( i \)-th sample, respectively, and \( \lambda > 0 \) is a regularization parameter;

- database: MNIST 8 and 9 digits (binary classification), dimension: 11800 x 784.

ADAM ALGORITHM [3]

Algorithm 1 Adam

1. Choose \( m_0, \eta, \epsilon \), and \( a_t, b_t \in [0, 1] \), \( a_0 \); initialze \( m_0 = 0, \epsilon_t = 0 \), \( t = 0 \)
2. for \( t \in \{0, \ldots, m_0\} \) do
3. \( t \leftarrow t + 1 \)
4. \( a_t \leftarrow \nabla f_i(x_t) \)
5. \( m_t = b_{t-1} m_{t-1} + (1 - b_t) \cdot a_t \)
6. \( v_t = b_{t-1} v_{t-1} + (1 - b_t) \cdot a_t \)
7. \( 0 < \beta_t = m_t^T m_{t-1} \)
8. \( 0 < \delta_t = \eta_{t-1} \cdot \epsilon / (\sqrt{\beta_t} + \epsilon) \)
9. \( x_{t+1} = x_t - \eta_t \cdot (m_t / (\sqrt{\beta_t} + \epsilon)) \)
end for
11. Result: \( x_f \)

EXPERIMENTAL RESULTS

- Adaptive steplenghts make the algorithms more robust than the standard SG methods and provide performances comparable with SG with best-tuned steplenghts [6], [5];
- further study to improve the adaptive steplenght rules also in the stochastic case;
- validation of the stochastic-Ritz version: experiments on other database and other loss-functions;
- exploit mini-batch of adaptive size; analyse the sensitivity of the step size rules to the mini-batch size, possible combination with inexact Line-Search.

CONCLUSION AND PERSPECTIVE WORK

REFERENCES