Steplength and mini-batch size selection in Convolutional Neural Networks

Giorgia Franchini frngrg@unife.it

BOS/SOR2020 Conference, Palais Staszic, Warsaw / virtual mode
Systems and Operational Research 2020

15 December 2020 – Tuesday

OASIS
Optimization
Algorithms and
Software for
Inverse problemS

RegioneEmilia-Romagna

Optimization problem in machine learning

The problem we consider is the unconstrained minimization of the form

$$\min_x F(x) = \mathbb{E}[f(x, \xi)]$$

where ξ is a multi-value random variable and f represents the cost function.

We haven't complete information about the probability distribution of ξ. In practice, we seek the solution of a problem that involves an estimate of the objective function $F(x)$.

Optimization problem in machine learning

For example: minimize the sum of cost functions depending on a finite training set, composed by sample data ξ_i, $i \in \{1 \dots n\}$:

$$\min_x F_n(x) = \frac{1}{n} \sum_{i=1}^{n} f(x, \xi_i) = \frac{1}{n} \sum_{i=1}^{n} f_i(x),$$

where n is the size of the training set and each $f_i(x) \equiv f(x, \xi_i)$ denotes the cost function related to the instance ξ_i of the training set elements.

Stochastic Gradient (SG)

For very large training set, the computation of $F_n(x)$ and $\nabla F_n(x)$ is prohibited and Stochastic Gradient (SG) method and its variants have been chosen as the main approaches to address the problem.

Algorithm 1 Stochastic Gradient (SG) method

```
Choose an initial iterate $x_1$.
for $k = 1, 2, \dots$ do
    Generate a realization of the random variable $\xi_k$.
    Compute a stochastic gradient $g(x_k, \xi_k)$.
    Choose a learning rate $\eta_k > 0$.
    Set the new iterate as $x_{k+1} \leftarrow x_k - \eta_k g(x_k, \xi_k)$.
end for
```

How to compute $g(x_k, \xi_k)$

In particular, we point out two different strategies for the choices of ξ_k and $g(x_k, \xi_k)$:

- **simple SG**: a realization of ξ_k may be given by the choice of a single sample element, or, in other words, a random index i_k is chosen from $\{1, 2, \ldots, n\}$ and the stochastic gradient is defined as

$$g(x_k, \xi_k) = \nabla f_{i_k}(x_k),$$

  where $\nabla f_{i_k}(x_k)$ denotes the gradient of the i_k-th component function at x_k;

- **mini-batch**: the random variable ξ_k may represents a small subset $S_k \subset \{1, ..., n\}$ of samples, randomly chosen at each iteration, so that the stochastic gradient is defined as

$$g(x_k, \xi_k) = \frac{1}{|S_k|} \sum_{i \in S_k} \nabla f_i(x_k).$$

Stochastic Gradient with Momentum

Algorithm 2 Momentum

```
Choose $maxit$, $\eta$, $\beta \in [0, 1)$, $x_0$;
initialize $m_0 \leftarrow 0$, $t \leftarrow 0$
for $t \in \{0, \dots, maxit\}$ do
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla f_{i_t}(x_{t-1})$
    $m_t \leftarrow \beta \cdot m_{t-1} + g_t$
    $x_t \leftarrow x_{t-1} - \eta_t \cdot m_t$
end for
Result: $x_t$
```

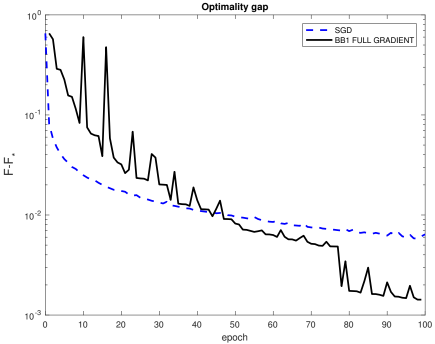
Kingma, Lei Ba, Adam: a method for stochastic optimization, ArXiv, 2017

Algorithm 3 Adam

```
Choose $maxit$, $\eta$, $\epsilon$, $\beta_1$ and $\beta_2 \in [0,1)$, $x_0$;
initialize $m_0 \leftarrow 0$, $v_0 \leftarrow 0$, $t \leftarrow 0$
for $t \in \{0, \dots, maxit\}$ do
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla f_{i_t}(x_{t-1})$
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$
    $\eta_t = \eta \frac{\sqrt{1-\beta_2^t}}{(1-\beta_1^t)}$
    $x_t \leftarrow x_{t-1} - \eta_t \cdot m_t / (\sqrt{v_t} + \hat{\epsilon})$
end for
Result: $x_t$
```

Stochastic versus deterministic

Numerical evidence shows very real advantages of SG with respect to a full gradient or other deterministic methods within the early epochs.



Theoretical results

Under a set of suitable assumptions on the smoothness of F and on the first and second moments of the stochastic directions $\{g(x_k, \xi_k)\}$, and on the steplength selection

$$0 < \eta_{min} \leq \eta_k \leq \eta_{max} \leq \frac{\nu}{L}$$

where L is the Lipschitz constant of the gradient and ν is a constant depending on the first and second moments of the stochastic process, we have that

$$\mathbb{E}[F(x_k) - F_*] \xrightarrow{k\to\infty} \eta_{max}\frac{L}{c}\gamma$$

where c is the strongly convexity constant of the function $F(x)$ and γ is again a constant depending on first and second moments of the stochastic process [Bottou et al, 2018].

Theoretical results

Under the same assumptions but without convexity, we have that

$$\mathbb{E}\left[\| \nabla F(x_k) \|_2^2\right] \xrightarrow{k\to\infty} \eta_{max} L \bar{\gamma}.$$

where γ is again a constant depending on first and second moments of the stochastic process [Bottou et al, 2018].

Limitation of the method

- This result shows that if the steplength is sufficiently small, then the expected objective values will converge to a neighborhood of the optimal value;
- in practice, since the constants related to the assumptions, such as the Lipschitz parameter, or the parameters involved in the bounds of the moments of the stochastic directions, are unknown and not easy to approximate, the steplength is selected as a fixed small value η;
- nevertheless, a too small steplength can give rise to a very slow learning process;
- for this reason, in [Sopyla et al, 2015; Tan et al, 2016] rules for an adaptive selection of the steplength have been proposed.

Contribution of my work: the steplength selection rule adopted in the limited memory gradient projection method [Fletcher, 2012] has been tailored to the SG framework.

Beyond the method limitations

In literature can be found two different approaches to solve the method limitations:

- Choose an appropriate steplenth
- Increase the mini-batches size

The two strategies have been addressed in my work, both separately and jointly.

[Franchini at al. (2018)**Artificial Neural Networks: the missing link between curiosity and accuracy**, Advances in Intelligent Systems and Computing, vol 941, Springer 2018]

[Franchini, Ruggiero, Zanni (2020) **On the Steplength Selection in Stochastic Gradient Methods**, Numerical Computations: Theory and Algorithms. vol 11973. Springer]

[Franchini, Ruggiero, Zanni (2020) **Ritz-like values in steplength selections for stochastic gradient methods**, Soft Computing]

Deterministic case in a nutshell

- In the strictly convex quadratic case,

  $$f(x) = \frac{1}{2}x^TAx - b^Tx$$

  in order to capture second order information of the considered problem, the steplengths are defined as the inverse of suitable approximations of the eigenvalues of the Hessian matrix, given by its Ritz values;
- the key point is to obtain the Ritz values in an inexpensive way;
- the basic idea is to divide the sequence of iterations into groups of m_R iterations referred to as *sweeps*, where m_R is a small positive integer, and to compute the steplengths for each sweep as the inverse of some Ritz values of the Hessian matrix A, computed by exploiting the gradients of the previous sweep.

How to compute Ritz values

At the iteration $k \geq m_R$, where m_R is a little integer, we denote by G and J the matrices obtained collecting m_R gradient vectors computed at previous iterates and the related steplengths:

$$G = [g_{k-m_R}, \dots, g_{k-1}], \quad J = \begin{pmatrix} \eta_{k-m_R}^{-1} & & \\ -\eta_{k-m_R}^{-1} & \ddots & \\ & \ddots & \eta_{k-1}^{-1} \\ & & -\eta_{k-1}^{-1} \end{pmatrix},$$

from the recurrent formula:

$$g_i = g_{i-1} - \eta_{i-1} A g_{i-1}, \quad i \geq 0$$

we can write

$$AG = [G, g_k] J.$$

Tridiagonal matrix

$$AG = [G, g_k]J.$$

This equation is useful to compute the tridiagonal matrix T resulting from the application of m_R iterations of the Lanczos process to the matrix A, with starting vector $q_1 = g_{k-m_R} / \parallel g_{k-m_R} \parallel$; this procedure generates an orthogonal matrix $Q = [q_1, \ldots, q_{m_R}]$, whose columns are a basis for the Krylov subspace $\{g_{k-m_R}, Ag_{k-m_R}, A^2 g_{k-m_R}, ..., A^{m_R-1} g_{k-m_R}\}$, such that

$$T = Q^T AQ$$

with $T \in \mathbb{R}^{m_R \times m_R}$.

How to use pseudo Ritz values as steplength

The steplengths for the next m_R gradient iterations are defined as the inverse of the eigenvalues θ_i of T, that are the so-called Ritz values:

$$\eta_{k-1+i} = \frac{1}{\theta_i}, \qquad i = 1, \dots, m_R.$$

Procedure to avoid the explicit computation of Q:

⇒

$$G = QR \quad \Rightarrow \quad R^T R = G^T G$$

where R is upper triangular;

⇒ then the matrix T can be obtained as follows:

$$T = R^{-T} G^T A G R^{-1} = R^{-T} G^T [G, g_k] J R^{-1} = [R, r] J R^{-1},$$

where the vector r is the solution of the linear system $R^T r = G^T g_k$.

General case

$$AG = [G, g_k]J.$$

- In the general case, the recurrence does not hold and the described procedure provides an Hessenberg matrix;
- in [Fletcher, 2012; Di Serafino et al., 2018] T is replaced by $\overline{T} = tril(T) + tril(T, -1)'$;
- the eigenvalues θ_i of $\overline{T}$ tend to approximate m_R eigenvalues of the Hessian matrix.

Harmonic Ritz values

A similar idea consists in obtaining the steplengths as the reciprocal of the eigenvalues of $T_j^{-1} P_j$, where

$$\begin{aligned} P_j &= R_j^{-T} J_j^T \begin{pmatrix} R_j & r_j \\ 0 & \rho_j \end{pmatrix}^T \begin{pmatrix} R_j & r_j \\ 0 & \rho_j \end{pmatrix} J_j R_j^{-1} = \\ &= \begin{pmatrix} T_j^T & t_j \end{pmatrix} \begin{pmatrix} T_j \\ t_j^T \end{pmatrix}, \end{aligned}$$

$\rho_j = \sqrt{g_{j+m}^T g_{j+m} - r_j^T r_j}$ and t_j is the solution of the linear system $R_j^T t_j = J_j^T \begin{pmatrix} 0 \\ \rho_j \end{pmatrix}$.

In QP problems, the eigenvalues of $T_j^{-1} P_j$ (**harmonic Ritz values**) are approximations of eigenvalues of the Hessian.

In general cases, replacing T_j by the non-singular tridiagonal matrix $\overline{T}_j$, a pentadiagonal matrix $\overline{P}_j$ is obtained and the eigenvalues of $\overline{T}_j^{-1} \overline{P}_j$ can be computed (**harmonic Ritz-like values**).

Stochastic Context

In the stochastic context, the implementation of this technique involves some crucial differences:

- stochastic gradients instead of full gradients in the formation of G:

$$G = [g_{k-m_R}(x_{k-m_R}, \xi_{k-m_R}), \dots, g_{k-1}(x_{k-1}, \xi_{k-1})];$$

- approximation of T by its symmetric part $\tilde{T} = (T + T^T)/2$;
- steplength selection of the next sweep as follows:

$$\eta_{k-i+1} = \max\left\{\min\left\{\eta_{max}, \frac{1}{\theta_i}\right\}, \eta_{min}\right\}, \quad i = 1, \dots, m_R.$$

  where θ_i is an eigenvalue of $\tilde{T}$;
- the thresholding procedure eliminates the negative eigenvalues and the ones out of the interval $[\eta_{min}, \eta_{max}]$.

A-R and AA-R versions of SG

The positive harmonic Ritz-like values generate shorter steplengths with respect to the ones defined by the corresponding Ritz-like values.

- **Alternate Ritz-like values (A-R)** method: simply **toggle** the use of the Ritz-like values to the one of the harmonic Ritz-like values at each sweep

- **Adaptive Alternate Ritz-like values (AA-R)** method: **links** the choice between Ritz-like and harmonic Ritz-like values to the selection of the size of the current subsample.
  In particular, when at the iteration k the size of the sample increases, **the stochastic gradients previously stored are related to subsamples of lower size**; then, we discard the available Ritz-like values and we exploit the current stored stochastic gradients to determine a set of harmonic Ritz-like values, using shorter steplengths in this transition phase.

Mini-batch SG based on increasing size

Recently, in [Bollagragrada, Byrd, Nocedal 2018], the norm test is replaced by an ***inner product test***, combined with an ***orthogonality test***, aimed to guarantee that **the negatives of the stochastic gradients based on subsamples of suitable size are descent directions in expectation**.

In view of the assumption

$$\mathbb{E}[g_k^{(n_k)}] = \nabla F(x^{(k)}) \quad \Rightarrow \quad \mathbb{E}[g_k^{(n_k)^T} \nabla F(x^{(k)})] = \|\nabla F(x^{(k)})\|^2$$

Condition on the sample size

The following condition can be imposed on the sample size n_k of $\xi^{(n_k)}$:

$$\mathbb{E}[({g_k^{(n_k)}}^T \nabla F(x^{(k)}) - \|\nabla F(x^{(k)})\|^2)^2] \leq \theta^2 \|\nabla F(x^{(k)})\|^4 \qquad \textbf{\textit{inner product test}}$$

$$\mathbb{E}[\|g_k^{(n_k)} - \frac{{g_k^{(n_k)}}^T \nabla F(x^{(k)})}{\|\nabla F(x^{(k)})\|^2} \nabla F(x^{(k)})\|^2] \leq \nu^2 \|\nabla F(x^{(k)})\|^2 \qquad \textbf{\textit{orthogonality test}}$$

for some $\theta, \nu > 0$.

The combination of the two tests is known as ***augmented inner product test***.

Numerical evidence highlights that the mechanism give rises to an increase of n_k slower than the one induced by the norm test.

Pratical point of view

From the practical point of view, in view of the previously used argument, n_k has to satisfy the following conditions:

$$\frac{\mathbb{E}[(\nabla f_i(x^{(k)})^T \nabla F(x^{(k)}) - \|\nabla F(x^{(k)})\|^2)^2]}{n_k} \leq \theta^2 \|\nabla F(x^{(k)})\|^4 \qquad \textbf{exact variance inner product test}$$

$$\frac{\mathbb{E}[\|\nabla f_i(x^{(k)}) - \frac{\nabla f_i(x^{(k)})^T \nabla F(x^{(k)})}{\|\nabla F(x^{(k)})\|^2} \nabla F(x^{(k)})\|^2]}{n_k} \leq \nu^2 \|\nabla F(x^{(k)})\|^2 \qquad \textbf{exact variance orthogonality test}$$

Approximating the variance with the sample variance and the gradient $\nabla F(x^{(k)})$ with a sample gradient, the conditions for n_k can be written as

$$\frac{\sum_{i \in S_k} (\nabla f_i(x^{(k)})^T g_k^{(n_k)} - \|g_k^{(n_k)}\|^2)^2}{n_k(n_k - 1)} \leq \theta^2 \|g_k^{(n_k)}\|^4 \qquad \textbf{approximate inner product test}$$

$$\frac{\sum_{i \in S_k} \|\nabla f_i(x^{(k)}) - \frac{\nabla f_i(x^{(k)})^T g_k^{(n_k)}}{\|g_k^{(n_k)}\|^2} g_k^{(n_k)}\|^2}{n_k(n_k - 1)} \leq \nu^2 \|g_k^{(n_k)}\|^2 \qquad \textbf{approximate variance orthogonality test}$$

Mini-batch size increasing rule

When these conditions are not satisfied by the current sample size, the sample size is increased:

$$n_k = \min(\lceil \max(Z_1, Z_2) \rceil, n)$$

where

$$Z_1 = \frac{\mathrm{Var}_{i \in S_k}(\nabla f_i(x^{(k)})^T g_k^{(n_k)})}{\theta^2 \|g_k^{(n_k)}\|^4}, \quad Z_2 = \frac{\mathrm{Var}_{i \in S_k}\left(\nabla f_i(x^{(k)}) - \frac{\nabla f_i(x^{(k)})^T g_k^{(n_k)}}{\|g_k^{(n_k)}\|^2} g_k^{(n_k)}\right)}{\nu^2 \|g_k^{(n_k)}\|^2}$$

Data-sets

In order to evaluate the effectiveness of the proposed steplength rule for SG methods, we consider the optimization problems arising in training binary classifiers for two well known data-sets:

- the *MNIST* data-set of handwritten digits, commonly used for testing different systems that process images; the images are in gray-scale $(0, 255)$, in our case normalized $(0, 1)$, centered in a box of 28×28 pixels. The database contains $60,000$ images for the train set and $10,000$ images for the test set.

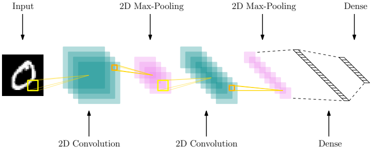
The minimization problem

We built multi-class classifier corresponding to a loss functions originating from a Convolutional Neural Network (CNN); a regularization term was added to avoid overfitting. Thus the minimization problem has the form

$$\min_x F_n(x) + \frac{\lambda}{2}\|x\|_2^2,$$

where $\lambda > 0$ is a regularization parameter.

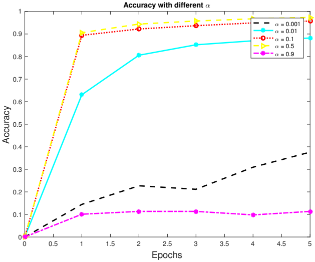
A non-convex problem: a Convolutional Neural Network



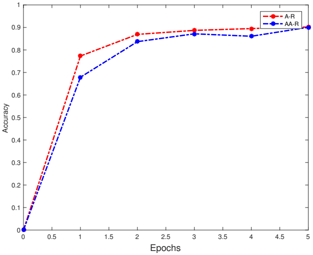
Convolutional Neural Network (CNN): an input layer, two sequences of convolutional and max-pooling layers, a fully connected layer and an output layer, given by a Rectified Linear Unit (ReLU) activations combined by a softmax function; the loss function is the cross entropy:

- regularization parameter $\delta = 10^{-4}$;
- the first convolutive layer is composed by 64 filters, each filter has 5×5 dimension; after we apply a max-pooling of size 2×2;
- the second convolutive layer is composed by 32 filters, each filter has 5×5 dimension; after we apply a max-pooling of size 2×2

CNN: results, SG case



CNN Accuracy in the **SG mini**, $|S| = 50$



Accuracy obtained with **A-R** and **AA-R**

The setting of **A-R** and **AA-R** methods is:

- for **A-R** method, $\alpha_{min} = 10^{-3}, \alpha_{max} = 1, \; n_0 = 10;$
- for **AA-R** method, $\alpha_{min} = 10^{-2}, \alpha_{max} = 1, \; n_0 = 3.$

$\overline{\alpha}$ is set as 0.1 in all cases.

The subsample size increases up to a maximum of 204 and 182 in **A-R** and **AA-R** methods respectively.

CNN: results, Momentum case

The setting for SG with momentum:

- $\beta = 0.9$;
- $|S| = 50$ subsample size.

The setting of **A-R** and **AA-R** methods is:

- for **A-R** method, $\alpha_{min} = 10^{-3}$, $n_0 = 10$;
- for **AA-R** method, $\alpha_{min} = 10^{-3}$,, $n_0 = 10$.

Table: Numerical results of the considered methods with Momentum optimiser after 5 epochs.

α	SG_{mom}	α_{max}	**A-R**	**AA-R**
0.01	0.8819	0.8	0.8783	0.9182
0.1	0.9573	0.9	0.8675	0.8829
0.5	0.9708	1	0.902	0.9269
0.9	0.0958	1.2	0.8733	0.8644

CNN: results, AdaM case

The setting for AdaM:

- $\beta_1 = 0.9$ and $\beta_2 = 0.999$;
- $\epsilon = 1e - 8$;
- $|S| = 50$ subsample size.

The setting of **A-R** and **AA-R** methods is:

- for **A-R** method, $\alpha_{min} = 10^{-3}$, $n_0 = 10$;
- for **AA-R** method, $\alpha_{min} = 10^{-3}$,, $n_0 = 10$.

Table: Accuracies of the considered methods with AdaM optimiser after 5 epochs.

α	SG_{AdaM}	α_{max}	**A-R**	**AA-R**
0.001	0.9705	0.1	0.8768	0.9061
0.01	0.9492	0.3	0.9557	0.9333
0.1	0.1148	0.5	0.8537	0.8372

Subsample size

Table: Subsample size of the considered methods with Momentum and AdaM optimiser after 5 epochs.

	Momentum			*AdaM*	
α_{max}	**A-R**	**AA-R**	α_{max}	**A-R**	**AA-R**
0.8	327	306	0.1	182	204
0.9	214	294	0.3	263	244
1	204	155	0.5	226	363

Conclusions and future works

- The deterministic procedure for obtaining the Ritz-like values is reformulated in the stochastic framework
- New adaptive subsampling strategy enables to control the variance of the stochastic directions
- Two different ways to select the current steplength, by simply toggling the Ritz-like values with the harmonic Ritz-like values (A-R method) or using the harmonic Ritz-like values only when the size of the subsample is increased (AA-R method)
- The novel methods enable to obtain an accuracy similar to the one obtained with SG mini-batch with fixed best-tuned steplength
- The approach appears sligthly dependent on the bounds imposed on the steplengths, making the parameters setting less expensive with respect to the SG framework
- The proposed technique provides a guidance on the learning rate selection and it allows to perform similarly to the SG approach equipped with the best-tuned steplength
- Combination with proximity operator?

Thanks for your attention!

frngrg@unife.it