Adaptive high order stochastic descent algorithms

Gabriel Turinici
partially in collaboration with I. Ayadi

CEREMADE, Université Paris Dauphine - PSL Research University

Numerical Analysis, Numerical Modeling, Approximation Theory (NA-NM-AT 2022) Conference
Cluj-Napoca, Romania Oct. 26-28 2022

Adaptive Stochastic Gradient Descent: motivations

• in statistical / machine learning, neural networks (NN) (e.g., classification, generation, reinforcement learning, ...) boils down to the minimization of a loss function : $f(X) := \frac{1}{N}\sum_{i=1}^{N} f(\omega_i, X)$, ω_i = available samples.

• Equivalent writing: $f(X) = \mathbb{E}_\omega f(\omega, X)$. $X \in \mathbb{R}^d$ = parameters of the NN, ω the training examples (or "states" in RL)

• optimization by gradient descent (classical) : $X_{n+1} = X_n - h\nabla f(X_n)$, $h > 0$ is the learning rate (="step size").

• PROBLEM : computing $\nabla f(X_n) = \mathbb{E}_\omega \nabla f(\omega, X_n)$ is too costly because of the average (many samples).

• $\nabla f(X_n)$ is replaced by an unbiased estimate to get the Stochastic Gradient Descent (SGD): $X_{n+1} = X_n - h\nabla f(\omega_{\gamma_n}, X_n)$
$(\gamma_n)_{n\geq 1}$ = i.i.d uniform random variables in $\{1, 2, .., N\}$.
Note: other unbiased estimates can be used beyond $\nabla f(\omega_{\gamma_n}, X_n)$ (e.g., in RL)

Adaptive Stochastic Gradient Descent

- Stochastic Gradient Descent $X_{n+1} = X_n - h\nabla f(\omega_{\gamma_n}, X_n)$
$(\gamma_n)_{n\geq 1}$ are i.i.d uniform in $\{1, 2, .., N\}$.
Problem: small h converge slowly, large h : stochastically unstable.

- MAIN QUESTION: how to (optimally) choose the learning rate (l.r.) h ?

- Flow interpretation : in the limit $h \to 0$ the minimization of $f(X)$ is some approximation of the 'continuous time' evolution equation $X'(t) = \nabla_X f(X(t))$. SGD: $X_n \simeq X(t_n)$, $t_n = n \cdot h$.

- MAIN HIGH ORDER + ADAPTATIVITY IDEA:
1/ construct a better approximation Y_{n+1} of $X(t_{n+1})$ such that $Y_{n+1} - X_{n+1}$ is an estimation of the error $X_{n+1} - X(t_{n+1})$.
2/ Using Y_{n+1} compute the largest l.r. h such that stability still holds
- Question 1: find a high order scheme consistent for the flow dynamics
- Question 2: is the procedure performing well in practice..

The second order Stochastic Runge Kutta "SRK" scheme

Stochastic Runge Kutta (SRK)

$$\tilde{Y}_{n+1} = Y_n - h\nabla f_{\gamma_n}(Y_n),\ \ Y_{n+1} = Y_n - \frac{h}{2}\left[\nabla f_{\gamma_n}(Y_n) + \nabla f_{\gamma_n}(\tilde{Y}_{n+1})\right]. \quad (1)$$

Rq: same γ_n in $\nabla f_{\gamma_n}(Y_n)$ and $\nabla f_{\gamma_n}(\tilde{Y}_{n+1})$!

Theorem (Convergence of SGD and SRK schemes, I.A., G.T. 2021)

Suppose $\forall k$, ∇f_k is a Lipschitz function, ∇f_k and its partial derivatives up to order 6 have at most polynomial increase at ∞ and ∇f_k increases at most linearly at infinity. Then the SGD scheme converges at (weak) order 1 (in h) while the SRK scheme (1) converges at (weak) order 2.

Proof idea: it is known (Q. Li, C. Tai W. E. 2017) that SGD weakly converges ($h \to 0$, match : $Y_n \simeq Z_{nh}$) to the solution of the SDE $dZ_t = -\nabla f(z)dt + \sqrt{hV(Z_t)}dW_t$, $Z(0) = X(0)$, W_t = Brownian motion, $V(z) = cov(f(\omega, z)) = \frac{1}{N}\sum_{k=1}^{N}(\nabla f(\omega_k, z) - \nabla f(z)) \cdot (\nabla f(\omega_k, z) - \nabla f(z))^T$

Adaptive step SGD: the SGD-G2 algorithm

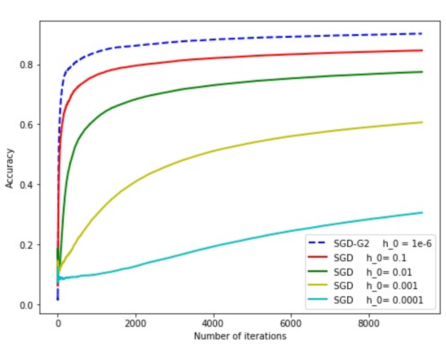
Algorithm 1 SGD-G2

Set hyper-parameter: β, mini-batch size M, choose stopping criterion
Input: initial learning rate h_0, initial guess X_0
Initialize iteration counter: $n = 0$
while stopping criterion not met **do**
  select next mini-batch γ_n^m, $m = 1, ..., M$
  Compute $g_n = \frac{1}{M}\sum_{m=1}^{M} \nabla f_{\gamma_n^m}(X_n)$
  Compute $\tilde{g}_n = \frac{1}{M}\sum_{m=1}^{M} \nabla f_{\gamma_n^m}(X_n - h_n g_n)$
  Compute $h_n^{opt} = \begin{cases} \frac{3}{2}\frac{h_n\langle g_n - \tilde{g}_n, g_n\rangle}{\|g_n - \tilde{g}_n\|^2} & \text{if } \langle g_n - \tilde{g}_n, g_n\rangle > 0 \\ h_n & \text{otherwise.} \end{cases}$
  if $h_n^{opt} > h_n$ **then**
    $h_{n+1} = \beta h_n + (1-\beta) h_n^{opt}$
  else
    $h_{n+1} = h_n^{opt}$
  end if
  Update $X_{n+1} = X_n - h_{n+1} g_n$
  Update $n \to n + 1$
end while

Remark: "stable + consistent thus convergent" algorithm

Empirical validation (MNIST / FMNIST / CIFAR10)

Results on standard datasets are performing well, start with h small then let it adapt itself.



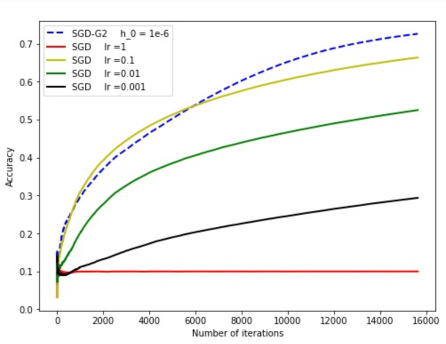
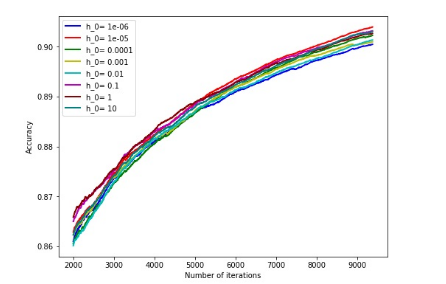


Figure: **Left:** SGD vs. SGD-G2 on FMNIST . **Right:** SGD vs. SGD-G2 on CIFAR10 (10 epochs).

Empirical validation on CIFAR100



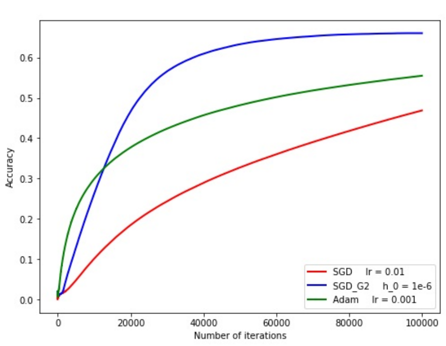


Figure: **Left:** Numerical results (over the first 5 epochs) for the SGD-G2 algorithm on the FMNIST dataset with several choices of the initial learning rate h_0; **right:** SGD , SGD-G2 and Adam (100 epochs) on CIFAR100.

Conclusion

• We presented a new adaptive learning rate procedure that performs well on standard datasets (MNIST, FMNIST, CIFAR10, CIFAR100)

• procedure robust with respect to the initial learning rate h_0

• in the process we came up with a proof for the convergence of the Stochastic Runge-Kutta second order scheme

• future work : to prove that $h \to 0$ and thus convergence to optimal X is reached (if possible general hypothesis) ; compare with other adaptive stochastic optimization algorithms.

Want to know more:
• these slides: https://doi.org/10.5281/zenodo.7257154 (DOI=10.5281/zenodo.7257154) ; also on https://turinici.com
• algorithm details paper (Arxiv ID= arXiv:2002.09304) : https://arxiv.org/abs/2002.09304
• self-contained SGD convergence proof (Arxiv ID=arXiv:2103.14350) : https://arxiv.org/abs/2103.14350
• related video: https://youtu.be/z_V2OIM0UmI