ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

are decaying as $\mu_t = \mu_0 \lambda^t, 0 < \lambda < 1$ and $\alpha_t = \alpha_0 t^{-c}, 0 < c < 1$; (A3) both the gradients and the variation of the parameter are bounded, $\|\boldsymbol{g}_t\|_\infty \leq C_g$ and $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_\infty \leq C_\theta$; (A4) the rate of change of the preconditioner is bounded, $\alpha_t p_{t,i} \leq \alpha_{t-1} p_{t-1,i}, \forall i$. Here we denote by index i the ith element of a vector. The first three assumptions are used in other algorithms, e.g ADAM [1].

To prove convergence under these assumptions, we upper-bound $|a_{t+1,i}|$, and $g_{t,i}\left(\theta_{t,i} - \theta_i^*\right)$. Using these bounds we find an upper bound for $R(T)$, as in Theorem 1. A sketch of the derivations is presented in Appendix A. We show in Corollary 1 that the cost function $f_t(\boldsymbol{\theta})$ asymptotically converges to $f_t(\boldsymbol{\theta}^*)$.

Theorem 1. *Under assumptions* (A1-4)*, $R(T)$ is bounded from above as*

$$\begin{aligned} R(T) \leq & \frac{C_p}{2\alpha_0} \sum_{i=1}^{d} \left[T^c C_\theta^2 + \alpha_0^2 C_{pg}^2 \left(\frac{1+2\mu_0}{1-\mu_0} \right)^2 \left(\frac{2(1-\mu_0)}{1-\lambda} + \frac{1}{(1-\lambda^2)^2} \right) + \right. \\ & \left. 2\mu_0^2 \alpha_0 C_\theta C_{pg} \left(\frac{1}{1-\lambda^2} + \frac{1}{(1-\mu_0)(1-\lambda^3)^2} \right) + \alpha_0^2 C_{pg}^2 \left(1+2\mu_0\right)^2 \sum_{t=1}^{T} t^{-c} \right] \quad (4) \\ \leq & \mathcal{O}\left(T^c + \sum_{t=1}^{T} t^{-c} \right). \end{aligned}$$

Corollary 1. *Under the assumptions from Theorem 1 it follows that the regret function grows slower than T, namely that* $\lim_{T\to\infty} \frac{R(T)}{T} = 0$.

Proof. In Theorem 1, all terms from (4) grow slower than T since the hyper-harmonic series, $\sum_{t=1}^{T} t^{-c}$, for $0 < c < 1$, divergence is proportional to T^{1-c}. ■

4 Simulations and results

We study the convergence properties of the proposed PA-SGD method in comparison with ADAM [1], AMSGrad [5] and NASGD [3]. For all simulations we use the MNIST hand written number database. For better visibility, we report the average evolution of the cost function values across iterations, for the best choice of configuration parameters selected via a grid search. We investigate the convergence on both convex problems and non-convex problems. We use a stochastic mini batch of size 128 and start all algorithms from the same initialization, $\boldsymbol{\theta}$ normally distributed with variance 0.1. For ADAM and AMSGrad we define $\beta_{1,t}$ as the gradient momentum parameter, β_2 as the equivalent of β from (2) in PA-SGD and α_t as the step size. We set $\beta = 0.999$, $\beta_2 = 0.999$, $\lambda = 1 - 10^{-8}$, and $\varepsilon = 10^{-8}$ for all experiments, similarly to ADAM.

For convex cost function, we present, in Fig. 1, a least squares regression directly with respect to the labels and a logistic regression experiment to classify digit 5. The step size decay used is $c = 0.5$. The convergence rate PA-SGD is better compared the other tested methods, reaching within $3\% - 4\%$ of the optimal LS solution in around 20 epochs. It shows the best performance for a larger acceleration parameters $\mu_0 = 0.99$ and $\alpha_0 = 10^{-3}$. The logistic regression simulations also show a good behavior, PA-SGD having a similar cost function value at 50 epochs as ADAM at 150. We note that the convergence plots of ADAM and AMSGrad almost overlap. We observed that μ_0 moderates the convergence rate together with the step size. For a smaller step size, a larger acceleration is beneficial, while for larger step sizes this becomes detrimental.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

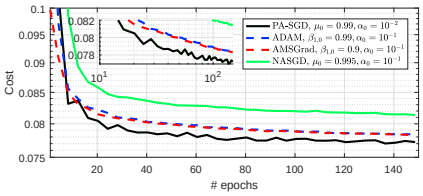

Fig. 1: Convex cost function: evolution for the best performing parameter configuration for (left) the least squares cost (right) the logistic regression cost. PA-SGD converges faster and achieves an almost optimal cost in $10-20$ epochs.

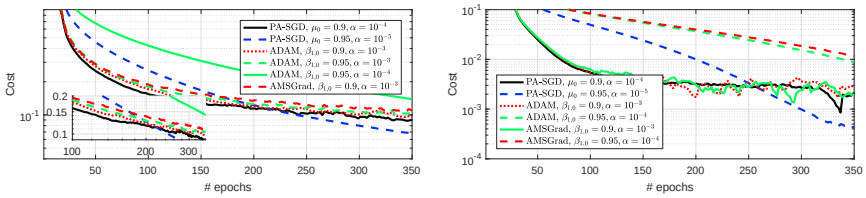

Fig. 2: Neural network training: (left) evolution of the mean squared error cost; (right) evolution of the multinomial logistic regression cost. For low value μ_0, PA-SGD is similar to ADAM and AMSGrad. For larger μ_0, PA-SGD has a faster final convergence albeit with slower start.

We also investigate the performance for neural network (NN) training. Here, we use constant step sizes α. For PA-SGD, the stability parameter s is set to 0. For a first experiment, we train a NN with, two, 32 neuron hidden layers, with tanh activation, to directly predict the MNIST numerical label. We use the mean squared error (MSE) as a cost function. The second experiment uses a similar NN with, two, 32 neuron hidden layers, with ReLu activation, resulting in a sub-gradient based optimization. We solve a softmax multinomial logistic regression with respect to the 10 digits. The convergence behavior is presented in Fig. 2. For the first experiment our method compares favorably with ADAM and AMSGrad. For the classification task, PA-SGD requires a lower step size and a larger acceleration to outperform the other methods.

Overall, a good practical range for the configuration parameters for PA-SGD is $\alpha \in [10^{-6}, 10^{-4}]$ and $\mu_0 \in [0.9, 0.99]$. We observed that, for NN training, a lower step size works best together with a larger acceleration parameter μ_0.

5 Conclusions

We proposed a preconditioned accelerated stochastic gradient method that combines Nesterov's accelerated gradient descent with a class of diagonal preconditioners. We analyzed the convergence for the minimization of convex cost functions and showcased empirically the behavior for convex and non-convex optimization tasks. The proposed method compares favorably with current stochastic optimization methods in terms of convergence speed while maintaining a low computational complexity, making it well suited for fast NN training.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

References

[1] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

[2] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[3] Y. Nesterov. A method of solving a convex programming problem with convergence rate $\mathcal{O}(\frac{1}{k^2})$. *Soviet Mathematics Doklady*, 1983.

[4] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Machine Learn. Res.*, 12(Jul):2121–2159, 2011.

[5] S.J. Reddi, S Kale, and S. Kumar. On the convergence of ADAM and beyond. In *ICLR*, 2018.

[6] T. Dozat. Incorporating Nesterov momentum into ADAM. In *ICLR*, 2016.

[7] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.

A Sketch for the convergence proof

Lemma 2. *Given a convex function $f : \mathbb{R}^d \to \mathbb{R}$, then for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ we have $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\dagger (\boldsymbol{y} - \boldsymbol{x})$.*

Property 3. *Under assumptions* (A1), (A2) *and* (A4)*, and for a given iteration t and index i of the accelerated gradient, $|a_{t+1,i}|$ is bounded from above as*

$$|a_{t+1,i}| \leq \alpha_0 C_{pg} \left(t^{-c} + \frac{\lambda^t}{1-\mu_0} \right). \tag{5}$$

Proof. From (1) we can expand $|a_{t+1,i}|$ such that

$$\begin{aligned} &|a_{t+1,i}| = |-\alpha_t p_{t,i} g_{t,i} - \textstyle\sum_{j=1}^{t-1} \alpha_j p_{j,i} g_{j,i} \prod_{k=j+1}^{t} \mu_k| \\ &\leq^{(A1)} \alpha_t p_{t,i} |g_{t,i}| + \textstyle\sum_{j=1}^{t-1} \alpha_j p_{j,i} |g_{j,i}| \prod_{k=j+1}^{t} \mu_k \leq^{(A1)} \alpha_t C_{pg} + \\ &\textstyle\sum_{j=1}^{t-1} \alpha_j C_{pg} \prod_{k=j+1}^{t} \mu_k \leq^{(A2)} \alpha_0 C_{pg} \left(t^{-c} + \sum_{j=1}^{t-1} j^{-c} \prod_{k=j+1}^{t} \mu_0 \lambda^k \right) \\ &\leq \alpha_0 C_{pg} \left(t^{-c} + \textstyle\sum_{j=1}^{t-1} j^{-c} \mu_0^{t-j} \lambda^t \right) \leq \alpha_0 C_{pg} \left(t^{-c} + \lambda^t \sum_{j=1}^{t-1} \mu_0^j \right). \end{aligned}$$

We have used $\prod_{k=j+1}^{t} \lambda^k \leq \lambda^t$ since $0 < \lambda < 1$ and $j^{-c} \leq 1$. Replacing the geometric progression with its upper bound we arrive at (5). ■

Property 4. *Under assumptions* (A1-3) *and for any given iteration t and index i the quantity $g_{t,i}\,(\theta_{t,i} - \theta_i^*)$ is bounded from above as*

$$\begin{aligned} g_{t,i}\,(\theta_{t,i} - \theta_i^*) \leq \tfrac{C_p}{2\alpha_0} \Big[\alpha_0^2 C_{pg}^2 \left(1 + 2\mu_0\right)^2 \left(t^{-c} + 2\tfrac{\lambda^t}{1-\mu_0} + \tfrac{\lambda^{2t} t^c}{(1-\mu_0)^2} \right) + \\ 2\mu_0^2 \alpha_0 C_\theta C_{pg} \left(\lambda^{2t} + \tfrac{\lambda^{3t} t^c}{1-\mu_0} \right) \Big] + \tfrac{1}{2\alpha_0 p_{t,i}} \Big[t^c \left(\theta_{t,i} - \theta_i^*\right)^2 - t^c \left(\theta_{t+1,i} - \theta_i^*\right)^2 \Big]. \end{aligned} \tag{6}$$

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

Proof. From (1) we have $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mu_t\mu_{t+1}\boldsymbol{a}_t - (1+\mu_{t+1})\,\alpha_t \boldsymbol{P}_t\boldsymbol{g}_t$. By expressing each component i, subtracting the ideal solution $\boldsymbol{\theta}^*$ and squaring we get

$$\begin{aligned}(\theta_{t+1,i}-\theta_i^*)^2 =& (\theta_{t,i}-\theta_i^*)^2 + \left(-\mu_t a_{t,i} + (1+\mu_{t+1})\,a_{t+1,i}\right)^2 + \\ & 2\left(\mu_t\mu_{t+1}a_{t,i} - (1+\mu_{t+1})\,\alpha_t p_{t,i} g_{t,i}\right)(\theta_{t,i}-\theta_i^*)\,.\end{aligned}$$

Additionally, we use (1) to replace $\mu_t\mu_{t+1}\boldsymbol{a}_t - (1+\mu_{t+1})\,\alpha_t\boldsymbol{P}_t\boldsymbol{g}_t$ by $-\mu_t\boldsymbol{a}_t + (1+\mu_{t+1})\,\boldsymbol{a}_{t+1}$. Rewriting the equality to express $g_{t,i}\,(\theta_{t,i}-\theta^*)$ we arrive at

$$\begin{aligned}g_{t,i}\,(\theta_{t,i}-\theta_i^*) &\leq \tfrac{1}{2(1+\mu_{t+1})\alpha_t p_{t,i}}\Big[\,(\theta_{t,i}-\theta_i^*)^2 - (\theta_{t+1,i}-\theta_i^*)^2 + \\ &\quad \left(\mu_t|a_{t,i}| + (1+\mu_{t+1})\,|a_{t+1,i}|\right)^2 + 2\mu_t\mu_{t+1}|a_{t,i}|\,|\,(\theta_{t,i}-\theta_i^*)\,|\Big] \\ &\leq^{(A3)} \tfrac{1}{2(1+\mu_{t+1})\alpha_t p_{t,i}}\Big[\,(\theta_{t,i}-\theta_i^*)^2 - (\theta_{t+1,i}-\theta_i^*)^2 + 2\mu_t\mu_{t+1}C_\theta|a_{t,i}| + \\ &\quad \left(\mu_t|a_{t,i}| + (1+\mu_{t+1})\,|a_{t+1,i}|\right)^2\Big] \leq^{(5),(A2)} \tfrac{1}{2(1+\mu_{t+1})\alpha_t p_{t,i}}\Big[\,(\theta_{t,i}-\theta_i^*)^2 - \\ &\quad (\theta_{t+1,i}-\theta_i^*)^2 + 2\mu_0^2\alpha_0 C_\theta C_{pg}\lambda^t\lambda^{t+1}\left(t^{-c} + \tfrac{\lambda^t}{1-\mu_0}\right) + \\ &\quad \alpha_0^2 C_{pg}^2\left(\mu_0\lambda^t\left(t^{-c} + \tfrac{\lambda^t}{1-\mu_0}\right) + \left(1+\mu_0\lambda^{t+1}\right)\left((t+1)^{-c} + \tfrac{\lambda^{t+1}}{1-\mu_0}\right)\right)^2\Big] \\ &\leq^{(A1,2)} \tfrac{t^c}{2(1+\mu_0\lambda^{t+1})\alpha_0 p_{t,i}}\Big[\,(\theta_{t,i}-\theta_i^*)^2 - (\theta_{t+1,i}-\theta_i^*)^2 + \\ &\quad \alpha_0^2 C_{pg}^2(1+2\mu_0\lambda^t)^2\left(t^{-c} + \tfrac{\lambda^t}{1-\mu_0}\right)^2 + 2\mu_0^2\alpha_0 C_\theta C_{pg}\lambda^t\lambda^{t+1}\left(t^{-c} + \tfrac{\lambda^t}{1-\mu_0}\right)\Big]\end{aligned}$$

which, under (A1) and (A2), further reduces to (6). ∎

Theorem 1. *Proof.* We construct an upper bound for (3) using Lemma 2. For each index i, summed for $t = 1:T$ in $f_t(\boldsymbol{\theta}_t) - f_t(\boldsymbol{\theta}^*) \leq \boldsymbol{g}_t^\dagger\,(\boldsymbol{\theta}_t - \boldsymbol{\theta}^*)$ we have

$$\begin{aligned}R(T) &\leq^{(6)} \textstyle\sum_{i=1}^d \sum_{t=1}^T \tfrac{1}{2\alpha_0 p_{t,i}}\Big[t^c\,(\theta_{t,i}-\theta_i^*)^2 - t^c\,(\theta_{t+1,i}-\theta_i^*)^2\Big] + \\ &\quad \tfrac{C_p}{2\alpha_0}\Big[\alpha_0^2 C_{pg}^2\,(1+2\mu_0)^2\left(\tfrac{1}{t^c} + 2\tfrac{\lambda^t}{1-\mu_0} + \tfrac{\lambda^{2t}t^c}{(1-\mu_0)^2}\right) + 2\mu_0^2\alpha_0 C_\theta C_{pg}\left(\lambda^{2t} + \tfrac{\lambda^{3t}t^c}{1-\mu_0}\right)\Big] \\ &\leq \tfrac{1}{2\alpha_0}\textstyle\sum_{i=1}^d\Big[\tfrac{1}{p_{1,i}}\,(\theta_{1,i}-\theta_i^*)^2 + \sum_{t=2}^T\left(\left(\tfrac{t^c}{p_{t,i}} - \tfrac{(t-1)^c}{p_{t-1,i}}\right)(\theta_{t,i}-\theta_i^*)^2\right) - \\ &\quad \tfrac{T^c}{p_{T,i}}\,(\theta_{T+1,i}-\theta_i^*)^2\Big] + \tfrac{C_p}{2\alpha_0}\Big[\alpha_0^2 C_{pg}^2\,(1+2\mu_0)^2\textstyle\sum_{t=1}^T\left(\tfrac{1}{t^c} + 2\tfrac{\lambda^t}{1-\mu_0} + \tfrac{\lambda^{2t}t^c}{(1-\mu_0)^2}\right) + \\ &\quad 2\mu_0^2\alpha_0 C_\theta C_{pg}\textstyle\sum_{t=1}^T\left(\lambda^{2t} + \tfrac{\lambda^{3t}t^c}{1-\mu_0}\right)\Big].\end{aligned}$$

Relying on (A3) and (A4) written as $\frac{t^c}{p_{t,i}} - \frac{(t-1)^c}{p_{t-1,i}} > 0$ for $\alpha_t = \alpha_0\frac{1}{t^c}$, we have

$$\begin{aligned}R(T) &\leq^{(A2)} \tfrac{1}{2\alpha_0}\textstyle\sum_{i=1}^d\Big[C_\theta^2 + C_\theta^2\sum_{t=2}^T\left(\tfrac{t^c}{p_{t,i}} - \tfrac{(t-1)^c}{p_{t-1,i}}\right) - \tfrac{T^c}{p_{t,i}}\,(\theta_{T+1,i}-\theta_i^*)^2\Big] \\ &\quad + \tfrac{C_p}{2\alpha_0}\Big[\alpha_0^2 C_{pg}^2\,(1+2\mu_0)^2\textstyle\sum_{t=1}^T\left(t^{-c} + 2\tfrac{\lambda^t}{1-\mu_0} + \tfrac{\lambda^{2t}t^c}{(1-\mu_0)^2}\right) + \\ &\quad 2\mu_0^2\alpha_0 C_\theta C_{pg}\textstyle\sum_{t=1}^T\left(\lambda^{2t} + \tfrac{\lambda^{3t}t^c}{1-\mu_0}\right)\Big] \\ &\leq^{(A4)} \tfrac{C_p}{2\alpha_0}\textstyle\sum_{i=1}^d\Big[T^c C_\theta^2 + \alpha_0^2 C_{pg}^2\,(1+2\mu_0)^2\sum_{t=1}^T\left(t^{-c} + 2\tfrac{\lambda^t}{1-\mu_0} + \tfrac{\lambda^{2t}t^c}{(1-\mu_0)^2}\right) + \\ &\quad 2\mu_0^2\alpha_0 C_\theta C_{pg}\textstyle\sum_{t=1}^T\left(\lambda^{2t} + \tfrac{\lambda^{3t}t^c}{1-\mu_0}\right)\Big].\end{aligned}$$

The resulting inequality can be bounded using the upper bounds for the geometric series $\sum_{t=1}^T\lambda^t \leq \frac{1}{1-\lambda}$ and $\sum_{t=1}^T\lambda^{2t} \leq \frac{1}{1-\lambda^2}$ and for the arithmetic-geometric series $\sum_{t=1}^T\lambda^{2t}t^c \leq \frac{1}{(1-\lambda^2)^2}$ and $\sum_{t=1}^T\lambda^{3t}t^c \leq \frac{1}{(1-\lambda^3)^2}$. This produces (4). ∎