Demonstrating How Exponential Moving Average Based Gradient Algorithms Fail in Certain Convex Settings

APPM 5630

Logan Barnhart, Tyler Jensen, and Eappen Nelluvelil

February 4, 2024

Department of Applied Mathematics

University of Colorado **Boulder**

Outline

1. Overview of SGD and Adaptive Gradient Descent Methods
2. The non-convergence of Adam in certain convex settings
3. Numerical Results and Conclusion

Overview of SGD and Adaptive Gradient Descent Methods

Stochastic gradient descent (SGD)

Gradient descent is the typical optimization algorithm used in ML settings, largely due to its simplicity.

Given an initial iterate $\mathbf{x}_1 \in \mathbb{R}^n$ and a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, for $i = 1, 2, \ldots$, we obtain a sequence of iterates $\mathbf{x}_1, \mathbf{x}_2, \ldots$

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \gamma \nabla f\left(\mathbf{x}_i\right),$$

where $\gamma > 0$ is the learning rate.

This can be improved upon by careful choice of our initial iterate, using a line search to use a different learning rate at each iteration, γ_i, etc.

Stochastic gradient descent (SGD) (cont.)

If our function f is of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}),$$

where each f_i is differentiable, then $\nabla f(\mathbf{x}) = \sum_{i=1}^{n} \nabla f_i(\mathbf{x})$

Gradient descent is an acceptable choice if calculating or evaluating each ∇f_i is inexpensive, but this might not always be the case

Stochastic gradient descent (SGD) (cont.)

Stochastic gradient descent (SGD) takes an alternate approach: instead of taking the descent direction to be $\nabla f(\mathbf{x}_i)$, it takes it to be $\nabla f_k(\mathbf{x}_i)$, $k \in [n]$

The typical SGD iteration is of the following form:

Randomly shuffle the f_k's

For $k = 1, 2, \ldots, n$:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \gamma \nabla f_k(\mathbf{x}_i)$$

A popular variant of SGD is "mini-batch" SGD, which uses $\sum_{k \in K} \nabla f_k(\mathbf{x}_i)$, where $K \subset [n]$ as the descent direction (empirically leads to better convergence rates since we're using more information about f)

A general framework for adaptive gradient methods

In practice, SGD is used as an optimizer with several modifications, e.g.,

- using momentum, i.e., use previous iterate and gradient to make new iterate (related to Nesterov acceleration as discussed in class)
- averaging over past parameters (instead of taking the most recent iterate as our final guess)
- modifying the learning rate for each parameter based on corresponding gradient entry sizes

These fall under the umbrella of adaptive gradient methods, with three of the most popular methods being AdaGrad, RMSProp, and Adam, and can be encapsulated by the framework given by Reddi et al. (Reddi, Kale, and Kumar 2019)

A general framework for adaptive gradient methods (cont.)

Input: $\mathbf{x}_1 \in \mathcal{F}$, where $\mathcal{F} \subset \mathbb{R}^d$ is feasible set; $\{\alpha_t > 0\}_{t=1}^T$, sequence of step-sizes; $\{\phi_t, \psi_t\}_{t=1}^T$, sequence of "averaging" functions

Output: $\mathbf{x}_{T+1}$, guess for the optimal parameter $\mathbf{x}$

begin

for $t = 1$ *to* T **do**

$\mathbf{g}_t \longleftarrow \nabla f_t(\mathbf{x}_t)$

$\mathbf{m}_t \longleftarrow \phi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t)$ // $\phi_t : \mathcal{F}^t \to \mathbb{R}^d$, 1^{st} moment estimate

$V_t \longleftarrow \psi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t)$ // $\psi_t : \mathcal{F}^t \to \mathcal{S}_+^d$

$\widehat{\mathbf{x}}_{t+1} \longleftarrow \mathbf{x}_t - \alpha_t (V_t)^{-\frac{1}{2}} \mathbf{m}_t$ // Update iterate with learning-rate $\alpha_t (V_t)^{-\frac{1}{2}}$, 2^{nd} moment estimate $\mathbf{v}_t = \alpha_t (V)_t^{-\frac{1}{2}} \mathbf{m}_t$

$\mathbf{x}_{t+1} \longleftarrow \prod_{\mathcal{F}, (V_t)^{\frac{1}{2}}} (\widehat{\mathbf{x}}_{t+1})$ // Project new iterate onto $\mathcal{F}$ $(\mathcal{F} = \mathbb{R}^d)$

A general framework for adaptive gradient methods (cont.)

The paper analyzes only diagonal variants of adaptive methods, i.e., $V_t = \operatorname{diag}(\mathbf{v}_t)$, which include the adaptive gradient methods mentioned earlier.

Under the earlier framework, Adam (Kingma and Ba 2014) is given by the following choices of averaging functions $\{\phi_t, \psi_t\}_{t=1}^{T}$:

$$\phi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = (1-\beta_1)\sum_{i=1}^{t}\beta_1^{t-i}\mathbf{g}_i \quad [\mathbf{m}_t = \beta_1\mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t]$$

$$\psi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = (1-\beta_2)\operatorname{diag}\left(\sum_{i=1}^{t}\beta_2^{t-i}\mathbf{g}_i^t\right) \quad \left[\mathbf{v}_t = \beta_2\mathbf{v}_{t-1} + (1-\beta_2)\mathbf{g}_t^2\right],$$

where $\beta_1, \beta_2 \in [0, 1)$ (typically, $\beta_1 = 0.9$ and $\beta_2 = 0.999$).

The non-convergence of Adam in certain convex settings

Adam diverges in certain convex settings

The authors focus on Adam due to its high popularity in the deep learning community (since publication in 2015, more than 142k citations)

However, Reddi et al. note that exponential moving average methods, e.g., RMSProp, Adam, their variants, etc., can and do fail in certain convex settings

The issue lies with the following quantity,

$$\Gamma_{t+1} = \left(\frac{1}{\alpha_{t+1}} \left(V_{t+1} \right)^{\frac{1}{2}} - \frac{1}{\alpha_t} \left(V_t \right)^{\frac{1}{2}} \right),$$

which is essentially the successive change in the inverse of the learning rate of the adaptive method (Reddi, Kale, and Kumar 2019)

Adam diverges in certain convex settings (cont.)

When using SGD and AdaGrad, $\Gamma_t \succeq 0$ for all $t \in [T]$ (this follows from their choice of averaging functions $\{\phi_t, \psi_t\}$), which implies "non-increasing" learning rates

For exponential moving average methods, such as Adam, Γ_t can be indefinite for $t \in [T]$, which potentially leads to non-convergence (Reddi, Kale, and Kumar 2019)

Example of Online Convex Problem on which Adam Diverges

Let $f_t : [-1, 1] \to \mathbb{R}$,

$$f_t(x) = \begin{cases} Cx & t \mod 3 = 1, \\ -x & \text{else.} \end{cases}$$

Note that the sequence of functions $\{f_t\}$ achieves a global minimum of -1 when $x = -1$.

It can be shown that if $x_0 > 0$ when using Adam, then $x_t > 0 \forall t$.

Notably, if $x_0 = 1$ then

$$x_{3t} = 1$$

$$x_{3t+1}, x_{3t+2} > 0$$

(Reddi, Kale, and Kumar 2019)

Example of Stochastic Convex Problem on Which Adam Diverges

Let $f_t : [-1, 1] \to \mathbb{R}$,

$$f_t(x) = \begin{cases} Cx & \text{with prob. } p = \frac{1+\delta}{C+1} \\ -x & \text{with prob. } 1 - p \end{cases}$$

The expected function from this distribution is thus $F(x) = \delta x$. Thus the optimal point is $x^\star = -1$

When using Adam, one can choose C (as a function of β_1, β_2 and δ) large enough such that $\mathbb{E}[x_t - x_{t-1}] \geq 0$. Thus we expect x_t to increase as $t \to \infty$ and thus move away from $x^\star = -1$

(Reddi, Kale, and Kumar 2019)

Potential Fixes to Successfully Solve the Earlier Convex Problems

Popular implementations of Adam use the following modification to the update rule to $\widehat{\mathbf{x}}_{t+1}$ to avoid the earlier non-convergence issues:

$$\widehat{\mathbf{x}}_{t+1} = \mathbf{x}_t - \alpha_t \left(V_t + \varepsilon I\right)^{-\frac{1}{2}} \mathbf{m}_t,$$

where $\varepsilon > 0$ and is chosen carefully

However, even with this update it can be shown that the optimizer does not converge in some online convex settings.

It can also be shown that large values of β_2 do not counteract this behavior either.

Although these problems were produced to demonstrate Adam's lack of non-convergence, the possibility of using problem-dependent ε, β_1, and β_2 to avoid non-convergence exists, which is undesirable (Reddi, Kale, and Kumar 2019)

AMSgrad Successfully Solves the Earlier Convex Problems

Reddi et al.'s modification to Adam, named AMSGrad, which they prove avoids Adam's non-convergence issues on the earlier problems, is given below:

$$
\begin{aligned}
\widehat{\mathbf{v}}_0 &= \mathbf{0}, \\
\widehat{\mathbf{v}}_t &= \max\left\{\widehat{\mathbf{v}}_{t-1}, \mathbf{v}_t\right\}, \\
\widehat{V}_t &= \operatorname{diag}\left(\widehat{\mathbf{v}}_t\right), \\
\mathbf{x}_{t+1} &= \prod_{\mathcal{F}, \widehat{V}_t^{\frac{1}{2}}} \left(\mathbf{x}_t - \alpha_t \left(\widehat{V}_t\right)^{-\frac{1}{2}} \mathbf{m}_t\right).
\end{aligned}
$$

The big difference between AMSGrad and Adam is that AMSGrad takes the **maximum** of all $\mathbf{v}_t$'s, and uses this to normalize the running average of the gradients, which leads to $\Gamma_T \succeq 0$

Numerical Results and Conclusion

Experiment 1: Synthetic Examples of Adam Diverging

Online Convex Synthetic Example

$$f_t : [-1, 1] \to \mathbb{R}$$

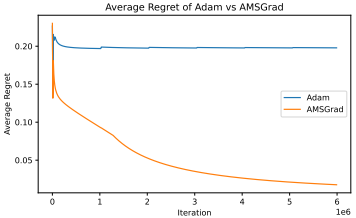
$$f_t(x) = \begin{cases} 1010x & t \mod 101 = 1, \\ -10x & \text{else.} \end{cases}$$

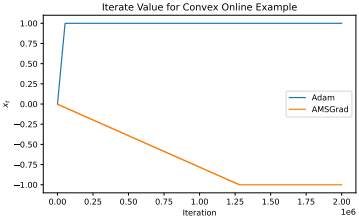
Stochastic Convex Synthetic Example

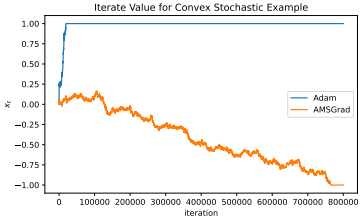
$$f_t : [-1, 1] \to \mathbb{R}$$

$$f_t(x) = \begin{cases} 1010x & \text{with prob. } p = 0.1 \\ -10x & \text{else.} \end{cases}$$

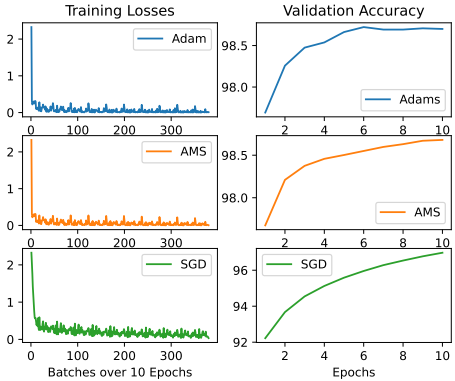
Experiment 1: Synthetic Examples of Adam Diverging





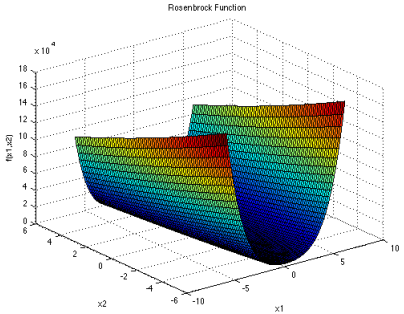


Experiment 2: Performance on Classical Machine Learning Task



Model Performance on MNIST Digits Dataset

Experiment 3: Testing Optimizer Robustness on Rosenbrock Function



Rosenbrock Function

$$f(\mathbf{x}) = \sum_{i=1}^{d-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$$

Experiment 3: Testing Optimizer Robustness on the Rosenbrock Function (cont.)

We applied regular gradient descent, Adam, and AMSGrad to find the minimum of the Rosenbrock function in the case where $d = 100$, with the following results:

Optimizer	2-norm of difference of the weights	∞-norm of the difference of the weights
GD	1.4430×10^{-13}	1.25011×10^{-13}
Adam	1.9033×10^{-6}	5.1294×10^{-6}
AMSGrad	8.2960	1.7580

Number of iterations $= 10^6$, learning rate $= 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$

Experiment 4: A Study on Ground-Truth Discovery

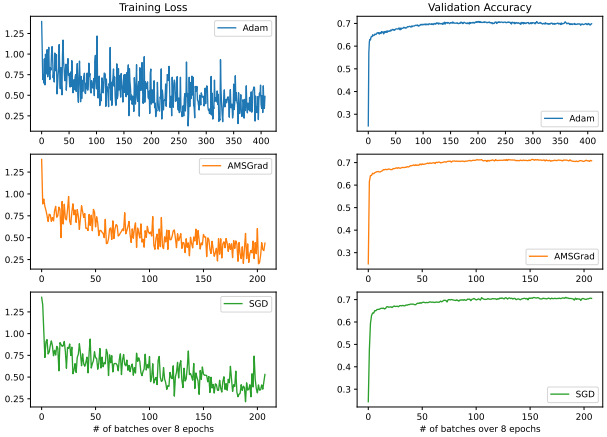
Context:

- We first initialize a neural net with randomly generated weights. This model behaves as our ground-truth or 'oracle'
- We then generate data points where each feature is i.i.d. $\sim \mathcal{N}(0, 1)$
- Passing these data pts through the 'oracle' produces the labels for later models to train on

The goal was twofold:

- Evaluate performance
- Evaluate weight recovery (if any is present)

Experiment 4: A Study on Ground-Truth Discovery - Results



Model	Inf-Norm of Difference in Weights
Adam	1.387
AMSGrad	.605
SGD	.958

Conclusions

Reddi et al.'s paper proves that Adam, and other adaptive gradient methods derived from it, do not work in all online and stochastic convex problems

However, the numerical results that Reddi et al. provide in their paper are not easily replicable, and our numerical results indicate that AMSGrad offers marginal convergence and performance benefits in some cases, but otherwise, performs as well as Adam

More papers have come out in recent years that address Adam's lack of convergence in such convex settings, including one that provides sufficient conditions under which RMSProp and Adam will converge in large-scale, non-convex settings (Zou et al. 2019), but such results need to be verified

Thank You

Questions?

References

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Reddi, Sashank J, Satyen Kale, and Sanjiv Kumar (2019). "On the convergence of adam and beyond". In: *arXiv preprint arXiv:1904.09237*.

Zou, Fangyu et al. (2019). "A sufficient condition for convergences of adam and rmsprop". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11127–11135.