Study of ADAM algorithm

Bharat Mohan
bmohan@iitg.ac.in

June 15, 2020

1 Motivation

Gradient Descent is applicable in the scenarios where the function is easily differentiable with respect to the parameters used in the network. It is easy to minimize continuous function than minimizing discrete functions. The weight update is performed after one epoch, where one epoch represents running through an entire dataset. This technique produces satisfactory results but it deteriorates if the training dataset size becomes large and does not converge well. It also may not lead to global minimum in case of the existence of multiple local minima.

Stochastic gradient descent overcomes this drawback by randomly selecting data samples and updating the parameters based on the cost function. Additionally, it converges faster than regular gradient descent and saves memory by not accumulating the intermediate weights. Adaptive Moment Estimation (ADAM) facilitates computation of learning rates for each parameter using first and second moment of gradient.Where SGD adds randomness to gradient descent, momentum accelerates the convergence, and adaptive gradient, as its name sounds, adapts to different learning rate for different parameters, Adam assembles the advantages of these algorithms and make it one.

2 Introduction

Adaptive Moment Estimation (Adam) is a method that computes adaptive learning rates for each parameter. In addition to storing an exponentially decaying average of past squared gradients (s) like Adadelta and RMSprop, Adam also keeps an exponentially decaying average of past gradients (v), similar to momentum. Whereas momentum can be seen as a ball running down a slope, Adam behaves like a heavy ball with friction, which thus prefers flat minima in the error surface. Being computationally efficient, ADAM requires less memory,outperforms on large datasets and typically requires zero (or little) tuning. It requires v_t,s_t,t to be initialized to 0, where v_t corresponds to 1st moment vector i.e. mean,s_t corresponds to 2nd moment vector i.e. uncentered variance and t represents timestep.
While considering $F(\theta)$ to be the stochastic objective function with parameters θ, proposed values of parameters in ADAM, are as follows:
$\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$.

3 Algorithm

Step 1 : while θ_t do not converge
do{
Step 2 : Calculate Gradient $g_t = \frac{\partial F(x,\theta)}{\partial \theta}$
Step 3 : Calculate $v_t = \beta_1 * v_{t-1} + (1-\beta_1) * g_t$
Step 4 : Calculate $s_t = \beta_2 * s_{t-1} + (1-\beta_2) * g_t$
Step 5: Calculate $v_t^{'} = \frac{v_t}{(1-\beta_1{}^t)}$
Step 6: Calculate $s_t^{'} = \frac{s_t}{(1-\beta_2{}^t)}$
Step 7: Update $\theta_t = \theta_{t-1} - \alpha * \frac{v_t^{'}}{\sqrt{s_t^{'}} + \epsilon} * g_t$
}
Step 8: return θ_t

4 Objective functions used for study

The following objective functions are taken for studying the convergence of ADAM over other approaches.

1. Ackley Function

$$f(x,y) = -20\exp\Big(-0.2\sqrt{0.5\big(x^2+y^2\big)}\Big) - \exp\Big(0.5\big(\cos 2\pi x + \cos 2\pi y\big)\Big) + \exp\big(1\big) + 20$$

2. Log Beale Function

$$f(x,y) = log((1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2 + 1)$$

3. Booth Function

$$f(x,y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

4. Cross-in-Tray Function

$$f(x,y) = -0.0001\left(\left|\sin(x)\sin(y)\exp\left(\left|100 - \frac{\sqrt{x^2+y^2}}{\pi}\right|\right)\right| + 1\right)^{0.1}$$

5. Easom Function

$$f(x,y) = -\cos(x)\cos(y)\exp\Big(-(x-\pi)^2 - (y-\pi)^2\Big)$$

6. Goldstein-Price Function

$$f(x,y) = \Big[1 + (x+y+1)^2\big(19 - 14x + 3x^2 - 14y + 6xy + 3y^2\big)\Big]$$

$$\times\Big[30 + (2x-3y)^2\big(18 - 32x + 12x^2 + 45y - 36xy + 27y^2\big)\Big]$$

7. Himmelblau Function

$$f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$

8. Holder Table Function

$$f(x,y) = -\left|\sin(x)\cos(y)\exp\left(\left|1 - \frac{\sqrt{x^2+y^2}}{\pi}\right|\right)\right|$$

9. Matyas Function

$$f(x,y) = 0.26(x^2 + y^2) - 0.48xy$$

10. Three-Hump Camel Function

$$f(x,y) = 2x^2 - 1.05x^4 + \frac{x^6}{6} + xy + y^2$$

11. Eggholder Function

$$f(x,y) = -(y+47)\sin\left(\sqrt{\left|y + \frac{x}{2} + 47\right|}\right) - x\sin\left(\sqrt{\left|x - (y+47)\right|}\right)$$

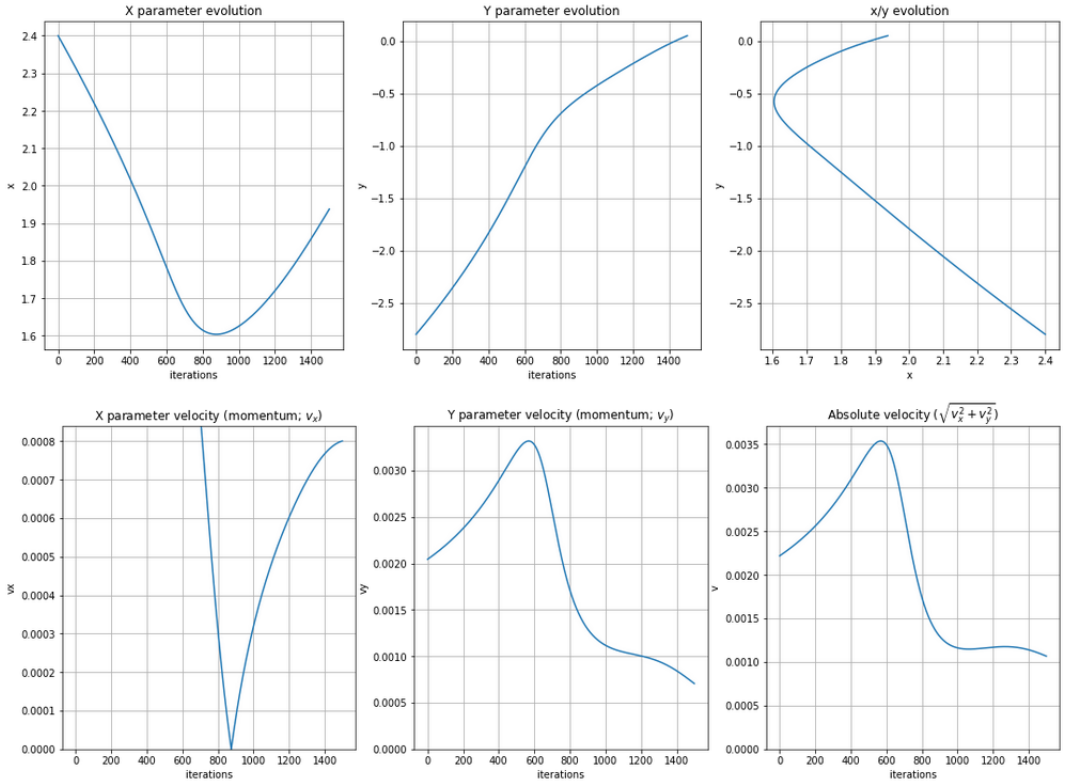
5 Analysis

Taking the example of log-Beale function to demonstrate the relevance of Adam algorithm,in comparison to SGD. The global minima for the same is at (3,0.5) which has a value of 0. The initial value is (2.4,-2.8). The evolution chart for SGD of x, y ,x/y values, and the velocity of the same are depicted in figure 1. The algorithm follows the negative gradient direction, which for the first steps is very strong due to the slope of the space. After that, it continues following the negative gradient but now the gradient signal is lighter (i.e. the space is much less inclined). Because of this algorithm follows the negative gradient direction with a step proportional to its magnitude, it presents a very slow convergence when it is small slopes.

Regarding the dynamic evolution, it clearly shows that this algorithm's velocity depends directly on the magnitude of the gradient. If the gradient is small, the algorithm advances really slowly; this is the problem that arises with this objective function (in which there are slopes of different orders of magnitude). That is the reason why the absolute velocity decays fast to a very small value after 900 iterations. The XY plot shows the trajectory followed by the algorithm.

The evolution chart for ADAM of x, y ,x/y values, and the velocity of the same are depicted in figure 2. It indicates a faster convergence compared to SGD even though the velocity of gradient decreases initially, it gains traction fast enough to converge in next 100 iterations.

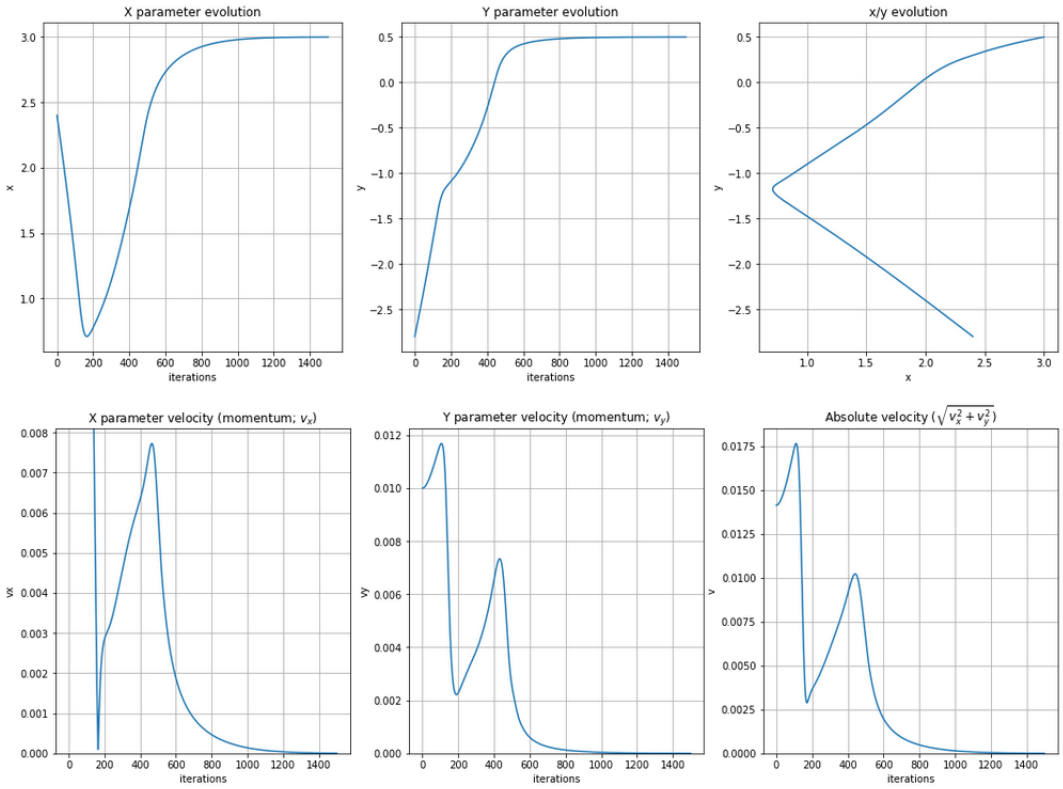
Figure 1: The evolution chart for SGD



6 Takeaways

- Adam combines Adagrad and RMSprop algorithms holding robustness and speed.
  1. Ability to handle very sparse gradients from AdaGrad.
  2. Ability to deal with non-stationary objectives from RMSprop.
- There is no need to perform a stepsize annealing because the first and second moments factor $\left(\frac{\mathbf{v_t}}{\sqrt{\mathbf{s_t}} + \epsilon}\right)$ does it automatically since the SNR value typically decreases closer to 0 towards an optimum.
- $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ by default. It is not usually necessary to change them.
- Adam's β_2 parameter can be increased when the data is very sparse (for example when working with bag-of-words data).
- AdaGrad is the same as Adam with $\beta_2 \to 1$, $\beta_1 = 0$ and annealing α.
- With convex and non-sparse objectives, SGD with Nesterov momentum performance is similar to Adam. Though in SGD the learning rate has to be manually picked.

Figure 2: The evolution chart for ADAM



- Functions like Easom which have steep ridges or valleys, convergence is still unattainable for gradient based methods like ADAM
- Plots reveal a smoother convergence for ADAM even though it isn't the fastest among all the techniques in all the cases

7 References

1. https://towardsdatascience.com/optimisation-algorithm-adaptive-moment-estimation-adam-92144d75e232
2. https://www.geeksforgeeks.org/adam-adaptive-moment-estimation-optimization-ml/
3. https://github.com/ivallesp/awesome-optimizers