Momentum and Adaptive Learning Rate

Ruixin Guo

Department of Computer Science
Kent State University

August 17, 2023

Contents

Contents

Momentum

Momentum and Adaptive Learning Rate are two main ways to accelerate gradient descent. The momentum based method include Heavy Ball Momentum and Nesterov's Accelerated Gradient Descent. The Adaptive Learning Rate based methods include AdaGrad, RMSProp, Adam and AMSGrad.

Heavy Ball Momentum is the first momentum based method, proposed by Boris Polyak in 1964 [3]. This method introduces a momentum term which serves as the inertia in Newton's Laws of Motion to gradient descent. We focus on the Heavy Ball Momentum in stochastic gradient descent.

Algorithm 1.1: Let Assumption $L_{\max}$-smooth hold. Let $x^0 \in \mathbb{R}^d$ and $m^{-1} = 0$. Let $\{t_k\}_{k\in\mathbb{N}} \subset (0, +\infty)$ be a sequence of step sizes, and let $\{\beta_k\}_{k\in\mathbb{N}} \subset [0, 1]$ be a sequence of momentum parameters. The Momentum algorithm defines a sequence $\{x^k\}_{k\in\mathbb{N}}$ satisfying for every $k \in \mathbb{N}$,

$$
\begin{aligned}
m^k &= \beta_k m^{k-1} + \nabla f_{i_k}(x^k) \\
x^{k+1} &= x^k - t_k m^k
\end{aligned}
$$

Heavy Ball

Algorithm 1.1 is the most basic momentum algorithm. It can be written in many ways. The following way is the original one in Polyak's paper [3].

Definition 1.2 (Heavy Ball): Let $\{\hat{t}_k\}_{k\in\mathbb{N}} \subset [0, +\infty]$ be another sequence of step sizes and $\{\hat{\beta}_k\}_{k\in\mathbb{N}} \subset [0, 1]$ be another sequence of momentum parameters. The Algorith 1.1 can also be written as

$$x^{k+1} = x^k - \hat{t}_k \nabla f_{i_k}(x^k) + \hat{\beta}_k (x^k - x^{k-1})$$

Lemma 1.3: The Algorithm 1.1 and Definition 1.2 (Heavy Ball) are equivalent by taking $\hat{t}_k = t_k$ and $\hat{\beta}_k = \frac{t_k}{t_{k-1}} \beta_k$.

Proof: In Algorithm 1.1 we have $m^k = \beta_k m^{k-1} + \nabla f_{i_k}(x^k)$ and $x^{k+1} = x^k - t_k m^k$. Plugging the first equation in the second equation, we get

$$x^{k+1} = x^k - t_k \nabla f_{i_k}(x^k) - t_k \beta_k m^{k-1}$$

Since $m^{k-1} = -\frac{x^k - x^{k-1}}{t_{k-1}}$, we have

$$x^{k+1} = x^k - t_k \nabla f_{i_k}(x^k) + t_k \beta_k \frac{x^k - x^{k-1}}{t_{k-1}}$$

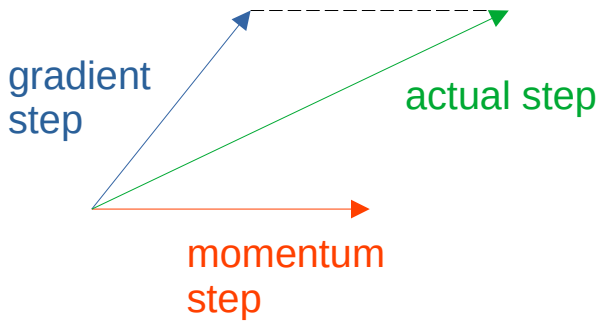
By taking $\hat{t}_k = t_k$ and $\hat{\beta}_k = \frac{t_k}{t_{k-1}} \beta_k$, the Algorithm 1.1 will be equivalent to Heavy Ball.

An Intuitive Explanation of Heavy Ball

Let's look at the Definition 1.2 (Heavy Ball):

$$x^{k+1} = x^k - \hat{t}_k \nabla f_{i_k}(x^k) + \hat{\beta}_k (x^k - x^{k-1})$$

The direction that x^k moves to x^{k+1} is a combination two vectors: the gradient (blue part) and the momentum (red part).



The momentum records the movement direction of the last iteration. When $\hat{\beta}_k = 0$, the Heavy Ball is reduced to SGD.

An Intuitive Explanation of Heavy Ball

Now let's consider what will happen if we add the momentum. Starting the iteration from 0:

$$
\begin{aligned}
x^1 &= x^0 - \hat{t}_0 \nabla f_{i_0}(x^0) \\
x^2 &= x^1 - \hat{t}_1 \nabla f_{i_1}(x^1) + \hat{\beta}_1 (x^1 - x^0) = x^1 - \hat{t}_1 \nabla f_{i_k}(x^1) - \hat{\beta}_1 \hat{t}_0 \nabla f_{i_k}(x^0) \\
x^3 &= x^2 - \hat{t}_2 \nabla f_{i_2}(x^2) + \hat{\beta}_2 (x^2 - x^1) \\
&= x^2 - \hat{t}_2 \nabla f_{i_2}(x^2) - \hat{\beta}_2 \hat{t}_1 \nabla f_{i_k}(x^1) - \hat{\beta}_2 \hat{\beta}_1 \hat{t}_0 \nabla f_{i_k}(x^0)
\end{aligned}
$$

Here you can see: the momentum term is a weighted sum of all past gradient descent movements. Since each $\hat{\beta}_k \in [0, 1]$, the older term will be assigned smaller weights. When all $\hat{\beta}_k$s are equal, the momentum term becomes an Exponential Moving Average (EMA)[1].

Therefore, if the gradient vectors of recent few iterations vary in direction or shorten in length, the momentum term will become small and make the movement slow down. Otherwise, the movement will be accelerated.

[1] https://en.wikipedia.org/wiki/Moving_average

Iterate Moving Average

Another way of writing momentum is Iterate Moving Average (IMA).

Lemma 1.4: The Algorithm 1.1 is equivalent to the following IMA algorithm: start from $z^{-1} = 0$ and iterate for $k \in \mathbb{N}$,

$$z^k = z^{k-1} - \eta_k \nabla f_{i_k}(x^k)$$
$$x^{k+1} = \frac{\lambda_{k+1}}{\lambda_{k+1}+1} x^k + \frac{1}{\lambda_{k+1}+1} z^k$$

By choosing any parameters (η_k, λ_k) satisfying

$$\lambda_k = (1+\lambda_{k+1})\frac{t_k}{t_{k-1}}\beta_k \text{ and } \eta_k = (1+\lambda_{k+1})t_k$$

Proof: Note that

$$x^{k+1} = \frac{\lambda_{k+1}}{\lambda_{k+1}+1} x^k + \frac{1}{\lambda_{k+1}+1} z^k \iff z^k = (1+\lambda_{k+1})x^{k+1} - \lambda_{k+1} x^k$$

Since by Heavy Ball,

$$x^{k+1} = x^k - \hat{t}_k \nabla f_{i_k}(x^k) + \hat{\beta}_k (x^k - x^{k-1})$$

We have

$$\begin{aligned} z^k &= (1+\lambda_{k+1}) x^{k+1} - \lambda_{k+1} x^k \\ &= (1+\lambda_{k+1})(x^k - \hat{t}_k \nabla f_{i_k}(x^k) + \hat{\beta}_k (x^k - x^{k-1})) - \lambda_{k+1} x^k \\ &= x^k - (1+\lambda_{k+1}) \hat{t}_k \nabla f_{i_k}(x^k) + (1+\lambda_{k+1}) \hat{\beta}_k (x^k - x^{k-1}) \end{aligned} \tag{1}$$

Since

$$z^{k-1} = (1+\lambda_k) x^k - \lambda_k x^{k-1} = x^k + \lambda_k (x^k - x^{k-1})$$

By letting $\lambda_k = (1+\lambda_{k+1})\hat{\beta}_k$ and $\eta_k = (1+\lambda_{k+1})\hat{t}_k$, Equation (1) can be written as

$$z^k = z^{k-1} - \eta_k \nabla f_{i_k}(x^k)$$

Since $\hat{\beta}_k = \frac{t_k}{t_{k-1}} \beta_k$ and $\hat{t}_k = t_k$, we have

$$\lambda_k = (1+\lambda_{k+1}) \frac{t_k}{t_{k-1}} \beta_k \text{ and } \eta_k = (1+\lambda_{k+1}) t_k$$

Convergence Analysis of Momentum

Theorem 1.5: Let Assumptions (Sum of $L_{\max}$) and (Sum of Convex) hold. Consider $\{x_k\}_{k\in\mathbb{N}}$ the iterates generated by the Algorithm 1.1 with step size and momentum parameters taken according to

$$t_k = \frac{2\eta}{k+3}, \quad \beta_k = \frac{k}{k+2}, \qquad \text{with } \eta \leq \frac{1}{4L_{\max}}$$

Then the iterates converge according to

$$\mathbb{E}[f(x^k) - \inf f] \leq \frac{\|x^0 - x^*\|^2}{\eta(k+1)} + 2\eta\sigma_f^*$$

Proof:
By IMA, we can write

$$\begin{aligned} z^k &= z^{k-1} - \eta_k \nabla f_{i_k}(x^k) \\ z^k &= x^{k+1} + \lambda_{k+1}(x^{k+1} - x^k) \end{aligned} \tag{2}$$

where

$$\lambda_k = (1+\lambda_{k+1})\frac{t_k}{t_{k-1}}\beta_k \text{ and } \eta_k = (1+\lambda_{k+1})t_k \tag{3}$$

Since $t_k = \frac{2\eta}{k+3}$ and $\beta_k = \frac{k}{k+2}$, we can make Equation (3) hold by taking

$$\lambda_k = \frac{k}{2}, \quad \eta_k = \eta$$

Note that this is not the only choice of (η_k, λ_k) to make IMA and Momentum be equivalent. We choose $\lambda_k = \frac{k}{2}, \eta_k = \eta$ because it is easy for analyzation.

Thus,

$$\begin{aligned}
\|z^k - x^*\|^2 &= \|z^{k-1} - x^* - \eta \nabla f_{i_k}(x^k)\|^2 \\
&= \|z^{k-1} - x^*\|^2 - 2\eta\langle \nabla f_{i_k}(x^k), z^{k-1} - x^*\rangle + \eta^2 \|\nabla f_{i_t}(x^k)\|^2 \\
&= \|z^{k-1} - x^*\|^2 - 2\eta\langle \nabla f_{i_k}(x^k), x^k - x^*\rangle \\
&\quad - 2\lambda_k \eta\langle \nabla f_{i_k}(x^k), x^k - x^{k-1}\rangle + \eta^2 \|\nabla f_{i_t}(x^k)\|^2 \qquad \text{[by (2)]}
\end{aligned}$$

Taking the expectation conditioned on x^k on both sides, we get

$$\begin{aligned}
\mathbb{E}[\|z^k - x^*\|^2 | x^k] \leq \mathbb{E}[\|z^{k-1} - x^*\|^2 | x^k] &- 2\eta\langle \nabla f(x^k), x^k - x^*\rangle \\
&- 2\lambda_k \eta\langle \nabla f(x^k), x^k - x^{k-1}\rangle + \eta^2 \mathbb{E}[\|\nabla f_{i_t}(x^k)\|^2]
\end{aligned}$$

By convexity, for any $x, y \in \mathbb{R}$,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$$

And by variance transfer,

$$\mathbb{E}[\|\nabla f_i(x)\|^2] \leq 4L_{\max}(f(x) - \inf f) + 2\sigma_f^*$$

Therefore,

$$\begin{aligned}
\mathbb{E}[\|z^k - x^*\|^2 | x^k] &\leq \mathbb{E}[\|z^{k-1} - x^*\|^2 | x^k] - 2\eta(f(x^k) - \inf f) \\
&\quad - 2\lambda_k \eta(f(x^k) - f(x^{k-1})) + \eta^2(4L_{\max}(f(x^k) - \inf f) + 2\sigma_f^*) \\
&= \mathbb{E}[\|z^{k-1} - x^*\|^2 | x^k] - 2\eta(1 + \lambda_k - 2\eta L_{\max})(f(x^k) - \inf f) \\
&\quad + 2\lambda_k \eta(f(x^{k-1}) - \inf f) + 2\eta^2 \sigma_f^*
\end{aligned}$$

Since $\eta \leq \frac{1}{4L_{\max}} \Rightarrow 2\eta L_{\max} \leq \frac{1}{2}$, we have $1 + \lambda_k - 2\eta L_{\max} \geq \lambda_k + \frac{1}{2} = \lambda_{k+1}$.
Thus,

$$\begin{aligned}
\mathbb{E}[\|z^k - x^*\|^2 | x^k] &\leq \mathbb{E}[\|z^{k-1} - x^*\|^2 | x^k] - 2\eta\lambda_{k+1}(f(x^k) - \inf f) \\
&\quad + 2\lambda_k \eta(f(x^{k-1}) - \inf f) + 2\eta^2 \sigma_f^*
\end{aligned}$$

Taking expectation with respect to x^k on both sides,

$$\mathbb{E}[\|z^k - x^*\|^2] \leq \mathbb{E}[\|z^{k-1} - x^*\|^2] - 2\eta\lambda_{k+1}\mathbb{E}[f(x^k) - \inf f] + 2\lambda_k\eta\mathbb{E}[f(x^{k-1}) - \inf f] + 2\eta^2\sigma_f^*$$

By Equation (2), let $k = -1$, then

$$z^{-1} = x^0 + \lambda_0(x^0 - x^{-1}) = x^0$$

since $\lambda_0 = 0$.
Summing up $k = 0, 1, ..., k$ on both sides, we get

$$\begin{aligned}\mathbb{E}[\|z^k - x^*\|^2] &\leq \|x^0 - x^*\|^2 - 2\eta\lambda_{k+1}\mathbb{E}[f(x^k) - \inf f] + 2(k+1)\eta^2\sigma_f^* \\ &= \|x^0 - x^*\|^2 - (k+1)\eta\mathbb{E}[f(x^k) - \inf f] + 2(k+1)\eta^2\sigma_f^*\end{aligned}$$

Since $\mathbb{E}[\|z^k - x^*\|^2] \geq 0$, we have

$$0 \leq \|x^0 - x^*\|^2 - \eta(k+1)\mathbb{E}[f(x^k) - \inf f] + 2(k+1)\eta^2\sigma_f^*$$

$$\mathbb{E}[f(x^k) - \inf f] \leq \frac{\|x^0 - x^*\|^2}{\eta(k+1)} + 2\eta\sigma_f^*$$

Convergence Rate

Corollary 1.6: Consider the setting of Theorem 1.4. Let T be the maximum number of iterations, by taking $\eta = \frac{1}{4L_{\max}} \frac{1}{\sqrt{T+1}}$, we have

$$\mathbb{E}[f(x^T) - \inf f] \leq \frac{4L_{\max}\|x^0 - x^*\|^2}{\sqrt{T+1}} + \frac{1}{\sqrt{T+1}} \frac{1}{2L_{\max}} \sigma_f^*$$

Thus, to guarantee $\mathbb{E}[f(x^T) - \inf f] \leq \epsilon$, we need

$$T \geq \left(4L_{\max}\|x^0 - x^*\|^2 + \frac{1}{2L_{\max}}\sigma_f^*\right)^2 \frac{1}{\epsilon^2} - 1$$

which means the iteration complexity is $O(\frac{1}{\epsilon^2})$. [2]

[2] Remember that the convergence rate of SGD is also $O(\frac{1}{\epsilon^2})$, but it is for $f(\bar{x}^k)$, where $\bar{x}^k$ is a weighted average of $x^0, ..., x^k$. This convergence rate is for the last iteration $f(x^k)$.

Contents

Nestrov's Accelerated Gradient Descent

Nesterov's Accelerated Gradient Descent is another momentum based method, proposed by Yurii Nesterov in 1983 [4]. I will call this method as Nestrov in the following text for simplicity.

Suppose the function f is convex and L-smooth. The iteration complexity of gradient descent is $O(1/\epsilon)$. By using Nesterov, we can improve the complexity to $O(1/\sqrt{\epsilon})$, which is the *optimal* complexity that can be achieved by gradient descent methods [5].

Algorithm 2.1 (Nesterov): Let the function $f : \mathbb{R}^d \to \mathbb{R}$ be convex and L-smooth. Let $\{t_k\}_{k\in\mathbb{N}} \in [\frac{1}{2L}, \frac{1}{L}]$ be a sequence of non-increasing step sizes. Let $\{\lambda_k\}_{k\in\mathbb{N}}$ be a sequence of momentum parameters satisfying:

$$\lambda_0 = 1 \quad , \quad \lambda_{k+1} = \frac{1 + \sqrt{4\lambda_k^2 + 1}}{2} \text{ for } k \geq 0$$

Let $y^0 \in \mathbb{R}^d$ be a starting point. The Nesterov defines a sequence $\{x^k\}_{k\in\mathbb{N}}$ satisfying for every $t \in \mathbb{N}$,

$$x^k = y^k - t_k \nabla f(y^k)$$
$$y^{k+1} = x^k + \frac{\lambda_k - 1}{\lambda_{k+1}}(x^k - x^{k-1})$$

Nesterov and Heavy Ball Momentum

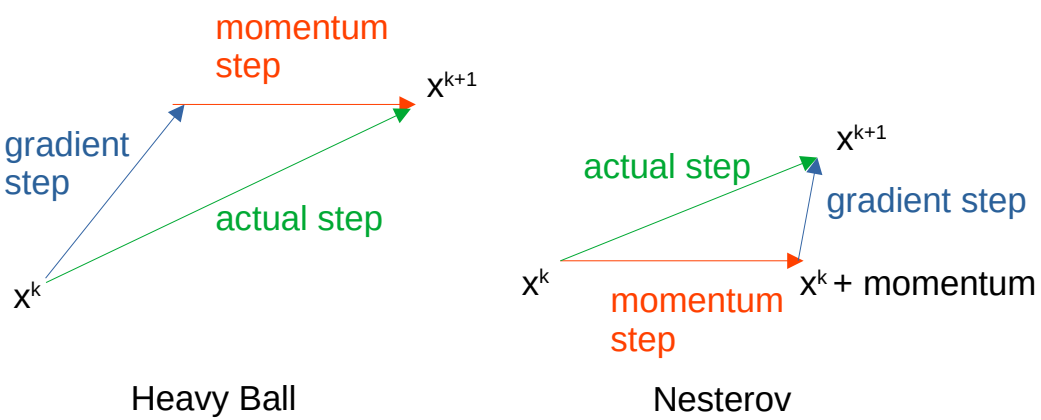
Both Algorithm 1.1 (Momentum) and Algorithm 2.1 (Nesterov) are based on momentum. Algorithm 1.1 is often referred to Heavy Ball Momentum, which can be written as

$$x^{k+1} = x^k - \underbrace{\hat{t}_k \nabla f_{i_k}(x^k)}_{\text{gradient}} + \underbrace{\hat{\beta}_k (x^k - x^{k-1})}_{\text{momentum}}$$

For Algorithm 2.1, let $\beta_k = \frac{\lambda_k - 1}{\lambda_{k+1}}$, we can write

$$x^{k+1} = x^k - \underbrace{t_{k+1} \nabla f(x^k + \beta_k (x^k - x^{k-1}))}_{\text{gradient}} + \underbrace{\beta_k (x^k - x^{k-1})}_{\text{momentum}}$$

So here comes the difference: The gradient of Heavy Ball depends only on x^k, while the gradient of Nesterov depends on x^k plus the momentum.



One can consider Heavy Ball applies gradient descent on x^k first, then applies momentum; Nesterov applies momentum on x^k first, then applies gradient descent on x^k + momentum.

Convergence Analysis of Nesterov

Theorem 2.2: Consider the settings of Algorithm 2.1, $f(x^k)$ satisfies

$$f(x^k) - \inf f \leq \frac{C}{(k+2)^2}$$

where $C = 4L\|y^0 - x^*\|^2$ is a constant.

Proof: Since f is L-smooth. Let $x, y \in \mathbb{R}^d$, we have

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2$$

Let $y = y^k - t_k \nabla f(y^k)$, $x = y^k$, we get

$$f(y^k) - f(y^k - t_k \nabla f(y^k)) \geq \frac{1}{2} t_k (2 - L t_k) \|\nabla f(y^k)\|^2$$

Since $\frac{1}{2L} \leq t_k \leq \frac{1}{L}$, we have $\frac{1}{2} t_k (2 - L t_k) \geq \frac{1}{2} t_k$, thus

$$f(y^k) - f(y^k - t_k \nabla f(y^k)) \geq \frac{1}{2} t_k \|\nabla f(y^k)\|^2 \tag{4}$$

Let $p_k = (\lambda_k - 1)(x^{k-1} - x^k)$, we have

$$y^{k+1} = x^k - \frac{p^k}{\lambda_{k+1}} \tag{5}$$

$$\begin{aligned} p_{k+1} - x^{k+1} &= (\lambda_{k+1} - 1)(x^k - x^{k+1}) - x^{k+1} \\ &= (\lambda_{k+1} - 1)x^k - \lambda_{k+1}x^{k+1} \\ &= (\lambda_{k+1} - 1)x^k - \lambda_{k+1}(y^{k+1} - t_{k+1}\nabla f(y^{k+1})) \\ &= (\lambda_{k+1} - 1)x^k - \lambda_{k+1}(x^k - \frac{p^k}{\lambda_{k+1}}) + \lambda_{k+1}t_{k+1}\nabla f(y^{k+1}) \\ &= p_k - x^k + \lambda_{k+1}t_{k+1}\nabla f(y^{k+1}) \end{aligned} \tag{6}$$

Then

$$\begin{aligned} \|p_{k+1} - x^{k+1} + x^*\|^2 = \|p_k - x^k + x^*\|^2 &+ 2\lambda_{k+1}t_{k+1}\langle \nabla f(y^{k+1}), p_k - x^k + x^*\rangle \\ &+ \lambda_{k+1}^2 t_{k+1}^2 \|\nabla f(y^{k+1})\|^2 \\ = \|p_k - x^k + x^*\|^2 &+ 2(\lambda_{k+1} - 1)t_{k+1}\langle \nabla f(y^{k+1}), p_k\rangle + \\ &2\lambda_{k+1}t_{k+1}\langle \nabla f(y^{k+1}), x^* - y^{k+1}\rangle + \lambda_{k+1}^2 t_{k+1}^2 \|\nabla f(y^{k+1})\|^2 \end{aligned}$$

The last equality replaces x^k by Equation (5).

By Inequality (4) we have

$$f(y^{k+1}) - f(x^{k+1}) \geq \frac{1}{2} t_{k+1} \|\nabla f(y^{k+1})\|^2 \tag{7}$$

By the convexity of f, for any $x, y \in \mathbb{R}^d$,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle \iff f(y) + \langle \nabla f(x), x - y \rangle \geq f(x)$$

Let $y = x^*, x = y^{k+1}$, then

$$\inf f + \langle \nabla f(y^{k+1}), y^{k+1} - x^* \rangle \geq f(y^{k+1}) \tag{8}$$

Let $y = x^k, x = y^{k+1}$, then

$$f(x^k) + \langle \nabla f(y^{k+1}), y^{k+1} - x^k \rangle = f(x^k) - \frac{1}{\lambda_{k+1}} \langle \nabla f(y^{k+1}), p_k \rangle \geq f(y^{k+1}) \tag{9}$$

Combining (7) and (8), we get

$$\langle \nabla f(y^{k+1}), y^{k+1} - x^* \rangle \geq f(x^{k+1}) - \inf f + \frac{1}{2} t_{k+1} \|\nabla f(y^{k+1})\|^2 \tag{10}$$

Combining (7) and (9), we get

$$f(x^k) - f(x^{k+1}) - \frac{1}{\lambda_{k+1}} \langle \nabla f(y^{k+1}), p_k \rangle \geq \frac{1}{2} t_{k+1} \|\nabla f(y^{k+1})\|^2 \tag{11}$$

By (10) and (11),

$$
\begin{aligned}
&\|p_{k+1} - x^{k+1} + x^*\|^2 - \|p_k - x^k + x^*\|^2 \\
&= 2(\lambda_{k+1} - 1)t_{k+1}\langle \nabla f(y^{k+1}), p_k\rangle + 2\lambda_{k+1}t_{k+1}\langle \nabla f(y^{k+1}), x^* - y^{k+1}\rangle \\
&\quad + \lambda_{k+1}^2 t_{k+1}^2 \|\nabla f(y^{k+1})\|^2 \\
&\le 2(\lambda_{k+1} - 1)t_{k+1}\lambda_{k+1}\left(f(x^k) - f(x^{k+1}) - \frac{1}{2}t_{k+1}\|\nabla f(y^{k+1})\|^2\right) - \\
&\quad 2\lambda_{k+1}t_{k+1}\left(f(x^{k+1}) - \inf f + \frac{1}{2}t_{k+1}\|\nabla f(y^{k+1})\|^2\right) + \lambda_{k+1}^2 t_{k+1}^2\|\nabla f(y^{k+1})\|^2 \\
&= 2(\lambda_{k+1} - 1)t_{k+1}\lambda_{k+1}f(x^k) + 2\lambda_{k+1}t_{k+1}\inf f - 2\lambda_{k+1}^2 t_{k+1}f(x^{k+1}) \\
&= 2(\lambda_{k+1} - 1)t_{k+1}\lambda_{k+1}(f(x^k) - \inf f) - 2\lambda_{k+1}^2 t_{k+1}(f(x^{k+1}) - \inf f) \\
&= 2\lambda_k^2 t_{k+1}(f(x^k) - \inf f) - 2\lambda_{k+1}^2 t_{k+1}(f(x^{k+1}) - \inf f) \qquad (12) \\
&\le 2\lambda_k^2 t_k(f(x^k) - \inf f) - 2\lambda_{k+1}^2 t_{k+1}(f(x^{k+1}) - \inf f) \qquad (13)
\end{aligned}
$$

(12) uses the fact that

$$
\lambda_{k+1} = \frac{1 + \sqrt{4\lambda_k^2 + 1}}{2} \Leftrightarrow \lambda_{k+1}^2 - \lambda_{k+1} - \lambda_k^2 = 0 \Leftrightarrow \lambda_{k+1}(\lambda_{k+1} - 1) = \lambda_k^2
$$

(13) is because the sequence $\{t_k\}$ is non-increasing, $t_{k+1} \le t_k$.

Therefore,

$$\begin{aligned}
2\lambda_{k+1}^2 t_{k+1}(f(x^{k+1}) - \inf f) &\le 2\lambda_{k+1}^2 t_{k+1}(f(x^{k+1}) - \inf f) + \|p_{k+1} - x^{k+1} + x^*\|^2 \\
&\le 2\lambda_k^2 t_k(f(x^k) - \inf f) + \|p_k - x^k + x^*\|^2 \\
&\le 2\lambda_0^2 t_0(f(x^0) - \inf f) + \|p_0 - x^0 + x^*\|^2 \quad \text{[By induction]} \\
&= 2t_0(f(x^0) - \inf f) + \|x^0 - x^*\|^2 \qquad (14) \\
&\le \|y^0 - x^*\|^2 \qquad (15)
\end{aligned}$$

(14) is because $\lambda_0 = 1$ and $p_0 = 0$. (15) is because

$$\begin{aligned}
\|y^0 - x^*\|^2 - \|x^0 - x^*\|^2 &= \|y^0 - x^*\|^2 - \|y^0 - t_0\nabla f(y^0) - x^*\|^2 \\
&= 2t_0\langle \nabla f(y^0), y^0 - x^*\rangle - t_0^2\|\nabla f(y^0)\|^2
\end{aligned}$$

In (10), letting $k = -1$, we get

$$\langle \nabla f(y^0), y^0 - x^*\rangle \ge f(x^0) - \inf f + \frac{1}{2}t_0\|\nabla f(y^0)\|^2$$

Thus,

$$\|y^0 - x^*\|^2 - \|x^0 - x^*\|^2 \ge 2t_0(f(x^0) - \inf f)$$

Therefore, we have

$$f(x^k) - \inf f \leq \frac{\|y^0 - x^*\|^2}{2t_k\lambda_k^2}$$

Since $\lambda_0 = 1$ and

$$\lambda_{k+1} = \frac{1 + \sqrt{4\lambda_k^2 + 1}}{2} \geq \frac{1}{2} + \lambda_k$$

thus $\lambda_k \geq 1 + \frac{1}{2}k$. And since $t_k \geq \frac{1}{2L}$, we have

$$f(x^k) - \inf f \leq \frac{\|y^0 - x^*\|^2}{2\frac{1}{2L}(1 + \frac{1}{2}k)^2} = \frac{4L\|y^0 - x^*\|^2}{(k+2)^2}$$

Corollary 2.3: Consider the settings of Theorem 2.2. To make $f(x^k) - \inf f \leq \epsilon$, we need

$$\frac{4L\|y^0 - x^*\|^2}{(k+2)^2} \leq \epsilon \Longrightarrow k \geq \frac{2\sqrt{L}\|y^0 - x^*\|}{\sqrt{\epsilon}} - 2$$

Thus, the iteration complexity is $O(\frac{1}{\sqrt{\epsilon}})$ [3].

[3] The Nestrov's paper uses the convergence rate $O(\frac{1}{k^2})$ to describe the performance of the algorithm. The convergence rate considers the maximum error as a function of number of iterations, while our iteration complexity considers the minimum number of iterations as a function of error. In a word, $O(\frac{1}{k^2})$ for ϵ is equivalent to $O(\frac{1}{\sqrt{\epsilon}})$ for k.

The Design of Nesterov's Accelerated Gradient

Now we introduce how the Algorithm 2.1 come. The contents are from Nesterov's book *Introductory lectures on convex programming* Section 2.2.1 [14].

Definition 2.4: A pair of sequences $\{\phi_k(x)\}_{k\in\mathbb{N}}$ and $\{\delta_k\}_{k\in\mathbb{N}}$ is called an estimate sequence of function $f(x)$ if for any $x \in \mathbb{R}^d$ and all $k > 0$ we have

$$\phi_k(x) \leq (1-\delta_k)f(x) + \delta_k\phi_0(x) \tag{16}$$

Lemma 2.5: If for a sequence $\{x_k\}_{k\in\mathbb{N}}$ we have

$$f(x^k) \leq \phi_k^* \equiv \min_{x\in\mathbb{R}^d} \phi_k(x)$$

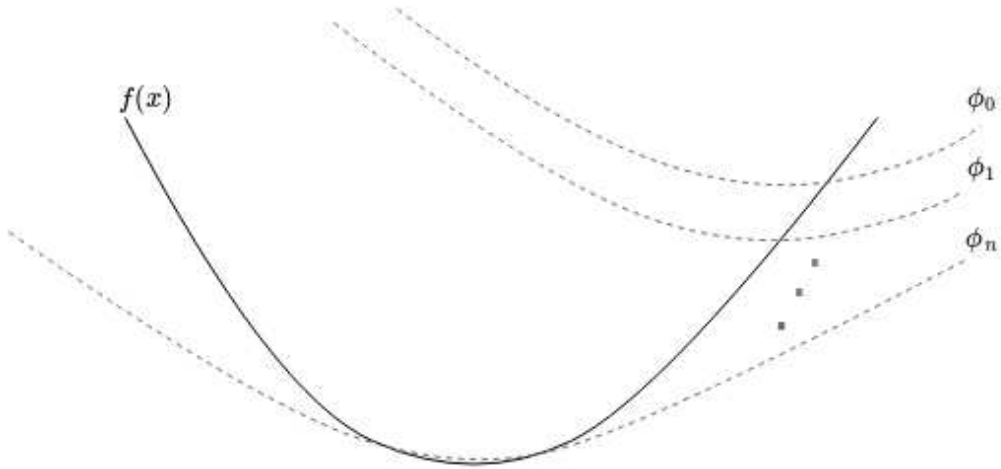
Then $f(x^k) - \inf f \leq \delta_k[\phi_0(x^*) - \inf f] \to 0$.

Proof:

$$f(x^k) \leq \phi_k^* = \min_{x\in\mathbb{R}^d} \phi_k(x) \leq \min_{x\in\mathbb{R}^d}[(1-\delta_k)f(x) + \delta_k\phi_0(x)] \leq (1-\delta_k)f(x^*) + \delta_k\phi_0(x^*)$$

$$\Longrightarrow f(x^k) - \inf f \leq \delta_k[\phi_0(x^*) - \inf f]$$

The main idea is, if we can construct the estimation sequence $\{\phi_k(x)\}_{k\in\mathbb{N}}$ and $\{\delta_k\}_{k\in\mathbb{N}}$ satisfying Definition 2.4, and find x^k such that $f(x^k) \leq \phi_k^*$ for all $k > 0$, then we can guarantee $x^k \to x^*$ when $k \to \infty$.



An illustration of estimation sequences

So the questions left are (1) how to find the estimation sequence $\{\phi_k(x)\}_{k\in\mathbb{N}}$ and $\{\delta_k\}_{k\in\mathbb{N}}$ and (2) how to find $\{x_k\}_{k\in\mathbb{N}}$ such that $f(x^k) \leq \phi_k^*$ for any $k > 0$.

Lemma 2.6: Let us assume that:
1. f is convex and L-smooth.
2. $\phi_0(x)$ is an arbitrary function on $\mathbb{R}^d$.
3. $\{y^k\}_{k\in\mathbb{N}}$ is an arbitrary sequence in $\mathbb{R}^n$.
4. $\{\alpha_k\}_{k\in\mathbb{N}}$, $\alpha_k \in (0,1)$, $\sum_{k=0}^{\infty} \alpha_k = \infty$.
5. $\delta_0 = 1$.

Then the pair of sequences $\{\phi_k(x)\}_{k\in\mathbb{N}}$ and $\{\delta_k\}_{k\in\mathbb{N}}$ defined by the following recursive rules:

$$\delta_{k+1} = (1-\alpha_k)\delta_k \tag{17}$$

$$\phi_{k+1}(x) = (1-\alpha_k)\phi_k(x) + \alpha_k[f(y^k) + \langle \nabla f(y^k), x - y^k\rangle] \tag{18}$$

is an estimate sequence.

Proof: Indeed, $\phi_0(x) = (1-\delta_0)f(x) + \delta_0\phi_0(x) \equiv \phi_0(x)$. By induction, for $k > 0$,

$$\begin{aligned}
\phi_{k+1}(x) &\le (1-\alpha_k)\phi_k(x) + \alpha_k f(x) && [f(x) \ge f(y^k) + \langle \nabla f(y^k), x - y^k\rangle] \\
&= (1 - (1-\alpha_k)\delta_k)f(x) + (1-\alpha_k)(\phi_k(x) - (1-\delta_k)f(x)) \\
&= (1 - (1-\alpha_k)\delta_k)f(x) + (1-\alpha_k)\delta_k\phi_0(x) && [\text{By (16)}] \\
&\le (1-\delta_{k+1})f(x) + \delta_{k+1}\phi_0(x) && [\text{We defined } \delta_{k+1} = (1-\alpha_k)\delta_k]
\end{aligned}$$

Thus, $\phi_{k+1}(x)$ also satisfies (16).

Lemma 2.6 tells us that we can define the estimation sequence $\{\phi_k(x)\}_{k\in\mathbb{N}}$ and $\{\delta_k\}_{k\in\mathbb{N}}$ by a recursive rule. Indeed, (17) will make $\delta_k \to 0$ when $k \to \infty$ as long as $(1-\alpha_k) \in (0,1)$ for any k and $\delta_0 = 1$ are guaranteed. So we only need to look at (18). We can let $\phi_k(x)$ be a simple quadratic function, as shown in the following Lemma:

Lemma 2.7: Let $\phi_0(x) = \phi_0^* + \frac{\gamma_0}{2}\|x - v_0\|^2$, then the process (18) forms:

$$\phi_k(x) = \phi_k^* + \frac{\gamma_k}{2}\|x - v_k\|^2 \tag{19}$$

where $\{\gamma_k\}, \{v_k\}, \{\phi_k^*\}$ are defined as follows:

$$\gamma_{k+1} = (1-\alpha_k)\gamma_k$$

$$v_{k+1} = v_k - \frac{\alpha_k}{\gamma_{k+1}}\nabla f(y^k)$$

$$\phi_{k+1}^* = (1-\alpha_k)\phi_k^* + \alpha_k f(y^k) - \frac{\alpha_k^2}{2\gamma_{k+1}}\|\nabla f(y^k)\|^2 + \alpha_k\langle \nabla f(y^k), v_k - y^k\rangle$$

Proof: We show that the induction rules of $\gamma_{k+1}, v_{k+1}, \phi_{k+1}^*$ can be obtained from (18), respectively. Look at $\{\gamma_k\}$ first. Since $\phi_0''(x) = \gamma_0 I_n$, and by (18),

$$\nabla^2\phi_{k+1}(x) = (1-\alpha_k)\nabla^2\phi_k(x)$$

By letting $\gamma_{k+1} = (1-\alpha_k)\gamma_k$, we have

$$\nabla^2\phi_k(x) = \prod_{i=0}^{k-1}(1-\alpha_i)\nabla^2\phi_0(x) = \prod_{i=0}^{k-1}(1-\alpha_i)\gamma_0 I_n = \gamma_k I_n$$

So the process $\gamma_{k+1} = (1-\alpha_k)\gamma_k$ will exactly give the γ_k in (19).
Now let's look at $\{v_k\}$. By (18) and (19) we have

$$\phi_{k+1}(x) = (1-\alpha_k)[\phi_k^* + \frac{\gamma_k}{2}\|x-v_k\|^2] + \alpha_k[f(y^k) + \langle\nabla f(y^k), x-y^k\rangle] \tag{20}$$

$$\nabla\phi_{k+1}(x) = (1-\alpha_k)\gamma_k(x-v_k) + \alpha_k\nabla f(y^k) = \gamma_{k+1}(x-v_k) + \alpha_k\nabla f(y^k)$$

Let $\nabla\phi_{k+1}(x) = 0$, we get $x = v_k - \frac{\alpha_k}{\gamma_{k+1}}\nabla f(y^k)$.
Note that by (19), $\nabla\phi_{k+1}(x) = 0$ also means $x = v_{k+1}$, thus $v_{k+1} = v_k - \frac{\alpha_k}{\gamma_{k+1}}\nabla f(y^k)$.
Finally, let's compute ϕ_{k+1}. By (19) and (20),

$$\phi_{k+1}^* + \frac{\gamma_{k+1}}{2}\|y^k - v_{k+1}\|^2 = \phi_{k+1}(y^k) = (1-\alpha_k)[\phi_k^* + \frac{\gamma_k}{2}\|y^k - v_k\|^2] + \alpha_k f(y^k)$$

Since

$$v_{k+1} - y^k = v_k - y^k - \frac{\alpha_k}{\gamma_{k+1}}\nabla f(y^k)$$

Therefore,

$$\phi_{k+1}^* + \frac{\gamma_{k+1}}{2}\|v_k - y^k\|^2 - \alpha_k\langle\nabla f(y^k), v_k - y^k\rangle + \frac{\alpha_k^2}{2\gamma_{k+1}}\|\nabla f(y^k)\|^2 = (1-\alpha_k)\phi_k^*$$
$$+ \frac{\gamma_{k+1}}{2}\|v_k - y^k\|^2 + \alpha_k f(y^k) \Longrightarrow$$
$$\phi_{k+1}^* = (1-\alpha_k)\phi_k^* + \alpha_k f(y^k) - \frac{\alpha_k^2}{2\gamma_{k+1}}\|\nabla f(y^k)\|^2 + \alpha_k\langle\nabla f(y^k), v_k - y^k\rangle$$

Hence we proved Lemma 2.7.

So far, we have found an estimation sequence $\{\gamma_k\}$ and $\{\phi_k(x)\}$. Note that $\{x^k\}$ and $\{y^k\}$ are not determined yet. How do we find x^k such that $f(x^k) \leq \phi_k^*$? There are multiple ways to achieve this, the following is one way:

By Lemma 2.7 we have

$$\phi_{k+1}^* = (1-\alpha_k)\phi_k^* + \alpha_k f(y^k) - \frac{\alpha_k^2}{2\gamma_{k+1}}\|\nabla f(y^k)\|^2 + \alpha_k\langle\nabla f(y^k), v_k - y^k\rangle$$

Assume $f(x^k) \leq \phi_k^*$, then

$$\phi_{k+1}^* \geq (1-\alpha_k)f(x^k) + \alpha_k f(y^k) - \frac{\alpha_k^2}{2\gamma_{k+1}}\|\nabla f(y^k)\|^2 + \alpha_k\langle\nabla f(y^k), v_k - y^k\rangle$$

Since $f(x^k) \geq f(y^k) + \langle \nabla f(y^k), x^k - y^k \rangle$, we have

$$\begin{aligned}\phi_{k+1}^* &\geq (1-\alpha_k)(f(y^k) + \langle \nabla f(y^k), x^k - y^k \rangle) + \alpha_k f(y^k) - \frac{\alpha_k^2}{2\gamma_{k+1}} \|\nabla f(y^k)\|^2 + \\ &\quad \alpha_k \langle \nabla f(y^k), v_k - y^k \rangle \\ &= f(y^k) + \alpha_k \langle \nabla f(y^k), \alpha_k(v_k - x^k) + x^k - y^k \rangle - \frac{\alpha_k^2}{2\gamma_{k+1}} \|\nabla f(y^k)\|^2\end{aligned}$$

Now we want $\phi_{k+1} \geq f(x^{k+1})$. Remember that in (4), when $\frac{1}{2L} \leq t_k \leq \frac{1}{L}$.

$$f(y^k) - f(y^k - t_k \nabla f(y^k)) \geq \frac{1}{2} t_k \|\nabla f(y^k)\|^2$$

We can use the gradient decent to get x^{k+1} by letting $x^{k+1} = y^k - t_k \nabla f(y^k)$. For simplicity, we let $t_k = \frac{1}{L}$, and choose $\{\alpha_k\}$ by letting

$$L\alpha_k^2 = \gamma_{k+1} = \prod_{i=0}^{k} (1 - \alpha_i)\gamma_0$$

Then,

$$\begin{aligned}\phi_{k+1}^* &\geq f(y^k) + \alpha_k \langle \nabla f(y^k), \alpha_k(v_k - x^k) + x^k - y^k \rangle - \frac{\alpha_k^2}{2\gamma_{k+1}} \|\nabla f(y^k)\|^2 \\ &\geq f(x^{k+1}) + \frac{1}{2L}\|\nabla f(y^k)\|^2 + \alpha_k \langle \nabla f(y^k), \alpha_k(v_k - x^k) + x^k - y^k \rangle \\ &\quad - \frac{1}{2L}\|\nabla f(y^k)\|^2 \\ &= f(x^{k+1}) + \alpha_k \langle \nabla f(y^k), \alpha_k(v_k - x^k) + x^k - y^k \rangle\end{aligned}$$

Now we can use our freedom in the choice of y^k. By taking

$$\alpha_k(v_k - x^k) + x^k - y^k = 0 \Rightarrow y^k = (1 - \alpha_k)x^k + \alpha_k v_k$$

We can guarantee $\phi_{k+1}^* \geq f(x^{k+1})$.

Algorithm 2.8: To sum up, $\{x^k\}_{k\in\mathbb{N}}$ can be obtained by the following algorithm:

0. Choose $x^0 \in \mathbb{R}^d$ and $\gamma_0 > 0$, set $v_0 = x^0$.
1. k-th iteration ($k \geq 0$)

    1.1 Compute $\alpha_k \in (0, 1)$ from the equation

$$L\alpha_k^2 = \prod_{i=0}^{k}(1-\alpha_i)\gamma_0$$

    1.2 Compute

$$y^k = (1-\alpha_k)x^k + \alpha_k v_k$$

    1.3 Compute

$$x^{k+1} = y^k - \frac{1}{L}\nabla f(y^k)$$

    1.4 Compute

$$v_{k+1} = v_k - \frac{\alpha_k}{\gamma_{k+1}}\nabla f(y^k)$$

We can make Algorithm 2.8 looks like Algorithm 2.1 by removing the v_k term. Note that

$$
\begin{aligned}
y^k &= (1-\alpha_k)x^k + \alpha_k v_k \\
v_k &= v_{k-1} - \frac{\alpha_{k-1}}{\gamma_k}\nabla f(y^{k-1}) \\
v_{k-1} &= \frac{1}{\alpha_{k-1}}y^{k-1} - (\frac{1}{\alpha_{k-1}} - 1)x^{k-1} \\
y^{k-1} &= x^k + \frac{1}{L}\nabla f(y^{k-1})
\end{aligned}
$$

By plugging in the variables with the same color, we get

$$
y^k = x^k + \alpha_k(\frac{1}{\alpha_{k-1}} - 1)(x^k - x^{k-1}) + \alpha_k\left(\frac{1}{\alpha_{k-1}L} - \frac{\alpha_{k-1}}{\gamma_k}\right)\nabla f(y^{k-1})
$$

Note that $L\alpha_{k-1}^2 = \gamma_k \Rightarrow \frac{1}{\alpha_{k-1}L} = \frac{\alpha_{k-1}}{\gamma_k}$, then

$$
y^k = x^k + \alpha_k(\frac{1}{\alpha_{k-1}} - 1)(x^k - x^{k-1})
$$

By letting $\lambda_k = \frac{1}{\alpha_k}$, we get

$$
y^k = x^k + \frac{\lambda_{k-1} - 1}{\lambda_k}(x^k - x^{k-1})
$$

So the iteration becomes

$$y^k = x^k + \frac{\lambda_{k-1} - 1}{\lambda_k}(x^k - x^{k-1})$$
$$x^{k+1} = y^k - \frac{1}{L}\nabla f(y^k)$$

Shifting all the indexes of $\{x^k\}_{k\in\mathbb{N}}$ by -1, then this iteration will be identical to Algorithm 2.1 except the following differences:

(1) The step size. Algorithm 2.8 fixes the step size to be $\frac{1}{L}$. While in Algorithm 2.1, the step sizes can vary in $[\frac{1}{2L}, \frac{1}{L}]$.

(2) The choice of $\{\lambda_k\}$. Algorithm 2.8 requires $\lambda_k = \frac{1}{\alpha_k}$ where $L\alpha_k^2 = \prod_{i=0}^{k}(1-\alpha_i)\gamma_0$, this will not result in $\lambda_{k+1} = \frac{1+\sqrt{4\lambda_k^2+1}}{2}$ as Algorithm 2.1.

By Theorem 2.2.2 in [14], Algorithm 2.8 converges at $O(\frac{1}{\sqrt{\epsilon}})$ complexity, which is the same as Algorithm 2.1. So Algorithm 2.1 and Algorithm 2.8 are basically one algorithm with different parameter settings. It needs to be verified that if the parameter settings in Algorithm 2.1 satisfy the conditions of Lemma 2.7.

Nesterov in Stochastic Gradient Descent

When Nestrov is run with SGD, there is no definitive theoretical convergence guarantees so far [7]. Leon Bottou et al [5] states that applying Nesterov to SGD will not improve the convergence rate, because the iteration complexity will be no better than $O(\frac{1}{\epsilon^2})$. The reasons are below:

We call gradient descent as first order method, because it only uses the first order information (first derivative) of f. Similarly, Newton's method uses the second order information (second derivative) of f and is called second order method.

Obtaining second order information is more computational costly than obtaining first order information. So first order method is faster than second order method in each iteration. However, since first order method obtains fewer information than second order method in each iteration, it needs more iterations to converge.

For SGD, when f is convex and L-smooth, the lower bound of iteration complexity is $O(\frac{1}{\epsilon^2})$ [6]. This lower bound is restricted by the limitation of first order information.

We have already proved that, when f is (Sum of Convex) and (Sum of $L_{\max}$-smooth): (1) Without Momentum, SGD converges at a complexity $O(\frac{1}{\epsilon^2})$ for $f(\bar{x}^k)$; (2) With Momentum, SGD converges at a complexity $O(\frac{1}{\epsilon^2})$ for $f(x^k)$. The complexity has reached lower bound, so it will not be improved even Nesterov is applied.

Contents

Online Learning Problem

Remember that SGD solves a (Sum of Functions) problem: we optimize a convex function f which can be written as a sum of functions $f_1, f_2, ..., f_n$:

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

The Assumption (Sum of Convex) says each f_i is convex. The Assumption (Sum of $L_{\max}$-Smooth) says each f_i is L_i-smooth and $\max_i\{L_i\} = L_{\max}$.

Now let's consider a new problem [9]:

Definition (Online Learning Regret): Given an arbitrary, unknown, infinite sequence of convex functions $f_1(x), f_2(x), f_3(x)....$ Our goal is to predict x_t at each time t, such that at any time T the regret

$$R(T) = \sum_{t=1}^{T} [f_t(x_t) - f_t(x^*)]$$

is minimized, where $x^* = \text{argmin}_x \sum_{t=1}^{T} f_t(x)$.

Online Learning Problem

Since $R(T)$ does not converge, we consider the average regret:

$$\bar{R}(T) = \frac{R(T)}{T} = \frac{1}{T}\sum_{t=1}^{T}[f_t(x_t) - f_t(x^*)]$$

Minimizing $R(T)$ is equivalent to minimizing $\bar{R}(T)$. To show the convergence, we need to show that $\bar{R}(T) \to 0$ when $T \to \infty$.

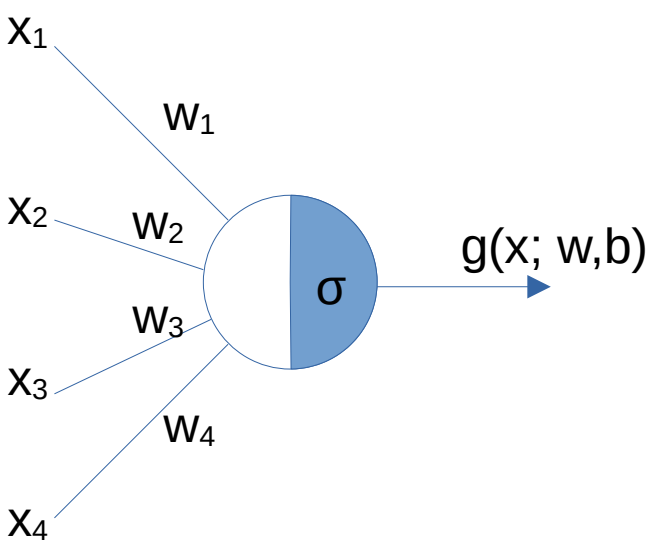
The differences between (Online Learning) problem and (Sum of Functions) problem are: In Online Learning, the sequence $f_1, f_2, f_3...$ is unknown in advance, and x^* is changing with the sequence increasing.

Sparse Gradient

Sparse gradient is a critical problem in training a neural network with sparse data. The gradients can be sparse on some dimensions while dense on other dimensions.

For a group of vectors, taking the i-th element of each vector, if most of them are 0, we say that the group of vectors is sparse on dimension i.

Let's see an example about how sparse gradient comes. Let $(\boldsymbol{x}, y)$ be a training sample where $\boldsymbol{x} = [x_1, x_2, x_3, x_4]^T \in \mathbb{R}^4$ and $y \in \mathbb{R}$. Suppose we have a neural network $g(\boldsymbol{x}; \boldsymbol{w}, b)$, where $\boldsymbol{w} = [w_1, w_2, w_3, w_4]^T \in \mathbb{R}^4$ is a weight vector and $b \in \mathbb{R}$ is a bias.



Let the activation function σ be sigmoid, we can write $g(\boldsymbol{x}; \boldsymbol{w}, b)$ as

$$g(\boldsymbol{x}; \boldsymbol{w}, b) = \frac{1}{1 + e^{-(\boldsymbol{w}^T \boldsymbol{x} + b)}}$$

And we define the loss function as

$$f(\boldsymbol{x}, y; \boldsymbol{w}, b) = \left(g(\boldsymbol{x}; \boldsymbol{w}, b) - y\right)^2$$

Sparse Gradient

The gradients of f with respect to $\boldsymbol{w}$ and b are calculated by

$$\frac{\partial f}{\partial w_i} = 2(g(\boldsymbol{x}) - y)(g(\boldsymbol{x}) - g(\boldsymbol{x})^2)x_i \quad [i = 1, 2, 3, 4]$$

$$\frac{\partial f}{\partial b} = 2(g(\boldsymbol{x}) - y)(g(\boldsymbol{x}) - g(\boldsymbol{x})^2)$$

It can be seen that $\frac{\partial f}{\partial w_i}$ depends on x_i. If $x_i = 0$, $\frac{\partial f}{\partial w_i} = 0$. Therefore, when we have a lot of training samples and they are sparse on dimension i, then the weight gradient vectors will be sparse on dimension i.

	Alice	Bob	Chase	David	Emily	Fred	George	Henry
The Godfather	5	3	2	1	5	4	5	2
Inception		5	3	3	4	5	5	1
The Matrix	4	3		4		5		4
Braveheart			5		5	1		
The Lion King	3		2		5			2
Blade Runner				2			1	
Dune				5				
Seven Samurai		4			3			
Whiplash				5				

The left figure shows an example of sparse data. It is a rating matrix: each row represents a movie, each column represents a user, each element is a rating of a user to a movie. Each rating is between 1 and 5.

We consider each column as a training sample. The ratings of Dune and Whiplash are very sparse. This may be because only a few users watched them. We usually fill those missing ratings with 0.

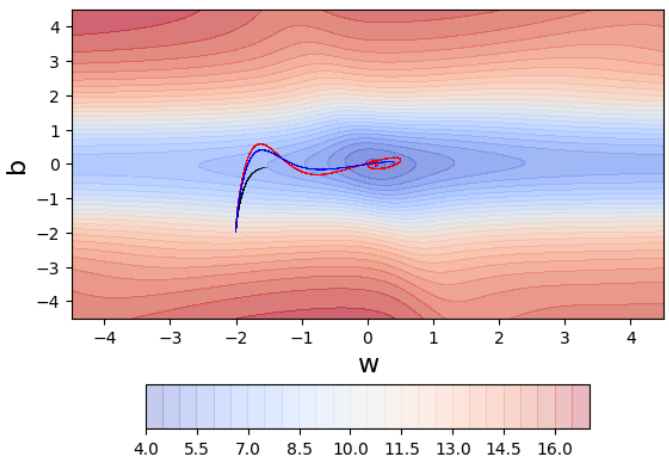
Sparse Gradient

What will happen if the gradients are sparse? Sparse gradient will slow down the convergence.

For easy visualization, we select two dimensions w_1 and b. Their gradient are calculated by

$$\frac{\partial f}{\partial w_1} = 2(g(\boldsymbol{x}) - y)(g(\boldsymbol{x}) - g(\boldsymbol{x})^2)x_1 \qquad \frac{\partial f}{\partial b} = 2(g(\boldsymbol{x}) - y)(g(\boldsymbol{x}) - g(\boldsymbol{x})^2)$$

Suppose we have a set of $(\boldsymbol{x}, y)$ pairs and roughly 80% of the x_1 in these pairs are 0. When we use gradient descent on f, the trajectory of (w_1, b) would be like:



Source: [8]

Black: GD
Red: Momentum
Blue: Nesterov

Note that the convergence is faster on b dimension. Because most gradients of w_1 dimension is 0, the convergence will be slow on w_1 dimension.

AdaGrad

The main idea of Adaptive Learning Rate is to assign different step size for each dimension of the gradient vector. The dense dimension will be assigned smaller step size while the sparse dimension will be assigned larger step size.

AdaGrad is the first algorithm based on Adaptive Learning Rate, proposed by Duchi et al [10].

Algorithm 3.1 (AdaGrad): Let $\{\alpha_t\}_{t\in\mathbb{N}}$ be a sequence of step sizes. Initialize $v_{-1} = 0$. AdaGrad defines a sequence $\{x_t\}_{t\in\mathbb{N}} \in \mathbb{R}^d$ satisfying for every $t \in \mathbb{N}$,

$$v_t = v_{t-1} + (\nabla f_t(x_t))^2$$
$$x_{t+1} = x_t - \alpha_t \frac{\nabla f_t(x_t)}{\sqrt{v_t + \epsilon}}$$

Where the square $(\cdot)^2$, square root $\sqrt{\cdot}$ and division $\frac{\cdot}{\cdot}$ of vectors are all element-wise [4]. ϵ is a nonzero vector with trivial values to prevent the calculation failure when some elements of v_t are 0.

[4] These notations are also used for Algorithm 3.2 and Algorithm 3.3.

Explanation of AdaGrad

Consider the following term in algorithm 3.1:

$$\frac{\nabla f_t(x_t)}{\sqrt{v_t}} = \frac{\nabla f_t(x_t)}{\sqrt{\sum_{j=0}^{t} \nabla f_j(x_j)^2}}$$

Let $g_t = \nabla f_t(x_t)$, then $g_t = [g_{t,1}, g_{t,2}, ..., g_{t,d}]^T$, where $g_{t,i}$ $(i \in \{1, 2, ..., d\})$ is the element of the i-th dimension of g_t.

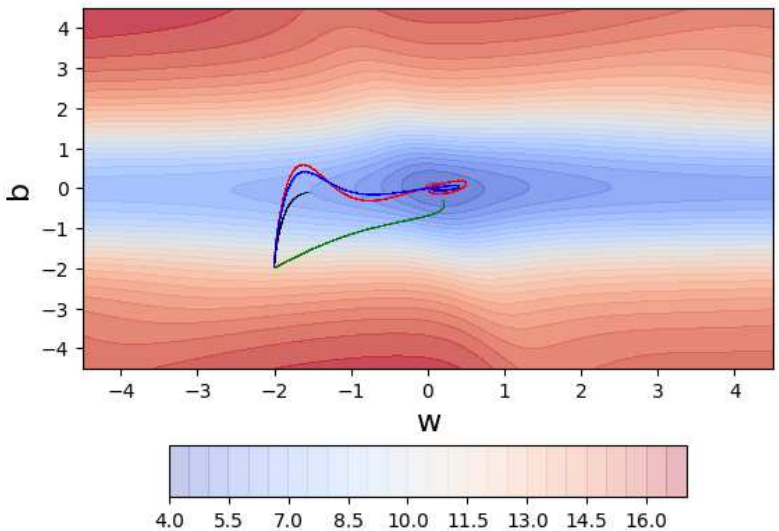
Thus, the element of the i-th dimension of $\frac{g_t}{\sqrt{v_t}}$ is

$$\frac{g_{t,i}}{\sqrt{\sum_{j=0}^{t} g_{j,i}^2}}$$

If the i-th dimension is sparse, most of $g_{j,i}$s for $j = 0, 1, ..., t$ will be 0, making the scale $\frac{1}{\sqrt{\sum_{j=0}^{t} g_{j,i}^2}}$ large for $g_{t,i}$. Otherwise, if the i-th dimension is dense, the scale $\frac{1}{\sqrt{\sum_{j=0}^{t} g_{j,i}^2}}$ will be small. In this way we assign different step size for each dimension.

Explanation of AdaGrad

We still use the example in sparse gradient. After AdaGrad is applied, the trajectory of (w_1, b) would be like:



Source: [8]

Black: GD
Red: Momentum
Blue: Nesterov
Green: AdaGrad

We can see that the trajectory of AdaGrad go towards the minimum, which means dimension w_1 and dimension d converge at roughly the same rate.

RMSProp

The problem of AdaGrad is, since $\frac{1}{\sqrt{v_t}} = \frac{1}{\sqrt{\sum_{j=0}^{t} \nabla f_j(x_j)^2}}$, the step size of each dimension decreases with t, which slows down the convergence.

Algorithm 3.2 (RMSProp)[5]: Let $\{\alpha_t\}_{t\in\mathbb{N}}$ be a sequence of step sizes. Initialize $v_{-1} = 0$. $\beta \in [0, 1]$ be the momentum parameter. RMSProp defines a sequence $\{\alpha_t\}_{t\in\mathbb{N}} \in \mathbb{R}^d$ satisfying for every $t \in \mathbb{N}$,

$$v_t = \beta v_{t-1} + (1-\beta)(\nabla f_t(x_t))^2$$
$$x_{t+1} = x_t - \alpha_t \frac{\nabla f_t(x_t)}{\sqrt{v_t + \epsilon}}$$

Note that RMSProp defines v_t as exponential moving average of $\nabla f_j(x_j)^2$s:

$$v_t = \beta v_{t-1} + (1-\beta)(\nabla f_t(x_t))^2 = (1-\beta)\sum_{j=0}^{t} \beta^{t-j} \nabla f_j(x_j)^2$$

So $\frac{1}{\sqrt{v_t}}$ will only be affected by the latest few $\nabla f_j(x_j)^2$ terms, which prevents it from decreasing with t.

[5] RMSProp is an unpublished work by Geoffrey Hinton on his Coursera class. http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

Adam

Adam is proposed by Kingma and Ba [11]. This paper has been cited 148k times by July 2023. In practice, Adam is the default optimization algorithm to train a large neural network for computer vision or natural language processing tasks.

Algorithm 3.2 (Adam): Let $\{\alpha_t\}_{t\in\mathbb{N}}$ be a sequence of step sizes, and $\beta_1, \beta_2 \in [0, 1)$ be the momentum parameters. Initialize $v_0 = 0, m_0 = 0$ and the starting point $x_1 \in \mathbb{R}^d$. Adam defines a sequence $\{x_t\} \in \mathbb{R}^d$ satisfying for every $t \geq 1$,

$$
\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1-\beta_1)\nabla f_t(x_t) \\
v_t &= \beta_2 v_{t-1} + (1-\beta_2)(\nabla f_t(x_t))^2 \\
\hat{m}_t &= \frac{m_t}{1-\beta_1^t}\ , \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t} \\
x_{t+1} &= x_t - \alpha_t \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}
\end{aligned}
$$

AdaGrad updates x by $x_{t+1} = x_t - \alpha_t \frac{\nabla f_t(x_t)}{\sqrt{v_t+\epsilon}}$, where $v_t = v_{t-1} + (\nabla f_t(x_t))^2$. RMSProp improves AdaGrad by making v_t a momentum term: $v_t = \beta_2 v_{t-1} + (1-\beta_2)(\nabla f_t(x_t))^2$. Adam in further improves RMSProp by making $\nabla f_t(x_t)$ a momentum term $m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla f_t(x_t)$.

Bias Correction in Adam

Adam uses $\hat{m}_t, \hat{v}_t$ instead of m_t, v_t to calculate the gradient for bias correction.

Use m_k for example, since

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla f_t(x_t) = (1-\beta_1)\sum_{i=1}^{t} \beta_1^{t-i} \nabla f_t(x_t)$$

Taking expectation on both sides

$$\mathbb{E}[m_t] = (1-\beta_1)\sum_{i=1}^{t} \beta_1^{t-i} \cdot \mathbb{E}[\nabla f_t(x_t)] = (1-\beta_1^t)\mathbb{E}[\nabla f_t(x_t)]$$

By taking $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$, we get

$$\mathbb{E}[\hat{m}_t] = \frac{1}{1-\beta_1^t}\mathbb{E}[m_t] = \mathbb{E}[\nabla f_t(x_t)]$$

Thus $\hat{m}_k$ is an unbiased estimator of $\mathbb{E}[\nabla f_t(x_t)]$. Similarly, Thus $\hat{v}_t$ is an unbiased estimator of $\mathbb{E}[\nabla f_t(x_t)^2]$.

About the Convergence of Adam

(1) Kingma and Ba proved that Adam converges at $O(\frac{1}{T})$ rate, see Theorem 4.1 in [11]. Unfortunately, this proof in wrong. There are few errors in the proof, one can be found here[6].

(2) Reddi et al [12] proved that there exists an online learning problem that Adam does not converge, and the main issue lies in the following quantity

$$\Gamma_{t+1} = \frac{\sqrt{v_{t+1}}}{\alpha_{t+1}} - \frac{\sqrt{v_t}}{\alpha_t}$$

Note that $\frac{\alpha_t}{\sqrt{v_t}}$ is the learning rate vector of step t. If $\Gamma_{t+1} \succeq 0$ [7] for any t, $\frac{\sqrt{v_t}}{\alpha_t}$ will be non-decreasing, and the learning rate $\frac{\alpha_t}{\sqrt{v_t}}$ will be non-increasing.

However, there is no guarantee of $\Gamma_{t+1} \succeq 0$. Since $v_{t+1} = \beta_2 v_t + (1-\beta_2)\nabla f_t(x_t)^2$ and $\beta_2 > 0$, we may have $v_{t+1,i} < v_{t,i}$ for some dimension i. Based on this, one can construct a online learning problem that makes Adam not converge.

---

[6] https://math.stackexchange.com/questions/4271692/a-possible-bug-in-a-highly-cited-paper-adam-gradient-descent

[7] For a vector $x \in \mathbb{R}^d$, $x \succeq 0$ means $x_i \geq 0$ for all i.

About the Convergence of Adam

We use the example given by Theorem 1 in [12]. Consider the following online learning problem: Suppose $x \in [-10, 10]$ and $t \geq 1$, the functions are defined as

$$f_t(x) = \begin{cases} Cx, & \text{for } t \bmod 3 = 1 \\ -x, & \text{otherwise} \end{cases}$$

where $C > 2$.

It is obvious that $x^* = -10$. This is because

$$\sum_{t=1}^{T} f_t(x) = (\lfloor \frac{T}{3}(C-2) \rfloor + A_T)x$$

where

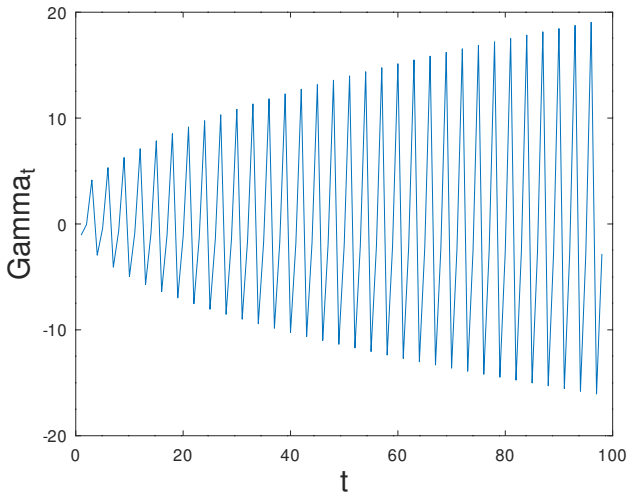
$$A_T = \begin{cases} 0 & T \bmod 3 = 0 \\ C & T \bmod 3 = 1 \\ C-1 & T \bmod 3 = 2 \end{cases}$$

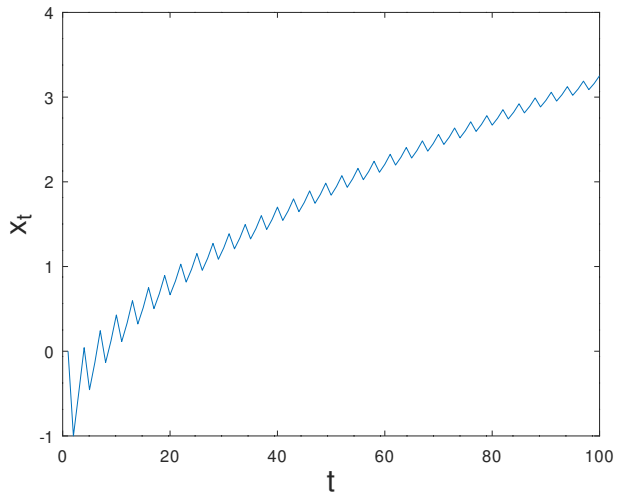
$\lfloor \frac{T}{3}(C-2) \rfloor + A_T$ is always positive, so $x^* = -10$.

About the Convergence of Adam

Suppose $\beta_1 = 0$, $\beta_2 = \frac{1}{1+C^2}$, $\alpha_t = \alpha/\sqrt{t}$ where $\alpha \leq \sqrt{1-\beta_2}$. It can be proved that by these settings, Adam will not converge to x^* on this online learning problem.

I would like to show how Adam performs intuitively. Let $C = 3, x_1 = 0$, then the results are shown below





The left Figure shows the trend of Γ_t. The right figure shows the trend of x_t.

In this case, $\Gamma_t \succeq 0$ does not hold, and x_t does not move towards x^*.

(3) One newest paper of 2022 by Defossez et al [13] proved that when used with default parameters, Adam does not converge. However, Adam moves away from initialization point faster than AdaGrad, which might explain its practical success. Adam is to Adagrad like constant step size SGD is to decaying step size SGD.

A Fixed Adam Algorithm

To solve the non-convergence problem of Adam, Reddi et al [12] proposed a fixed Adam algorithm – AMSGrad.

Algorithm 3.3 (AMSGrad): Let $t \in \mathbb{N}$ and $t \geq 1$. Let $\{\alpha_t\}$ be a sequence of non-increasing step sizes, $\{\beta_{1t}\} \in [0, 1)$ be a set of non-increasing momentum parameters, and $\beta_2 \in [0, 1)$ be another momentum parameter. Initialize $v_0 = 0, m_0 = 0$ and the starting point $x_1 \in \mathbb{R}^d$. Adam defines a sequence $\{x_t\} \in \mathbb{R}^d$ satisfying

$$
\begin{aligned}
m_t &= \beta_{1t} m_{t-1} + (1 - \beta_{1t}) \nabla f_t(x_t) \\
v_t &= \beta_2 v_{t-1} + (1 - \beta_2) (\nabla f_t(x_t))^2 \\
\hat{v}_t &= \max(\hat{v}_{t-1}, v_t) \\
x_{t+1} &= x_t - \alpha_t \frac{m_t}{\sqrt{\hat{v}_t + \epsilon}}
\end{aligned}
$$

Here $\max(\cdot)$ is also a element-wise maximum operation.

By taking $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$, we have $\hat{v}_t \succeq \hat{v}_{t-1}$. Since $\alpha_t \leq \alpha_{t-1}$, we must have

$$
\Gamma_t = \frac{\sqrt{\hat{v}_t}}{\alpha_t} - \frac{\sqrt{\hat{v}_{t-1}}}{\alpha_{t-1}} \succeq 0
$$

The authors removed the bias correction step in AMSGrad for simplicity, but AMSGrad can also be applied to the bias correction case.

Notations

Before we go to the proof, I would like to clarify some notations:

(1) For any vector $x \in \mathbb{R}^d$, we denote the i-th element of x as x_i. For example, the i-th element of m_t is denoted as $m_{t,i}$.

(2) For the vector $\hat{v}_t = [\hat{v}_{t,1}, \hat{v}_{t,2}, ..., \hat{v}_{t,d}]$, we denote $\hat{V}_t = \text{diag}[\hat{v}_{t,1}, \hat{v}_{t,2}, ..., \hat{v}_{t,d}]$ as a diagonal matrix. For any $p \in \mathbb{R}$, $\hat{V}_t^p = \text{diag}[\hat{v}_{t,1}^p, \hat{v}_{t,2}^p, ..., \hat{v}_{t,d}^p]$. Thus, for any vector $x \in \mathbb{R}^d$ the element-wise division $\frac{x}{\hat{v}_t}$ can be written as the matrix-vector multiplication $\hat{V}^{-1}x$.

Convergence Analysis of AMSGrad

Theorem 3.4: Let $\{x_t\}$ and $\{v_t\}$ be the sequence obtained from Algorithm 3.3 (AMSGrad). Let $\alpha_t = \alpha/\sqrt{t}$, $\beta_1 = \beta_{11}$ and $\beta_{1(t+1)} \leq \beta_{1t}$ for $t \geq 1$. Choose β_2 such that $\gamma = \beta_1/\sqrt{\beta_2} < 1$. Assume that $\|\nabla f_t(x)\|_\infty \leq G_\infty$ and $\|x_t - x^*\| \leq D_\infty$ for any $t \geq 1$. For $\{x_t\}$ generated using the Algorithm 3.3 (AMSGrad), we have the following bound on regret

$$R(T) \leq \frac{D_\infty^2 \sqrt{T}}{\alpha(1-\beta_1)} \sum_{i=1}^{d} \hat{v}_{T,i}^{1/2} + \frac{D_\infty^2}{2(1-\beta_1)} \sum_{t=1}^{T} \sum_{i=1}^{d} \frac{\beta_{1t} \hat{v}_{t,i}^{1/2}}{\alpha_t} + \frac{\alpha\sqrt{1+\ln T}}{(1-\beta_1)^2(1-\gamma)\sqrt{1-\beta_2}} \sum_{i=1}^{d} \|g_{1:T,i}\|$$

Proof: We denote $g_t = \nabla f_t(x_t)$. Since $x_{t+1} = x_t - \alpha_t \hat{V}_t^{-1/2} m_t$, we have

$$\begin{aligned}
\|\hat{V}_t^{1/4}(x_{t+1} - x^*)\|^2 &= \|\hat{V}_t^{1/4}(x_t - \alpha_t \hat{V}_t^{-1/2} m_t - x^*)\|^2 \\
&= \|\hat{V}_t^{1/4}(x_t - x^*)\|^2 + \alpha_t^2 \|\hat{V}_t^{-1/4} m_t\|^2 - 2\alpha_t \langle m_t, x_t - x^* \rangle \\
&= \|\hat{V}_t^{1/4}(x_t - x^*)\|^2 + \alpha_t^2 \|\hat{V}_t^{-1/4} m_t\|^2 \\
&\quad - 2\alpha_t \langle \beta_{1t} m_{t-1} + (1-\beta_{1t}) g_t, x_t - x^* \rangle
\end{aligned}$$

Rearranging the above equality, we have

$$\begin{aligned}
\langle g_t, x_t - x^* \rangle = \frac{1}{2\alpha_t(1-\beta_{1t})} \left[\|\hat{V}_t^{1/4}(x_t - x^*)\|^2 - \|\hat{V}_t^{1/4}(x_{t+1} - x^*)\|^2 \right] + \\
\frac{\alpha_t}{2(1-\beta_{1t})} \|\hat{V}_t^{-1/4} m_t\|^2 - \frac{\beta_{1t}}{1-\beta_{1t}} \langle m_{t-1}, x_t - x^* \rangle
\end{aligned}$$

Since

$$
\begin{aligned}
-\frac{\beta_{1t}}{1-\beta_{1t}}\langle m_{t-1}, x_t - x^*\rangle &\le \frac{\beta_{1t}}{1-\beta_{1t}}|\langle m_{t-1}, x_t - x^*\rangle| \\
&= \frac{\beta_{1t}}{1-\beta_{1t}}|\langle \alpha_t^{1/2}\hat{V}_{t-1}^{-1/4} m_{t-1}, \alpha_t^{-1/2}\hat{V}_{t-1}^{1/4}(x_t - x^*)\rangle| \\
&\le \frac{\beta_{1t}}{1-\beta_{1t}}|\left(\alpha_t^{1/2}\|\hat{V}_{t-1}^{-1/4} m_{t-1}\|\right)\left(\alpha_t^{-1/2}\|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|\right) \qquad \text{[Cauchy]} \\
&\le \frac{\beta_{1t}}{1-\beta_{1t}}\left(\frac{\alpha_t}{2}\|\hat{V}_{t-1}^{-1/4} m_{t-1}\|^2 + \frac{1}{2\alpha_t}\|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2\right) \quad [ab \le \frac{a^2+b^2}{2}]
\end{aligned}
$$

We have

$$
\langle g_t, x_t - x^*\rangle \le \frac{1}{2\alpha_t(1-\beta_{1t})}\left[\|\hat{V}_t^{1/4}(x_t - x^*)\|^2 - \|\hat{V}_t^{1/4}(x_{t+1} - x^*)\|^2\right] +
$$

$$
\frac{\alpha_t}{2(1-\beta_{1t})}\|\hat{V}_t^{-1/4} m_t\|^2 + \frac{\alpha_t\beta_{1t}}{2(1-\beta_{1t})}\|\hat{V}_{t-1}^{-1/4} m_{t-1}\|^2 + \frac{\beta_{1t}}{2\alpha_t(1-\beta_{1t})}\|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2
$$

Therefore,

$$R(T) = \sum_{t=1}^{T}[f_t(x_t) - f_t(x^*)] \leq \sum_{t=1}^{T}\langle g_t, x_t - x^*\rangle \qquad \text{[by the convexity of } f_t\text{]}$$

$$\leq \sum_{t=1}^{T}\Bigg[\frac{1}{2\alpha_t(1-\beta_{1t})}\left[\|\hat{V}_t^{1/4}(x_t - x^*)\|^2 - \|\hat{V}_t^{1/4}(x_{t+1} - x^*)\|^2\right] +$$

$$\frac{\alpha_t}{2(1-\beta_{1t})}\|\hat{V}_t^{-1/4}m_t\|^2 + \frac{\alpha_t\beta_{1t}}{2(1-\beta_{1t})}\|\hat{V}_{t-1}^{-1/4}m_{t-1}\|^2 +$$

$$\frac{\beta_{1t}}{2\alpha_t(1-\beta_{1t})}\|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2\Bigg] \tag{21}$$

Since $m_0 = 0$, we have

$$\sum_{t=1}^{T}\left[\frac{\alpha_t}{2(1-\beta_{1t})}\|\hat{V}_t^{-1/4}m_t\|^2 + \frac{\alpha_t\beta_{1t}}{2(1-\beta_{1t})}\|\hat{V}_{t-1}^{-1/4}m_{t-1}\|^2\right]$$

$$= \frac{\alpha_T}{2(1-\beta_{1T})}\|\hat{V}_T^{-1/4}m_T\|^2 + \sum_{t=1}^{T-1}\left[\frac{\alpha_t}{2(1-\beta_{1t})}\|\hat{V}_t^{-1/4}m_t\|^2 + \frac{\alpha_{t+1}\beta_{1(t+1)}}{2(1-\beta_{1(t+1)})}\|\hat{V}_t^{-1/4}m_t\|^2\right]$$

$$\leq \frac{\alpha_T}{2(1-\beta_{1T})}\|\hat{V}_T^{-1/4}m_T\|^2 + \sum_{t=1}^{T-1}\left[\frac{\alpha_t}{2(1-\beta_{1t})}\|\hat{V}_t^{-1/4}m_t\|^2 + \frac{\alpha_t}{2(1-\beta_{1t})}\|\hat{V}_t^{-1/4}m_t\|^2\right]$$

$$\leq \sum_{t=1}^{T}\frac{\alpha_t}{(1-\beta_{1t})}\|\hat{V}_t^{-1/4}m_t\|^2$$

Thus, (21) can be written as

$$R(T) \leq \sum_{t=1}^{T} \Bigg[\frac{1}{2\alpha_t(1-\beta_{1t})} \Big[\|\hat{V}_t^{1/4}(x_t - x^*)\|^2 - \|\hat{V}_t^{1/4}(x_{t+1} - x^*)\|^2 \Big] + \frac{\alpha_t}{(1-\beta_{1t})} \|\hat{V}_t^{-1/4} m_t\|^2 + \frac{\beta_{1t}}{2\alpha_t(1-\beta_{1t})} \|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2 \Bigg] \tag{22}$$

We would like to bound the term $\alpha_t \|\hat{V}_t^{-1/4} m_t\|^2$ first.

Lemma 3.5: For the parameter settings and conditions assumed in Theorem 3.4, we have

$$\sum_{t=1}^{T} \alpha_t \|\hat{V}_t^{-1/4} m_t\|^2 \leq \frac{\alpha\sqrt{1+\ln T}}{(1-\beta_1)(1-\gamma)\sqrt{1-\beta_2}} \sum_{i=1}^{d} \|g_{1:T,i}\|$$

Proof:

$$\sum_{t=1}^{T} \alpha_t \|\hat{V}_t^{-1/4} m_t\|^2 = \sum_{t=1}^{T-1} \alpha_t \|\hat{V}_t^{-1/4} m_t\|^2 + \alpha_T \sum_{i=1}^{d} \frac{m_{T,i}^2}{\sqrt{\hat{v}_{T,i}}}$$

$$\leq \sum_{t=1}^{T-1} \alpha_t \|\hat{V}_t^{-1/4} m_t\|^2 + \alpha_T \sum_{i=1}^{d} \frac{m_{T,i}^2}{\sqrt{v_{T,i}}} \qquad [v_{T,i} \leq \hat{v}_{T,i}]$$

$$= \sum_{t=1}^{T-1} \alpha_t \|\hat{V}_t^{-1/4} m_t\|^2 + \frac{\alpha}{\sqrt{T}} \sum_{i=1}^{d} \frac{\left(\sum_{j=1}^{T}(1-\beta_{1j})(\prod_{k=1}^{T-j}\beta_{1(T-k+1)})g_{j,i}\right)^2}{\sqrt{(1-\beta_2)\sum_{j=1}^{T}\beta_2^{T-j} g_{j,i}^2}}$$

$$= \sum_{t=1}^{T-1} \alpha_t \|\hat{V}_t^{-1/4} m_t\|^2 +$$

$$\frac{\alpha}{\sqrt{T}} \sum_{i=1}^{d} \frac{(\sum_{j=1}^{T} [\sqrt{(1-\beta_{1j})(\prod_{k=1}^{T-j} \beta_{1(T-k+1)})}][\sqrt{(1-\beta_{1j})(\prod_{k=1}^{T-j} \beta_{1(T-k+1)})} g_{j,i}])^2}{\sqrt{(1-\beta_2)\sum_{j=1}^{T} \beta_2^{T-j} g_{j,i}^2}}$$

$$\leq \sum_{t=1}^{T-1} \alpha_t \|\hat{V}_t^{-1/4} m_t\|^2 +$$

$$\frac{\alpha}{\sqrt{T}} \sum_{i=1}^{d} \frac{[\sum_{j=1}^{T} (1-\beta_{1j})(\prod_{k=1}^{T-j} \beta_{1(T-k+1)})][\sum_{j=1}^{T} (1-\beta_{1j})(\prod_{k=1}^{T-j} \beta_{1(T-k+1)}) g_{j,i}^2]}{\sqrt{(1-\beta_2)\sum_{j=1}^{T} \beta_2^{T-j} g_{j,i}^2}}$$

[Cauchy]

$$\leq \sum_{t=1}^{T-1} \alpha_t \|\hat{V}_t^{-1/4} m_t\|^2 + \frac{\alpha}{\sqrt{T}} \sum_{i=1}^{d} \frac{[\sum_{j=1}^{T} \beta_1^{T-j}][\sum_{j=1}^{T} \beta_1^{T-j} g_{j,i}^2]}{\sqrt{(1-\beta_2)\sum_{j=1}^{T} \beta_2^{T-j} g_{j,i}^2}}$$

$$[1-\beta_{1j} \leq 1, \prod_{k=1}^{T-j} \beta_{1(T-k+1)} \leq \beta_1^{T-j}]$$

$$\leq \sum_{t=1}^{T-1} \alpha_t \|\hat{V}_t^{-1/4} m_t\|^2 + \frac{\alpha}{(1-\beta_1)\sqrt{T}} \sum_{i=1}^{d} \frac{\sum_{j=1}^{T} \beta_1^{T-j} g_{j,i}^2}{\sqrt{(1-\beta_2)\sum_{j=1}^{T} \beta_2^{T-j} g_{j,i}^2}} \quad [\sum_{j=1}^{T} \beta_1^{T-j} \leq \frac{1}{1-\beta_1}]$$

$$= \sum_{t=1}^{T-1} \alpha_t \|\hat{V}_t^{-1/4} m_t\|^2 + \frac{\alpha}{(1-\beta_1)\sqrt{T(1-\beta_2)}} \sum_{i=1}^{d} \frac{\sum_{j=1}^{T} \beta_1^{T-j} g_{j,i}^2}{\sqrt{\sum_{j=1}^{T} \beta_2^{T-j} g_{j,i}^2}}$$

$$\leq \sum_{t=1}^{T-1} \alpha_t \|\hat{V}_t^{-1/4} m_t\|^2 + \frac{\alpha}{(1-\beta_1)\sqrt{T(1-\beta_2)}} \sum_{i=1}^{d} \sum_{j=1}^{T} \frac{\beta_1^{T-j} g_{j,i}^2}{\sqrt{\beta_2^{T-j} g_{j,i}^2}} \quad [\beta_2^{T-j} g_{j,i}^2 \leq \sum_{j=1}^{T} \beta_2^{T-j} g_{j,i}^2]$$

$$= \sum_{t=1}^{T-1} \alpha_t \|\hat{V}_t^{-1/4} m_t\|^2 + \frac{\alpha}{(1-\beta_1)\sqrt{T(1-\beta_2)}} \sum_{i=1}^{d} \sum_{j=1}^{T} \gamma^{T-j} |g_{j,i}|$$

Therefore, by induction,

$$\begin{aligned}
\sum_{t=1}^{T} \alpha_t \|\hat{V}_t^{-1/4} m_t\|^2 &\leq \sum_{t=1}^{T} \frac{\alpha}{(1-\beta_1)\sqrt{t(1-\beta_2)}} \sum_{i=1}^{d} \sum_{j=1}^{t} \gamma^{t-j} |g_{j,i}| \\
&= \frac{\alpha}{(1-\beta_1)\sqrt{(1-\beta_2)}} \sum_{i=1}^{d} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sum_{j=1}^{t} \gamma^{t-j} |g_{j,i}| \\
&= \frac{\alpha}{(1-\beta_1)\sqrt{(1-\beta_2)}} \sum_{i=1}^{d} \sum_{t=1}^{T} |g_{t,i}| \sum_{j=t}^{T} \frac{\gamma^{j-t}}{\sqrt{j}} \qquad (23) \\
&\leq \frac{\alpha}{(1-\beta_1)\sqrt{(1-\beta_2)}} \sum_{i=1}^{d} \sum_{t=1}^{T} |g_{t,i}| \sum_{j=t}^{T} \frac{\gamma^{j-t}}{\sqrt{t}}
\end{aligned}$$

$$\leq \frac{\alpha}{(1-\beta_1)\sqrt{(1-\beta_2)}} \sum_{i=1}^{d} \sum_{t=1}^{T} |g_{t,i}| \frac{1}{(1-\gamma)\sqrt{t}}$$

$$= \frac{\alpha}{(1-\beta_1)(1-\gamma)\sqrt{(1-\beta_2)}} \sum_{i=1}^{d} \sum_{t=1}^{T} |g_{t,i}| \frac{1}{\sqrt{t}}$$

$$= \frac{\alpha}{(1-\beta_1)(1-\gamma)\sqrt{(1-\beta_2)}} \sum_{i=1}^{d} \|g_{1:T,i}\| \sqrt{\sum_{t=1}^{T} \frac{1}{t}} \tag{24}$$

$$\leq \frac{\alpha\sqrt{\ln T + 1}}{(1-\beta_1)(1-\gamma)\sqrt{(1-\beta_2)}} \sum_{i=1}^{d} \|g_{1:T,i}\| \tag{25}$$

(23) is because, we can list the terms of $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sum_{j=1}^{t} \gamma^{t-j} |g_{j,i}|$ as

$$\begin{array}{ccccccc}
|g_{1,i}| & & & & & & \\
\frac{\gamma|g_{1,i}|}{\sqrt{2}} & & \frac{|g_{2,i}|}{\sqrt{2}} & & & & \\
\frac{\gamma^2|g_{1,i}|}{\sqrt{3}} & & \frac{\gamma|g_{2,i}|}{\sqrt{3}} & & \frac{|g_{3,i}|}{\sqrt{3}} & & \\
\cdots & & & & & & \\
\frac{\gamma^{T-1}|g_{1,i}|}{\sqrt{T}} & & \frac{\gamma^{T-2}|g_{2,i}|}{\sqrt{T}} & & \frac{\gamma^{T-3}|g_{3,i}|}{\sqrt{T}} & \cdots & \frac{|g_{T,i}|}{\sqrt{T}}
\end{array}$$

Summing up these terms by columns, we get $\sum_{t=1}^{T} |g_{t,i}| \sum_{j=t}^{T} \frac{\gamma^{j-t}}{\sqrt{j}}$, thus

$$\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sum_{j=1}^{t} \gamma^{t-j} |g_{j,i}| = \sum_{t=1}^{T} |g_{t,i}| \sum_{j=t}^{T} \frac{\gamma^{j-t}}{\sqrt{j}}$$

(24) is because, by Cauchy Inequality,

$$\sum_{t=1}^{T} |g_{t,i}| \frac{1}{\sqrt{t}} = \sqrt{\sum_{t=1}^{T} |g_{t,i}|^2} \sqrt{\sum_{t=1}^{T} \frac{1}{t}} = \|g_{1:T,i}\| \sqrt{\sum_{t=1}^{T} \frac{1}{t}}$$

(25) is because

$$\sum_{t=1}^{T} \frac{1}{t} \leq \ln T + 1$$

Thus we proved Lemma 3.5.

We now return to the proof of Theorem 3.4. Using Lemma 3.5 and $1 - \beta_{1t} \geq 1 - \beta_1$, (22) can be written as

$$R(T) \leq \sum_{t=1}^{T} \left[\frac{1}{2\alpha_t(1-\beta_{1t})} \left[\|\hat{V}_t^{1/4}(x_t - x^*)\|^2 - \|\hat{V}_t^{1/4}(x_{t+1} - x^*)\|^2 \right] + \right.$$
$$\left. \frac{\beta_{1t}}{2\alpha_t(1-\beta_{1t})} \|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2 \right] + \frac{\alpha\sqrt{1+\ln T}}{(1-\beta_1)^2(1-\gamma)\sqrt{1-\beta_2}} \sum_{i=1}^{d} \|g_{1:T,i}\|$$

$$= \frac{1}{2\alpha_1(1-\beta_1)}\|\hat{V}_1^{1/4}(x_1 - x^*)\|^2 + \frac{1}{2}\sum_{t=2}^{T}\left[\frac{\|\hat{V}_t^{1/4}(x_t - x^*)\|^2}{\alpha_t(1-\beta_{1t})} - \frac{\|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2}{\alpha_{t-1}(1-\beta_{1(t-1)})}\right] +$$

$$\sum_{t=1}^{T}\frac{\beta_{1t}}{2\alpha_t(1-\beta_{1t})}\|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2 + \frac{\alpha\sqrt{1+\ln T}}{(1-\beta_1)^2(1-\gamma)\sqrt{1-\beta_2}}\sum_{i=1}^{d}\|g_{1:T,i}\| \tag{26}$$

For any vector $x \in \mathbb{R}^d$, $\|\hat{V}_t x\|^2 \geq \|\hat{V}_{t-1}x\|^2$. This is because $\hat{v}_{t,i} > \hat{v}_{t-1,i} > 0$ for any i, t. Thus

$$\|\hat{V}_t x\|^2 = \sum_{i=1}^{d}\hat{v}_{t,i}^2 x_i^2 \geq \sum_{i=1}^{d}\hat{v}_{t-1,i}^2 x_i^2 = \|\hat{V}_{t-1}x\|^2$$

Therefore,

$$\|\hat{V}_t^{1/4}(x_t - x^*)\|^2 \geq \|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2$$

Since $\alpha_t \leq \alpha_{t-1}$, we have

$$\frac{\|\hat{V}_t^{1/4}(x_t - x^*)\|^2}{\alpha_t} \geq \frac{\|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2}{\alpha_{t-1}}$$

Since $\beta_{1t} < \beta_{1(t-1)} \leq \beta_1 \Rightarrow \frac{1}{1-\beta_1} - \frac{1}{1-\beta_{t-1}} \leq \frac{1}{1-\beta_1} - \frac{1}{1-\beta_t}$, we have

$$\left(\frac{1}{1-\beta_1} - \frac{1}{1-\beta_t}\right)\frac{\|\hat{V}_t^{1/4}(x_t - x^*)\|^2}{\alpha_t} \geq \left(\frac{1}{1-\beta_1} - \frac{1}{1-\beta_{t-1}}\right)\frac{\|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2}{\alpha_{t-1}} \iff$$

$$\frac{1}{1-\beta_1}\left(\frac{\|\hat{V}_t^{1/4}(x_t - x^*)\|^2}{\alpha_t} - \frac{\|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2}{\alpha_{t-1}}\right) \geq \frac{\|\hat{V}_t^{1/4}(x_t - x^*)\|^2}{\alpha_t(1-\beta_t)} - \frac{\|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2}{\alpha_{t-1}(1-\beta_{t-1})} \tag{27}$$

Plugging (27) in (26), we get

$$\begin{aligned}
R(T) &\leq \frac{1}{2\alpha_1(1-\beta_1)}\|\hat{V}_1^{1/4}(x_1 - x^*)\|^2 + \\
&\frac{1}{2(1-\beta_1)}\sum_{t=2}^{T}\left[\frac{\|\hat{V}_t^{1/4}(x_t - x^*)\|^2}{\alpha_t} - \frac{\|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2}{\alpha_{t-1}}\right] + \\
&\sum_{t=1}^{T}\frac{\beta_{1t}}{2\alpha_t(1-\beta_{1t})}\|\hat{V}_{t-1}^{1/4}(x_t - x^*)\|^2 + \frac{\alpha\sqrt{1+\ln T}}{(1-\beta_1)^2(1-\gamma)\sqrt{1-\beta_2}}\sum_{i=1}^{d}\|g_{1:T,i}\| \\
&\leq \frac{1}{2\alpha_1(1-\beta_1)}\sum_{i=1}^{d}\hat{v}_{1,i}^{1/2}(x_{1,i} - x_i^*)^2 + \frac{1}{2(1-\beta_1)}\sum_{t=2}^{T}\sum_{i=1}^{d}(x_{1,i} - x_i^*)^2\left[\frac{\hat{v}_{t,i}^{1/2}}{\alpha_t} - \frac{\hat{v}_{t-1,i}^{1/2}}{\alpha_{t-1}}\right] \\
&\frac{1}{2(1-\beta_1)}\sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\beta_{1t}(x_{1,i} - x_i^*)^2\hat{v}_{t,i}^{1/2}}{\alpha_t} + \frac{\alpha\sqrt{1+\ln T}}{(1-\beta_1)^2(1-\gamma)\sqrt{1-\beta_2}}\sum_{i=1}^{d}\|g_{1:T,i}\|
\end{aligned} \tag{28}$$

The second inequality of (28) uses the fact $2(1-\beta_{1t}) > 2(1-\beta_1)$. The original paper uses $2(1-\beta_{1t}) > (1-\beta_1)^2$. This is correct but the previous one gives a tighter bound.

Since $(x_{t,i} - x^*)^2 \leq D_\infty^2$, by (28) we have

$$
\begin{aligned}
R(T) &\leq \frac{1}{2\alpha_1(1-\beta_1)} \sum_{i=1}^{d} \hat{v}_{1,i}^{1/2} D_\infty^2 + \frac{1}{2(1-\beta_1)} \sum_{t=2}^{T} \sum_{i=1}^{d} D_\infty^2 \left[\frac{\hat{v}_{t,i}^{1/2}}{\alpha_t} - \frac{\hat{v}_{t-1,i}^{1/2}}{\alpha_{t-1}} \right] \\
&\quad \frac{1}{2(1-\beta_1)} \sum_{t=1}^{T} \sum_{i=1}^{d} \frac{\beta_{1t} D_\infty^2 \hat{v}_{t,i}^{1/2}}{\alpha_t} + \frac{\alpha\sqrt{1+\ln T}}{(1-\beta_1)^2(1-\gamma)\sqrt{1-\beta_2}} \sum_{i=1}^{d} \|g_{1:T,i}\| \\
&= \frac{D_\infty^2}{2\alpha_T(1-\beta_1)} \sum_{i=1}^{d} \hat{v}_{T,i}^{1/2} + \frac{D_\infty^2}{2(1-\beta_1)} \sum_{t=1}^{T} \sum_{i=1}^{d} \frac{\beta_{1t} \hat{v}_{t,i}^{1/2}}{\alpha_t} + \\
&\quad \frac{\alpha\sqrt{1+\ln T}}{(1-\beta_1)^2(1-\gamma)\sqrt{1-\beta_2}} \sum_{i=1}^{d} \|g_{1:T,i}\|
\end{aligned}
$$

By plugging in $\alpha_T = \alpha/\sqrt{T}$, we prove Theorem 3.4.

Convergence Rate of AMSGrad

Corollary 3.6: Consider the settings of Theorem 3.4. When taking $\beta_{1t} = \beta_1/t$, the $\bar{R}(T)$ generated by AMSGrad converges at a rate of $O(\sqrt{\frac{\ln T}{T}})$.

Proof: By Theorem 3.4,

$$R(T) \leq \frac{D_\infty^2 \sqrt{T}}{\alpha(1-\beta_1)} \sum_{i=1}^{d} \hat{v}_{T,i}^{1/2} + \frac{D_\infty^2}{2(1-\beta_1)} \sum_{t=1}^{T} \sum_{i=1}^{d} \frac{\beta_{1t} \hat{v}_{t,i}^{1/2}}{\alpha_t} + \frac{\alpha\sqrt{1+\ln T}}{(1-\beta_1)^2(1-\gamma)\sqrt{1-\beta_2}} \sum_{i=1}^{d} \|g_{1:T,i}\|$$

Since

$$\sum_{i=1}^{d} \hat{v}_{T,i}^{1/2} = \sum_{i=1}^{d} \max_t \{v_{t,i}^{1/2}\} = \sum_{i=1}^{d} \max_t \left\{ \sqrt{(1-\beta_2) \sum_{j=1}^{T} \beta_2^{T-j} g_j^2} \right\} \leq$$

$$\sum_{i=1}^{d} \sqrt{(1-\beta_2) \sum_{j=1}^{T} \beta_2^{T-j} G_\infty^2} \leq d G_\infty$$

$$\sum_{i=1}^{d} \|g_{1:T,i}\| = \sum_{i=1}^{d} \sqrt{\sum_{t=1}^{T} g_{t,i}^2} \leq \sum_{i=1}^{d} \sqrt{T G_\infty^2} = d G_\infty \sqrt{T}$$

$$\sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\beta_{1t}\hat{v}_{t,i}^{1/2}}{\alpha_t} = \frac{\beta_1}{\alpha}\sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\hat{v}_{t,i}^{1/2}}{\sqrt{t}} = \frac{\beta_1 dG_\infty}{\alpha}\sum_{t=1}^{T}\frac{1}{\sqrt{t}}$$
$$\leq \frac{\beta_1 dG_\infty}{\alpha}\int_0^T \frac{1}{\sqrt{t}}dt = \frac{\beta_1 dG_\infty}{\alpha}\ 2\sqrt{t}\Big|_0^T = \frac{2\beta_1 dG_\infty}{\alpha}\sqrt{T}$$

Therefore,

$$R(T) \leq \frac{D_\infty^2 dG_\infty\sqrt{T}}{\alpha(1-\beta_1)} + \frac{D_\infty^2\beta_1 dG_\infty\sqrt{T}}{\alpha(1-\beta_1)} + \frac{\alpha dG_\infty\sqrt{(1+\ln T)T}}{(1-\beta_1)^2(1-\gamma)\sqrt{1-\beta_2}} = O(\sqrt{T\ln T})$$

Thus,

$$\bar{R}(T) = \frac{R(T)}{T} = O(\sqrt{\frac{\ln T}{T}})$$

which means $\bar{R}(T) \to 0$ when $T \to \infty$.

References

[1] Handbook of Convergence Theorems for (Stochastic) Gradient Methods. https://gowerrobert.github.io/pdf/M2_statistique_optimisation/grad_conv.pdf. (Important!)

[2] Heavy Ball Momentum. https://www.cs.toronto.edu/~rgrosse/courses/csc2541_2021/slides/lec09.pdf

[3] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. USSR computational mathematics and mathematical physics, 4(5):1–17, 1964. http://vsokolov.org/courses/750/files/polyak64.pdf

[4] Yurii Evgen'evich Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In Doklady Akademii Nauk, volume 269, number 3, pages 543–547. Russian Academy of Sciences, 1983. https://vsokolov.org/courses/750/2018/files/nesterov.pdf

[5] Leon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. SIAM review, 60(2):223–311, 2018. https://arxiv.org/pdf/1606.04838.pdf

References

[6] Alekh Agarwal, Martin J Wainwright, Peter Bartlett, and Pradeep Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. Advances in Neural Information Processing Systems, 22, 2009. https://arxiv.org/pdf/1009.0571.pdf

[7] Mahmoud Assran and Mike Rabbat. On the convergence of nesterov's accelerated gradient method in stochastic settings. In International Conference on Machine Learning, pages 410–420. PMLR, 2020. https://arxiv.org/pdf/2002.12414.pdf

[8] Gradient Descent with Adaptive Learning Rate. Video: https://www.youtube.com/watch?v=FKCV76N9Ys0&list=PLyqSpQzTE6M9gCgajvQbc68Hk_JKGBAYT&index=43 Slides: http://www.cse.iitm.ac.in/~miteshk/CS7015/Slides/Teaching/pdf/Lecture5.pdf

[9] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In Proceedings of the 20th International Conference on Machine Learning, pages 928–936, 2003. https://people.eecs.berkeley.edu/~brecht/cs294docs/week1/03.Zinkevich.pdf

References

[10] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research, 12:2121–2159, 2011.
https://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf

[11] Diederik P. Kingma, Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. Proceedings of the 3rd International Conference on Learning Representations, 2015.
https://arxiv.org/pdf/1412.6980.pdf

[12] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In International Conference on Learning Representations, 2018.
https://arxiv.org/abs/1904.09237

[13] Alexandre Defossez, Leon Bottou, Francis Bach, and Nicolas Usunier. A simple convergence proof of Adam and Adagrad. Transactions on Machine Learning Research, 2022. https://openreview.net/pdf?id=ZPQhzTSWA7

[14] Yurii Nesterov. Introductory lectures on convex programming volume I: Basic course. Lecture notes 3.4 (1998): 5.