New Perspectives on the Polyak Stepsize: Surrogate Functions and Negative Results

Francesco Orabona and Ryan D'Orazio

Workshop on Regret, Optimization, and Games, 2025

Motivation

- This presentation is about better understanding previous results on Polyak stepsize

Motivation

- This presentation is about better understanding previous results on Polyak stepsize
- Polyak stepsize adapts to lipschitzness, strong convexity, smoothness, sharpness [e.g., Hazan&Kakade, 2019]

Motivation

- This presentation is about better understanding previous results on Polyak stepsize
- Polyak stepsize adapts to lipschitzness, strong convexity, smoothness, sharpness [e.g., Hazan&Kakade, 2019]
- But, **why?**

Motivation

- This presentation is about better understanding previous results on Polyak stepsize
- Polyak stepsize adapts to lipschitzness, strong convexity, smoothness, sharpness [e.g., Hazan&Kakade, 2019]
- But, **why?**
- In optimization for ML, a convergence rate is rarely predictive of reality, so the **why** is actually more important than a non-informative theorem

Motivation

- This presentation is about better understanding previous results on Polyak stepsize
- Polyak stepsize adapts to lipschitzness, strong convexity, smoothness, sharpness [e.g., Hazan&Kakade, 2019]
- But, **why?**
- In optimization for ML, a convergence rate is rarely predictive of reality, so the **why** is actually more important than a non-informative theorem
- In this talk I'll present a way to *truly understand and explain* the behaviour of the Polyak stepsize

Motivation

- This presentation is about better understanding previous results on Polyak stepsize
- Polyak stepsize adapts to lipschitzness, strong convexity, smoothness, sharpness [e.g., Hazan&Kakade, 2019]
- But, **why?**
- In optimization for ML, a convergence rate is rarely predictive of reality, so the **why** is actually more important than a non-informative theorem

- In this talk I'll present a way to *truly understand and explain* the behaviour of the Polyak stepsize
- No really new rates, but many negative results

The Challenge of Stepsize Selection

Gradient Descent (GD)

The foundational first-order optimization algorithm:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \nabla f(\boldsymbol{x}_t)$$

The Challenge of Stepsize Selection

Gradient Descent (GD)

The foundational first-order optimization algorithm:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \nabla f(\boldsymbol{x}_t)$$

- The stepsize (or learning rate) η_t is critical

The Challenge of Stepsize Selection

Gradient Descent (GD)

The foundational first-order optimization algorithm:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \nabla f(\boldsymbol{x}_t)$$

- The stepsize (or learning rate) η_t is critical
- Tuning η_t often requires knowledge of problem parameters (e.g., smoothness constant L, distance to optimum) that are unknown

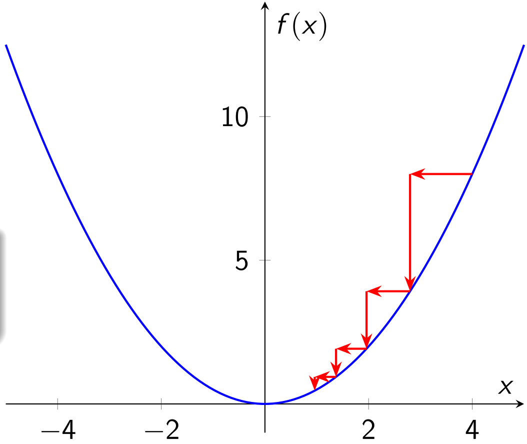
Gradient Descent on Smooth f: Step Size Too Small

Function: $f(x) = \frac{1}{2}x^2$

- Curvature (Smoothness) $L = 1$
- Step size $\eta = 0.3$

Observation

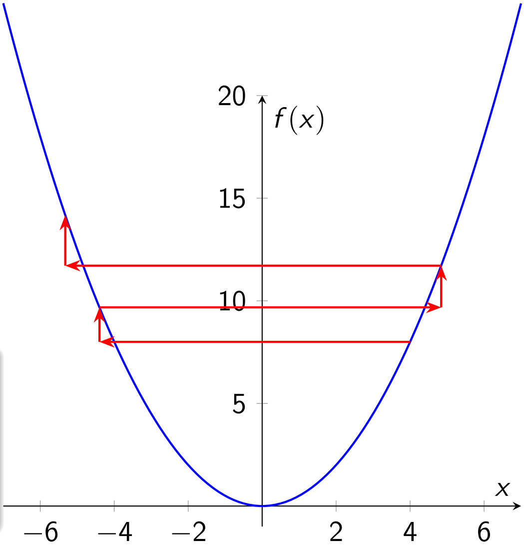
The algorithm converges, but takes many small steps



Gradient Descent on Smooth f: Step Size Too Large

Function: $f(x) = \frac{1}{2}x^2$

- Curvature (Smoothness) $L = 1$
- Step size $\eta = 2.1$
- This is larger than the divergence threshold $\eta = 2/L = 2$



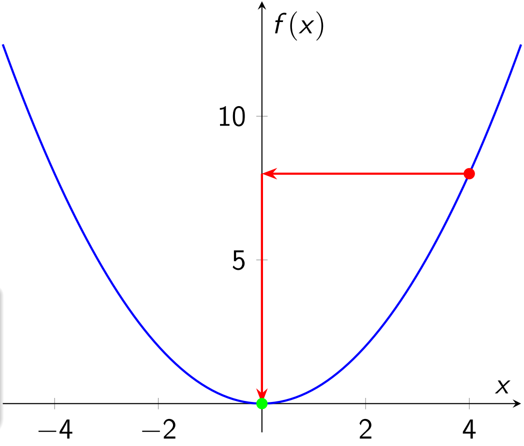
Observation

Each step overshoots the minimum by a larger amount, and the iterates move further away from the solution

Gradient Descent on Smooth f: Optimal Constant Step Size

Function: $f(x) = \frac{1}{2}x^2$

- Curvature (Smoothness) $L = 1$
- Optimal step size $\eta = 1/L = 1$



Observation

This stepsize maximizes the worst-case decrease of the function

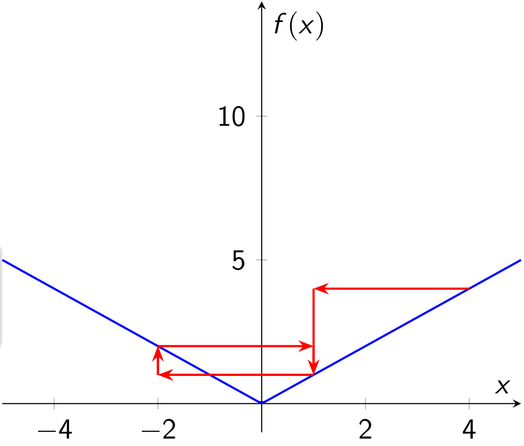
Gradient Descent on Non-Smooth f

Function: $f(x) = |x|$

- Step size $\eta = 3$

Observation

The learning rate is too large, it will oscillate



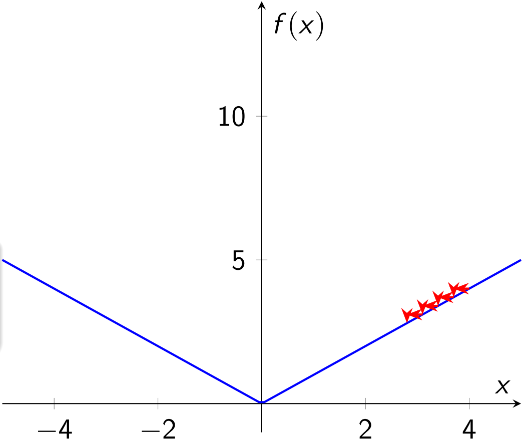
Gradient Descent on Non-Smooth f

Function: $f(x) = |x|$

- Step size $\eta = .3$

Observation

The learning rate is too small, it will converge very slowly



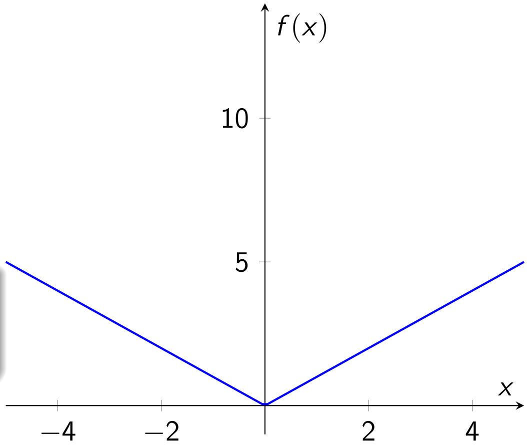
Gradient Descent on Non-Smooth f

Function: $f(x) = |x|$

- Optimal step size $\eta = \frac{\|x_1 - x^\star\|}{\sqrt{T}}$

Observation

The optimal learning rate depends on where you start



The "Magic" of the Polyak Stepsize

Proposed by Boris Polyak in 1969, the stepsize is defined as

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \frac{f(\boldsymbol{x}_t) - f^\star}{\|\boldsymbol{g}_t\|_2^2}\boldsymbol{g}_t,$$

where $\boldsymbol{g}_t \in \partial f(\boldsymbol{x}_t)$ is a subgradient and $f^\star = \min_{\boldsymbol{x}} f(\boldsymbol{x})$

The "Magic" of the Polyak Stepsize

Proposed by Boris Polyak in 1969, the stepsize is defined as

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \frac{f(\boldsymbol{x}_t) - f^\star}{\|\boldsymbol{g}_t\|_2^2}\boldsymbol{g}_t,$$

where $\boldsymbol{g}_t \in \partial f(\boldsymbol{x}_t)$ is a subgradient and $f^\star = \min_{\boldsymbol{x}} f(\boldsymbol{x})$

Remarkable Adaptivity

A single update rule achieves near-optimal rates for:

- Non-smooth convex functions: $\mathcal{O}(1/\sqrt{T})$
- Smooth convex functions: $\mathcal{O}(1/T)$
- Smooth & strongly convex functions: Linear convergence

...all without knowing smoothness or strong convexity constants!

The Central Research Question

Despite a resurgence of interest and many new variants [e.g., Rolinek&Martius, NeurIPS'18; Berrada et al., ICML'20; Loizou et al., AISTATS'21; Prazeres&Oberman, 2021], a fundamental question remains:

What makes the Polyak stepsize so adaptive, and when can it fail?

The Central Research Question

Despite a resurgence of interest and many new variants [e.g., Rolinek&Martius, NeurIPS'18; Berrada et al., ICML'20; Loizou et al., AISTATS'21; Prazeres&Oberman, 2021], a fundamental question remains:

What makes the Polyak stepsize so adaptive, and when can it fail?

Our Contributions

1. A new, unified perspective: Polyak's method is just **Gradient Descent on a surrogate function**

The Central Research Question

Despite a resurgence of interest and many new variants [e.g., Rolinek&Martius, NeurIPS'18; Berrada et al., ICML'20; Loizou et al., AISTATS'21; Prazeres&Oberman, 2021], a fundamental question remains:

What makes the Polyak stepsize so adaptive, and when can it fail?

Our Contributions

1. A new, unified perspective: Polyak's method is just **Gradient Descent on a surrogate function**
2. This surrogate is **always locally smooth and we know the smoothness constant**

The Central Research Question

Despite a resurgence of interest and many new variants [e.g., Rolinek&Martius, NeurIPS'18; Berrada et al., ICML'20; Loizou et al., AISTATS'21; Prazeres&Oberman, 2021], a fundamental question remains:

What makes the Polyak stepsize so adaptive, and when can it fail?

Our Contributions

1. A new, unified perspective: Polyak's method is just **Gradient Descent on a surrogate function**
2. This surrogate is **always locally smooth and we know the smoothness constant**
3. We use this framework to analyze a general family of Polyak-like algorithms

The Central Research Question

Despite a resurgence of interest and many new variants [e.g., Rolinek&Martius, NeurIPS'18; Berrada et al., ICML'20; Loizou et al., AISTATS'21; Prazeres&Oberman, 2021], a fundamental question remains:

What makes the Polyak stepsize so adaptive, and when can it fail?

Our Contributions

1. A new, unified perspective: Polyak's method is just **Gradient Descent on a surrogate function**
2. This surrogate is **always locally smooth and we know the smoothness constant**
3. We use this framework to analyze a general family of Polyak-like algorithms
4. We prove several **negative results**, showing that the non-convergence seen in some analyses is real, not an artifact

Polyak Stepsize as GD on a Surrogate

Let f be convex with minimizer $\boldsymbol{x}^\star$

Polyak Stepsize as GD on a Surrogate

Let f be convex with minimizer $\boldsymbol{x}^\star$
Instead of minimizing $f(\boldsymbol{x})$, consider minimizing a new surrogate function:

$$\phi(\boldsymbol{x}) = \frac{1}{2}\left(f(\boldsymbol{x}) - f(\boldsymbol{x}^\star)\right)^2$$

Polyak Stepsize as GD on a Surrogate

Let f be convex with minimizer $\boldsymbol{x}^\star$
Instead of minimizing $f(\boldsymbol{x})$, consider minimizing a new surrogate function:

$$\phi(\boldsymbol{x}) = \frac{1}{2}\left(f(\boldsymbol{x}) - f(\boldsymbol{x}^\star)\right)^2$$

Key Insight

The subgradient of $\phi(\boldsymbol{x})$ is $\nabla\phi(\boldsymbol{x}) = (f(\boldsymbol{x}) - f^\star)\boldsymbol{g}_{\boldsymbol{x}}$, where $\boldsymbol{g}_{\boldsymbol{x}} \in \partial f(\boldsymbol{x})$

A subgradient step on $\phi(\boldsymbol{x})$ with stepsize $\eta = \frac{1}{\|\boldsymbol{g}_{\boldsymbol{x}}\|_2^2}$ is:

$$\boldsymbol{x} - \eta\nabla\phi(\boldsymbol{x}) = \boldsymbol{x} - \frac{1}{\|\boldsymbol{g}_{\boldsymbol{x}}\|_2^2}(f(\boldsymbol{x}) - f^\star)\boldsymbol{g}_{\boldsymbol{x}}$$

This is exactly the Polyak update!

A New Notion of Local Curvature

So, the Polyak stepsize is GD on ϕ with stepsize $\eta'_t = 1/\|\boldsymbol{g}_t\|_2^2$. But why is this a good stepsize?

A New Notion of Local Curvature

So, the Polyak stepsize is GD on ϕ with stepsize $\eta'_t = 1/\|\boldsymbol{g}_t\|_2^2$. But why is this a good stepsize?

Definition (Local Star Upper Curvature - LSUC)

A function ϕ has $\lambda_{\boldsymbol{y}}$-LSUC around $\boldsymbol{y}$ if

$$\phi(\boldsymbol{x}^\star) - \langle \nabla\phi(\boldsymbol{y}), \boldsymbol{x}^\star - \boldsymbol{y}\rangle - \frac{1}{2\lambda_{\boldsymbol{y}}}\|\nabla\phi(\boldsymbol{y})\|_2^2 \geqslant \phi(\boldsymbol{y})$$

A New Notion of Local Curvature

So, the Polyak stepsize is GD on ϕ with stepsize $\eta'_t = 1/\|\boldsymbol{g}_t\|_2^2$. But why is this a good stepsize?

Definition (Local Star Upper Curvature - LSUC)

A function ϕ has $\lambda_{\boldsymbol{y}}$-LSUC around $\boldsymbol{y}$ if

$$\phi(\boldsymbol{x}^\star) - \langle \nabla \phi(\boldsymbol{y}), \boldsymbol{x}^\star - \boldsymbol{y} \rangle - \frac{1}{2\lambda_{\boldsymbol{y}}} \|\nabla \phi(\boldsymbol{y})\|_2^2 \geqslant \phi(\boldsymbol{y})$$

- L-smooth functions are L-LSUC everywhere, in fact convex smooth functions satisfy

$$f(\boldsymbol{x}) - \langle \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle - \frac{1}{2L} \|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|^2 \geqslant f(\boldsymbol{y}), \ \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$$

- This is a local smoothness-like condition

The Local Curvature is the Source of Adaptivity

Theorem 1: Curvature of the Polyak Surrogate

For any f convex, the surrogate $\phi(\boldsymbol{x}) = \frac{1}{2}(f(\boldsymbol{x}) - f^\star)^2$ is $\|\boldsymbol{g}_{\boldsymbol{x}}\|_2^2$-**LSUC** around any $\boldsymbol{x}$

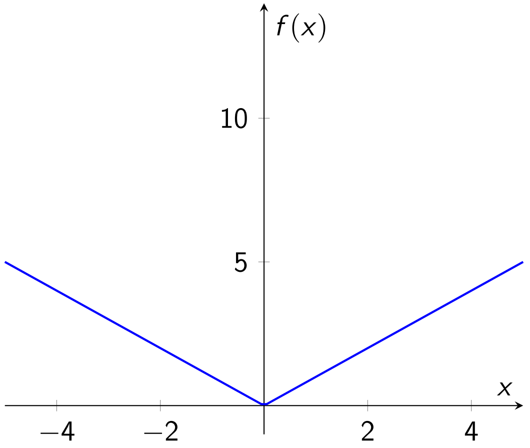
The Local Curvature is the Source of Adaptivity

Theorem 1: Curvature of the Polyak Surrogate

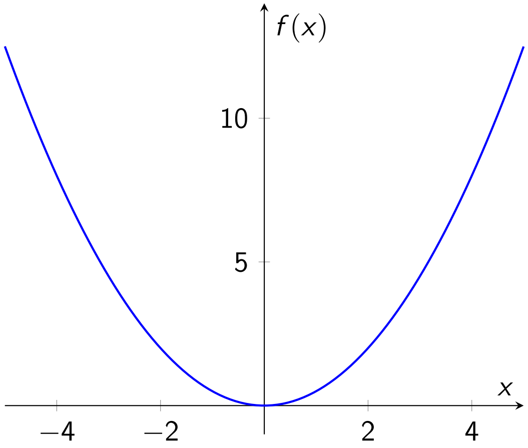
For any f convex, the surrogate $\phi(\boldsymbol{x}) = \frac{1}{2}(f(\boldsymbol{x}) - f^\star)^2$ is $\|\boldsymbol{g}_{\boldsymbol{x}}\|_2^2$-**LSUC** around any $\boldsymbol{x}$

- **This is the magic!** The surrogate ϕ is *always* "locally smooth"
- The adaptive stepsize $1/\|\boldsymbol{g}_t\|_2^2$ is simply the inverse of this local curvature constant!

Non-smooth: Choice of Stepsize is Difficult



Smooth: Choice of Stepsize is Easy, *Knowing Smoothness*



Recovering Convergence Rates

Using this perspective, we can easily derive convergence guarantees

Lemma (One-step Progress)

Using stepsize $\eta'_t = 1/\lambda_{\boldsymbol{x}_t} = 1/\|\boldsymbol{g}_t\|_2^2$ on ϕ gives:

$$\eta'_t \left(\phi(\boldsymbol{x}_t) - \phi(\boldsymbol{x}^\star)\right) \leqslant \frac{1}{2}\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|_2^2 - \frac{1}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|_2^2$$

Recovering Convergence Rates

Using this perspective, we can easily derive convergence guarantees

Lemma (One-step Progress)

Using stepsize $\eta'_t = 1/\lambda_{\boldsymbol{x}_t} = 1/\|\boldsymbol{g}_t\|_2^2$ on ϕ gives:

$$\eta'_t \left(\phi(\boldsymbol{x}_t) - \phi(\boldsymbol{x}^\star)\right) \leqslant \frac{1}{2}\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|_2^2 - \frac{1}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|_2^2$$

Summing over T steps:

$$\sum_{t=1}^{T} \eta'_t \phi(\boldsymbol{x}_t) \leqslant \frac{1}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}^\star\|_2^2$$

Recovering Convergence Rates

Using this perspective, we can easily derive convergence guarantees

Lemma (One-step Progress)

Using stepsize $\eta'_t = 1/\lambda_{\boldsymbol{x}_t} = 1/\|\boldsymbol{g}_t\|_2^2$ on ϕ gives:

$$\eta'_t \left(\phi(\boldsymbol{x}_t) - \phi(\boldsymbol{x}^\star)\right) \leqslant \frac{1}{2}\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|_2^2 - \frac{1}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|_2^2$$

Summing over T steps:

$$\sum_{t=1}^{T} \eta'_t \phi(\boldsymbol{x}_t) \leqslant \frac{1}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}^\star\|_2^2$$

- **If f is G-Lipschitz:** $\sum \eta'_t \geqslant T/G^2 \Rightarrow \phi(\bar{\boldsymbol{x}}_T) = \mathcal{O}(1/T)$
- **If f is L-self-bounded:** We recover $\phi(\bar{\boldsymbol{x}}_T) = \mathcal{O}(1/T^2)$
- **If f is sharp:** We recover linear convergence

From the Surrogate to the Original Function

- We can easily convert a rate on the surrogate to a rate on the original function by inverting the surrogate
- For example for smooth losses we have

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^\star) = \sqrt{2\phi(\boldsymbol{x})} = \sqrt{\mathcal{O}(1/T^2)} = \mathcal{O}(1/T)$$

Not a Completely New Idea

- Gower et al. [ArXiv'21] showed that the stochastic Polyak stepsize can be casted as online convex optimization problem on surrogate losses
- The adversarial nature of online convex optimization means that it is not possible to say that we are minimizing a specific function
- For the same reason, they need slightly stronger assumptions

A Family of Surrogates

We can generalize this idea beyond knowing $f^\star$

General Surrogate

Consider $\psi(\boldsymbol{x}) = \frac{1}{2}h^2(\boldsymbol{x})$, where $h : \mathbb{R}^d \to \mathbb{R}_{\geqslant 0}$ is convex

A Family of Surrogates

We can generalize this idea beyond knowing $f^\star$

General Surrogate

Consider $\psi(\boldsymbol{x}) = \frac{1}{2}h^2(\boldsymbol{x})$, where $h : \mathbb{R}^d \to \mathbb{R}_{\geqslant 0}$ is convex

Examples of $h(\boldsymbol{x})$:

- Original Polyak: $h(\boldsymbol{x}) = f(\boldsymbol{x}) - f^\star$
- Unknown $f^\star$: $h(\boldsymbol{x}) = (f(\boldsymbol{x}) - c)_+$ for some estimate c
- Stochastic Variants, for example, SPS$_+$ [Garrigos et al., 2023]: $h(\boldsymbol{x}, \boldsymbol{\xi}) = |f(\boldsymbol{x}, \boldsymbol{\xi}) - f(\boldsymbol{x}^\star, \boldsymbol{\xi})|_+$

A Family of Surrogates

We can generalize this idea beyond knowing $f^\star$

General Surrogate

Consider $\psi(\boldsymbol{x}) = \frac{1}{2}h^2(\boldsymbol{x})$, where $h : \mathbb{R}^d \to \mathbb{R}_{\geqslant 0}$ is convex

Examples of $h(\boldsymbol{x})$:

- Original Polyak: $h(\boldsymbol{x}) = f(\boldsymbol{x}) - f^\star$
- Unknown $f^\star$: $h(\boldsymbol{x}) = (f(\boldsymbol{x}) - c)_+$ for some estimate c
- Stochastic Variants, for example, SPS$_+$ [Garrigos et al., 2023]: $h(\boldsymbol{x}, \boldsymbol{\xi}) = |f(\boldsymbol{x}, \boldsymbol{\xi}) - f(\boldsymbol{x}^\star, \boldsymbol{\xi})|_+$

Problem

If the minimum of h is not zero (i.e., $h(\boldsymbol{x}^\star) > 0$), the surrogate only has *approximate* local curvature. This leads to convergence to a **neighborhood**, not the true optimum.

The Stochastic Setting

Consider minimizing $F(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}\sim D}[f(\boldsymbol{x}, \boldsymbol{\xi})]$.
Stochastic Polyak variants use a surrogate based on a single sample $\boldsymbol{\xi}_t$: $\frac{1}{2}h(\boldsymbol{x}, \boldsymbol{\xi}_t)$, and update with

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t h(\boldsymbol{x}_t, \boldsymbol{\xi}_t)\boldsymbol{g}_t, \text{ where } \boldsymbol{g}_t \in \partial h(\boldsymbol{x}, \boldsymbol{\xi}_t)$$

The Stochastic Setting

Consider minimizing $F(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}\sim D}[f(\boldsymbol{x},\boldsymbol{\xi})]$.
Stochastic Polyak variants use a surrogate based on a single sample $\boldsymbol{\xi}_t$: $\frac{1}{2}h(\boldsymbol{x},\boldsymbol{\xi}_t)$, and update with

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t h(\boldsymbol{x}_t,\boldsymbol{\xi}_t)\boldsymbol{g}_t, \text{ where } \boldsymbol{g}_t \in \partial h(\boldsymbol{x},\boldsymbol{\xi}_t)$$

A Fundamental Mismatch

The algorithm is effectively minimizing the expectation of the *surrogate*:

$$\mathbb{E}_{\boldsymbol{\xi}\sim D}\left[\frac{1}{2}h^2(\boldsymbol{x},\boldsymbol{\xi})\right]$$

This is generally **not** the same as minimizing the original objective $F(\boldsymbol{x})$!

$$\operatorname*{argmin}_{\boldsymbol{x}} \ \mathbb{E}[h^2(\boldsymbol{x},\boldsymbol{\xi})] \neq \operatorname*{argmin}_{\boldsymbol{x}} \ \mathbb{E}[f(\boldsymbol{x},\boldsymbol{\xi})]$$

The Stochastic Setting

Consider minimizing $F(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}\sim D}[f(\boldsymbol{x}, \boldsymbol{\xi})]$.
Stochastic Polyak variants use a surrogate based on a single sample $\boldsymbol{\xi}_t$: $\frac{1}{2}h(\boldsymbol{x}, \boldsymbol{\xi}_t)$, and update with

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t h(\boldsymbol{x}_t, \boldsymbol{\xi}_t)\boldsymbol{g}_t, \text{ where } \boldsymbol{g}_t \in \partial h(\boldsymbol{x}, \boldsymbol{\xi}_t)$$

A Fundamental Mismatch

The algorithm is effectively minimizing the expectation of the *surrogate*:

$$\mathbb{E}_{\boldsymbol{\xi}\sim D}\left[\frac{1}{2}h^2(\boldsymbol{x}, \boldsymbol{\xi})\right]$$

This is generally **not** the same as minimizing the original objective $F(\boldsymbol{x})$!

$$\operatorname*{argmin}_{\boldsymbol{x}} \ \mathbb{E}[h^2(\boldsymbol{x}, \boldsymbol{\xi})] \neq \operatorname*{argmin}_{\boldsymbol{x}} \ \mathbb{E}[f(\boldsymbol{x}, \boldsymbol{\xi})]$$

Warning: Minimizing a different loss function is problematic for a ML point of view

Unified Analysis of Stochastic Variants

We propose a generalized algorithm with a clipped stepsize, covering methods like $\mathtt{SPS}_{\max}$, $\mathtt{SPS}_{+}$, etc.

for $t = 1, \ldots, T$ **do**
    Sample $\boldsymbol{\xi}_t$
    Get subgradient $\boldsymbol{g}_t \in \partial h(\boldsymbol{x}_t, \boldsymbol{\xi}_t)$
    $\eta_t = \min\left(\frac{1}{\|\boldsymbol{g}_t\|_2^2}, \frac{\gamma}{h(\boldsymbol{x}_t, \boldsymbol{\xi}_t)}\right)$
    $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t h(\boldsymbol{x}_t, \boldsymbol{\xi}_t) \boldsymbol{g}_t$
end for

General Convergence Guarantees

Theorem

- Let $H(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi} \sim D}[h(\boldsymbol{x}, \boldsymbol{\xi})]$. If $h(\cdot, \boldsymbol{\xi}_t)$ is L-self bounded, we have

$$\min\left(\frac{1}{2L}, \gamma\right) \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[H(\boldsymbol{x}_t)] \leqslant \frac{\|\boldsymbol{x}_1 - \boldsymbol{x}^\star\|^2}{T} + 2\gamma H(\boldsymbol{x}^\star)$$

General Convergence Guarantees

Theorem

- Let $H(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi} \sim D}[h(\boldsymbol{x}, \boldsymbol{\xi})]$. If $h(\cdot, \boldsymbol{\xi}_t)$ is L-self bounded, we have

$$\min\left(\frac{1}{2L}, \gamma\right) \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[H(\boldsymbol{x}_t)] \leqslant \frac{\|\boldsymbol{x}_1 - \boldsymbol{x}^\star\|^2}{T} + 2\gamma H(\boldsymbol{x}^\star)$$

- If $h(\cdot, \boldsymbol{\xi}_t)$ is G-Lipschitz, then we have

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[H(\boldsymbol{x}_t)] \leqslant \frac{\|\boldsymbol{x}_1 - \boldsymbol{x}^\star\|^2}{\gamma T} + 2H(\boldsymbol{x}^\star) + \frac{G\|\boldsymbol{x}_1 - \boldsymbol{x}^\star\|}{\sqrt{T}} + G\sqrt{2\gamma H(\boldsymbol{x}^\star)}$$

General Convergence Guarantees

Theorem

- Let $H(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}\sim D}[h(\boldsymbol{x}, \boldsymbol{\xi})]$. If $h(\cdot, \boldsymbol{\xi}_t)$ is L-self bounded, we have

$$\min\left(\frac{1}{2L}, \gamma\right) \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[H(\boldsymbol{x}_t)] \leqslant \frac{\|\boldsymbol{x}_1 - \boldsymbol{x}^\star\|^2}{T} + 2\gamma H(\boldsymbol{x}^\star)$$

- If $h(\cdot, \boldsymbol{\xi}_t)$ is G-Lipschitz, then we have

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[H(\boldsymbol{x}_t)] \leqslant \frac{\|\boldsymbol{x}_1 - \boldsymbol{x}^\star\|^2}{\gamma T} + 2H(\boldsymbol{x}^\star) + \frac{G\|\boldsymbol{x}_1 - \boldsymbol{x}^\star\|}{\sqrt{T}} + G\sqrt{2\gamma H(\boldsymbol{x}^\star)}$$

- If $h(\cdot, \boldsymbol{\xi})$ is L-self-bounded and $H(\boldsymbol{x})$ has μ-quadratic growth, then

$$\mathbb{E}\left[\|\boldsymbol{x}_{T+1} - \boldsymbol{x}^\star\|^2\right] \leqslant \mathbb{E}\left[\|\boldsymbol{x}_1 - \boldsymbol{x}^\star\|^2\right] a^{T+1} + b\frac{1 - a^{T+1}}{1 - a} H(\boldsymbol{x}^\star),$$

where $a = \frac{\mu}{2} \min\left(\frac{1}{2L}, \gamma\right)$ and $b = 2\gamma - \min\left(\frac{1}{2L}, \gamma\right)$

When Things Go Wrong: $h(\boldsymbol{x}^\star) > 0$

What happens when the minimum of our surrogate-generator h is strictly positive?

- **Deterministic case:** We underestimate $f^\star$ (e.g., $h(\boldsymbol{x}) = f(\boldsymbol{x}) - c$ with $c < f^\star$)
- The correct stepsize for the underlying surrogate $\psi = \frac{1}{2}(h - h^\star)^2$ is $\frac{1}{\lambda_t}$, but the algorithm uses a stepsize for $\frac{1}{2}h^2$:

$$\eta'_t = \left(\frac{h(\boldsymbol{x}_t)}{h(\boldsymbol{x}_t) - h^\star}\right)\frac{1}{\lambda_t}$$

When Things Go Wrong: $h(\boldsymbol{x}^\star) > 0$

What happens when the minimum of our surrogate-generator h is strictly positive?

- **Deterministic case:** We underestimate $f^\star$ (e.g., $h(\boldsymbol{x}) = f(\boldsymbol{x}) - c$ with $c < f^\star$)
- The correct stepsize for the underlying surrogate $\psi = \frac{1}{2}(h - h^\star)^2$ is $\frac{1}{\lambda_t}$, but the algorithm uses a stepsize for $\frac{1}{2}h^2$:

$$\eta_t' = \left(\frac{h(\boldsymbol{x}_t)}{h(\boldsymbol{x}_t) - h^\star}\right)\frac{1}{\lambda_t}$$

Problem

As $\boldsymbol{x}_t \to \boldsymbol{x}^\star$, we have $h(\boldsymbol{x}_t) \to h^\star > 0$. The term $\frac{h(\boldsymbol{x}_t)}{h(\boldsymbol{x}_t)-h^\star}$ **blows up to** $+\infty$**!** The stepsize becomes enormous near the minimum, causing instability. Moreover, clipping will not fix this issue.

Unstable Fixed Points

This intuition is formalized in following Proposition

Proposition (Unstable Fixed Point)

For a wide class of functions h (e.g., self-bounded with quadratic growth), if $h(\boldsymbol{x}^\star) > 0$, then the minimizer $\boldsymbol{x}^\star$ is an **unstable fixed point**.
There exists a neighborhood around $\boldsymbol{x}^\star$ where if you enter, the next step will take you **further away** from $\boldsymbol{x}^\star$.

Unstable Fixed Points

This intuition is formalized in following Proposition

Proposition (Unstable Fixed Point)

For a wide class of functions h (e.g., self-bounded with quadratic growth), if $h(\boldsymbol{x}^\star) > 0$, then the minimizer $\boldsymbol{x}^\star$ is an **unstable fixed point**.
There exists a neighborhood around $\boldsymbol{x}^\star$ where if you enter, the next step will take you **further away** from $\boldsymbol{x}^\star$.

- This confirms that the neighborhood of non-convergence is not just an artifact of the analysis

Cycling of the Iterates

This instability can lead to more than just a failure to converge; it can lead to cycles

Proposition (Cycling)

Consider the simple 1D function $h(x) = x^2 + 1$. Here $h^\star = 1 > 0$. There exists an initial point $\boldsymbol{x}_1$ such that the update rule

$$x_{t+1} = x_t - \frac{h(x_t)}{\|\nabla h(x_t)\|_2^2} \nabla h(x_t)$$

cycles on points different than $x^\star = 0$. Moreover, the suboptimality on the average of the iterates also fails to converge.

Proof by Gemini 2.5

The update is

$$x_{t+1} = x_t - \frac{h(x_t)}{\|\nabla h(x_t)\|_2^2} \nabla h(x_t) = x_t - \frac{x_t^2 + 1}{2x_t} = \frac{x_t^2 - 1}{2x_t}$$

Proof by Gemini 2.5

The update is

$$x_{t+1} = x_t - \frac{h(x_t)}{\|\nabla h(x_t)\|_2^2} \nabla h(x_t) = x_t - \frac{x_t^2 + 1}{2x_t} = \frac{x_t^2 - 1}{2x_t}$$

If we start at $x_1 = \cot(\theta)$, from the identity of $\cot(2x) = \frac{\cot^2 \theta - 1}{2 \cot \theta}$

Proof by Gemini 2.5

The update is

$$x_{t+1} = x_t - \frac{h(x_t)}{\|\nabla h(x_t)\|_2^2}\nabla h(x_t) = x_t - \frac{x_t^2+1}{2x_t} = \frac{x_t^2-1}{2x_t}$$

If we start at $x_1 = \cot(\theta)$, from the identity of $\cot(2x) = \frac{\cot^2\theta - 1}{2\cot\theta}$
Hence, set $x_1 = \cot \pi/7$, to have

$$x_1 = \cot(\pi/7) \to x_2 = \cot(2\pi/7) \to x_3 = \cot(4\pi/7) \to x_4 = x_1$$

The Set of Good Initial Point has Measure Zero

- In the previous Proposition we found a very specific initial point such that the algorithm cycles
- Was it just a very unlucky initial point?

The Set of Good Initial Point has Measure Zero

- In the previous Proposition we found a very specific initial point such that the algorithm cycles
- Was it just a very unlucky initial point?

Proposition

For $h(x) = x^2/2 + a$, the set of initial points where the update $x_{t+1} = x_t - \frac{h(x_t)}{\|\nabla h(x_t)\|_2^2} \nabla h(x_t)$ converges to the minimum has **measure zero**

Failure in the Stochastic Case

Proposition

There exist f_1 and f_2 quadratic 1-d functions and a starting point x_1 such that SPS on $F(x) = 0.5(f_1(x) + f_2(x))$ satisfies

$$\mathbb{E}[F(x_t)] - \min_x F(x) \geqslant 2/3, \ \forall t$$

Summary and Takeaways

The Good: A Unifying Perspective

- The Polyak stepsize is equivalent to GD on a surrogate $\phi(\boldsymbol{x}) = \frac{1}{2}(f - f^\star)^2$
- The adaptivity comes from the fact that ϕ is always locally "smooth" with a known curvature constant $\|\boldsymbol{g}\|_2^2$
- This framework simplifies and unifies the analysis of many Polyak-like methods

Summary and Takeaways

The Good: A Unifying Perspective

- The Polyak stepsize is equivalent to GD on a surrogate $\phi(\boldsymbol{x}) = \frac{1}{2}(f - f^\star)^2$
- The adaptivity comes from the fact that ϕ is always locally "smooth" with a known curvature constant $\|\boldsymbol{g}\|_2^2$
- This framework simplifies and unifies the analysis of many Polyak-like methods

The Bad: Fundamental Instability

- When the surrogate's minimum value is positive (e.g., $f^\star$ underestimated or no interpolation), the dynamics change drastically
- The algorithm becomes unstable near the optimum, leading to cycles and non-convergence
- This neighborhood of convergence is not an analysis artifact but a fundamental property of the method

"New Perspectives on the Polyak Stepsize: Surrogate Functions and Negative Results" Francesco Orabona, Ryan D'Orazio, NeurIPS'25

P.S. We have multiple PhD/Post-Doc/Research Scientist positions, and 3000 GH200 GPUs