TRƯỜNG ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

VIỆN TOÁN ỨNG
DỤNG VÀ TIN HỌC
SCHOOL OF APPLIED
MATHEMATICS AND INFORMATICS

Self-adaptive algorithms for quasiconvex programming and applications to machine learning

Nhóm 3

Tháng 12, 2025

ONE LOVE. ONE FUTURE

1.	Nguyễn Phong	MSSV: 202400066
2.	Nguyễn Đăng Long	MSSV: 202400057
3.	Phạm Gia Linh	MSSV: 202416262
4.	Nguyễn Tuấn Long	MSSV: 202416269
5.	Đoàn Đức Mạnh	MSSV: 202400058
6.	Trần Đình Nam	MSSV: 202400063
7.	Hoàng Thị Thu Phương	MSSV: 202400068
8.	Đặng Bảo Quân	MSSV: 202416319
9.	Phạm Thị Bích Phương	MSSV: 202400069
10.	Dương Thanh Minh	MSSV: 20230047
11.	Nguyễn Hải Yến Nhi	MSSV: 202400064
12.	Lương Đức Mạnh	MSSV: 20235152

- ◇ Giả sử C là một tập hợp **khác rỗng**, **đóng** và **lồi** trong $\mathbb{R}^m$.
- ◇ Hàm $f: \mathbb{R}^m \to \mathbb{R}$ là một hàm khả vi trên một tập mở chứa C.
- ◇ Ánh xạ ∇f là liên tục L-Lipschitz, tức là tồn tại $L > 0$ sao cho:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y} \in C$$

- ◇ Giả sử C là một tập hợp **khác rỗng**, **đóng** và **lồi** trong $\mathbb{R}^m$.
- ◇ Hàm $f: \mathbb{R}^m \to \mathbb{R}$ là một hàm khả vi trên một tập mở chứa C.
- ◇ Ánh xạ ∇f là liên tục L-Lipschitz, tức là tồn tại $L > 0$ sao cho:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y} \in C$$

Bài toán tối ưu:

$$\min_{\mathbf{x} \in C} f(\mathbf{x}), \quad OP(f, C)$$

◇ Giả sử tập nghiệm của $OP(f, C)$ là **không rỗng**.

Với $\mathbf{x} \in \mathbb{R}^m$, ký hiệu $P_C(\mathbf{x})$ là **phép chiếu** của $\mathbf{x}$ lên C:

$$P_C(\mathbf{x}) := \arg\min\{\|\mathbf{z} - \mathbf{x}\| : \mathbf{z} \in C\}.$$

◇ Giả sử tập nghiệm của $OP(f, C)$ là **không rỗng**.

Với $\mathbf{x} \in \mathbb{R}^m$, ký hiệu $P_C(\mathbf{x})$ là **phép chiếu** của $\mathbf{x}$ lên C:

$$P_C(\mathbf{x}) := \arg\min\{\|\mathbf{z} - \mathbf{x}\| : \mathbf{z} \in C\}.$$

Mệnh đề 1 (Bauschke và Combettes 2011)

Các khẳng định sau là đúng:

1. $\|P_C(\mathbf{x}) - P_C(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\|$ với mọi $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$.
2. $(\mathbf{y} - P_C(\mathbf{x}))^T(\mathbf{x} - P_C(\mathbf{x})) \leq 0$ với mọi $\mathbf{x} \in \mathbb{R}^m, \mathbf{y} \in C$.

Định nghĩa 1 (Mangasarian 1965)

Hàm $f : \mathbb{R}^m \to \mathbb{R}$ được gọi là:

◇ ***Lồi** (convex) trên C nếu $\forall \mathbf{x}, \mathbf{y} \in C, \lambda \in [0, 1]$:*

$$f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y}).$$

Định nghĩa 1 (Mangasarian 1965)

Hàm $f : \mathbb{R}^m \to \mathbb{R}$ được gọi là:

◇ ***Lồi** (convex) trên C nếu $\forall \mathbf{x}, \mathbf{y} \in C, \lambda \in [0, 1]$:*

$$f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y}).$$

◇ ***Giả lồi** (pseudoconvex) trên C nếu $\forall \mathbf{x}, \mathbf{y} \in C$:*

$$\nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) \geq 0 \implies f(\mathbf{y}) \geq f(\mathbf{x}).$$

Định nghĩa 1 (Mangasarian 1965)

Hàm $f: \mathbb{R}^m \to \mathbb{R}$ được gọi là:

- ◇ ***Lồi** (convex) trên C nếu $\forall \mathbf{x}, \mathbf{y} \in C, \lambda \in [0,1]$:*

$$f(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}).$$

- ◇ ***Giả lồi** (pseudoconvex) trên C nếu $\forall \mathbf{x}, \mathbf{y} \in C$:*

$$\nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) \geq 0 \implies f(\mathbf{y}) \geq f(\mathbf{x}).$$

- ◇ ***Tựa lồi** (quasiconvex) trên C nếu $\forall \mathbf{x}, \mathbf{y} \in C, \lambda \in [0,1]$:*

$$f(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) \leq \max\{f(\mathbf{x}); f(\mathbf{y})\}.$$

Mệnh đề 2: Đặc trưng tựa lồi

Hàm khả vi f là tựa lồi trên C khi và chỉ khi:

$$f(\mathbf{y}) \leq f(\mathbf{x}) \Rightarrow \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \leq 0, \quad \forall \mathbf{x}, \mathbf{y} \in C.$$

Mệnh đề 2: Đặc trưng tựa lồi

Hàm khả vi f là tựa lồi trên C khi và chỉ khi:

$$f(\mathbf{y}) \leq f(\mathbf{x}) \Rightarrow \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) \leq 0, \quad \forall \mathbf{x}, \mathbf{y} \in C.$$

Mối quan hệ bao hàm

Hàm lồi $\Longrightarrow$ Hàm giả lồi $\Longrightarrow$ Hàm tựa lồi

Mệnh đề 2: Đặc trưng tựa lồi

Hàm khả vi f là tựa lồi trên C khi và chỉ khi:

$$f(\mathbf{y}) \leq f(\mathbf{x}) \Rightarrow \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) \leq 0, \quad \forall \mathbf{x}, \mathbf{y} \in C.$$

Mối quan hệ bao hàm

Hàm lồi $\Longrightarrow$ Hàm giả lồi $\Longrightarrow$ Hàm tựa lồi

Mệnh đề 3: Tính chất Lipschitz

Giả sử ∇f là liên tục L-Lipschitz trên C. Khi đó:

$$|f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x})| \leq \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|^2.$$

Lemma 1 (Xu 2002)

Cho $\{a_k\}$ và $\{b_k\}$ là các dãy số dương sao cho:

$$a_{k+1} \leq a_k + b_k \quad \forall k \geq 0 \quad \text{và} \quad \sum_{k=0}^{\infty} b_k < \infty.$$

Khi đó, tồn tại giới hạn $\lim_{k\to\infty} a_k = c \in \mathbb{R}$.

Algorithm 1 (Gradient Descent Adaptive Algorithm-GDA)

Input: $x^0 \in C$; $\lambda_0 > 0$; $\sigma, \kappa \in (0,1)$; $k = 0$
Output: Stationary point x^*
while true **do**
    $x^{k+1} \leftarrow P_C\big(x^k - \lambda_k \nabla f(x^k)\big)$
    if $f(x^{k+1}) \leq f(x^k) - \sigma\langle \nabla f(x^k), x^k - x^{k+1}\rangle$ **then**
        $\lambda_{k+1} \leftarrow \lambda_k$
    else
        $\lambda_{k+1} \leftarrow \kappa\lambda_k$
    end if
    if $x^{k+1} = x^k$ **then**
        break
    end if
    $k \leftarrow k + 1$
end while

Nhận xét 1

Nếu thuật toán GDA dừng tại bước k, thì $\mathbf{x}^k$ là một điểm dừng của bài toán $OP(f, C)$.

Nhận xét 1

Nếu thuật toán GDA dừng tại bước k, thì $\mathbf{x}^k$ là một điểm dừng của bài toán $OP(f, C)$.

Chứng minh:

Do $\mathbf{x}^{k+1} = P_C(\mathbf{x}^k - \lambda_k \nabla f(\mathbf{x}^k))$, áp dụng Mệnh đề 1-(ii), ta có:

$$\langle \mathbf{z} - \mathbf{x}^{k+1}, \mathbf{x}^k - \lambda_k \nabla f(\mathbf{x}^k) - \mathbf{x}^{k+1} \rangle \leq 0, \quad \forall \mathbf{z} \in C \tag{1}$$

Nhận xét 1

Nếu thuật toán GDA dừng tại bước k, thì $\mathbf{x}^k$ là một điểm dừng của bài toán $OP(f, C)$.

Chứng minh:

Do $\mathbf{x}^{k+1} = P_C(\mathbf{x}^k - \lambda_k \nabla f(\mathbf{x}^k))$, áp dụng Mệnh đề 1-(ii), ta có:

$$\langle \mathbf{z} - \mathbf{x}^{k+1}, \mathbf{x}^k - \lambda_k \nabla f(\mathbf{x}^k) - \mathbf{x}^{k+1} \rangle \leq 0, \quad \forall \mathbf{z} \in C \tag{1}$$

Nếu $\mathbf{x}^{k+1} = \mathbf{x}^k$, ta thu được:

$$\langle \nabla f(\mathbf{x}^k), \mathbf{z} - \mathbf{x}^k \rangle \geq 0, \quad \forall \mathbf{z} \in C \tag{2}$$

Nhận xét 1

Nếu thuật toán GDA dừng tại bước k, thì $\mathbf{x}^k$ là một điểm dừng của bài toán $OP(f, C)$.

Chứng minh:

Do $\mathbf{x}^{k+1} = P_C(\mathbf{x}^k - \lambda_k \nabla f(\mathbf{x}^k))$, áp dụng Mệnh đề 1-(ii), ta có:

$$\langle \mathbf{z} - \mathbf{x}^{k+1}, \mathbf{x}^k - \lambda_k \nabla f(\mathbf{x}^k) - \mathbf{x}^{k+1} \rangle \leq 0, \quad \forall \mathbf{z} \in C \tag{1}$$

Nếu $\mathbf{x}^{k+1} = \mathbf{x}^k$, ta thu được:

$$\langle \nabla f(\mathbf{x}^k), \mathbf{z} - \mathbf{x}^k \rangle \geq 0, \quad \forall \mathbf{z} \in C \tag{2}$$

Khi đó $\mathbf{x}^k$ là một điểm dừng của bài toán.

Định lý 1

Giả sử dãy $\{\mathbf{x}^k\}$ được sinh bởi Thuật toán GDA. Khi đó, dãy $\{f(\mathbf{x}^k)\}$ hội tụ và mỗi điểm giới hạn (nếu có) của dãy $\{\mathbf{x}^k\}$ là một điểm dừng của bài toán. Hơn nữa:

- ◇ *Nếu f là hàm tựa lồi trên C, thì dãy $\{\mathbf{x}^k\}$ hội tụ về một điểm dừng của bài toán.*
- ◇ *Nếu f là hàm giả lồi trên C, thì dãy $\{\mathbf{x}^k\}$ hội tụ về một nghiệm của bài toán.*

1. Bất đẳng thức cơ bản:

Kết hợp tính chất L-Lipschitz và tính chất của phép chiếu P_C:

$$\begin{cases} f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^k \rangle + \frac{L}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 \\ \langle \nabla f(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^k \rangle \leq -\frac{1}{\lambda_k} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 \end{cases}$$

1. Bất đẳng thức cơ bản:

Kết hợp tính chất L-Lipschitz và tính chất của phép chiếu P_C:

$$\begin{cases} f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^k \rangle + \frac{L}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 \\ \langle \nabla f(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^k \rangle \leq -\frac{1}{\lambda_k} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 \end{cases}$$

$\Longrightarrow$ Ta thu được đánh giá quan trọng:

$$\boxed{f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) - \sigma \langle \nabla f(\mathbf{x}^k), \mathbf{x}^k - \mathbf{x}^{k+1} \rangle - \left(\frac{1-\sigma}{\lambda_k} - \frac{L}{2} \right) \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2} \quad (3)$$

2. Sự hội tụ:
Giả sử $\lambda_k \to 0$, khi đó tồn tại $k_0 \in \mathbb{N}$ thỏa mãn

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) - \sigma\langle\nabla f(\mathbf{x}^k), \mathbf{x}^k - \mathbf{x}^{k+1}\rangle \quad \forall k \geq k_0. \tag{4}$$

Theo cách xây dựng λ_k thì $\lambda_k = \lambda_{k_0} \quad \forall k \geq k_0$ (mâu thuẫn), do đó:

$$\exists k_1 : \lambda_k = \lambda_{k_1} = \bar{\lambda} > 0, \quad \forall k \geq k_1.$$

và thỏa mãn (4) với mọi $k \geq k_1$.
Từ $\langle\nabla f(\mathbf{x}^k), \mathbf{x}^k - \mathbf{x}^{k+1}\rangle \geq 0$ ta suy ra $\{f(\mathbf{x}^k)\}$ hội tụ và:

$$\sum_{k=0}^{\infty}\langle\nabla f(\mathbf{x}^k), \mathbf{x}^k - \mathbf{x}^{k+1}\rangle < \infty; \quad \sum_{k=0}^{\infty}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 < \infty. \tag{5}$$

3. Điểm dừng:
Từ tính chất phép chiếu với mọi $\mathbf{z} \in C$:

$$\|\mathbf{x}^{k+1} - \mathbf{z}\|^2 \leq \|\mathbf{x}^k - \mathbf{z}\|^2 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + 2\lambda_k \langle \nabla f(\mathbf{x}^k), \mathbf{z} - \mathbf{x}^{k+1} \rangle \tag{6}$$

Xét dãy con $\mathbf{x}^{k_i} \to \bar{\mathbf{x}}$. Thay k_i vào (6) và cho $i \to \infty$, kết hợp tính liên tục của ∇f:

$$\langle \nabla f(\bar{\mathbf{x}}), \mathbf{z} - \bar{\mathbf{x}} \rangle \geq 0, \quad \forall \mathbf{z} \in C. \tag{7}$$

$\Longrightarrow$ $\bar{\mathbf{x}}$ là một **điểm dừng** của bài toán.

4. Trường hợp f Tựa lồi / Giả lồi:
Đặt $U := \{\mathbf{x} \in C : f(\mathbf{x}) \leq f(\mathbf{x}^k), \forall k\}$. Với mọi $\hat{\mathbf{x}} \in U$:

$$\langle \nabla f(\mathbf{x}^k), \hat{\mathbf{x}} - \mathbf{x}^k \rangle \leq 0$$

kết hợp (6) ta thu được:

$$\|\mathbf{x}^{k+1} - \hat{\mathbf{x}}\|^2 \leq \|\mathbf{x}^k - \hat{\mathbf{x}}\|^2 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + 2\lambda_k \langle \nabla f(\mathbf{x}^k), \mathbf{x}^k - \mathbf{x}^{k+1} \rangle$$

Áp dụng **Bổ đề 1** với $a_k = \|\mathbf{x}^{k+1} - \hat{\mathbf{x}}\|^2$ và $b_k = 2\lambda_k \nabla f(\mathbf{x}^k)^T(\mathbf{x}^k - \mathbf{x}^{k+1})$:
$\Longrightarrow$ $\{\|\mathbf{x}^k - \hat{\mathbf{x}}\|\}$ hội tụ với mọi $\hat{\mathbf{x}} \in U$.
$\Longrightarrow$ Tồn tại $\{\mathbf{x}^{k_j}\} \subset \{\mathbf{x}^k\}$ hội tụ về $\bar{\mathbf{x}} \in C$.
Từ (4) ta có $\lim_{k\to\infty} f(\mathbf{x}^k) = f(\bar{\mathbf{x}})$ và $f(\bar{\mathbf{x}}) \leq f(\mathbf{x}^k)$ với mọi $k \geq 0$
$\Longrightarrow$ $\bar{\mathbf{x}} \in U$ và $\lim_{k\to\infty} \|\mathbf{x}^k - \bar{\mathbf{x}}\| = \lim_{j\to\infty} \|\mathbf{x}^{k_j} - \bar{\mathbf{x}}\| = 0$.
Kết luận: Dãy $\{\mathbf{x}^k\}$ hội tụ duy nhất về điểm dừng $\bar{\mathbf{x}}$. Nếu f giả lồi, $\bar{\mathbf{x}}$ là **nghiệm tối ưu**. ■

Nhận xét 2

◇ Trong Thuật toán GDA, ta có thể chọn $\lambda_0 = \lambda$, với hằng số $\lambda \leq 2(1-\sigma)/L$. Thuật toán GDA áp dụng được cho bước nhảy hằng số $\lambda \leq 2(1-\sigma)/L$.

Nhận xét 2

- ◇ Trong Thuật toán GDA, ta có thể chọn $\lambda_0 = \lambda$, với hằng số $\lambda \leq 2(1-\sigma)/L$. Thuật toán GDA áp dụng được cho bước nhảy hằng số $\lambda \leq 2(1-\sigma)/L$.
- ◇ Nếu giá trị hằng số Lipschitz L được biết trước, ta có thể chọn bước nhảy hằng số $\lambda \in (0, 2/L)$ như trong thuật toán gradient descent (GD) để giải các bài toán quy hoạch lồi.

Algorithm 2 (Gradient Descent Algorithm-GD)

Input: $x^0 \in C$; $\lambda \in (0, 2/L)$; $k = 0$
Output: Stationary point x^*
while true **do**
    $x^{k+1} \leftarrow P_C\big(x^k - \lambda \nabla f(x^k)\big)$
    if $x^{k+1} = x^k$ **then**
        break
    end if
    $k \leftarrow k + 1$
end while

Hệ quả 1:

Giả sử f lồi, $C = \mathbb{R}^m$ và $\{x^k\}$ là dãy được sinh ra bởi thuật toán GDA. Khi đó,

$$f(x^k) - f(x^*) = O\left(\frac{1}{k}\right).$$

với x^* là nghiệm tối ưu của bài toán.

1. Thiết lập bài toán: Gọi x^* là nghiệm tối ưu. Đặt $\Delta_k := f(x^k) - f(x^*) \geq 0$.
2. Các đánh giá cơ bản: Từ (4) kết hợp $x^k - x^{k+1} = \lambda \nabla f(x^k)$, ta có:

$$\Delta_{k+1} \leq \Delta_k - \sigma\lambda \|\nabla f(x^k)\|^2, \quad \forall k \geq k_1 \tag{8}$$

Do dãy $\{x^k\}$ bị chặn và f lồi nên :

$$\begin{aligned} \Delta_k &\leq \langle \nabla f(x^k), x^k - x^* \rangle \\ &\leq \underbrace{\|x^k - x^*\|}_{\leq M} \|\nabla f(x^k)\| \implies \|\nabla f(x^k)\| \geq \frac{\Delta_k}{M} \end{aligned} \tag{9}$$

trong đó $M := \sup\{\|x^k - x^*\| : k \geq k_1\} < \infty$.

3. Bất đẳng thức truy hồi:
Thay (9) vào (8), đặt $Q := \frac{\sigma\lambda}{M^2}$:

$$\Delta_{k+1} \leq \Delta_k - \sigma\lambda\frac{\Delta_k^2}{M^2} = \Delta_k - Q\Delta_k^2 \tag{10}$$

4. Đánh giá tốc độ: Vì $\Delta_{k+1} \leq \Delta_k$, chia hai vế (10) cho $\Delta_k\Delta_{k+1}$:

$$\frac{1}{\Delta_{k+1}} \geq \frac{1}{\Delta_k} + Q\frac{\Delta_k}{\Delta_{k+1}} \geq \frac{1}{\Delta_k} + Q \implies \frac{1}{\Delta_k} \geq \frac{1}{\Delta_{k_1}} + (k-k_1)Q$$
$$\implies \Delta_k \leq \frac{1}{(k-k_1)Q}$$

Kết luận:

$$f(x^k) - f(x^*) = O\left(\frac{1}{k}\right).$$

1. Động lực Ứng dụng trong Học sâu quy mô lớn (Large-scale Deep Learning) nơi việc tính toán toàn bộ gradient $\nabla f(x)$ là bất khả thi.

2. Bài toán Tối ưu Ngẫu nhiên: Thay vì tối thiểu hóa hàm mục tiêu xác định, ta xét bài toán kỳ vọng:

$$\min_x \mathbb{E}[f_\xi(x)]. \tag{11}$$

Trong đó:

- x: Biến quyết định (tham số mô hình).
- ξ: Biến ngẫu nhiên với phân phối xác định.
- $f_\xi(x)$: Hàm L-smooth đối với mỗi ξ.

3. Xấp xỉ Gradient Tại mỗi bước lặp k, thay vì tính $\nabla\Phi(x^k) = \mathbb{E}[\nabla f_\xi(x^k)]$, ta lấy mẫu:

- ◇ Lấy mẫu ngẫu nhiên ξ tại mỗi lần lặp k.
- ◇ Tính gradient xấp xỉ: $\nabla f_{\xi^k}(x^k)$.

4. Các thành phần trong Học máy:

Ký hiệu	Ý nghĩa thực tế
x	Trọng số mạng nơ-ron
ξ	Tính ngẫu nhiên của việc chọn dữ liệu
$f_\xi(x)$	Hàm mất mát trên một batch dữ liệu
$\Phi(x)$	Hàm mất mát tổng quát

**Lưu ý: Các phân tích lý thuyết chặt chẽ cho SGDA được dành cho nghiên cứu tương lai.*

Algorithm 3 (Stochastic Gradient Descent Algorithm-SGDA)

Input: $x^0 \in C$; $\lambda_0 > 0$; $\sigma, \kappa \in (0,1)$; $k = 0$
Output: Stationary point x^*
while true **do**
    Lấy mẫu ngẫu nhiên ξ_k
    $x_{k+1} \leftarrow P_C\big(x_k - \lambda_k \nabla f_{\xi_k}(x_k)\big)$
    if $f_{\xi_k}(x_{k+1}) \leq f_{\xi_k}(x_k) - \sigma\langle \nabla f_{\xi_k}(x_k), x_k - x_{k+1}\rangle$ **then**
        $\lambda_{k+1} \leftarrow \lambda_k$
    else
        $\lambda_{k+1} \leftarrow \kappa\lambda_k$
    end if
    if $x_{k+1} = x_k$ **then**
        break
    end if
    $k \leftarrow k + 1$
end while

Mục tiêu

Đánh giá hiệu quả của GDA thông qua đối sánh với các phương pháp:

- ◇ **Gradient Descent (GD)**: Cho **Bài toán Lồi** (Ví dụ 3).
- ◇ **Neurodynamic (RNN)**: Cho **Bài toán Phi/Giả lồi** (Ví dụ 1, 2, 4).

Quy ước & Tiêu chí dừng

- ◇ **Điều kiện dừng thuật toán:**
  - Số vòng lặp đạt giới hạn tối đa (Iterations $\geq$ #Iter).
  - Hoặc khi **đạt điều kiện hội tụ** (Convergence condition met).
- ◇ **Các ký hiệu sử dụng:**
  - x^*: Điểm giới hạn (nghiệm) của dãy lặp $\{x_k\}$.
  - **Time**: Thời gian CPU thực chạy (không tính thời gian khởi tạo).
  - **Value**: Giá trị hàm mục tiêu tại điểm dừng.

Nội dung

Mô hình bài toán

Hàm mục tiêu phi lồi:

$$\min f(x) = \frac{x_1^2 + x_2^2 + 3}{1 + 2x_1 + 8x_2}$$

Ràng buộc tập C:

$$C = \{(x_1, x_2)^\top \in \mathbb{R}^2 \mid -x_1^2 - 2x_1x_2 \leq -4,\ x_1, x_2 \geq 0\}$$

Nhận xét

Hàm f là giả lồi (pseudoconvex) trên C. Mục tiêu là so sánh khả năng hội tụ về nghiệm tối ưu toàn cục so với RNN.

Kết quả hội tụ

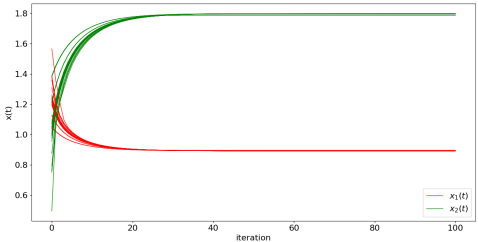
- Nghiệm tối ưu x^*:

$$(0.893870, 1.790527)$$

- Value (GDA): **0.409362** (Tốt hơn RNN: 0.410100).

Đánh giá

- **Tính ổn định:** Quỹ đạo nghiệm hội tụ mượt, không nhạy với nhiễu.
- **Hiệu quả:** Cân bằng tốt giữa *chất lượng nghiệm* và *chi phí tính toán*.



Hình 1: Quỹ đạo nghiệm hội tụ

Bảng 1: Kết quả các lần chạy (Results of Iterations)

#	Time (s)	Iters	Value	x^{*t}
1	1.489085	**28**	0.409364	[0.894247, 1.789397]
2	1.235719	32	**0.409362**	[0.893870, 1.790527]
3	1.082686	31	0.409363	[0.893946, 1.790301]
4	**1.047740**	31	0.409363	[0.894092, 1.789862]
5	1.440842	32	0.409364	[0.894284, 1.789285]
6	1.139069	30	0.409367	[0.895131, 1.786744]
7	1.126395	32	0.409363	[0.894047, 1.789995]
8	1.564762	33	0.409363	[0.893926, 1.790359]
9	2.008716	33	0.409363	[0.893998, 1.790141]
10	1.510566	32	**0.409362**	[0.893911, 1.790403]

Dữ liệu thực nghiệm 10 lần chạy đầu tiên

Hàm mục tiêu (Không trơn):

$$\min f(x) = \frac{e^{|x_2-3|} - 30}{x_1^2 + x_3^2 + 2x_4^2 + 4}$$

Hệ ràng buộc phức tạp:

$$g_1(x) = (x_1 + x_3)^3 + 2x_4^2 \leq 10$$
$$g_2(x) = (x_2 - 1)^2 \leq 1, \quad h(x) = 2x_1 + 4x_2 + x_3 = -1$$

Xử lý đạo hàm tại điểm không trơn:

$$\nabla|x_2 - 3| = \frac{x_2 - 3}{|x_2 - 3|} \quad (\text{với } x_2 \neq 3)$$

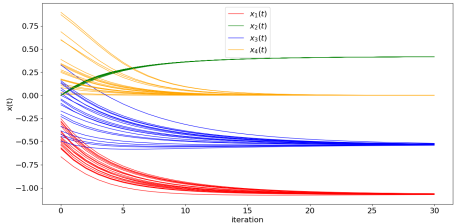
Kết quả mong đợi: Tìm được nghiệm tốt hơn so với RNN (vốn hay bị kẹt ở cực tiểu địa phương đối với hàm không trơn).

Kết quả hội tụ

- Nghiệm tối ưu x^*: $(-1.067557, 0.417538, -0.535037, 0.000020)^\top$
- Value (GDA): **-3.090768**.
- Value (RNN): -3.084900.

Đánh giá

- **Tính ổn định:** Hội tụ về cùng một nghiệm tối ưu bất kể điểm khởi tạo ban đầu khác nhau.
- **Hiệu quả:** GDA vượt trội RNN cả về **độ chính xác**, **thời gian tính toán** và cấu trúc đơn giản hơn.



Hình 2: Sự hội tụ đồng nhất của 4 biến số

Bảng 2: Kết quả tối ưu (Optimization Results)

#	Time (s)	Iters	Value	x^{*t}
1	3.125859	38	-3.090767	[-1.069016, 0.417647, -0.532558, 0.000022]
2	3.376504	42	-3.090767	[-1.069795, 0.417966, -0.532273, 0.000000]
3	3.359701	41	-3.090767	[-1.069610, 0.417896, -0.532363, 0.000014]
4	3.596762	44	-3.090767	[-1.069956, 0.418058, -0.532320, 0.000005]
5	3.124562	41	-3.090767	[-1.069584, 0.417901, -0.532434, 0.000007]
6	3.700269	47	-3.090767	[-1.070102, 0.418157, -0.532424, 0.000004]
7	3.406564	44	-3.090767	[-1.069881, 0.418059, -0.532475, 0.000001]
8	**2.954161**	**37**	**-3.090768**	[-1.067557, 0.417538, -0.535037, 0.000020]
9	3.307867	47	-3.090767	[-1.070151, 0.418162, -0.532345, 0.000000]
10	3.687243	44	-3.090767	[-1.069958, 0.418055, -0.532303, 0.000009]

Giá trị hàm mục tiêu hội tụ ổn định

Cho $e := (1, \ldots, n) \in \mathbb{R}^n$, với $\alpha > 0$, $\beta > 0$ thỏa mãn

$$2\alpha > 3\beta^{3/2}\sqrt{n}.$$

Xét bài toán tối ưu $\mathrm{OP}(f, C)$ với

$$f(x) := a^\top x + \alpha x^\top x + \frac{\beta}{\sqrt{1 + \beta x^\top x}} e^\top x.$$

$$C = \left\{ x \in \mathbb{R}^n_{++} : x_1 x_2 \cdots x_n \geq 1 \right\}.$$

$$\beta = 0.741271, \qquad \alpha = 3\beta^{3/2}\sqrt{n+1}.$$

$$L = 4\beta^{3/2}\sqrt{n} + 3\alpha.$$

So sánh hai phương pháp:

- ◇ **GD:** $\lambda = 1/L$.
- ◇ **GDA:** $\lambda_0 = 5/L$, điều chỉnh thích nghi.

n	GDA			GD		
	$f(x^*)$	#Iter	Time(s)	$f(x^*)$	#Iter	Time(s)
10	**80.080568**	**3**	**0.2447**	80.080573	7	0.3834
20	**219.714451**	**4**	**0.6099**	219.714455	8	0.6726
50	**851.700027**	**4**	**0.9028**	851.700030	9	2.5383
100	**2390.314031**	**4**	**5.6882**	2390.314040	9	13.3452
200	**6721.095882**	**5**	**41.3877**	6721.095887	10	53.0749
500	**26426.158664**	**5**	**91.4714**	26426.158682	10	262.8033
600	**34711.054331**	**5**	**160.1987**	34711.054335	11	353.6806
700	**43704.962204**	**5**	**199.9534**	43704.962212	16	1726.5777
800	**53372.127462**	**5**	**352.0395**	53372.127469	11	930.0608
900	**63653.989656**	**11**	**1589.4307**	63653.989663	11	2123.5983
1000	**74522.902256**	**7**	**1260.0110**	74522.902265	11	1617.5378

Bảng 1: So sánh GDA và GD

- ◇ GDA và GD cho giá trị hàm mục tiêu tối ưu gần như tương đương.
- ◇ GDA thường hội tụ với ít vòng lặp hơn so với GD.
- ◇ Hiệu quả của GDA thể hiện rõ hơn khi kích thước bài toán tăng.
- ◇ Cơ chế cỡ bước tự thích nghi giúp GDA cải thiện hiệu suất tính toán.

Xét bài toán $\mathrm{OP}(f, C)$ với hàm mục tiêu

$$f(x) = -\exp\left(-\sum_{i=1}^{n} \frac{x_i^2}{e_i^2}\right),$$

trong đó $x \in \mathbb{R}^n$, $e = (e_1, \ldots, e_n)^\top$, $e_i > 0$.
Hàm f là giả lồi trên miền ràng buộc lồi

$$C := \{Ax = b,\ g(x) \leq 0\}.$$

Ma trận $A = (a_1, \ldots, a_n) \in \mathbb{R}^{1\times n}$ được xác định bởi

$$a_i = \begin{cases} 1, & 1 \le i \le n/2, \\ 3, & n/2 < i \le n, \end{cases} \qquad b = 16.$$

Các ràng buộc bất đẳng thức:

$$g_i(x) = \sum_{j=1}^{10} x^2_{10(i-1)+j} - 20, \qquad i = 1, 2, \ldots, \frac{n}{10}.$$

So sánh hai phương pháp:

- ◇ **GDA**: thuật toán đề xuất.
- ◇ **RNN**: thuật toán của Liu et al. (2022).

Chỉ số đánh giá được sử dụng:

$$-\ln(-f(x^*)),$$

trong đó x^* là nghiệm gần đúng thu được từ mỗi thuật toán.
Hàm $-\ln(-z)$ là đơn điệu tăng với mọi $z < 0$.

n	GDA			RNN		
	$-\ln(-f(x^*))$	#Iter	Time	$-\ln(-f(x^*))$	#Iter	Time
10	**5.1200**	**50**	**2.597**	5.2014	5000	1123
20	**2.56000**	**50**	**4.542**	2.66910	5000	1264
30	**1.70667**	**50**	**5.448**	2.0824	5000	1562
40	**1.28000**	**50**	**16.3**	1.38552	5000	1662.3
50	**1.02400**	**50**	**12.31**	1.13281	5000	1741.3
60	**0.85334**	**500**	**248.5**	1.07326	5000	1984.8
70	**0.73143**	**500**	**78.4**	2.30464	5000	1443.2
80	**0.64000**	**500**	**298**	7.98621	5000	2338.6
90	**0.56889**	**500**	**272.3**	9.12685	5000	2743.2
100	**0.51200**	**500**	**617.64**	10.88262	5000	2985.26
150	**0.34134**	**500**	**1508**	17.69445	5000	3240.36
200	**0.25600**	**500**	**1748.3**	18.99267	5000	4004.87

Bảng 2: Kết quả tính toán cho Ví dụ 4

- ◇ GDA cho **giá trị tối ưu gần đúng tốt hơn** so với RNN.
- ◇ GDA hội tụ với **số vòng lặp ít hơn đáng kể**.
- ◇ **Thời gian tính toán thấp hơn nhiều**, đặc biệt khi n lớn.

Kết quả cho thấy GDA vượt trội hơn RNN trong các bài toán giả lồi kích thước lớn.

Mục tiêu nghiên cứu

Phương pháp được đề xuất, giống như thuật toán GD, có nhiều ứng dụng trong học máy. Chúng tôi phân tích ba ứng dụng phổ biến để chứng minh độ chính xác và hiệu quả tính toán so với các phương pháp thay thế khác:

1. **Lựa chọn đặc trưng có giám sát** (Supervised feature selection).
2. **Hồi quy Logistic** (Regression).
3. **Phân loại** (Classification).

◇ **1. Lựa chọn đặc trưng:**
- *Mô hình:* Cực tiểu hóa hàm phân thức **giả lồi** trên tập lồi (lớp con của bài toán $OP(f, C)$).
- *Mục đích:* So sánh với phương pháp RNN.

◇ **2. Hồi quy Logistic đa biến:**
- *Mô hình:* Bài toán **quy hoạch lồi**.
- *Giải pháp:* Dùng GDA và các biến thể GD.

◇ **3. Mạng Nơ-ron (Phân loại ảnh):**
- *Mô hình:* Hàm mục tiêu **không lồi, không tựa lồi**.
- *Giải pháp:* Dùng biến thể ngẫu nhiên **SGDA** (heuristic).
- *Hội tụ:* Nếu dãy điểm có giới hạn $\rightarrow$ hội tụ về điểm dừng (Định lý 1).

Đầu vào:

- ◇ Tập p-đặc trưng $\mathcal{F} = \{F_1, ..., F_p\}$.
- ◇ Tập n-mẫu $\{(x_i, y_i) | i = 1, ..., n\}$.
- ◇ x_i: vector đặc trưng p-chiều; $y_i \in \{1, ..., m\}$: nhãn lớp.

Mục tiêu: Chọn tập con tối ưu $\{F_1, ..., F_k\} \subseteq \mathcal{F}$ thỏa mãn:

1. **Sự dư thừa ít nhất (Min Redundancy):** Cực tiểu hóa $w^T Q w$ (Q là ma trận bán xác định dương).
2. **Mức độ liên quan cao nhất (Max Relevance):** Cực đại hóa $\rho^T w$ (ρ là vector tham số liên quan).

Kết hợp hai mục tiêu trên, ta có bài toán tối ưu hàm phân thức:

$$\begin{aligned} &\text{minimize} \quad \frac{w^T Q w}{\rho^T w} \\ &\text{subject to} \quad e^T w = 1, \quad w \geq 0 \end{aligned} \tag{12}$$

Trong đó:

- $\diamond$ $w = (w_1, ..., w_p)^T$: Vector điểm số đặc trưng cần xác định.

Nhận xét quan trọng: Vì hàm mục tiêu là phân thức của một hàm lồi trên một hàm tuyến tính dương $\rightarrow$ Nó là **hàm giả lồi** trên tập ràng buộc. $\Rightarrow$ Có thể giải bằng **Thuật toán GDA**.

1. Ma trận hệ số tương đồng $Q = \delta I_p + S$: Với $S = (s_{ij})$ xác định bởi:

$$s_{ij} = \max\left\{0, \frac{I(F_i; F_j; y)}{H(F_i) + H(F_j)}\right\}$$

Trong đó:

- ◇ Entropy thông tin: $H(\hat{X}) = -\sum_{\hat{x}\in\hat{X}} p(\hat{x}) \log p(\hat{x})$
- ◇ Đa thông tin: $I(\hat{X}; \hat{Y}; \hat{Z}) = I(\hat{X}; \hat{Y}) - I(\hat{X}; \hat{Y}|\hat{Z})$
- ◇ Thông tin tương hỗ có điều kiện: $I(\hat{X}; \hat{Y}|\hat{Z}) = \sum\sum\sum p(\hat{x}, \hat{y}, \hat{z}) \log \frac{p(\hat{x},\hat{y}|\hat{z})}{p(\hat{x}|\hat{z})p(\hat{y}|\hat{z})}$

2. Vector mức độ liên quan ρ (Fisher score):

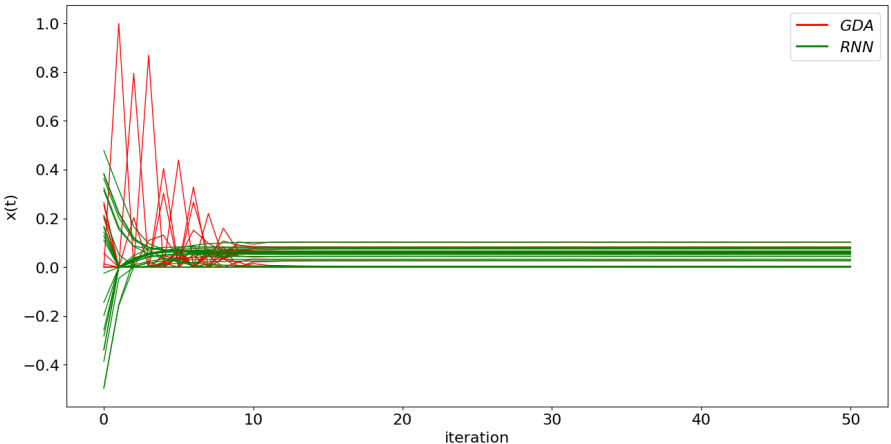
$$\rho(F_i) = \frac{\sum_{j=1}^{K} n_j(\mu_{ij} - \mu_i)^2}{\sum_{j=1}^{K} n_j \sigma_{ij}^2}$$

So sánh Thuật toán GDA (đề xuất) và Thuật toán RNN (Wang et al. 2021):

Chỉ số	**Thuật toán GDA**	**Thuật toán RNN**
Giá trị tối ưu $f(w^*)$	**0.153478**	0.153911
Thời gian tính toán (T)	**10.034473s**	49.324688s

Kết luận:

- ◇ Thuật toán GDA tìm được giá trị tối ưu tốt hơn (nhỏ hơn).
- ◇ Thời gian tính toán nhanh hơn gần gấp 5 lần so với RNN.
- ◇ GDA vượt trội cả về độ chính xác và hiệu quả tính toán.



Hình 3: Kết quả cho bài toán lựa chọn đặc trưng

Bài toán:

- ◇ Dữ liệu: N quan sát $(a_i, b_i) \in \mathbb{R}^d \times \mathbb{R}$.
- ◇ Hàm mất mát (Cross-entropy + chuẩn hóa L_2):

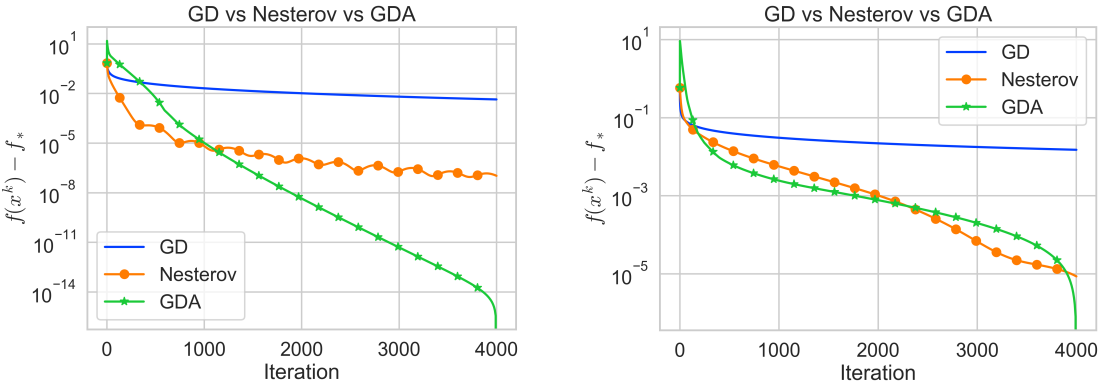
$$\bar{J}(x) = -\sum_{i=1}^{N} \left(b_i \log(\sigma(-x^T a_i)) + (1 - b_i) \log(1 - \sigma(-x^T a_i)) \right) + \frac{1}{2N} \|x\|^2$$

Tham số thuật toán:

- ◇ Hệ số Lipschitz ước lượng: $L \approx \frac{1}{2N}(\|A\|^2/2 + 1)$.
- ◇ So sánh với:
  1. GD (bước nhảy $1/L$).
  2. Nesterov's accelerated method.

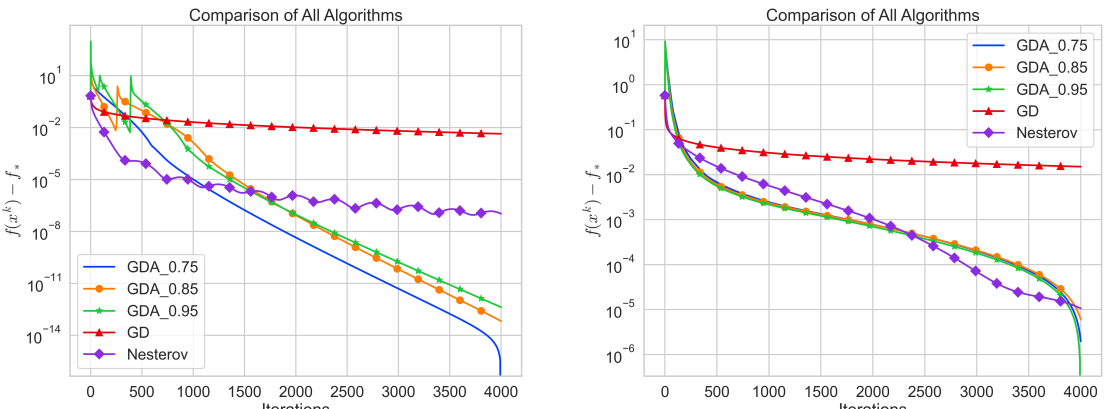
Phân tích biểu đồ (Dataset: Mushrooms và W8a):

- GDA vượt trội hơn GD và Nesterov về giá trị hàm mục tiêu.



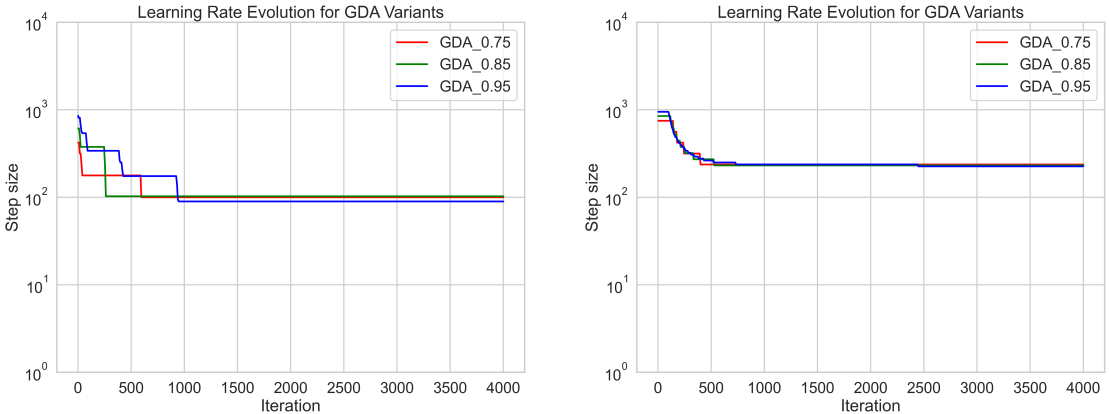
Hình 4: Kết quả so sánh đối với dataset Mushrooms (bên trái) và W8a (bên phải)

• **Hình 5** minh họa cơ chế tự thích nghi: Bước nhảy giảm dần từ giá trị ban đầu (với hệ số $\kappa = 0.75$)



Hình 5: Kết quả so sánh đối với dataset Mushrooms (bên trái) và W8a (bên phải)

• **Hình 6** trình bày sự giảm kích thước bước nhảy từ một kích thước bước nhảy ban đầu liên quan đến các kết quả trong Hình 5.



Hình 6: Sự thay đổi bước nhảy đối với dataset Mushrooms (bên trái) và W8a (bên phải)

Mục tiêu: Triển khai thuật toán đề xuất vào mô hình huấn luyện mạng nơ-ron (bài toán không lồi).

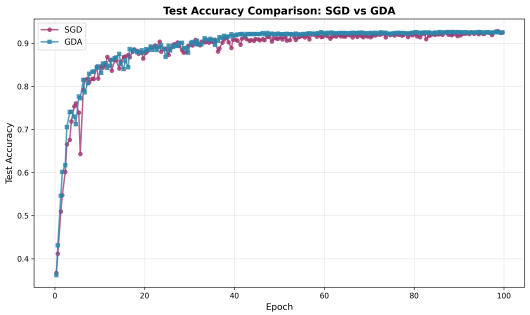
Cấu hình thực nghiệm:

- ◇ **Mô hình:** Kiến trúc ResNet-18 tiêu chuẩn (PyTorch).
- ◇ **Dữ liệu:** Cifar10 (Phân loại ảnh).
- ◇ **Hàm mất mát:** Cross-entropy.
- ◇ **Tham số:** Sử dụng thiết lập mặc định của Adam.

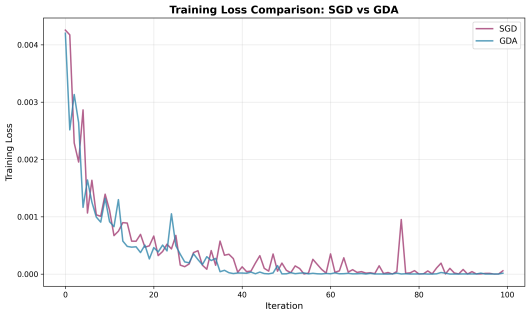
Phương pháp so sánh:

- ◇ Biến thể ngẫu nhiên của GDA (**Thuật toán SGDA**).
- ◇ So với: Stochastic Gradient Descent (**SGD**).

Kết quả (Hình 7 & 8): SGDA vượt trội hơn SGD về cả hai chỉ số quan trọng:

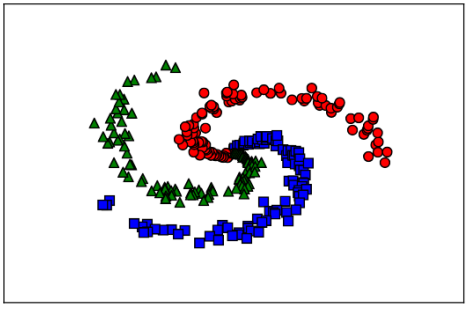


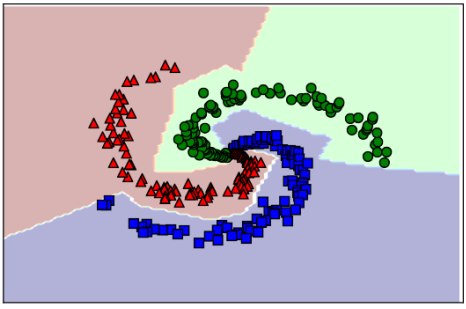
Hình 7: Độ chính xác kiểm thử (test accuracy) theo số vòng lặp khi huấn luyện ResNet-18.



Hình 8: Hàm mất mất huấn luyện (training loss) theo số vòng lặp cho ResNet-18.

Bài toán phân loại 3 lớp, biểu diễn trên đồ thị 2D như sau:





Mục tiêu là tạo ra mô hình Neural network có thể tạo ra ranh giới tốt giữa 3 lớp.

Hàm mất mát của mô hình là hàm cross-entropy:

$$J \triangleq J(\mathbf{W}, \mathbf{b}; \mathbf{X}, \mathbf{Y}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} y_{ji} \log(\hat{y}_{ji})$$

Bài toán tối ưu (training objective)

$$\min_{x \in \mathbb{R}^n} f(x),$$

trong đó x là vector tham số (weights) của mô hình, và $f(x)$ là training loss.

Gradient Descent (GD)

$$x^{k+1} = x^k - \eta \nabla f(x^k), \qquad \eta > 0 \text{ fixed.}$$

- ◇ **Điểm mạnh:** mỗi iteration rẻ, triển khai đơn giản.
- ◇ **Điểm yếu:** nhạy với learning rate η; η quá lớn gây dao động, quá nhỏ gây hội tụ chậm.
- ◇ Động lực của các biến thể GDA: **điều chỉnh step size** để ổn định hơn hoặc giảm công tuning.

GDA-1 (paper)

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

và kiểm tra giảm đủ (sufficient decrease):

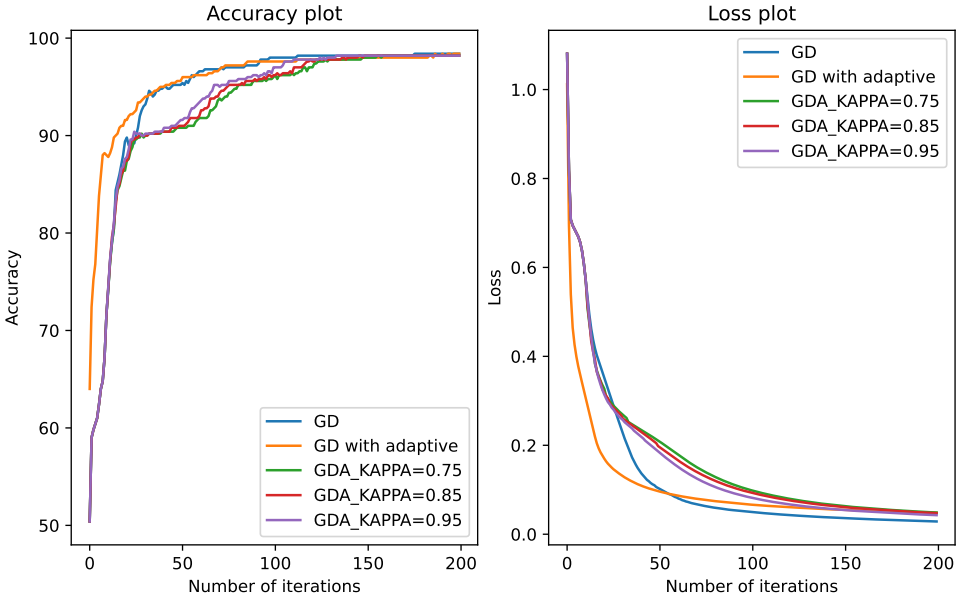
$$f(x^{k+1}) \le f(x^k) - \sigma \langle \nabla f(x^k),\, x^k - x^{k+1} \rangle, \quad \sigma \in (0,1).$$

Nếu không đạt, shrink step size: $\lambda_{k+1} = \kappa \lambda_k$, $\kappa \in (0,1)$.

GDA-2 (added)

$$x^{k+1} = x^k - \lambda_k \nabla f(x^k), \qquad \lambda_k = \frac{\alpha_k}{\max\{1,\ \|\nabla f(x^k)\|\}}.$$

- ◇ Khi $\|\nabla f(x^k)\|$ lớn: λ_k nhỏ lại $\Rightarrow$ hạn chế bước đi quá mạnh (stabilization).
- ◇ Khi $\|\nabla f(x^k)\| \le 1$: $\lambda_k \approx \alpha_k \Rightarrow$ gần giống GD với step size α_k.


Accuracy plot
Accuracy
Number of iterations
GD
GD with adaptive
GDA_KAPPA=0.75
GDA_KAPPA=0.85
GDA_KAPPA=0.95
Loss plot
Loss
Number of iterations
GD
GD with adaptive
GDA_KAPPA=0.75
GDA_KAPPA=0.85
GDA_KAPPA=0.95

Runtime

Method	Time (s)	vs GD
GD	1.999849	1.00×
GDA ($\kappa = 0.75$)	3.019701	1.51×
GDA ($\kappa = 0.85$)	3.046629	1.52×
GDA ($\kappa = 0.95$)	3.140818	1.57×
Adaptive GD ($p = 0.51$)	2.249212	1.12×

Cùng thiết lập thí nghiệm; khác biệt đến từ optimizer.

Kết quả

- ◇ **Accuracy cuối:** cả 3 nhóm tiệm cận gần như tương đương (xấp xỉ 98–99%).
- ◇ **Loss:** GD thường đạt **loss cuối thấp nhất**; Adaptive GD giảm rất nhanh giai đoạn đầu; GDA-κ giảm chậm hơn trong cùng số iteration.
- ◇ **Trade-off quan sát được:** GDA-κ tăng chi phí thời gian $\approx$51–57% so với GD, trong khi chất lượng cuối không cải thiện rõ trong setup này.

1. Bauschke HH, Combettes PL (2011) Convex analysis and monotone operator theory in hilbert spaces. Springer Convex analysis and monotone operator theory in hilbert spaces
2. Bian W, Ma L, Qin S, Xue X (2018) Neural network for nonsmooth pseudoconvex optimization with general convex constraints. Neural Netw 101: 1–14
3. Boyd SP, Vandenberghe L (2009) Convex optimization. Cambridge University Press, Cambridge
4. Cevher V, Becker S, Schmidt M (2014) Convex optimization for big data: Sparse and distributed algorithms. Signal Process Mag 31:32-4
5. Dennis JE, Schnabel RB (1983) Numerical methods for unconstrained optimization and nonlinear equations. Prentice-Hall,New Jersey
6. Ferreira OP, Sosa WS (2022) On the Frank-Wolfe algorithm for non-convex optimization problems. Optimization 71(1): 197-211
7. Hu Y, Li X, Wei K (2020) Convergence rates of subgradient methods for quasiconvex optimization problems. Comput Optim Appl 77: 183-212
8. Kiwiel KC (2001) Convergence rates of subgradient methods for quasiconvex minimization. Math Program Ser A 90:1-25
9. Konnov IV (2018) Simplified versions of the conditional gradient method. Optimization 67(12): 2275-2290
10. Lan GH (2020) First-order and stochastic optimization methods for machine learning. Springer series in data sciences. Springer Nature
11. Liu N, Wang J, Qin S (20232) A one-layer recurrent neural network for nonsmooth pseudoconvex optimization with quasiconvex inequality and affine equality constraints. Neural Netw 147: 1-9

12. Malitsky Y, Mishchenko K (2020) Adaptive gradient descent without descent. Proc Mach Learn Res 119: 6702-6712
13. Mangasarian O (1965) Pseudo-convex functions. SIAM Control 3: 281-290
14. Nesterov Y (2013) Introductory lectures on convex optimization: a basic course, vol 87. Springer Science & Business Media
15. Rockafellar RT (1970) Convex analysis. Princeton University Press, Princeton
16. Thang TN, Solanki VK, Dao TA, Anh NTN, Hai PV (2020) A monotonic optimization approach for solving strictly quasiconvex multiobjective problems. J Intell Fuzzy Syst 38: 6053-6063
17. Wang Y, Li X, Wang J (2021) A neurodynamic optimization approach to supervised feature selection via fractional programming. Neural Netw 136: 194-206
18. Xu HK (2002) Iterative algorithms for nonlinear operators. J Lond Math Soc 66: 240-256
19. Yu CK, Hu Y, Yang X, Choy SK (2019) Abstract convergence theorem for quasi-convex programming problems with applications. Optimization 68(7): 1289-1304
20. `https://github.com/nguyenphongg233/NMPPTU-Team3-SAAQP`

Cảm ơn mọi người đã chú ý lắng nghe!