Thuật toán tự thích nghi cho lập trình tựa lồi và ứng dụng trong học máy

Trần Ngọc Thăng[1] Trịnh Ngọc Hải[1]

[1]Khoa Toán - Tin
Đại học Bách Khoa Hà Nội, Hai Bà Trưng, Hà Nội, Việt Nam

Thông tin nhóm

Nhóm thuyết trình: Nhóm 1

Các thành viên:

Nguyễn Ngọc Tuấn Anh	202400029
Trần Tuấn Anh	202416124
Nguyễn Hữu Chính	202416143
Trần Mạnh Cường	202416147
Đỗ Hải Đăng	202400035
Bùi Tuấn Đạt	202400096
Nguyễn Hồng Đạt	202416153
Phạm Đình Minh Đức	202400038
Bùi Tiến Dũng	202416167
Trần Huy Dương	20230025
Nguyễn Khắc Duy	202400100
Nguyễn Hoàng Gia	202400040

Mục lục

Đặt vấn đề

- **Vai trò của GD:** Là công cụ phổ biến giải quyết các bài toán quy hoạch từ lồi đến không lồi, với nhiều ứng dụng thực tế. [3] [4] [10]
- **Thực trạng:** Các bài toán học máy thường có số chiều lớn và hàm mục tiêu **không lồi**. [4] [10]
- **Hạn chế của một số phương pháp hiện tại:**
  - Line-search (Tìm kiếm theo tia): Chi phí tính toán quá lớn cho mỗi lần lặp, đặc biệt khi tính giá trị hàm phức tạp. [3]
  - Sử dụng hằng số Lipschitz (biết trước): Dẫn đến hội tụ chậm do chỉ dùng một phần ước lượng không chính xác, hoặc thông tin này không có sẵn. [8] [14]
  - Độ dài bước giảm dần (Diminishing step-size): Gây ra tốc độ hội tụ chậm (ví dụ: phương pháp của Kiwiel cho quy hoạch tựa lồi). [8]

Phương pháp Gradient Descent Adaptive [17]

- **Mục tiêu của bài báo:** Đề xuất thuật toán độ dài bước tự thích nghi mới.
- **Phạm vi áp dụng:**
  - Hàm mục tiêu: Không lồi và trơn.
  - Tập ràng buộc: Không bị chặn, đóng và lồi.
- **Cơ chế hoạt động:** Giảm dần độ dài bước một cách ổn định cho đến khi thỏa mãn một điều kiện xác định, không cần biết trước hằng số Lipschitz.
- **Ứng dụng:** Hiệu quả cho các bài toán quy mô lớn như hồi quy logistic đa biến và mạng nơ-ron.

Mục lục

Thiết lập bài toán

Bài toán tối ưu hóa tổng quát (OP)

Chúng ta xét bài toán tối ưu hóa tổng quát $(OP(f, C))$:

$$\min_{x \in C} f(x)$$

Các giả thiết cơ bản:

- $C \subset \mathbb{R}^m$: Là tập **lồi**, **đóng** và khác rỗng (có thể không bị chặn).
- $f : \mathbb{R}^m \to \mathbb{R}$: Là hàm khả vi trên tập mở chứa C.
- ∇f: Có tính chất **liên tục Lipschitz** trên C, tức là tồn tại $L > 0$ sao cho:
$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y \in C$$
- Giả sử tập nghiệm của $(OP(f, C))$ là khác rỗng.

Phép chiếu lên tập lồi

Thuật toán sử dụng phép chiếu $P_C(x)$ để đảm bảo nghiệm luôn nằm trong vùng chấp nhận.

Định nghĩa 1

$$P_C(x) := \text{argmin}\{\|z - x\| : z \in C\}$$

Mệnh đề 1 (Bauschke và Combettes 2011 [1])

Với mọi $x, y \in \mathbb{R}^m$ và $z \in C$:

1. $\|P_C(x) - P_C(y)\| \leq \|x - y\|$ (Tính không giãn).
2. $\langle z - P_C(x), x - P_C(x)\rangle \leq 0$

Định nghĩa

Định nghĩa 2 (Mangasarian (1965) [13])

Hàm f được gọi là:

- **Lồi (Convex):** $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$.
- **Giả lồi (Pseudoconvex):** $\langle \nabla f(x), y - x \rangle \geq 0 \Rightarrow f(y) \geq f(x)$.
- **Tựa lồi (Quasiconvex):** $f(\lambda x + (1-\lambda)y) \leq \max\{f(x); f(y)\}$.

Với mọi $x, y \in C$ và $\lambda \in [0, 1]$.

Mối quan hệ bao hàm:

$$\text{Lồi} \implies \text{Giả lồi} \implies \text{Tựa lồi}$$

Điều kiện của Hàm tựa lồi

Để chứng minh sự hội tụ của thuật toán trên hàm tựa lồi, ta cần tính chất sau:

Mệnh đề 2 (Dennis và Schnabel 1983 [5])

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

$$f(y) \leq f(x) \implies \langle \nabla f(x), y - x \rangle \leq 0$$

Ý nghĩa: Nếu giá trị hàm tại y nhỏ hơn tại x, thì hướng từ x đến y phải tạo góc tù với gradient của f tại x.

Bất đẳng thức Descent

Mệnh đề 3 (Dennis và Schnabel 1983 [5])

Giả sử ∇f là L-Lipschitz liên tục trên C. Với mọi $x, y \in C$, ta có:

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

Ý nghĩa: Bất đẳng thức này chặn trên sai số giữa hàm số $f(y)$ và xấp xỉ tuyến tính của nó tại x.

Bổ đề chứng minh hội tụ

Bổ đề 1 (Xu 2002 [19])

Cho các dãy số thực $\{a_k\}, \{b_k\} \subset (0, \infty)$ thỏa mãn điều kiện:

$$a_{k+1} \leq a_k + b_k, \quad \forall k \geq 0$$

và chuỗi hội tụ: $\sum_{k=0}^{\infty} b_k < \infty$
Khi đó, tồn tại giới hạn hữu hạn: $\lim_{k \to \infty} a_k = c \in \mathbb{R}$

Mục lục

Thuật toán Gradient Descent

Thuật toán Gradient Descent (GD)

Bước 0. Chọn điểm khởi tạo $x^0 \in C$, tham số bước $\lambda \in (0, 2/L)$. Đặt $k = 0$.

Bước 1. Với x^k đã cho, tính

$$x^{k+1} = P_C\big(x^k - \lambda \nabla f(x^k)\big).$$

Bước 2.

- Nếu $x^{k+1} = x^k$ thì **dừng**.
- Ngược lại, đặt $k := k + 1$, quay lại **Bước 1**.

Thuật toán Gradient Descent (GD)

- x^k : nghiệm xấp xỉ tại vòng lặp k.
- $-\nabla f(x^k)$: hướng dốc nhất làm giảm hàm mục tiêu.
- λ : cỡ bước (step-size), cố định trong khoảng (0,2/L).
- P_C : phép chiếu lên tập ràng buộc C:
- Nếu $C = \mathbb{R}^m$ thì P_C là ánh xạ đồng nhất $\Rightarrow$ GD không ràng buộc.

Ý tưởng chính: đi theo hướng $-\nabla f(x^k)$, sau đó kéo nghiệm trở lại miền khả thi C.

Thuật toán Gradient Descent Adaptive (GDA) [17]

Thuật toán GDA

Bước 0. Chọn $x^0 \in C$, $\lambda_0 > 0$, hệ số $\sigma, \kappa \in (0,1)$. Đặt $k := 0$.

Bước 1. Với x^k, λ_k đã cho, tính

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

Nếu $f(x^{k+1}) \leq f(x^k) - \sigma\langle \nabla f(x^k), x^k - x^{k+1}\rangle$ thì $\lambda_{k+1} := \lambda_k$, ngược lại $\lambda_{k+1} := \kappa\lambda_k$.

Bước 2.

- Nếu $x^{k+1} = x^k$ thì **dừng**.
- Ngược lại, đặt $k := k+1$, quay lại **Bước 1**.

- Ý tưởng: giảm dần cỡ bước một cách ổn định cho đến khi thỏa mãn điều kiện, không cần chi phí tính toán lớn như phương pháp tìm kiếm theo tia hay phải biết trước hằng số Lipschitz.

Nhận xét

Chú ý 1

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

Thật vậy, vì $x^{k+1} = P_C(x^k - \lambda_k \nabla f(x^k))$, áp dụng Mệnh đề 1-(2), ta có:

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

Nếu $x^{k+1} = x^k$, ta thu được:

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

điều này có nghĩa là x^k là điểm dừng của bài toán. Hơn nữa, nếu f là hàm giả lồi, từ (2) suy ra $f(z) \geq f(x^k)$ với mọi $z \in C$, hay x^k là nghiệm của $\mathrm{OP}(f, C)$.

Định lý hội tụ của GDA

Định lý

Giả sử dãy $\{x^k\}$ được sinh bởi thuật toán GDA. Khi đó:

- Dãy $\{f(x^k)\}$ hội tụ và mọi điểm giới hạn (nếu có) của $\{x^k\}$ là 1 điểm dừng của bài toán.
- Nếu f **tựa lồi** trên C thì $\{x^k\}$ hội tụ tới 1 điểm dừng của bài toán.
- Nếu f **giả lồi** trên C thì $\{x^k\}$ hội tụ tới 1 nghiệm của bài toán.

Chứng minh định lý

Dãy $\{f(x^k)\}$ hội tụ và mọi điểm giới hạn (nếu có) của $\{x^k\}$ là 1 điểm dừng của bài toán.

Chứng minh. Áp dụng Mệnh đề 3, ta có:

$$f(x^{k+1}) \leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2. \quad (3)$$

Trong (1), chọn $z = x^k \in C$, ta thu được:

$$\langle \nabla f(x^k), x^{k+1} - x^k \rangle \leq -\frac{1}{\lambda_k}\|x^{k+1} - x^k\|^2. \quad (4)$$

Kết hợp (3) và (4), ta thu được:

$$f(x^{k+1}) \leq f(x^k) - \sigma\langle \nabla f(x^k), x^k - x^{k+1} \rangle - \left(\frac{1-\sigma}{\lambda_k} - \frac{L}{2}\right)\|x^{k+1} - x^k\|^2. \quad (5)$$

Chứng minh định lý

Ta sẽ chứng minh rằng dãy $\{\lambda_k\}$ bị chặn dưới bởi một số dương, hay nói cách khác, độ dài bước nhảy chỉ thay đổi hữu hạn lần.
Thật vậy, giả sử phản chứng rằng $\lambda_k \to 0$. Từ (5), tồn tại $k_0 \in \mathbb{N}$ thỏa mãn:

$$f(x^{k+1}) \leq f(x^k) - \sigma\langle \nabla f(x^k), x^k - x^{k+1}\rangle \quad \forall k \geq k_0. \tag{6}$$

Theo cách xây dựng của λ_k, bất đẳng thức trên kéo theo rằng $\lambda_k = \lambda_{k_0}$ với mọi $k \geq k_0$. Điều này là mâu thuẫn.
Do đó, tồn tại $k_1 \in \mathbb{N}$ sao cho với mọi $k \geq k_1$, ta có $\lambda_k = \lambda_{k_1}$ và:

$$f(x^{k+1}) \leq f(x^k) - \sigma\langle \nabla f(x^k), x^k - x^{k+1}\rangle. \tag{7}$$

Lưu ý rằng $\langle \nabla f(x^k), x^k - x^{k+1}\rangle \geq 0$, ta suy ra dãy $\{f(x^k)\}$ hội tụ và:

$$\sum_{k=0}^{\infty}\langle \nabla f(x^k), x^k - x^{k+1}\rangle < \infty; \quad \sum_{k=0}^{\infty}\|x^{k+1} - x^k\|^2 < \infty \tag{8}$$

Chứng minh định lý

Từ (1), với mọi $z \in C$, ta có

$$\begin{aligned} \|x^{k+1} - z\|^2 &= \|x^k - z\|^2 - \|x^{k+1} - x^k\|^2 + 2\langle x^{k+1} - x^k, x^{k+1} - z\rangle \\ &\leq \|x^k - z\|^2 - \|x^{k+1} - x^k\|^2 + 2\lambda_k \langle \nabla f(x^k), z - x^{k+1}\rangle. \end{aligned} \tag{9}$$

Giả sử $\bar{x}$ là một điểm giới hạn của dãy $\{x^k\}$. Tồn tại một dãy con $\{x^{k_i}\} \subset \{x^k\}$ sao cho $\lim_{i\to\infty} x^{k_i} = \bar{x}$. Với bất đẳng thức trên cho $k = k_i$ và lấy giới hạn khi $i \to \infty$. Chú ý rằng $\|x^k - x^{k+1}\| \to 0$, và ∇f là hàm liên tục, ta nhận được

$$\langle \nabla f(\bar{x}), z - \bar{x}\rangle \geq 0 \quad \forall z \in C, \tag{10}$$

Do đó $\bar{x}$ là điểm dừng.

Chứng minh định lý

Nếu f **tựa lồi** trên C thì $\{x^k\}$ hội tụ tới 1 điểm dừng của bài toán.

Đặt

$$U := \left\{ x \in C : f(x) \leq f(x^k) \quad \forall k \geq 0 \right\}.$$

Vì U chứa tập nghiệm của $OP(f, C)$, và do đó, U không rỗng. Lấy $\hat{x} \in U$. Vì $f(x^k) \geq f(\hat{x})$ với mọi $k \geq 0$ và f là hàm tựa lồi nên

$$\langle \nabla f(x^k), \hat{x} - x^k \rangle \leq 0 \quad \forall k \geq 0. \tag{11}$$

Kết hợp (9) và (11), ta có

$$\|x^{k+1} - \hat{x}\|^2 \leq \|x^k - \hat{x}\|^2 - \|x^{k+1} - x^k\|^2 + 2\lambda_k \langle \nabla f(x^k), x^k - x^{k+1} \rangle. \tag{12}$$

Áp dụng Bổ đề 1 với $a_k = \|x^k - \hat{x}\|^2$, $b_k = 2\lambda_k \langle \nabla f(x^k), x^k - x^{k+1} \rangle$, ta suy ra rằng dãy $\{\|x^k - \hat{x}\|\}$ hội tụ với mọi $\hat{x} \in U$.

Chứng minh định lý

Lại có dãy $\{x^k\}$ bị chặn, tồn tại một dãy con $\{x^{k_i}\} \subset \{x^k\}$ sao cho

$$\lim_{i \to \infty} x^{k_i} = \bar{x} \in C$$

Ta có dãy $\{f(x^k)\}$ là dãy không tăng và hội tụ. Nên $\lim_{k \to \infty} f(x^k) = f(\bar{x})$ và $f(\bar{x}) \leq f(x^k)$ với mọi $k \geq 0$. Do đó $\bar{x} \in U$ và dãy $\{\|x^k - \bar{x}\|\}$ hội tụ. Từ đó suy ra,

$$\lim_{k \to \infty} \|x^k - \bar{x}\| = \lim_{i \to \infty} \|x^{k_i} - \bar{x}\| = 0.$$

Ta đã có mỗi điểm tụ của $\{x^k\}$ là một điểm dừng của bài toán. Khi đó, toàn bộ dãy $\{x^k\}$ hội tụ đến $\bar{x}$ – một điểm dừng của bài toán.

Chứng minh định lý

Nếu f **giả lồi** trên C thì $\{x^k\}$ hội tụ tới 1 nghiệm của bài toán.

Từ (10) có: $\langle \nabla f(\bar{x}), z - \bar{x} \rangle \geq 0 \quad \forall z \in C$
Do f là giả lồi nên $f(\bar{x}) \leq f(z) \quad \forall z \in C$
Do đó $\{x^k\}$ hội tụ tới 1 nghiệm của bài toán.

Thuật toán Gradient Descent Adaptive (GDA)

Chú ý 2

Với mọi $\lambda \in (0, 2/L)$ luôn tồn tại $\sigma \in (0, 1)$ sao cho $\lambda \leq 2(1-\sigma)/L$.
Khi đó chọn $\lambda_0 \in (0, 2(1-\sigma)/L)$ thì $\lambda_k = \lambda_0$ với mọi k.

- Như vậy GD là 1 trường hợp đặc biệt của GDA.
- Nếu L biết trước có thể chọn λ cố định thì theo định lý trên thuật toán vẫn hội tụ.
- Mọi khẳng định của định lý trên vẫn đúng với dãy $\{x^k\}$ được sinh ra bởi Thuật toán GD.

Sự hội tụ của GDA

Bây giờ, chúng ta sẽ ước lượng tốc độ hội tụ của Thuật toán GDA khi giải các bài toán tối ưu không ràng buộc.

Hệ quả 1

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

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

trong đó x^* là một nghiệm của bài toán.

Chứng minh: Gọi x^* là một nghiệm của bài toán. Đặt $\Delta_k := f(x^k) - f(x^*)$. Từ (6), lưu ý rằng $x^k - x^{k+1} = \lambda_k \nabla f(x^k)$, ta có:

$$\Delta_{k+1} \leq \Delta_k - \sigma \lambda_{k_1} \|\nabla f(x^k)\|^2 \quad \forall k \geq k_1. \tag{13}$$

Sự hội tụ của GDA

Mặt khác, vì dãy $\{x^k\}$ bị chặn và f là hàm lồi, ta có:

$$\begin{aligned} 0 \leq \Delta_k &\leq \langle \nabla f(x^k), x^k - x^* \rangle \\ &\leq M \|\nabla f(x^k)\|, \end{aligned} \tag{14}$$

trong đó $M := \sup\left\{\|x^k - x^*\| : k \geq k_1\right\} < \infty$. Từ (13) và (14), ta thu được:

$$\Delta_{k+1} \leq \Delta_k - Q\Delta_k^2 \quad \forall k \geq k_1 \tag{15}$$

với $Q := \frac{\sigma \lambda_{k_1}}{M^2}$. Lưu ý rằng $\Delta_{k+1} \leq \Delta_k$, từ (15), ta có:

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

điều này suy ra

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

Stochastic Gradient Descent Adaptive Algorithm-SGDA

Thuật toán GD Thích nghi Ngẫu nhiên (SGDA)

Bước 0. Chọn $x^0 \in C$, $\lambda_0 \in (0, +\infty)$, $\sigma, \kappa \in (0, 1)$. Đặt $k = 0$.

Bước 1. Lấy mẫu ξ^k và tính x^{k+1} cùng λ_{k+1}:

$$x^{k+1} = P_C(x^k - \lambda_k \nabla f_{\xi^k}(x^k))$$

Nếu $f_{\xi^k}(x^{k+1}) \leq f_{\xi^k}(x^k) - \sigma\langle \nabla f_{\xi^k}(x^k), x^k - x^{k+1}\rangle$
thì $\lambda_{k+1} := \lambda_k$; **ngược lại** $\lambda_{k+1} := \kappa\lambda_k$.

Bước 2.

- Nếu $x^{k+1} = x^k$ thì **dừng**.
- Ngược lại, đặt $k := k + 1$, quay lại **Bước 1**.

Mục lục

Thực nghiệm toán học

Ví dụ 1 (Ví dụ 5.2, Liu et. al., 2022) [11]

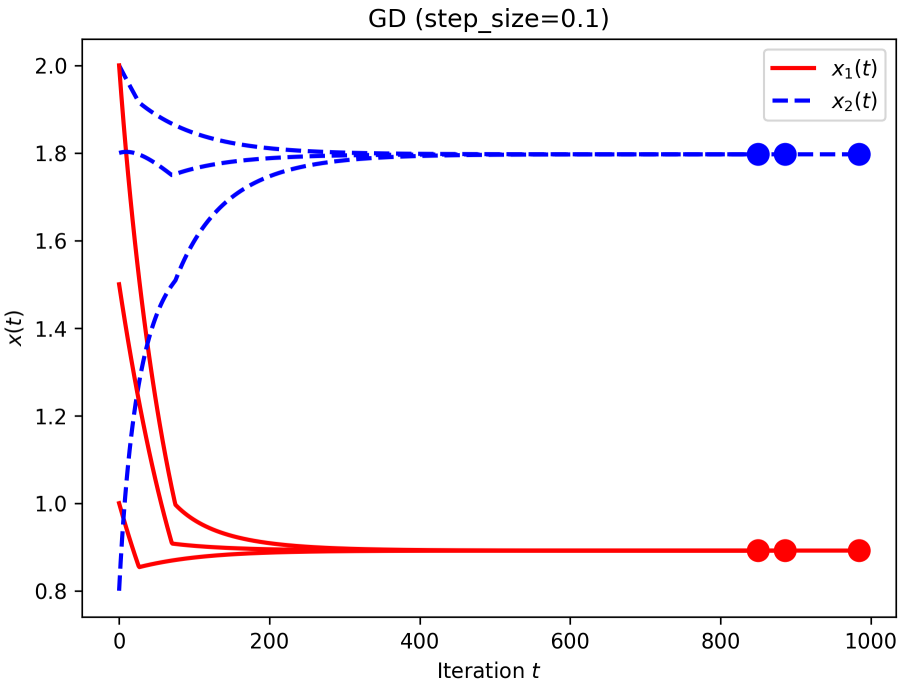
Xét bài toán (P):

$$\min f(x) = \frac{x_1^2 + x_2^2 + 3}{1 + 2x_1 + 8x_2}$$
$$\text{v.đ.k } x \in C,$$

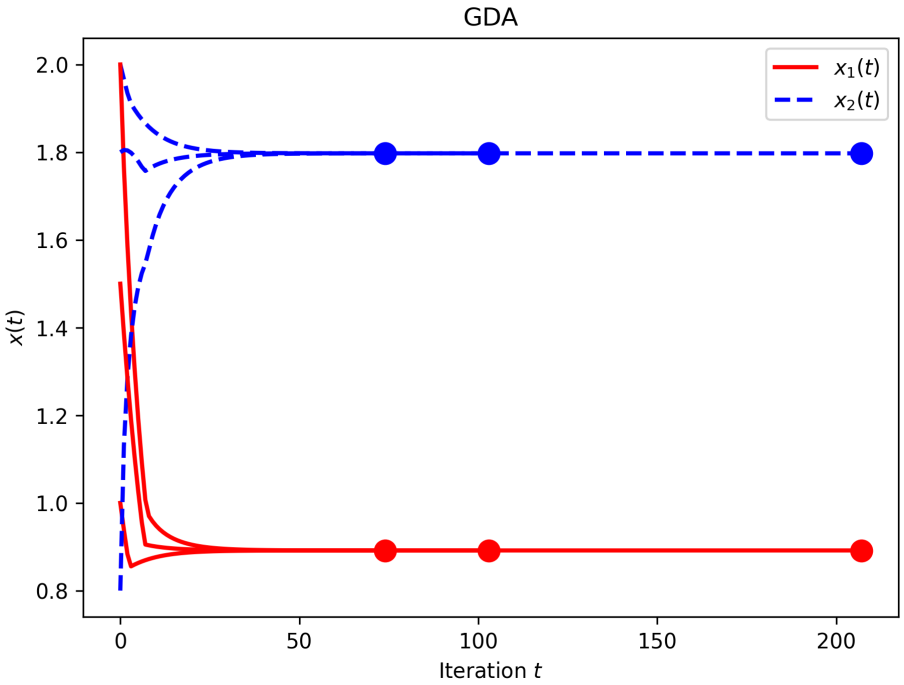
với $C = \{ x = (x_1, x_2)^\top \in \mathbb{R}^2 \mid g_1(x) = -x_1^2 - 2x_1x_2 \leq -4,\ x_1, x_2 \geq 0\}$.
Hàm mục tiêu f là giả lồi trên tập lồi C.

Thuật toán	**f(x*)**	**Iter**
GDA	0.40935	130
GD	0.40935	910
RNN [11]	0.4101	-

Thực nghiệm toán học



(a) GD



(b) GDA

Hình: Kết quả chạy thực nghiệm *ví dụ 1*

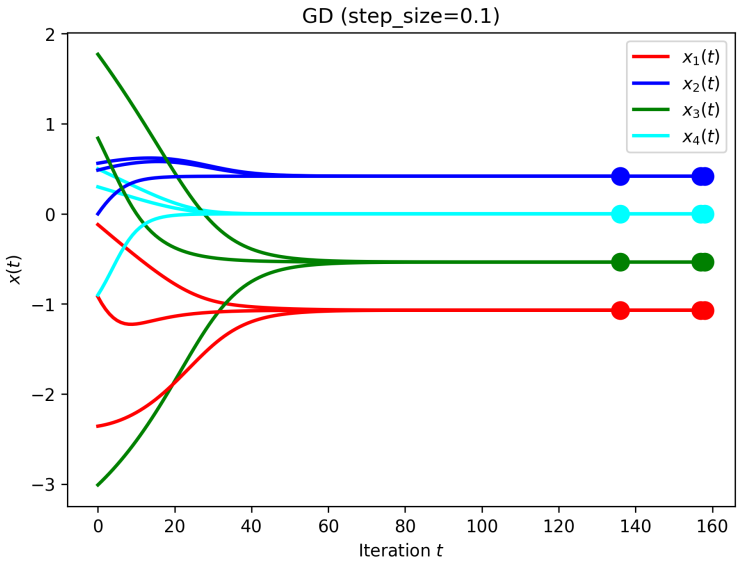
Ví dụ 2 (Ví dụ 5.1, Liu et. al., 2022 [11])

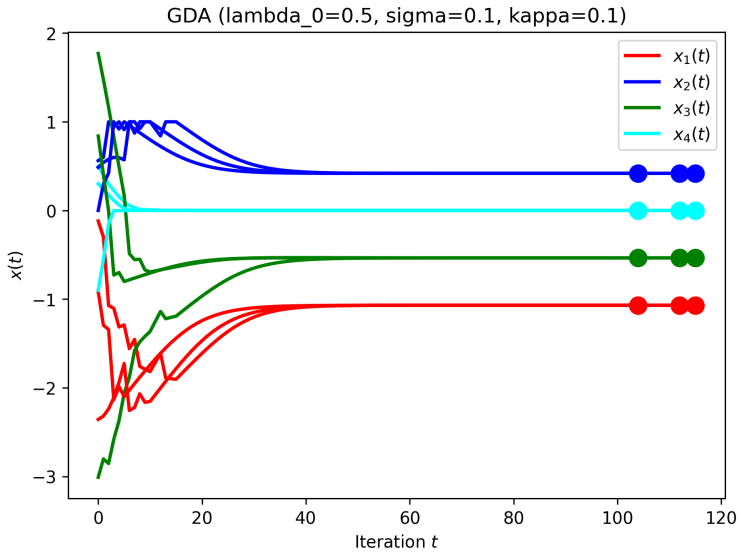
Xét bài toán tối ưu giả lồi không trơn sau đây:

$$\begin{aligned} \min \quad & f(x) = \frac{e^{|x_2-3|} - 30}{x_1^2 + x_3^2 + 2x_4^2 + 4}, \\ \text{v.đ.k} \quad & g_1(x) = (x_1 + x_3)^3 + 2x_4^2 \leq 10, \\ & g_2(x) = (x_2 - 1)^2 \leq 1, \\ & 2x_1 + 4x_2 + x_3 = -1, \end{aligned}$$

trong đó $x = (x_1, x_2, x_3, x_4)^\top \in \mathbb{R}^4$. Hàm mục tiêu f(x) là hàm giả lồi không trơn trên miền C.

Thực nghiệm toán học





Hình: Kết quả chạy thực nghiệm *ví dụ 2*

Thuật toán	**f(x*)**	**Iter**
GDA	-3.0908	110
GD	-3.0908	150
RNN [11]	-3.0849	-

Ví dụ 3 (Ví dụ 4.5, Ferreira et. al., 2022 [6])

Cho $e := (1, \ldots, n) \in \mathbb{R}^n$ là một vector, $\alpha > 0$ và $\beta > 0$ là các hằng số thỏa mãn $2\alpha > 3\beta^{3/2}\sqrt{n}$. Xét bài toán (P) với hàm liên quan:

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

với $a \in \mathbb{R}^n_{++}$ và tập ràng buộc không lồi $C := \{x \in \mathbb{R}^n_{++} : 1 \leq x_1 \cdots x_n\}$

Kết quả chạy thực nghiệm *ví dụ 3* với $\beta = 0.741271, \alpha = 4\beta^{3/2}\sqrt{n+1}$.

Thực nghiệm toán học

Results with Adaptive Step Size (gda_multiplier=5.0)						
n	Algorithm GDA (proposed)			Algorithm GD		
	f_opt	time (ms)	iters	f_opt	time (ms)	iters
10	80.080567	5.187273	18	80.080567	5.340099	19
20	219.590296	4.446507	17	219.590296	4.818201	19
50	852.496543	5.086899	17	852.496543	3.758669	19
100	2394.338032	2.675772	17	2394.338032	9.163141	20
200	6732.212169	8.964062	17	6732.212169	8.924246	20
500	26429.055560	11.014938	17	26429.055560	3.299236	20
1000	74531.265821	4.306316	17	74531.265821	8.611441	20
2000	210371.422765	6.401777	17	210371.422765	7.229328	20
3000	386136.150348	7.884979	17	386136.150348	7.285357	20
10000	2345979.128179	20.025969	17	2345979.128180	19.420147	20

Hình: GDA với $\lambda_0 = 5/L$.

Thực nghiệm toán học

Results with Adaptive Step Size (gda_multiplier=2.0)						
n	Algorithm GDA (proposed)			Algorithm GD		
	f_opt	time (ms)	iters	f_opt	time (ms)	iters
10	80.080567	5.170822	8	80.080567	10.428429	19
20	219.590296	3.693819	8	219.590296	5.551577	19
50	852.496543	5.488157	8	852.496543	12.067556	19
100	2394.338032	2.471209	8	2394.338032	9.481430	20
200	6732.212169	0.000000	8	6732.212169	4.953146	20
500	26429.055560	5.886316	9	26429.055560	4.676104	20
1000	74531.265821	2.899885	9	74531.265821	5.962849	20
2000	210371.422765	1.003504	9	210371.422765	0.000000	20
3000	386136.150348	12.046814	9	386136.150348	8.279085	20
10000	2345979.128179	8.660316	9	2345979.128180	19.539595	20

Hình: GDA với $\lambda_0 = 2/L$.

Thực nghiệm toán học

Results with Fixed Step Size (gda_multiplier=5.0, step_size=0.1)						
n	Algorithm GDA (proposed)			Algorithm GD		
	f_opt	time (ms)	iters	f_opt	time (ms)	iters
10	80.080567	15.069962	46	80.080567	0.000000	8
20	219.590296	17.896414	58	219.590296	2.617359	13
50	852.496543	21.991491	90	852.496543	5.728245	21
100	2394.338032	30.868053	111	2394.338032	8.029699	29
200	6732.212169	27.752638	100	6732.212169	11.726856	41
500	26429.055560	40.580988	130	26429.055560	20.916939	66
1000	74531.265821	44.952154	112	74531.265821	34.481049	92
2000	210371.422765	61.488390	138	210371.422765	57.462692	129
3000	386136.150348	50.976515	92	386136.150348	94.659328	161
10000	2345979.128179	137.495756	101	2345979.128179	426.794291	293

Hình: GDA với $\lambda_0 = 0.5$.

Thực nghiệm toán học

Results with Fixed Step Size (gda_multiplier=2.0, step_size=0.1)							
n	Algorithm GDA (proposed)			Algorithm GD			
	f_opt	time (ms)	iters	f_opt	time (ms)	iters	
10	80.080567	8.711338	19	80.080567	0.000000	8	
20	219.590296	8.127928	26	219.590296	3.469229	13	
50	852.496543	11.977434	41	852.496543	6.461382	21	
100	2394.338032	9.682655	55	2394.338032	11.731386	29	
200	6732.212169	19.770622	78	6732.212169	10.179520	41	
500	26429.055560	31.902075	109	26429.055560	22.556067	66	
1000	74531.265821	32.226086	103	74531.265821	36.172628	92	
2000	210371.422765	51.384926	114	210371.422765	65.045834	129	
3000	386136.150348	84.401369	130	386136.150348	95.977306	161	
10000	2345979.128179	172.491789	125	2345979.128179	400.845051	293	

Hình: GDA với $\lambda_0 = 0.2$.

Ứng dụng trong học máy

Phương pháp đề xuất được đánh giá qua 03 bài toán phổ biến trong Machine Learning để chứng minh hiệu quả tính toán và độ chính xác:

1. **Lựa chọn đặc trưng có giám sát (Supervised feature selection):**
2. **Hồi quy Logistic đa biến (Multi-variable logistic regression):**
3. **Phân loại ảnh với Mạng nơ-ron (Neural networks for classification):**

Các bài toán trên được mô hình hóa dưới dạng các bài toán tối ưu.

Lựa chọn đặc trưng có giám sát [18]

- **Mô hình:** Tối thiểu hóa hàm phân thức giả lồi trên tập lồi

$$\text{minimize} \quad \frac{w^T Q w}{\rho^T w}$$
$$\text{với điều kiện} \quad e^T w = 1; w \geq 0,$$

  - w: Vector trọng số đặc trưng cần tìm (Feature score).
  - Q: Ma trận dư thừa thông tin.
  - ρ: Vector mức độ liên quan đến nhãn.
  - e: Vector đơn vị $(1, \ldots, 1)^T$, ràng buộc tổng trọng số $= 1$.
- **Dữ liệu:** Tập dữ liệu Parkinsons (23 đặc trưng, 195 mẫu).
- **So sánh:** Thuật toán GDA đề xuất với Thuật toán mạng nơ-ron động (RNN).

Lựa chọn đặc trưng có giám sát[18]

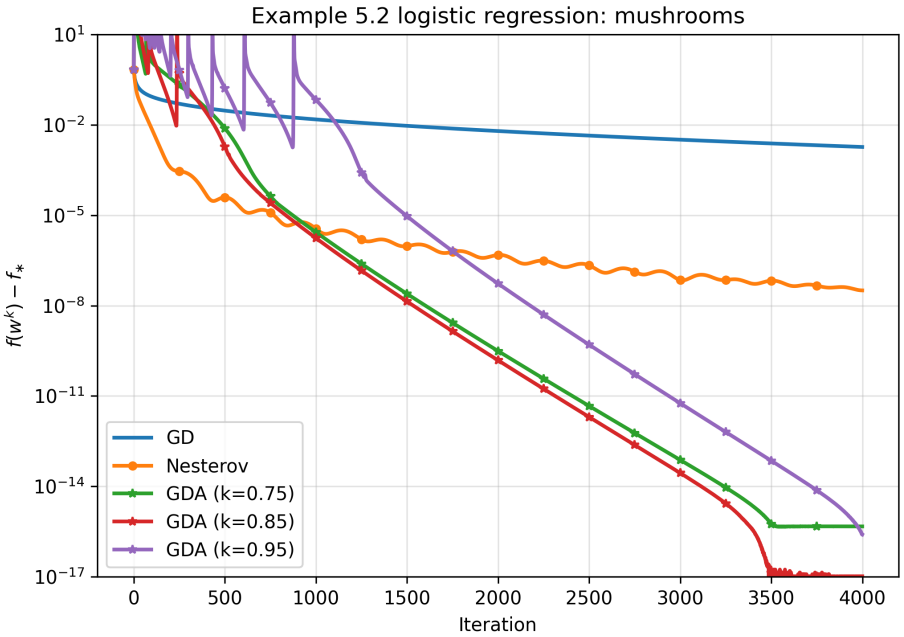
Comparing Scipy vs GDA						
seed	Algorithm GDA (proposed)			Scipy		
	f_opt	time (ms)	iters	f_opt	time (ms)	iters
1	0.152478	0.018614	32	0.152478	0.006164	19
11	0.153480	0.096673	51	0.153480	0.030229	15
21	0.153834	0.015980	31	0.153834	0.006214	18
31	0.153501	0.029897	55	0.153501	0.005530	16
41	0.154281	0.026025	50	0.154281	0.004012	13
51	0.154185	0.031173	60	0.154185	0.005908	15
61	0.154878	0.034290	57	0.154878	0.006208	16
71	0.153465	0.021693	33	0.153465	0.005638	16
81	0.153517	0.031792	59	0.153517	0.003538	13
91	0.153269	0.030206	52	0.153269	0.005027	15

Hình: Time qua các vòng lặp.

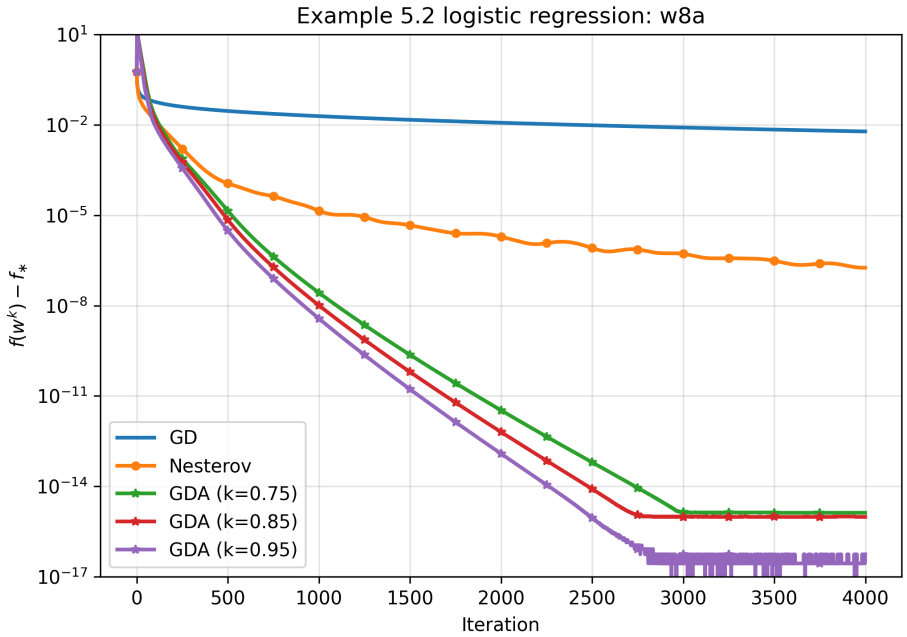
Hồi quy Logistic đa biến [12]

- **Bài toán:** Tối ưu hàm mất mát Cross-entropy có điều chuẩn L_2 (Convex programming).
- **Dữ liệu:** Mushrooms và W8a.
- **So sánh:**
  - Gradient Descent (GD) truyền thống.
  - Phương pháp tăng tốc Nesterov.
- **Nhận xét:**
  - GDA cho giá trị hàm mục tiêu tốt hơn qua các vòng lặp so với GD và Nesterov trên cả 2 tập dữ liệu.
  - Cỡ bước tự thích nghi giảm dần theo thời gian giúp hội tụ ổn định.

Hồi quy Logistic đa biến[12]



(a) Mushrooms



(b) W8a

Hình: Kết quả chạy thực nghiệm *Hồi quy Logistic đa biến.*

Phân loại ảnh với Mạng nơ-ron

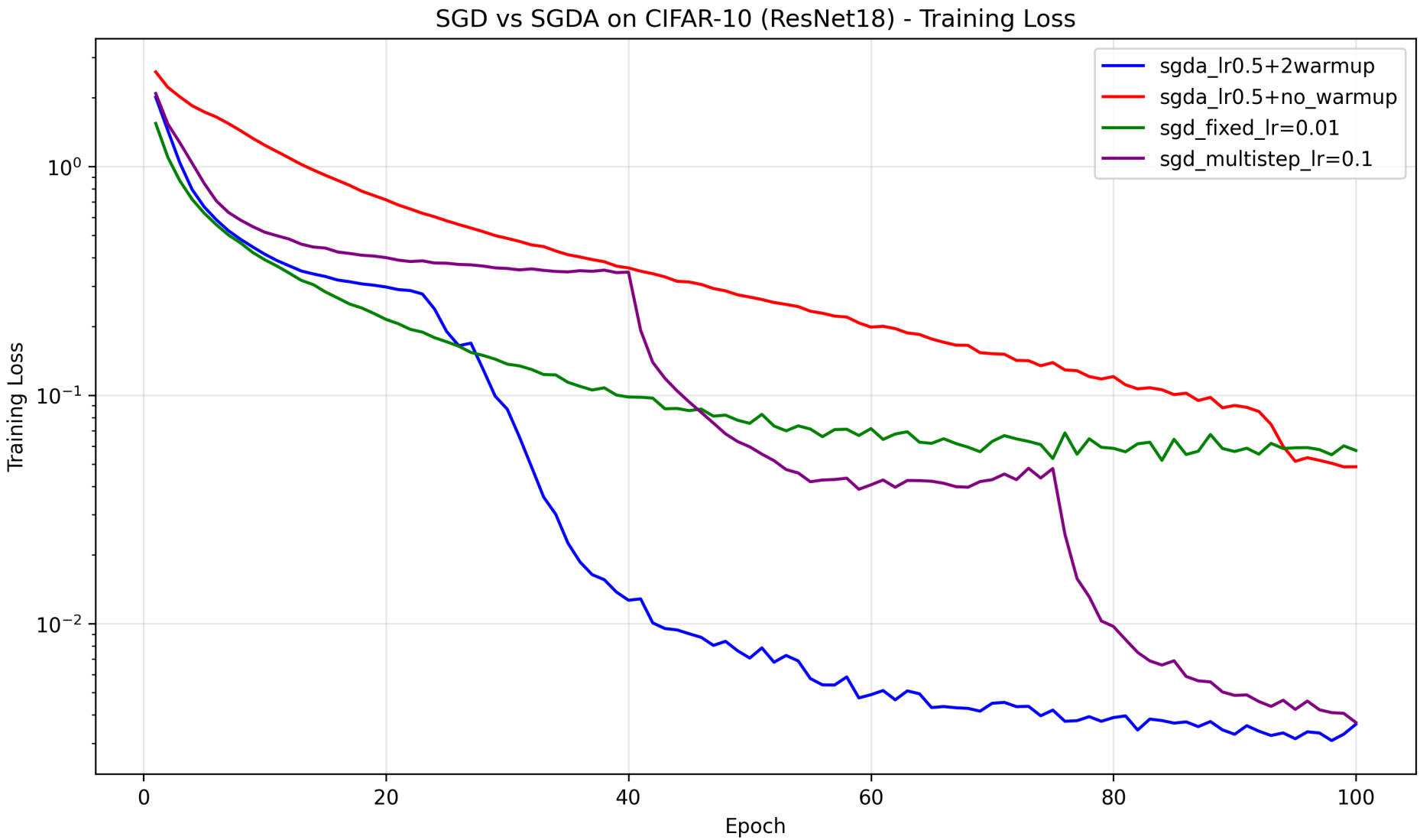
Thiết lập thực nghiệm

- **Mô hình:** Kiến trúc ResNet-18.
- **Dữ liệu:** CIFAR-10 (Dữ liệu ảnh).
- **Thuật toán:** Sử dụng biến thể ngẫu nhiên **SGDA** (do hàm mục tiêu không lồi/không giả lồi).

Nhận xét khi so sánh với SGD

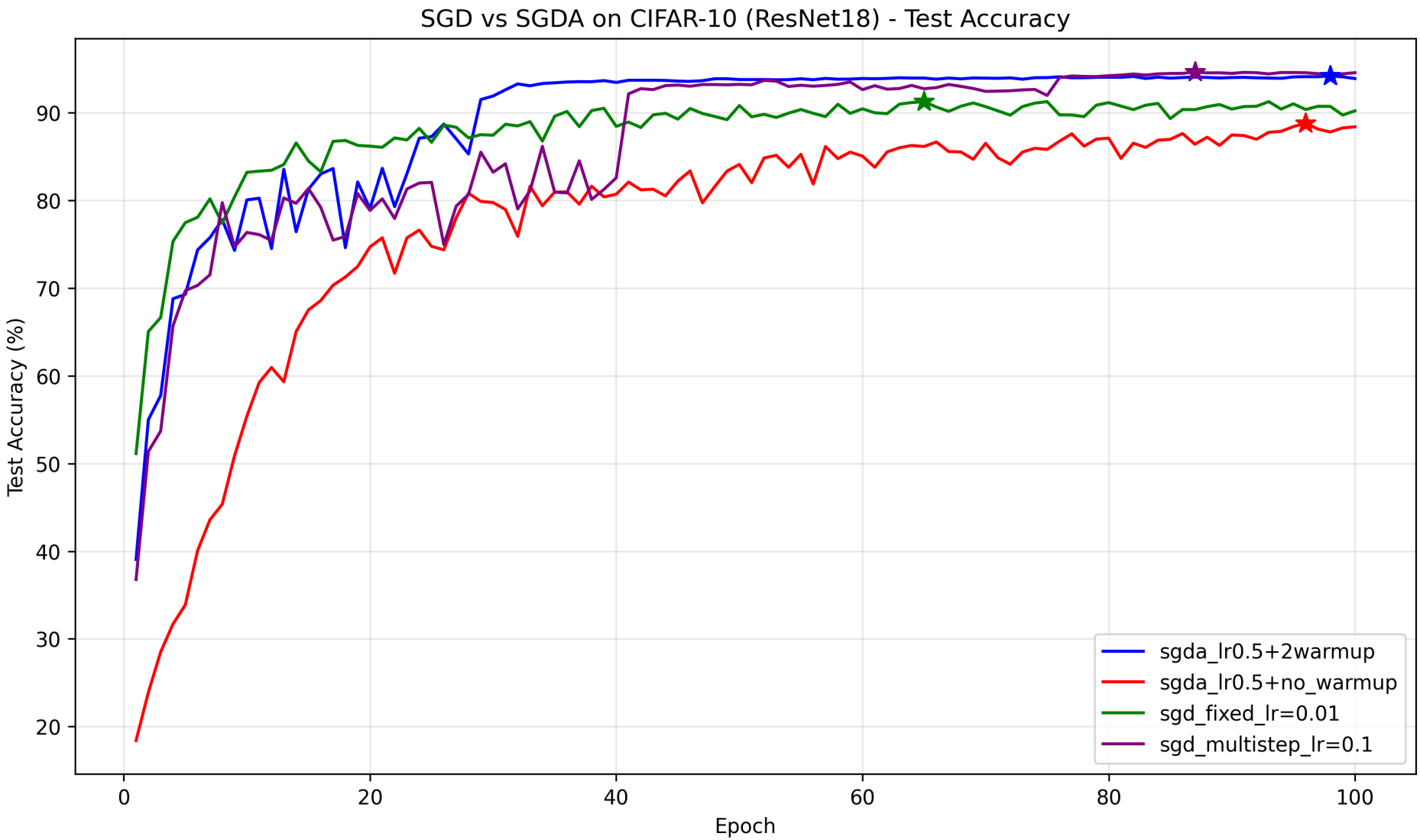
- **Test Accuracy:** SGDA đạt độ chính xác kiểm tra cao hơn qua các vòng lặp.
- **Training Loss:** SGDA giảm hàm mất mát huấn luyện tốt hơn so với SGD.
- *Lưu ý:* Dù không đảm bảo hội tụ lý thuyết mạnh như trường hợp lồi, SGDA vẫn hoạt động hiệu quả như một kỹ thuật heuristic.

Phân loại ảnh với mạng nơ-ron



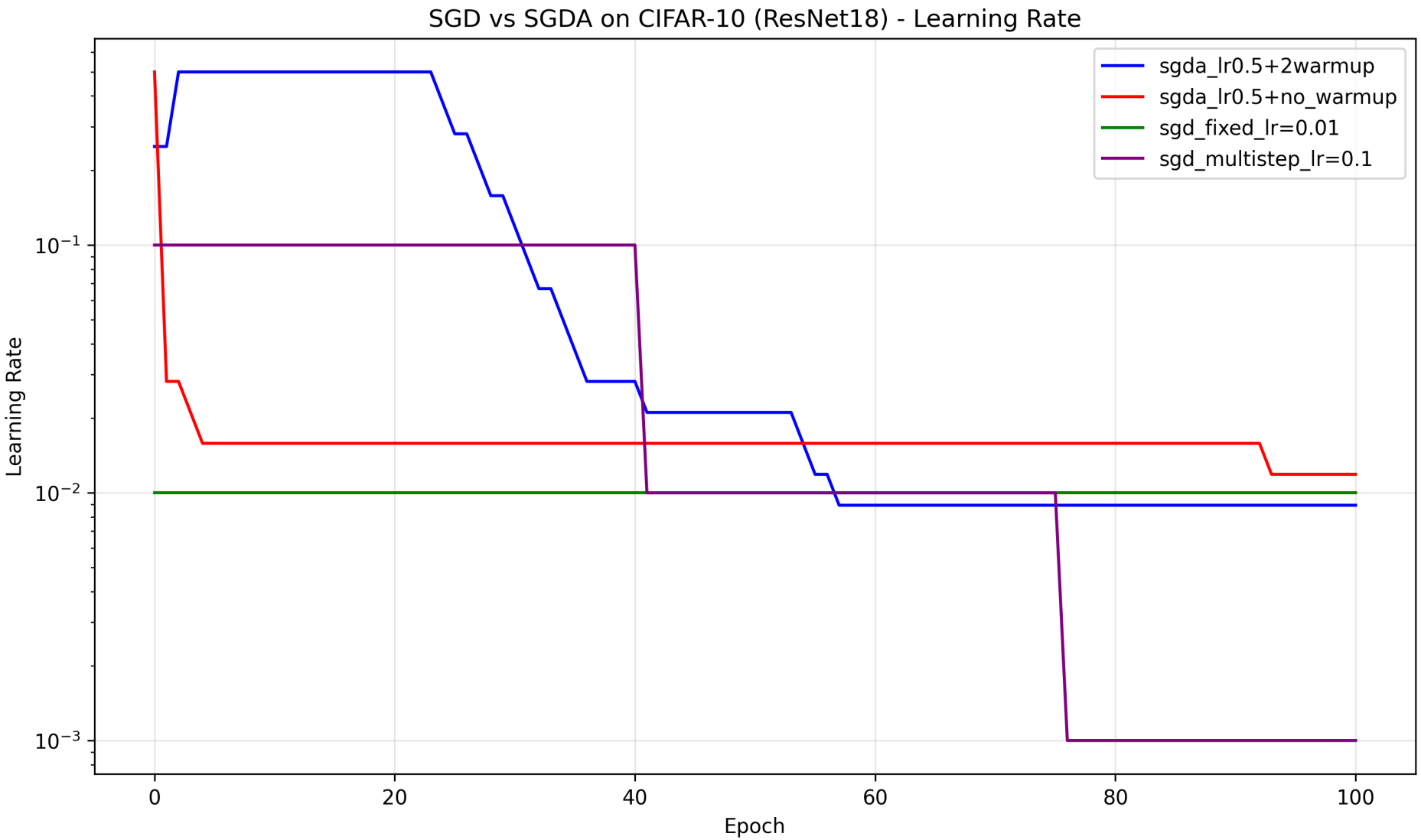
Hình: Training loss qua các vòng lặp.

Phân loại ảnh với mạng nơ-ron



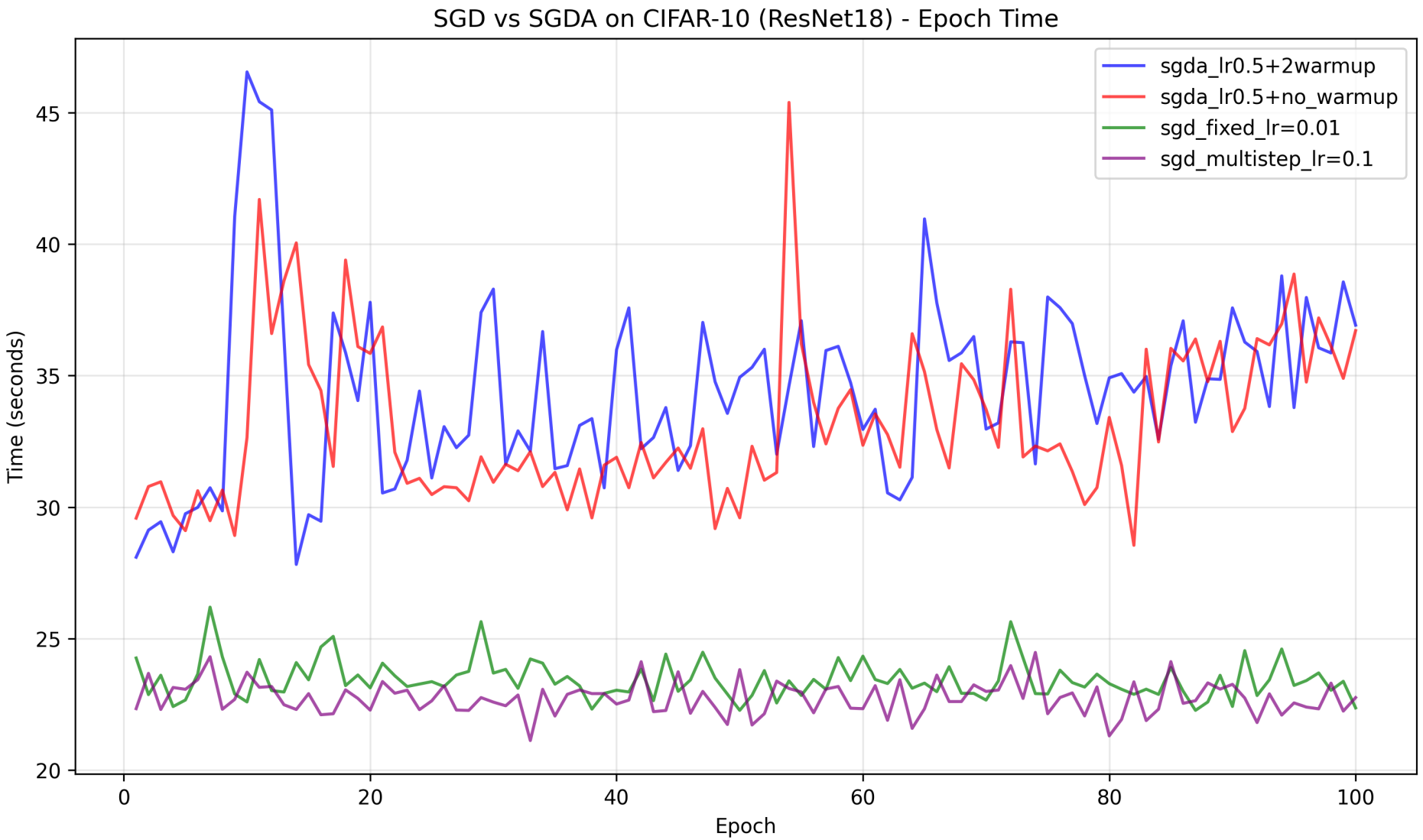
Hình: Test Accuracy qua các vòng lặp.

Phân loại ảnh với mạng nơ-ron



Hình: Learning rate qua các vòng lặp.

Phân loại ảnh với mạng nơ-ron



Hình: Time qua các vòng lặp.

Tài liệu tham khảo I


Bauschke HH, Combettes PL.
Convex Analysis and Monotone Operator Theory in Hilbert Spaces.
Springer, 2011.


Bian W, Ma L, Qin S, Xue X.
Neural network for nonsmooth pseudoconvex optimization with general convex constraints.
Neural Networks, 101:1–14, 2018.


Boyd SP, Vandenberghe L.
Convex Optimization.
Cambridge University Press, Cambridge, 2009.


Cevher V, Becker S, Schmidt M.
Convex optimization for big data.
IEEE Signal Processing Magazine, 31:32–44, 2014.

Tài liệu tham khảo II

Dennis JE, Schnabel RB.
Numerical Methods for Unconstrained Optimization and Nonlinear Equations.
Prentice-Hall, New Jersey, 1983.

Ferreira OP, Sosa WS.
On the frank–wolfe algorithm for non-compact constrained optimization problems.
Optimization, 71(1):197–211, 2022.

Hu Y, Li J, Yu CK.
Convergence rates of subgradient methods for quasiconvex optimization problems.
Computational Optimization and Applications, 77:183–212, 2020.

Tài liệu tham khảo III

Kiwiel KC.

Convergence and efficiency of subgradient methods for quasiconvex minimization.

Mathematical Programming, Series A, 90(1):1–25, 2001.

Konnov IV.

Simplified versions of the conditional gradient method.

Optimization, 67(12):2275–2290, 2018.

Lan GH.

First-order and Stochastic Optimization Methods for Machine Learning.

Springer Series in the Data Sciences. Springer Nature, 2020.

Tài liệu tham khảo IV

Liu N, Wang J, Qin S.
A one-layer recurrent neural network for nonsmooth pseudoconvex optimization with quasiconvex inequality and affine equality constraints.
Neural Networks, 147:1–14, 2022.

Malitsky Y, Mishchenko K.
Adaptive gradient descent without descent.
In *Proceedings of Machine Learning Research*, volume 119, pages 6702–6712, 2020.

Mangasarian O.
Pseudo-convex functions.
SIAM Journal on Control, 8:281–289, 1965.

Tài liệu tham khảo V


Nesterov Y.
Introductory Lectures on Convex Optimization: A Basic Course, volume 87.
Springer Science & Business Media, 2013.


Rockafellar RT.
Convex Analysis.
Princeton University Press, Princeton, 1970.


Thang TN, Solanki VK, Dao TA, Anh NTN, Hai PV.
A monotonic optimization approach for solving strictly quasiconvex multiobjective programming problems.
Journal of Intelligent & Fuzzy Systems, 38:6053–6063, 2020.

Tài liệu tham khảo VI


Trần Ngọc Thăng, Trịnh Ngọc Hải.
Self-adaptive algorithms for quasiconvex programming and applications to machine learning.
Computational and Applied Mathematics, 43(249), 2024.


Wang Y, Li X, Wang J.
A neurodynamic optimization approach to supervised feature selection via fractional programming.
Neural Networks, 136:194–206, 2021.


Xu HK.
Iterative algorithms for nonlinear operators.
Journal of the London Mathematical Society, 66:240–256, 2002.

Tài liệu tham khảo VII


Yu CK, Hu Y, Yang X, Choy SK.
Abstract convergence theorem for quasi-convex optimization problems with applications.
Optimization, 68(7):1289–1304, 2019.

Thông tin liên hệ

Nhóm trưởng: Bùi Tuấn Đạt

- **Email:** Dat.BT2400096@sis.hust.edu.vn
- **Github:** `https://github.com/tuasananh/NMPPTU-Team1-SAAQP`

Xin chân thành cảm ơn mọi người đã lắng nghe bài thuyết trình của nhóm!