Mathematic Fundamentals of
Computer Science
12/2025

MOMENTUM IN OPTIMIZATION

PRESENTED BY:
Trần Thái Học - 2570426
Ngô Duy Bảo - 2570395
Lê Thái Bình - 2570081
Tiêu Trí Kiệt - 2570436

PRESENTED TO:
Dr. Nguyễn An Khương

Agenda

01	Giới thiệu
02	Ứng dụng
03	Bài tập
04	Tổng kết

1 GIỚI THIỆU

1.1. Phát biểu bài toán

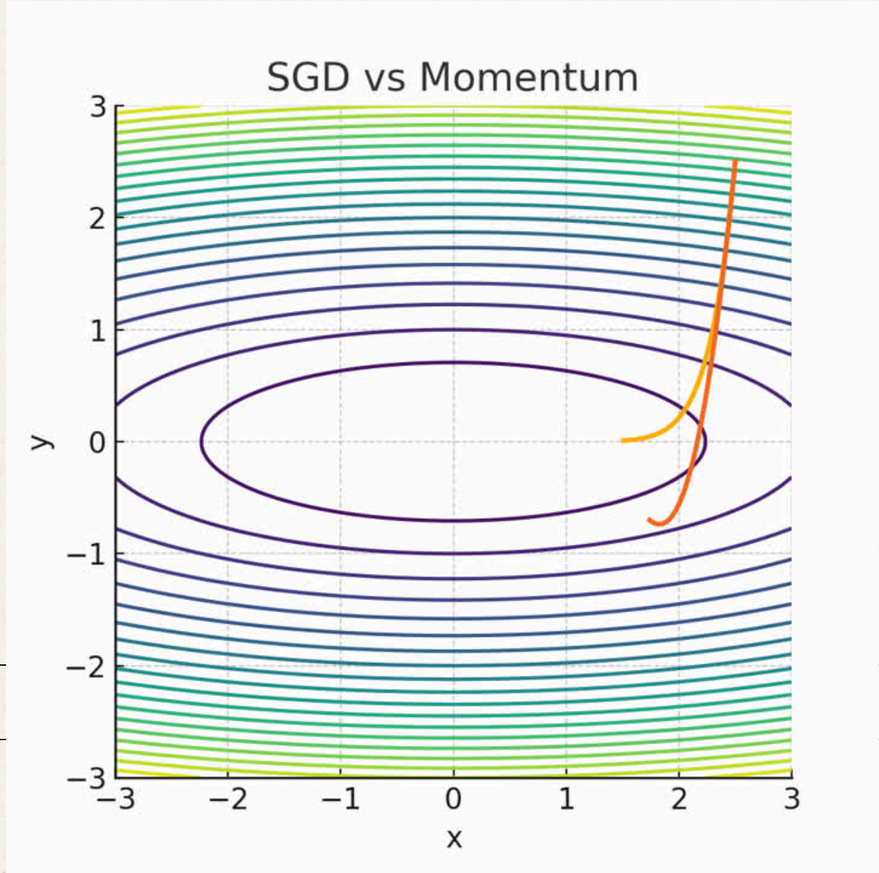
Trong học sâu, ta thường tìm bộ tham số w tối ưu để tối ưu một hàm mất mát: $min_w f(\mathbf{w})$

Gradient Descent cập nhật tham số theo: $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla f(\mathbf{w})$

Tuy nhiên, Gradient Descent thường gặp các vấn đề:

- **Hội tụ chậm**, đặc biệt ở các vùng rãnh dốc hoặc nơi đạo hàm thay đổi mạnh theo từng chiều.
- **Dễ bị dao động** (oscillation) khi gradient ở hướng này rất lớn nhưng ở hướng kia rất nhỏ.
- Khó thoát khỏi các vùng "bằng phẳng" (flat regions).

Ví dụ: $f(x, y) = 0.1x^2 + y^2$



1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

Gradient Descent

$$\mathbf{w} \leftarrow \mathbf{w} - \eta_t \mathbf{g}_t$$

$$\mathbf{g}_t = \partial_{\mathbf{w}} f(\mathbf{x}_t, \mathbf{w})$$

Trong đó:

$\mathbf{w}$	Tham số mô hình
η_t	Hệ số học (learning rate)
$\mathbf{g}_t$	Gradient của hàm loss tại thời điểm t
$\mathbf{x}_t$	Dữ liệu đầu vào
$f(\mathbf{x}_t, \mathbf{w})$	Hàm loss khi dự đoán trên mẫu x_t
$\partial_{\mathbf{w}}$	Đạo hàm riêng theo **w**

Minibatch Stochastic Gradient Descent

$$\mathbf{g}_{t,t-1} = \partial_{\mathbf{w}} \frac{1}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} f(\mathbf{x}_i, \mathbf{w}_{t-1})$$

$$= \frac{1}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} \mathbf{h}_{i,t-1}$$

Trong đó:

$\mathbf{h}_{i,t-1}$	$\partial_{\mathbf{w}} f(\mathbf{x}_i, \mathbf{w}_{t-1})$
$\mathbf{w}_{t-1}$	Tham số bước trước cập nhật
$\lvert\mathcal{B}_t\rvert$	Kích thước batch
$\mathbf{g}_{t,t-1}$	Gradient trung bình của mini-batch

Mini-batch SGD giúp giảm nhiễu trong từng batch, nhưng vẫn còn nhiễu giữa các batch

=> Cần Leaky Average

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

LEAKY AVERAGES

Ý tưởng: Lưu trữ Gradient tích lũy nhiều bước trước

Vận tốc (momentum-like)

$$\mathbf{v}_t = \beta \mathbf{v}_{t-1} + \mathbf{g}_{t,t-1}$$

Hệ số ghi nhớ $0 \leq \beta < 1$

- Beta = 0: chỉ dùng gradient hiện tại
- Beta lớn: dùng nhiều thông tin lịch sử, giảm nhiễu nhưng chậm đổi hướng

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

LEAKY AVERAGES

Ý tưởng: Lưu trữ Gradient tích lũy nhiều bước trước

$$\mathbf{v}_t = \beta \mathbf{v}_{t-1} + \mathbf{g}_{t,t-1}$$

Khai triển 2 bước:

$$\begin{aligned}\mathbf{v}_t &= \beta(\beta \mathbf{v}_{t-2} + \mathbf{g}_{t-1,t-2}) + \mathbf{g}_{t,t-1} \\ &= \beta^2 \mathbf{v}_{t-2} + \beta \mathbf{g}_{t-1,t-2} + \mathbf{g}_{t,t-1}\end{aligned}$$

Khai triển 3 bước:

$$\mathbf{v}_t = \beta^3 \mathbf{v}_{t-3} + \beta^2 \mathbf{g}_{t-2,t-3} + \beta\, \mathbf{g}_{t-1,t-2} + \mathbf{g}_{t,t-1}$$

Khai triển k bước:

$$\mathbf{v}_t = \beta^k \mathbf{v}_{t-k} + \sum_{\tau=0}^{k-1} \beta^\tau \mathbf{g}_{t-\tau,\, t-\tau-1}$$

Khai triển t bước:

$$\mathbf{v}_t = \beta^t \mathbf{v}_0 + \sum_{\tau=0}^{t-1} \beta^\tau \mathbf{g}_{t-\tau,\, t-\tau-1}$$

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

LEAKY AVERAGES

Ý tưởng: Lưu trữ Gradient tích lũy nhiều bước trước

Khai triển t bước:

$$\mathbf{v}_t = \beta^t \mathbf{v}_0 + \sum_{\tau=0}^{t-1} \beta^\tau \mathbf{g}_{t-\tau,\, t-\tau-1}$$

Khởi tạo $\mathbf{v}_0 = 0$

$$\begin{aligned}\mathbf{v}_t &= \sum_{\tau=0}^{t-1} \beta^\tau \mathbf{g}_{t-\tau,\, t-\tau-1} \\ &= \mathbf{g}_{t,t-1} + \beta \mathbf{g}_{t-1,t-2} + \beta^2 \mathbf{g}_{t-2,t-3} + \cdots\end{aligned}$$

Với $0 \leq \beta < 1$

$$1 + \beta + \beta^2 + \cdots = \frac{1}{1-\beta}$$

- Chỉ là tổ hợp tuyến tính giảm dần theo β^τ
- Tổng trọng số > 1 nên không phải trung bình chuẩn hóa.
- Giúp làm mượt gradient, giảm nhiễu giữa các batch.

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

AN ILL-CONDITIONED PROBLEM

Ví dụ, ta có hàm f(x):

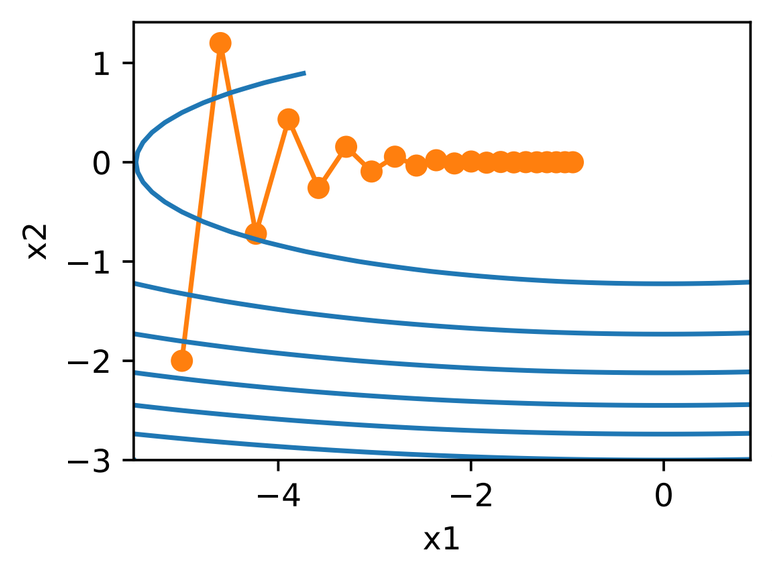
$$f(\mathbf{x}) = 0.1x_1^2 + 2x_2^2$$

Hàm bậc 2 lồi

Có điểm cực tiểu tại (0,0)

$$\nabla f(x) = (0.2x_1, 4x_2)^\top$$

=> gradient theo hướng x_2 nhanh gấp 20 lần hướng x_1



Với learning rate = 0.4, gradient theo hướng x_2 lớn hơn và thay đổi nhanh hơn nhiều so với x_1.

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

AN ILL-CONDITIONED PROBLEM

!?
Gradient Descent gặp khó khăn khi giải quyết bài toán ill-conditioned.

Ví dụ, ta có hàm f(x):

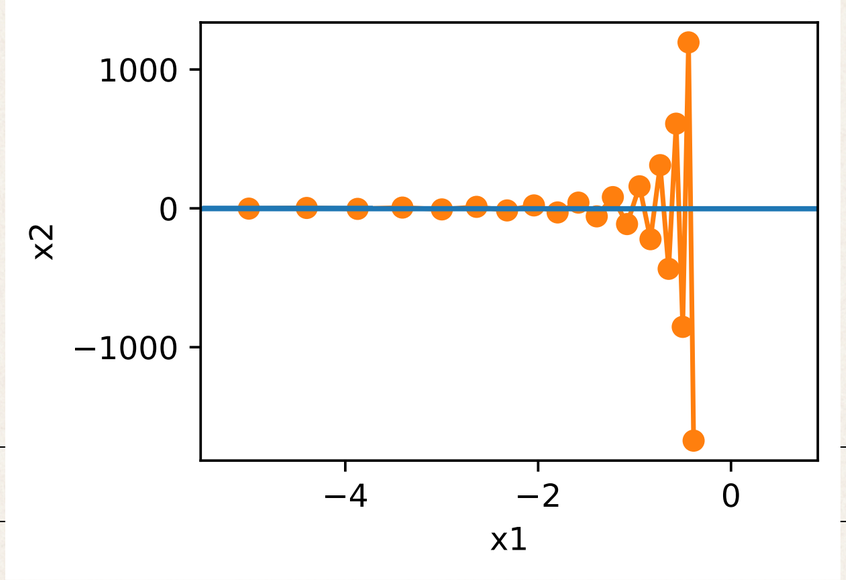
$$f(\mathbf{x}) = 0.1x_1^2 + 2x_2^2$$

Hàm bậc 2 lồi

Có điểm cực tiểu tại (0,0)

$$\nabla f(x) = (0.2x_1, 4x_2)^\top$$

=> gradient theo hướng x_2 nhanh gấp 20 lần hướng x_1



Với learning rate = 0.4, gradient theo hướng x_2 lớn hơn và thay đổi nhanh hơn nhiều so với x_1.

Với learning rate = 0.6, tiến nhanh theo hướng x_1, nhưng lại phân kỳ theo hướng x_2.

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

1.3. Định nghĩa

Momentum được mô tả như việc ta tích lũy Gradient theo thời gian bằng một biến vận tốc (velocity).

$$\begin{aligned} \mathbf{v}_t &\leftarrow \beta \mathbf{v}_{t-1} + \mathbf{g}_{t,t-1} \\ \mathbf{x}_t &\leftarrow \mathbf{x}_{t-1} - \eta_t \mathbf{v}_t \end{aligned} \tag{12.6.5}$$

Trong đó:

β là hệ số momentum

$\mathbf{v}$ là quán tính hướng cập nhật (velocity)

η là hệ số học (learning rate)

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

1.3. Định nghĩa

1.4. Ảnh hưởng trọng lượng mẫu

Effective Sample Weight

Hệ số ghi nhớ

$$\mathbf{v}_t = \sum_{\tau=0}^{t-1} \beta^\tau \mathbf{g}_{t-\tau,\, t-\tau-1}$$

Nghĩa là gradient sử dụng lúc cập nhật tại thời điểm t là một "**trung bình có trọng số**" của rất nhiều gradient trong quá khứ.

Với $0 \leq \beta < 1$ và mở rộng vô hạn thời gian

$$\sum_{\tau=0}^{\infty} \beta^\tau = \frac{1}{1-\beta} = 1 + \beta + \beta^2 + \cdots$$

Tăng "quy mô" hiệu quả của gradient (như dùng hiều sample), giúp giảm phương sai (variance) của quá trình cập nhật vì đỡ nhiễu từ gradient đơn lẻ.

Vẫn giữ được hướng đi "mượt" và ổn định nhờ trung bình gradient theo thời gian.

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

1.3. Định nghĩa

1.4. Ảnh hưởng trọng lượng mẫu

1.5. Phân tích lý thuyết

Theoretical Analysis

Momentum là một kỹ thuật tăng tốc (acceleration technique) rất quan trọng trong tối ưu hóa hiện đại.

Ý tưởng cốt lõi của momentum là thay vì chỉ sử dụng gradient hiện tại để cập nhật tham số, ta sẽ tích luỹ một đại lượng gọi là vận tốc (velocity).

$$\begin{aligned} \mathbf{v}_t &\leftarrow \beta \mathbf{v}_{t-1} + \mathbf{g}_{t,t-1} \\ \mathbf{x}_t &\leftarrow \mathbf{x}_{t-1} - \eta_t \mathbf{v}_t \end{aligned} \tag{12.6.5}$$

Mục tiêu:

- Hiểu rõ tai sao Momentum giúp tăng tốc quá trình tối ưu hóa.
- Hiểu rõ cách Momentum giảm dao động trong những vùng ill-conditioned.
- Phân tích động học của thuật toán thông qua việc xét dạng hàm bậc hai lồi (quadratic convex func)
- Chỉ ra điều kiện hội tụ và tốc độ hội tụ.
- Đưa ra mô hình một chiều (scalar function) để trực quan hóa cơ chế dao đợng và suy giảm.

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

1.3. Định nghĩa

1.4. Ảnh hưởng trọng lượng mẫu

1.5. Phân tích lý thuyết

Hàm bậc hai lồi (Quadratic Convex Function)

$$h(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top Q\mathbf{x} + \mathbf{x}^\top c + b, \qquad Q \succ 0$$ (Q là ma trận đối xứng xác định dương.)

Khi đó, gradient của hàm là: $\partial_x h(\mathbf{x}) = Q\mathbf{x} + c = 0$

Do đó, nghiệm tối ưu là: $x^* = -Q^{-1}c$

Viết lại hàm dưới dạng dịch chuyển quanh nghiệm tối ưu:

$$h(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - Q^{-1}c)^\top Q(\mathbf{x} - Q^{-1}c) + b - \frac{1}{2}c^\top Q^{-1}c$$

Trong dạng này, tất cả thông tin về độ cong của hàm đều được chứa trong ma trận Q.

Đạo hàm khi dùng dạng dịch chuyển là:

$$\partial_x h(\mathbf{x}) = Q(\mathbf{x} - Q^{-1}c)$$

Biểu thức này cho thấy gradient luôn hướng từ x về nghiệm tối ưu và độ lớn được điều chỉnh bởi trị riêng của Q.

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

1.3. Định nghĩa

1.4. Ảnh hưởng trọng lượng mẫu

1.5. Phân tích lý thuyết

Hàm bậc hai lồi (Quadratic Convex Function)

Vì Q đối xứng nên ta có thể phân rã Q thành:

$$Q = O^\top \Lambda O$$

O là ma trận trực giao thỏa $O^\top O = I$

$\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$

Ma trận O đóng vai trò xoay hệ trục tọa độ sao cho trong hệ trục mới, ma trận Q trở thành ma trận đường chéo.

Để chuyển sang hệ tọa độ mới, đặt: $z_t = O^\top x_t$

Mô tả giá trị hàm trong hệ eigen như sau: $h(z) = \frac{1}{2} z^\top \Lambda z + b'$

Dạng này chỉ ra rằng mỗi thành phần z_i đóng góp một lượng $\frac{1}{2}\lambda_i z_i^2$ độc lập với các thành phần khác.

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

1.3. Định nghĩa

1.4. Ảnh hưởng trọng lượng mẫu

1.5. Phân tích lý thuyết

Hàm bậc hai lồi (Quadratic Convex Function)

Cập nhật Gradient Descent trong hệ mới:

$$x_t = x_{t-1} - Q(x_{t-1} - x^*) \xrightarrow{\text{Đổi sang hệ eigen, ta được:}} z_t = z_{t-1} - \Lambda z_{t-1}$$

Viết ở dạng thu gọn, ta được: $z_t = (I - \Lambda) z_{t-1}$

Dạng đường chéo của Λ cho phép tách riêng từng chiều $z_t^{(i)} = (1 - \lambda_i)\, z_{t-1}^{(i)}$

- **Tiến trình hội tụ của Gradient Descent chính là sự hội tụ độc lập của từng chuỗi một chiều.**
- **Tốc độ hội tụ của từng chiều phụ thuộc trực tiếp vào trị riêng λ_i**

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

1.3. Định nghĩa

1.4. Ảnh hưởng trọng lượng mẫu

1.5. Phân tích lý thuyết

Hàm bậc hai lồi (Quadratic Convex Function)

Cập nhật Momentum trong hệ mới:

$$v_t = \beta v_{t-1} + \Lambda z_{t-1} \xrightarrow{\text{Đổi sang hệ eigen, ta được:}} z_t = z_{t-1} - \eta(\beta v_{t-1} + \Lambda z_{t-1})$$

Thay biểu thức v_t vào z_t $\quad z_t = z_{t-1} - \eta\Lambda z_{t-1} - \eta\beta v_{t-1}$

Viết ở dạng thu gọn, ta được: $\quad z_t = (I - \eta\Lambda)\, z_{t-1} - \eta\beta v_{t-1}$

Dạng đường chéo của Λ cho phép tách riêng từng chiều, đồng thời các chiều bây giờ trở thành hệ động học bậc hai vì xuất hiện v_{t-1}

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

1.3. Định nghĩa

1.4. Ảnh hưởng trọng lượng mẫu

1.5. Phân tích lý thuyết

Hàm bậc hai lồi (Quadratic Convex Function)

Cập nhật Momentum trong hệ mới:

$$v_t = \beta v_{t-1} + \Lambda z_{t-1} \xrightarrow{\text{Đổi sang hệ eigen, ta được:}} z_t = z_{t-1} - \eta(\beta v_{t-1} + \Lambda z_{t-1})$$

Kết hợp với biểu thức, ta có: $v_t = \beta v_{t-1} + \lambda_i z_{t-1}^{(i)}$

Thu được phương trình truy hồi bậc hai cho từng chiều: $z_t^{(i)} = (1 - \eta\lambda_i + \beta)\, z_{t-1}^{(i)} - \beta\, z_{t-2}^{(i)}$

Phương trình đặc trưng: $r^2 - (1 - \eta\lambda_i + \beta)r + \beta = 0$

Thuật toán hội tụ khi nghiệm của phương trình trên thỏa $|r| < 1$ cho mọi trị riêng λ_i

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

1.3. Định nghĩa

1.4. Ảnh hưởng trọng lượng mẫu

1.5. Phân tích lý thuyết

Hàm bậc hai lồi (Quadratic Convex Function)

Nhờ phân rã eigen, bài toán tối ưu nhiều chiều giảm xuống thành nhiều bài toán một chiều độc lập. Mỗi chiều có dạng tương ứng với một hệ dao động tuyến tính bậc hai.

Từ đó ta có thể dễ dàng phân tích:

- Điều kiện hội tụ
- Dao động
- Tốc độ hội tụ
- Lựa chọn η, β tối ưu

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

1.3. Định nghĩa

1.4. Ảnh hưởng trọng lượng mẫu

1.5. Phân tích lý thuyết

Hàm một biến (Scalar function)

Xét hàm số: $f(x) = \frac{1}{2}\lambda x^2, \qquad \lambda > 0$

Ta có gradient của hàm số: $f'(x) = \lambda x$

Nghiệm tối ưu là: $x^* = 0$

Gradient Descent cập nhật theo quy tắc: $x_{t+1} = x_t - \eta f'(x_t)$

Thay đạo hàm của f(x), ta có: $x_{t+1} = x_t - \eta\lambda x_t$

Đặt x_t ra ngoài, ta có $x_{t+1} = (1 - \eta\lambda)x_t$

Đây là phương trình truy hồi bậc nhất cho chuỗi x_t . Mỗi bước cập nhật nhân với hệ số $1 - \eta\lambda$

Để thuật toán hội tụ về 0, yêu cầu $|1 - \eta\lambda| < 1$

Suy ra, điều kiện của learning rate: $0 < \eta < \frac{2}{\lambda}$

- λ càng lớn, hàm cong hơn, thì η phải càng nhỏ.
- Khi $\eta \to \frac{2}{\lambda}$, thuật toán bắt đầu dao động.

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

1.3. Định nghĩa

1.4. Ảnh hưởng trọng lượng mẫu

1.5. Phân tích lý thuyết

Hàm một biến (Scalar function)

Trong bài toán nhiều chiều với $Q = O^\top \Lambda O$, mỗi trị riêng λ_i đóng vai trò hoàn toàn giống λ trong bài toán vô hướng.

Hệ truy hồi trong từng chiều eigen là:

$$z_{t+1}^{(i)} = (1 - \eta\lambda_i) z_t^{(i)}$$

=> Phân tích hội tụ trong không gian nhiều chiều là kiểm tra điều kiện:

$$|1 - \eta\lambda_i| < 1, \quad \forall \lambda_i$$

=> Learning rate trong nhiều chiều phải thỏa điều kiện:

$$0 < \eta < \frac{2}{\lambda_{\max}}$$

=> Phân tích vô hướng do đó đóng vai trò mô hình chuẩn cho từng chiều eigen khi phân tích theo ma trận Hessian.

1 GIỚI THIỆU

1.1. Phát biểu bài toán

1.2. Các bài toán

1.3. Định nghĩa

1.4. Ảnh hưởng trọng lượng mẫu

1.5. Phân tích lý thuyết

Hàm một biến (Scalar function)

Gradient Descent hội tụ tuyến tính với tỉ lệ hội tụ phụ thuộc vào λ

Khi độ cong lớn, thuật toán dễ dao động nếu η không đủ nhỏ

Công thức $1 - \eta\lambda$ chính là phiên bản 1 chiều của phương trình đa chiều $I - \eta Q$

Tất cả phân tích đối với bài toán nhiều chiều đều quy về phân tích vô hướng với mỗi trị riêng λ_i

2 ỨNG DỤNG

2.1. Ứng dụng của Momentum

Huấn luyện Deep Neural Networks

Trong các mạng lớn như CNN, RNN, Transformer, bề mặt loss rất phức tạp thì Momentum giúp:

- Giảm dao động gradient theo các chiều dốc.
- Tăng tốc theo các chiều phẳng (flat/sloppy directions).
- Hội tụ nhanh hơn nhiều so với SGD thông thường.

Giảm vấn đề mạng Vanishing Gradient

Ở những vùng gradient rất nhỏ, Momentum có thể tích lũy nhiều bước để vẫn tạo ra cập nhật đủ lớn để thoát vùng “chết”.

2 ỨNG DỤNG

2.1. Ứng dụng của Momentum

Tối ưu bài toán Machine Learning cổ điển

Momentum giúp các thuật toán này tiến nhanh hơn và ít dao động.

- Logistic regression
- Linear regression
- Matrix factorization
- Collaborative filtering
- PCA optimization (stochastic gradient)

Tối ưu trong Robotics / Control / Physics Simulation

Do bản chất mô phỏng từ vật lý (momentum), thuật toán hoạt động hiệu quả trong hệ động lực.

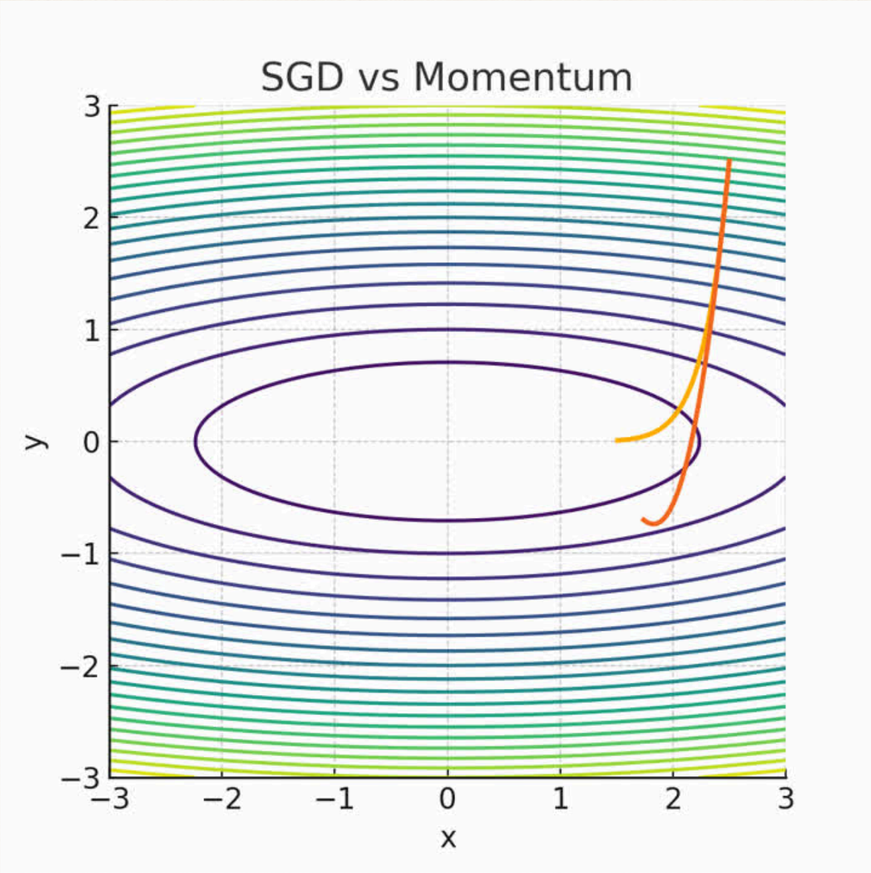
2 ỨNG DỤNG

2.1. Ứng dụng của Momentum

2.2. Trong Khoa học Máy tính

Bài toán rãnh hẹp (ravine problem)

$$f(x, y) = 0.1x^2 + y^2$$



Gradient descent thường:

- dao động mạnh ở chiều y
- tiến chậm ở chiều x

Momentum giúp lấy hướng trung bình, giúp giảm zig-zag.

2 ỨNG DỤNG

2.1. Ứng dụng của Momentum

2.2. Trong Khoa học Máy tính

Bài toán logistic regression với dữ liệu không chuẩn hóa

Khi một số feature có scale lớn – nhỏ khác nhau, gradient descent rất "nảy".
Momentum giúp cân bằng việc cập nhật và giảm jitter.

Bài toán tối ưu mạng sâu (deep network loss landscape)

Landscape của deep learning thường có:

- Saddle points
- Flat region
- Sharp minima

SGD thường đứng yên ở flat region, nên Momentum tích lũy gradient và "đẩy qua" vùng này.

Bài toán SGD với batch nhỏ

Khi kích thước batch nhỏ, gây ra nhiễu cho SGD.
Nên Momentum giúp làm mượt (smoothing) gradient theo thời gian.

3 BÀI TẬP

3.1. Bài tập 1

Use other combinations of momentum hyper-parameters and learning rates and observe and analyze the different experimental results.

$$f(x) = 0.1x_1^2 + 2x_2^2$$

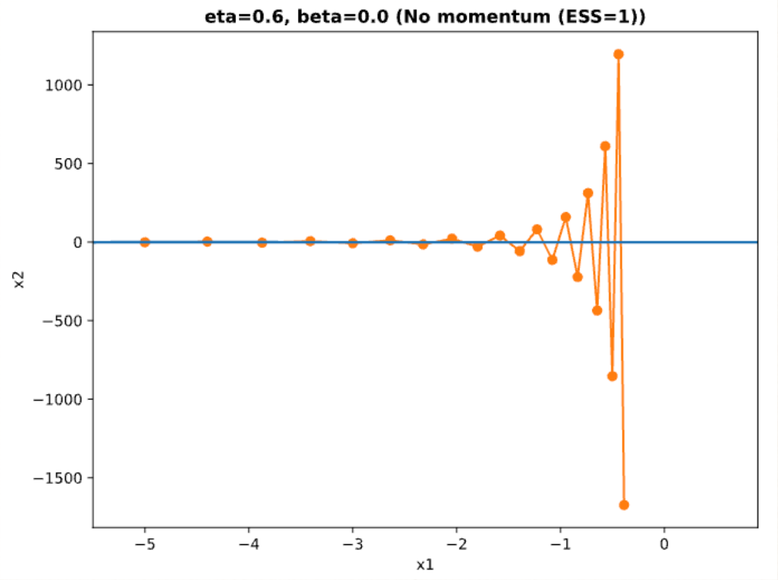
Trong đó: Gradient của hai biến tại thời điểm t lần lượt là

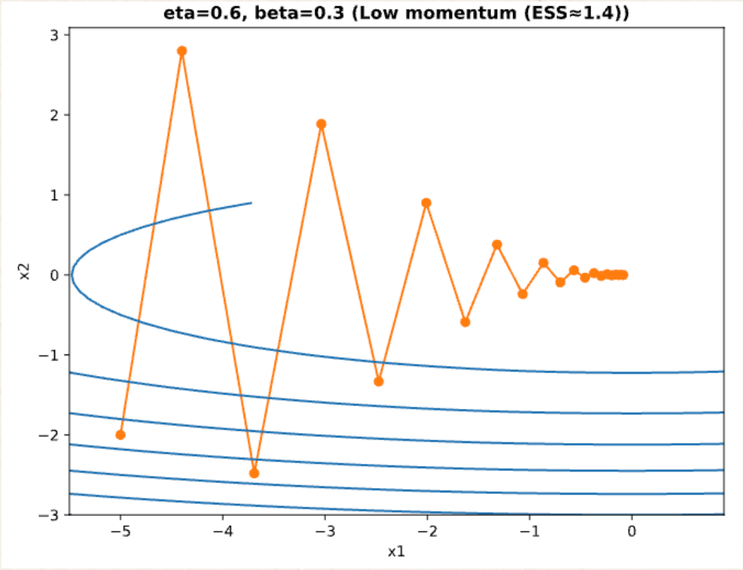
$$\mathbf{g}_{1,t} = 0.2x_1$$
$$\mathbf{g}_{2,t} = 4x_2$$

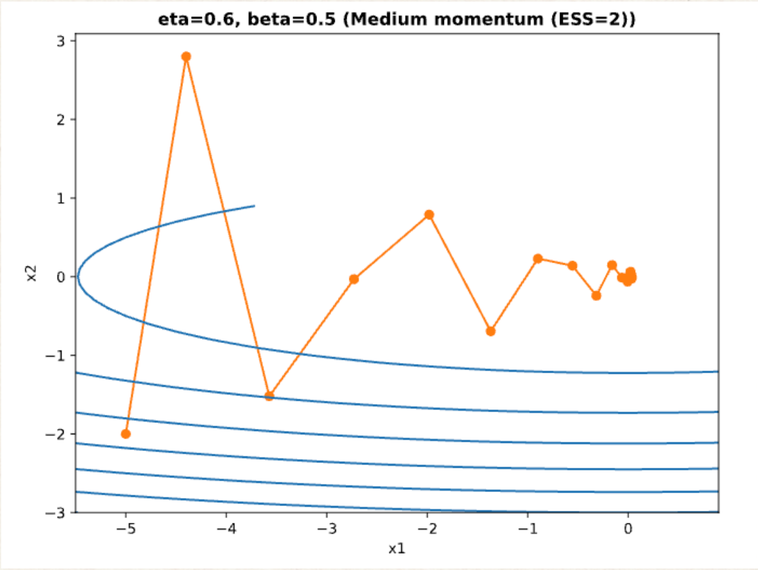
Thí nghiệm sẽ tiến hành trong ba trường hợp:

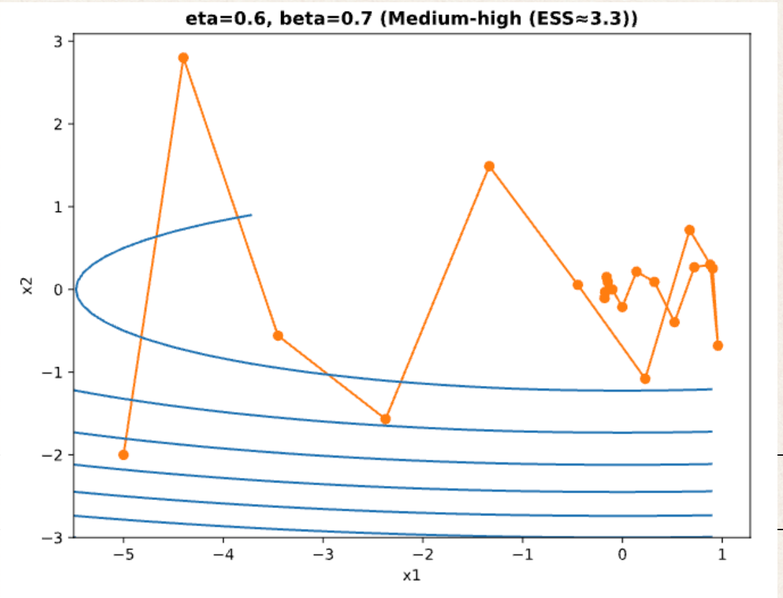
1. Thay đổi Beta, cố định Eta.
2. Cố định Beta, thay đổi Eta.
3. Thay đổi Beta và Eta.

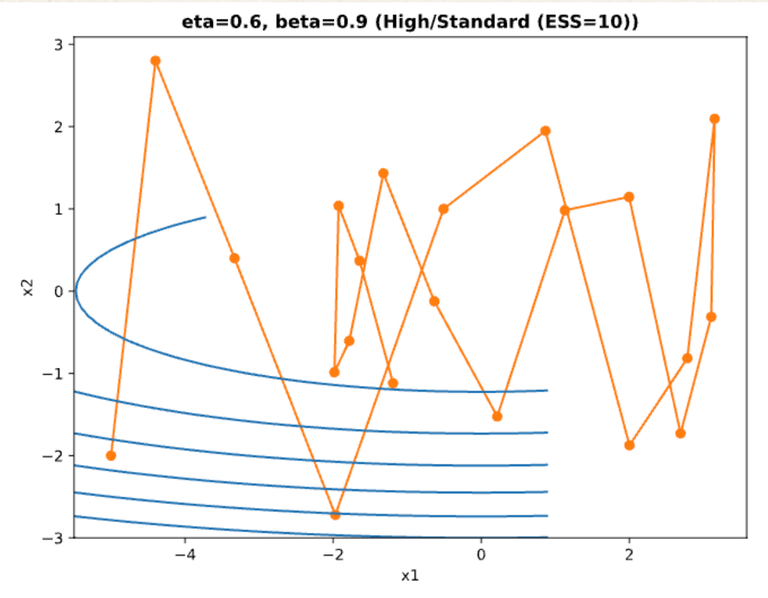
Thí nghiệm 1: Cố định eta, thay đổi Beta = 0; 0.3; 0.5; 0.7; 0.9; 0.95

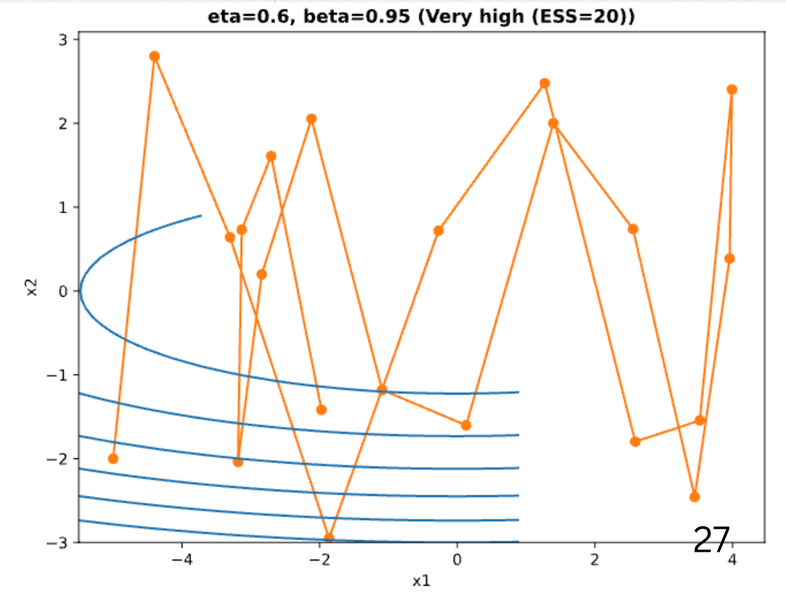




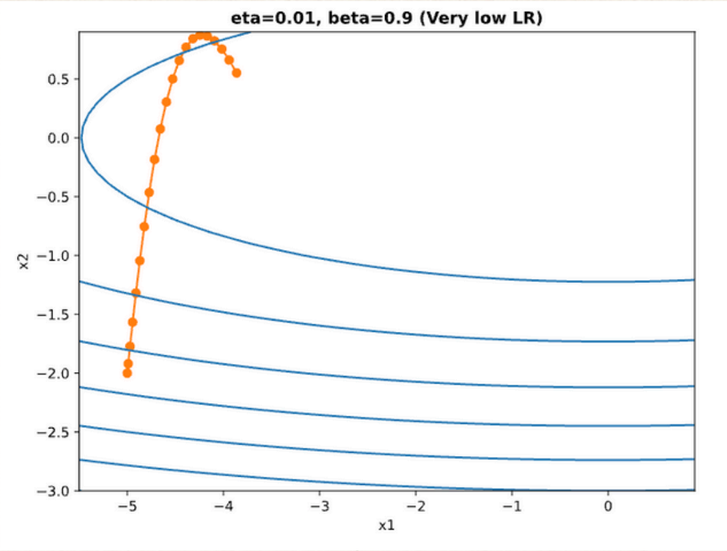


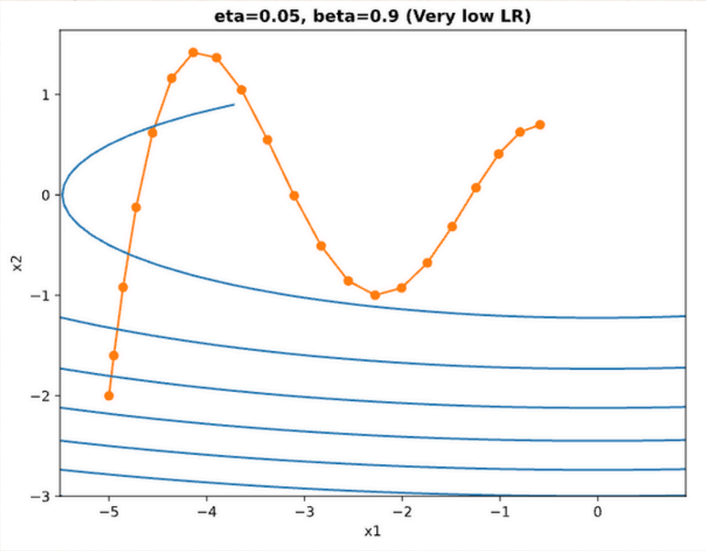


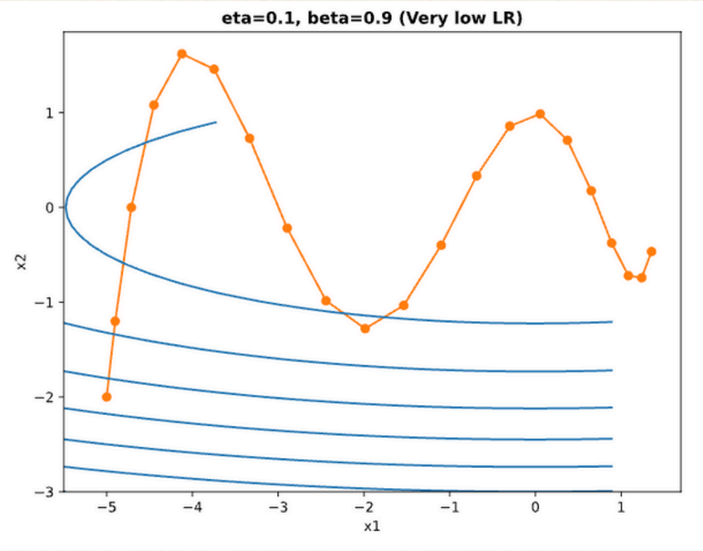


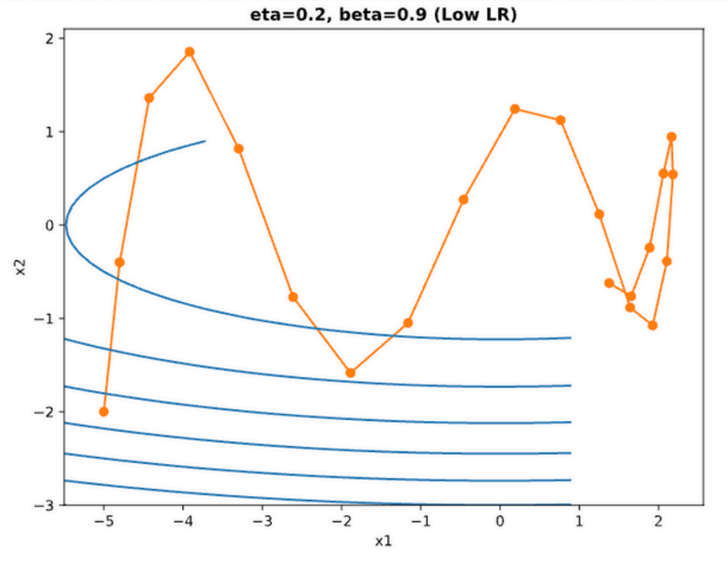


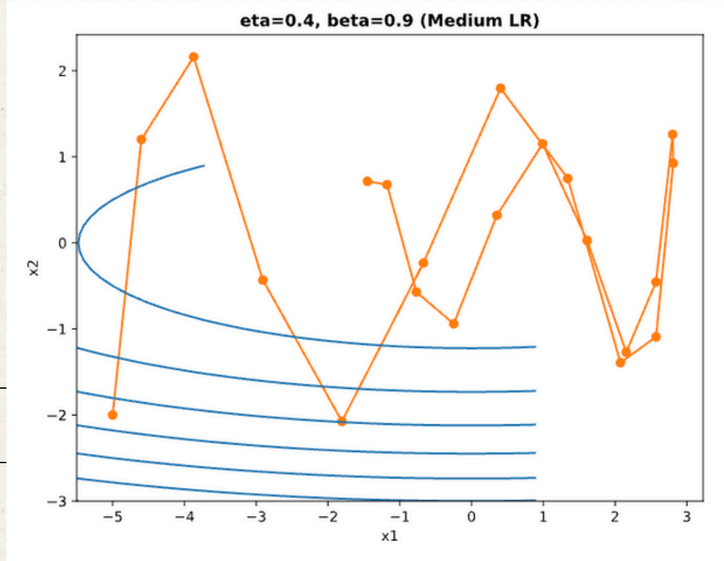
Thí nghiệm 2: Cố định beta = 0.9, thay đổi eta = 0.01; 0.05; 0.1; 0.2; 0.4; 0.6

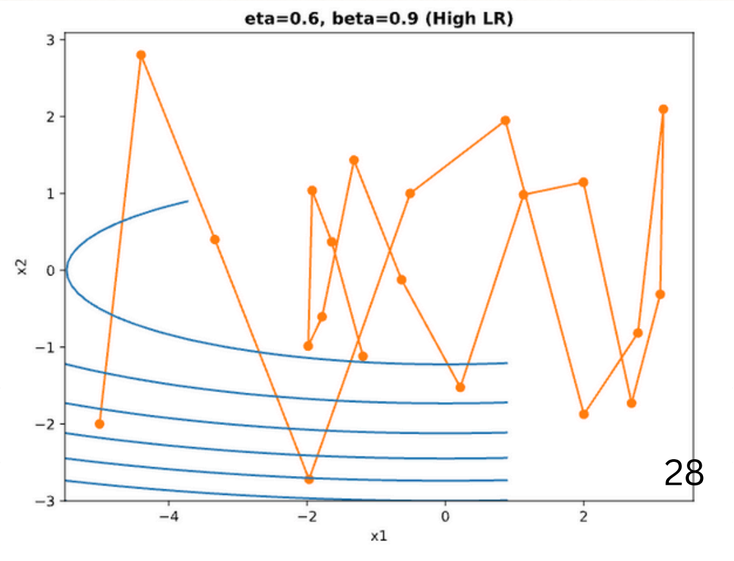




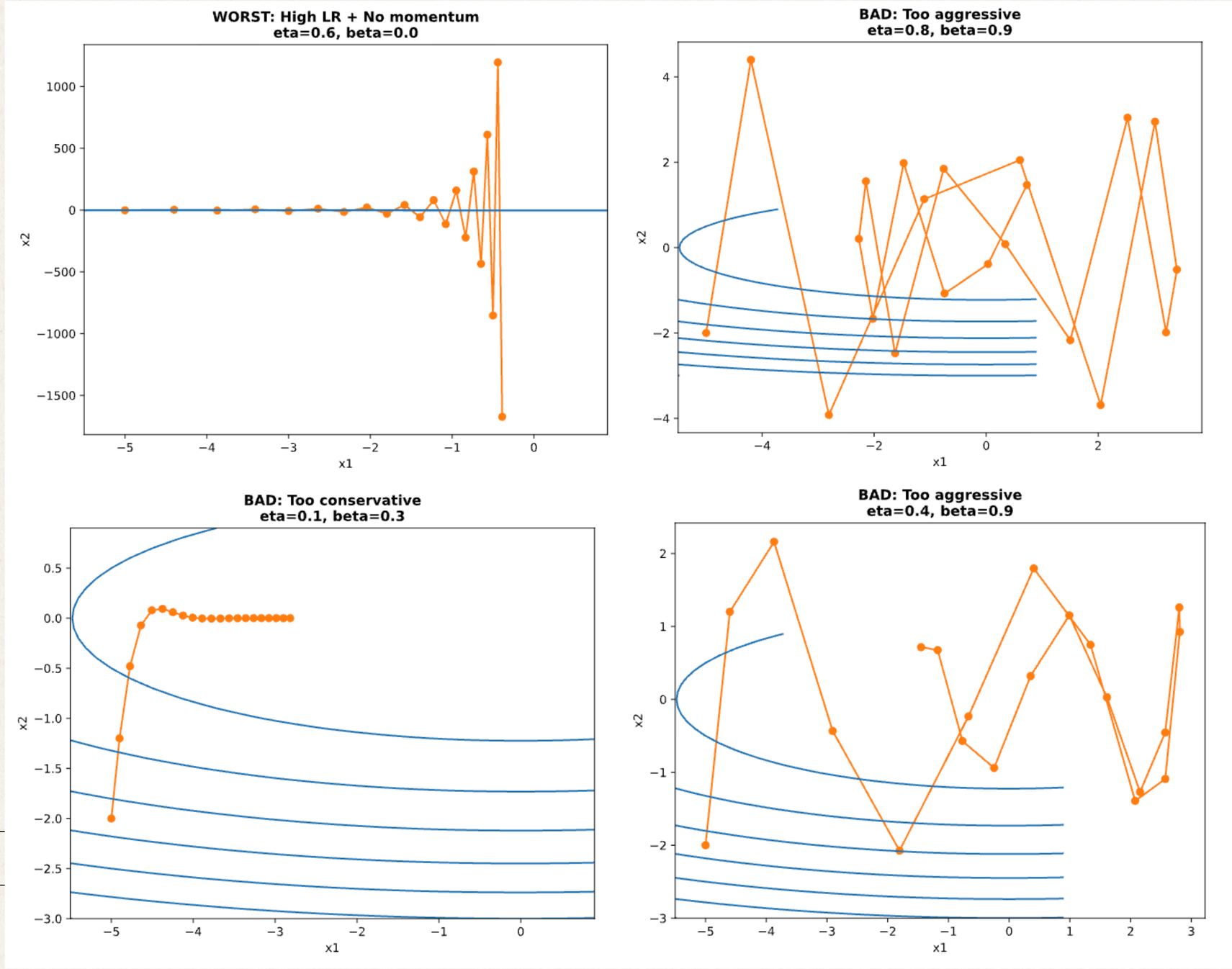




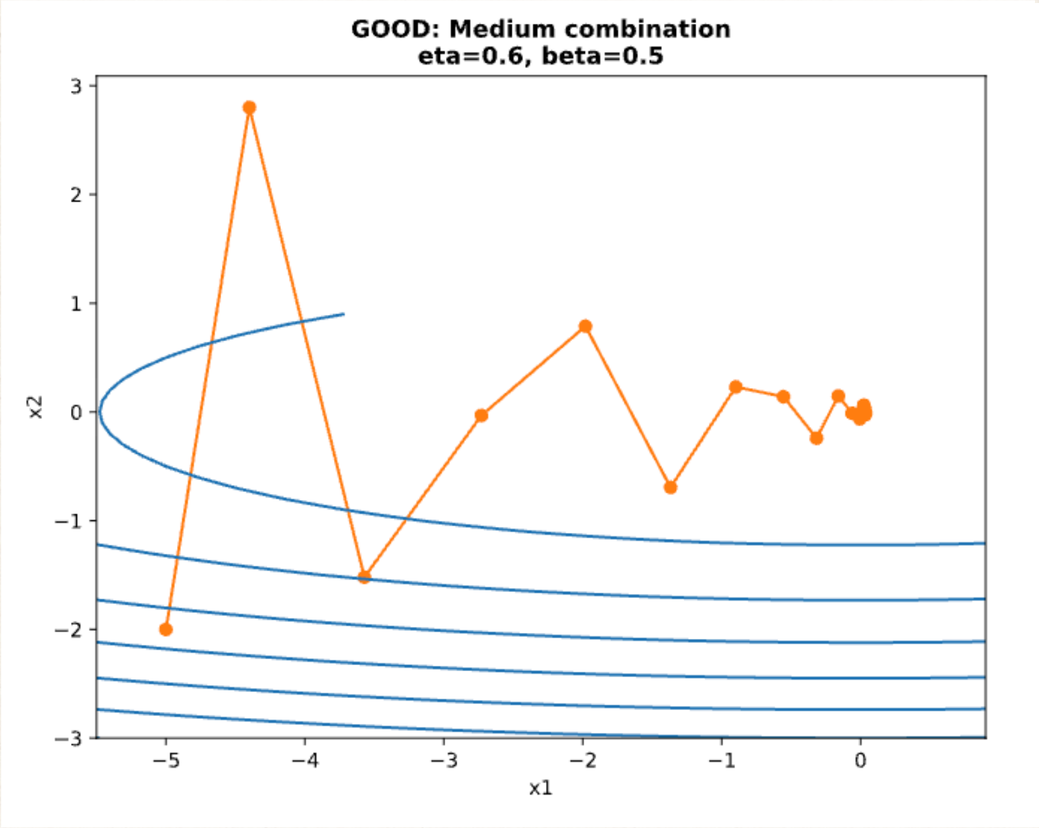


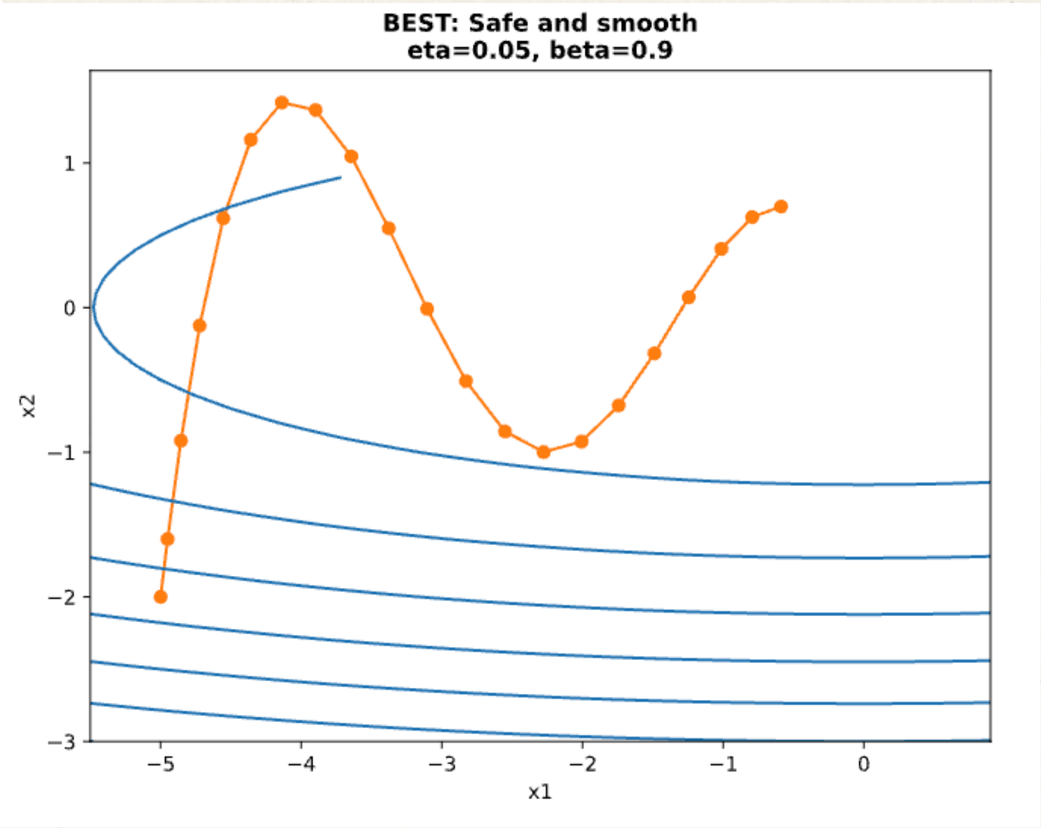


Thí nghiệm 3: Thay đổi hỗn hợp



Thí nghiệm 3: Thay đổi hỗn hợp





3 BÀI TẬP

3.1. Bài tập 1

3.2. Bài tập 2

Try out gradient descent and momentum for a quadratic problem where you have multiple eigenvalues, i.e.,

$$f(x) = \frac{1}{2}\sum_i \lambda_i x_i^2, \qquad \lambda_i = 2^{-i}$$

Ta có:

$$f(x) = \frac{1}{2}\sum_i \lambda_i x_i^2, \qquad \lambda_i = 2^{-i}$$

$$= \frac{1}{2}\sum_i 2^{-i} x_i^2$$

Gọi

$$\mathbf{x} = [x_1, x_2, \ldots, x_i]^\top$$

Điểm tối ưu sẽ là vector

$$\mathbf{x}^* = [0, 0, \ldots, 0]^\top$$

3 BÀI TẬP

3.1. Bài tập 1

3.2. Bài tập 2

$$f(x) = \frac{1}{2}\sum_i \lambda_i x_i^2, \qquad \lambda_i = 2^{-i}$$
$$= \frac{1}{2}\sum_i 2^{-i} x_i^2$$

Try out gradient descent and momentum for a quadratic problem where you have multiple eigenvalues, i.e.,

Với Gradient Descent

$$\mathbf{x}_0 = [1, 1, \ldots, 1]^\top$$
$$\eta = 0.1$$

$$\nabla f(x) = [\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_i}]^\top$$
$$= [\frac{\partial(\frac{1}{2}2^{-1}x_1^2)}{\partial x_1}, \frac{\partial(\frac{1}{2}2^{-2}x_2^2)}{\partial x_2}, \ldots, \frac{\partial(\frac{1}{2}2^{-i}x_i^2)}{\partial x_i},]^\top$$
$$= [2^{-1}x_1, 2^{-2}x_2, \ldots, 2^{-i}x_i]^\top$$

Vòng lặp 1:

$$\nabla f(x_0) = [2^{-1}, 2^{-2}, \ldots, 2^{-i}]$$

$$\mathbf{x}_1 = \mathbf{x}_0 - \eta \nabla f(x_0)$$
$$= [1, 1, \ldots, 1]^\top - 0.1[2^{-1}x_1, 2^{-2}x_2, \ldots, 2^{-i}x_i]^\top$$
$$= [0.95, 0.975, \ldots, 1 - 0.1 \cdot 2^{-i}]$$

=> Phương pháp Gradient Descent sẽ làm cho các vị trí x_i với i lớn sẽ về điểm $\mathcal{O}$ rất lâu. Nếu $i \to \infty$, thậm chí còn không di chuyển.

3 BÀI TẬP

3.1. Bài tập 1

3.2. Bài tập 2

$$f(x) = \frac{1}{2}\sum_i \lambda_i x_i^2, \qquad \lambda_i = 2^{-i}$$
$$= \frac{1}{2}\sum_i 2^{-i} x_i^2$$

Try out gradient descent and momentum for a quadratic problem where you have multiple eigenvalues, i.e.,

Với **Momentum**

$$\begin{aligned}
\beta &= 0.5 \\
\eta &= 0.1 \\
\mathbf{v}_0 &= [0, 0, \ldots, 0]^\top \\
\mathbf{x}_0 &= [1, 1, \ldots, 1]^\top
\end{aligned}$$

Vòng lặp 1:

$$\nabla f(x_0) = [2^{-1}, 2^{-2}, \ldots, 2^{-i}]^\top$$

$$\begin{aligned}
\mathbf{v}_1 &= \beta \mathbf{v}_0 + \nabla f(x_0) \\
&= 0.5[0, 0, \ldots, 0]^\top + [2^{-1}, 2^{-2}, \ldots, 2^{-i}]^\top \\
&= [2^{-1}, 2^{-2}, \ldots, 2^{-i}]^\top \\
&= [\lambda_1, \lambda_2, \ldots, \lambda_i]^\top
\end{aligned}$$

$$\begin{aligned}
\mathbf{x}_1 &= \mathbf{x}_0 - \eta \mathbf{v}_1 \\
&= [1, 1, \ldots, 1]^\top - 0.1[2^{-1}, 2^{-2}, \ldots, 2^{-i}]^\top \\
&= [0.95, 0.975, \ldots, 1 - 0.1 \cdot 2^{-i}]^\top \\
&= [1 - 0.1\lambda_1, 1 - 0.1\lambda_2, \ldots, 1 - 0.1\lambda_i]^\top
\end{aligned}$$

3 BÀI TẬP

3.1. Bài tập 1

3.2. Bài tập 2

$$f(x) = \frac{1}{2}\sum_i \lambda_i x_i^2, \qquad \lambda_i = 2^{-i}$$
$$= \frac{1}{2}\sum_i 2^{-i} x_i^2$$

Try out gradient descent and momentum for a quadratic problem where you have multiple eigenvalues, i.e.,

Với **Momentum**

$$\beta = 0.5$$
$$\eta = 0.1$$
$$\mathbf{v}_0 = [0, 0, \ldots, 0]^\top$$
$$\mathbf{x}_0 = [1, 1, \ldots, 1]^\top$$

Vòng lặp 1:

$$\nabla f(x_0) = [2^{-1}, 2^{-2}, \ldots, 2^{-i}]^\top$$
$$\mathbf{v}_1 = [\lambda_1, \lambda_2, \ldots, \lambda_i]^\top$$
$$\mathbf{x}_1 = [1 - 0.1\lambda_1, 1 - 0.1\lambda_2, \ldots, 1 - 0.1\lambda_i]^\top$$

Vòng lặp 2:

$$\begin{aligned}\nabla f(x_1) &= [\lambda_1(1 - 0.1\lambda_1), \lambda_2(1 - 0.1\lambda_2), \ldots, \lambda_i(1 - 0.1\lambda_i)]^\top \\ &= [\lambda_1 - 0.1\lambda_1^2, \lambda_2 - 0.1\lambda_2^2, \ldots, \lambda_i - 0.1\lambda_i^2]^\top\end{aligned}$$

$$\begin{aligned}\mathbf{v}_2 &= \beta\mathbf{v}_1 + \nabla f(x_1) \\ &= 0.5[\lambda_1, \lambda_2, \ldots, \lambda_i]^\top + [\lambda_1 - 0.1\lambda_1^2, \lambda_2 - 0.1\lambda_2^2, \ldots, \lambda_i - 0.1\lambda_i^2]^\top \\ &= [1.5\lambda_1 - 0.1\lambda_1^2, 1.5\lambda_2 - 0.1\lambda_2^2, \ldots, 1.5\lambda_i - 0.1\lambda_i^2]^\top\end{aligned}$$

$$\begin{aligned}\mathbf{x}_2 &= \mathbf{x}_1 - \eta\mathbf{v}_2 \\ &= [1 - 0.1\lambda_1, 1 - 0.1\lambda_2, \ldots, 1 - 0.1\lambda_i]^\top - 0.1[1.5\lambda_1 - 0.1\lambda_1^2, 1.5\lambda_2 - 0.1\lambda_2^2, \ldots, 1.5\lambda_i - 0.1\lambda_i^2]^\top \\ &= [1 - 0.25\lambda_1 - 0.01\lambda_1^2, 1 - 0.25\lambda_2 - 0.01\lambda_2^2, \ldots, 1 - 0.25\lambda_i - 0.01\lambda_i^2]^\top\end{aligned}$$

3 BÀI TẬP

3.1. Bài tập 1

3.2. Bài tập 2

3.3. Bài tập 3

Derive minimum value and minimizer for $h(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{x}^\top \mathbf{c} + b$

Trong đó: $\mathbf{x}^\top = [x_1, x_2, \ldots, x_n]^\top \Rightarrow \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} Q_{11} & Q12 & \cdots & Q_{1n} \\ Q_{21} & Q22 & \cdots & Q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Q_{n1} & Qn2 & \cdots & Q_{nn} \end{bmatrix}$$

3 BÀI TẬP

3.1. Bài tập 1

3.2. Bài tập 2

3.3. Bài tập 3

Derive minimum value and minimizer for $h(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{x}^\top \mathbf{c} + b$

B1. Tính Gradient của hàm

Theo định nghĩa của ma trận $(\mathbf{Q}\mathbf{x})_i = \sum_{j=1}^{n} \mathbf{Q}_{ij}\mathbf{x}_j$, nghĩa là: $\mathbf{Q}\mathbf{x} = \begin{bmatrix} \sum_{j=1}^{n} \mathbf{Q}_{1j}\mathbf{x}_j \\ \sum_{j=1}^{n} \mathbf{Q}_{2j}\mathbf{x}_j \\ \vdots \\ \sum_{j=1}^{n} \mathbf{Q}_{nj}\mathbf{x}_j \end{bmatrix}$

$$\Rightarrow \mathbf{x}^\top Q\mathbf{x} = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{x}_i Q_{ij}\mathbf{x}_j, \qquad \mathbf{x}^\top \mathbf{c} = \sum_{i=1}^{n} \mathbf{x}_i c_i$$

Nếu (i = k), thì hạng tử có dạng $(\mathbf{x}_k Q_{kj}\mathbf{x}_j)$ sẽ có đạo hàm theo $(\mathbf{x}_k)$ là: $(Q_{kj}\mathbf{x}_j)$
Nếu (j = k), thì hạng tử có dạng $(\mathbf{x}_i Q_{ik}\mathbf{x}_k)$ sẽ có đạo hàm theo $(\mathbf{x}_k)$ là: $(\mathbf{x}_i Q_{ik})$

Do đó: $$\frac{\partial}{\partial \mathbf{x}_k}\left(\frac{1}{2}\sum_{i,j} \mathbf{x}_i Q_{ij}\mathbf{x}_j\right) = \frac{1}{2}\sum_{j} Q_{kj}\mathbf{x}_j + \frac{1}{2}\sum_{i} \mathbf{x}_i Q_{ik}$$

Cộng $\mathbf{x}^\top \mathbf{c}$ và hằng số: $$\frac{\partial h}{\partial \mathbf{x}_k} = \frac{1}{2}\sum_{j} Q_{kj}\mathbf{x}_j + \frac{1}{2}\sum_{i} \mathbf{x}_i Q_{ik} + c_k$$

$$\Rightarrow \nabla h(\mathbf{x}) = \frac{1}{2}(Q + Q^\top)\mathbf{x} + \mathbf{c}$$

Khi Q đối xứng $$\Rightarrow \nabla h(\mathbf{x}) = Q\mathbf{x} + \mathbf{c}$$

3 BÀI TẬP

3.1. Bài tập 1

3.2. Bài tập 2

3.3. Bài tập 3

Derive minimum value and minimizer for $h(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{x}^\top \mathbf{c} + b$

B2. Tìm nghiệm đóng

Điều kiện điểm tới hạn: $\nabla h(\mathbf{x}^*) = 0 \iff Q\mathbf{x}^* + \mathbf{c} = 0 \iff Q\mathbf{x}^* = -\mathbf{c}$

Nhân hai vế với $\mathbf{Q}^{-1}$, ta có: $\mathbf{Q}^{-1}\mathbf{Q}\mathbf{x}^* = -\mathbf{Q}^{-1}\mathbf{c}$

Vì Q khả nghịch, nên: $\mathbf{Q}^{-1}\mathbf{Q} = 1$

Khi đó, nghiệm tối thiểu là: $\mathbf{x}^* = -\mathbf{Q}^{-1}\mathbf{c}$

Giá trị tối thiểu là:

$$\begin{aligned} h(\mathbf{x}^*) &= \tfrac{1}{2}(\mathbf{x}^*)^\top \mathbf{Q}\mathbf{x}^* + (\mathbf{x}^*)^\top \mathbf{c} + b \\ &= \tfrac{1}{2}(-\mathbf{Q}^{-1}\mathbf{c})^\top \mathbf{Q}(-\mathbf{Q}^{-1}\mathbf{c}) + (-\mathbf{Q}^{-1}\mathbf{c})^\top \mathbf{c} + b \\ &= \tfrac{1}{2}\mathbf{c}^\top \mathbf{Q}^{-1}\mathbf{c} - \mathbf{c}^\top \mathbf{Q}^{-1}\mathbf{c} + b \\ &= b - \tfrac{1}{2}\mathbf{c}^\top \mathbf{Q}^{-1}\mathbf{c} \end{aligned}$$

3 BÀI TẬP

3.1. Bài tập 1

3.2. Bài tập 2

3.3. Bài tập 3

3.4. Bài tập 4

What changes when we perform stochastic gradient descent with momentum?
What happens when we use mini-batch stochastic gradient descent with momentum?
Experiment with the parameters?

$$\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \eta \mathbf{g}_{t,t-1}$$

Full Gradient

$$\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \eta \nabla f(\mathbf{x}) = \mathbf{x}_{t-1} - \eta \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\mathbf{x})$$

SGD

$$\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \eta \nabla f_i(\mathbf{x})$$

Mini SGD

$$\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \eta \frac{1}{\mathcal{B}_t} \sum_{i \in \mathcal{B}_t} \nabla f_i(\mathbf{x}_t)$$

SGD w. Momentum

$$\mathbf{v}_t \leftarrow \beta \mathbf{v}_{t-1} + \nabla f_i(\mathbf{x})$$
$$\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \eta \mathbf{v}_t$$

Mini SGD w. Momentum

$$\mathbf{v}_t \leftarrow \beta \mathbf{v}_{t-1} + \frac{1}{\mathcal{B}_t} \sum_{i \in \mathcal{B}_t} \nabla f_i(\mathbf{x}_t)$$
$$\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \eta \mathbf{v}_t$$

3 BÀI TẬP

3.1. Bài tập 1

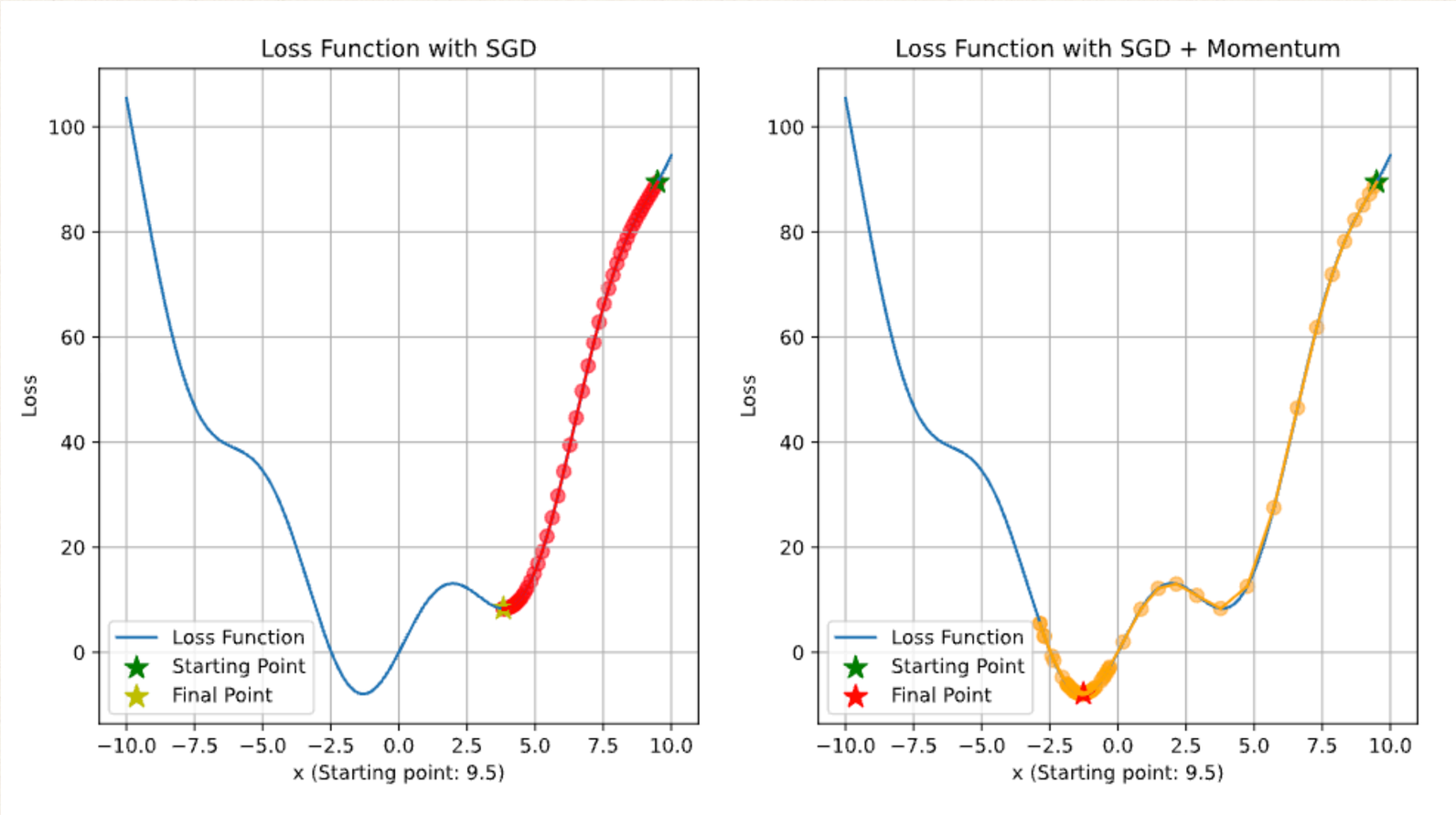
3.2. Bài tập 2

3.3. Bài tập 3

3.4. Bài tập 4

$$f(x) = x^2 + 10sin(x)$$

Với điểm bắt đầu là 9.5; Beta = 0.9; hệ số learning rate = 0.01



SGD đến điểm local minimum trong 137 vòng lặp.
Trong khi SGD với Momentum đến điểm minimum trong 74 vòng lặp.

3 BÀI TẬP

3.1. Bài tập 1

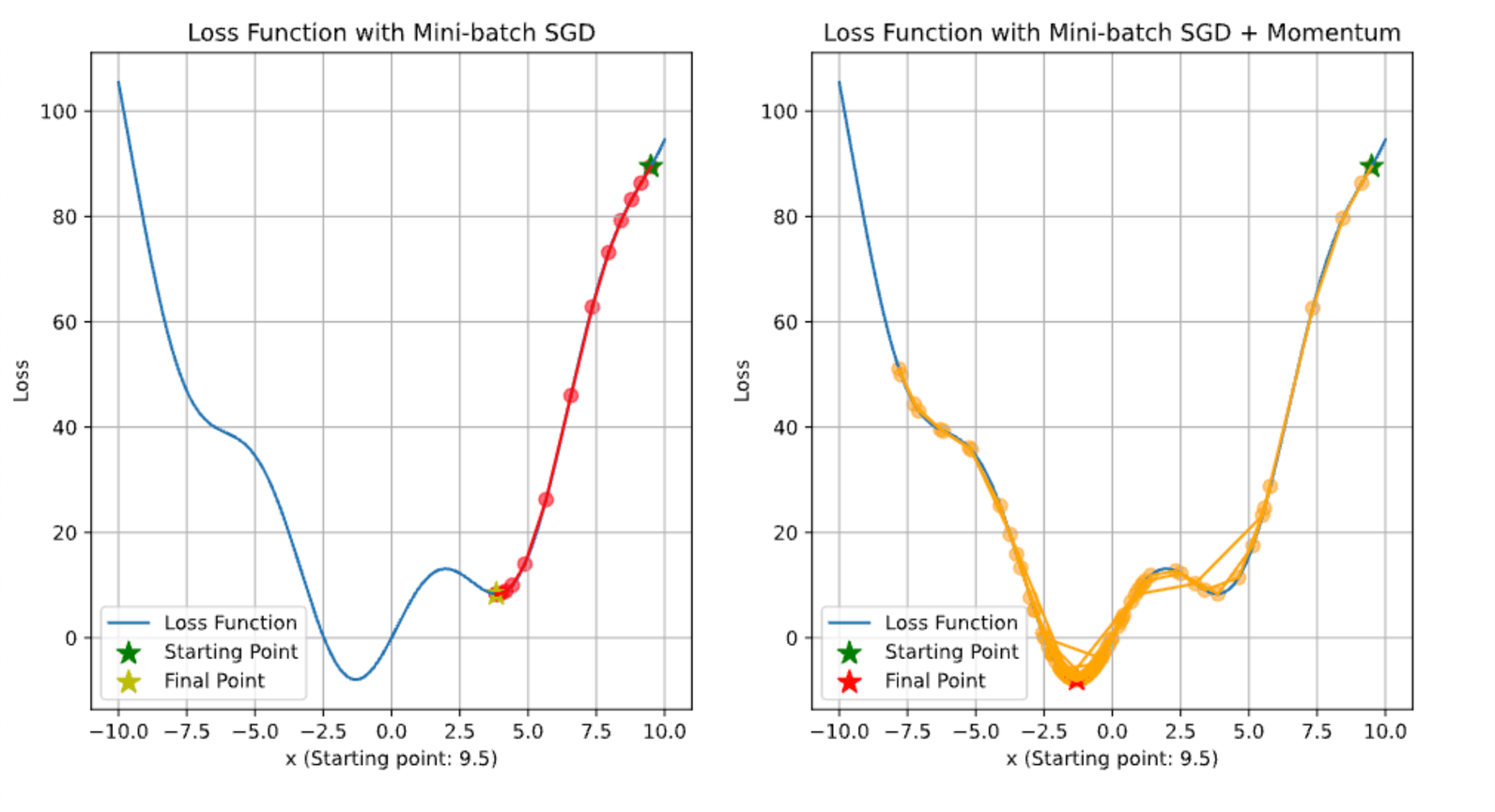
3.2. Bài tập 2

3.3. Bài tập 3

3.4. Bài tập 4

$$f(x) = x^2 + 10sin(x)$$

Với điểm bắt đầu là 9.5; Beta = 0.95; hệ số learning rate = 0.04



Mini-SGD đến điểm local minimum trong 29 vòng lặp.
Trong khi SGD với Momentum đến điểm minimum trong 306 vòng lặp.

4 TỔNG KẾT

Momentum thay thế gradient bằng một giá trị trung bình không hoàn chỉnh trên các gradient trong quá khứ. Điều này giúp tăng tốc độ hội tụ đáng kể.

Nó hữu ích cho cả phương pháp giảm gradient không nhiễu và phương pháp giảm gradient ngẫu nhiên (có nhiễu).

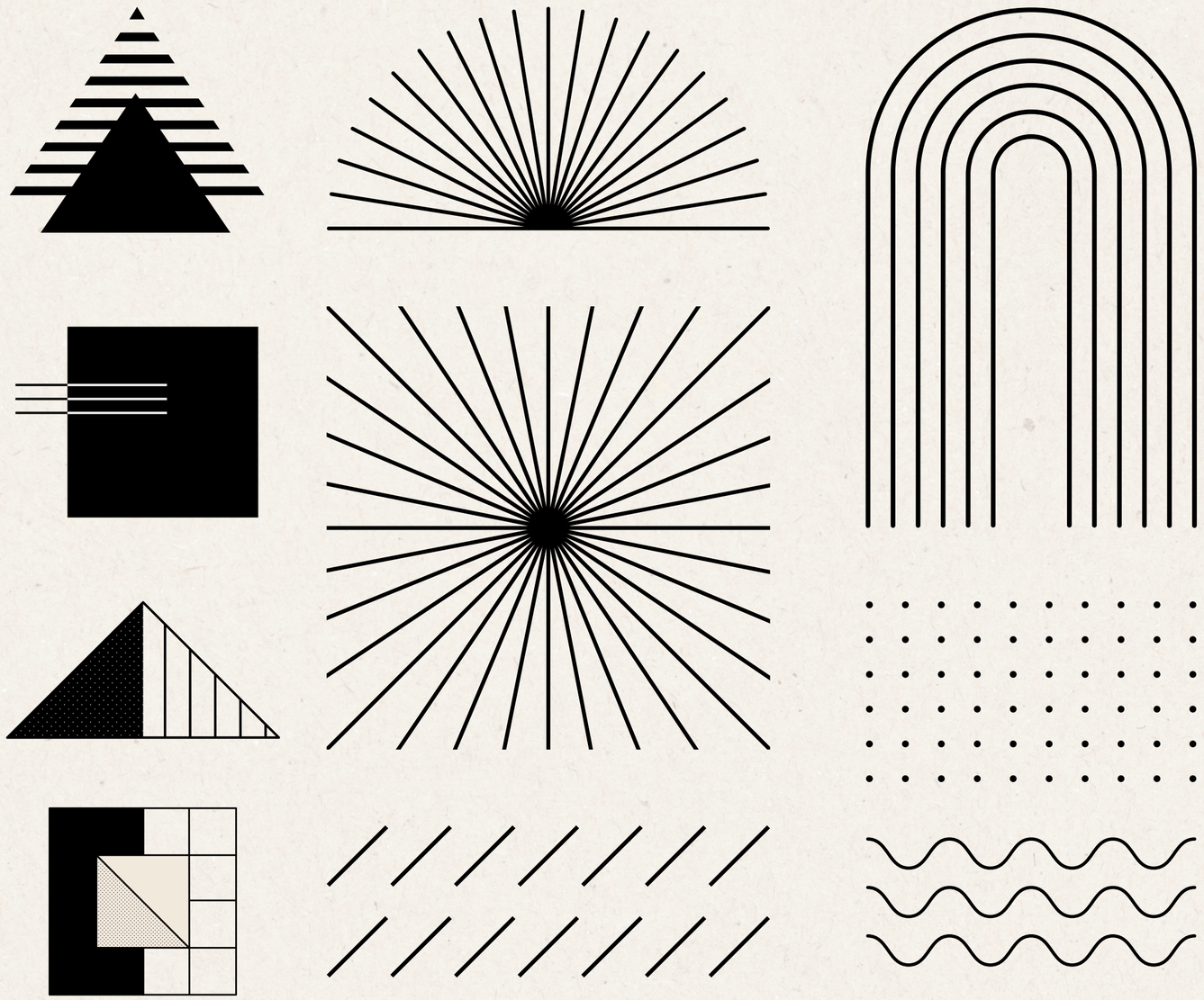
Momentum ngăn chặn sự đình trệ của quá trình tối ưu hóa, điều này dễ xảy ra hơn nhiều đối với phương pháp giảm gradient ngẫu nhiên.

Số lượng đạo hàm hiệu quả được cho bởi $\frac{1}{1-\beta}$ do việc giảm trọng số theo lũy thừa của dữ liệu trong quá khứ.

Việc triển khai khá đơn giản nhưng yêu cầu chúng ta lưu trữ thêm một vectơ trạng thái (vận tốc).

Momentum mang lại tốc độ hội tụ cao hơn và độ ổn định tốt hơn khi tối ưu, là công cụ thiết yếu trong các hệ tối ưu hiện đại và có tính ứng dụng rộng rãi trong thực tế.

Thank you for listening

HERE’RE ALL DEMONSTRATIONS OF MY TEAM ABOUT MOMENTUM IN OPTIMIZATION PROBLEMS!

IT’S QUESTIONING AND ANSWERING TIME!