ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

TOÁN CHO KHOA HỌC MÁY TÍNH

CÁC PHƯƠNG PHÁP GRADIENT

GRADIENT METHODS

TS. Lương Ngọc Hoàng

NỘI DUNG

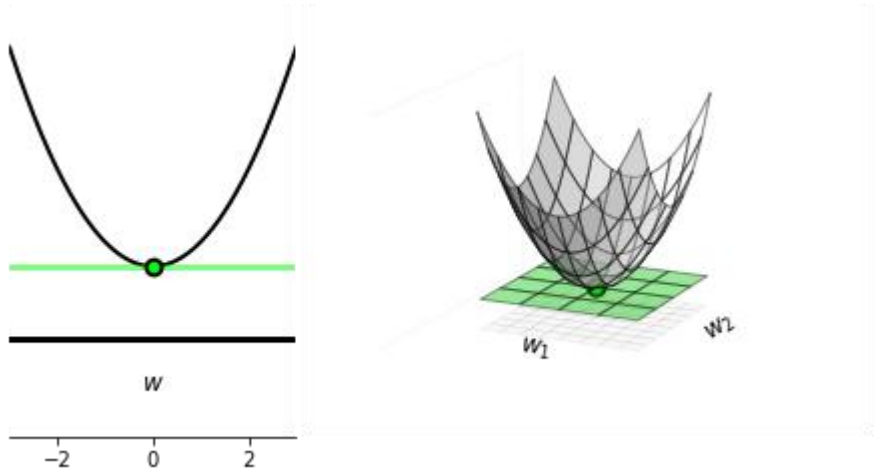
1. Điều kiện tối ưu bậc nhất
2. Phương pháp Gradient Descent
3. Các nhược điểm của Gradient Descent
4. Tăng tốc thuật toán Gradient Descent với Momentum
5. Gradient Descent được chuẩn hóa
6. Một số phương pháp Gradient Descent cải tiến khác

ĐIỀU KIỆN TỐI ƯU BẬC NHẤT

THE FIRST-ORDER OPTIMALITY CONDITION

Điều kiện bậc nhất

- Điều kiện tối ưu bậc nhất mô tả đặc điểm của đạo hàm bậc nhất của bất kỳ hàm khả vi nào tại điểm cực tiểu.



- Ta vẽ xấp xỉ bậc nhất (một đường tiếp tuyến (tangent line) hoặc siêu phẳng (hyperplane) tiếp tuyến) tại điểm cực tiểu của hàm số.
- Trong cả hai ví dụ, đường tiếp tuyến hoặc siêu phẳng tiếp tuyến đều hoàn toàn phẳng, cho thấy rằng đạo hàm bậc nhất tại điểm cực tiểu bằng đúng 0.

Điều kiện bậc nhất

- Giá trị của các đạo hàm bậc nhất giúp xác định các điểm cực tiểu của một hàm g.
- Khi $N = 1$, bất kỳ điểm v nào mà tại đó

$$\frac{d}{dw} g(v) = 0$$

có thể là một điểm cực tiểu.

- Khi $N > 1$, bất kỳ điểm $\boldsymbol{v}$ nào trong không gian N chiều mà tại đó tất cả các đạo hàm riêng của g đều bằng 0:

$$\frac{\partial}{\partial w_1} g(\boldsymbol{v}) = 0, \frac{\partial}{\partial w_2} g(\boldsymbol{v}) = 0, \ ..., \frac{\partial}{\partial w_N} g(\boldsymbol{v}) = 0$$

thì điểm đó có thể là một điểm cực tiểu. Hệ gồm N phương trình này thường được gọi là hệ phương trình bậc nhất, và có thể được viết gọn lại như sau:

$$\nabla g(\boldsymbol{v}) = \mathbf{0}_{N \times 1}$$

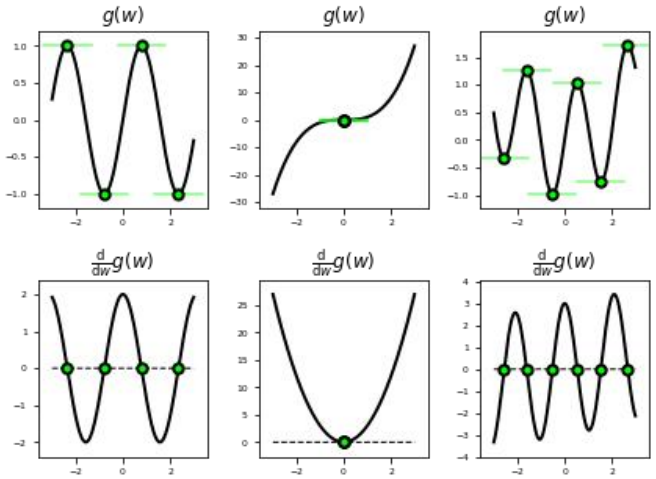
Điều kiện bậc nhất

- Điều kiện tối ưu bậc nhất chuyển bài toán tìm các điểm cực tiểu của một hàm thành bài toán giải một hệ gồm N phương trình đạo hàm bậc nhất.
- Tuy nhiên, có hai vấn đề:

➢ Hầu như không thể (ngoại trừ một số trường hợp đặc biệt) giải hệ phương trình bậc nhất của một hàm tổng quát để thu được nghiệm dạng đóng (closed-form solutions).

➢ Điều kiện tối ưu bậc nhất không chỉ tương ứng với các điểm cực tiểu (minima), mà còn cho cả các điểm cực đại (maxima) và điểm yên ngựa (saddle points) của hàm số.

Điều kiện bậc nhất



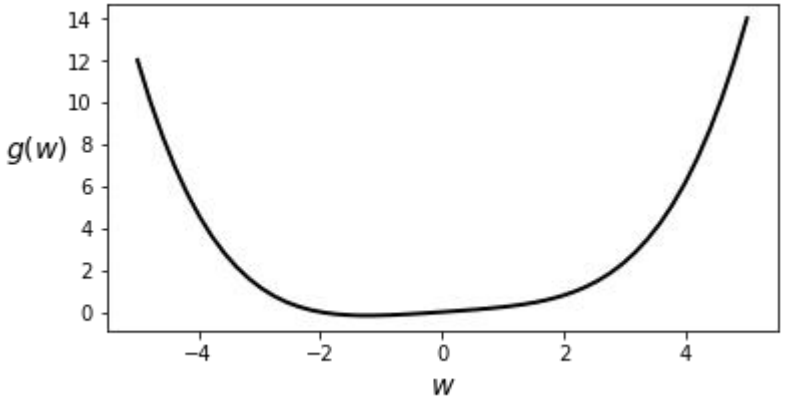
- Không chỉ các điểm cực tiểu toàn cục có đạo hàm bằng không, mà cả các điểm khác như cực tiểu cục bộ, cực đại cục bộ và toàn cục, cũng như các điểm yên ngựa, cũng có thể có đạo hàm bằng không. Các điểm mà tại đó đạo hàm (hoặc gradient) bằng không được gọi chung là các điểm dừng / điểm bất động (stationary points) hay điểm tới hạn (critical points).
- Điều kiện tối ưu bậc nhất phát biểu rằng: các điểm dừng của một hàm g (bao gồm cực tiểu, cực đại và điểm yên ngựa) phải thỏa mãn điều kiện bậc nhất:

$$\nabla g(\boldsymbol{v}) = \mathbf{0}_{N\times 1}$$

- Nếu một hàm là lồi (convex, ví dụ: hàm bậc hai), thì bất kỳ điểm nào thỏa mãn điều kiện bậc nhất đều là điểm cực tiểu toàn cục. Hàm lồi không có điểm cực đại cũng như điểm yên ngựa.

Ví dụ

$$g(w) = \frac{1}{50}(w^4 + w^2 + 10w)$$



$$\frac{d}{dw}g(w) = \frac{1}{50}(4w^3 + 2w + 10) = 0$$

Hàm số đạt cực tiểu tại:

$$w = \frac{\sqrt[3]{\sqrt{2031} - 45}}{6^{\frac{2}{3}}} - \frac{1}{\sqrt[3]{6\left(\sqrt{2031} - 45\right)}}$$

Ví dụ

- Xét một hàm bậc hai đa biến (multi-input quadratic function):

$$g(\boldsymbol{w}) = a + \boldsymbol{b}^T\boldsymbol{w} + \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{C}\boldsymbol{w}$$

với $\boldsymbol{C}$ là ma trận đối xứng $N \times N$, $\boldsymbol{b}$ là vector $N \times 1$, và a là giá trị vô hướng (scalar).

- Tính đạo hàm bậc nhất (gradient):

$$\nabla g(w) = 2\boldsymbol{C}\boldsymbol{w} + \boldsymbol{b}$$

- Đặt đạo hàm (gradient) bằng 0 cho ta một hệ các phương trình tuyến tính có dạng:

$$\boldsymbol{C}\boldsymbol{w} = -\frac{1}{2}\boldsymbol{b}$$

với nghiệm là các điểm dừng (stationary points) của hàm $g(\boldsymbol{w})$.

Thuật toán Coordinate Descent

- Xét hàm $g(\boldsymbol{w})$ có đầu vào là $\boldsymbol{w} \in \mathbb{R}^N$, các điểm dừng (bao gồm các điểm cực tiểu) của hàm g thỏa mãn: $\nabla g(\boldsymbol{v}) = \mathbf{0}_{N\times 1}$.
- Tức là:

$$\frac{\partial}{\partial w_1} g(\boldsymbol{v}) = 0$$

$$\frac{\partial}{\partial w_2} g(\boldsymbol{v}) = 0$$

$$\dots$$

$$\frac{\partial}{\partial w_N} g(\boldsymbol{v}) = 0$$

- Thay vì giải đồng thời tất cả các phương trình trên, ta lần lượt giải từng phương trình $\frac{\partial}{\partial w_n} g(\boldsymbol{v}) = 0$ nếu mỗi phương trình này có thể giải được một cách hiệu quả.

Thuật toán Coordinate Descent

- Ta bắt đầu với một lời giải khởi tạo $\boldsymbol{w}^0$ và tìm nghiệm của phương trình:

$$\frac{\partial}{\partial w_1} g(\boldsymbol{w}^0) = 0$$

tương ứng với giá trị w_1^*.

- Ta cập nhật tọa độ (coordinate) đầu tiên w_1 của vector $\boldsymbol{w}^0$ với giá trị w_1^* này, và gọi vector sau khi cập nhật là $\boldsymbol{w}^1$.
- Ta tiếp tục quy trình này để cập nhật tọa độ thứ n bằng cách giải:

$$\frac{\partial}{\partial w_n} g(\boldsymbol{w}^{n-1}) = 0$$

tương ứng với giá trị w_n^* để có được vector $\boldsymbol{w}^n$.

- Sau khi quét qua N tọa độ, ta có thể tiếp tục quay lại cập nhật lần lượt các tọa độ $w_1,\ldots, w_N$ lần nữa. Ở mỗi vòng lặp cập nhật thứ k, tọa độ thứ n được cập nhật với nghiệm của phương trình:

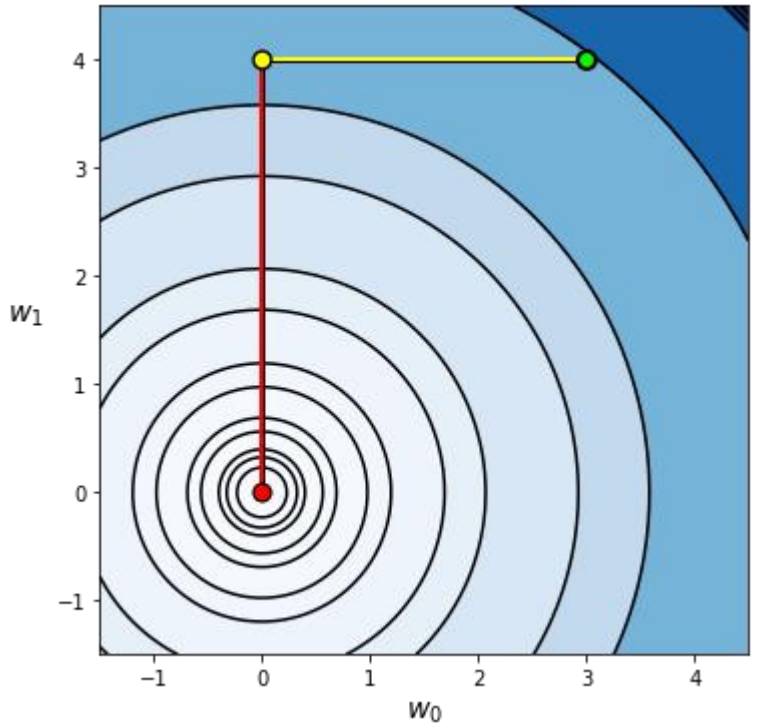
$$\frac{\partial}{\partial w_n} g(\boldsymbol{w}^{k+n-1}) = 0$$

Ví dụ

- Ta muốn cực tiểu hóa hàm bậc 2 sau đây: $g(w_0, w_1) = w_0^2 + w_1^2 + 2$.

$$g(w_0, w_1) = a + \boldsymbol{b}^T\boldsymbol{w} + \boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} = 2 + [0 \quad 0]\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} + [w_0 \quad w_1]\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

- Khởi tạo tại $\boldsymbol{w} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$ và chạy 1 vòng lặp cập nhật của Coordinate Descent.

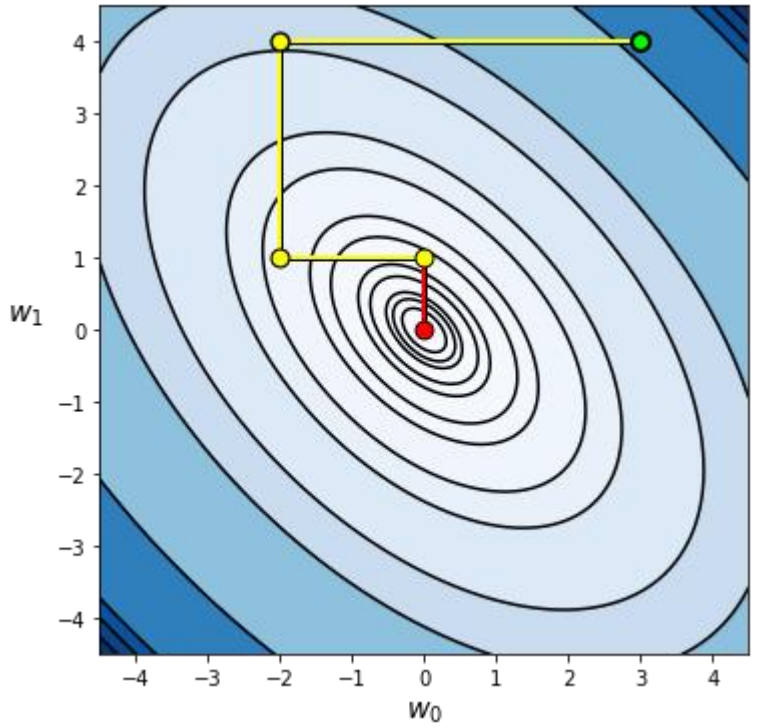


Ví dụ

- Ta muốn cực tiểu hóa hàm bậc 2 sau đây: $g(w_0, w_1)$

$$g(w_0, w_1) = a + \boldsymbol{b}^T\boldsymbol{w} + \boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} = 20 + [0 \quad 0]\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} + [w_0 \quad w_1]\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

- Khởi tạo tại $\boldsymbol{w} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$ và chạy 2 vòng lặp cập nhật của Coordinate Descent.



GRADIENT DESCENT

Cấu trúc các siêu phẳng (hyperplanes)

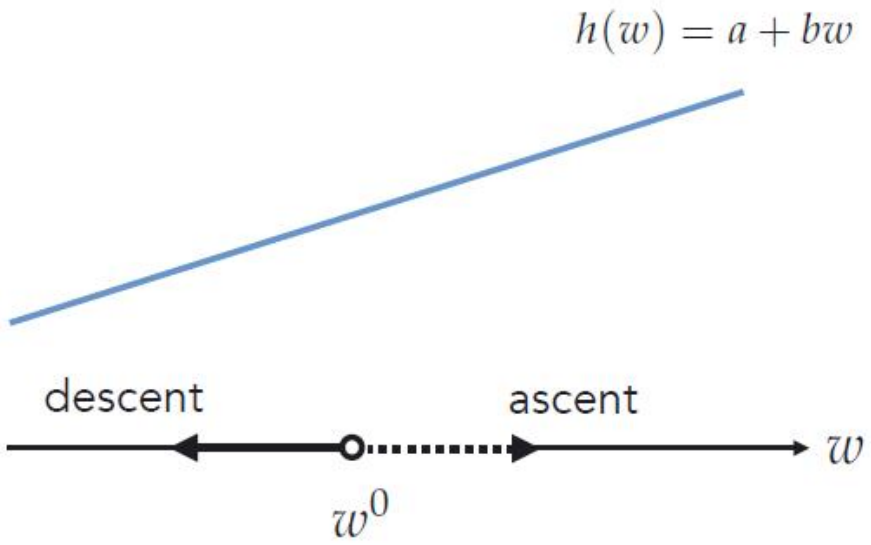
- Một siêu phẳng (hyperplane) tổng quát trong không gian N chiều được biểu diễn:

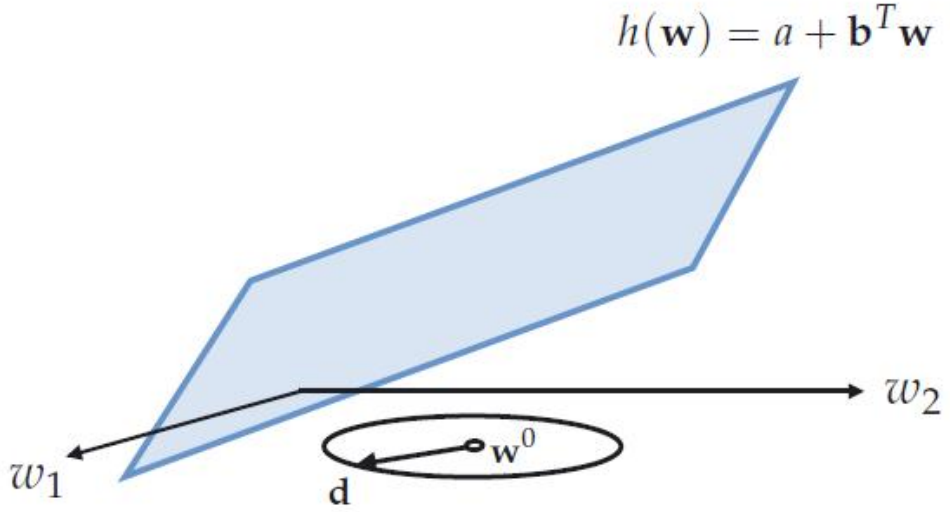
$$h(w_1, w_2, ..., w_N) = a + b_1 w_1 + b_2 w_2 + ... + b_N w_N$$

trong đó $a, b_1, b_2, ..., b_N$ là các tham số (scalar). Ta có thể viết gọn biểu thức h:

$$h(\boldsymbol{w}) = a + \boldsymbol{b}^T \boldsymbol{w}$$

- Khi $N = 1$, ta có $h(w) = a + bw$ là công thức của đường thẳng 1 chiều.
- Khi $N > 1$, ta có $h(\boldsymbol{w}) = a + \boldsymbol{b}^T \boldsymbol{w}$ là siêu phẳng N chiều.





Cấu trúc các siêu phẳng (hyperplanes)

- Ta muốn tìm hướng tăng nhanh nhất (steepest descent direction) tại một điểm $\boldsymbol{w}^0$:

$$\max_{\boldsymbol{d}} h(\boldsymbol{w}^0 + \boldsymbol{d})$$

với $\boldsymbol{d}$ là các vector đơn vị. Ta có:

$$h(\boldsymbol{w}^0 + \boldsymbol{d}) = a + \boldsymbol{b}^T(\boldsymbol{w}^0 + \boldsymbol{d}) = a + \boldsymbol{b}^T\boldsymbol{w}^0 + \boldsymbol{b}^T\boldsymbol{d}$$

- Hai thành phần đầu tiên không phụ thuộc $\boldsymbol{d}$, do đó: $\max_{\boldsymbol{d}} h(\boldsymbol{w}^0 + \boldsymbol{d}) \sim \max_{\boldsymbol{d}} \boldsymbol{b}^T\boldsymbol{d}$.
- Ta có $\boldsymbol{b}^T\boldsymbol{d} = \|\boldsymbol{b}\|_2\|\boldsymbol{d}\|_2 \cos(\theta)$ với$\|\boldsymbol{b}\|_2$ không thay đổi theo $\boldsymbol{d}$, và $\|\boldsymbol{d}\|_2 = \mathbf{1}$:

$$\max_{\boldsymbol{d}} h(\boldsymbol{w}^0 + \boldsymbol{d}) \sim \max_{\boldsymbol{d}} \boldsymbol{b}^T\boldsymbol{d} \sim \max_{\theta} \cos(\theta)$$

➢ Trong các vector đơn vị, $d = \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|_2}$ là hướng tăng nhanh nhất (steepest ascent) do $\theta = 0$.

➢ Trong các vector đơn vị, $d = \frac{-\boldsymbol{b}}{\|\boldsymbol{b}\|_2}$ là hướng giảm nhanh nhất (steepest descent) do $\theta = \pi$.

Gradient và hướng tăng/giảm nhanh nhất

- Một hàm số $g(\boldsymbol{w})$ có thể được xấp xỉ cục bộ (locally approximated) xung quanh một điểm $\boldsymbol{w}^0$ bởi một siêu phẳng (hyperplane) $h(\boldsymbol{w})$

$$h(\boldsymbol{w}) = g(\boldsymbol{w}^0) + \nabla g(\boldsymbol{w}^0)^T(\boldsymbol{w} - \boldsymbol{w}^0)$$

- Nếu biểu diễn dưới dạng $h(\boldsymbol{w}) = a + \boldsymbol{b}^T\boldsymbol{w}$ ta có:

$$a = g(\boldsymbol{w}^0) - \nabla g(\boldsymbol{w}^0)^T\boldsymbol{w}^0 \text{ và } \boldsymbol{b} = \nabla g(\boldsymbol{w}^0)$$

- Siêu phẳng này là xấp xỉ Taylor bậc nhất của $g(\boldsymbol{w})$ tại $\boldsymbol{w}^0$, và là tiếp tuyến với $g(\boldsymbol{w})$ tại điểm $\boldsymbol{w}^0$ này.
- Bởi vì $h(\boldsymbol{w})$ được xây dựng để xấp xỉ $g(\boldsymbol{w})$ ở những điểm xung quanh $\boldsymbol{w}^0$, hướng tăng/giảm nhanh nhất của h cho ta biết hướng dịch chuyển để tăng hoặc giảm giá trị hàm g tại $\boldsymbol{w}^0$. Đây chính là vector gradient $\boldsymbol{b} = \nabla g(\boldsymbol{w}^0)$.

Gradient Descent

- Các phương pháp tối ưu hóa cục bộ (local optimization methods) tìm kiếm lời giải cực tiểu của một hàm số $g(\boldsymbol{w})$ bằng cách bắt đầu từ một lời giải khởi tạo $\boldsymbol{w}^0$, và lần lượt đi từng bước $\boldsymbol{w}^1$, $\boldsymbol{w}^2$, … như sau:

$$\boldsymbol{w}^k = \boldsymbol{w}^{k-1} + \alpha \boldsymbol{d}^k$$

với:

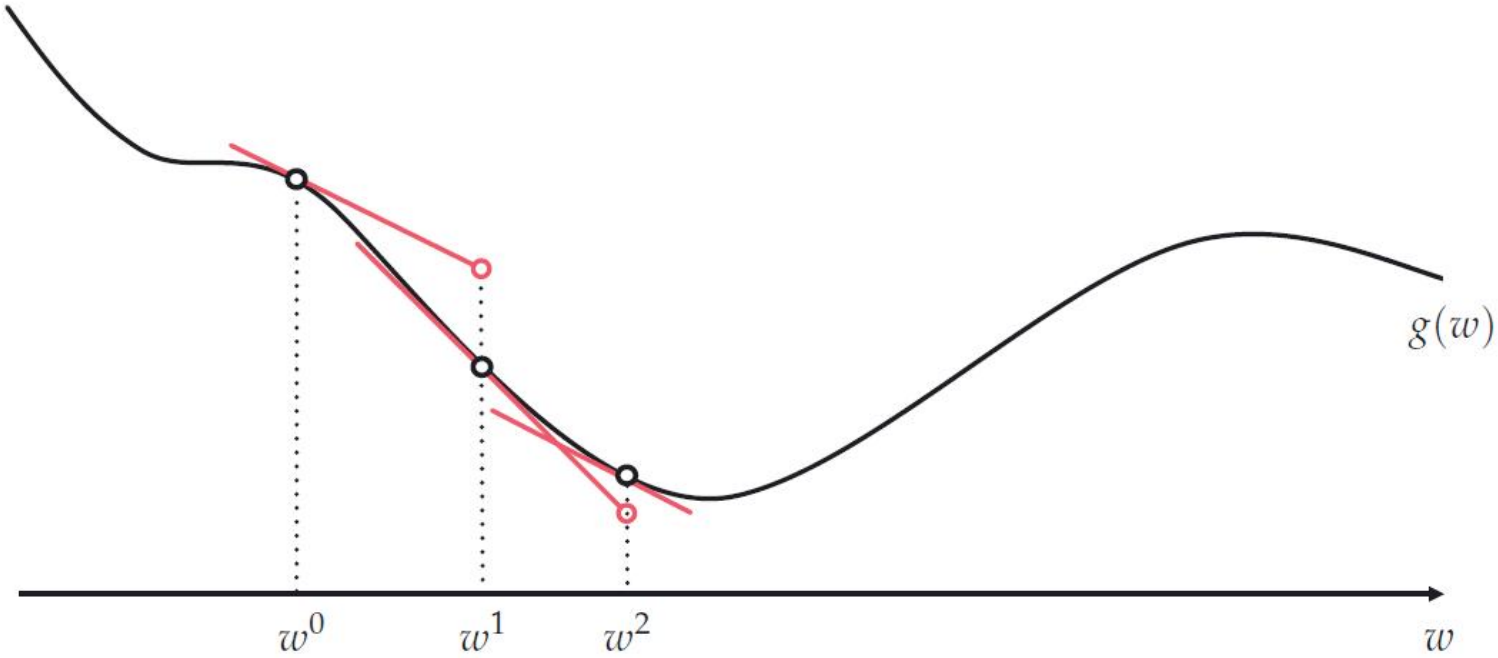
- $\boldsymbol{d}^k$ là các hướng giảm (descent direction)
- α là tham số độ dài bước (steplength parameter)

- Gradient âm (negative gradient) $-\nabla g(\boldsymbol{w})$ của hàm $g(\boldsymbol{w})$ tại một điểm nào đó cho ta một hướng giảm (descent direction) tại vị trí đó.
- Nếu ta chọn $d^k = -\nabla g(\boldsymbol{w}^{k-1})$, dãy các bước đi của thuật toán sẽ có dạng:

$$\boldsymbol{w}^k = \boldsymbol{w}^{k-1} - \alpha \nabla g(\boldsymbol{w}^{k-1})$$

- Phương pháp tối ưu hóa cục bộ với bước cập nhật (update step) như trên được gọi là thuật toán Gradient Descent.

Gradient Descent



- Bắt đầu từ w^0, ta thực hiện xấp xỉ $g(w)$ tại điểm $(w^0, g(w^0))$ với xấp xỉ Taylor bậc nhất.
- Cập nhật theo hướng gradient âm của phép xấp xỉ trên, ta có:

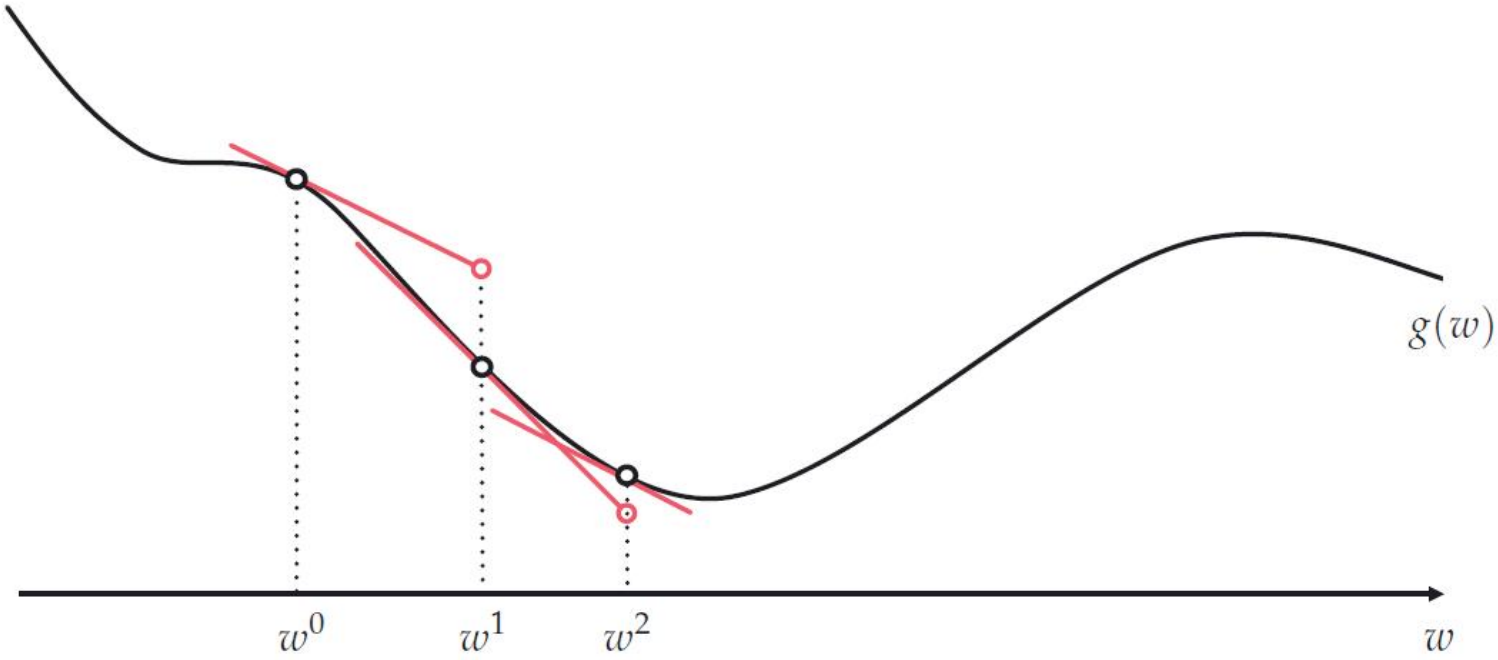
$$w^1 = w^0 - \alpha \frac{d}{dw} g(w^0)$$

- Lặp lại quy trình trên tại w^1, cập nhật theo hướng gradient âm:

$$w^2 = w^1 - \alpha \frac{d}{dw} g(w^1)$$

- …

Gradient Descent



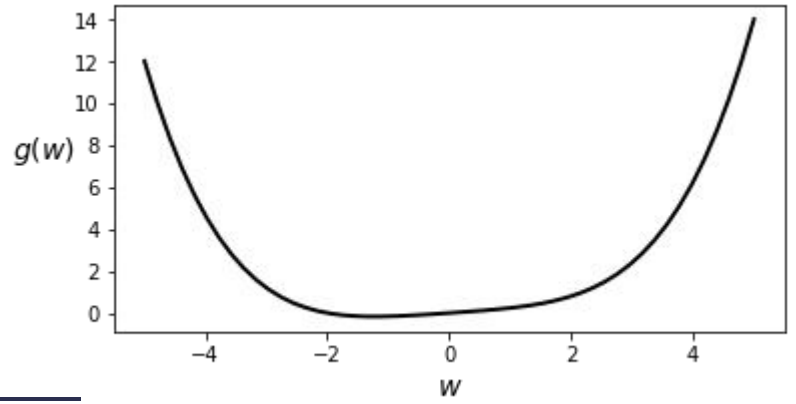
- ***Khi nào thuật toán Gradient Descent có thể dừng?***
- Nếu độ dài bước (steplength) α được lựa chọn phù hợp, thuật toán sẽ dùng tại các điểm dừng / điểm bất động (stationary points) của hàm số, thường là các cực tiểu hoặc điểm yên ngựa.
- Nếu bước cập nhật

$$\boldsymbol{w}^k = \boldsymbol{w}^{k-1} - \alpha \nabla g(\boldsymbol{w}^{k-1})$$

không thay đổi đáng kể giữa $\boldsymbol{w}^{k-1}$ và $\boldsymbol{w}^k$ thì nghĩa là hướng dịch chuyển đang bị triệt tiêu (vanishing), tức là $\nabla g(\boldsymbol{w}^{k-1}) \approx \boldsymbol{0}_{N\times 1}$. Thuật toán đang tiệm cận một điểm dừng (stationary point) nào đó của hàm số $g(\boldsymbol{w})$.

Ví dụ

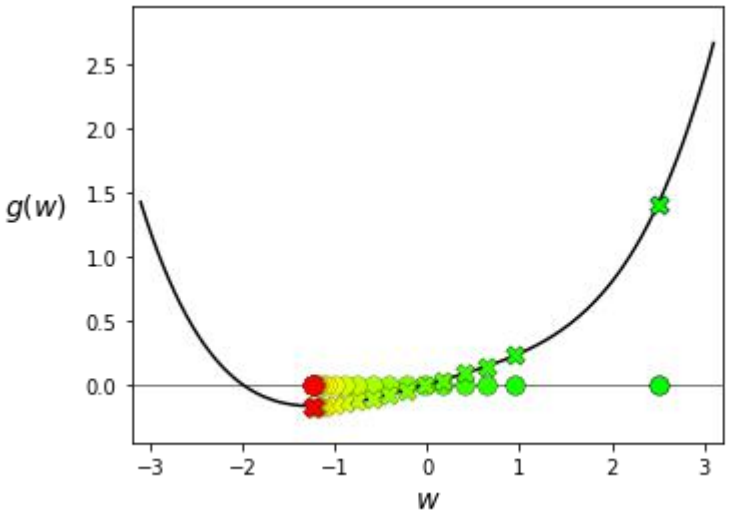
$$g(w) = \frac{1}{50}(w^4 + w^2 + 10w)$$

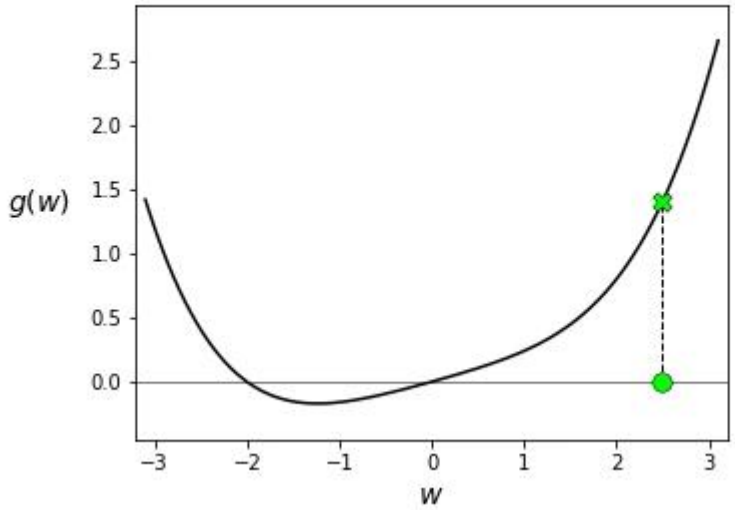


- Hàm số đạt cực tiểu tại:

$$w = \frac{\sqrt[3]{\sqrt{2031} - 45}}{6^{\frac{2}{3}}} - \frac{1}{\sqrt[3]{6\left(\sqrt{2031} - 45\right)}}$$

- Chạy thuật toán Gradient Descent 25 bước với khởi tao tai $w^0 = 2.5$, đô dài bước $\alpha = 1$.

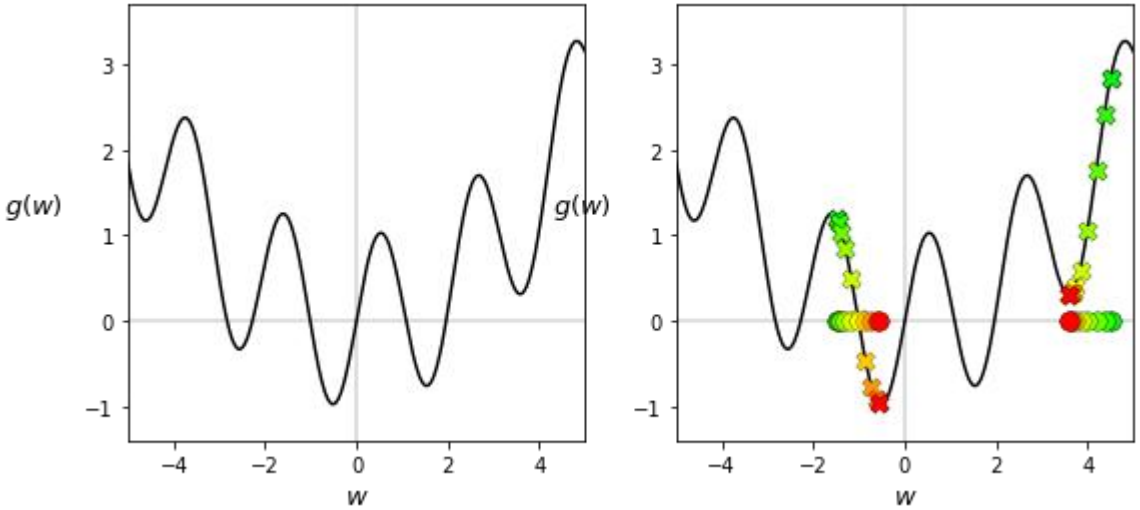




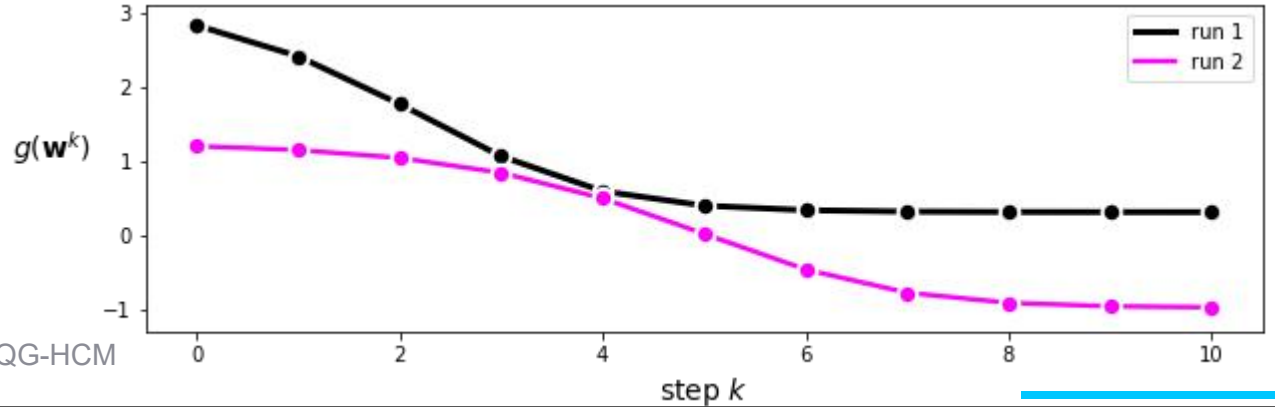
Ví dụ

$$g(w) = \sin(3w) + 0.1w^2$$

- Chạy 2 lượt Gradient Descent tại 2 vị trí khởi tạo $w^0 = 4.5$ và $w^0 = -1.5$.
- Ta sử dụng độ dài bước $\alpha = 0.05$ với 10 bước cập nhật.



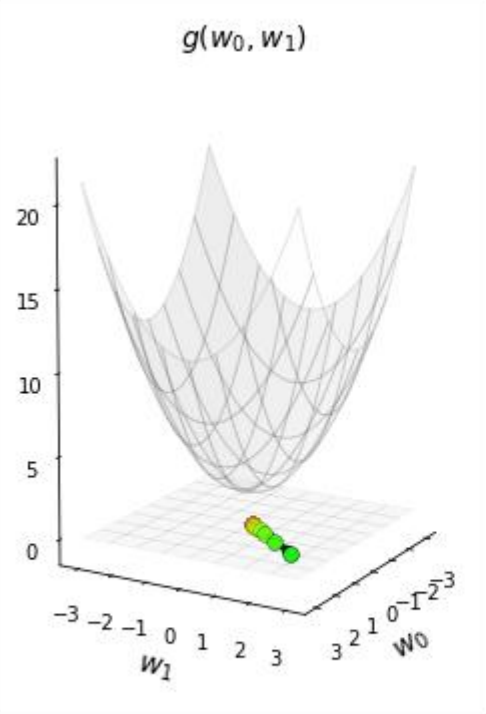
- Tùy thuộc vị trí khởi tạo mà ta thu được các lời giải cực trị khác nhau.

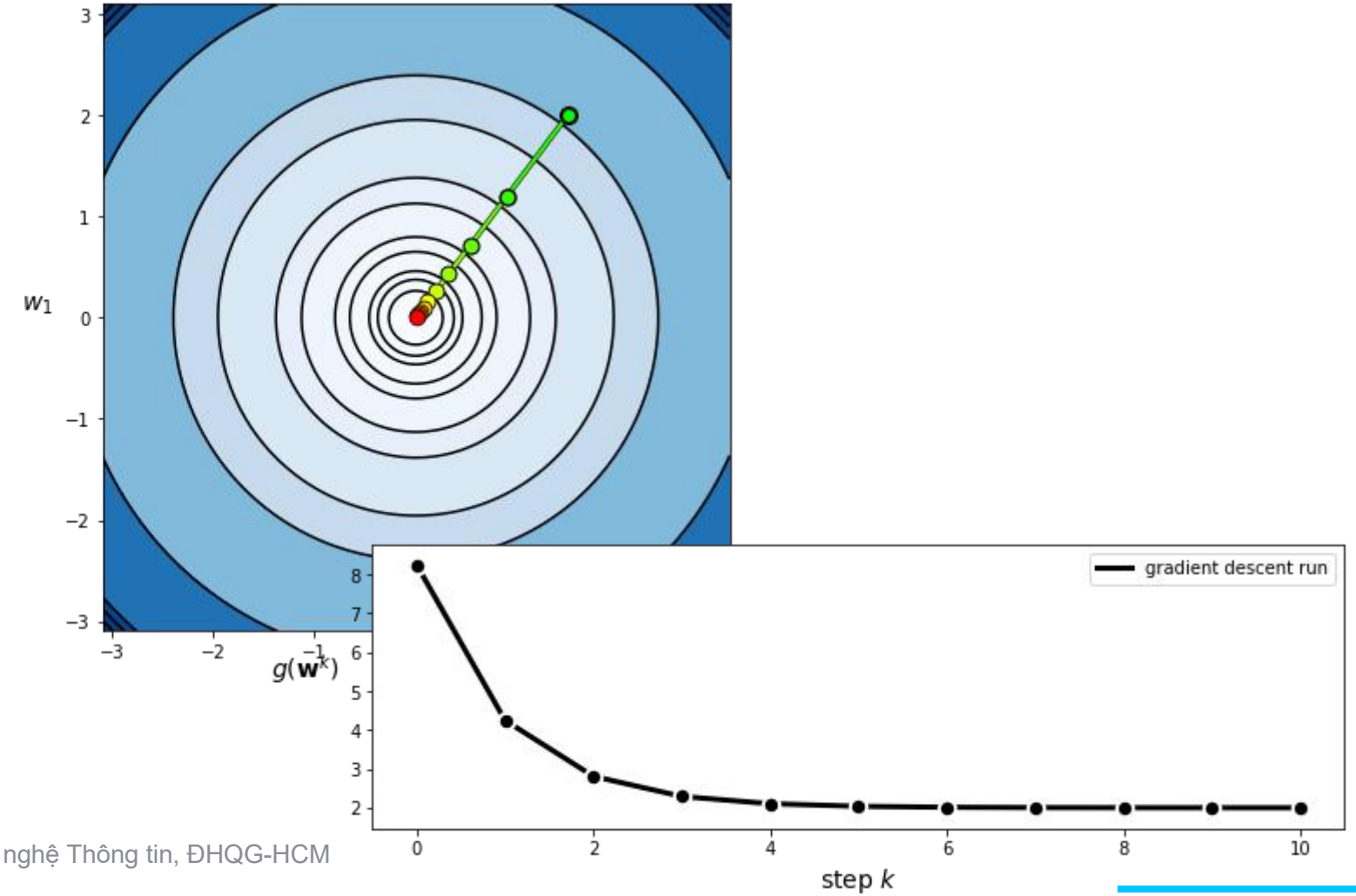


Ví dụ

$$g(w_1, w_2) = w_1^2 + w_2^2 + 2$$

- Chạy 10 bước Gradient Descent với độ dài bước $\alpha = 0.1$





Lựa chọn độ dài bước (Steplength choices)

- Độ dài bước (steplength), hay còn gọi là hệ số học (learning rate), là tham số quan trọng cần được lựa chọn phù hợp với bài toán.
- Hai lựa chọn phổ biến:

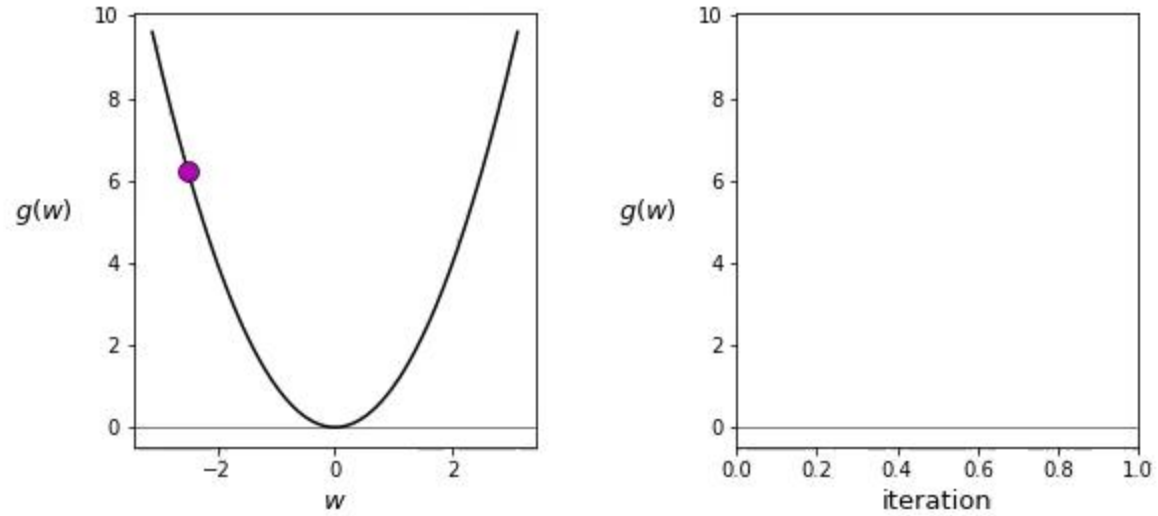
➢ Sử dụng một giá trị α cố định: $\alpha = 10^{\gamma}$ với γ thường là các số nguyên âm.

➢ Sử dụng độ dài bước giảm dần (diminishing steplenght): ví dụ: $\alpha = \frac{1}{k}$ cho bước thứ k trong một lượt chạy.

- Ta muốn lựa chọn giá trị độ dài bước α lớn nhất có thể sao đạt được tốc độ hội tụ hợp lý.

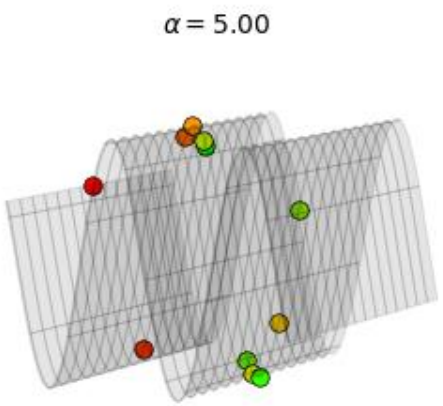
Ví dụ

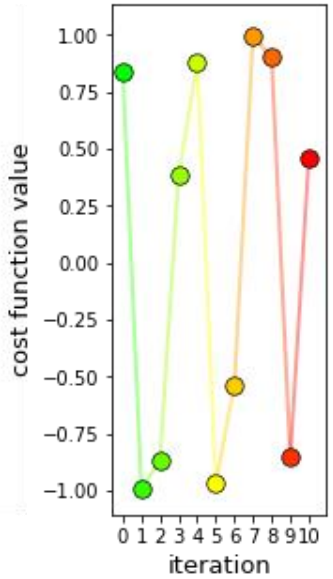
$$g(w) = w^2$$

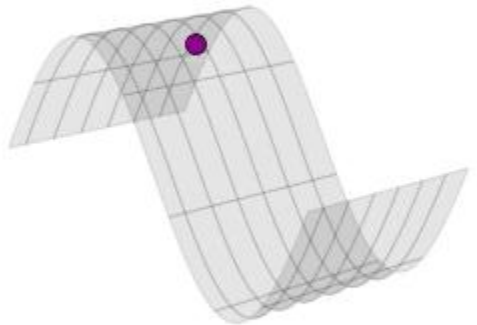


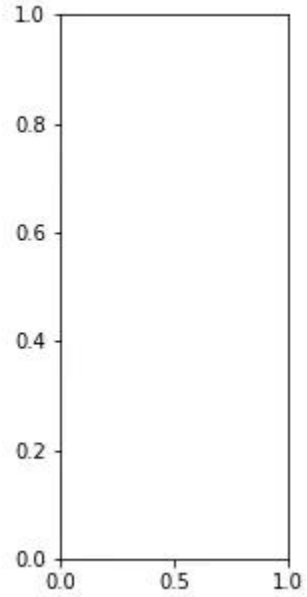
Ví dụ

$$g(w_1, w_2) = \sin(w_1)$$









Ví dụ

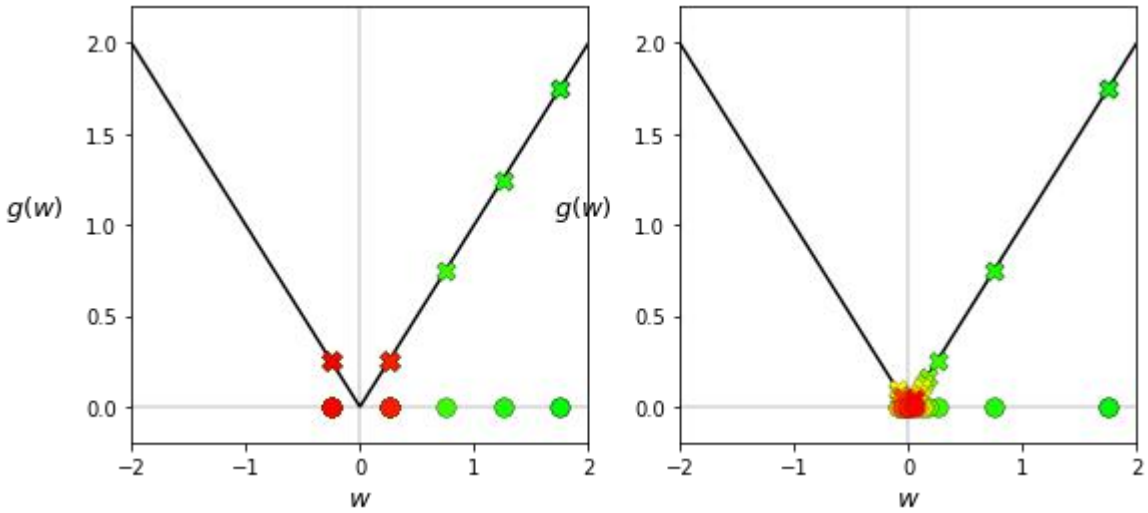
$$g(w) = |w|$$

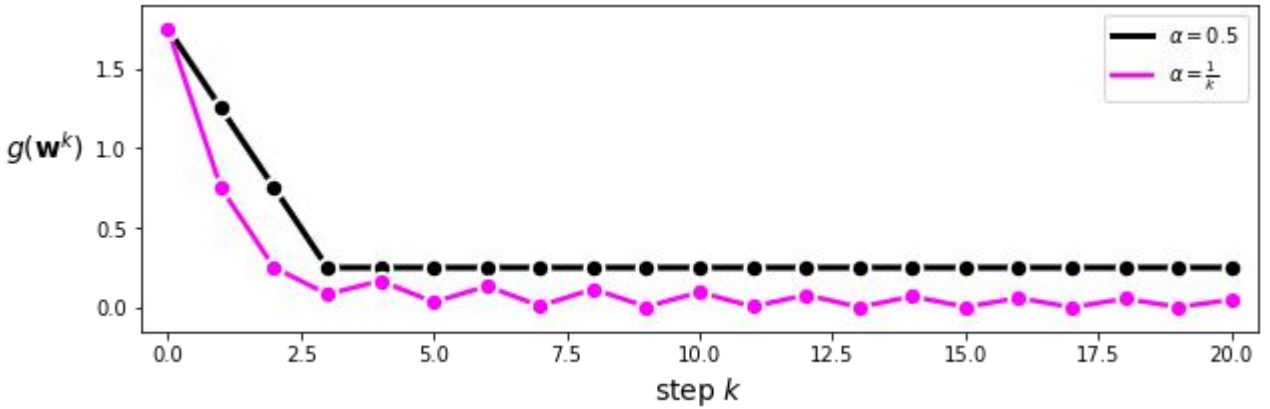
Hàm số này có cực tiểu toàn cục (global minimum) duy nhất tại $w = 0$, và đạo hàm là:

$$\frac{d}{dw} g(w) = \begin{cases} +1, & w > 0 \\ -1, & w < 0 \end{cases}$$

Ta chạy 2 lượt chạy Gradient Descent khởi tạo tại $w^0 = 2$, mỗi lượt gồm 20 bước cập nhật với:

- Độ dài bước cố định $\alpha = 0.5$
- Độ dài bước giảm dần $\alpha = \frac{1}{k}$



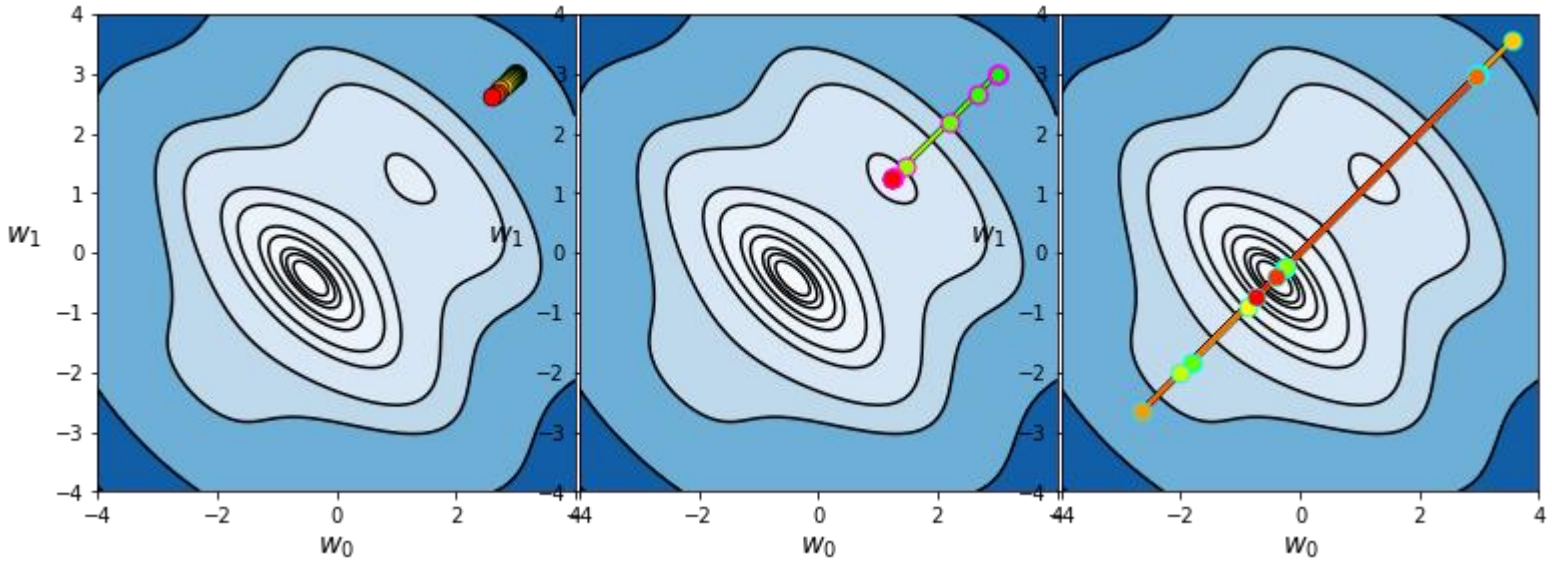


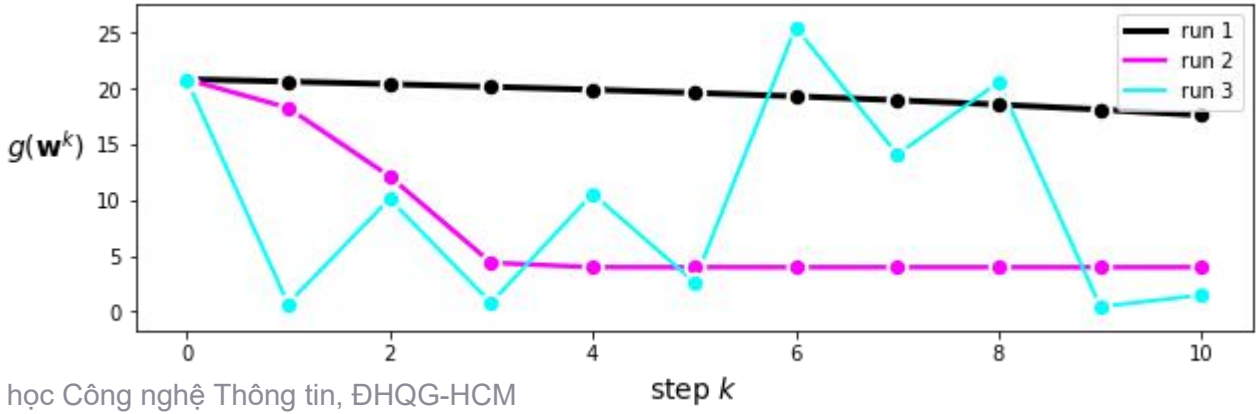
Ví dụ

$$g(\boldsymbol{w}) = w_0^2 + w_1^2 + 2\sin\left(1.5(w_0 + w_1)\right)^2 + 2$$

Ta chạy 3 lượt chạy Gradient Descent khởi tạo tại $\boldsymbol{w}^0 = (3,3)$, mỗi lượt gồm 10 bước đi với:

1. Độ dài bước $\alpha = 10^{-2}$
2. Độ dài bước $\alpha = 10^{-1}$
3. Độ dài bước $\alpha = 10^{0}$





Gradient Descent

- ***Khi nào thuật toán Gradient Descent có thể dừng?***

Nếu bước cập nhật

$$\boldsymbol{w}^k = \boldsymbol{w}^{k-1} - \alpha \nabla g(\boldsymbol{w}^{k-1})$$

không thay đổi đáng kể giữa $\boldsymbol{w}^{k-1}$ và $\boldsymbol{w}^k$ thì nghĩa là hướng dịch chuyển đang bị triệt tiêu (vanishing), tức là $\nabla g(\boldsymbol{w}^{k-1}) \approx \boldsymbol{0}_{N\times 1}$. Thuật toán đang tiệm cận một điểm dừng (stationary point) nào đó của hàm số $g(\boldsymbol{w})$.

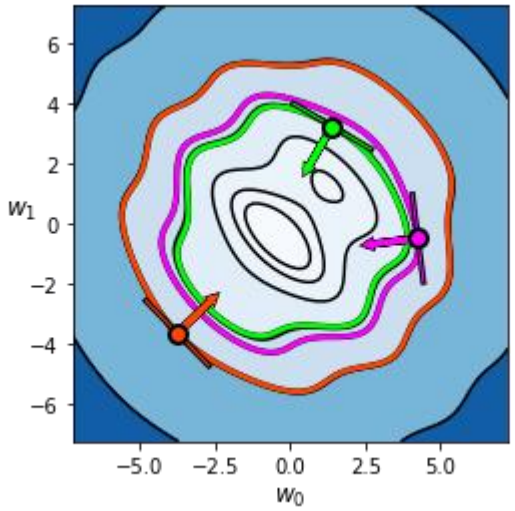
- Khi thuật toán tiệm cận rất gần một điểm dừng: $\left\|\nabla g\left(\boldsymbol{w}^{k-1}\right)\right\|_2$ đủ nhỏ
- Khi các bước cập nhật không có tiến triển đáng kể: $\frac{1}{N}\left\|\boldsymbol{w}^k - \boldsymbol{w}^{k-1}\right\|_2 < \varepsilon$
- Khi hàm mục tiêu tối ưu hóa không có thay đổi đáng kể: $\left|g\left(\boldsymbol{w}^k\right) - g(\boldsymbol{w}^{k-1})\right| < \varepsilon$
- Trong thực tế, ta thường thiết lập để Gradient Descent dừng lại sau một số hữu hạn bước cập nhật (maximum number of iterations).

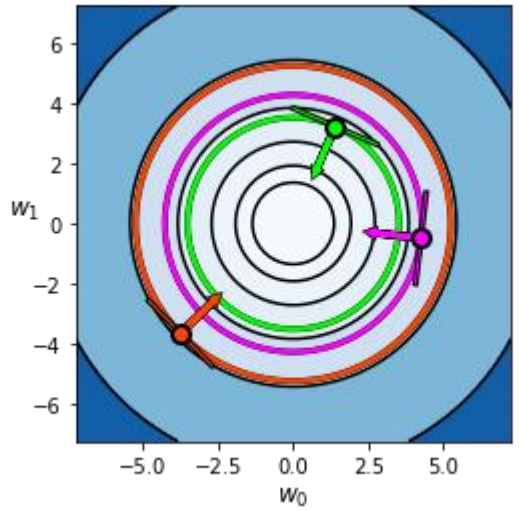
NHƯỢC ĐIỂM CỦA GRADIENT DESCENT

WEAKNESSES OF GRADIENT DESCENT

Gradient vuông góc các đường đồng mức

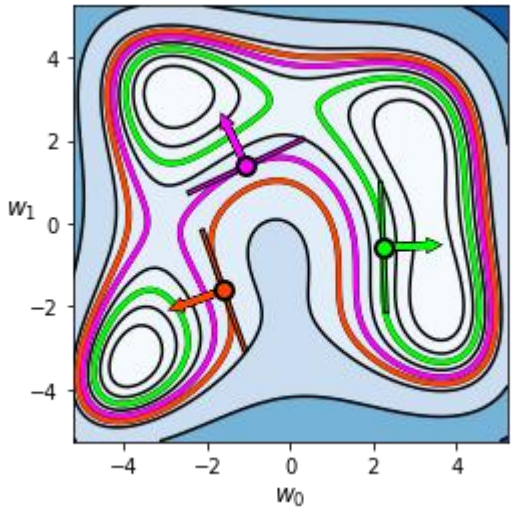
$$g(\boldsymbol{w}) = w_0^2 + w_1^2 + 2$$





$$g(\boldsymbol{w}) = w_0^2 + w_1^2 + 2\sin\left(1.5(w_0 + w_1)\right)^2 + 2$$

$$g(\boldsymbol{w}) = (w_0^2 + w_1 - 11)^2 + (w_0 + w_1^2 - 7)^2$$



Gradient vuông góc các đường đồng mức

- Gọi $g(\boldsymbol{w})$ là một hàm khả vi và $\boldsymbol{a}$ là một điểm đầu vào.
- $\boldsymbol{a}$ nằm trên một đường đồng mức (contour) định nghĩa bởi tất cả các điểm mà $g(\boldsymbol{w}) = g(\boldsymbol{a}) = c$ với c là một hằng số.
- Nếu ta chọn một điểm $\boldsymbol{b}$ nằm trên đường đồng mức này và $\boldsymbol{b}$ rất gần với $\boldsymbol{a}$ thì ta có vector $\boldsymbol{a} - \boldsymbol{b}$ có thể xem như vuông góc với gradient $\nabla g(\boldsymbol{a})$ bởi vì

$$\nabla g(\boldsymbol{a})^T(\boldsymbol{a} - \boldsymbol{b}) = 0$$

định nghĩa đường thẳng trong không gian đầu vào mà vector pháp tuyến chính là $\nabla g(\boldsymbol{a})$.

- Do đó, các hướng tăng/giảm (ascent/descent directions) định nghĩa bởi gradient $\nabla g(\boldsymbol{a})$ thì vuông góc với đường đồng mức của $\boldsymbol{a}$.
- Điều này thỏa với mọi điểm đầu vào $\boldsymbol{a}$.

Hiện tượng zig-zag của Gradient Descent

- Trong thực tế, việc gradient âm luôn hướng vuông góc với các đường đồng mức của một hàm có thể khiến hướng gradient âm dao động nhanh hoặc chuyển động kiểu "zig-zag" trong quá trình thực hiện Gradient Descent.
- Điều này dẫn đến chính các bước cập nhật của Gradient Descent cũng biểu hiện hành vi zig-zag.
- Việc zig-zag quá nhiều sẽ làm chậm quá trình tối thiểu hóa, khiến cần nhiều bước Gradient Descent hơn để tối ưu hóa hàm mục tiêu.

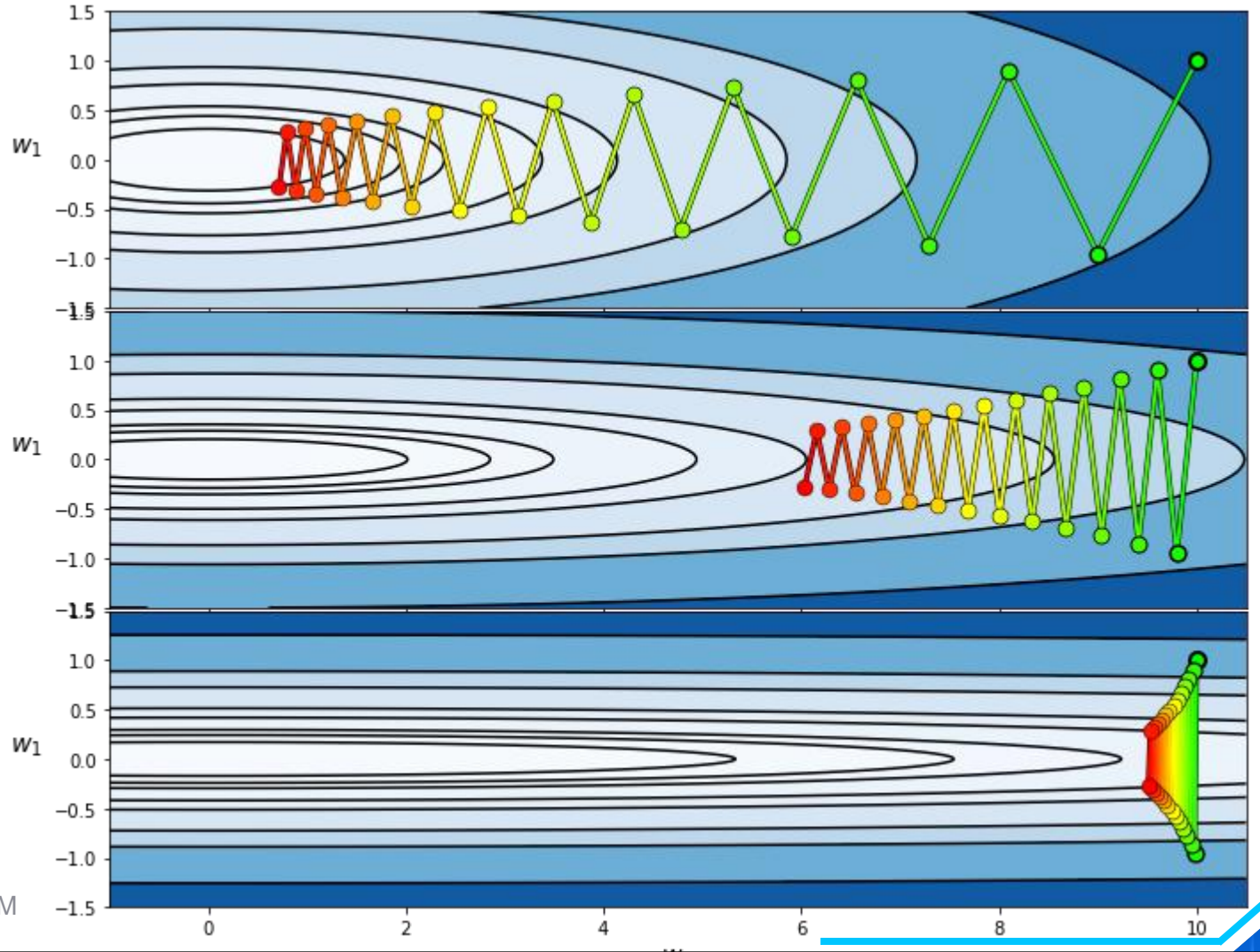
Ví dụ

- Sử dụng Gradient Descent với $\alpha = 0.1$ cực tiểu hóa 3 hàm bậc hai có dạng:

$$g(w) = a + \boldsymbol{b}^T\boldsymbol{w} + \boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w}$$

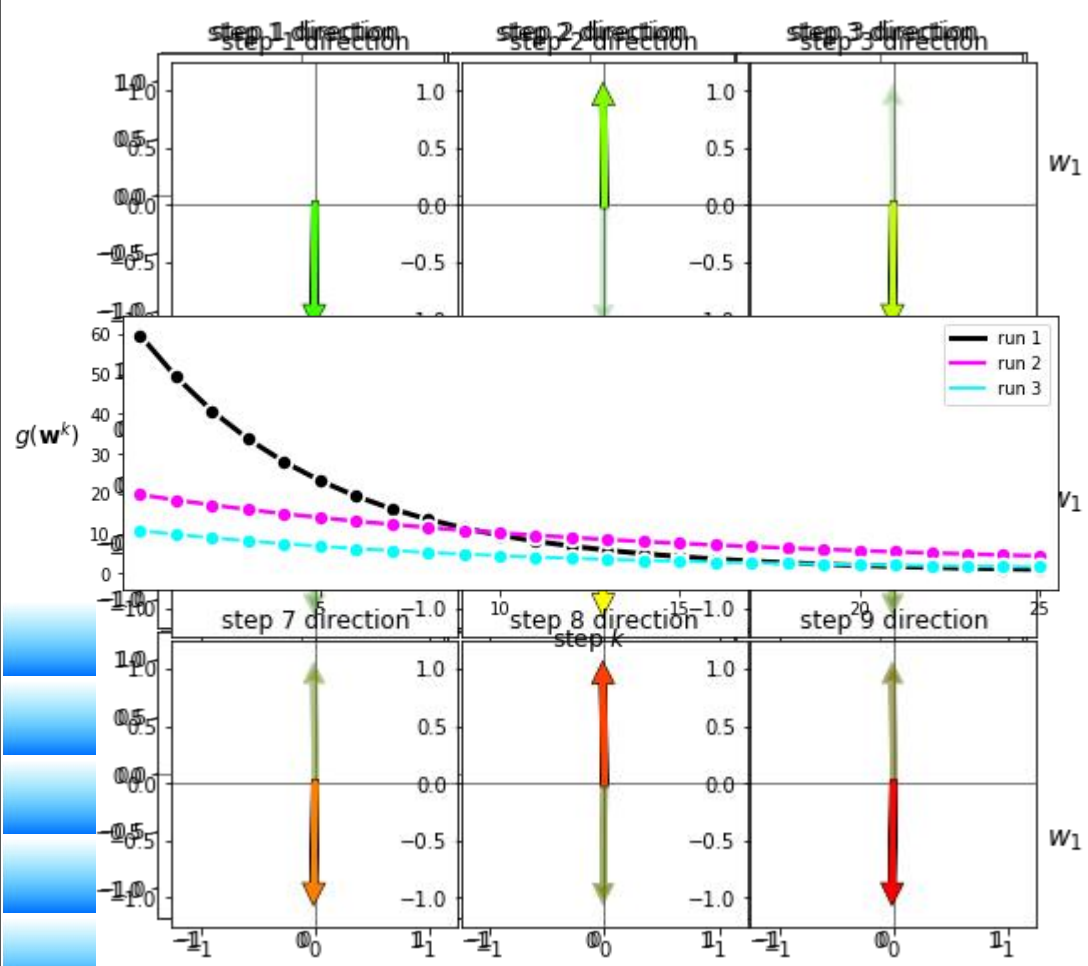
Với $a = 0$, $\boldsymbol{b} = (0,0)$, và $\boldsymbol{C}$ là

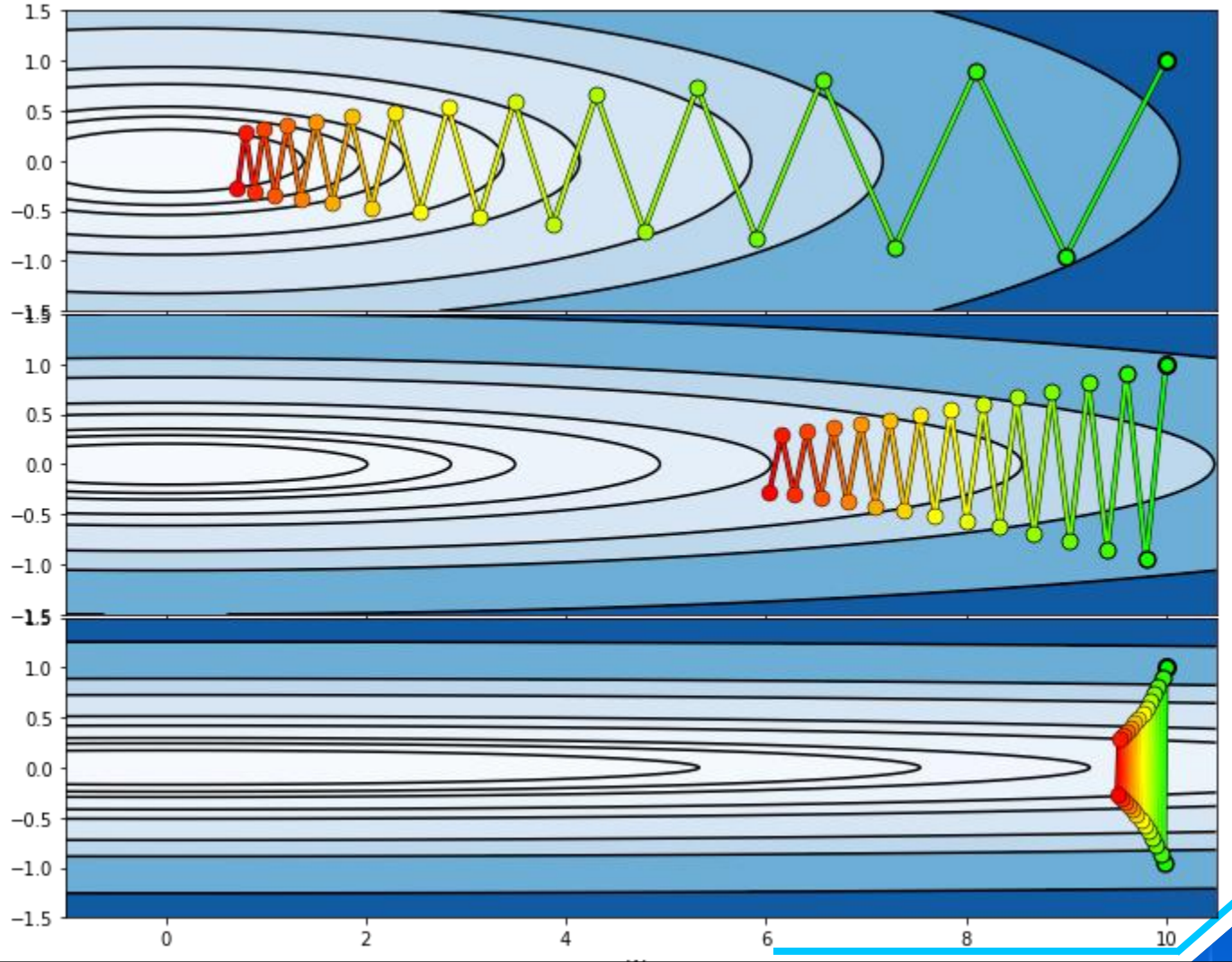
- $\boldsymbol{C} = \begin{bmatrix} 0.5 & 0 \\ 0 & 12 \end{bmatrix}$
- $\boldsymbol{C} = \begin{bmatrix} 0.1 & 0 \\ 0 & 12 \end{bmatrix}$
- $\boldsymbol{C} = \begin{bmatrix} 0.01 & 0 \\ 0 & 12 \end{bmatrix}$



Ví dụ

- Hành vi zig-zag của gradient descent xuất phát từ sự thay đổi nhanh chóng của hướng gradient âm trong mỗi bước của thuật toán.





Gradient Descent triệt tiêu gần các điểm dừng

- Từ điều kiện tối ưu bậc nhất, ta biết gradient (âm) sẽ triệt tiêu tại các điểm dừng. Nghĩa là, nếu $\boldsymbol{w}$ là một điểm cực tiểu, cực đại hoặc điểm yên ngựa thì ta có $\nabla g(\boldsymbol{w}) = \boldsymbol{0}$.
- Độ lớn của gradient triệt tiêu tại các điểm dừng: $\|\nabla g(\boldsymbol{w})\|_2 = 0$.
- Gradient (âm) tại các điểm gần một điểm dừng sẽ có độ lớn rất nhỏ $\|\nabla g(\boldsymbol{w})\|_2 \approx 0$. Do độ lớn của gradient tiệm cận về 0 gần các điểm dừng, các bước của Gradient Descent tiến rất chậm khi ở gần các điểm này.
- Tại mỗi bước của một thuật toán tối ưu hóa cục bộ:

$$\boldsymbol{w}^k = \boldsymbol{w}^{k-1} + \alpha \boldsymbol{d}^k$$

nếu $\boldsymbol{d}^k$ là một vector đơn vị thì khoảng cách của bước cập nhật chính là

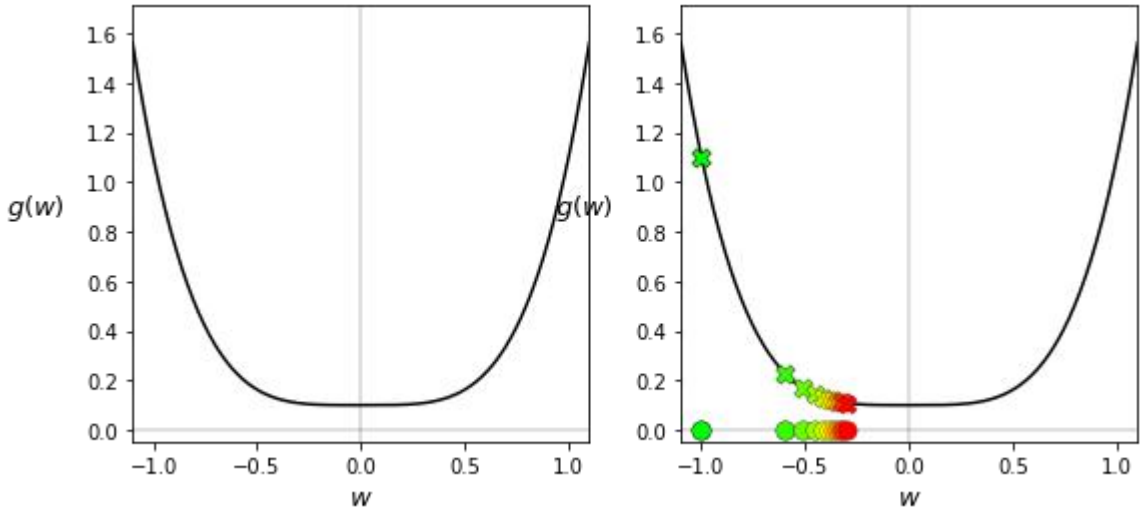
$$\left\|\boldsymbol{w}^k - \boldsymbol{w}^{k-1}\right\|_2 = \|\boldsymbol{w}^{k-1} + \alpha \boldsymbol{d}^k - \boldsymbol{w}^{k-1}\|_2 = \left\|\alpha \boldsymbol{d}^k\right\|_2 = \alpha$$

- Tuy nhiên, đối với Gradient Descent, hướng giảm $\boldsymbol{d}^k = -\nabla g(\boldsymbol{w}^{k-1})$ không đảm bảo có độ dài là 1, nên khoảng cách của bước cập nhật tỉ lệ thuận với độ lớn gradient $\alpha\left\|\nabla g(\boldsymbol{w}^{k-1})\right\|_2$.

Gradient Descent triệt tiêu gần các điểm dừng

$$g(w) = w^4 + 0.1$$

- Sử dụng Gradient Descent với $\alpha = 0.1$ và khởi tạo xa vị trí cực trị.

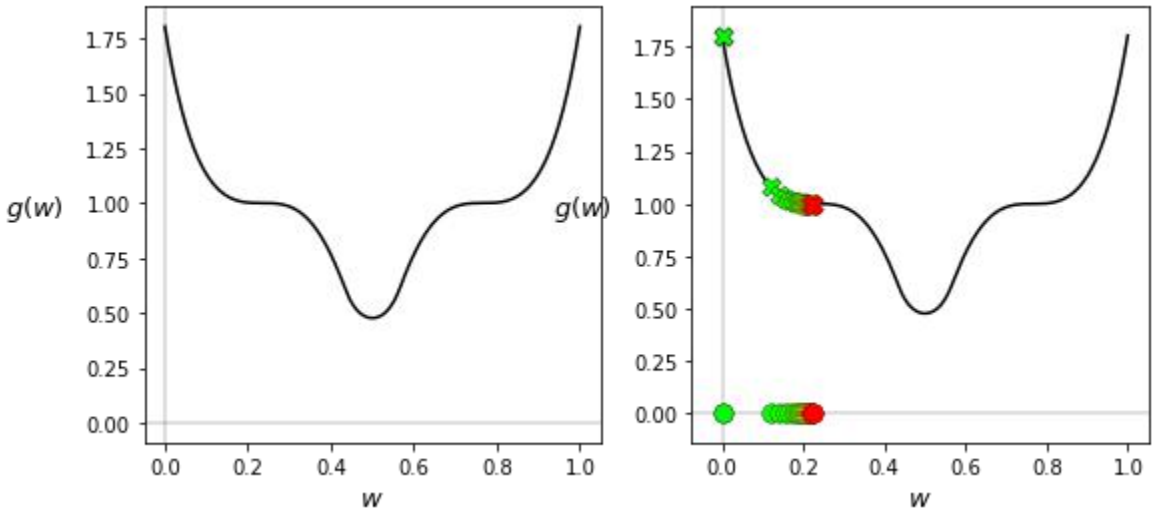


- Gradient Descent tiến rất chậm khi tiến gần đến điểm cực tiểu vì độ lớn của gradient tiệm cận về 0 tại đó.

Gradient Descent triệt tiêu gần các điểm dừng

$$g(w) = \max(0, (3w - 2.3)^3 + 1)^2 + \max(0, (-3w + 0.7)^3 + 1)^2$$

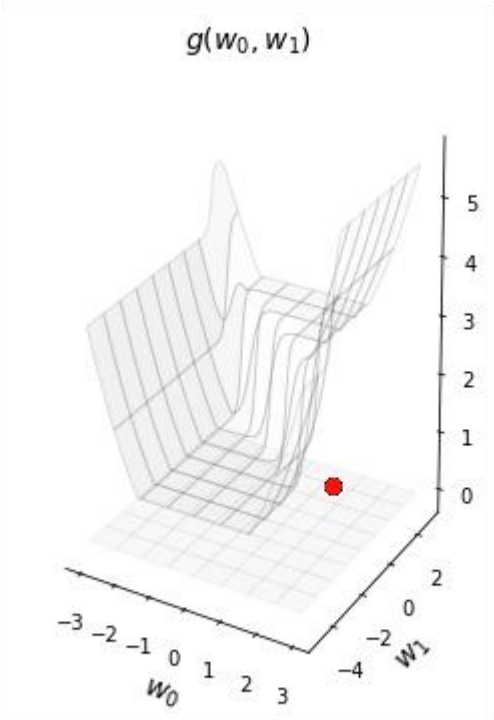
- Có cực tiểu tại $w = \frac{1}{2}$ và các điểm yên ngựa tại $w = \frac{7}{30}$ và $w = \frac{23}{30}$.
- Thực hiện Gradient Descent với 50 bước cập nhật và $\alpha = 0.01$.

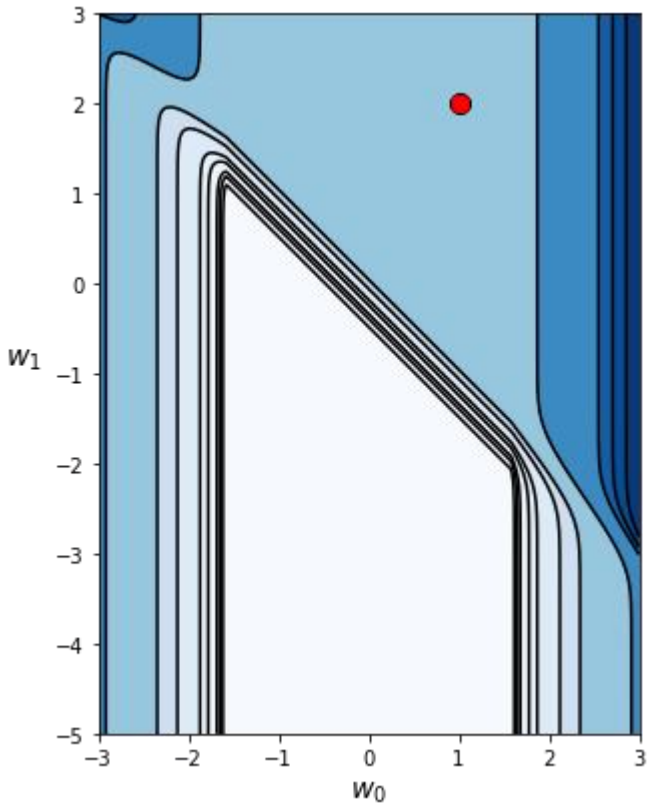


Gradient Descent triệt tiêu gần các điểm dừng

$$g(w) = \tanh(4w_0 + 4w_1) + \max(1,\ 0.4w_0^2) + 1$$

- Khởi tạo Gradient Descent tại điểm $w^0 = (1,2)$.
- Thực hiện 1000 bước cập nhật với $\alpha = 0.1$.



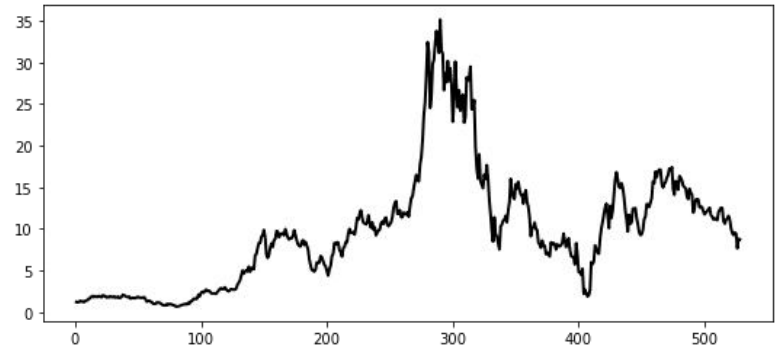


GRADIENT DESCENT VỚI MOMENTUM

MOMENTUM-ACCELERATED GRADIENT DESCENT

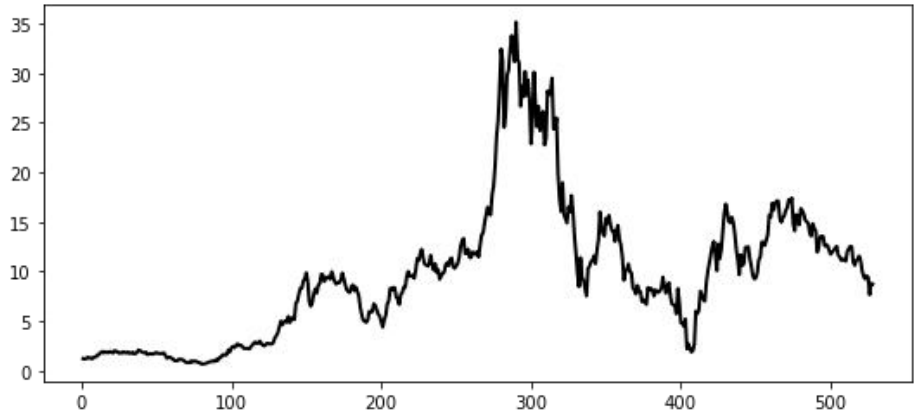
Gradient Descent với momentum

- Một vấn đề với hướng của gradient âm: tùy thuộc vào hàm mục tiêu đang được tối thiểu hóa, gradient có thể dao động nhanh chóng, dẫn đến các bước gradient descent đi theo đường zig-zag và làm chậm quá trình tối ưu.
- Một cải tiến để giải quyết hiện tượng zig-zag này trong phương pháp Gradient Descent tiêu chuẩn là gia tốc động lượng (momentum acceleration).
- Ý tưởng cốt lõi đến từ một công cụ dùng để làm trơn dữ liệu chuỗi thời gian (time series data), được gọi là trung bình lũy thừa / trung bình hàm mũ (exponential average). Các giá trị thô của một chuỗi thời gian thường dao động lên xuống bất thường; do đó, người ta thường làm trơn chúng để dễ trực quan hóa hoặc phân tích.
- Ví dụ: dữ liệu chuỗi thời gian là một dãy gồm K điểm dữ liệu có thứ tự $w^1, w^2, ..., w^K$. Giả sử ta có lịch sử giá cổ phiếu tại $K = 528$ thời điểm.



Trung bình tích lũy (cumulative average)

- Ta xét cách tính trung bình tích lũy của K điểm đầu vào $w^1, w^2, ..., w^K$. Ta tính trung bình của hai điểm đầu tiên, trung bình của ba điểm đầu tiên, và cứ thế tiếp tục.
- $h^1 = w^1$
- $h^2 = \frac{w^1+w^2}{2}$
- $h^3 = \frac{w^1+w^2+w^3}{3}$
- …
- $h^k = \frac{w^1+w^2+w^3+....+w^k}{k}$



- Ở mỗi bước, h^k tính giá trị trung bình dựa trên các điểm dữ liệu $w^1, w^2, ..., w^k$.
- Ta có thể tính trung bình tích lũy bằng cách biểu diễn h^k $(k > 1)$ theo dạng đệ quy, sử dụng trung bình tích lũy trước đó h^{k-1} và giá trị tại bước thứ k của dữ liệu chuỗi thời gian:

$$h^k = \frac{k-1}{k} h^{k-1} + \frac{1}{k} w^k$$

Trung bình lũy thừa (exponential average)

- Ở mỗi bước thứ k, trung bình tích lũy h^k $(k > 1)$:

$$h^k = \frac{k-1}{k} h^{k-1} + \frac{1}{k} w^k$$

- Hai hệ số của mỗi bước cập nhật luôn có tổng bằng 1. Khi k lớn dần lên thì hệ số của h^{k-1} tiệm cận về 1, và hệ số của w^k tiệm cận về 0.
- Để tạo ra trung bình lũy thừa / trung bình hàm mũ, thì ta cố định hệ số của h^{k-1} là một hằng số $\beta \in [0,1]$ và hệ số của w^k là $1 - \beta$.

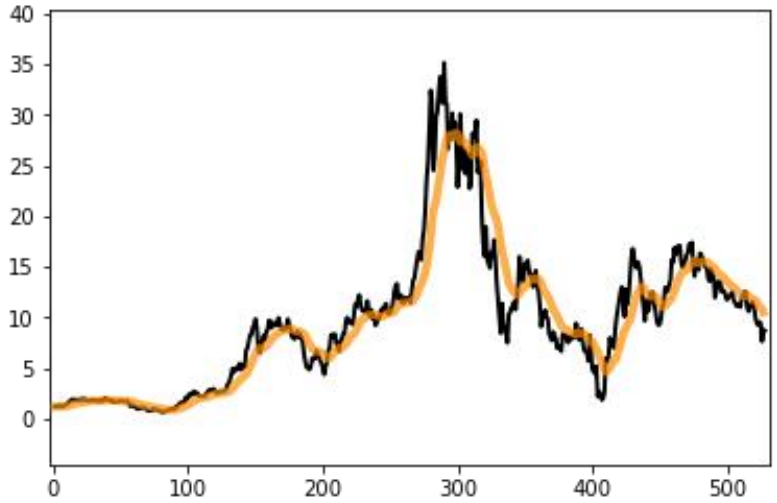
$$\begin{aligned} h^k &= \beta h^{k-1} + (1-\beta) w^k \\ &= \beta(\beta h^{k-2} + (1-\beta) w^{k-1}) + (1-\beta) w^k \\ &= \beta^2 h^{k-2} + \beta(1-\beta) w^{k-1} + (1-\beta) w^k \\ &= \beta^2(\beta h^{k-3} + (1-\beta) w^{k-2}) + \beta(1-\beta) w^{k-1} + (1-\beta) w^k = \ \ldots\ldots \\ &= \beta^k h^0 + \beta^{k-1}(1-\beta) w^1 + \beta^{k-2}(1-\beta) w^2 + \ldots + \beta(1-\beta) w^{k-1} + (1-\beta) w^k \end{aligned}$$

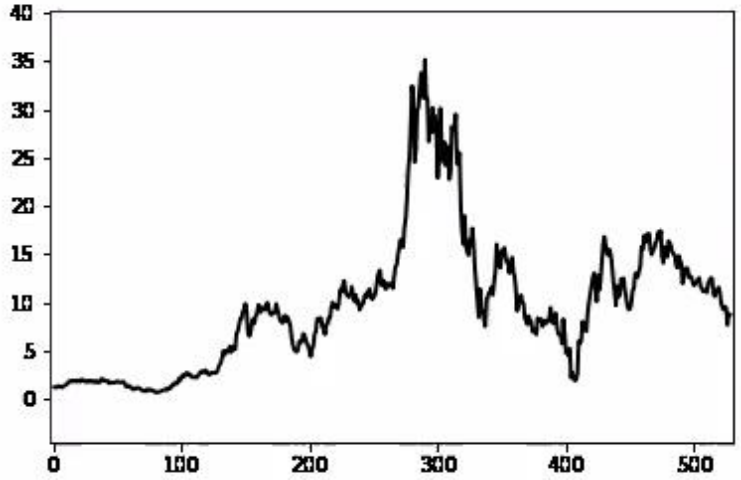
- β kiểm soát đánh đổi (trade-off): β càng nhỏ thì trung bình lũy thừa càng giống dữ liệu thô zig-zag. β càng lớn thì giá trị trung bình tích lũy được cập nhật càng giống giá trị trước đó.

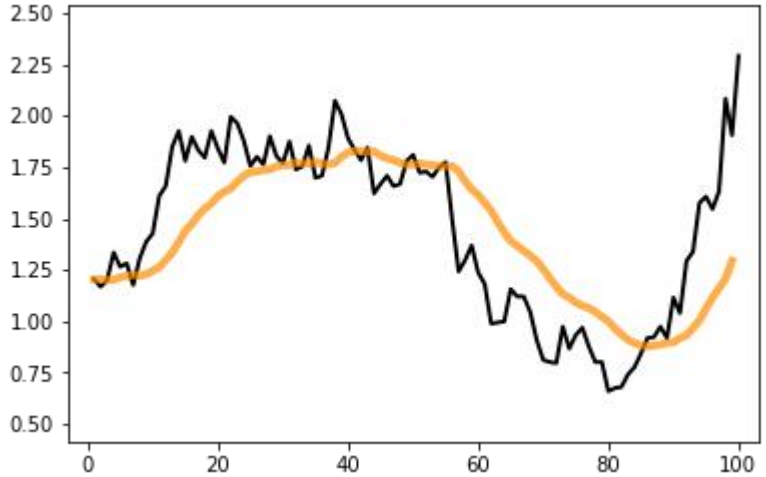
Trung bình lũy thừa (exponential average)

- Ở mỗi bước thứ k, trung bình tích lũy h^k:

$$h^k = \beta h^{k-1} + (1-\beta)w^k$$







Trung bình lũy thừa (exponential average)

- Cập nhật ở mỗi bước của Gradient Descent:

$$\boldsymbol{w}^{k} = \boldsymbol{w}^{k-1} - \alpha \nabla g(\boldsymbol{w}^{k-1})$$

- Dãy các bước đi $\boldsymbol{w}^{1}, \boldsymbol{w}^{2}, ..., \boldsymbol{w}^{K}$ và dãy các gradient âm $-\nabla g(\boldsymbol{w}^{0}), -\nabla g(\boldsymbol{w}^{1}), ..., -\nabla g(\boldsymbol{w}^{K-1})$ cũng có thể xem như một dạng dữ liệu chuỗi thời gian.
- Nếu các hướng giảm $-\nabla g(\boldsymbol{w}^{0}), -\nabla g(\boldsymbol{w}^{1}), ..., -\nabla g(\boldsymbol{w}^{K-1})$ có hiện tượng zig-zag, thì các bước đi của Gradient Descent cũng có biểu hiện zig-zag. Ta có thể sử dụng trung bình lũy thừa để "làm trơn" đường đi của Gradient Descent.
- Ta khởi tạo $\boldsymbol{d}^{0} = -\nabla g(\boldsymbol{w}^{0})$ và thực hiện tính trung bình lũy thừa các hướng giảm (exponentially averaged descent direction):

$$\boldsymbol{d}^{k-1} = \beta \boldsymbol{d}^{k-2} + (1-\beta)(-\nabla g(\boldsymbol{w}^{k-1}))$$

- Bước cập nhật trong Gradient Descent được thay đổi thành:

$$\boldsymbol{d}^{k-1} = \beta \boldsymbol{d}^{k-2} + (1-\beta)\left(-\nabla g(\boldsymbol{w}^{k-1})\right)$$
$$\boldsymbol{w}^{k} = \boldsymbol{w}^{k-1} + \alpha \boldsymbol{d}^{k-1}$$

Gradient Descent với momentum

- Bước cập nhật trong Gradient Descent được sửa đổi thành:

$$\boldsymbol{d}^{k-1} = \beta \boldsymbol{d}^{k-2} + (1-\beta)\left(-\nabla g(\boldsymbol{w}^{k-1})\right)$$
$$\boldsymbol{w}^{k} = \boldsymbol{w}^{k-1} + \alpha \boldsymbol{d}^{k-1}$$

- Phiên bản Gradient Descent trên gọi là Gradient Descent được gia tốc bằng động lượng (momentum). Momentum chính là trung bình lũy thừa của hướng giảm (exponentially averaged descent direction) $\boldsymbol{d}^{k-1}$.
- Thay vì sử dụng tính trung bình lũy thừa các hướng giảm (gradient âm) thì ta có thể tính momentum của các vector gradient, và thực hiện cập nhật theo hướng momentum âm.

$$\boldsymbol{d}^{k-1} = \beta \boldsymbol{d}^{k-2} + (1-\beta)\nabla g(\boldsymbol{w}^{k-1})$$
$$\boldsymbol{w}^{k} = \boldsymbol{w}^{k-1} - \alpha \boldsymbol{d}^{k-1}$$

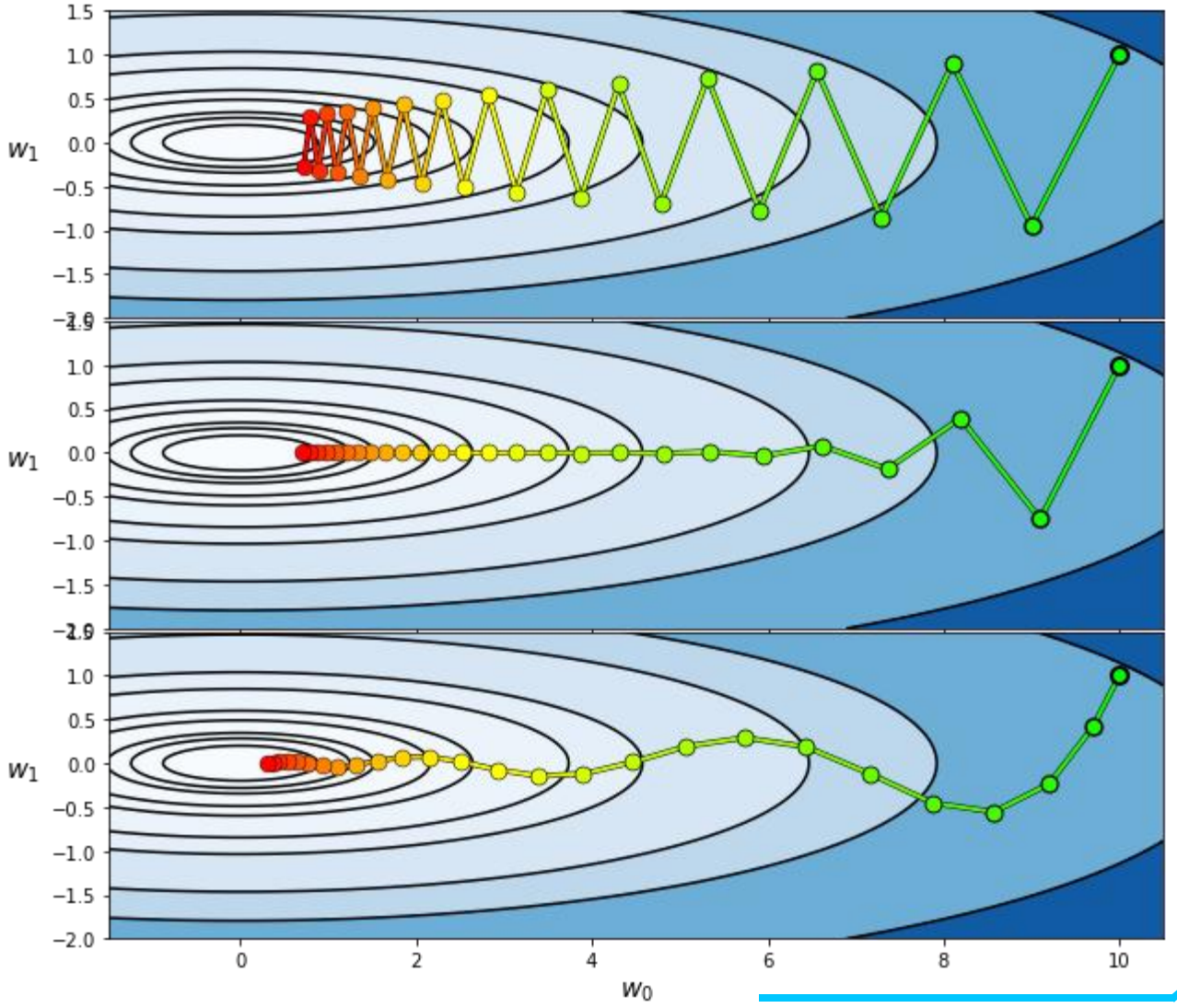
Ví dụ

- Sử dụng Gradient Descent với $\alpha = 0.1$ cực tiểu hóa hàm bậc hai có dạng:

$$g(w) = a + \boldsymbol{b}^T\boldsymbol{w} + \boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w}$$

với $a = 0$, $\boldsymbol{b} = (0,0)$, và $\boldsymbol{C} = \begin{bmatrix} 0.5 & 0 \\ 0 & 9.75 \end{bmatrix}$.

- Thực hiện 3 lượt chạy:
  - ➢ Không sử dụng momentum,
  - ➢ Sử dụng momentum với $\beta = 0.2$,
  - ➢ Sử dụng momentum với $\beta = 0.7$.



GRADIENT DESCENT ĐƯỢC CHUẨN HÓA

NORMALIZED GRADIENT DESCENT

Chuẩn hóa gradient

- Một vấn đề cơ bản của thuật toán Gradient Descent là độ lớn của gradient (âm) trở nên rất nhỏ khi gần các điểm dừng (stationary points). Gradient Descent bước rất chậm khi tiến gần các điểm dừng, và có thể bị dừng lại gần các điểm yên ngựa (saddle points).
- Trong thuật toán Gradient Descent tiêu chuẩn, kích thước bước cập nhật tỉ lệ thuận với độ lớn của gradient $\alpha\left\|\nabla g(\boldsymbol{w}^{k-1})\right\|_2$ nên nếu $\left\|\nabla g(\boldsymbol{w}^{k-1})\right\|_2 \approx 0$ thì bước đi của thuật toán sẽ rất nhỏ.
- Ta có thể chuẩn hóa vector gradient ở mỗi bước cập nhật như sau:

$$w^k = w^{k-1} - \alpha \frac{\nabla g(\boldsymbol{w}^{k-1})}{\left\|\nabla g(\boldsymbol{w}^{k-1})\right\|_2}$$

- Kích thước bước cập nhật của Gradient Descent được chuẩn hóa sẽ là:

$$\left\|\boldsymbol{w}^k - \boldsymbol{w}^{k-1}\right\|_2 = \left\|\boldsymbol{w}^{k-1} - \alpha \frac{\nabla g(\boldsymbol{w}^{k-1})}{\left\|\nabla g(\boldsymbol{w}^{k-1})\right\|_2} - \boldsymbol{w}^{k-1}\right\|_2 = \left\|-\alpha \frac{\nabla g(\boldsymbol{w}^{k-1})}{\left\|\nabla g(\boldsymbol{w}^{k-1})\right\|_2}\right\|_2 = \alpha$$

- Kích thước bước cập nhật của thuật toán được quy định hoàn toàn bởi siêu tham số độ dài bước (steplength) α.

Gradient Descent được chuẩn hóa

$$w^k = w^{k-1} - \alpha \frac{\nabla g(\boldsymbol{w}^{k-1})}{\left\|\nabla g(\boldsymbol{w}^{k-1})\right\|_2}$$

- Ta có thể viết lại công thức của bước cập nhật được chuẩn hóa như sau:

$$w^k = w^{k-1} - \frac{\alpha}{\left\|\nabla g(\boldsymbol{w}^{k-1})\right\|_2} \nabla g(\boldsymbol{w}^{k-1})$$

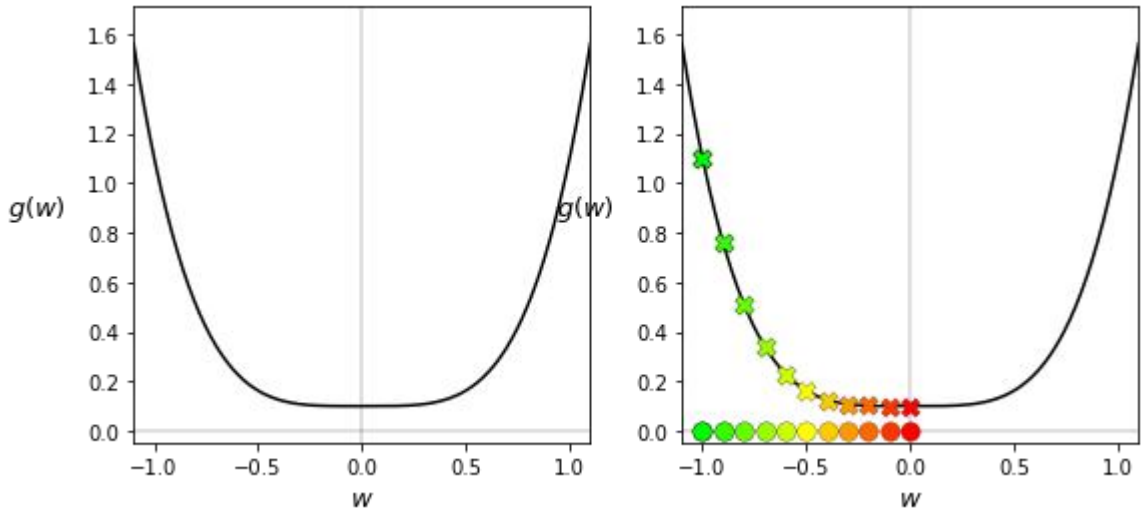
- Ta có thể diễn giải công thức cập nhật trên như là một bước của thuật toán Gradient Descent tiêu chuẩn trong đó siêu tham số độ dài bước (steplength) / hệ số học (learning rate) $\frac{\alpha}{\left\|\nabla g(\boldsymbol{w}^{k-1})\right\|_2}$ được thích ứng (adapt) ở mỗi bước cập nhật dựa trên độ lớn (magnitude) của gradient sao cho kích thước của bước cập nhật bằng đúng α.
- Một hằng số ε nhỏ (ví dụ, 10^{-7} hoặc nhỏ hơn) có thể được thêm vào để tránh việc chia cho 0:

$$w^k = w^{k-1} - \frac{\alpha}{\left\|\nabla g(\boldsymbol{w}^{k-1})\right\|_2 + \varepsilon} \nabla g(\boldsymbol{w}^{k-1})$$

Gradient Descent được chuẩn hóa

$$g(w) = w^4 + 0.1$$

- Sử dụng Gradient Descent được chuẩn hóa với $\alpha = 0.1$ và khởi tạo xa vị trí cực trị.

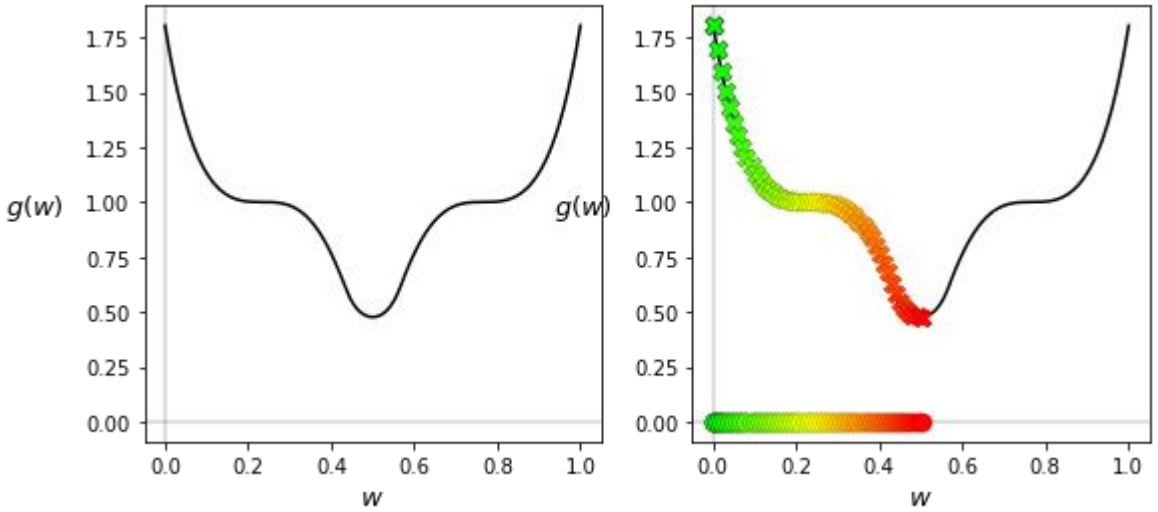


- Gradient Descent được chuẩn hóa (normalized gradient descent) có các bước đi với kích thước bằng với siêu tham số độ dài bước α.

Gradient Descent được chuẩn hóa

$$g(w) = \max(0, (3w - 2.3)^3 + 1)^2 + \max(0, (-3w + 0.7)^3 + 1)^2$$

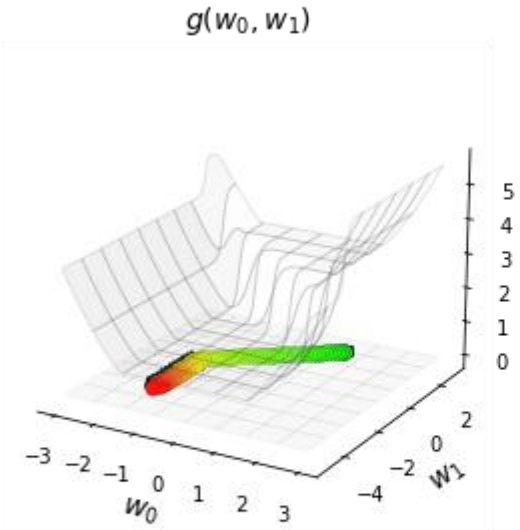
- Có cực tiểu tại $w = \frac{1}{2}$ và các điểm yên ngựa tại $w = \frac{7}{30}$ và $w = \frac{23}{30}$.
- Thực hiện Gradient Descent được chuẩn hóa với 50 bước cập nhật và $\alpha = 0.01$.

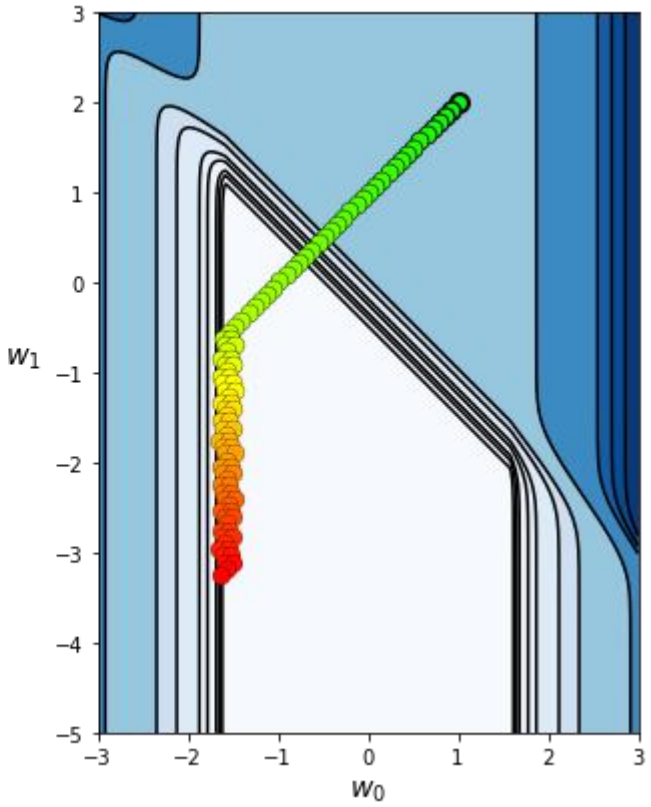


Gradient Descent được chuẩn hóa

$$g(w) = \tanh(4w_0 + 4w_1) + \max(1,\ 0.4w_0^2) + 1$$

- Khởi tạo Gradient Descent tại điểm $w^0 = (1,2)$.
- Thực hiện 100 bước cập nhật với $\alpha = 0.1$.





Chuẩn hóa độ lớn của gradient

- Vector gradient bao gồm N đạo hàm riêng (partial derivatives):

$$\nabla g(\boldsymbol{w}) = \begin{bmatrix} \frac{\partial}{\partial w_1} g(\boldsymbol{w}) \\ \vdots \\ \frac{\partial}{\partial w_N} g(\boldsymbol{w}) \end{bmatrix}$$

với đạo hàm riêng $\frac{\partial}{\partial w_j} g(\boldsymbol{w})$ mô tả hành vi của gradient dọc theo trục tọa độ thứ j.

- Khi ta thực hiện chuẩn hóa gradient, thì mỗi thành phần đạo hàm riêng được điều chỉnh:

$$\frac{\frac{\partial}{\partial w_j} g(\boldsymbol{w})}{\|\nabla g(\boldsymbol{w})\|_2} = \frac{\frac{\partial}{\partial w_j} g(\boldsymbol{w})}{\sqrt{\sum_{n=1}^{N} \left(\frac{\partial}{\partial w_n} g(\boldsymbol{w})\right)^2}}$$

- Ta thấy đạo hàm riêng thứ j được chuẩn hóa bằng cách chia cho tổng độ lớn của tất cả các đạo hàm riêng.

Chuẩn hóa độ lớn của gradient

$$\frac{\frac{\partial}{\partial w_j} g(\boldsymbol{w})}{\left\|\nabla g(\boldsymbol{w}^{k-1})\right\|_2} = \frac{\frac{\partial}{\partial w_j} g(\boldsymbol{w})}{\sqrt{\sum_{n=1}^{N} \left(\frac{\partial}{\partial w_n} g(\boldsymbol{w})\right)^2}}$$

- Nếu đạo hàm riêng thứ j đã có độ lớn rất nhỏ thì thao tác chuẩn hóa này sẽ triệt tiêu đóng góp của nó đến mỗi bước cập nhật của thuật toán.
- Việc này khiến quá trình tối ưu hóa không hiệu quả nếu hàm số ta cần tối ưu có chứa các vùng bằng phẳng chỉ theo một vài hướng nào đó.
- Một giải pháp là ta thực hiện chuẩn hóa gradient theo từng thành phần (component-wise):

$$\frac{\frac{\partial}{\partial w_j} g(\boldsymbol{w})}{\sqrt{\left(\frac{\partial}{\partial w_j} g(\boldsymbol{w})\right)^2}} = \frac{\frac{\partial}{\partial w_j} g(\boldsymbol{w})}{\left|\frac{\partial}{\partial w_j} g(\boldsymbol{w})\right|} = \text{sign}\left(\frac{\partial}{\partial w_j} g(\boldsymbol{w})\right)$$

Chuẩn hóa gradient theo thành phần

- Ở mỗi bước đi của Gradient Descent, thành phần thứ j được cập nhật với đạo hàm riêng được chuẩn hóa như sau:

$$w_j^k = w_j^{k-1} - \alpha \frac{\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})}{\sqrt{\left(\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})\right)^2}} = w_j^{k-1} - \alpha \text{sign}\left(\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})\right)$$

- Toàn bộ bước đi có thể được biểu diễn:

$$\boldsymbol{w}^k = \boldsymbol{w}^{k-1} - \alpha \text{sign}\left(\nabla g(\boldsymbol{w}^{k-1})\right)$$

- Kích thước một bước cập nhật của Gradient Descent được chuẩn hóa theo thành phần là:

$$\left\|\boldsymbol{w}^k - \boldsymbol{w}^{k-1}\right\|_2 = \left\|-\alpha \text{sign}\left(\nabla g(\boldsymbol{w}^{k-1})\right)\right\|_2 = \sqrt{N}\alpha$$

Chuẩn hóa gradient theo thành phần

$$w_j^k = w_j^{k-1} - \alpha \frac{\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})}{\sqrt{\left(\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})\right)^2}}$$

- Ta có thể viết lại công thức cập nhật tại mỗi bước đi của Gradient Descent được chuẩn hóa, thành phần thứ j được cập nhật với đạo hàm riêng được chuẩn hóa như sau:

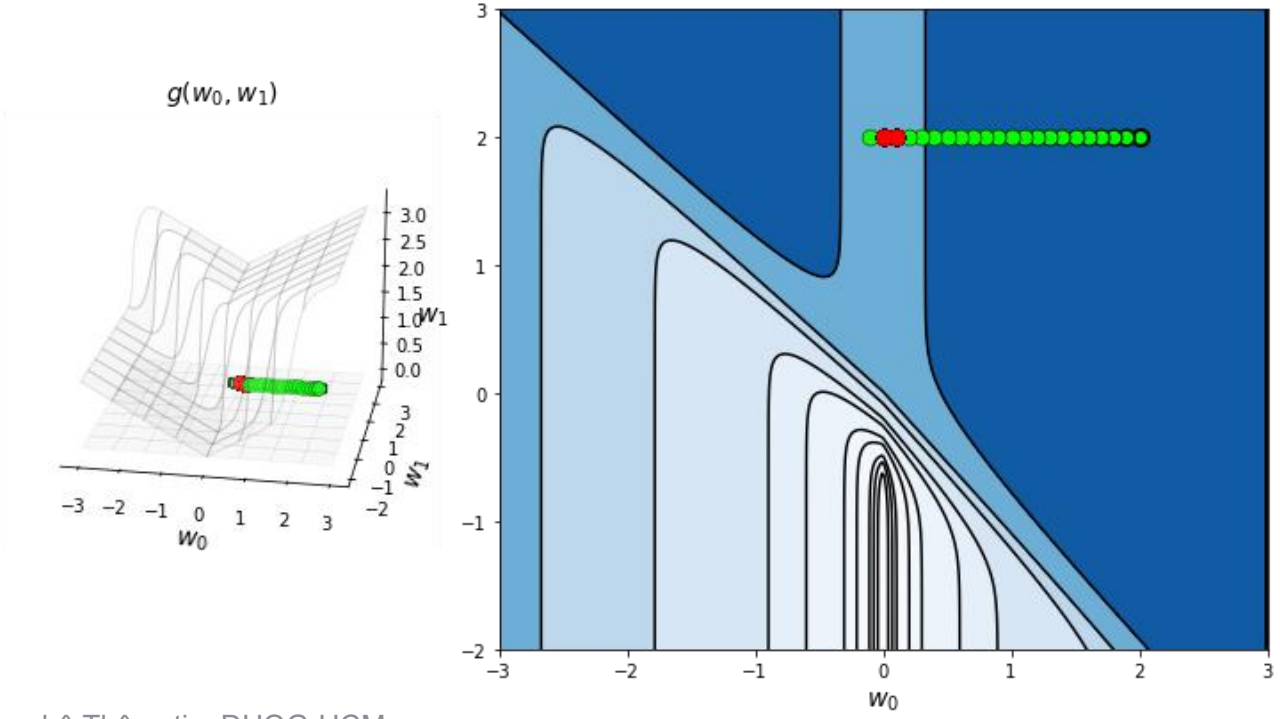
$$w_j^k = w_j^{k-1} - \frac{\alpha}{\sqrt{\left(\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})\right)^2}} \frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})$$

- Ta có thể diễn giải công thức cập nhật trên như là một bước của thuật toán Gradient Descent tiêu chuẩn với siêu tham số độ dài bước (steplength) / hệ số học (learning rate) riêng cho từng thành phần được thích ứng (adapt) dựa trên độ lớn của đạo hàm riêng sao cho kích thước của

Gradient Descent được chuẩn hóa

$$g(w) = \max(0, \tanh(4w_0 + 4w_1)) + \max(0, \text{abs}(0.4w_0)) + 1$$

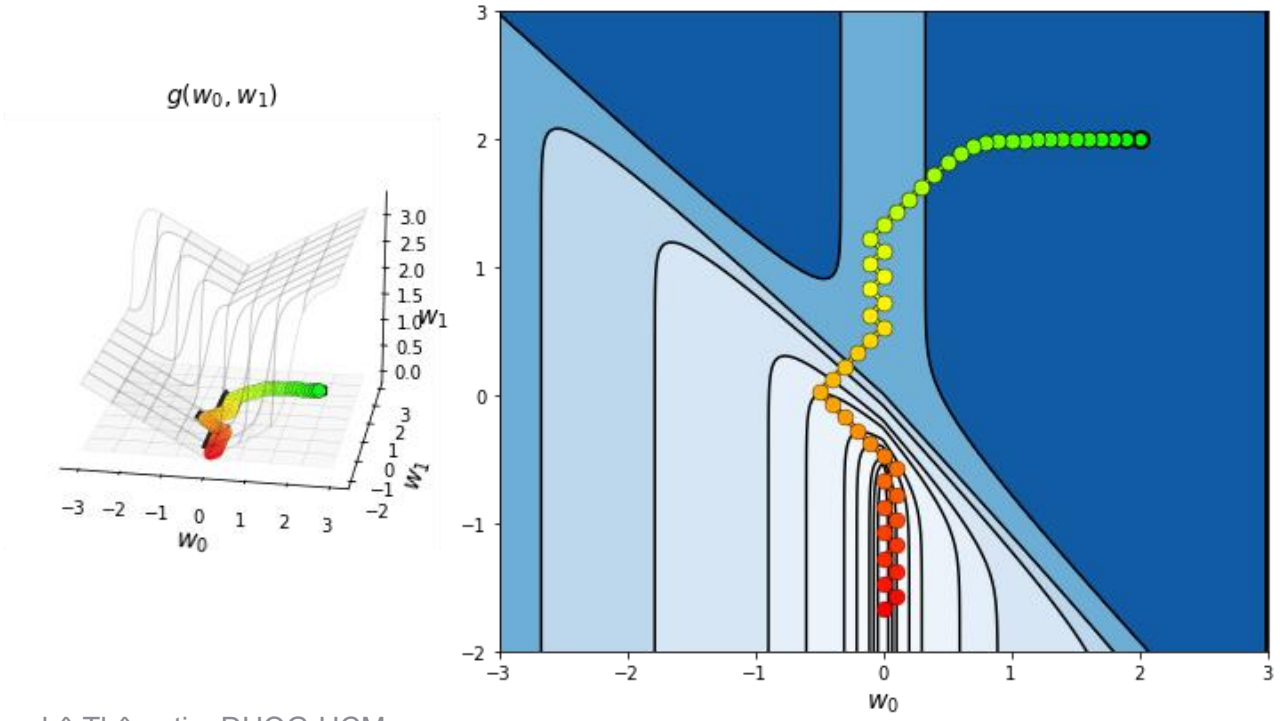
- Hàm số này rất phẳng theo trục tọa độ w_1 với mỗi giá trị của w_0. Có một thung lũng rất hẹp dẫn đến vị trí cực tiểu theo hướng w_1 tại giá trị $w_0 = 0$.
- Thực hiện Gradient Descent với 1000 bước cập nhật được chuẩn hóa độ lớn đầy đủ. Độ lớn của đạo hàm riêng của w_1 bị triệt tiêu nên gần như thành phần w_1 không được cập nhật.



Gradient Descent được chuẩn hóa

$$g(w) = \max(0, \tanh(4w_0 + 4w_1)) + \max(0, \text{abs}(0.4w_0)) + 1$$

- Hàm số này rất phẳng theo trục tọa độ w_1 với mỗi giá trị của w_0. Có một thung lũng rất hẹp dẫn đến vị trí cực tiểu theo hướng w_1 tại giá trị $w_0 = 0$.
- Thực hiện Gradient Descent với 50 bước cập nhật được chuẩn hóa độ lớn theo từng thành phần. Độ lớn của đạo hàm riêng của w_1 do đó được cải thiện.



MỘT SỐ PHƯƠNG PHÁP GD CẢI TIẾN KHÁC

ADVANCED GRADIENT-BASED METHODS

Hai kỹ thuật cải thiện Gradient Descent

- Gradient Descent với momentum có thể khắc phục tình trạng zig-zag của Gradient Descent tiêu chuẩn. Động lượng (momentum) của hướng giảm $\boldsymbol{d}^{k-1}$ là trung bình lũy thừa của các hướng đi của Gradient Descent tại các bước cập nhật của thuật toán:

$$\boldsymbol{d}^{k-1} = \beta \boldsymbol{d}^{k-2} + (1-\beta)\left(-\nabla g(\boldsymbol{w}^{k-1})\right)$$
$$\boldsymbol{w}^{k} = \boldsymbol{w}^{k-1} + \alpha \boldsymbol{d}^{k-1}$$

với $\beta \in [0,1)$.

- Gradient Descent được chuẩn hóa độ lớn theo từng thành phần giúp khắc phục vấn đề của Gradient Desent tiêu chuẩn khi đi qua những vùng bằng phẳng trong không gian tìm kiếm. Thành phần thứ j được cập nhật ở mỗi bước theo dạng:

$$\boldsymbol{w}_j^{k} = \boldsymbol{w}_j^{k-1} - \alpha \frac{\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})}{\sqrt{\left(\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})\right)^2}}$$

Gradient Descent chuẩn hóa với momentum

Ý tưởng: kết hợp momentum với kỹ thuật chuẩn hóa các hướng đi của gradient descent.

- Ta thực hiện chuẩn hóa theo từng thành phần của vector động lượng các hướng giảm của thuật toán. Tại mỗi bước của thuật toán, thành phần thứ j sẽ được cập nhật bởi động lượng của đạo hàm riêng theo w_j được chuẩn hóa :

$$d_j^{k-1} = \beta d_j^{k-2} - (1-\beta)\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})$$

$$d_j^{k-1} \leftarrow \frac{d_j^{k-1}}{\sqrt{\left(d_j^{k-1}\right)^2}}$$

- Vector cập nhật $\boldsymbol{d}^{k-1}$ có các thành phần d_j^{k-1} được tính theo công thức trên. Mỗi bước đi của thuật toán, phép cập nhật là:

$$\boldsymbol{w}^k = \boldsymbol{w}^{k-1} + \alpha \boldsymbol{d}^{k-1}$$

- Có nhiều phương pháp khác nhau để hiện thực hóa các phép kết hợp momentum và chuẩn hóa.

RMSprop – Root Mean Squared Propagation

- Trong thuật toán Gradient Descent được chuẩn hóa theo từng thành phần, thì mỗi thành phần của gradient được chuẩn hóa với độ lớn của đạo hàm riêng tương ứng.
- Ta có thể chuẩn hóa từng thành phần của gradient với trung bình lũy thừa của độ lớn của đạo hàm riêng tương ứng dựa trên gradient của các bước đi trước đó.
- Trung bình lũy thừa của bình phương độ lớn của đạo hàm riêng theo w_j:

$$h_j^{k-1} = \gamma h_j^{k-2} + (1-\gamma)\left(\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})\right)^2$$

- Mỗi bước của RMSprop cập nhật thành phần thứ j như sau:

$$w_j^k = w_j^{k-1} - \alpha \frac{\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})}{\sqrt{h_j^{k-1}}}$$

với mẫu số thường được cộng thêm một đại lượng $\varepsilon > 0$ nhỏ (ví dụ, 10^{-8}) để tránh chia cho 0.

RMSprop – Root Mean Squared Propagation

$$w_j^k = w_j^{k-1} - \alpha \frac{\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})}{\sqrt{h_j^{k-1}}}$$

- Bước cập nhật trên có thể được viết lại như sau:

$$w_j^k = w_j^{k-1} - \frac{\alpha}{\sqrt{h_j^{k-1}}} \frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})$$

- Ta có thể diễn giải bước cập nhật của RMSprop là một bước cập nhật của Gradient Descent tiêu chuẩn với độ dài bước (steplength) / hệ số học (learning rate) riêng cho từng thành phần, và được thích ứng (adapt) tại mỗi bước đi của thuật toán dựa trên động lượng (momentum) của (bình phương) độ lớn của đạo hàm riêng tương ứng với mỗi thành phần.

Adam – Adaptive Moment Estimation

- Thành phần thứ j của bước cập nhật được tính dựa trên:

➢ Trung bình lũy thừa của đạo hàm riêng theo w_j:

$$d_j^{k-1} = \beta_1 d_j^{k-2} + (1-\beta_1)\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})$$

➢ Trung bình lũy thừa của bình phương độ lớn của đạo hàm riêng theo w_j:

$$h_j^{k-1} = \beta_2 h_j^{k-2} + (1-\beta_2)\left(\frac{\partial}{\partial w_j} g(\boldsymbol{w}^{k-1})\right)^2$$

với $\beta_1, \beta_2 \in [0,1]$ và ta có thể chọn lựa $\beta_1 = 0.9, \beta_2 = 0.999$.

- Ta khởi tạo $d_j^0 = \frac{\partial}{\partial w_j} g(\boldsymbol{w}^0)$ và $h_j^0 = \left(\frac{\partial}{\partial w_j} g(\boldsymbol{w}^0)\right)^2$.
- Thuật toán Adam tiêu chuẩn có sử dụng thêm các kỹ thuật hiệu chỉnh d_j^{k-1} và h_j^{k-1} khác.

Adam – Adaptive Moment Estimation

- Thành phần thứ j được cập nhật bởi:

$$w_j^k = w_j^{k-1} - \alpha \frac{d_j^{k-1}}{\sqrt{h_j^{k-1}}}$$

với mẫu số có thể được thêm một đại lượng $\varepsilon > 0$ nhỏ (ví dụ, 10^{-8}) để tránh chia cho 0.

- Nếu ta viết lại công thức cập nhật:

$$w_j^k = w_j^{k-1} - \frac{\alpha}{\sqrt{h_j^{k-1}}} d_j^{k-1}$$

thì có thể diễn giải một bước cập nhật của Adam như là một bước của Gradient Descent được gia tốc với động lượng (momentum) của gradient, và vector động lượng này được chuẩn hóa theo từng thành phần tùy theo động lượng của (bình phương) độ lớn của đạo hàm riêng của mỗi thành phần.

Tài liệu tham khảo

- J. Watt, R. Borhani, A. K. Katsaggelos (2020): *Machine Learning Refined. Foundations, Algorithms, and Applications* (2nd Edition). Cambridge University Press. https://www.mlrefined.com/
- A. Zhang, Z. Lipton, M. Li, A.J. Smola (2023): *Dive into Deep Learning*. Cambridge University Press. https://d2l.ai/
- K. P. Murphy (2022): *Probabilistic Machine Learning – An Introduction*. The MIT Press. https://probml.github.io/pml-book/book1.html