Huấn luyện mạng nơ-ron nhiều tầng ẩn bằng thuật toán Adam

Nhóm sinh viên thực hiện:

- Nguyễn Ngọc Lan Như - 1712644
- Hoàng Minh Quân – 1712688

Giáo viên hướng dẫn: Th.S. Trần Trung Kiên

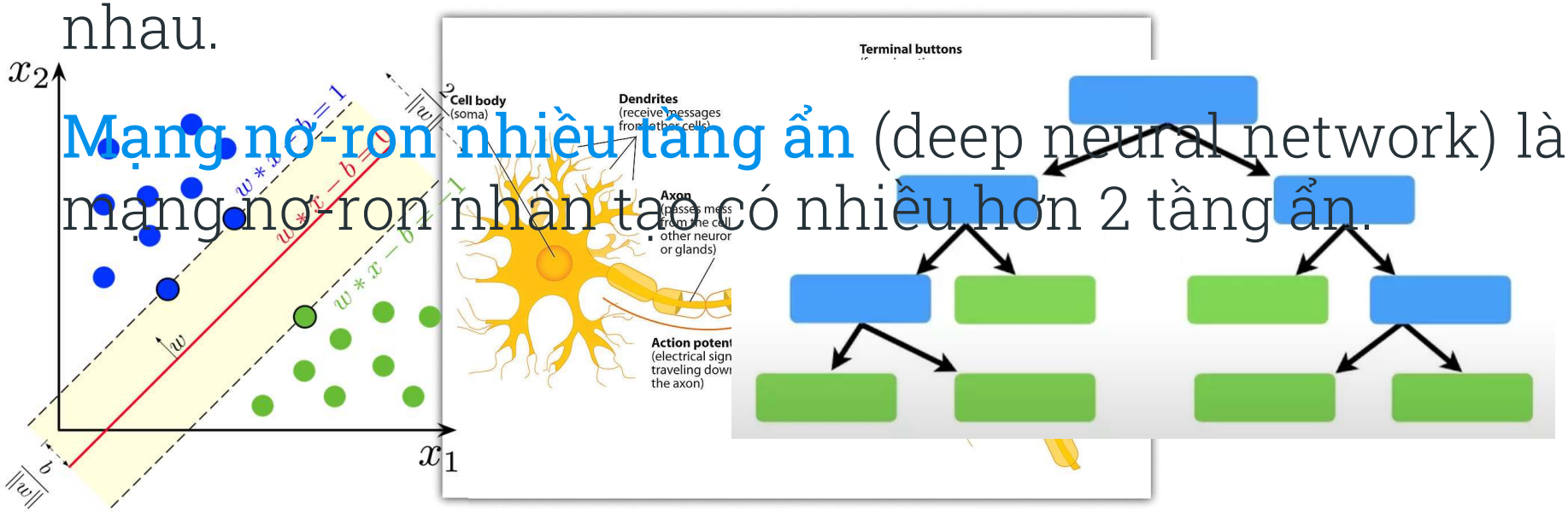
Mục lục

1.
Giới thiệu đề tài

Giới thiệu đề tài

Giới thiệu khái niệm

- **Học máy** (machine learning) là một chương trình máy tính có khả năng tự tìm kiếm các đặc trưng của dữ liệu đó để đưa ra dự đoán.
- **Mạng nơ-ron nhân tạo** (artificial neural network) là một mô hình học máy gồm các nơ-ron kết nối với nhau.
- **Mạng nơ-ron nhiều tầng ẩn** (deep neural network) là mạng nơ-ron nhân tạo có nhiều hơn 2 tầng ẩn.



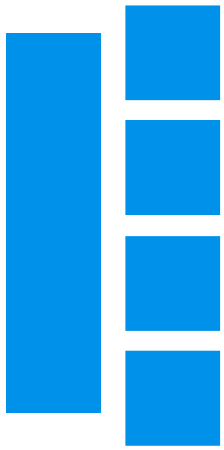
Giới thiệu đề tài

Giới thiệu khái niệm

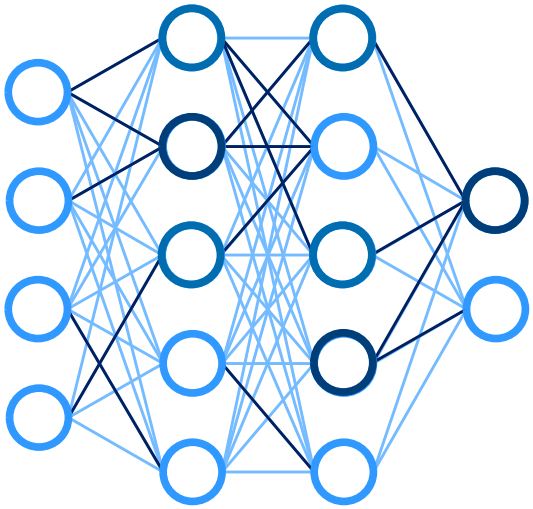
- **Hàm chi phí** cho biết độ lỗi trung bình của mạng nơ-ron trên tập dữ liệu huấn luyện.
- **Độ lỗi** là sự sai biệt giữa giá trị dự đoán của mạng với giá trị đúng.

Giới thiệu đề tài

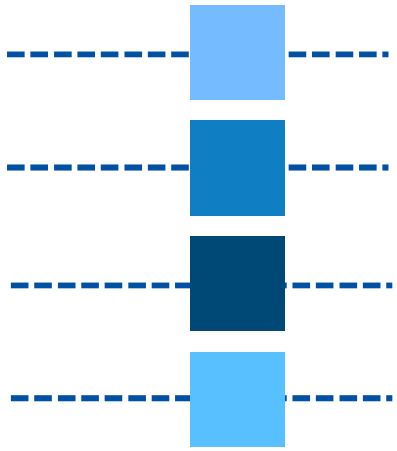
Giới thiệu khái niệm



Dữ liệu huấn luyện



Mạng nơ-ron
nhiều tầng ẩn



Output

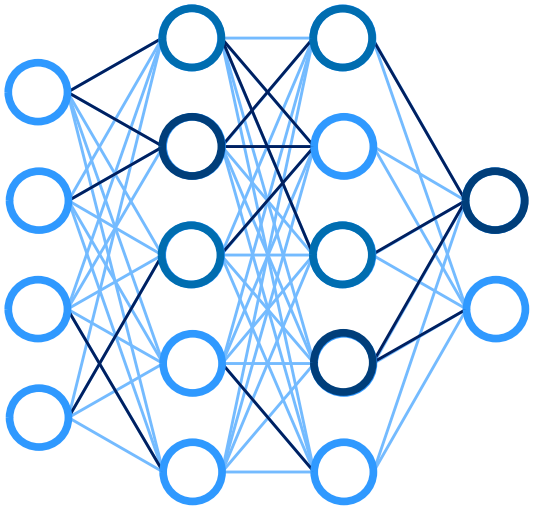
Hàm chi phí

0.38

Nhãn

Giới thiệu đề tài

Giới thiệu khái niệm



Mạng nơ-ron
nhiều tầng ẩn

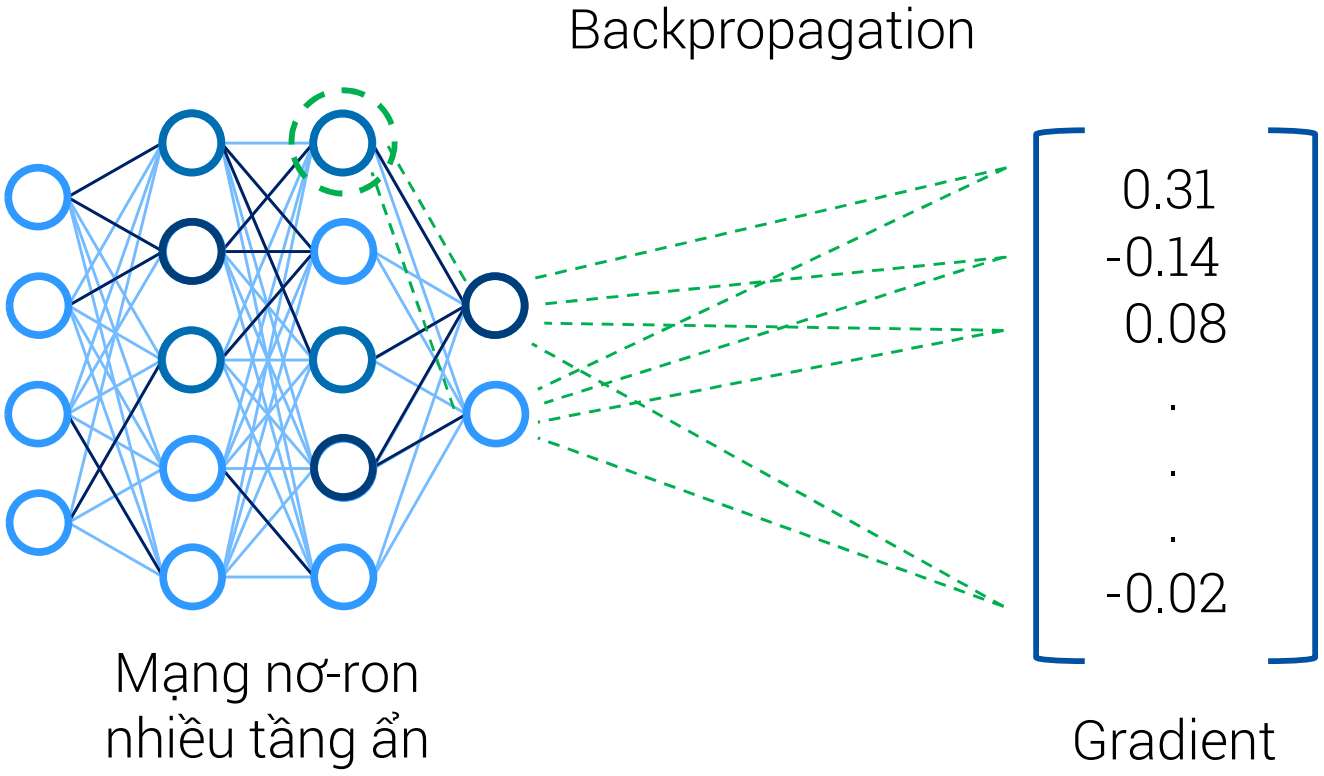
∇ Hàm chi phí

$$\begin{bmatrix} 0.81 \\ -0.14 \\ 0.08 \\ . \\ . \\ . \\ -0.02 \end{bmatrix}$$

Gradient

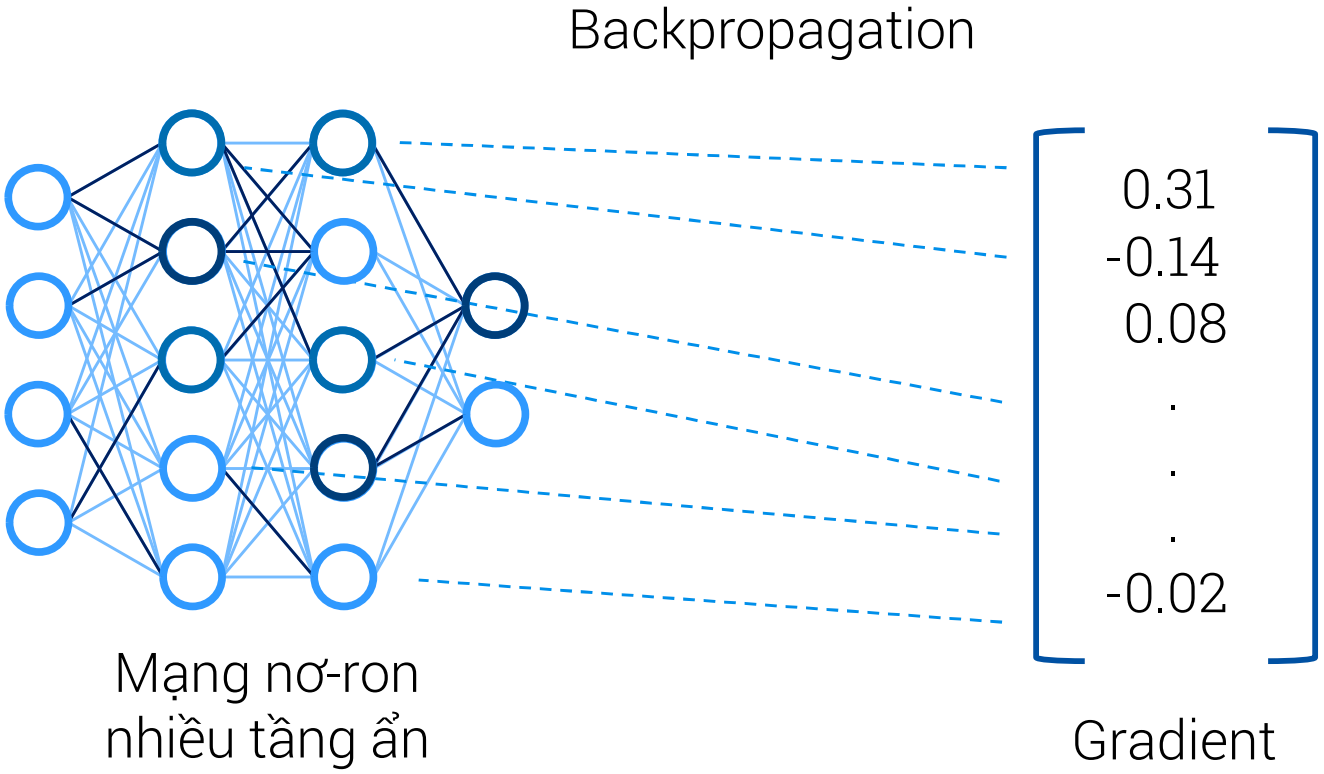
Giới thiệu đề tài

Giới thiệu khái niệm



Giới thiệu đề tài

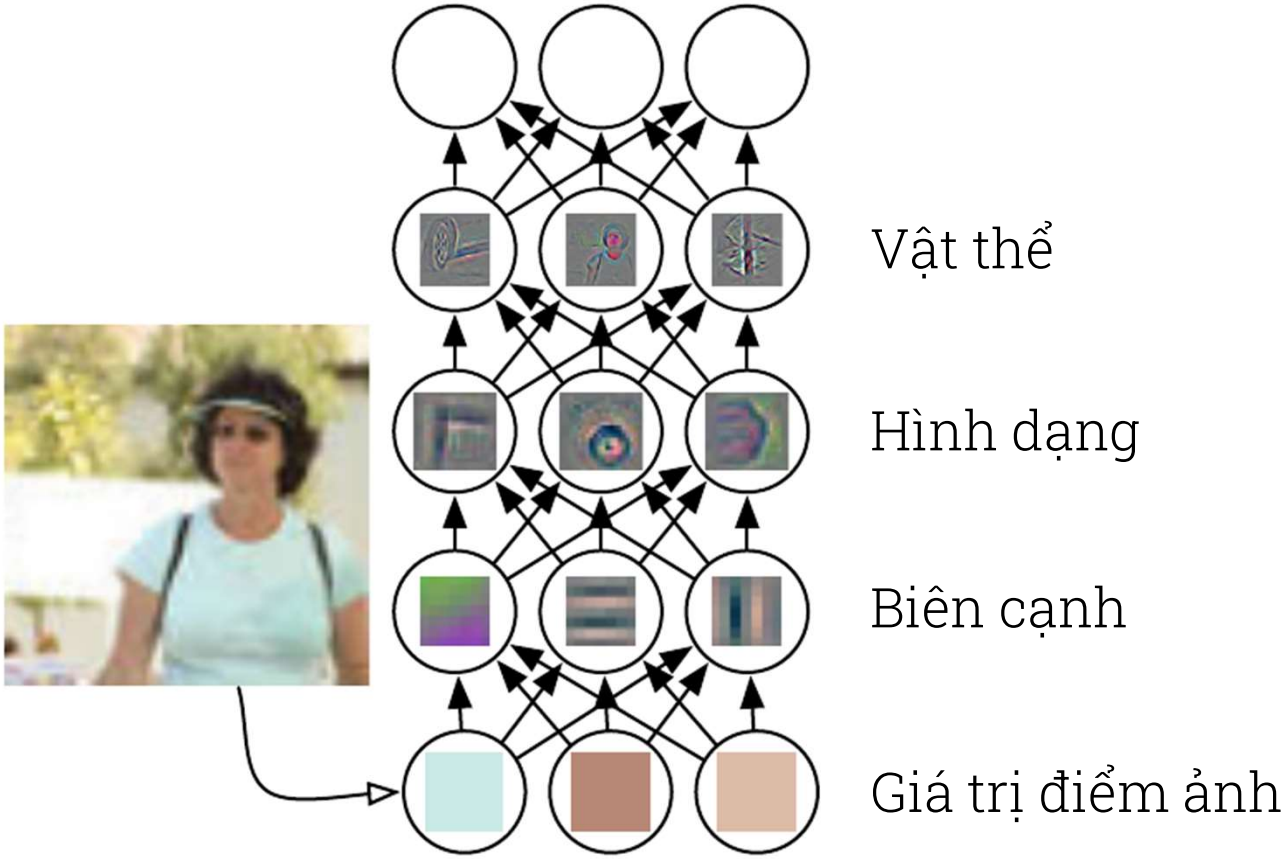
Giới thiệu khái niệm



Giới thiệu đề tài

Tại sao lại sử dụng mạng nơ-ron nhiều tầng ẩn?

- Rút trích đặc trưng từ đơn giản đến phức tạp.



Giới thiệu đề tài

Tại sao lại sử dụng mạng nơ-ron nhiều tầng ẩn?

- Độ phức tạp tính toán của các mạch boolean
  (Computational complexity of boolean circuit)
  - Một mạng nơ-ron có ít tầng ẩn sẽ cần thêm rất nhiều nơ-ron trong mỗi tầng để xấp xỉ được hàm số của một mạng nơ-ron nhiều tầng ẩn.

Giới thiệu đề tài

Phát biểu bài toán

- **Input:** Hàm chi phí với các tham số là bộ trọng số của mạng nơ-ron nhiều tầng ẩn. Trong hàm chi phí có thể có thêm lượng "regularization" để củng cố kết quả mong muốn.
- **Output:** Bộ trọng số của mạng nơ-ron nhiều tầng ẩn cho giá trị của hàm chi phí là nhỏ nhất (hoặc tương đối nhỏ).

regularization: một phương pháp giới hạn độ phức tạp của mô hình để giúp mô hình hoạt động tốt hơn trên dữ liệu ngoài tập huấn luyện.

Giới thiệu đề tài

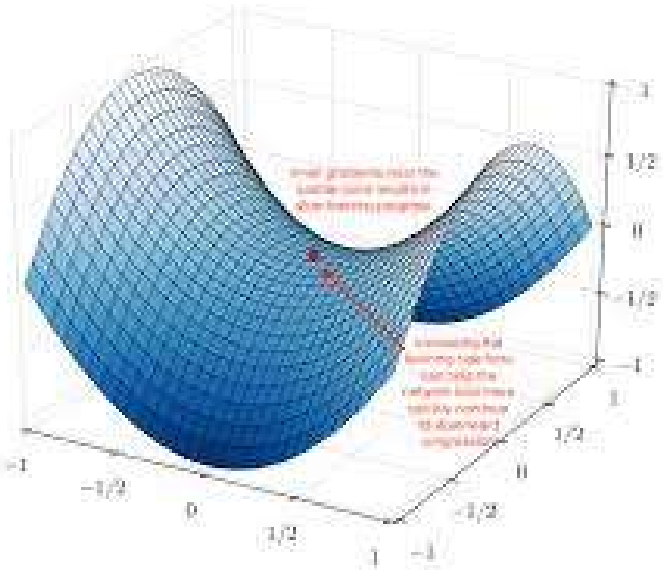
Khó khăn khi huấn luyện mạng nơ-ron nhiều tầng ẩn

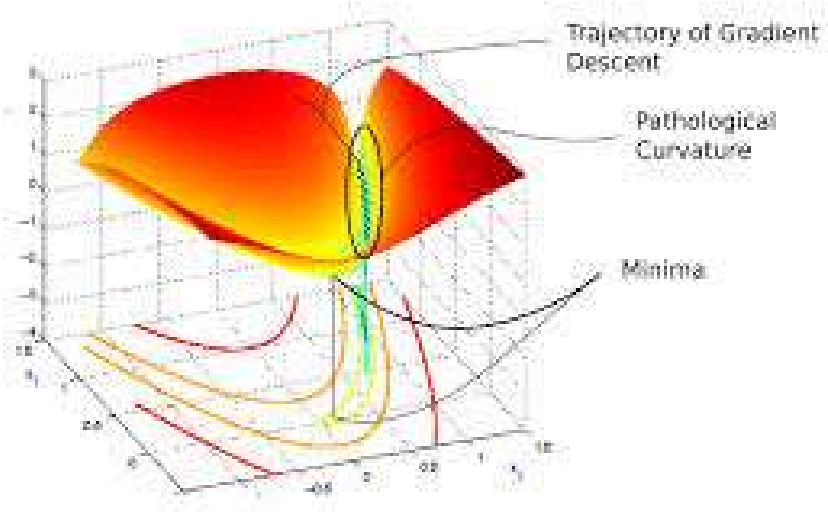
- Mặt phẳng lỗi phức tạp
- Biến động về đạo hàm.
  - Đạo hàm tiêu biến (Vanishing gradient)
  - Đạo hàm cực biến (Exploding gradient)
- Giá trị khởi tạo ban đầu của mạng ảnh hưởng nhiều đến quá trình học.
- Cần tinh chỉnh siêu tham số cẩn thận.

Giới thiệu đề tài

Mặt phẳng lỗi

- Non-convex
- Nhiều **điểm yên ngựa** trong vùng bằng phẳng (plateau) có độ lỗi cao và rãnh hẹp
- Nhiều **cực tiểu địa phương**





Giới thiệu đề tài

Đề tài liên quan

- Hướng tiếp cận truyền thống: Stochastic Gradient Descent (SGD).
  - Sử dụng đạo hàm bậc nhất để xác định hướng đi có sự thay đổi lớn nhất.
  - Lấy một phần giá trị đạo hàm làm độ dài bước nhảy.

Giới thiệu đề tài

Đề tài liên quan: SGD

- Sử dụng đạo hàm bậc nhất.
  - Rất khó di chuyển khi đạo hàm tiệm cận 0.
  - Hướng cập nhật tiếp theo luôn luôn vuông góc với hướng của bước trước đó → Khó di chuyển trong các vùng hẹp.
  - Cập nhật một lượng chung cho tất cả tham số.

Giới thiệu đề tài

Đề tài liên quan: SGD

- Thực hiện cập nhật trên từng điểm dữ liệu.
  - Tính toán nhanh.
  - Tạo ra sự ngẫu nhiên (stochasticity) giúp vượt qua critical point.
  - Sự ngẫu nhiên có thể khiến độ lỗi dao động phức tạp.

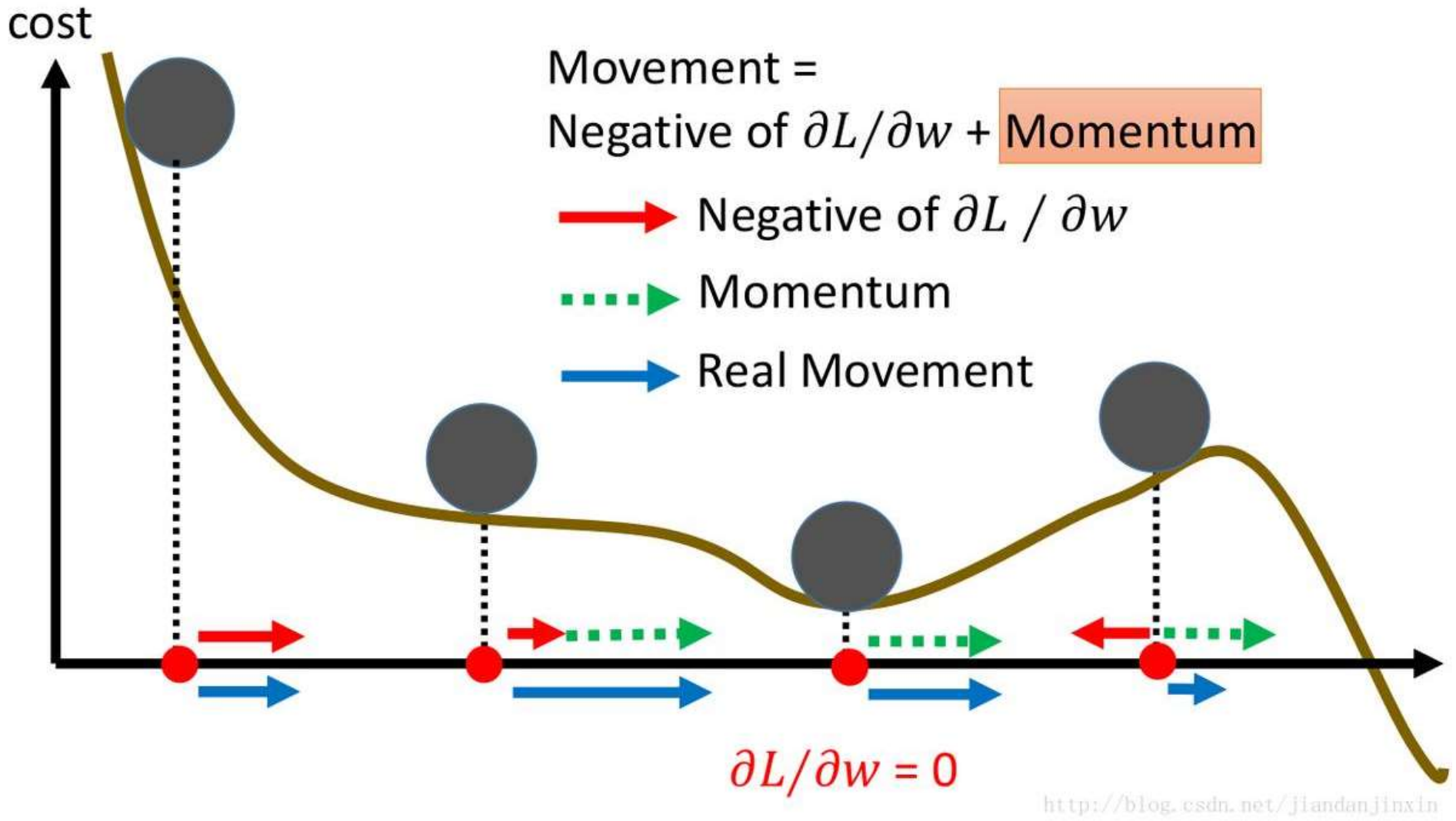
Giới thiệu đề tài

Đề tài liên quan

- Cải tiến: Stochastic Gradient Descent with Momentum (Momentum).
  - Áp dụng nguyên lý lực quán tính.
  - Di chuyển nhanh hơn khi gặp hướng dốc.

Giới thiệu đề tài

Đề tài liên quan: Momentum



Giới thiệu đề tài

Đề tài liên quan: Momentum

- Tăng dần tốc độ khi "lăn xuống".
  - Giảm độ lỗi nhanh hơn.
- Giúp tăng tốc trên các hướng có độ dốc cao.
  - Giảm dao động quanh vùng rãnh hẹp.
- Cộng một lượng quán tính vào giá trị đạo hàm.
  - Vượt qua các điểm có đạo hàm tiệm cận 0.
  - Di chuyển nhanh hơn trong các vùng bằng phẳng.
  - Có thể đi vượt qua các điểm cực tiểu.

Giới thiệu đề tài

Đề tài liên quan

- Cải tiến: Nesterov Accelerated Descent (NAG).
  - Tính đạo hàm tại (điểm hiện tại + quán tính) để lấy hướng cập nhật tiếp theo rồi mới cộng quán tính vào lượng cập nhật.

Giới thiệu đề tài

Đề tài liên quan: NAG

- Đạo hàm tại hướng dự đoán.
  - Cho biết trước hệ quả khi cập nhật để thực hiện "sửa sai".
  - Ổn định hơn so với Momentum.
  - Hạn chế đi vượt qua các điểm cực tiểu.

Giới thiệu đề tài

Đề tài liên quan: Adaptive

- Hướng tiếp cận mới: adaptive learning rate.
- adaptive: tự điều chỉnh tỷ lệ học tương ứng với từng trọng số.

⇒ Đây là hướng mà chúng em sẽ tập trung tìm hiểu.

Giới thiệu đề tài

Đề tài liên quan: AdaGrad

- Là thuật toán đầu tiên áp dụng tỷ lệ học riêng biệt cho từng trọng số của từng tầng tại từng bước nhảy của mạng nơ-ron.

2.
Kiến thức nền tảng

Kiến thức nền tảng

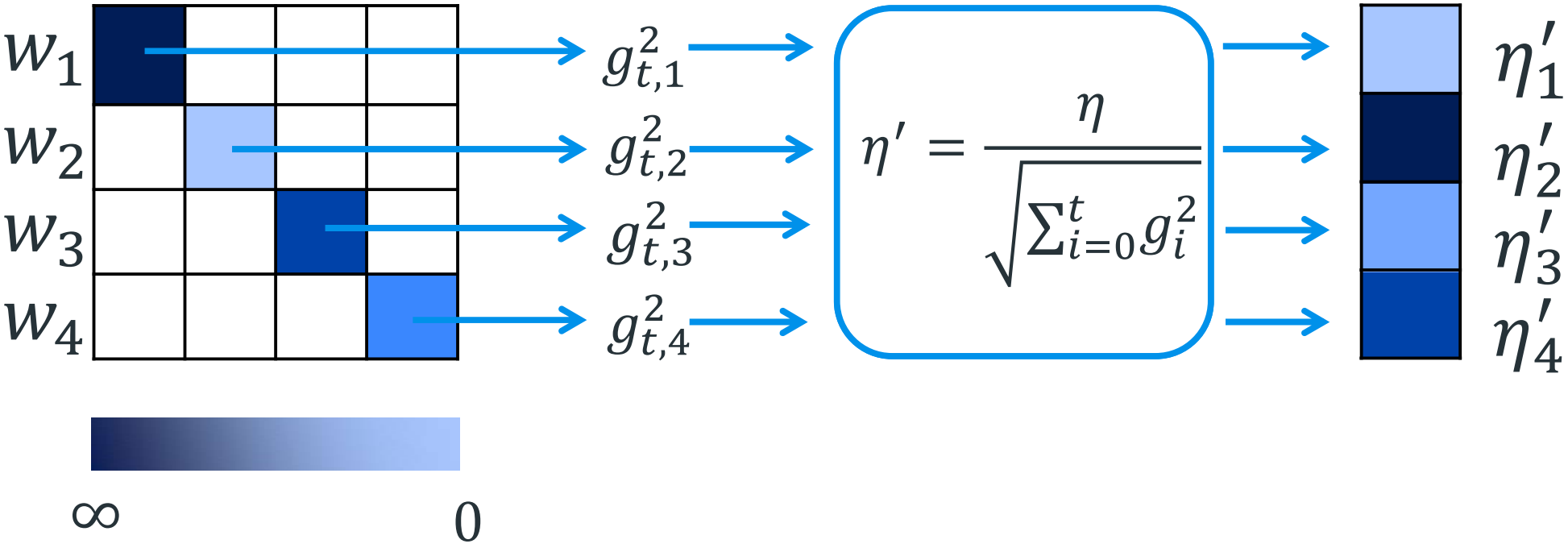
Tối ưu hóa (optimization)

Kiến thức nền tảng

Tối ưu hóa (optimization)

Kiến thức nền tảng

AdaGrad



3.
Thuật toán Adam

Thuật toán Adam

Pseudo-code

Algorithm 1: *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. g_t^2 indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t.

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  while θ_t not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return θ_t (Resulting parameters)

Thuật toán Adam

Siêu tham số

$\boldsymbol{\alpha}$: tỉ lệ học

$\boldsymbol{\beta_1}$, $\boldsymbol{\beta_2}$: tỉ lệ suy biến của trung bình đạo hàm và bình phương đạo hàm (mặc định lần lượt là 0.9 và 0.999)

ϵ: hệ số nhỏ

Thuật toán Adam

Khởi tạo

$\boldsymbol{m_0 = 0}$: khởi tạo trung bình đạo hàm

$\boldsymbol{v_0 = 0}$: khởi tạo trung bình bình phương đạo hàm

$\boldsymbol{t = 0}$: khởi tạo bước chạy

Thuật toán Adam

Các bước thực hiện

Tăng bước chạy t

$$t = t + 1$$

Tính đạo hàm của hàm chi phí trên từng tham số

$$g_t = \nabla_\theta f_t(\theta_{t-1})$$

Thuật toán Adam

Các bước thực hiện

Cập nhật trung bình đạo hàm m_t và trung bình bình phương đạo hàm v_t

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$
$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$$

Tính bias-correction của m_t và v_t

$$\widehat{m_t} = {m_t}/{(1 - \beta_1^t)}$$
$$\widehat{v_t} = {v_t}/{(1 - \beta_2^t)}$$

Cập nhật trọng số

$$\theta_t = \theta_{t-1} - \alpha \cdot {\widehat{m}_t}/{\sqrt{\widehat{v}_t} + \epsilon}$$

4.
Thí nghiệm

5.
Tổng kết