

# **TÓM TẮT TUẦN 2 – NHẬP MÔN HỌC MÁY**

Đại học Quốc gia Thành phố Hồ Chí Minh – Trường Đại Học Khoa Học  
Tự Nhiên

Giảng viên: Bùi Tiến Lên

Sinh viên: Vy Quốc Huy.

MSSV: 22120142

## I. Gradient-Based Learning

Gradient-Based Learning là một phương pháp tối ưu hóa phổ biến trong Machine Learning, trong đó ta sử dụng **gradient descent** để cập nhật các tham số của mô hình sao cho hàm mất mát giảm dần

### 1. cách hoạt động

Gradient Descent là một thuật toán lặp để tìm cực tiểu của một hàm mất mát  $L(\theta)$ .

- Công thức cập nhật tham số:

$$\theta := \theta - \eta \nabla L(\theta)$$

- Trong đó:
  - $\theta$  là tập tham số của mô hình.
  - $\eta$  (learning rate) là tốc độ cập nhật.
  - $\nabla L(\theta)$  là gradient của hàm mất mát.

### 2. Các Biến Thể của Gradient Descent

- **Batch Gradient Descent (BGD):** Sử dụng toàn bộ dữ liệu để tính gradient.
- **Stochastic Gradient Descent (SGD):** Cập nhật tham số sau mỗi mẫu dữ liệu, nhanh nhưng nhiễu.
- **Mini-Batch Gradient Descent:** Kết hợp cả hai bằng cách dùng một nhóm nhỏ dữ liệu mỗi lần cập nhật.

### 3. Gradient-Based Learning trong Machine Learning

- **Hồi quy tuyến tính/logistic:** Dùng gradient descent để tối ưu hàm mất mát (MSE, Cross-Entropy).
- **Mạng nơ-ron (Neural Networks):** Sử dụng thuật toán lan truyền ngược (*backpropagation*) để tính gradient và cập nhật trọng số.
- **Các thuật toán tối ưu nâng cao:** Adam, RMSprop, Momentum giúp cải thiện tốc độ hội tụ.

## II. Computational Graph (Đồ thị tính toán)

Yêu cầu: Tính toán giá trị biểu thức

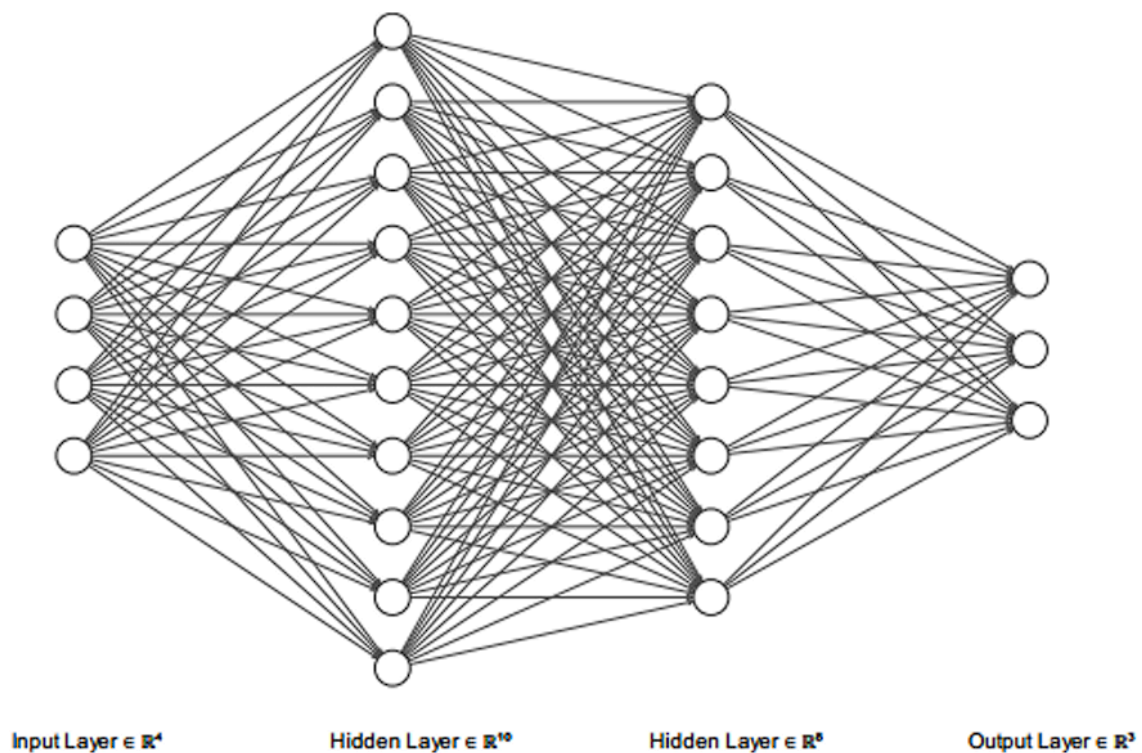
$$e \leftarrow (a+b) * (b+1)$$

Để giải bài toán này, ta có thể chuyển biểu thức trên thành các biểu thức đơn như sau:

1.  $a \leftarrow$
2.  $b \leftarrow$
3.  $c \leftarrow a+b$
4.  $d \leftarrow b+1$
5.  $e \leftarrow c*d$

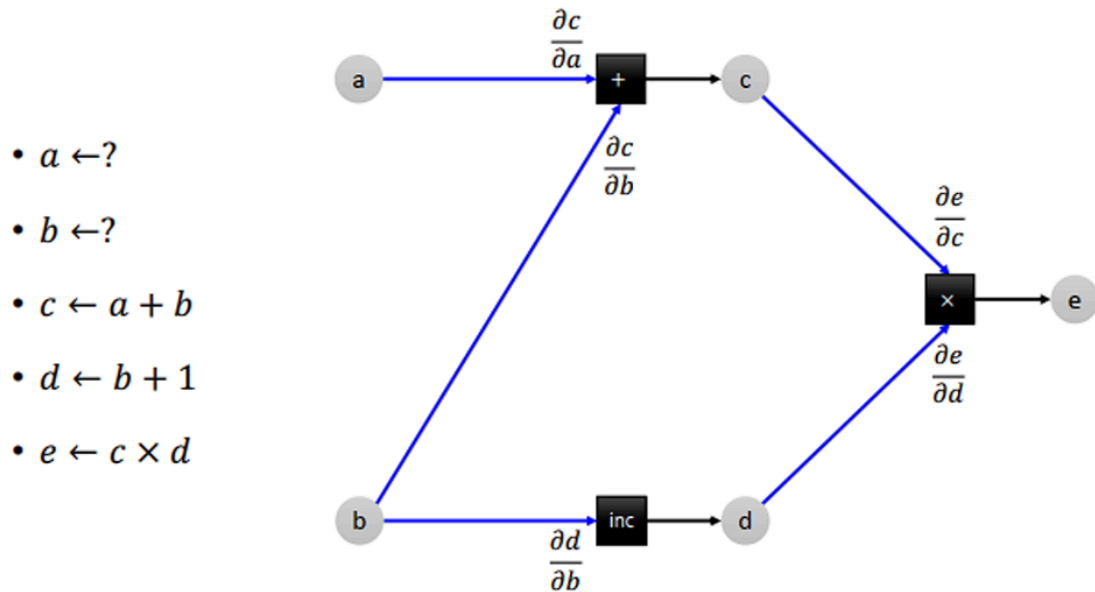
#### 1. Biểu diễn mô hình tính toán

- Đồ thị tính toán (computational graph) là cấu trúc dữ liệu đồ thị DAG (directed acyclic graph) biểu diễn một mô hình tính toán (ví dụ, mạng nơ-ron, mạng Bayes)
- Mỗi đỉnh tương ứng với một biến hoặc tham số
- Mỗi đỉnh tương ứng với một hàm, phép toán, cổng (function, operator, gate, ...)



Hình 2: Đồ thị tính toán

Quay lại bài toán trên, ta có đồ thị tính toán đầy đủ có dạng như hình vẽ



Hình 3: Đồ thị tính toán đầy đủ

Nếu ta thay đổi giá trị đầu vào ta thấy rằng ta không cần phải thay đổi toàn bộ giá trị các đỉnh, ta chỉ cần thay đổi các đỉnh lên quan tới đỉnh a. Điều này thực sự quan trọng trong các mô hình lớn, nơi có tới hàng trăm triệu tới hàng tỷ tham số

## 2. Đạo hàm từng phần

## 3. Các phương pháp tính đạo hàm

- Ba phương pháp chính:
- Đạo hàm số (Numerical Differentiation):
  - Công thức xấp xỉ:

$$\frac{\partial f(x_0)}{\partial x} \approx \frac{f(x_0 + h) - f(x_0)}{h}$$

■

- Nhược điểm: Sai số lớn.
- Đạo hàm ký hiệu (Symbolic Differentiation):
  - Ví dụ:  $\cos(x)' = -\sin(x)$
  - Có thể khó tính toán cho các biểu thức phức tạp.
- Đạo hàm tự động (Automatic Differentiation - AD):
  - Kết hợp cả hai phương pháp trên.

- Được sử dụng trong các framework như TensorFlow, PyTorch.

#### 4. Phương pháp đạo hàm tự động

Dùng quy tắc **chuỗi (chain rule)** để tính toán gradient.

**Ví dụ:** Chuỗi tính toán:

$$x \rightarrow y \rightarrow z \rightarrow w$$

- **Phương pháp đạo hàm tiến (Forward Mode Differentiation)**

- Đi từ biến đầu vào **x** đến biến đầu ra **w**.
- Công thức:

$$\frac{\partial w}{\partial x} = \frac{\partial x}{\partial x} * \frac{\partial y}{\partial x} * \frac{\partial z}{\partial y} * \frac{\partial w}{\partial z}$$

- **Phương pháp đạo hàm lùi (Backward Mode Differentiation - Backpropagation)**

- Đi từ biến đầu ra **w** về biến đầu vào **x**.
- Công thức:

$$\frac{\partial w}{\partial x} = \frac{\partial w}{\partial w} * \frac{\partial w}{\partial z} * \frac{\partial z}{\partial y} * \frac{\partial y}{\partial x}$$

- Đây chính là cơ sở của **Backpropagation** trong mạng nơ-ron.

- **Phương pháp tổng (Summation Rule)**

- Nếu từ **x** tới **y** có nhiều đường, đạo hàm tổng sẽ là tổng của tất cả đạo hàm từng đường đi:

$$\frac{\partial y}{\partial x} = \sum_{i=1}^n \frac{\partial y_i}{\partial x}$$

## **TÀI LIỆU THAM KHẢO:**

1. Slide Nhập Môn Học Máy – Thầy Bùi Tiến Lên.
2. Trang Machine Learning cơ bản: <https://machinelearningcoban.com/>