TÓM TẮT TUẦN 3 – 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. Tensor

1. Khái niệm Tensor

Trong Machine Learning, Tensor là dạng cấu trúc dữ liệu đa chiều được sử dụng để biểu diễn dự liệu. Nó là một khái niệm mở rộng từ các đối tượng toán học quen thuộc như **scalar (vô hướng), vector (một chiều) và matrix (ma trận hai chiều)**. Tensor có thể có nhiều chiều hơn (3D, 4D, hoặc cao hơn), giúp biểu diễn dữ liệu phức tạp hơn.

Các loại Tensor phổ biến:

- Tensor bậc 0 (Scala): một số đơn lẻ, vd x = 5
- Tensor bậc 1 (Vector): một mảng 1D, vd [1,2,3]
- Tensor bậc 2 (Matrix): mảng ma trận m x n, vd $\begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}$

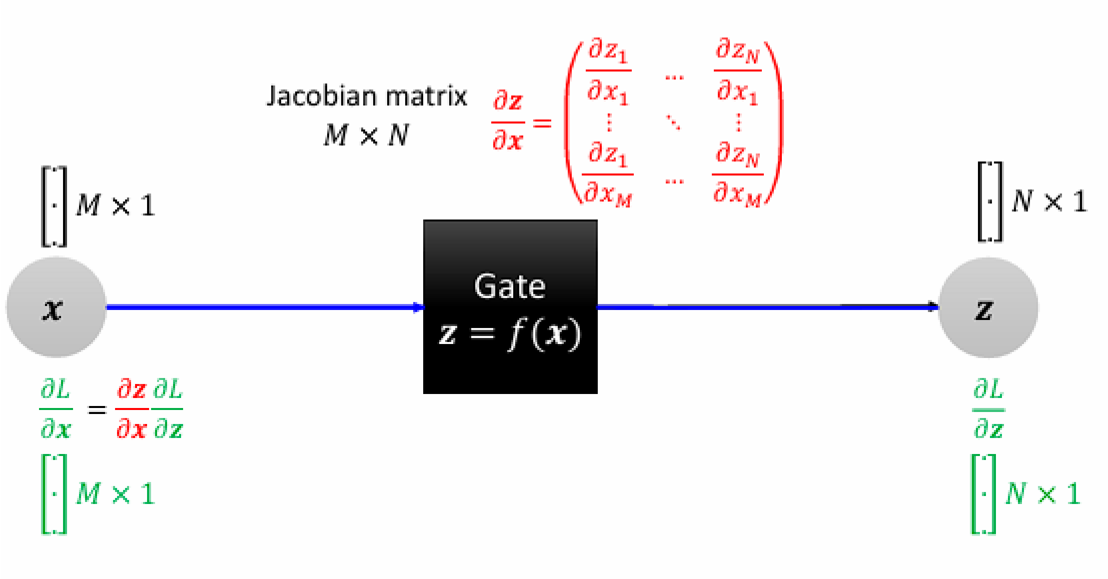
2. Các nhóm phép toán trên Tensor

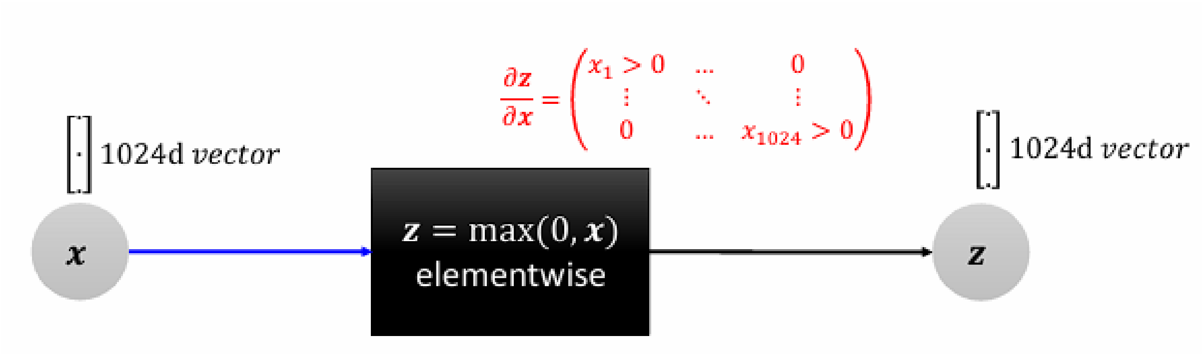
- Nhóm element-wise: cộng, trừ, nhân (hadamard), chia, sigmoid, tanh, cos, sin, ReLU, ...
- Nhóm non-element-wise: nhân ma trận với ma trận, nhân ma trận với vector,

3. Bài toán đạo hàm từng phần

Giả định L là biến mất mát (loss) đầu cuối của đồ thị tính toán. L là biến tensor bậc 0. Bài toán: Tính đạo hàm từng phần (gradient) của L với các biến tensor

4. Nhược điểm của quy tắc nhân





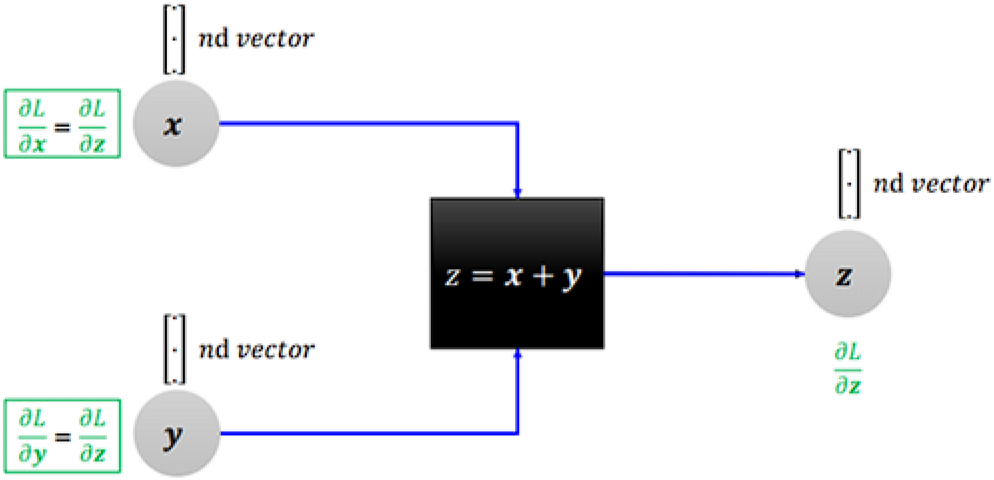
Jacobian matrix có kích thước rất lớn (M x N) chiều nhưng lại rất thưa, khi tính toán sẽ không hiệu quả

II. Một số công thức tính đạo hàm thường gặp

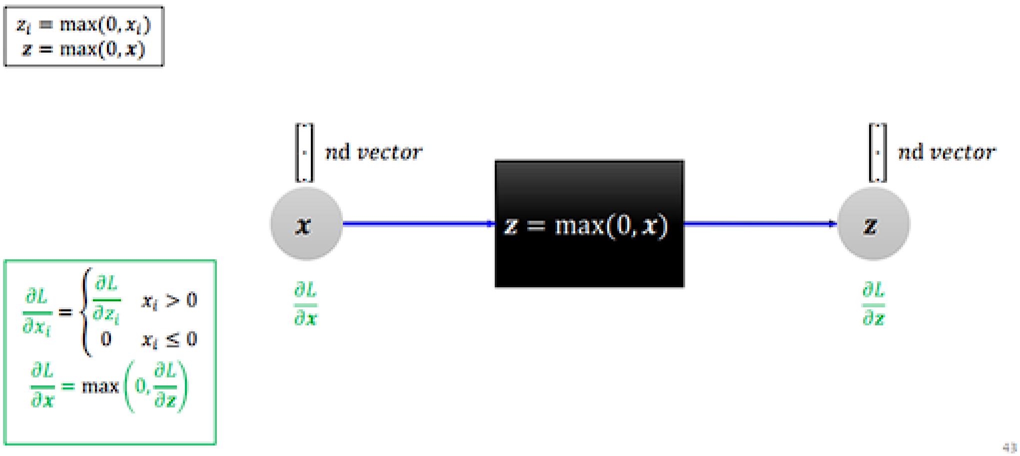
Lưu ý:

- Chỉ sử dụng phương pháp đạo hàm lùi
- Nguyên tắc "nhân"không còn đúng trong trường hợp tổng quát
- Nguyên tắc "cộng"vẫn đúng

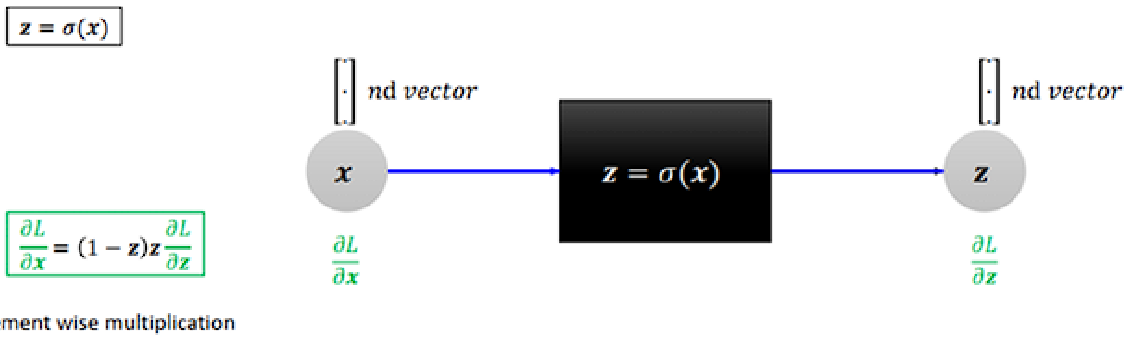
1. **Cổng phép cộng**: z=x+y



2. **Cổng max**: z=max(0,x)



3. **Cổng Sigmoid**: $z = \sigma(x) = \dfrac{1}{1+e^{-x}}$



4. **Cổng L2 (norm)**: $z = L_2 x = \sum_{i=1}^{n} x_i^2$

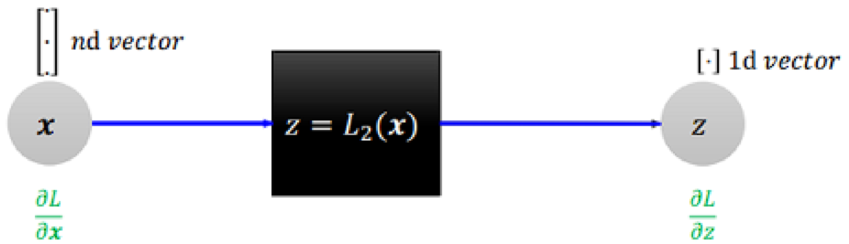
$$\boxed{z = L_2(x) = \sum_{i=1}^{n} x_i^2}$$

Lưu ý: có thể mở rộng phép toán L_2 cho tensor bất kỳ

$$\boxed{\frac{\partial z}{\partial x_i} = 2x_i}$$

$$\boxed{\frac{\partial L}{\partial x_i} = 2x_i \frac{\partial L}{\partial z}}$$

$$\boxed{\frac{\partial L}{\partial x} = 2x \frac{\partial L}{\partial z}}$$

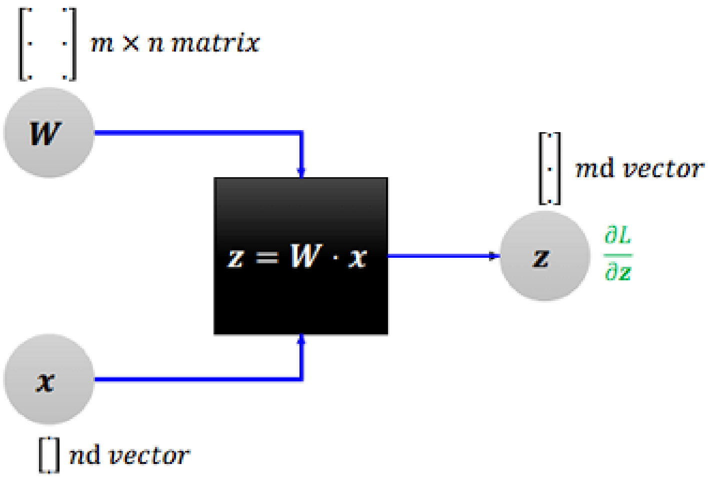


5. **Cổng nhân ma trận với vector**: z=W · x

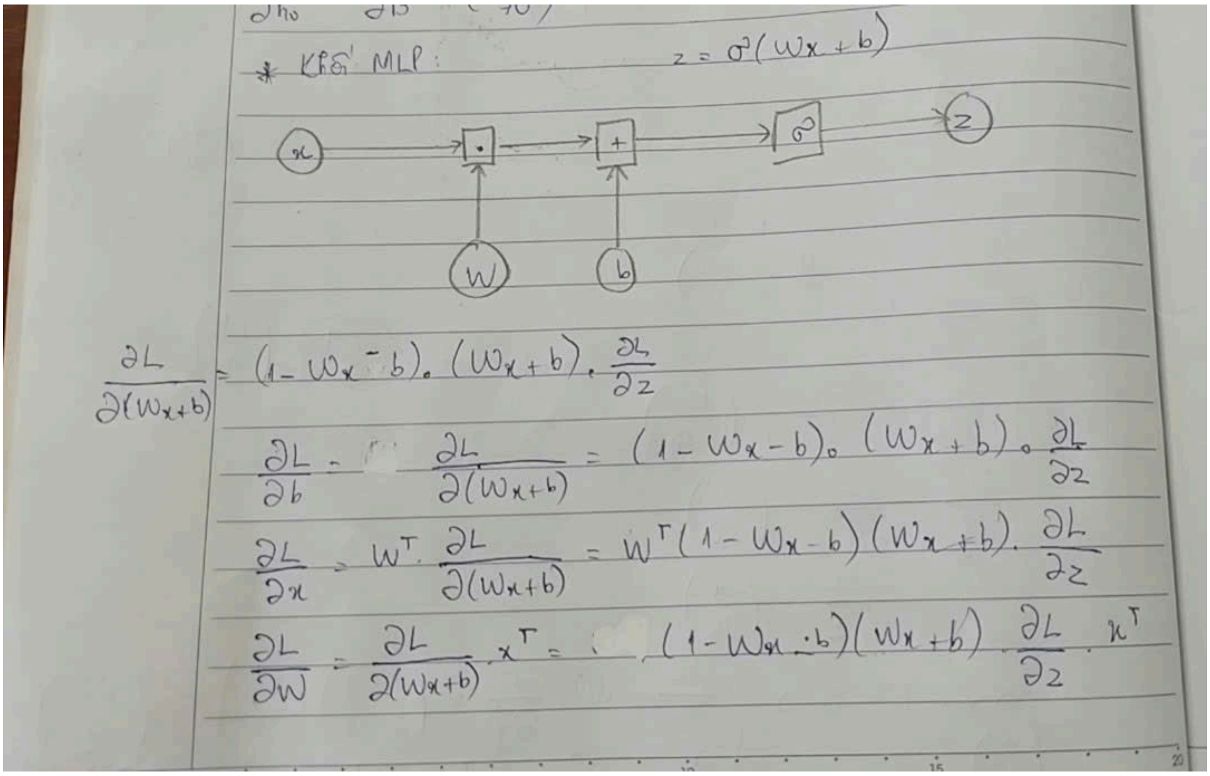
$$\boxed{z = W \cdot x}$$

$$\boxed{\frac{\partial L}{\partial W} = \frac{\partial L}{\partial z} x^T}$$

$$\boxed{\frac{\partial L}{\partial x} = W^T \frac{\partial L}{\partial z}}$$



6. **Tính đạo hàm cho khối phép toán**:
    a. Khối MLP



III. Các biến thể của Hill-Climbing

Hill Climbing (leo đồi) là một thuật toán tối ưu hóa tìm kiếm cục bộ được sử dụng trong trí tuệ nhân tạo và tối ưu hóa. Nó hoạt động bằng cách liên tục di chuyển đến trạng thái lân cận tốt hơn cho đến khi không còn cải thiện được nữa.

Tuy nhiên, do dễ mắc kẹt ở cực trị cục bộ, cao nguyên và sườn dốc nhẹ, có nhiều biến thể khác nhau của thuật toán Hill Climbing để giải quyết các hạn chế này. Dưới đây là các biến thể chính:

- **Các biến thể của thuật toán Hill-Climbing**
    - **Simple Hill Climbing**: Kiểm tra từng trạng thái lân cận một cách tuần tự, dễ mắc kẹt ở cực trị cục bộ.
    - **Steepest-Ascent Hill Climbing**: Chọn trạng thái lân cận tốt nhất, tránh lựa chọn kém nhưng tốn nhiều thời gian hơn.

- **Stochastic Hill Climbing**: Chọn trạng thái ngẫu nhiên thay vì trạng thái tốt nhất để giảm nguy cơ mắc kẹt.
- **First-Choice Hill Climbing**: Chọn trạng thái lân cận tốt đầu tiên gặp phải, giúp tăng tốc nhưng có thể bị mắc kẹt sớm.
- **Random-Restart Hill Climbing**: Chạy nhiều lần từ các điểm khởi tạo khác nhau để tránh cực trị cục bộ.
- **Simulated Annealing**: Kết hợp Hill Climbing với xác suất chấp nhận bước đi kém hơn, giúp tìm kiếm nghiệm tối ưu hơn khi nhiệt độ giảm dần.

TÀI LIỆU THAM KHẢO:

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