

Ghi chép bài giảng: Stanford CS231n

Lecture 2: Image Classification with Linear Classifiers

Ho Hong Phuc Nguyen

25 tháng 1, 2026

Mục lục

1	Bài toán Phân loại hình ảnh (Image Classification)	3
1.1	Góc nhìn của máy tính	3
1.2	Các thử thách chính	3
2	Thuật toán K-Nearest Neighbors (KNN)	3
2.1	Tính toán khoảng cách (Distance Metrics)	3
2.2	Lựa chọn L1 hay L2?	3
2.3	Bất lợi của KNN	3
3	Bộ phân loại tuyến tính (Linear Classifiers)	4
3.1	Mô hình toán học	4
3.2	Hàm mất mát (Loss Function) và Softmax	4
3.2.1	Hàm Softmax	4
3.2.2	Cross-entropy Loss	4

1 Bài toán Phân loại hình ảnh (Image Classification)

Phân loại hình ảnh là nhiệm vụ cốt lõi trong Computer Vision. Hệ thống nhận đầu vào là một hình ảnh và gán cho nó một nhãn (label) từ một tập hợp các nhãn đã định nghĩa trước.

1.1 Góc nhìn của máy tính

Đối với con người, hình ảnh là các thực thể có ý nghĩa, nhưng máy tính chỉ nhìn thấy các khối dữ liệu số (Tensors). Một ảnh màu độ phân giải 800×600 là một ma trận số có kích thước $800 \times 600 \times 3$ (RGB), với mỗi pixel có giá trị từ 0 đến 255.

1.2 Các thử thách chính

Việc lập trình cứng (hard-coding) để phân loại là bất khả thi do các biến số:

- **Viewpoint variation:** Sự thay đổi góc nhìn làm thay đổi toàn bộ giá trị pixel.
- **Illumination:** Điều kiện ánh sáng thay đổi thông số RGB của cùng một đối tượng.
- **Deformation:** Đối tượng thay đổi hình dáng (ví dụ: tư thế của con mèo).
- **Occlusion:** Đối tượng bị che khuất một phần.
- **Background Clutter:** Đối tượng hòa lẫn vào môi trường xung quanh.
- **Intra-class variation:** Sự khác biệt giữa các cá thể trong cùng một lớp (màu sắc, kích thước).

2 Thuật toán K-Nearest Neighbors (KNN)

KNN là phương pháp tiếp cận dựa trên dữ liệu (data-driven approach). Thay vì xây dựng logic cụ thể, nó ghi nhớ tập dữ liệu huấn luyện và tìm các mẫu tương đồng nhất.

2.1 Tính toán khoảng cách (Distance Metrics)

Có hai phương pháp phổ biến để so sánh hai hình ảnh I_1 và I_2 :

- **L1 distance (Manhattan distance):** Tổng giá trị tuyệt đối của sự chênh lệch giữa các pixel:

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

- **L2 distance (Euclidean distance):** Căn bậc hai của tổng bình phương sự chênh lệch:

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$

2.2 Lựa chọn L1 hay L2?

- **Dùng L1:** Khi các đặc trưng đầu vào có ý nghĩa riêng biệt và cần bảo toàn cấu trúc cụ thể của từng đặc trưng.
- **Dùng L2:** Khi các đặc trưng mang tính chất ngẫu nhiên hơn và ranh giới quyết định cần sự mượt mà.

2.3 Bất lợi của KNN

- **Tốc độ dự đoán:** Quá trình test rất chậm ($O(N)$) vì phải duyệt toàn bộ dữ liệu mẫu.
- **Độ chính xác:** Khoảng cách pixel không phản ánh đúng ngữ nghĩa của hình ảnh.
- **Chiều dữ liệu:** Không hiệu quả với hình ảnh có số chiều rất lớn.

3 Bộ phân loại tuyến tính (Linear Classifiers)

Đây là nền tảng của mạng Neural hiện đại, sử dụng phương pháp tham số (Parametric approach).

3.1 Mô hình toán học

Thay vì lưu ảnh mẫu, ta học ma trận trọng số W và vector bias b :

$$f(x, W, b) = Wx + b$$

Trong đó:

- x : Vector hình ảnh đầu vào (kích thước $D \times 1$).
- W : Ma trận trọng số (kích thước $C \times D$, với C là số lớp).
- b : Vector độ lệch giúp dịch chuyển ranh giới quyết định.

3.2 Hàm mất mát (Loss Function) và Softmax

Để tối ưu hóa mô hình, ta cần định nghĩa hàm mất mát để đo lường mức độ sai sót.

3.2.1 Hàm Softmax

Chuyển đổi các điểm số thô thành phân phối xác suất:

$$P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Trong đó s là điểm số đầu ra của hàm tuyến tính.

3.2.2 Cross-entropy Loss

Mục tiêu là tối thiểu hóa giá trị âm logarit của xác suất đúng:

$$L_i = -\log(P(Y = y_i | X = x_i))$$

Nếu mô hình dự đoán xác suất lớp đúng càng cao, Loss sẽ càng tiến về 0.