

ĐẠI HỌC QUỐC GIA TP HCM  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN

---

## Báo cái bài thực hành 2

Đề tài: Hồi quy Softmax cho bài toán nhận diện chữ số viết tay

---

Môn học: Nhập môn học máy

*Sinh viên thực hiện:*

Tô Hữu Danh (23127336)

Nguyễn Đăng Hưng (23127050)

Lê Phú Cường (23127164)

Nguyễn Bá Đăng Khoa (23127392)

*Giáo viên hướng dẫn:*

Thầy Võ Nhật Tân

Ngày 3 tháng 12 năm 2025



# Mục lục

<b>1</b>	<b>Mô tả và tiền xử lý dữ liệu</b>	<b>1</b>
<b>2</b>	<b>Nền tảng toán học và triển khai mô hình</b>	<b>2</b>
2.1	Nền tảng toán học . . . . .	2
2.1.1	Mô hình Softmax Regression . . . . .	2
2.1.2	Hàm mất mát (Loss Function) . . . . .	2
2.1.3	Giải thuật tối ưu Gradient Descent . . . . .	3
<b>3</b>	<b>Thử nghiệm thiết kế thuộc tính</b>	<b>4</b>
<b>4</b>	<b>Đánh giá và phân tích kết quả</b>	<b>5</b>
<b>5</b>	<b>Mô tả ứng dụng</b>	<b>6</b>
<b>6</b>	<b>Kết luận và nhận định</b>	<b>7</b>
<b>7</b>	<b>Tài liệu tham khảo</b>	<b>8</b>

## Danh sách bảng

## Danh sách hình vẽ

## 1 Mô tả và tiền xử lý dữ liệu

## 2 Nền tảng toán học và triển khai mô hình

### 2.1 Nền tảng toán học

#### 2.1.1 Mô hình Softmax Regression

Mô hình *Softmax Regression* được sử dụng để giải quyết bài toán phân loại đa lớp (*multi-class classification*) trên tập dữ liệu MNIST. Với một mẫu đầu vào được biểu diễn bởi vector đặc trưng  $x \in \mathbb{R}^d$  (trong đó  $d$  là số chiều của không gian đặc trưng), mô hình trước tiên tính toán điểm số tuyến tính (logits) tương ứng với  $K$  lớp thông qua phép biến đổi sau:

$$z = Wx + b.$$

Trong đó:

- $W \in \mathbb{R}^{K \times d}$  là ma trận trọng số;
- $b \in \mathbb{R}^K$  là vector bias;
- $z \in \mathbb{R}^K$  là vector điểm số.

Để chuyển các điểm số tuyến tính  $z$  thành một phân phối xác suất hợp lệ trên không gian nhãn, mô hình sử dụng hàm kích hoạt *softmax*. Xác suất để mẫu  $x$  thuộc lớp  $k$  được xác định bởi:

$$\hat{y}_k = P(y = k|x) = \text{softmax}(z_k) = \frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}}.$$

Trong đó,  $z_k$  là điểm số ứng với lớp  $k$ , và mẫu số  $\sum_{j=1}^K e^{z_j}$  đảm bảo rằng tổng các xác suất bằng 1, tạo thành một phân phối xác suất chuẩn hóa trên  $K$  lớp.

#### 2.1.2 Hàm mất mát (Loss Function)

Để huấn luyện mô hình, ta sử dụng hàm mất mát *Cross-Entropy* nhằm đo lường mức độ khác biệt giữa phân phối xác suất dự đoán và nhãn thực tế. Đồng thời, nhằm hạn chế hiện tượng quá khớp (*overfitting*), một thành phần điều chuẩn L2 (*L2 Regularization*) được bổ sung vào hàm mất mát tổng quát. Cụ thể, hàm mất mát được định nghĩa như sau:

$$J(W, b) = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K y_{i,k} \log \hat{y}_{ik} + \frac{\lambda}{2} \sum_l \sum_m W_{lm}^2.$$

Trong đó:

- $N$  là số lượng mẫu trong một batch;
- $y_{i,k}$  là thành phần của vector one-hot label (bằng 1 nếu mẫu  $i$  thuộc lớp  $k$ , ngược lại bằng 0);
- $\lambda$  (ký hiệu là **reg** trong mã nguồn) là hệ số điều chuẩn.

Trong biểu thức trên, hạng tử cross-entropy (hạng tử thứ nhất) thúc đẩy mô hình tối đa hóa xác suất dự đoán đúng, trong khi hạng tử điều chuẩn (hạng tử thứ hai) giúp giới hạn độ lớn của các trọng số, từ đó nâng cao khả năng tổng quát hóa của mô hình.

### 2.1.3 Giải thuật tối ưu Gradient Descent

Quá trình huấn luyện mô hình được thực hiện bằng thuật toán *Mini-batch Gradient Descent*. Ở mỗi bước lặp, ta tính gradient của hàm mất mát đối với các tham số  $W$  và  $b$ . Nhờ đặc tính giải tích thuận lợi của hàm softmax khi kết hợp với hàm mất mát cross-entropy, các biểu thức đạo hàm thu được có dạng đơn giản và hiệu quả như sau:

$$\frac{\partial J}{\partial z} = \hat{y} - y, \quad \frac{\partial J}{\partial W} = \frac{1}{N} X^T (\hat{y} - y) + \lambda W, \quad \frac{\partial J}{\partial b} = \frac{1}{N} \sum (\hat{y} - y).$$

Trong đó  $X \in \mathbb{R}^{N \times d}$  là ma trận dữ liệu đầu vào của mini-batch gồm  $N$  mẫu.

Quy tắc cập nhật tham số với tốc độ học  $\alpha$  (*learning rate*) được thể hiện như sau:

$$W \leftarrow W - \alpha \frac{\partial J}{\partial W}, \quad b \leftarrow b - \alpha \frac{\partial J}{\partial b}.$$

Nhờ việc sử dụng mini-batch, quá trình tối ưu vừa duy trì sự ổn định của gradient, vừa đảm bảo hiệu năng tính toán phù hợp cho các tập dữ liệu lớn.

### 3 Thử nghiệm thiết kế thuộc tính



## 4 Đánh giá và phân tích kết quả

## 5 Mô tả ứng dụng

## 6 Kết luận và nhận định

## 7 Tài liệu tham khảo