[ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM

KHOA CÔNG NGHỆ THÔNG TIN

Logo

Description automatically generated

THỐNG KÊ MÁY TÍNH VÀ ỨNG DỤNG - CQ2018-22

**BÁO CÁO ĐỒ ÁN THỰC HÀNH**

**ĐỒ ÁN 2: SỬ DỤNG LINEAR REGESSION TRÊN TẬP DỮ LIỆU MNIST**

**Thông tin nhóm**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | MSSV | Họ tên | Email |
| 1 | 18120009 | Vương Gia Bảo | 18120009@student.hcmus.edu.vn |
| 2 | 18120374 | Nguyễn Minh Hiếu | 18120374@student.hcmus.edu.vn |

**Phân công công việc**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | MSSV | Họ tên | Nội dung công việc | Hoàn thành |
| 1 | 18120009 | Vương Gia Bảo | Tìm hiểu cách ứng HMM cho bài toán đặt ra, đọc hiểu code và viết báo cáo mục 4,5 | 100% |
| 2 | 18120374 | Nguyễn Minh Hiếu | Tìm hiểu lý thuyết HMM, mô hình HMM cho bài toán đặt ra, viết báo cáo mục 1,2,3 | 100% |

# Tập dữ liệu

MNIST là tập dữ liệu bao gồm các chữ số từ 0-9 viết tay.

Calendar

Description automatically generated

* Tập dữ liệu bao gồm tập Train (60000 ảnh) và Test (10000 ảnh).
* Mỗi ảnh là ảnh xám (1 channel) và kích thước 28x28.
* Giá trị mỗi pixels trong ản nằm trong đoạn [0:255], càng nhỏ thì càng tối và ngược lại.
* Dữ liệu có dáng chuẩn, không bị xoay ngang dọc.

# Tiền xử lý

# Áp dụng Logistic Regression

Phần trước ta đã thực hiện vài biến đổi trên các ảnh, tại đây các kết quả đấy sẽ là input cho mô hình nhằm giúp mô hình học được và gán nhãn chính xác.

Logistic Regression thuộc mô hình học máy có giám sát, là mô hình giúp nhận biết được dữ liệu đã cho có “nằm ở” lớp đó hay không.

Các thành phần:

* Các features của đầu vào **X.** Với mỗi **xi** = [x1, x2,…, xn]
* Hàm phân lớp để tính ŷ = P(y|**x**). Ở đây sẽ dùng sigmoid cho 2 lớp và softmax cho nhiều lớp.
* Hàm đặc biệt để tính toán tham số phù hợp cho mô hình, mục tiêu là cần phải giảm thiểu lỗi nhiều nhất có thể khi huấn luyện. Ở đây sẽ giới thiệu về cross-entrophy loss function.
* Cách tối ưu hóa hàm trên.

Vậy tổng thể cần 2 bước:

* Training: Xác định tham số **w** và bias thích hợp cho mô hình đựa vào cross-entropy loss function và tối hưu hàm đó.
* Test: Đưa một vào **x** để kiểm tra. Ta sẽ tính P(y|**x**) là trả về giá trị P cao hơn với y = 1 hoặc y = 0.

## Sigmoid

Giả sử ta có một vector feature **x** = [x1, x2, …, xn]. Giờ ta muốn xác định P(y = 1 | **x**) và P(y = 0 | **x**). Giống như là xác định 1 comment có phải tích cực hay không. Logistic Regression sẽ sử dụng một vector gồm bias và weight. Weight đại diện cho mức quan trọng của một feature còn bias là tham số giúp phân lớp chính xác hơn. Cuối cùng ta cần tính:

*z* = + b

hay còn viết là: z = **w** *·* **x +** b

Sau đó đưa z vào hàm sigmoid, hàm này biến mọi giá trị x số thực thành giá trị trong đoạn [0,1]

Chart, line chart

Description automatically generated

Sigmoid function:

A picture containing box and whisker chart

Description automatically generated

Giá trị của y có thể dùng làm xác suất:

Text, letter

Description automatically generated

Giờ ta tính được P(y = 1 | **x**)**.** Ta sẽ đưa ra quyết định như sau

Text

Description automatically generated

0.5 ở trên coi như là một mức độ, vượt qua nó thì ŷ sẽ là 1.

## Cross Entropy loss function

Ta muốn dánh giá được được ước lượng ŷ = σ(**w** *·* **x +** b) gần sát với y thực tế (y = 1 hoặc 0) như thế nào thì có thể sử dụng Cross-Entropy loss.

Đầu tiên sử dụng hàm:



y = 1 -> p(y | x) = ŷ

y = 0 -> p(y|x) = 1 - ŷ

Lấy log 2 về:

A picture containing text, watch, clock

Description automatically generated

Đổi dấu để chuyễn thành loss function (một hàm mà ta cần cực tiểu hóa)



Vậy cuối cùng hàm Cross-Entropy loss function sẽ là:



## Tối ưu hàm Cross Entropy bằng thuật toán Gradient Descent

Mục tiêu là cần tìm weight để cực tiểu hóa hàm loss. Ta có dạng sau:

Với θ = w, b

Text

Description automatically generated with medium confidence

Dùng thuật toán Gradient Descent để giải quyết bài toán này. Thuật toán sẽ tìm Gradient của hàm loss tại 1 điểm và di chuyển đến vị trí đối nghịch.

Diagram

Description automatically generated

Trong hình trên, nếu ta muốn thuật toán đi nhanh hay chậm thì sẽ sử dụng tham số learning rate: η



Áp dụng:



Với

Text, letter

Description automatically generated

Ta tính được đạo hàm của hàm log:



Text, letter

Description automatically generated

Trong model được xây dựng với Sklearn, sử dụng [Logistic Regression](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html). Đặt các hypermeter **multi\_class = *multinomial,* solver *= lbfgs***

* **multi\_class: Nếu data set gồm 2 label thì sẽ chọn ‘OVR’ còn nhiều label thì chọn ‘multinominal’. Khi ta đặt là multinominal thì hàm training sẽ là cross-entropy loss.**

**Hàm cross-entropy loss cho data nhiều label có dạng:**

**Graphical user interface

Description automatically generated with low confidence**

**Text

Description automatically generated**

**Với y ở dạng one-hot vector: yi = 1, yj = 0** ∀j ≠ i

* **solver:** Các thuật toán để tối ưu vấn đề, tùy data thì sẽ có thuật toán thích hợp. Nó sẽ giúp giảm thời gian so với thuật toán “tầm thường”, điển hình là ta không cần phải chọn learning rate.

# Kết quả

## Không có tiền xử lý.

Tại đây, chúng ta để mọi thứ tự nhiên, không tác động gì:

* Kết quả:
* Nhận xét:

## Chuẩn hóa giá trị pixel [0:1]

Tại đây, chúng ta chuẩn hóa giá trị pixel về [0:1] để làm giảm sự quá nổi bật của các pixel này:

* Kết quả:
* Nhận xét: