

# Thị Giác Máy Tính

## Báo cáo Lab 2:

### MNIST Handwritten Dataset

Huỳnh Thiết Gia  
20120070@student.hcmus.edu.vn

Ngày 9 tháng 6 năm 2023

#### Tóm tắt nội dung

Đây là báo cáo cho bài thực hành lab 2, chủ đề: phân loại ảnh chữ số viết tay MNIST sử dụng mô hình CNN. Nhóm lấy dữ liệu từ Kaggle, sử dụng thư viện Pytorch và tự xây dựng mô hình để chạy local trên máy tính, không dùng Google Colab. Chương trình đã đạt được điểm số 0.98285 trên bảng xếp hạng của Kaggle.

## 1 Giới thiệu

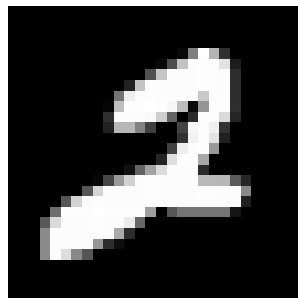
### 1.1 Yêu cầu bài toán

Input: Ảnh trắng đen (Giá trị nằm trong khoảng từ 0 tới 255), kích thước 28x28, tổng 784 điểm ảnh.

Output: Một số từ 0 tới 9 dự đoán lớp mà ảnh đó thuộc về.

### 1.2 Dataset

**MNIST Database (Modified National Institute of Standards and Technology database):** Gồm 2 tập train và test, cả hai đều chứa dữ liệu là ảnh nhưng tập train có gán nhãn đầy đủ còn tập test thì không, người dùng sau khi huấn luyện với tập train



Hình 1: Một ảnh trong tập dữ liệu

```
summary(net.cuda(), (1,28,28))
```

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 40, 26, 26]	400
MaxPool2d-2	[-1, 40, 13, 13]	0
Conv2d-3	[-1, 80, 11, 11]	28,880
MaxPool2d-4	[-1, 80, 5, 5]	0
Conv2d-5	[-1, 160, 3, 3]	115,360
MaxPool2d-6	[-1, 160, 1, 1]	0
Linear-7	[-1, 64]	10,304
Linear-8	[-1, 32]	2,080
Linear-9	[-1, 10]	330

```

Total params: 157,354
Trainable params: 157,354
Non-trainable params: 0

Input size (MB): 0.00
Forward/backward pass size (MB): 0.36
Params size (MB): 0.60
Estimated Total Size (MB): 0.96

```

Hình 2: Cấu trúc của model

xong sẽ thực hiện dự đoán trên tập test và upload kết quả lên Kaggle để so sánh. Dữ liệu được lưu lại ở định dạng file csv, trong đó mỗi dòng có 784 giá trị ngăn cách bởi dấu phẩy, tương ứng với 784 điểm ảnh, riêng tập train có thêm một giá trị ở đầu thể hiện nhãn của ảnh.

### 1.3 Phương pháp đánh giá kết quả

Điểm số trên Kaggle được tính theo phần trăm số ảnh dự đoán đúng, ví dụ 0.97 có nghĩa là 3% ảnh bị dự đoán sai.


## 2 Giải quyết bài toán

### 2.1 Mô hình

Nhóm sử dụng Pytorch, xây dựng model CNN đơn giản với Input và Output như trên. Cấu trúc của model gồm 3 cặp lớp Convolution và MaxPooling, sau đó các Feature Maps sẽ được Flattened và đưa vào 3 lớp Linear.

### 2.2 Các hàm phụ trợ

Các hàm chuyển mảng một chiều từ file csv sang dữ liệu ảnh đầu vào, tạo thành các batch để huấn luyện trên gpu được nhanh hơn, xuất kết quả trên tập test để đẩy lên Kaggle đều là hàng cây nhà lá vườn tự code hết.

Overview	Data	Code	Discussion	Leaderboard	Rules	Team	Submissions	Submit Predictions	...
887	Jakab Dolia						0.98289	10	2mo
888	Gia Huynh						0.98285	5	3d
 <b>Your Best Entry!</b> Your submission scored 0.98285, which is not an improvement of your previous score. Keep trying!									

Hình 3: Vị trí trên bảng xếp hạng Kaggle

### 3 Kết quả

Trên tập huấn luyện, mô hình đạt được độ chính xác là 98%. Trên bảng xếp hạng của cuộc thi MNIST trên Kaggle, mô hình đạt được độ chính xác là 98% luôn. Trong quá trình huấn luyện, nhóm cũng nhận thấy rằng việc huấn luyện trên gpu sẽ nhanh hơn trên cpu rất nhiều lần, chỉ khi nhóm sử dụng batch size lớn để tận dụng khả năng tính toán đa luồng của gpu. Với Batch size là 64, thời gian huấn luyện được cắt giảm đi còn phân nửa so với dùng cpu (111 giây trên cpu so với 60 giây).

### 4 Cấu trúc bài nộp

Vì nhóm không dùng Google Colab nên trong bài nộp sẽ được đính kèm 2 video ngắn thể hiện minh chứng cho quá trình huấn luyện và dự đoán kết quả (Training.mkv và Testing.mkv trong thư mục Release).

Nguyên nhân không dùng Google Colab là vì nhóm không biết cách gọi hàm gì đó để copy dataset từ nguồn ngoài vào môi trường của Colab, còn dùng DataLoaders có sẵn trên Pytorch thì đơn giản quá (Mọi thứ bày sẵn ra, không thử thách gì cả).

### 5 Hướng dẫn chạy

Nhóm không khuyến khích thầy chạy code trên máy thầy vì không đảm bảo môi trường hai máy giống nhau, nhưng nếu thầy muốn thử thì hướng dẫn ở phần dưới.

#### 5.1 Yêu cầu thư viện

Pytorch, Numpy, OpenCv2 (Chỉ khi chạy hàm "Sanity\_check" do hàm phải xuất ra hình ảnh để ta đối chiếu)

#### 5.2 Chạy chương trình

Trong thư mục Sources, chạy file PytorchCode.py, biến use\_gpu cứ để "-1" để chạy trên cpu, nếu máy thầy có gpu và đảm bảo là chạy được thì đổi biến đó thành 0 hoặc lớn hơn tùy vào số thứ tự gpu thầy cần.

## Tài liệu

[https://pytorch.org/tutorials/beginner/blitz/cifar10\\_tutorial.html](https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html)