**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙢🕮🙠-----

**BÁO CÁO BÀI TẬP LỚN THUỘC HỌC PHẦN: KHO DỮ LIỆU VÀ CÁC PHƯƠNG PHÁP KHAI PHÁ**

**ĐỀ TÀI: NHẬN DẠNG CHỮ SỐ TỪ 0 ĐẾN 9 TRONG ẢNH CÓ SẴN SỬ DỤNG MẠNG NƠ RON MLP**

**GVHD: Lê Thị Thủy**

**Nhóm 4:**

1. **Phạm Hoàng Sơn - 2017602027**
2. **Đặng Hoàng Hiệp - 2017603162**
3. **Nguyễn Văn Trường - 2017601659**
4. **Vũ Trí Trung Đông - 2017602525**

**Lớp: KHMT2** - **Khoá: K12**

**Hà Nội – 2021**

**MỤC LỤC**

[**CHƯƠNG 1: TỔNG QUAN VỀ KHO DỮ LIỆU VÀ CÁC PHƯƠNG PHÁP KHAI PHÁ** 3](#_Toc72861800)

[**1. Khái niệm:** 3](#_Toc72861801)

[**1.1. Kho dữ liệu:** 3](#_Toc72861802)

[**1.2. Khai phá dữ liệu:** 3](#_Toc72861803)

[**2. Các bước thực hiện khai phá dữ liệu: (Data Mining)** 4](#_Toc72861804)

[**2.1. Nghiên cứu lĩnh vực:** 4](#_Toc72861805)

[**2.2. Tạo tập dữ liệu đầu vào:** 4](#_Toc72861806)

[**2.3. Tiền xử lý, làm sạch, mã hóa:** 4](#_Toc72861807)

[**2.4. Rút gọn chiều:** 4](#_Toc72861808)

[**2.5. Chọn tác vụ khai phá dữ liệu:** 5](#_Toc72861809)

[**2.6. Chọn các thuật giải khai phá dữ liệu.** 5](#_Toc72861810)

[**2.7. Khai phá tìm kiếm tri thức:** 5](#_Toc72861811)

[**2.8. Đánh giá mẫu tìm được:** 5](#_Toc72861812)

[**2.9. Biểu diễn tri thức:** 5](#_Toc72861813)

[**2.10. Sử dụng tri thức.** 5](#_Toc72861814)

[**3. Ứng dụng: (Data mining)** 5](#_Toc72861815)

[**4. Các công cụ khai phá dữ liệu:** 6](#_Toc72861816)

[**5. Hướng nghiên cứu:** 7](#_Toc72861817)

[**CHƯƠNG 2: CÁC PHƯƠNG PHÁP KHAI PHÁ DỮ LIỆU** 8](#_Toc72861818)

[**1. Tổng quan:** 8](#_Toc72861819)

[**2. Trình bày:** 8](#_Toc72861820)

[**2.1. Thuật toán ID3:** 8](#_Toc72861821)

[**2.2. Thuật toán Naive Bayes:** 9](#_Toc72861822)

[**2.3. Mạng nơ ron:** 11](#_Toc72861823)

[**CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH** 14](#_Toc72861824)

[**1. Thuật toán sử dụng:** 14](#_Toc72861825)

[**2. Mô tả bài toán:** 15](#_Toc72861826)

[**2.1. Giới thiệu:** 15](#_Toc72861827)

[**2.2. Quy trình thực hiện:** 16](#_Toc72861828)

[**KẾT LUẬN** 24](#_Toc72861829)

[**TÀI LIỆU THAM KHẢO** 25](#_Toc72861830)

# **CHƯƠNG 1: TỔNG QUAN VỀ KHO DỮ LIỆU VÀ CÁC PHƯƠNG PHÁP KHAI PHÁ**

## **1. Khái niệm:**

### **1.1. Kho dữ liệu:**

- Kho dữ liệu được dùng để hỗ trợ cho việc xử lý đưa ra các quyết định của người quản lý một dự án nào đó.

- Kho dữ liệu bao gồm các tuyển tập dữ liệu như sau:

+ Dữ liệu có tính hướng theo chủ đề: dữ liệu được tổ chức theo một số chủ đề chính của doanh nghiệp chứ không theo các ứng dụng.

+ Dữ liệu được tích hợp: là kết quả của việc kết hợp dữ liệu từ nhiều nguồn dữ liệu ứng dụng khác nhau với các định dạng khác nhau.

+ Dữ liệu bị biến đổi theo thời gian: chỉ đúng và chính xác tại một thời điểm nào đó. Các cơ sở dữ liệu lưu trữ dữ liệu có giá trị hiện thời.

+ Dữ liệu có tính ổn định: dữ liệu mới sẽ được đưa vào kho dữ liệu như là dữ liệu cung cấp thêm chứ không thay đổi cái đã có.

### **1.2. Khai phá dữ liệu:**

- Khai phá dữ liệu được hiểu là một công việc lấy “chất xám” tri thức một cách tự động và hiệu quả từ một khối dữ liệu lớn.

- Quả trình phát hiện tri thức bao gồm:

+ Làm sạch dữ liệu: những dữ liệu không cần thiết nên được loại bỏ.

+ Tích hợp dữ liệu: tổng hợp những dữ liệu từ nhiều nguồn khác nhau.

+ Lựa chọn dữ liệu: những dữ liệu được tích hợp với nhiệm vụ là phân tích sẽ được “học” từ cơ sở dữ liệu.

+ Chuyển đổi dữ liệu: dữ liệu sau khi được lựa chọn phù hợp sẽ chuyển về dạng hợp nhất cho việc khai phá.

+ Khai phá: là quá trình quan trọng mà trong đó các phương pháp khai phá sẽ được sử dụng nhằm mục đích rút ra các mẫu dữ liệu tiêu chuẩn.

+ Đánh giá: việc phân tích dữ liệu sẽ được dựa trên một số đo quy chuẩn nào đó để có thể xác định độ quan trọng.

+ Biểu diễn tri thức: hiển thị tri thức sẽ được sử dụng để đưa tri thức đã lấy ra từ việc khai phá tìm dữ liệu tiêu chuẩn đến người dùng.

## **2. Các bước thực hiện khai phá dữ liệu: (Data Mining)**

🡪 Quy trình khai phá bao gồm:

### **2.1. Nghiên cứu lĩnh vực:**

- Ta cần phải nghiên cứu các lĩnh vực cần sử dụng tới Data Mining để có thể xác định được những tri thức ta cần chọn lọc tránh mất thời gian cho các tri thức không cần thiết.

### **2.2. Tạo tập dữ liệu đầu vào:**

- Xây dựng một tập tin dữ liệu để lưu trữ đầu vào cho máy tính có thể xử lý chúng.

### **2.3. Tiền xử lý, làm sạch, mã hóa:**

- Tiến hành loại bỏ bớt những dữ liệu không cần thiết, tùy chình là cấu trúc dữ liệu cho phù hợp và mã hóa chúng tiện cho quá trình xử lý.

### **2.4. Rút gọn chiều:**

- Thường thì dữ liệu có chiều lớn sinh ra một khối lượng dữ liệu khổng lồ. Ví dụ với n chiều ta sẽ có 2^n tổ hợp. Vì thế đây là bước quan trọng giúp giảm thiểu đáng kể hao tốn tài nguyên trong quá trình xử lý tri thức.

### **2.5. Chọn tác vụ khai phá dữ liệu:**

- Đặc trưng

- Phân biệt

- Kết hợp

- Phân lớp

- Gom cụm

- Xu thế

- Phân tích độ lệch

- Phân tích hiếm

### **2.6. Chọn các thuật giải khai phá dữ liệu.**

### **2.7. Khai phá tìm kiếm tri thức:**

- Sau khi lựa chọn thuật giải ta sẽ tiến hành khai phá dữ liệu và tím kiếm tri thức.

### **2.8. Đánh giá mẫu tìm được:**

- Cần đánh giá lại trong các tri thức tìm được ta sẽ có những tri thức nào, xem cái nào dư thừa, không cần thiết.

### **2.9. Biểu diễn tri thức:**

- Biểu diễn tri thức thu thập được dưới dạng các hình thức sao cho người dùng có thể hiểu được.

### **2.10. Sử dụng tri thức.**

## **3. Ứng dụng: (Data mining)**

- Có các ứng dụng liên quan tới data mining cụ thể như:

+ Phân tích thị trường và chứng khoán

+ Phân hiện gian lận

+ Quản lý sự rủi ro và phân tích hệ thống doanh nghiệp

+ Phân tích giá trị

+ … còn rất nhiều những ứng dụng khác của data mining

- Lĩnh vực ứng dụng:

+ Lĩnh vực tài chính

+ Lĩnh vực chăm sóc sức khỏe

+ Lĩnh vực viễn thông

+ Marketing và Sales

+ Thương mại điện tử

+ Giáo dục

+ … Còn rất nhiều lĩnh vực khác nữa

## **4. Các công cụ khai phá dữ liệu:**

- RapidMiner: đây là công cụ phổ biến nhất, nó được viết dựa trên nền tảng là ngôn ngữ Java, có nhiều chức năng khai phá như: tiền xử lý data, biểu diễn data, lọc, phân cụm, …

- Weka: là phần mềm mã nguồn mở, không mã hóa và sử dụng GUI đơn giản, có thể gọi trực tiếp thuật toán hoc máy hoặc nhúng mã Java.

- KNime: công cụ này chủ yếu sử dụng cho tiền xử lý dữ liệu.

- Apache Mahout: là phần mở rộng của Big Data Hadoop.

- Oracle Datamining: cho phép người dùng khai phá dữ liệu trên cơ sở dữ liệu SQL để phân tích khung hình và biểu đồ.

- Tera Data: được hiểu như một dịch vụ kho chứa các công cụ khai phá dữ liệu.

- Orange: công cụ này tích hợp các công cụ khai phá dữ liệu và học máy, được viết bằng Python.

## **5. Hướng nghiên cứu:**

- Theo như đề tài của nhóm 4 là nhận dạng chữ số từ 0 đến 9 sử dụng mạng nơ ron MLP, nên dữ liệu đầu vào sẽ phải được xử lý để tìm kiếm tri thức.

- Nhóm sẽ chia hướng nghiên cứu thành các bước như sau:

+ Dữ liệu sẽ được lấy về từ website cung cấp data bao gồm ảnh chứa các số từ 0 đến 9.

+ Bởi vì dữ liệu và số lượng ảnh sẽ là khá lớn nên cần phải qua bước xử lý ảnh để loại bỏ bớt những ảnh chứa các kĩ tự đặc biệt không cần thiết.

+ Sau khi có các dữ liệu đã được khai phá và cần thiết, nhóm tiến hành xây dựng model theo đề tài sử dụng mạng MLP.

+ Tiếp đến tiến hành xây dựng các files như train, test và cuối cùng là nhận dạng được các số có trong ảnh.

# **CHƯƠNG 2: CÁC PHƯƠNG PHÁP KHAI PHÁ DỮ LIỆU**

## **1. Tổng quan:**

- Có 3 kỹ thuật được sử dụng trong khai phá dữ liệu là: luật kết hợp, phân lớp, gom cụm.

+ Kỹ thuật luật kết hợp: có các thuật toán phổ biến như Apriori, FP – growth, …

+ Kỹ thuật phân lớp: thuật toán ID3, C4.5 là dựa trên cây quyết định. Thuật toán Naïve Bayes là dựa trên xác suất và thuật giải mạng nơ ron.

+ Kỹ thuật gom cụm: gom cụm bằng phân hoạch thì có k – means, phân cấp thì có thuật toán gộp.

## **2. Trình bày:**

- Để tài của nhóm có sử dụng đến mạng nơ ron MLP nên phân này sẽ trình bày các thuật toán liên quan tới kỹ thuật phân lớp.

### **2.1. Thuật toán ID3:**

- Là một thuật giải học có giám sát, tạo cây quyết định dựa trên dữ liệu có sẵn, cây kết quả sử dụng cho mẫu mới được gán nhãn.

- Giả sử cho X là tập các vector đặc trưng: X = {x1, x2, …, xn} trong đó xi = (xi1, xi2, …, xim) với i = 1, 2, …, n. C là tập các giá trị phân lớp, c: X 🡪 C thực hiện gán nhãn cho phần tử trong X. D = {(x1, c1(x1)), x2, c2(x2), …, (xn, c(xn)} là X \* C.

- Xây dựng cây quyết định bằng mã như sau:

ID3 (D, Attributes, Target)

{

t = createNode()

if

return(t) endif

if

return(t) rendif

label(t) = mostCommonClass(D, Target)

if

A\* =

For Each a Do

Da ={(x, c(x)) D: x | A\* = a

if Da = THEN

t’ = createNode()

label(t’) = mostCommonClass(D, Target)

createEdge(t, a, t’)

else

createEdge(t, a, ID3(Da, Attributes – {A\*}, Target))

endif

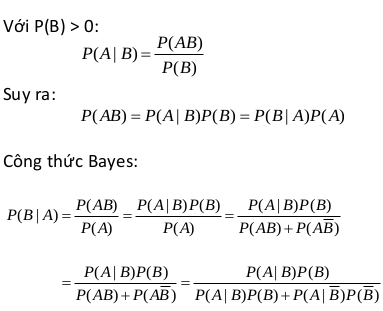
enddo

return(t)

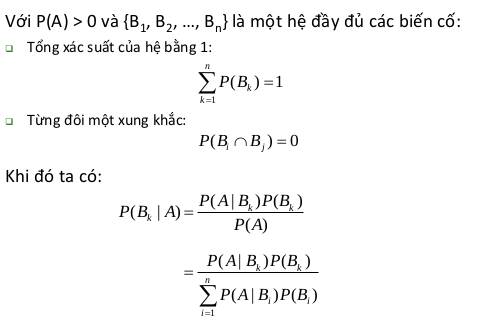
### **2.2. Thuật toán Naive Bayes:**

- Định lí Bayes:

Gọi A, B là 2 biến số:



Công thức Bayes tổng quát:



- Phân lớp Naive Bayes:

Gọi D là tập huấn luyện, trong mỗi phần tử X một vector chứa n giá trị thuộc tính A1, A2, …, An = {X1, X2, …, Xn)

Giả sử có m lớp C1, C2, …, Cm cho mỗi phần tử dữ liệu X, cụ thể

P (Cj | X) > P (Cj | X) (1<=i, j <= m, I != j)

Để tìm xác suất lớn nhất, các giá trị P(X) không cần tính vì giống nhau với mọi lớp. Nên chỉ cần tìm giá trị lớn nhất của P(X|Ci) \* P(Ci). (P(Ci) được ước lượng bằng |Di|/|D|), trong đó Di là tập các phần tử dữ liệu thuộc lớp Ci. Nếu xác suất P(Ci) không xác định được thì ta coi chúng bằng nhau P(C1) = P(C2) = ... = P(Cm), khi đó chỉ cần tìm giá trị P(X|Ci) lớn nhất.

Khi số lượng các thuộc tính mô tả dữ liệu là lớn thì chi phí tính toàn P(X|Ci) là rất lớn, dó đó có thể giảm độ phức tạp của thuật toán Naive Bayes giả thiết các thuộc tính độc lập nhau. Khi đó ta có thể tính:  
P(X|Ci) = P(x1|Ci) ...P(xn|Ci).

### **2.3. Mạng nơ ron:**

**2.3.1. Tổng quan mạng nơ ron:**

- Mạng nơron nhân tạo (artificial neural networks) là một hệ thống xử lý song song và phân tán cỡ lớn. Nó hình thành nên một kỹ thuật tính toán mới được xây dựng dựa vào sự bắt chước việc xử lý thông tin của con người.

- Các ứng dụng ban đầu của mạng nơron đã chứng minh nó là công cụ mạnh cho phân tích tín hiệu, trích chọn đặc trưng, phân lớp dữ liệu, nhận dạng… Do có khả năng học và khái quát hóa từ dữ liệu thu thập nên mạng nơron được chấp nhận rộng rãi bởi các nhà nghiên cứu và các kỹ sư như là công cụ để xử lý dữ liệu thực nghiệm.

- Mạng nơron nhân tạo được sử dụng rộng rãi trong thực tế do chúng có một số đặc điểm chính sau:

+ Khả năng tạo ra ánh xạ phi tuyến giữa một tập dữ liệu quá khứ với dữ liệu tương lai trong chuỗi dữ liệu

+ Khả năng xấp xỉ hàm số. Cho phép sử dụng mạng nơron để mô hình hóa cho một hàm phi tuyến liên tục bất kỳ với độ chính xác tùy ý

+ Khả năng học và khái quát từ dữ liệu mẫu.

- Mỗi nơron sinh học có các thành phần chính như hình dưới đây:



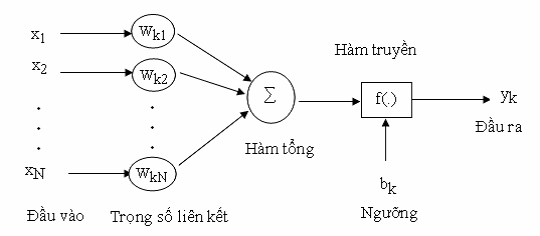
Hình 1: Cấu tạo của nơron sinh học

+ Cell body chứa nhân

+ Dendrites dùng để truyền tín hiệu từ nơron lân cận đến cell body

+ Axon dùng để truyền tín hiệu từ cell body này đến cell body của nơron khác.

- Từ cấu tạo trên ta có thể mô hình hóa nơron nhân tạo như hình dưới:



Hình 2: Mô hình tổng quan nơron nhân tạo

**2.3.2. Hàm truyền đạt:**

- Hàm truyền đạt có thể là tuyến tính hoặc phi tuyến, việc chọn hàm truyền phù hợp cho nơron tùy thuộc vào mục đích sử dụng để giải quyết những vấn đề cụ thể.

- Một số hàm truyền thường được sử dụng là:

+ Hàm truyền hard limit (giới hạn cứng):

f(n) =

+ Hàm truyền đạt tuyến tính (linear transfer function):

f(n) = n

+ Hàm truyền đạt log-sigmoid:

f(n) =

+ Hàm truyền đạt tangenet-sigmoid: f(n) =

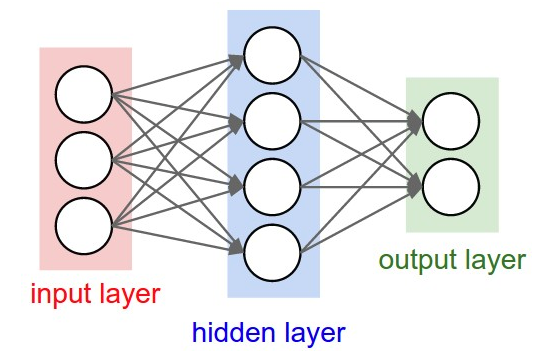
# **CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH**

## **1. Thuật toán sử dụng:**

- Ở đề tài này nhóm chọn mạng nơ ron nhân tạo MLP (multilayer perceptron) để nhận dạng chữ số từ 0 đến 9 có trong ảnh.

- Cấu trúc mạng nơ ron:

+ Mạng nơron nhân tạo là sự kết hợp của các tầng perceptron hay còn gọi là perceptron đa tầng (MLP – multilayer perceptron) như hình dưới:



Hình 3: Cấu trúc mạng nơ ron nhân tạo

- Một mạng nơron nhân tạo sẽ bao gồm 3 kiểu lớp:

+ Tầng vào (input layer): thể hiện cho các đầu vào của mạng

+ Tầng ẩn (hidden layer): là tầng nằm giữa tầng vào và tầng ra thể hiện cho việc suy luận logic của mạng

+ Tầng ra (output layer): thể hiện cho các đầu ra của mạng.

🡪1 mạng nơron nhân tạo chỉ có 1 tầng vào và 1 tầng ra nhưng có thể có nhiều tầng ẩn.

## **2. Mô tả bài toán:**

### **2.1. Giới thiệu:**

**9**

**1**

**0**

**128**

**W1**

**W2**

**W3**

**Input**

**Hidden 1**

**Hidden 2**

**Ouput**

Sigmoid

Linear

Softmax

**56**

**10**

Hình 4: Mô hình cấu trúc mạng

- Mô tả: cấu trúc mạng gồm các lớp:

+ Tầng input: là một ảnh dữ liệu đầu vào đã được chuẩn hóa về kích thước 28x28

+ Lớp hidden 1: là một lớp gồm 128 nơron lấy đầu vào từ lớp input với hàm truyền là hàm sigmoid

+ Lớp hidden 2: là một lớp gồm 56 nơron lấy đầu vào là đầu ra của lớp hidden 1 với hàm truyền là hàm linear

+ Lớp output: là một lớp gồm 10 nơron theo 10 số từ 0-9, sử dụng hàm SoftMax

- Bộ kiểm tra: để thực hiện khảo sát với các lớp hidden có số nơron khác nhau thì độ chính xác sẽ khác nhau như thế nào:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | input | Hidden 1 | Hidden 2 | output | Độ chính xác | epochs |
| 0 | 784 | 128 | 56 | 10 | 96.2226% | 50 |
| 1 | 784 | 100 | 30 | 10 | 96.3331% | 50 |
| 2 | 784 | 110 | 40 | 10 | 96.3504% | 50 |
| 3 | 784 | 120 | 47 | 10 | 96.2956% | 50 |
| 4 | 784 | 125 | 50 | 10 | 96.4416% | 50 |
| 5 | 784 | 20 | 10 | 10 | 93.4% | 50 |
| 6 | 784 | 512 | 128 | 10 | 96.04% | 50 |

🡪 Qua bảng trên ta thấy sau khi huấn luyện với số nơron khác nhau ở các lớp khác nhau ở các lớp khác nhau thì độ chính xác không quá khác biệt nhưng tốc độ đạt được độ chính xác cao nhất không giống nhau.

### **2.2. Quy trình thực hiện:**

- Gồm 4 giai đoạn chính: tiền xử lý, xây dựng model, huấn luyện (train+test), nhận dạng.

- Sử dụng ngôn ngữ lập trình: Python 3.7

- Cơ sở dữ liệu (dataset):

+ Nhóm em sử dụng tập dữ liệu trên trang web <https://kaggle.com/data> để làm cơ sở dữ liệu đánh giá hệ thống. Dữ liệu huấn luyện bao gồm hơn 25000 bức ảnh. Tất cả hình ảnh đều là đen trắng có in số từ 0 đến 9 có kích thước nhỏ.

+ Nguồn download dataset:

Link: <https://www.kaggle.com/moghya/numbers-and-plusminus-in-google-fonts>

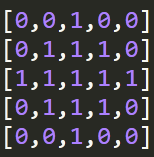
**2.2.1. Tiền xử lý:**

- Ngôn ngữ lập trình: Python

- Thư viện sử dụng: OpenCv, Os, Tensorflow, Keras

- Công việc:

B1: Dùng các phép biến đổi hình thái để loại bỏ khoảng trống nhỏ có thể có trong ảnh trước khi bắt đầu xử lý. Ở đây sử dụng phép Closing với mặt nạ hình thoi 5x5.



B2: Làm mờ ảnh bằng gaussian để tăng độ mịn các biên.

B3: Chuyển sang nhị phân (0-255)

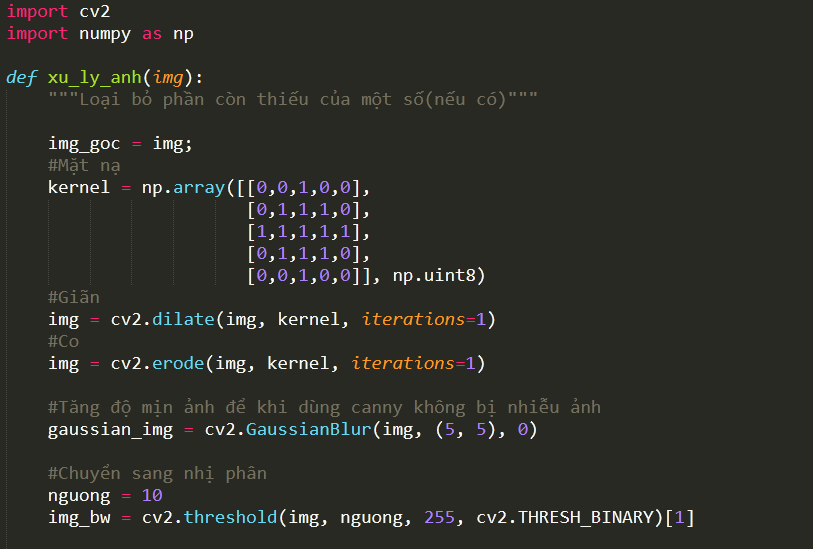
B4: Tìm các contour có trong ảnh. Contours ở đây là một đường cong liên kết toàn bộ các điểm liên tục mà có màu sắc hoặc giá trị cường độ giống nhau.

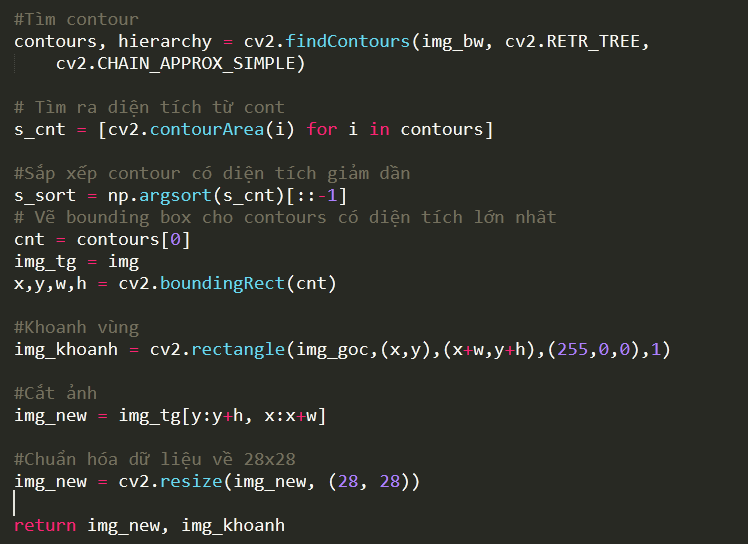
B5: Dùng hàm contourArea() để lấy được các diện tích của các tontour đó vì lúc này contour chỉ là nhưng điểm ảnh không thể sử dụng để xác định vật thể.

B6: Tính x, y, chiều cao, chiều rộng từ diện tích các contour đã có dùng hàm boundingRect()

B7: Cắt ảnh theo x,y,w,h và chuẩn hóa về kích thước 28x28.

**Code chương trình:**





**2.2.2. Xây dựng model:**

- Dataset: gồm dữ liệu vào đã được chuẩn hóa về kích thước 28x28

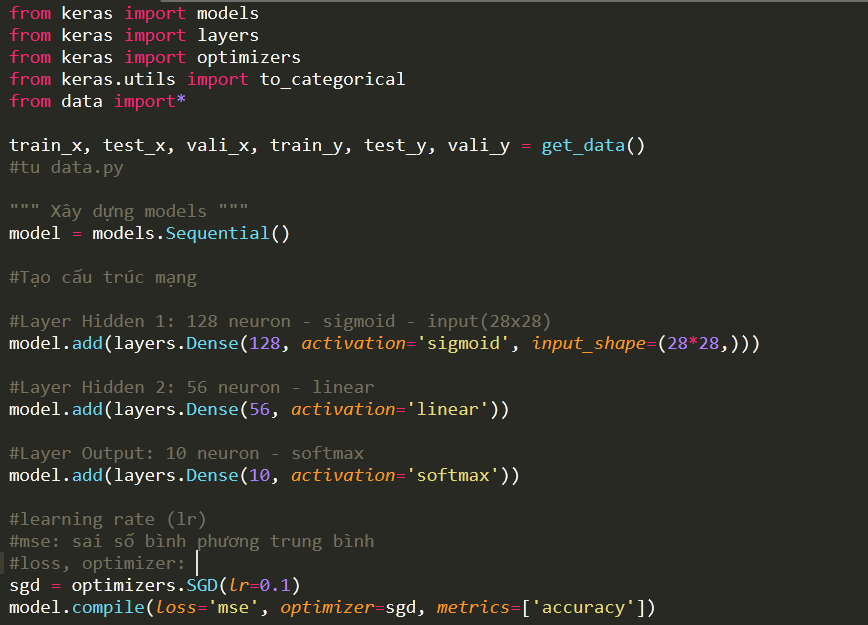
- Mô tả:

+ Xây dựng model MLP bằng thư viện keras

+ Model gồm 4 tầng: input – hidden – hidden - ouptut theo cấu trúc mạng.

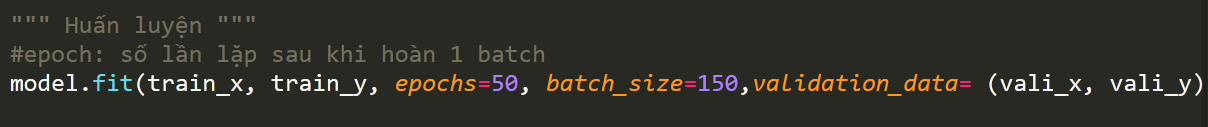
+ Learning rate model là 0.1

**Code chương trình:**

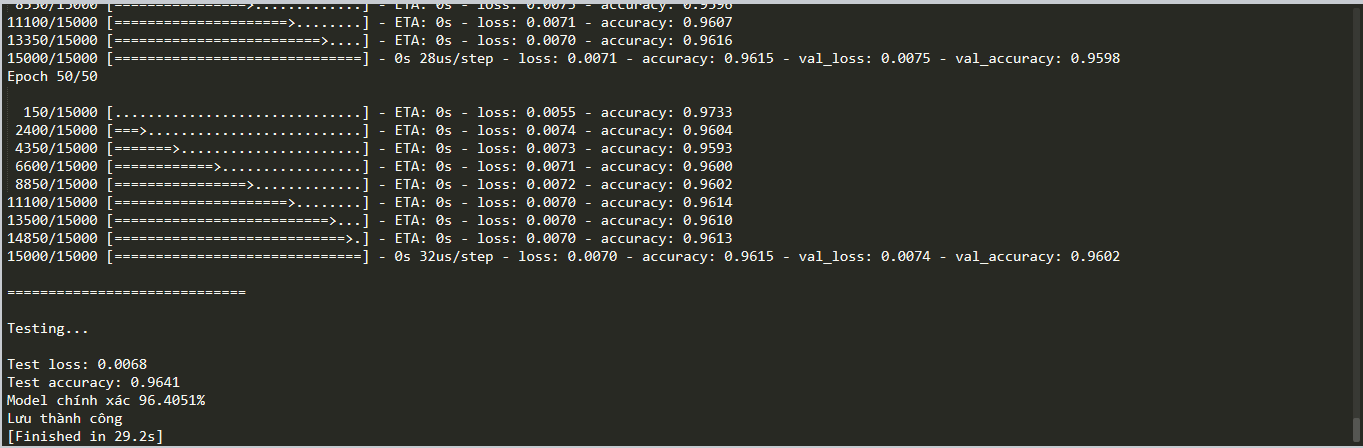


**2.2.3. Huấn luyện:** (train + test)

- Train:



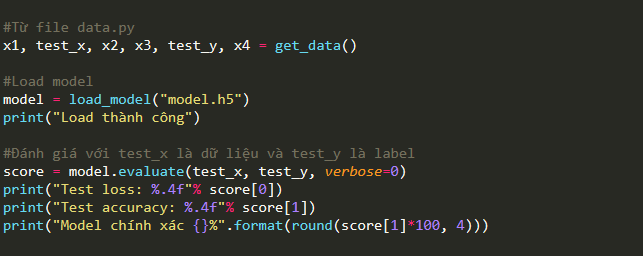
🡪 Kết quả sau khi train:



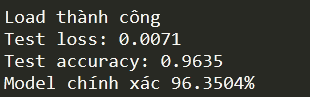
🡪 Ta thấy model có độ chính xác là 96.4051% với dữ liệu train là 95.74% với dữ liệu validation sau 50 lần lặp.

- Test:

🡪 Lấy model từ file model.h5 và test



🡪 Kết quả:



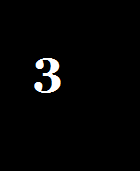
🡪 Sau khi huấn luyện ta tiến hành test lại với số nơron các lớp như ở trên và có độ chính xác 96.3504%.

**2.2.4: Nhận dạng:**

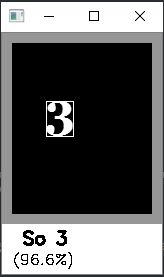
- Kết quả: Nhận dạng thành công được hầu hết các ảnh đầu vào và in ra label số đó là gì

- Ví dụ: Truyền vào 1 hình ảnh có chứa số và khoanh vùng, nhận dạng số có trong ảnh đó:

+ Input: 1 ảnh đầu vào



+ Output: sau khi xử lý tSa nhận dạng được ảnh như hình:



=> Có thể nhận dạng ra ảnh số 3 này với độ chắc chắn 96.6%.

# **KẾT LUẬN**

Thông qua đề tài bài tập lớn này, nhóm chúng em đã rút ra được một số kinh nghiệm sau:

- Hiểu được khái niệm về mạng nơron nhân tạo MLP, mô hình và cách thực hiện của mạng nơron nhân tạo MLP.

- Hiểu được cách sử dụng ngôn ngữ lập trình Python trong ứng dụng vào bài toán nhận dạng

- Hiểu được quá trình thực hiện bài toán nhận dạng

- Quan trọng của quá trình là phải có được một dữ liệu đủ tốt để thực hiện nhận dạng

- Học được cách phối hợp cùng các thành viên khác trong nhóm trong quá trình thực hiện bài tập lớn.

Trong quá trình làm bài tập lớn, tuy gặp nhiều khó khăn trong việc tìm hiểu và thực hiện, nhóm chúng em đã hoàn thiện đề tài đúng thời hạn một cách hiệu quả và nhanh chóng. Ngoài ra chúng em còn rút ra được một số kinh nghiệm như và kiến thức như trên. Chúng em xin cảm ơn thầy đã giúp chúng em trong quá trình làm bài tập lớn một cách thuận lợi và có hiệu quả.

# **TÀI LIỆU THAM KHẢO**

[1]. Giáo trình Khai phá dữ liệu – Trường ĐH Công Nghiệp HN

[2]. https://phamdinhkhanh.github.io/2020/01/06/ImagePreprocessing.html

[3]. <https://viblo.asia/p/xu-ly-anh-erosion-dilation-opening-closing-4dbZNpWq5YM>

[4]. <https://machinelearningmastery.com/how-to-load-large-datasets-from-directories-for-deep-learning-with-keras/>

[5]. [https://machinelearningmastery.com/tutorial-first-neural-network-python- keras/](https://machinelearningmastery.com/tutorial-first-neural-network-python-%20%20%20%20%20%20keras/)

[6]. <https://towardsdatascience.com/loading-custom-image-dataset-for-deep-learning-models-part-1-d64fa7aaeca6>

[7]. <http://itechseeker.com/tutorials/tensorflow/xay-dung-mo-hinh-deep-learning-voi-keras/>

[8]. <https://machinelearningcoban.com/2018/07/06/deeplearning/>