**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THÍ NGHIỆM / THỰC NGHIỆM**

**HỌC PHẦN: TRÍ TUỆ NHÂN TẠO**

**Đề tài: Nhận dạng chữ số in trên bằng tốt nghiệp**

**Giảng viên : ThS.Nguyễn Lan Anh**

**Lớp : 20221IT6043004**

**Nhóm : Nhóm 24**

**Sinh viên thực hiện : Dương Ngọc Anh - 2020600274**

**Nguyễn Thành Lâm - 2020601227**

***Hà Nội, năm 2022***

# LỜI NÓI ĐẦU

AI - Artificial Intelligence (Trí Tuệ Nhân Tạo), và cụ thể hơn là Machine Learning (Máy Học) nổi lên như một minh chứng của cuộc cách mạng công nghiệp lần thứ tư. AI hiện diện trong mọi lĩnh vực của đời sống con người, từ kinh tế, giáo dục, y khoa cho đến những công việc nhà, giải trí hay thậm chí là trong quân sự. Những ứng dụng nổi bật trong việc phát triển AI đến từ nhiều lĩnh vực để giải quyết nhiều vấn đề khác nhau. Nhưng những đột phá phần nhiều đến từ Deep Learning (học sâu) - một mảng nhỏ đang mở rộng dần đến từng loại công việc, từ đơn giản đến phức tạp. Deep Learning đã giúp máy tính thực thi những việc tưởng chừng như không thể vào 15 năm trước: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người, hay thậm chí cả sáng tác văn, phim, ảnh, âm nhạc.

Trong quá trình thực hiện đề tài nhóm chúng em xin gửi lời cám ơn chân thành tới cô Nguyễn Lan Anh - giảng viên hướng dẫn nhóm chúng em thực hiện đề tài này. Trong quá trình nghiên cứu và thực hiện đề tài được sự chỉ bảo tận tình của cô, nhóm chúng em đã cố gắng hết sức để hoàn thiện đề tài. Tuy nhiên chúng em rất mong nhận được sự góp ý của cô và các bạn.

Nhóm em xin trân trọng cảm ơn!

**MỤC LỤC**

[LỜI NÓI ĐẦU 2](#_Toc123065204)

[CHƯƠNG 1: GIỚI THIỆU VÀ MỤC TIÊU CỦA ĐỀ TÀI 5](#_Toc123065205)

[1.1 Giới thiệu đề tài 5](#_Toc123065206)

[1.2 Mục tiêu của đề tài 5](#_Toc123065207)

[1.3 Cách tiếp cận và phương pháp nghiên cứu 5](#_Toc123065208)

[1.4 Kết quả dự kiến đạt được 5](#_Toc123065209)

[CHƯƠNG 2: TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO 6](#_Toc123065210)

[2.1 Khái niệm về Trí tuệ nhân tạo 6](#_Toc123065211)

[2.1.1 Hành động như người: Turning test 6](#_Toc123065212)

[2.1.2 Suy nghĩ như người 6](#_Toc123065213)

[2.1.3 Suy nghĩ có lí trí 7](#_Toc123065214)

[2.1.4 Hành động có lí trí 7](#_Toc123065215)

[2.2 Các lĩnh vực liên quan đến TTNT 7](#_Toc123065216)

[2.3 Lịch sử hình thành khoa học TTNT 7](#_Toc123065217)

[2.4 Các lĩnh vực ứng dụng của TTNT 8](#_Toc123065218)

[2.5 Một số ví dụ về TTNT 9](#_Toc123065219)

[2.6 Những vấn đề chưa được giải quyết 9](#_Toc123065220)

[CHƯƠNG 3: TỔNG QUAN VỀ MẠNG NƠ-RON 10](#_Toc123065221)

[3.1 Giới thiệu về Neural network 10](#_Toc123065222)

[3.2 Mô hình Neural network 11](#_Toc123065223)

[3.2.1 Logistic regression 11](#_Toc123065224)

[3.2.2 Mô hình tổng quát 13](#_Toc123065225)

[3.2.3 Kí hiệu 14](#_Toc123065226)

[3.3 Ảnh trong máy tính 14](#_Toc123065227)

[3.4 Convolutional neural network (CNN) 22](#_Toc123065228)

[CHƯƠNG 4: XÂY DỰNG CHƯƠNG TRÌNH 27](#_Toc123065229)

[4.1 Chuẩn bị dữ liệu 27](#_Toc123065230)

[4.2 Thực hiện chương trình 28](#_Toc123065231)

[4.3 Một số hạn chế khi chạy chương trình 41](#_Toc123065232)

[CHƯƠNG 5: TỔNG KẾT 44](#_Toc123065233)

[5.1 Tài liệu tham khảo 44](#_Toc123065234)

[5.2 Phụ lục 44](#_Toc123065235)

# CHƯƠNG 1: GIỚI THIỆU VÀ MỤC TIÊU CỦA ĐỀ TÀI

## 1.1 Giới thiệu đề tài

Với mục tiêu phát triển nền công nghiệp hiện đại hóa, tự động hóa, nhóm chúng em đã lên ý tưởng *“Nhận dạng chữ số in trên bằng tốt nghiệp”* nhằm tránh việc tìm kiếm thủ công dễ gây nhầm lẫn thông tin bằng tốt nghiệp gây ra các sai sót ngoài ý muốn, nhận dạng chữ số sẽ giúp tìm kiếm dễ dàng hơn với các thuật toán có độ chính xác cao, đem đến cho con người 1 cái nhìn mới về Trí tuệ nhân tạo (AI) và cải thiện sức lao động và trí óc của con người.

## 1.2 Mục tiêu của đề tài

* Xây dựng được một chương trình để nhận dạng các chữ số in trên bằng tốt nghiệp, sau đó chuyển đổi chúng thành các dạng số kĩ thuật trên máy, giúp tạo ra 1 danh sách tự động, giảm thiểu việc viết bằng tay và nhận dạng bằng mắt thường.
* Sử dụng các thuật toán đã được học trong môn Trí tuệ nhân tạo để nhận dạng hình ảnh các con số để chuyển đổi thành số kĩ thuật số.

## 1.3 Cách tiếp cận và phương pháp nghiên cứu

* Tìm hiểu và viết báo cáo về nhận dạng chữ số.
* Nghiên cứu các thuật toán nhận dạng chữ số.
* Tìm hiểu và chọn một ngôn ngữ lập trình để lập trình thuật toán nghiên cứu được.
* Phân tích, thiết kế và ứng dụng nhận dạng chữ số trong việc tìm kiếm thông tin.

## 1.4 Kết quả dự kiến đạt được

* Xây dựng thành công chương trình nhận dạng và chuyển đổi số in hoặc viết tay sang số kĩ thuật số.

# CHƯƠNG 2: TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO

## 2.1 Khái niệm về Trí tuệ nhân tạo

* Hiện nay, trên thế giới có nhiều định nghĩa khác nhau về trí tuệ nhân tạo. Tuy nhiên, vẫn chưa thống nhất một dạng định nghĩa.
* Mặc dù vậy, có 2 trường phái về khái niệm AI:
* Strong AI: Có thể tạo ra thiết bị có trí thông minh và các chương trình máy tính thông minh hơn người.
* Weak AI: Chương trình máy tính có thể mô phỏng các hành vi thông minh của con người.
* Có 4 quan điểm về AI:
* Suy nghĩ như người
* Suy nghĩ có lí trí
* Hành động như người
* Hành động có lí trí

### 2.1.1 Hành động như người: Turning test

* Turing (1950) "Computing machinery and intelligence":
* “Máy tính có thể nghĩ?" => “Máy tính có thể hành động thông minh?"
* Turing Test: Trò chơi bắt chước người.
* Ưu điểm của Turing Test:
* Khái niệm khách quan về trí tuệ
* Tránh đi những thảo luận về quá trình bên trong và ý thức
* Loại trừ định kiến thiên vị của người thẩm vấn
* Các ý kiến phản đối Turing Test
* Thiên vị các nhiệm vụ giải quyết vấn đề bằng ký hiệu.
* Trói buộc sự thông minh máy tính theo kiểu con người, trong khi con người có: Bộ nhớ giới hạn, có khuynh hướng nhầm lẫn
* Tuy nhiên, trắc nghiệm Turing đã cung cấp một cơ sở cho nhiều sơ đồ đánh giá dùng thực sự cho các chương trình TTNT hiện đại.

### 2.1.2 Suy nghĩ như người

* Hai cách tiếp cận:
* Trên xuống: Tâm lý học nhân tạo Symbolism (Simon & Newell, 1961).
* Dưới lên: Neural and Brain Science (Mc Culloch, Pitt 1950s) => Artificial Neural Networks.

### 2.1.3 Suy nghĩ có lí trí

* Bắt đầu từ thời Hylạp cổ đại (Rule of Arguments) cho đến G. Boole (Mathematical Model of Thoughts), cho đến Hilbert: Logics. (nhưng không phải các hành vi thông minh đều có thể biểu diễn bằng Logic!)

### 2.1.4 Hành động có lí trí

- Hành vi được coi là thông minh nếu giúp cho tác nhân (agent) thực hiện hành vi tăng cơ hội thực hiện được đích đặt ra cho nó với điều kiện thông tin phương tiện cho phép của môi trường mà nó đang tồn tại.

- Như vậy ưu điểm là:

* + Thông minh không nhất thiết phải là con người hay giống người
  + Hành vi thông minh không nhất thiết phải thực hiện thông qua suy nghĩ, lý luận

## 2.2 Các lĩnh vực liên quan đến TTNT

* Tâm lý học nhận thức.
* Thần kinh học.
* Lý thuyết về hệ thống (cybernetics).
* Toán Logic và Logic học.
* Sinh học tiến hoá.
* Khoa học về hành vi bầy đàn.
* Tổ chức học.
* Thống kê học.
* .....

## 2.3 Lịch sử hình thành khoa học TTNT

* Ba giai đoạn:
* Symbolism (70-80) (Automated Reasoning and Proofing, Expert Systems, Logic Programming,...).
* Connectionism (80s-90s) (Neural Networks, Statistical Learning, Support Vector Machines, Probabilistic Graph Learning,....).
* Evolutionary Computation (90s-?) (Evolutionary Programming, Evolutionary Strategies, Genetic Algorithms), Intelligent Multi Agent Systems.
* 1930-A.M.Turing đưa ra các kết quả nghiên cứu về máy thông minh, chương trình thông minh đến trắc nghiệm thông minh, đồng thời đưa ra các kết quả cơ sở quan trọng về máy Turing.
* Phát hiện quan trọng của Turing là chương trình có thể lưu trữ trong bộ nhớ để sau đó được thực hiện trên cơ sở các phép toán cơ bản thao tác với các đại lượng là số 0 và 1 của hệ đếm nhị phân.
* Việc lưu giữ chương trình trong máy cho phép thay đổi chức năng của nó một cách nhanh chóng và dễ dàng thông qua việc nạp chương trình mới khác vào bộ nhớ.
* Điều trên làm cho máy có khả năng học và suy nghĩ đáy chính là biểu hiện đầu tiên của các máy tính được trang bị TTNT.
* 1956-Chương trình tìm dẫn xuất trong các hệ hình thức.
* 1959-Máy giải toán vạn năng (MP3).
* 1960-Mc Kathy đưa ra ngôn ngữ trí tuệ nhân tạo (LispList Processing).
* 1961-Minsky đưa ra ngôn ngữ AI
* Tri thức + Điều khiển = chương trình.
* 1962- Tính tích phân bất định
* 1963- Chương trình Heuristic-(gợi mở).
* 1964-Giải phương trình đại số sơ cấp.
* Chương trình ELIDA - phân tích tâm lý.
* 1966- Phân tích và tổng hợp tiếng nói.
* 1968-Robot.
* Học nói.
* 1972-A. Camerauls (ngôn ngữ Prolog-chương trình Logic)
* 1970-1980: Xử lý ngôn ngữ tự nhiên.
* Cuối 80: Hệ chuyên gia xử lý ngôn ngữ tự nhiên.
* 1981-đề án tạo ra các máy tính thế hệ 5 của Nhật.
* 1986,1987 đến nay: Phát triển mạng Neural và ứng dụng.

## 2.4 Các lĩnh vực ứng dụng của TTNT

* Xử lý ngôn ngữ tự nhiên và giao diện người máy.
  + Lập luận và giải quyết vấn đề tự động.
  + Chuẩn đoán, chưa trị với tri thức chuyên gia.
  + Nhìn và nhận dạng.
  + Xử lý âm thanh tiếng nói.
  + Phát hiện tri thức tự động từ dữ liệu.
  + Lập lịch, kế hoạch tự động.
  + Xây dựng các trò chơi thông minh.
  + Mô phỏng thông minh.
  + Giải các bài toán xã hội, thiên nhiên thông qua mô phỏng thông minh.
  + Cuộc sống nhân tạo.
  + ........

## 2.5 Một số ví dụ về TTNT

* Chương trình chơi cờ trên máy Deep Blue đánh bại đại kiện tướng Kasparov (1997).
* Hệ chuyên gia MYCIN (1984, Standford) không thua kém chuyên gia người trong việc chuẩn đoán bệnh.
* Chiến tranh vùng vịnh 1991, Kỹ thuật TTNT được dùng để lập lịch và lên kế hoạch hậu cần.
* Chiến tranh vùng vịnh lần 2 (2003). Chiến tranh mô phỏng trên máy tính.
* Chương trình lập lịch và điều khiển thông minh trên xe tự hành và Robot tự hành của NASA.
* Máy nhận dạng mắt người tại sân bay Heathrow.

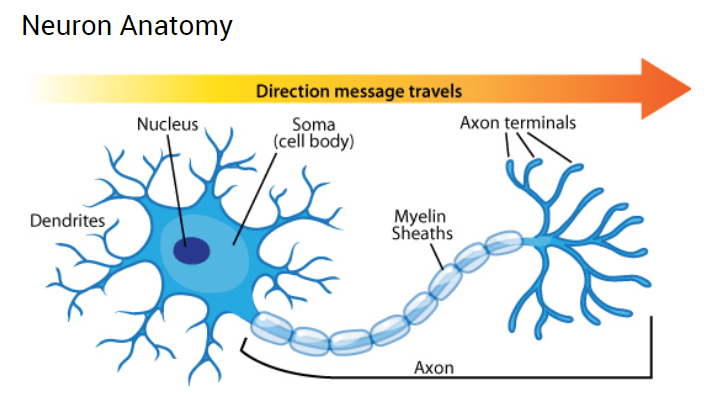
## 2.6 Những vấn đề chưa được giải quyết

* Chương trình chưa tự sinh ra được heuristic
* Chưa có khả năng xử lý song song của con người
* Chưa có khả năng diễn giải một vấn đề theo nhiều phương pháp khác nhau như con người.
* Chưa có khả năng xử lý thông tin trong môi trường liên tục như con người.
* Chưa có khả năng học như con người.
* Chưa có khả năng tự thích nghi với môi trường.

# CHƯƠNG 3: TỔNG QUAN VỀ MẠNG NƠ-RON

## 3.1 Giới thiệu về Neural network

* Neural là tính từ của neuron (nơ-ron), network chỉ cấu trúc đồ thị nên neural network (NN) là một hệ thống tính toán lấy cảm hứng từ sự hoạt động của các nơ-ron trong hệ thần kinh.
* Hoạt động của các nơ-ron:



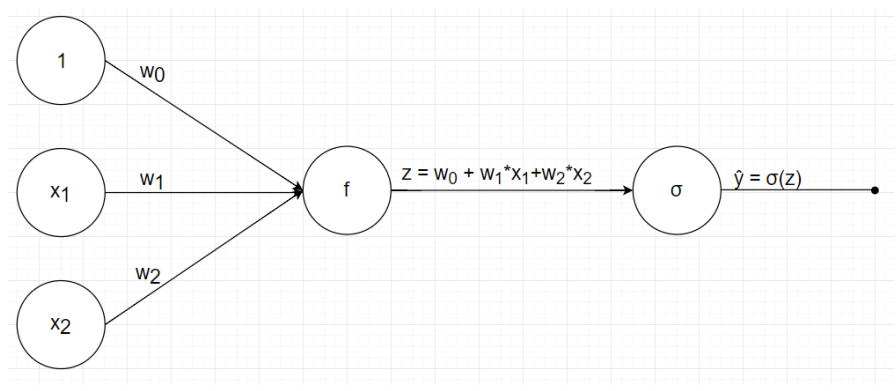
Hình 1: Tế bào nơ-ron

* Nơ-ron là đơn vị cơ bản cấu tạo hệ thống thần kinh và là một phần quan trọng nhất của não. Não chúng ta gồm khoảng 10 triệu nơ-ron và mỗi nơ-ron liên kết với 10.000 nơ-ron khác.
* Ở mỗi nơ-ron có phần thân (soma) chứa nhân, các tín hiệu đầu vào qua sợi nhánh (dendrites) và các tín hiệu đầu ra qua sợi trục (axon) kết nối với các nơ-ron khác. Hiểu đơn giản mỗi nơ-ron nhận dữ liệu đầu vào qua sợi nhánh và truyền dữ liệu đầu ra qua sợi trục, đến các sợi nhánh của các nơ-ron khác.
* Mỗi nơ-ron nhận xung điện từ các nơ-ron khác qua sợi nhánh. Nếu các xung điện này đủ lớn để kích hoạt nơ-ron, thì tín hiệu này đi qua sợi trục đến các sợi nhánh của các nơ-ron khác.
* Mô hình neural network lấy cảm hứng từ não bộ và cách nó hoạt động, chứ không phải bắt chước toàn bộ các chức năng của nó. Việc chính của chúng ta là dùng mô hình đấy đi giải quyết các bài toán chúng ta cần.

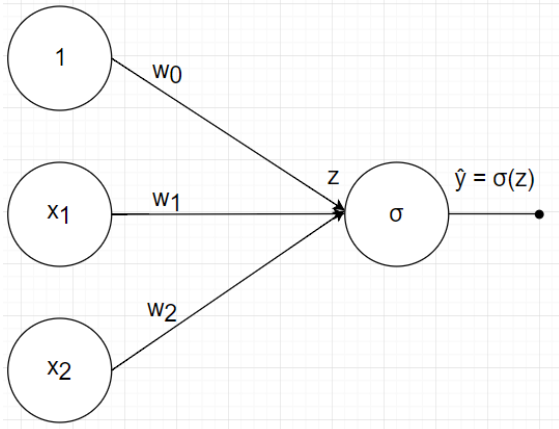
## 3.2 Mô hình Neural network

### 3.2.1 Logistic regression

* Logistic regression là mô hình neural network đơn giản nhất chỉ với input layer và output layer.
* Mô hình của logistic regression là: yˆ = s(w0 +w1 ∗ x1 +w2 ∗ x2). Có 2 bước:
  + Tính tổng linear: z = 1∗ w0 +x1 ∗ w1 +x2 ∗ w2
  + Áp dụng sigmoid function: yˆ = s(z)



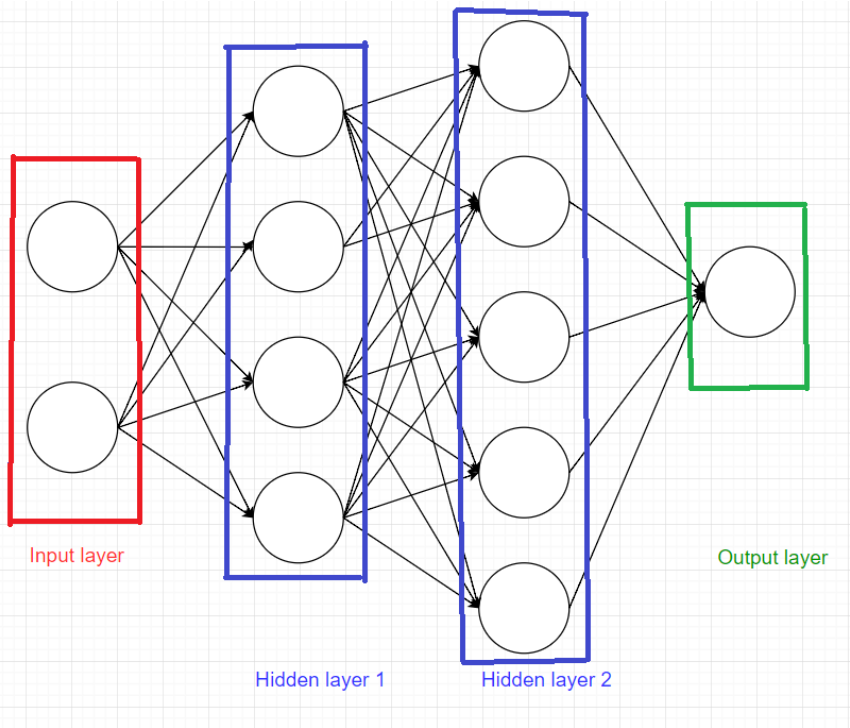
* Để biểu diễn gọn lại ta sẽ gộp hai bước trên thành một trên biểu đồ dưới đây:



Hình 2: Mô hình Logistic regression

* Hệ số w0 được gọi là bias. Để ý từ những bài trước đến giờ dữ liệu khi tính toán luôn được thêm 1 để tính hệ số bias w0 . Tại sao lại cần hệ số bias? Phương trình đường thẳng sẽ thế nào nếu bỏ w0, phương trình giờ có dạng: y = w1 ∗ x, sẽ luôn đi qua gốc tọa độ và nó không tổng quát hóa phương trình đường thẳng nên có thể không tìm được phương trình mong muốn. => Việc thêm bias (hệ số tự do) là rất quan trọng.
* Hàm sigmoid ở đây được gọi là activation function.

### 3.2.2 Mô hình tổng quát

****

Hình 3: Mô hình neural network

- Layer đầu tiên là input layer, các layer ở giữa được gọi là hidden layer, layer cuối cùng được gọi là output layer. Các hình tròn được gọi là node.

- Mỗi mô hình luôn có 1 input layer, 1 output layer, có thể có hoặc không các hidden layer. Tổng số layer trong mô hình được quy ước là số layer – 1 (Không tính input layer).

- Mỗi node trong hidden layer và output layer:

* + Liên kết với tất cả các node ở layer trước đó với các hệ số w riêng.
  + Mỗi node có 1 hệ số bias b riêng.
* Diễn ra 2 bước: tính tổng linear và áp dụng activation function. Trong đó tổng linear là tổng các giá trị đầu vào, activation function là hàm kích hoạt của mạng.

### **3.2.3 Kí hiệu**

* Số node trong hidden layer thứ i là l(i).
* Ma trận W(k) kích thước l(k−1) ∗ l(k) là ma trận hệ số giữa layer (k-1) và layer k, trong đó w(k)ij là hệ số kết nối từ node thứ i của layer k-1 đến node thứ j của layer k.
* Vector b(k) kích thước lk ∗ 1 là hệ số bias của các node trong layer k, trong đó b(k)i là bias của node thứ i trong layer k.
* Với node thứ i trong layer l có bias b(l)i thực hiện 2 bước:
* Tính tổng linear: z(l)i = ∑lj(=l−11) a(jl−1) ∗w(ji l) +b(il), là tổng tất cả các node trong layer trước nhânvới hệ số w tương ứng, rồi cộng với bias b.
* Áp dụng activation function: a(l)i = s(z(l)i)
* Vector z(k) kích thước l(k) ∗ 1 là giá trị các node trong layer k sau bước tính tổng linear.
* Vector a(k) kích thước l(k) ∗ 1 là giá trị của các node trong layer k sau khi áp dụng hàm activation function

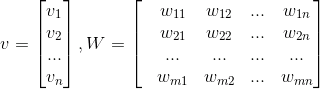
## 3.3 Ảnh trong máy tính

Hệ màu RGB:

* RGB viết tắt của red (đỏ), green (xanh lục), blue (xanh lam), là ba màu chính của ánh sáng khi tách ra từ lăng kính. Khi trộn ba màu trên theo tỉ lệ nhất định có thể tạo thành các màu khác nhau.
* Với mỗi bộ 3 số r, g, b nguyên trong khoảng [0, 255] sẽ cho ra một màu khác nhau. Do có 256 cách chọn r, 256 cách chọn màu g, 256 cách chọn b => tổng số màu có thể tạo ra bằng hệ màu RGB là: 256 \* 256 \* 256 = 16777216 màu.
* Ảnh màu là một ma trận các pixel mà mỗi pixel biểu diễn một điểm màu. Mỗi điểm màu được biểu diễn bằng bộ 3 số (r,g,b). Để tiện cho việc xử lý ảnh thì sẽ tách ma trận pixel ra 3 channel red, green, blue.

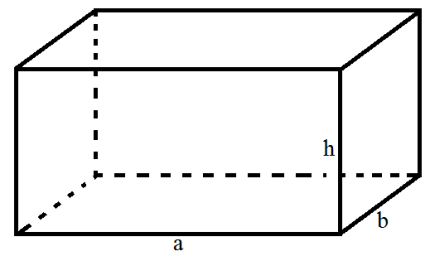
Tensor:

* Khi dữ liệu biểu diễn dạng 1 chiều, người ta gọi là vector, mặc định khi viết vector sẽ viết dưới dạng cột.
* Khi dữ liệu dạng 2 chiều, người ta gọi là ma trận, kích thước là số hàng \* số cột.



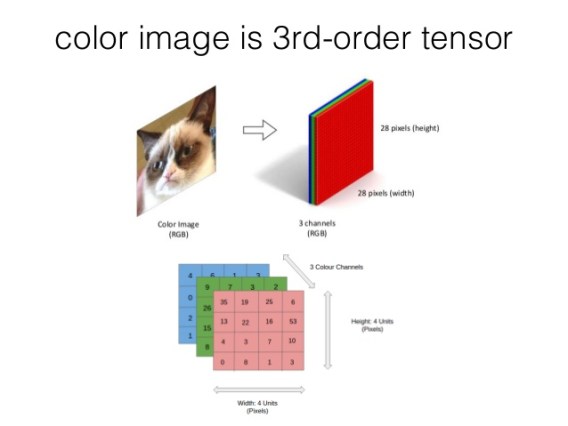
Hình 4: Vector v kích thước n, ma trận W kích thước m\*n

* Khi dữ liệu nhiều hơn 2 chiều thì sẽ được gọi là tensor



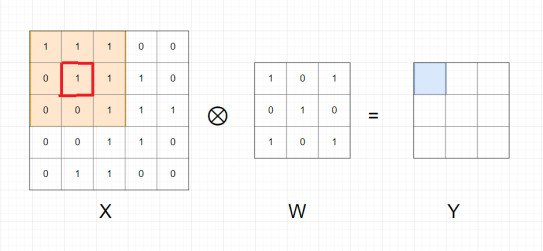
Hình 5: Hình hộp chữ nhật kích thước a\*b\*h

* Tưởng tượng mặt đáy là một ma trận kích thước a \* b, được tạo bởi b vector kích thước a. Cả hình hộp là tensor 3 chiều kích thước a\*b\*h, được tạo bởi xếp h ma trận kích thước a\*b lên nhau.



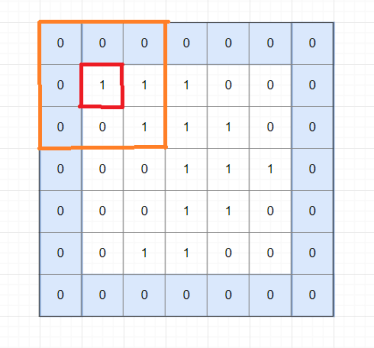
Hình 6: Ảnh màu kích thước 28\*28, biểu diễn dưới dạng tensor 28\*28\*3

* Phép tính tích chập (convolution)



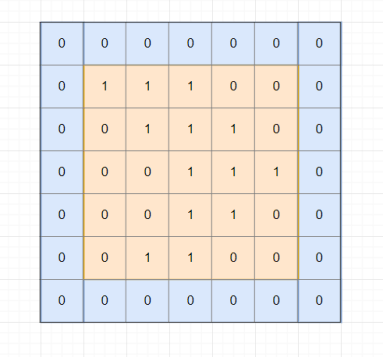
Hình 7: Phép tính tích chập

* Padding
* Mỗi lần thực hiện phép tính convolution xong thì kích thước ma trận Y đều nhỏ hơn X. Nếu muốn ma trận Y thu được có kích thước bằng ma trận X chỉ cần thêm giá trị 0 ở viền ngoài ma trận X.



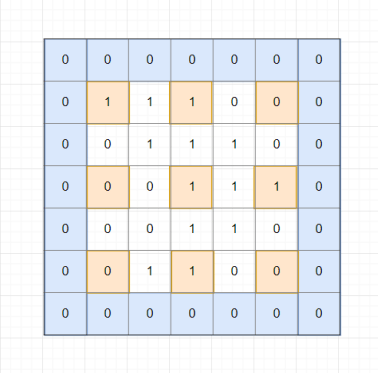
Hình 8: Ma trận X khi thêm viền 0 bên ngoài

* Phép tính này gọi là convolution với padding = 1. Padding = k nghĩa là thêm k vector 0 vào mỗi phía của ma trận.
* Stride



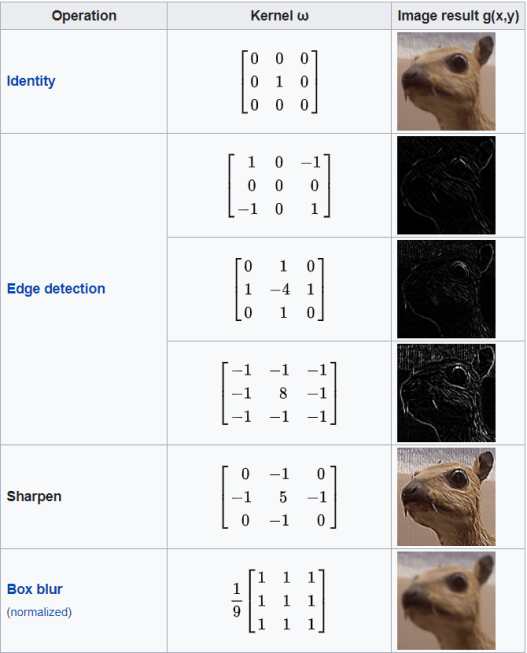
Hình 9: stride=1, padding=1

* Ta thực hiện tuần tự các phần tử trong ma trận X, thu được ma trận Y cùng kích thước ma trận X, ta gọi là stride=1.
* Nếu stride=k (k > 1) thì ta chỉ thực hiện phép tính convolution trên các phần tử từ vị trí đầu tiên sau đó nhảy k bước theo chiều dọc và ngang cho đến hết ma trận X.



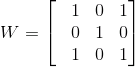
Hình 10: padding=1, stride=2

* Ý nghĩa của phép tính convolution
* Mục đích của phép tính convolution trên ảnh là làm mở, làm nét ảnh; xác định các đường;… Mỗi kernel khác nhau thì sẽ phép tính convolution sẽ có ý nghĩa khác nhau.



Hình 11: Ý nghĩa của phép tính tích chập

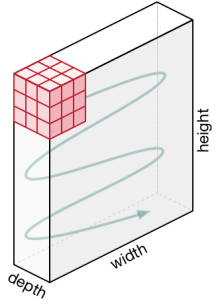
* Kernel là một ma trận vuông kích thước k\*k sử dụng trong phép tính tích chập



Hình 12: kernel kích thước 3\*3

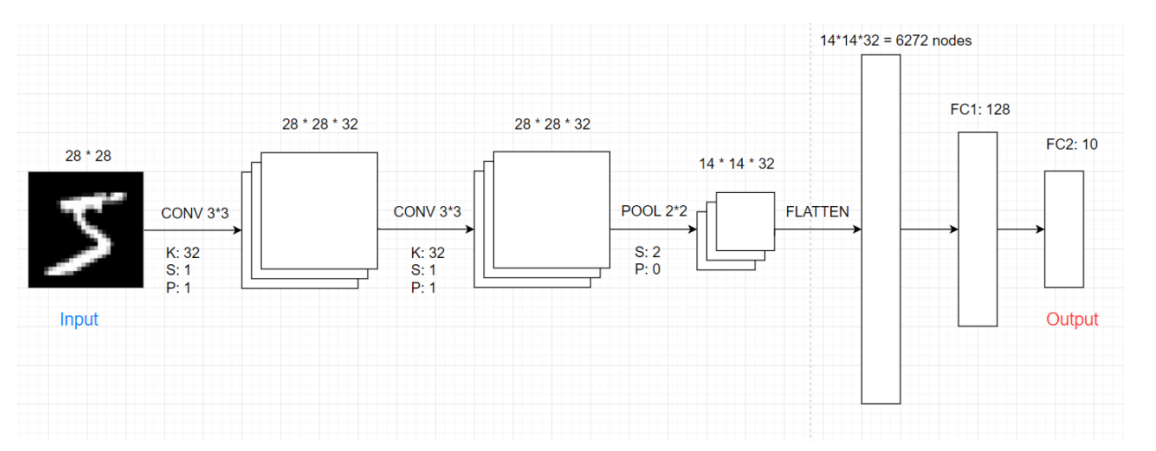
## 3.4 Convolutional neural network (CNN)

* Mạng CNN còn gọi là mạng nơ-ron tích chập, là một lớp của mạng thần kinh sâu (deep neural network), áp dụng phổ biến nhất để phân tích hình ảnh trực quan.
* CNN có nhiều ứng dụng trong thị giác máy tính, hệ thống gợi ý, phân loại hình ảnh, tính toán hình ảnh y tế (điện toán hình ảnh y tế), xử lý ngôn ngữ tự nhiên, và chuỗi thời gian tài chính.
* Trong bài toán xử lý ảnh, CNN áp dụng các phép tính tích chập để lấy ra các đặc trưng của ảnh.
* Xử lý ảnh với 3 kênh màu (RGB), mạng sẽ sử dụng kernal là 1 tensor 3 chiều kích thước k\*k\*3



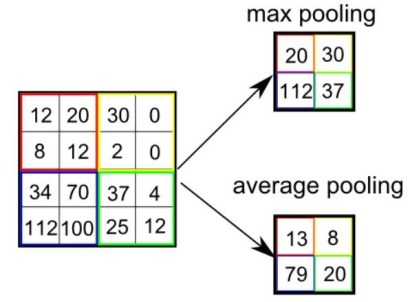
Hình 13: Phép tính convolution trên ảnh màu với k=3

* CNN có quy tắc với padding và stride tương tự như xử lý ảnh ở máy tính.



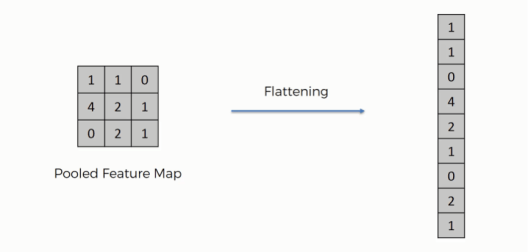
Hình 14: Mô hình CNN

* Tổng quát:
* Input image -> Convolutional layer (Conv) + Pooling layer (Pool) -> Fully connected layer (FC) -> Output.
* Conv là các phép tính tích chập
* Pooling layer thường được dùng giữa các convolutional layer, để giảm kích thước dữ liệu nhưng vẫn giữ được các thuộc tính quan trọng. Kích thước dữ liệu giảm giúp giảm việc tính toán trong model.
* Có 2 loại pooling layer phổ biến là: max pooling và average pooling.



Hình 15: Ví dụ về pooling layer

* Fully connected layer để kết hợp các đặc điểm của ảnh để ra được output của model.
* Sau khi ảnh được truyền qua nhiều convolutional layer và pooling layer thì model đã học được tương đối các đặc điểm của ảnh (ví dụ mắt, mũi, khung mặt, …) thì tensor của output của layer cuối cùng, kích thước H\*W\*D, sẽ được chuyển về 1 vector kích thước (H\*W\*D)

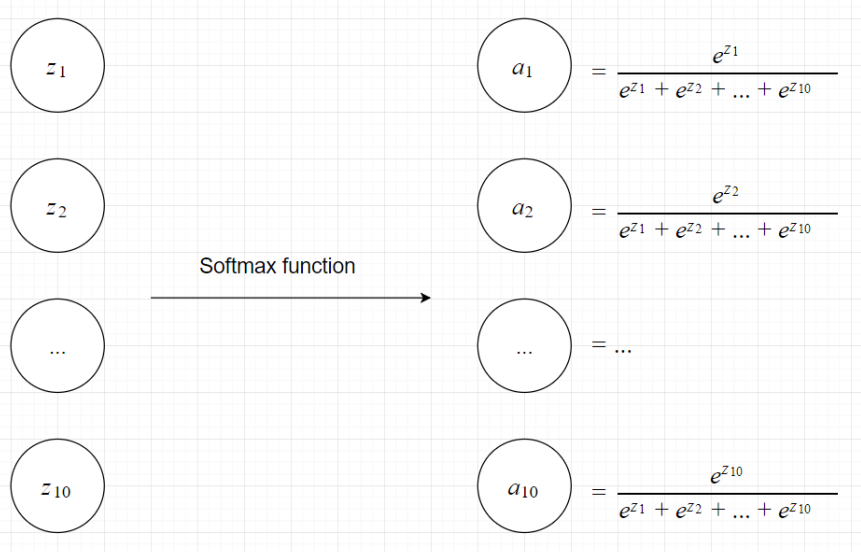


Hình 16: Fully connected layer

# CHƯƠNG 4: XÂY DỰNG CHƯƠNG TRÌNH

## 4.1 Chuẩn bị dữ liệu

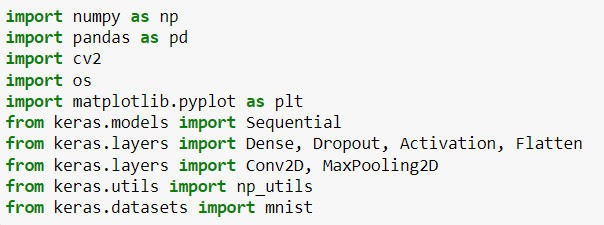
* Dữ liệu đầu vào là số lượng ảnh các bằng tốt nghiệp, cụ thể là số hiệu cấp trên bằng tốt nghiệp.
* Yêu cầu đầu vào của ảnh chụp bằng tốt nghiệp:
  + Ảnh cần được chụp đầy đủ khung hình.
  + Ảnh cần có chất lượng, độ phân giải khá trở lên (Nếu ảnh có chất lượng thấp hoặc độ phân giải kém sẽ ảnh hưởng tới quá trình nhận diện của chương trình).
* MNIST là bộ cơ sở dữ liệu về chữ số viết tay, bao gồm 2 tập con: training set gồm 60.000 ảnh các chữ số viết tay và test set gồm 10.000 ảnh các chữ số**.**
* Mục đích cuối cùng của machine learning không phải là predict tốt với dữ liệu mình đang có mà là để predict tốt các dữ liệu mới khi đi vào thực tế, vậy nên ta mong muốn model học được tổng quan hóa dữ liệu (generalization) thay vì chỉ nhớ các dữ liệu trong dataset. Để đánh giá xem model có học không hay chỉ nhớ cũng như khi dùng ngoài thực tế thì performance sẽ thế nào, người ta chia dataset làm 3 tập traning set, validation set và test set.
* Softmax function:
* Giống như bài logistic regression, thay vì chỉ muốn kết quả là ảnh là số mấy, ta muốn dự đoán phần trăm của ảnh là số nào
* Nhắc lại bài neural network, ở mỗi layer sẽ thực hiện 2 bước: tính tổng linear các node ở layer trước và thực hiện activation function (ví dụ sigmoid function, softmax function). Do sau bước tính tổng linear cho ra các giá trị thực nên cần dùng softmax function dùng để chuyển đổi giá trị thực trong các node ở output layer sang giá trị phần trăm.
* Vì mỗi ảnh sẽ thuộc 1 class từ 0 đến 9, nên tất cả sẽ có 10 class. Nên output layer sẽ có 10 node để tương ứng với phần trăm ảnh là số 0,1,..,9.



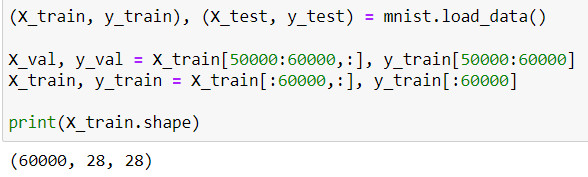
Hình 17: Softmax function

## 4.2 Thực hiện chương trình

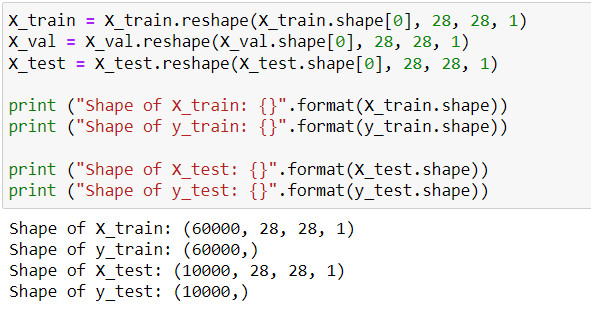
* Khai báo các thư viện cần sử dụng:

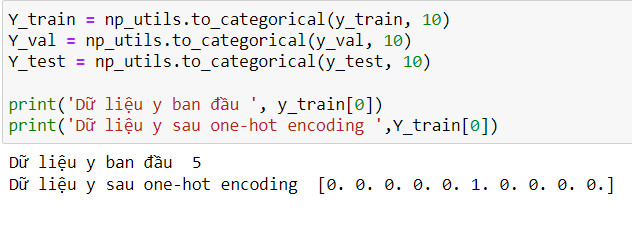
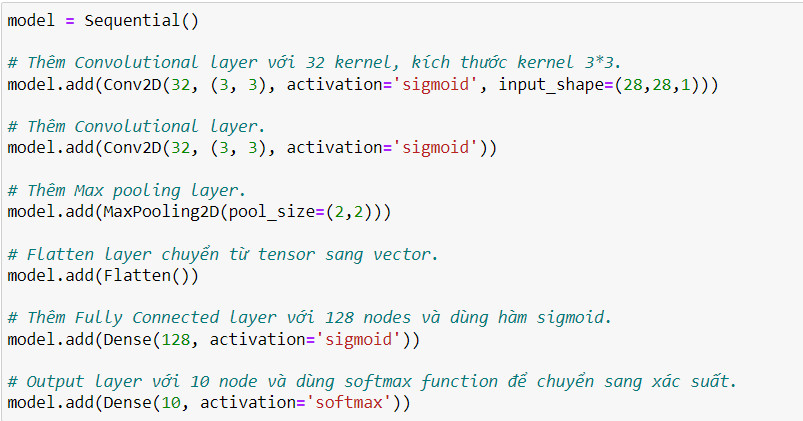
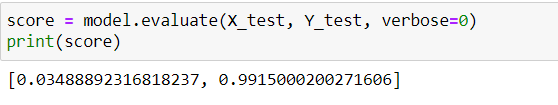
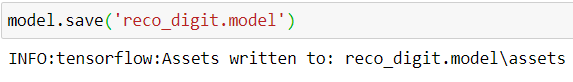
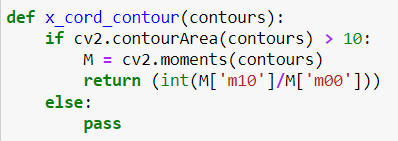
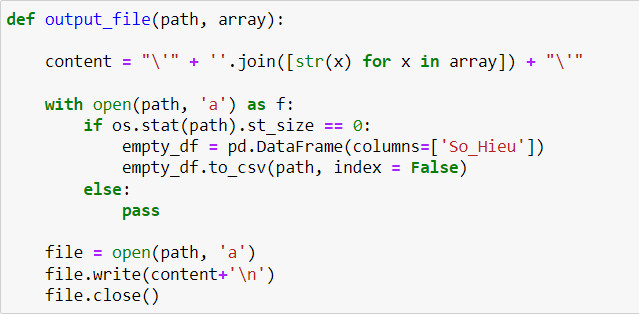
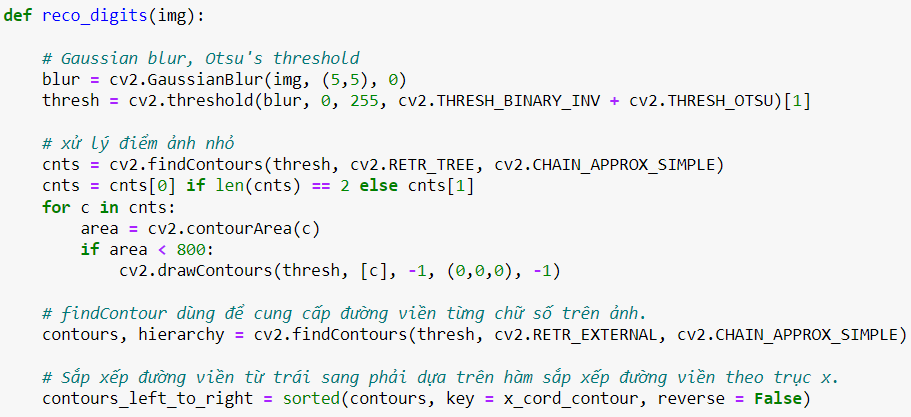
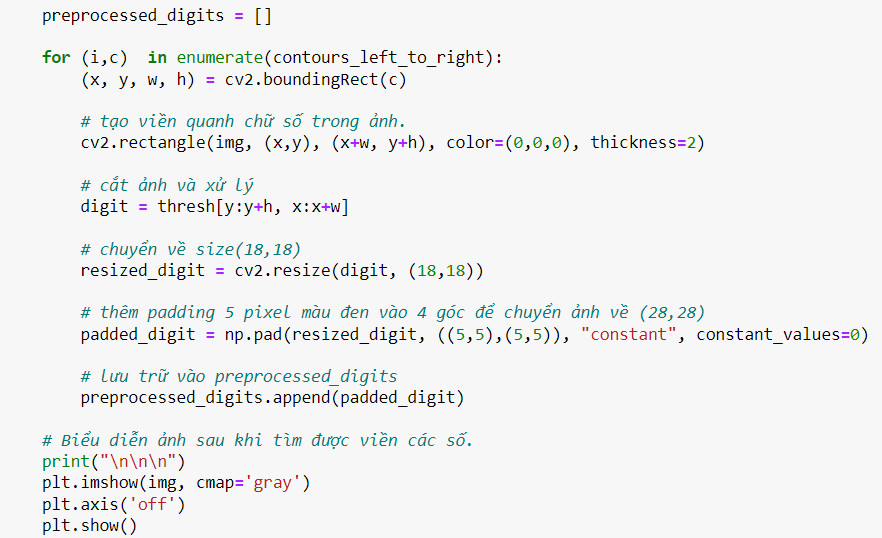
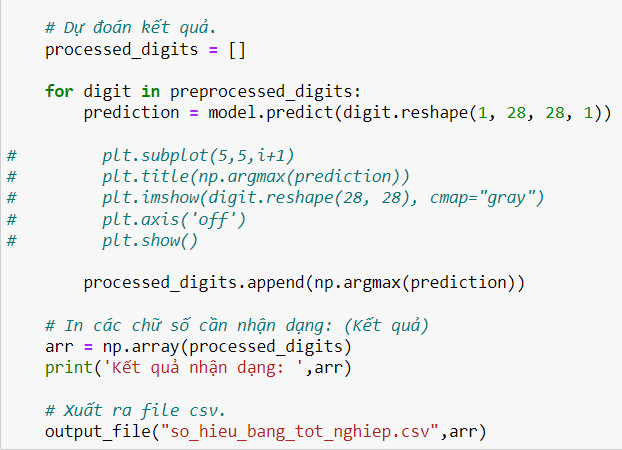


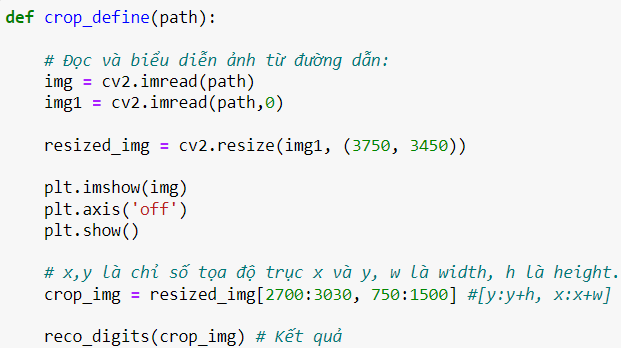
* Load dữ liệu từ tệp MNIST

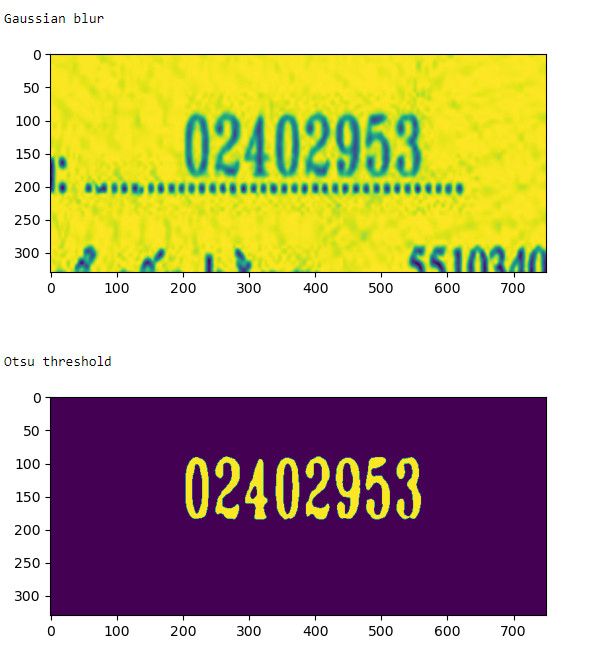


* Định dạng lại dữ liệu cho đúng kích thước mà thư viện Keras yêu cầu

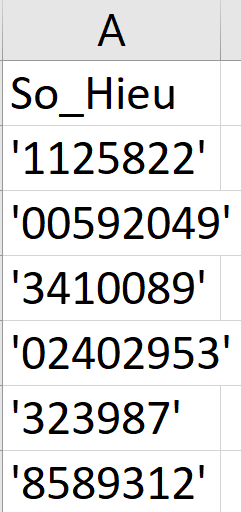


* Áp dụng thuật toán One hot encoding  
    
  
* Định nghĩa mô hình  
    
  
* Biên dịch chương trình  
    
  
* Thực hiện huấn luyện mô hình với tệp dữ liệu MNIST  
    
  
* Đánh giá mô hình với dữ liệu test  
    
  
* Lưu mô hình  
    
  
* Dự đoán kết quả với tệp test của MNIST  
    
  
* Sắp xếp viền theo trục x  
    
  
* Hàm xuất dữ liệu ra file CSV (Excel)   
    
  
* Hàm nhận dạng chữ số và xuất file  
    
    
    
    
    
  

* Hàm cắt ảnh và hiển thị kết quả từ ảnh được chọn  
    
  

* Chạy thử nghiệm chương trình với ảnh bằng tốt nghiệp  
    
  
* Minh họa quá trình xử lý ảnh của chương trình  
    
    
    
  

* Kết quả được xuất ra file CSV (Excel) có kiểu như sau



## 4.3 Một số hạn chế khi chạy chương trình

* Khi chất lượng ảnh đưa vào quá thấp thì chương trình sẽ không thể sử dụng các hàm hay dữ liệu đã được huấn luyện để nhận dạng các chữ số in trên bằng tốt nghiệp. Cụ thể được minh họa bằng hình ảnh sau đây:



Hình 18: Ảnh minh họa lỗi nhận dạng chữ số

* Khi ảnh đưa vào có số hiệu là [0 0 3 9 9 3 3 3] thì kết quả nhận dạng của chương trình sai hoàn toàn là [4 9 9 3 2 4]. Điều này chính là hạn chế mà nhóm chúng em đã gặp phải và khắc phục bằng cách tìm và đưa vào chương trình ảnh có độ phân giải hoặc chất lượng tốt nhất để sai số của chương trình đạt ở mức nhỏ nhất.

# CHƯƠNG 5: TỔNG KẾT

## 5.1 Tài liệu tham khảo

* Nguyễn Thanh Tuấn, 08/2020, Sách Deep Learning cơ bản V2, [tr.109 – 147]:

(1) <https://nttuan8.com/sach-deep-learning-co-ban/>

(2) <https://nttuan8.com/bai-7-gioi-thieu-keras-va-bai-toan-phan-loai-anh/>

* OpenCV:

<https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html>

* Phạm Đình Khánh, 01/2020, Bài 21: Tiền xử lý ảnh OpenCV:

<https://phamdinhkhanh.github.io/2020/01/06/ImagePreprocessing.html>

## 5.2 Phụ lục

[Hình 1: Tế bào nơ-ron 11](#_Toc121870677)

[Hình 2: Mô hình Logistic regression 13](#_Toc121870678)

[Hình 3: Mô hình neural network 14](#_Toc121870679)

[Hình 4: Vector v kích thước n, ma trận W kích thước m\*n 16](#_Toc121870680)

[Hình 5: Hình hộp chữ nhật kích thước a\*b\*h 16](#_Toc121870681)

[Hình 6: Ảnh màu kích thước 28\*28, biểu diễn dưới dạng tensor 28\*28\*3 17](#_Toc121870682)

[Hình 7: Phép tính tích chập 18](#_Toc121870683)

[Hình 8: Ma trận X khi thêm viền 0 bên ngoài 19](#_Toc121870684)

[Hình 9: stride=1, padding=1 20](#_Toc121870685)

[Hình 10: padding=1, stride=2 21](#_Toc121870686)

[Hình 11: Ý nghĩa của phép tính tích chập 22](#_Toc121870687)

[Hình 12: kernel kích thước 3\*3 23](#_Toc121870688)

[Hình 13: Phép tính convolution trên ảnh màu với k=3 24](#_Toc121870689)

[Hình 14: Mô hình CNN 25](#_Toc121870690)

[Hình 15: Ví dụ về pooling layer 26](#_Toc121870691)

[Hình 16: Fully connected layer 27](#_Toc121870692)

[Hình 17: Softmax function 29](#_Toc121870693)

[Hình 18: Ảnh minh họa lỗi nhận dạng chữ số 42](#_Toc121870694)