

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

=====***=====



BÁO CÁO THỰC NGHIỆM

HỌC PHẦN: HỆ HỖ TRỢ QUYẾT ĐỊNH

**Đề tài: Tìm hiểu về Deep Learning trong việc nhận diện chữ số
viết tay**

GVHD: ThS. Trần Thanh Hùng

Nhóm: 13

Thành viên: Lưu Minh Chiến - 2021601446
Nguyễn Hoàng Hải - 2021602114
Nguyễn Huy Hoàng - 2021601725

Lớp: 20234IT6058001 - Khoá: 16

Hà Nội, 2024

MỤC LỤC

LỜI NÓI ĐẦU	1
CHƯƠNG 1: LÝ THUYẾT HỆ HỖ TRỢ RA QUYẾT ĐỊNH.....	2
1.1 Hệ hỗ trợ ra quyết định là gì?	2
1.1.1 DSS (Decision Support System) là gì?	2
1.1.2 Khái niệm DSS (Decision Support System).....	2
1.1.3 Đặc điểm của hệ thống hỗ trợ ra quyết định DSS là gì?.....	2
1.1.4 Lợi ích của DSS mang lại cho doanh nghiệp.....	3
1.2 Kiến trúc chung hệ hỗ trợ ra quyết định	5
1.3 Mô hình ra quyết định	6
1.4 Mục đích xây dựng hệ hỗ trợ ra quyết định	7
1.5 Tính cần thiết và phạm vi ứng dụng của hệ hỗ trợ ra quyết định.....	8
1.5.1 Tính cần thiết của hệ hỗ trợ ra quyết định	8
1.5.2 Phạm vi ứng dụng của hệ hỗ trợ ra quyết định	8
1.6 Môi trường ra quyết định và các yếu tố ảnh hưởng	9
1.6.1 Môi trường ra quyết định.....	9
1.6.2 Các yếu tố ảnh hưởng đến ra quyết định	10
CHƯƠNG 2 : TÌM HIỂU VỀ DEEP LEARNING VÀ MÔ HÌNH CONVOLUTIONAL NEURAL NETWORK	12
2.1 Deep Learning là gì?.....	12
2.2 Deep Learning hoạt động như thế nào?	13
2.3 Ưu và nhược điểm của Deep Learning	14
2.4 Ứng dụng của Deep Learning vào đời sống	15

2.4.1 Digital Marketing	15
2.4.2 Chế tạo Robot.....	16
2.4.3 Hệ thống lái xe tự động	17
2.4.4 Trợ lý ảo	18
2.4.5 Nghiên cứu y học	19
2.4.6 Nhận diện khuôn mặt	20
2.4.7 Mua sắm và giải trí.....	21
2.5. Tổng quan về mạng CNN	22
2.5.1 Khái niệm chung	22
2.5.2 Các kiểu tầng trong mạng CNN	23
2.5.3 Ví dụ minh họa mô hình CNN	28
CHƯƠNG 3: XÂY DỰNG HỆ QUYẾT ĐỊNH SỬ DỤNG MÔ HÌNH CNN TRONG BÀI TOÁN NHẬN DIỆN CHỮ SỐ VIẾT TAY.....	31
3.1 Thuật toán áp dụng	31
3.2 Dữ liệu đầu vào.....	32
3.3 Cài đặt.....	33
3.3.1 Khai báo thư viện	33
3.3.2 Lấy cơ sở dữ liệu MNIST	33
3.3.3 Tập kiểm thử - Validation	34
3.3.4 Chuẩn hóa dữ liệu	34
3.3.5 Xây dựng mô hình CNN	35
3.3.6 Huấn luyện mô hình	35

3.3.7 Đánh giá và lưu mô hình huấn luyện	36
3.3.8 Biểu đồ đánh giá.....	36
3.3.9 Thử nghiệm mô hình hình ảnh	37
TỔNG KẾT	40
TÀI LIỆU THAM KHẢO.....	41

LỜI NÓI ĐẦU

Trong thời đại số hóa ngày nay, công nghệ thông tin và trí tuệ nhân tạo đang ngày càng chiếm giữ một vị trí quan trọng trong cuộc sống và các ngành công nghiệp. Đặc biệt, hệ thống hỗ trợ quyết định, với khả năng xử lý lượng thông tin lớn và tự động hóa quy trình ra quyết định, đang trở thành một công cụ không thể thiếu trong nhiều lĩnh vực.

Mạng nơ-ron tích chập (Convolutional Neural Network - CNN) đã trở thành một trong những công cụ quan trọng trong lĩnh vực trí tuệ nhân tạo và xử lý hình ảnh. Với khả năng nắm bắt thông tin không gian và cấu trúc của dữ liệu hình ảnh, CNN đã mang lại những đột phá đáng kể trong việc nhận diện đối tượng, phân loại hình ảnh và thậm chí tạo ra nội dung sáng tạo. Với những ưu điểm vượt trội này, đề tài chúng ta hôm nay đã chọn lựa mô hình CNN để xây dựng hệ thống nhận diện chữ số viết tay.

Trong trình bày tiếp theo, chúng ta sẽ tìm hiểu chi tiết về cấu trúc và hoạt động của mạng CNN, cách nó áp dụng các khối tích chập và tổng hợp để trích xuất đặc trưng, và sử dụng trong các ứng dụng thực tế. Hi vọng rằng qua việc tìm hiểu về CNN, chúng ta sẽ có cái nhìn rõ ràng hơn về sự phát triển và tiềm năng của trí tuệ nhân tạo trong xử lý hình ảnh và các lĩnh vực liên quan.

Chúng em xin gửi lời cảm ơn chân thành nhất đến thầy Trần Thanh Hùng. Trong suốt thời gian quá trình thực hiện, chúng em nhận được sự góp ý, nhận xét từ thầy. Bài báo cáo của nhóm có thể còn nhiều sai sót, rất mong sự đóng góp của quý thầy cô để báo cáo được hoàn thiện hơn.

Một lần nữa, xin chân thành cảm ơn thầy

CHƯƠNG 1: LÝ THUYẾT HỆ HỖ TRỢ RA QUYẾT ĐỊNH

1.1 Hệ hỗ trợ ra quyết định là gì?

1.1.1 DSS (Decision Support System) là gì?

DSS là viết tắt của cụm từ Decision Support System nghĩa là hệ thống hỗ trợ ra quyết định hoặc hệ hỗ trợ quyết định. Đúng như tên gọi của mình, DSS được tạo ra với mục đích hỗ trợ trong việc đưa ra quyết định cũng như đưa ra những dự đoán, chiều hướng hành động của một tổ chức hoặc một doanh nghiệp.

1.1.2 Khái niệm DSS (Decision Support System)

DSS hay hệ hỗ trợ ra quyết định là một hệ thống phần mềm tương tác và thu thập mọi thông tin liên quan từ rất nhiều nguồn như vận hành, thu nhập, chi phí, thị trường, xu hướng, mô hình doanh nghiệp. Sau đó DSS sẽ sàng lọc và phân tích những dữ liệu này, tổng hợp lại thành các thông tin một cách toàn diện từ đó có thể sử dụng để hỗ trợ trong việc đưa ra quyết định.

DSS có thể được điều khiển thủ công bởi con người hoặc máy tính hóa hoàn toàn. Trong một vài trường hợp, DSS có thể kết hợp cả hai phương pháp với nhau. Hệ thống DSS lý tưởng sẽ phân tích dữ liệu và tổng hợp thông tin để tăng độ chính xác cũng như tốc độ trong việc đưa ra quyết định.

1.1.3 Đặc điểm của hệ thống hỗ trợ ra quyết định DSS là gì?

Hệ hỗ trợ ra quyết định - DSS được lập trình để tạo ra nhiều loại báo cáo khác nhau, tất cả đều tùy chỉnh được bởi thông số kỹ thuật của người dùng. Điều này là rất thuận tiện và hữu ích vì nó giúp việc trình bày những thông tin tới người xem trở nên dễ hiểu và phù hợp hơn. Thay vì một báo cáo đi qua nhiều mức quản lý như trước, với DSS có thể tạo ra các báo cáo khác nhau cho

từng mức quản lý từ đó họ có thể đưa ra quyết định thích hợp tại mức riêng của họ.



Hình 1.1: Đặc điểm của hệ thống hỗ trợ ra quyết định DSS là gì?

1.1.4 Lợi ích của DSS mang lại cho doanh nghiệp

Sau khi hiểu DSS là gì? Có thể thấy lợi ích lớn nhất và hệ hỗ trợ ra quyết định - DSS mang lại là cung cấp những thông tin bổ ích cần thiết cho các nhà quản trị để từ đó họ đưa ra những quyết định chính xác hơn. Ngoài ra, DSS chỉ là một phần mềm vi tính nên người dùng có thể sử dụng vô cùng thuận tiện và linh hoạt. Cung cấp thông tin cho người dùng mọi lúc mọi nơi cũng như phù hợp với từng mức quản lý khác nhau với khả năng tạo ra nhiều loại báo cáo. Tất cả để phục vụ mục đích chất của DSS là cải thiện quy trình đưa ra quyết định giúp kiểm soát doanh nghiệp một cách hiệu quả hơn.

1.1.1.1. Vai trò hệ hỗ trợ ra quyết định

Lý do sử dụng DSS:

- Cải thiện tốc độ tính toán;
- Tăng năng suất của cá nhân liên đới;

- Cải tiến kỹ thuật trong việc lưu trữ, tìm kiếm, trao đổi dữ liệu trong và ngoài tổ chức theo hướng nhanh và kinh tế;
- Nâng cao chất lượng của các quyết định đưa ra;
- Tăng cường năng lực cạnh tranh của tổ chức;
- Khắc phục khả năng hạn chế của con người trong việc xử lý và lưu chứa thông tin.

Các hỗ trợ từ DSS:

- Cung cấp thông tin trạng thái và dl thô
- Khả năng phân tích tổng quát
- Mô hình biểu diễn (cân đối tài chính), mô hình nhân quả (dự báo, chuẩn đoán)
- Đề nghị giải pháp, đánh giá
- Chọn lựa giải pháp

1.1.1.2. Quá trình ra quyết định của hệ hỗ trợ ra quyết định

a) Phân loại quyết định

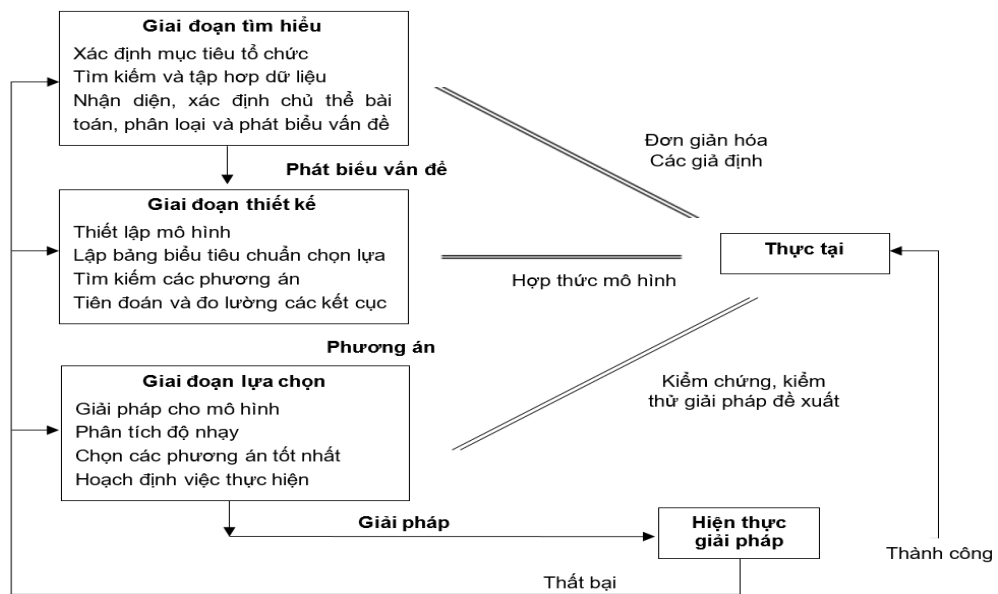
Có thể phân ra 4 loại quyết định sau:

- Quyết định có cấu trúc (Structured Decision): các quyết định mà người ra quyết định biết chắc chắn đúng.
- Quyết định không có cấu trúc (Nonstructured Decision): các quyết định mà người ra quyết định biết là có nhiều câu trả lời gần đúng và không có cách nào tìm ra câu trả lời chính xác nhất.
- Quyết định đệ quy (Recurring Decision): các quyết định lặp đi, lặp lại
- Quyết định không đệ quy (Nonrecurring Decision): các quyết định không xảy ra thường xuyên.

b) Các giai đoạn của quá trình ra quyết định

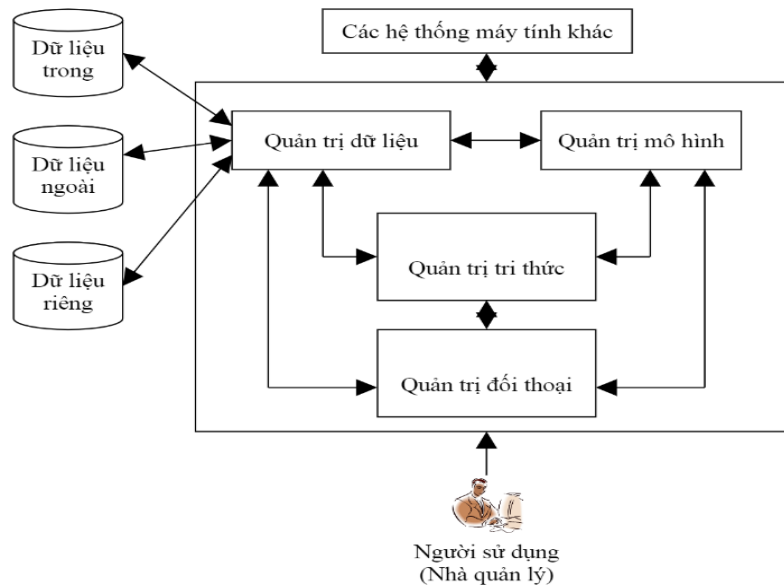
Theo Simon, các giai đoạn của quá trình ra quyết định gồm các pha:

- Tìm hiểu - bài toán dẫn đến quyết định;
- Thiết kế - phân tích và xây dựng các diễn trình hành động;
- Chọn lựa - chọn một diễn trình trong tập diễn trình;
- Thực hiện - thực hiện các quyết định để có được kết quả.



Hình 1. 2: Sơ đồ các giai đoạn ra quyết định

1.2 Kiến trúc chung hệ hỗ trợ ra quyết định



Hình 1. 3: Kiến trúc chung hệ hỗ trợ quyết định

Trong đó:

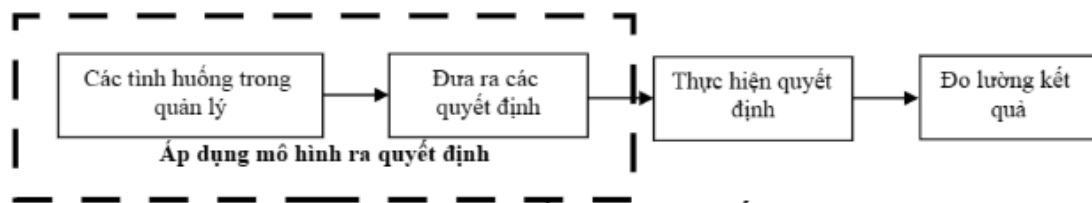
- Quản trị dữ liệu: bao gồm các CSDL chứa dữ liệu liên quan đến một tình huống và được quản lý bởi phần mềm là hệ quản trị CSDL (quản lý và khai thác).
- Quản trị mô hình: cho phép khai thác và quản lý các mô hình định lượng (xử lý) khác nhau, cung cấp khả năng phân tích cho hệ thống.
- Quản trị đối thoại: cung cấp giao diện cho người dùng để liên lạc và ra lệnh cho Hệ hỗ trợ quyết định.
- Quản trị tri thức: hoạt động như 1 thành phần độc lập, hoặc có thể trợ giúp cho bất kỳ 1 hệ thống nào trong 3 hệ thống nói trên.

1.3 Mô hình ra quyết định

Mô hình ra quyết định là một lĩnh vực khoa học quản trị nhằm tìm ra phương pháp tối ưu hoặc hiệu quả nhất của việc sử dụng các nguồn lực có hạn để có thể đạt được các mục tiêu của một cá nhân hoặc một doanh nghiệp đưa ra.

Vì lý do này, mô hình ra quyết định thường được hiểu với một nghĩa khác là *Tối ưu hóa*.

Mô hình ra quyết định thường chỉ áp dụng hai giai đoạn đầu của tiến trình ra quyết định đó là các tình huống quản lý và đưa ra các quyết định còn lại các bước thực hiện quyết định và đo lường kết quả đạt được khi ra quyết định thì không được đề cập đến.



Hình 1. 4 : Mô hình ra quyết định

Triết lý về ra quyết định mỗi cách tiếp cận đối với các vấn đề khác nhau, phụ thuộc rất nhiều vào môi trường xung quanh, nền tảng kiến thức, kinh nghiệm, trạng thái tâm lý.

Có nhiều mô hình ra quyết định, nổi bật nhất là:

Mô hình tỷ lệ: xem ra quyết định là một quá trình có cấu trúc, rút gọn bài toán thành lập các tham số đo được.

Mô hình tổ chức: quan tâm nhiều đến các chính sách, định hướng tiếp theo, quan tâm nhiều tới việc định tính.

Mô hình chính trị: kết quả của liên kết nhóm, thể hiện các khả năng cá nhân.

1.4 Mục đích xây dựng hệ hỗ trợ ra quyết định

Mục đích chính của việc sử dụng DSS là trình bày thông tin cho khách hàng một cách dễ hiểu. Một DSS rất hữu ích vì nó có thể được lập trình để tạo nhiều loại báo cáo, tất cả dựa trên thông số kỹ thuật của người dùng.

DSS trợ giúp các hoạt động ra quyết định. Hệ thống này có rất nhiều ưu điểm nổi trội, cung cấp cho người dùng những thông tin hữu ích một cách linh hoạt.

Cung cấp các công cụ trợ giúp việc phát triển và cải thiện các mô hình nhận thức (về nhân và quả) của người ra quyết định bằng cách cung cấp dữ liệu nhanh, đúng & áp dụng các mô hình toán học.

1.5 Tính cần thiết và phạm vi ứng dụng của hệ hỗ trợ ra quyết định

1.5.1 Tính cần thiết của hệ hỗ trợ ra quyết định

- Nghiên cứu và hoạch định tiếp thị: Chính sách giá cho khách hàng, dự báo sản phẩm tiêu thụ...
- Hoạch định chiến lược và vận hành: Theo dõi, phân tích và báo cáo về xu hướng thị trường...
- Hỗ trợ bán hàng: Chi tiết và tổng hợp tình hình bán hàng, so sánh và phân tích xu hướng bán hàng...

1.5.2 Phạm vi ứng dụng của hệ hỗ trợ ra quyết định

Ở đâu:

- Ở các quyết định hàng ngày.
- Các quyết định có tính chiến thuật: chọn 1 phương sách thích hợp để đáp ứng 1 mục tiêu nào đó như chuẩn bị ngân sách, quản lý tài chính.
- Các quyết định có tính chiến lược: quyết định chính sách dài hạn, đầu tư dài hạn, tổ chức lại cơ quan, chiến lược tiếp thị.

Khi nào:

- Tài chính: mua trang thiết bị, phần mềm.
- Tổ chức: thích hợp.
- Hiệu quả và rủi ro: giảm rủi ro, không phải là hoàn vốn

Các loại bài toán áp dụng:

- Đơn giản.
- Có nhiều phương án chọn hơn, hàm đánh giá đơn giản.
- Phức tạp: đa chỉ tiêu.
- Rất phức tạp: vượt quá khả năng xử lý của con người.

Như thế nào:

- Thân thiện với người sử dụng về giao diện.
- Tính cập nhật thường xuyên cập nhật trong môi trường biến động.
- Độ chi tiết đảm bảo yêu cầu người sử dụng. Tần suất sử dụng cao, tính thuận tiện: mềm dẻo, thiết kế tốt

1.6 Môi trường ra quyết định và các yếu tố ảnh hưởng

1.6.1 Môi trường ra quyết định

Các yếu tố tác động: trực tiếp dễ nhìn ra, dễ lượng hoá, dễ nhìn, dễ hiểu, dễ cấu trúc hoá.

Các yếu tố về tổ chức tác động đến mọi bài toán quyết định:

- Chính sách: luật, mệnh lệnh, quan hệ, vay, trả.
- Cấu trúc tổ chức: cách quản lý, cách điều hành, quy chế.
- Uy tín của tổ chức có thể lượng hoá.
- Con người trong tổ chức: hành vi văn hoá, thái độ nhân cách.

Các yếu tố ngoại cảnh:

- Tình hình kinh tế.
- Tình hình thị trường.
- Tình hình môi trường.
- Pháp luật: chính sách có thể thay đổi còn pháp luật khó thay đổi.
- Sự chấp nhận của khách hàng.

Các yếu tố thông tin:

- Khả năng liên lạc.
- Độ bảo mật của thông tin.
- Độ tin cậy của thông tin: sự chính xác, mức độ cập nhật của thông tin.
- Các giải pháp thông tin tổng hợp, đa dạng, tỉ mỉ chi tiết.
- Giá cả của thông tin: thu nhập, truyền, xử lý.

Các mục tiêu về quản lý:

- Sự vận hành: kinh tế, hiệu quả, chất lượng, an toàn.
- Định lượng hay định tính.
- Mức độ rõ ràng của môi trường ra quyết định: rõ, ẩn hay mờ.

Phân loại môi trường ra quyết định:

- Môi trường truy nhập được và không truy nhập được.
- Môi trường tất định, không tất định.
- Môi trường tĩnh, động.
- Môi trường chắc chắn, không chắc chắn.

1.6.2 Các yếu tố ảnh hưởng đến ra quyết định

Trực tiếp, dễ nhìn thấy, dễ hiểu như là:

- Hạn chế về tài nguyên (ràng buộc ngân sách, khả năng phát triển sản xuất,...)
- Điều kiện vật lý (các tham số đo, các khoảng cách liên hệ,...)
- Các tham số chức năng ảnh hưởng đến hiệu quả (tỷ lệ sản xuất, độ tin cậy của nhà cung cấp, chất lượng sản phẩm, tay nghề nhân công, tỷ lệ sai hỏng, chỉ số kinh tế,...)

Các yếu tố tổ chức:

- Chính sách: các luật mệnh lệnh, các quan hệ, sự vay trả, sự định hướng thực hiện.

- Cấu trúc: vị trí địa lý, cách quản lý, nghiên cứu – phát triển, cách điều hành (tập trung, phân cấp, ma trận).
- Hình ảnh (uy tín): uy tín xã hội (công cộng), uy tín kinh doanh (thỏa mãn khách hàng, đảm bảo tài chính), uy tín trong nội bộ (không mâu thuẫn).
- Con người: điều kiện xã hội, ý thức chấp hành, hành vi văn hóa, thái độ, nhân cách.

Các yếu tố ngoại cảnh:

- Pháp luật, các quy định cần tuân thủ (thời gian sản xuất, môi trường sản xuất, luật chất thải, giới hạn giá cả, xuất khẩu,...)
- Kinh tế: sự đầu tư tài chính, giá cả, thuế.
- Môi trường: thời tiết, yếu tố địa lý, thiên tai.
- Thị trường: sự cạnh tranh, phát triển công nghệ mới.
- Đòi hỏi của khách hàng, nhu cầu của khách hàng.

Yếu tố thông tin:

- Khả năng thông tin: độ bảo mật, khả năng truyền thông, phương tiện lưu trữ, nguồn, kênh thông tin (bên trong, bên ngoài).
- Độ tin cậy: sự chính xác, cập nhật, chuẩn mực.
- Giải pháp: thông tin tỉ mỉ, tổng hợp, đa dạng.
- Giá cả: cho thu thập, chuẩn bị, kiểm tra, bảo hành thông tin, cả về thời gian tính và đòi hỏi phần cứng.

CHƯƠNG 2 : TÌM HIỂU VỀ DEEP LEARNING VÀ MÔ HÌNH CONVOLUTIONAL NEURAL NETWORK

2.1 Deep Learning là gì?

Deep Learning là một nhánh của Machine Learning và Khoa học dữ liệu bắt chước cách con người thu được những kiến thức cụ thể, giúp máy tính tự huấn luyện chính nó thực hiện những tác vụ tương tự con người, hỗ trợ cho việc dịch ngôn ngữ, phân loại các hình ảnh, nhận dạng giọng nói.

Các thuật toán Deep Learning là nhân tố quan trọng đằng sau những chiếc xe không người lái, cho phép chúng nhận ra biển báo dừng hoặc phân biệt người đi bộ với cột đèn. Công nghệ này là chìa khóa để con người có thể dùng giọng nói điều khiển các thiết bị tiêu dùng như điện thoại, máy tính bảng, TV và loa cầm tay. Qua từng bước phát triển, Deep Learning đang đạt được những kết quả chưa từng có trước đây.



Hình 2.1: Ô tô không người lái ứng dụng Deep Learning

2.2 Deep Learning hoạt động như thế nào?

Sau khi hiểu về định nghĩa **Deep Learning là gì** thì chúng ta cần hiểu cách thức nó vận hành ra sao để có cái nhìn tổng quát nhất.

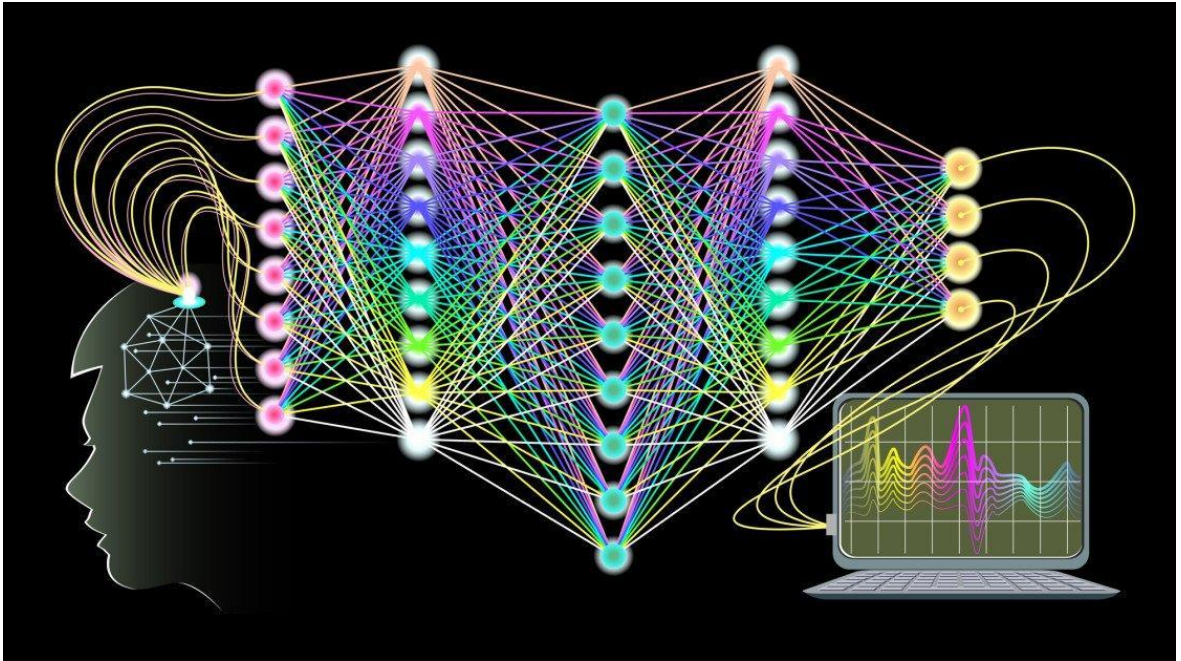
Deep Learning về cơ bản là một mạng nơ-ron gồm ba lớp trở lên, những mạng lưới nơ-ron này thu thập từ dữ liệu lớn (Big Data), học từ những dữ liệu đó và cố gắng “bắt chước” hành vi của não bộ con người. Công nghệ này hoạt động bằng cách khám phá các cấu trúc phức tạp trong dữ liệu mà chúng trải nghiệm, cụ thể là xây dựng các mô hình tính toán bao gồm nhiều lớp xử lý, mạng có thể tạo ra nhiều mức trừu tượng để biểu diễn dữ liệu.

Học sâu hoạt động bằng cách khám phá các cấu trúc phức tạp trong dữ liệu mà chúng thu thập. Các giai đoạn quá trình làm việc của Deep Learning như sau:

- ANN đặt ra một loạt các truy vấn đúng/sai nhị phân.
- Trích xuất các số từ các khối dữ liệu.
- Sắp xếp dữ liệu thành các danh mục dựa trên các phản hồi.
- Ghi nhãn dữ liệu.

Deep Learning rút ra kết luận và gắn nhãn dữ liệu mới, chưa được khám phá trong giai đoạn suy luận bằng cách sử dụng kiến thức và kinh nghiệm trước đó của nó.

Ví dụ: Một mô hình Deep Learning được gọi là mạng nơ-ron tích chập có thể được tạo ra bằng cách sử dụng số lượng lớn (lên đến hàng triệu) hình ảnh và mạng này học hỏi từ các pixel có trong hình ảnh mà nó thu được. Nó có thể phân loại các nhóm pixel đại diện cho các đặc điểm của mèo, với các nhóm đặc điểm như móng vuốt, tai và mắt cho biết sự hiện diện của mèo trong hình ảnh.



Hình 2.2: Minh họa mạng Neural network

2.3 Ưu và nhược điểm của Deep Learning

Một số ưu và nhược điểm của Deep Learning có thể kể đến như sau:

Ưu điểm	Nhược điểm
<ul style="list-style-type: none"> • Deep Learning được tối ưu tốt hơn nhờ sự bùng nổ của Internet và khả năng phát triển mạnh mẽ từ khả năng tính toán của máy tính. • Độ chính xác cao, đảm bảo cho các thiết bị điện tử tiêu dùng có thể đáp ứng được mọi nhu cầu cũng như mong đợi của người dùng. • Cung cấp hệ thống gợi ý trên các nền tảng lớn như: Facebook, Amazon, 	<ul style="list-style-type: none"> • Đòi hỏi một nguồn dữ liệu đầu vào khổng lồ để máy tính học hỏi. Quy trình này mất nhiều thời gian và công sức để xử lý mà chỉ có các Server chuyên nghiệp mới có thể làm được. Trường hợp không đủ dữ liệu đầu vào để xử lý, mọi thứ sẽ không thể diễn ra đúng như dự định của máy tính đưa ra.

<p>Netflix,..., giúp tăng độ tương tác của người dùng lên cao đáng kể.</p> <ul style="list-style-type: none"> • Khả năng nhận diện hình ảnh tốt, không chỉ xác định được đối tượng trong ảnh mà còn hiểu được nội dung và ngữ cảnh trong bức hình. • Độ an toàn cao, được tin dùng trong hệ thống lái xe tự động. 	<ul style="list-style-type: none"> • Chưa thể nhận biết được những thứ phức tạp. Bởi, kỹ thuật của Deep Learning hiện chưa đảm bảo tốt để trí tuệ nhân tạo có thể rút ra những kết luận một cách logic.
---	--

2.4 Ứng dụng của Deep Learning vào đời sống

Học sâu là một thuật ngữ tương đối xa lạ với những người chưa từng tìm hiểu về chủ đề này tuy nhiên Deep Learning đã được ứng dụng vào rất nhiều khía cạnh cũng như lĩnh vực trong đời sống của chúng ta, giúp ích rất nhiều cho các nhà khoa học dữ liệu trong việc thu thập, giải thích và phân tích khối lượng dữ liệu khổng lồ một cách dễ dàng và nhanh chóng. Deep Learning đã mang lại nhiều ứng dụng tích cực và đa dạng trong đời sống hàng ngày, từ nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, đến tự động lái xe và chăm sóc sức khỏe. Các ứng dụng này không chỉ cải thiện hiệu suất và tiện ích của nhiều lĩnh vực, mà còn mở ra những triển vọng mới trong việc giải quyết các thách thức xã hội và kinh tế. Đồng thời, Deep Learning đang chứng minh vai trò quan trọng trong việc thúc đẩy sự phát triển và tiến bộ trong các lĩnh vực quan trọng của xã hội. Dưới đây là một số những ví dụ cụ thể về hình thức mà Deep Learning hiện diện và giúp ích cho cuộc sống của con người.

2.4.1 Digital Marketing

Thời đại chuyển đổi số trên khắp mọi lĩnh vực hiện nay khiến việc ứng dụng trí tuệ nhân tạo trong Digital Marketing không còn quá xa lạ đối với dân

kinh doanh lẫn dân công nghệ. Việc sử dụng Deep Learning trong chiến dịch Marketing sẽ giúp mang lại trải nghiệm tốt hơn cho khách hàng và tăng doanh thu cho doanh nghiệp. Một số ví dụ điển hình như lợi ích của hệ thống Chatbot, xây dựng nội dung cá nhân hóa, phân tích người dùng bằng cách nhận diện hình ảnh, ...



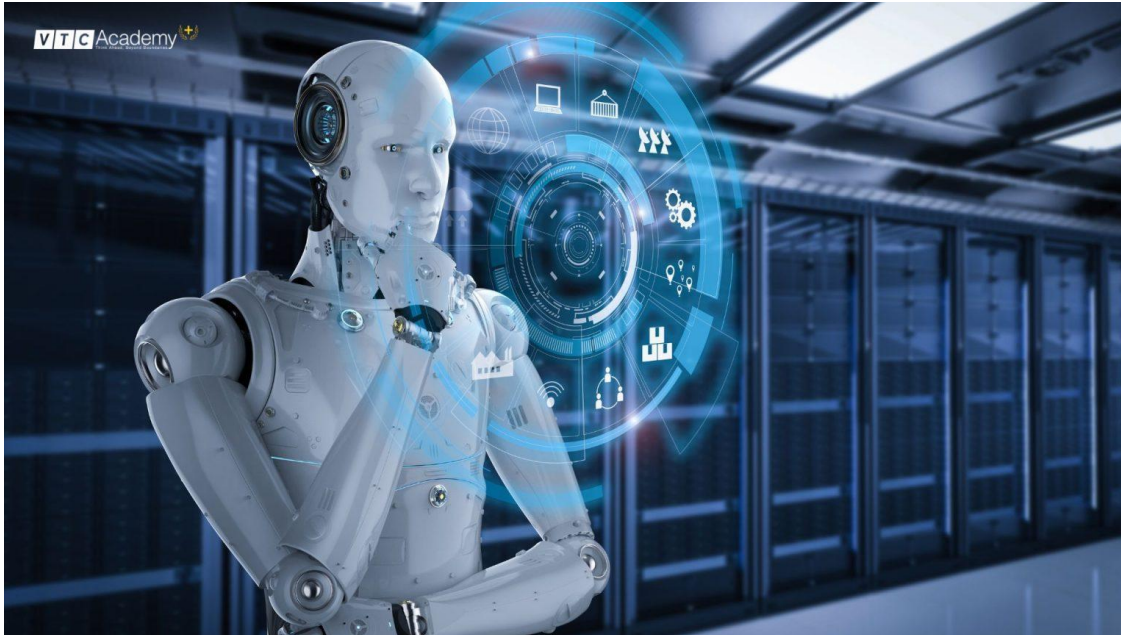
Hình 2.3: Ứng dụng Deep Learning vào Digital Marketing giúp chiến lược Marketing thành công hơn

2.4.2 Chế tạo Robot

Nhiều sự phát triển gần đây trong chế tạo robot đã được thúc đẩy bởi những tiến bộ trong AI và Deep Learning. Ví dụ, AI cho phép robot cảm nhận và phản ứng với môi trường quanh nó, điều hướng, phân loại và xử lý các đồ vật không bằng phẳng, dễ vỡ hoặc sắp xếp những vật lộn xộn lại với nhau.

Sự phát triển của AI có nghĩa là chúng ta có thể mong đợi các robot trong tương lai sẽ ngày càng được sử dụng làm trợ lý cho con người. Chúng sẽ không chỉ được sử dụng để hiểu và trả lời các câu hỏi, như một số được sử dụng ngày nay. Người dùng kỳ vọng chúng sẽ có thể thực hiện các lệnh bằng giọng nói và cử chỉ, thậm chí dự đoán hành động tiếp theo của con người. Ngày nay, rô-bốt đã

hoạt động cộng tác cùng với con người, trong đó con người và robot thực hiện các nhiệm vụ riêng biệt phù hợp nhất với thế mạnh của mỗi bên .



Hình 2.4: Lĩnh vực chế tạo Robot được dự đoán sẽ phát triển hơn trong tương lai nhờ sự bùng nổ của AI

2.4.3 Hệ thống lái xe tự động

Một trong những sản phẩm áp dụng Deep Learning mới và hiện đại nhất hiện nay là hệ thống lái xe tự động, được xây dựng dựa trên các mạng nơ-ron cấp cao.

Các mô hình Deep Learning đóng vai trò nhận diện các đối tượng trong môi trường xung quanh xe, tính toán khoảng cách giữa xe và các phương tiện khác, xác định vị trí làn đường, tín hiệu giao thông, nhận diện chướng ngại vật,... từ đó đưa ra được các quyết định tối ưu và nhanh chóng nhất. Các thuật toán Học sâu càng nhận được nhiều dữ liệu thì càng có khả năng hành động giống như con người trong quá trình xử lý thông tin — phân biệt được biển báo dừng dù bị phủ tuyết ...

Một trong những hãng xe đình đám và tiên phong trong việc sản xuất xe tự lái hiện nay là Tesla.



Hình 2.5: Tesla – hãng xe lái tự động nổi tiếng nhất hiện nay

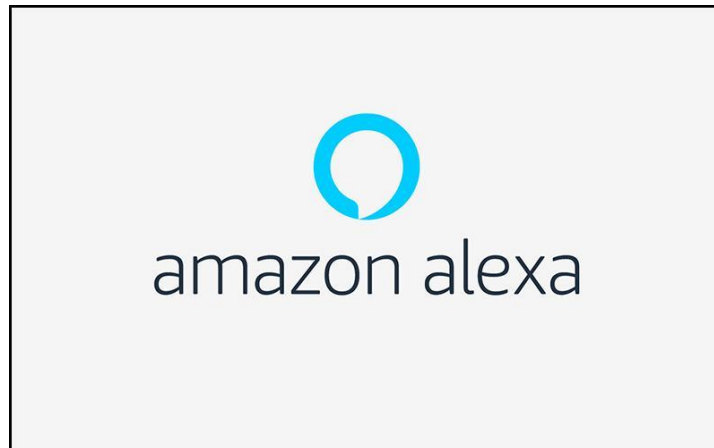
2.4.4 Trợ lý ảo

Trợ lý ảo không còn quá xa lạ với công chúng trong những năm gần đây bởi những hiệu quả vượt trội mà chúng mang lại. Trợ lý ảo hay Virtual Assistant đang được ứng dụng rất nhiều trong đời sống hàng ngày, phổ biến nhất nhất có thể kể đến Alexa hay Siri, Google Assistant. Mỗi tương tác với các trợ lý này cung cấp cho họ cơ hội tìm hiểu thêm về giọng nói và ngữ điệu của bạn, từ đó cung cấp cho bạn trải nghiệm tương tác như phiên bản thứ 2 của con người.

Trợ lý ảo sử dụng Deep Learning để biết thêm về người dùng, từ sở thích ăn uống đến các điểm truy cập nhiều nhất hoặc các bài hát yêu thích của bạn. Công nghệ này học cách hiểu và thực hiện các mệnh lệnh thông qua đánh giá ngôn ngữ tự nhiên của con người.

Ngoài ra Deep Learning còn có khả năng giúp trợ lý ảo dịch bài phát biểu thành văn bản, ghi chú lại và đặt lịch hẹn cho bạn; nhắc lịch hẹn và tự động trả lời

các cuộc gọi cụ thể để phối hợp các nhiệm vụ giữa bạn và các thành viên chung team; tạo hoặc gửi bản sao email phù hợp.



Hình 2.6: Alexa – trợ lý ảo thông minh được phát triển bởi Amazon

2.4.5 Nghiên cứu y học

Từ chẩn đoán bệnh, chẩn đoán khối u cho đến các loại thuốc được tạo riêng cho từng bộ gen cá nhân, ứng dụng của Deep Learning vào y học đang nhận được sự chú ý từ nhiều công ty dược phẩm và y tế lớn nhất hiện nay.

Các nhà nghiên cứu ung thư đang sử dụng Deep Learning để tự động phát hiện tế bào ung thư. Các nhóm nghiên cứu tại UCLA đã chế tạo một kính hiển vi tiên tiến mang lại bộ dữ liệu đa chiều được sử dụng để đào tạo một ứng dụng Deep Learning nhằm xác định chính xác các tế bào ung thư.

Số hóa hồ sơ bệnh nhân và đọc hình ảnh cũng là minh chứng nổi bật của việc ứng dụng công nghệ này vào trong ngành y tế. Deep Learning phân tích tự động hình ảnh chụp cắt lớp của bệnh nhân, giúp bác sĩ chẩn đoán chính xác và tình trạng bệnh được phát hiện nhanh chóng, có biện pháp điều trị từ sớm. Quá trình kiểm tra sức khỏe cũng trở nên đơn giản khi mọi dữ liệu của người bệnh đã được ghi nhận trên hệ thống chung.



Hình 2.7: Deep Learning ứng dụng rộng rãi trong nghiên cứu y học

2.4.6 Nhận diện khuôn mặt

Deep Learning được sử dụng trong nhận diện khuôn mặt không chỉ nhằm phục vụ cho mục đích bảo mật mà còn được ứng dụng việc phân loại hình ảnh. Chắc hẳn, chúng ta đều đã từng thấy máy tính tự động nhận diện và phân loại các hình ảnh trên điện thoại.

Ví dụ: Facebook có thể tự động gắn thẻ chính bạn và bạn bè của bạn, hoặc Google Photos có thể tự động gắn nhãn ảnh của bạn để tìm kiếm dễ dàng hơn. Với Deep Learning, bạn có thể dễ dàng tìm và phân loại các hình ảnh theo ngày, sự kiện mà không phải dùng thao tác thủ công mất thời gian. Và trong tương lai gần, người dùng còn có thể thanh toán các hóa đơn bằng chính khuôn mặt của mình.

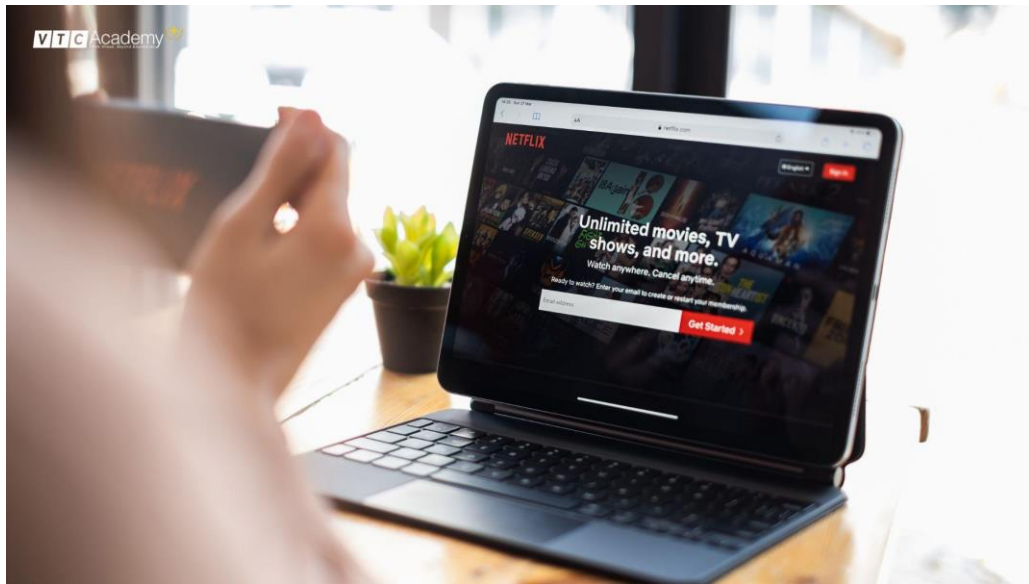
Tuy nhiên, thách thức đặt ra đối với Deep Learning khi nhận diện khuôn mặt là làm thế nào để xác định được chính xác một người ngay cả khi người đó thay đổi kiểu tóc, để râu hay cạo râu, hoặc khi chất lượng ảnh kém do ánh sáng hoặc môi trường xung quanh...



Hình 2.8: Deep Learning được sử dụng trong Face ID

2.4.7 Mua sắm và giải trí

Bạn đã bao giờ thắc mắc làm thế nào Tiktok đưa ra các đề xuất về nội dung bạn nên xem tiếp theo chưa? Hoặc Shopee đưa ra ý tưởng về những gì bạn nên mua tiếp theo và những gợi ý đó chính xác là những gì bạn cần nhưng chưa từng biết trước đây? Đúng, đó là cách mà các thuật toán Deep Learning đang hoạt động, len lỏi vào trong các hoạt động giải trí của chúng ta. Deep Learning càng “học” được từ nhiều dữ liệu, chúng càng trở nên tốt hơn, từ đó những gợi ý đưa ra cũng chính xác hơn.



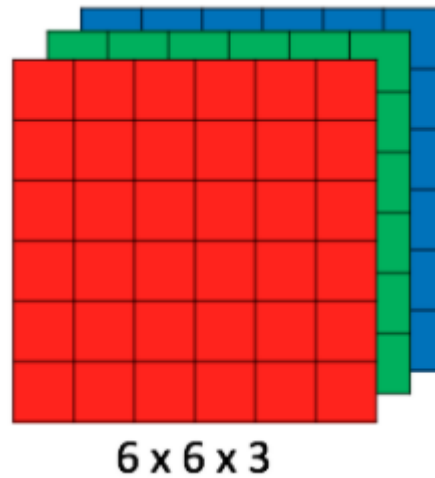
Hình 2.9: Netflix sử dụng Deep Learning để gợi ý những bộ phim phù hợp với sở thích người xem

2.5. Tổng quan về mạng CNN

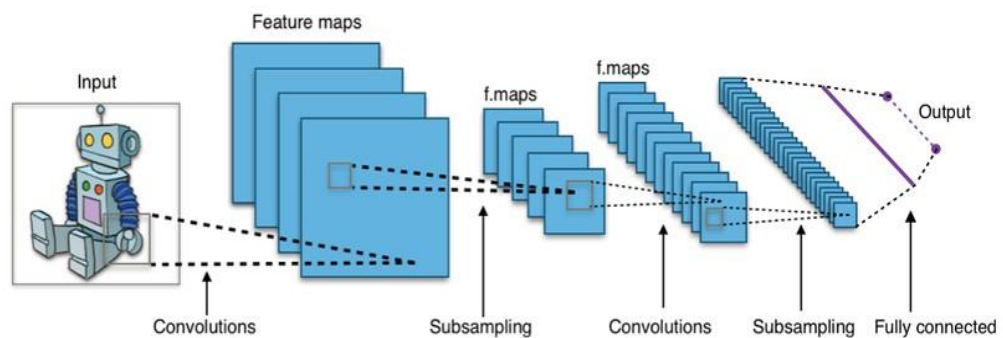
2.5.1 Khái niệm chung

- Trong mạng neural, mô hình mạng neural tích chập (CNN) là 1 trong những mô hình để nhận dạng và phân loại hình ảnh. Trong đó, xác định đối tượng và nhận dạng khuôn mặt là 1 trong số những lĩnh vực mà CNN được sử dụng rộng rãi.

- CNN phân loại hình ảnh bằng cách lấy 1 hình ảnh đầu vào, xử lý và phân loại nó theo các hạng mục nhất định (Ví dụ: Chó, Mèo, Hổ, ...). Máy tính coi hình ảnh đầu vào là 1 mảng pixel và nó phụ thuộc vào độ phân giải của hình ảnh. Dựa trên độ phân giải hình ảnh, máy tính sẽ thấy $H \times W \times D$ (H: Chiều cao, W: Chiều rộng, D: Độ dày).



Về kỹ thuật, mô hình CNN để training và kiểm tra, mỗi hình ảnh đầu vào sẽ chuyển nó qua 1 loạt các lớp tích chập với các bộ lọc (Kernels), tổng hợp lại các lớp được kết nối đầy đủ (Full Connected) và áp dụng hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1.



2.5.2 Các kiểu tầng trong mạng CNN

a, Convolution Layer

Convolution (lớp tích chập) là lớp đầu tiên trích xuất các đặc tính từ hình ảnh. Tham số lớp này bao gồm một tập hợp các bộ lọc có thể học được. Các bộ lọc đều nhỏ thường có kích cỡ hai chiều đầu tiên khoảng 3×3 hoặc 5×5 , và có độ sâu bằng với độ sâu của đầu vào đầu vào. Bằng cách trượt dần bộ lọc theo chiều ngang và dọc trên ảnh, chúng thu được một Feature Map chứa các đặc trưng được trích xuất từ trên hình ảnh đầu vào.

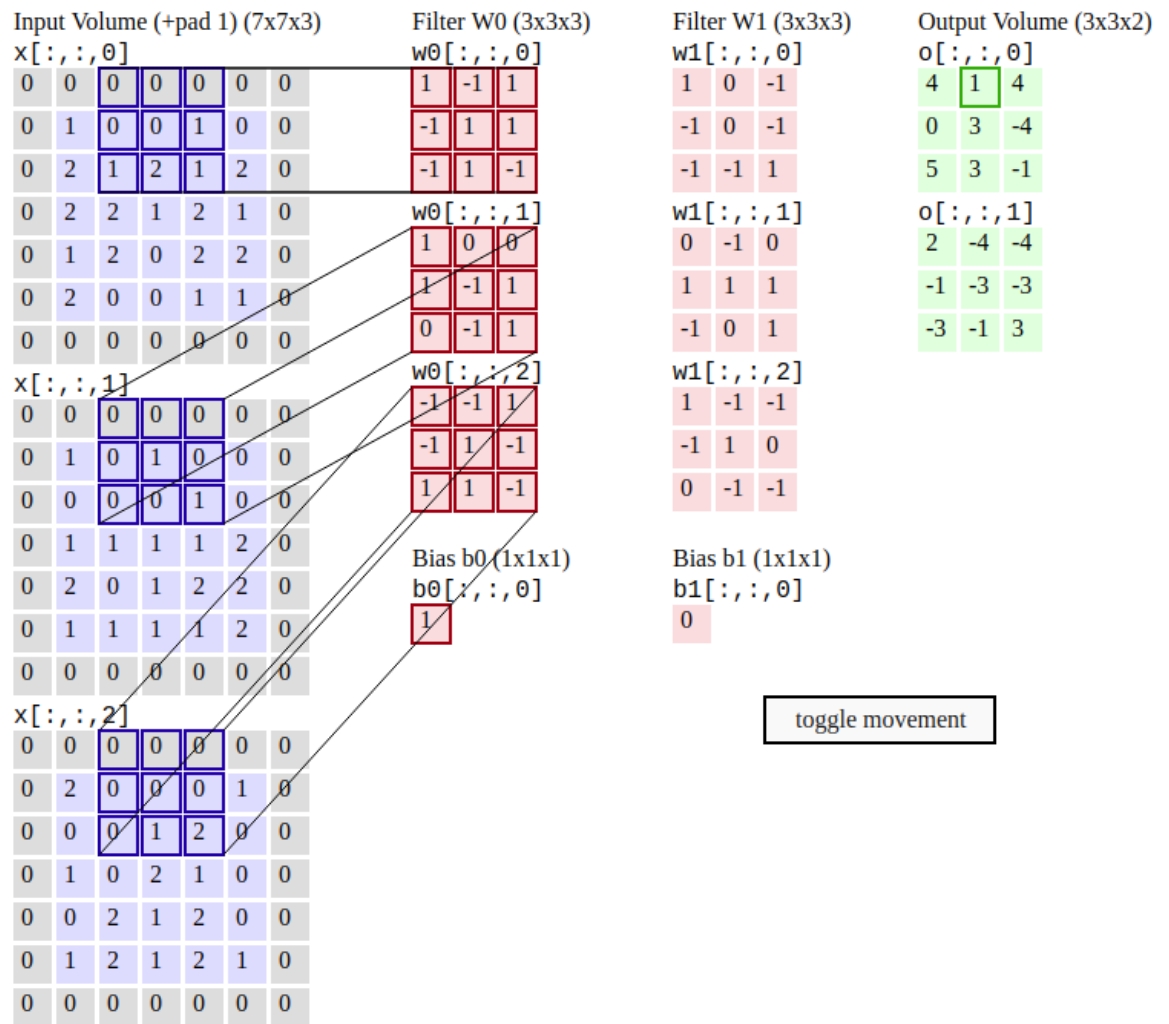
Quá trình trượt các bộ lọc thường có các giá trị được quy định bao gồm:

- Bước nhảy – Stride: Stride là số pixel thay đổi trên ma trận đầu vào. Khi stride là 1 thì ta di chuyển các kernel 1 pixel. Khi stride là 2 thì ta di chuyển các kernel đi 2 pixel và tiếp tục như vậy. Hình dưới là lớp tích chập hoạt động với stride là 2.
- Đường viền – Padding: Đôi khi kernel không phù hợp với hình ảnh đầu vào. Ta có 2 lựa chọn:
 - + Chèn thêm các số 0 vào 4 đường biên của hình ảnh (padding).
 - + Cắt bớt hình ảnh tại những điểm không phù hợp với kernel.
- Hàm phi tuyến – ReLU:

ReLU viết tắt của Rectified Linear Unit, là 1 hàm phi tuyến. Với đầu ra là: $f(x) = \max(0, x)$.

Tại sao ReLU lại quan trọng: ReLU giới thiệu tính phi tuyến trong ConvNet. Vì dữ liệu trong thế giới mà chúng ta tìm hiểu là các giá trị tuyến tính không âm.

Có 1 số hàm phi tuyến khác như tanh, sigmoid cũng có thể được sử dụng thay cho ReLU. Hầu hết người ta thường dùng ReLU vì nó có hiệu suất tốt.



Với mỗi kernel khác nhau ta sẽ học được những đặc trưng khác nhau của ảnh, nên trong mỗi convolutional layer ta sẽ dùng nhiều kernel để học được nhiều thuộc tính của ảnh. Vì mỗi kernel cho ra output là 1 matrix nên k kernel sẽ cho ra k output matrix. Ta kết hợp k output matrix này lại thành 1 tensor 3 chiều có chiều sâu k. Output của convolutional layer sẽ qua hàm activation function trước khi trở thành input của convolutional layer tiếp theo.

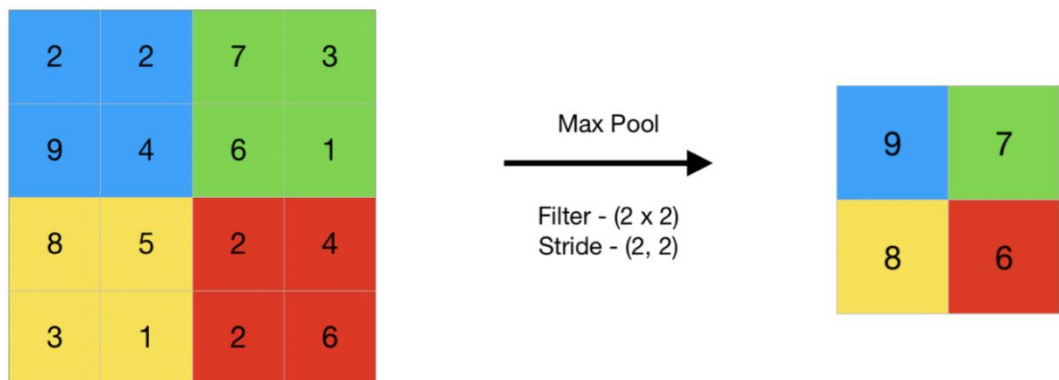
b, Pooling layer

Mục tiêu của lớp gộp là lấy các tính năng quan trọng nhất từ ma trận phức tạp. Điều này được thực hiện bằng cách áp dụng một số thao tác tổng hợp, giúp

giảm kích thước của bản đồ đặc trưng (ma trận phức tạp), do đó giảm bộ nhớ được sử dụng trong khi huấn luyện mạng. Pooling cũng có liên quan để giảm thiểu tình trạng tràn bị quá mức.

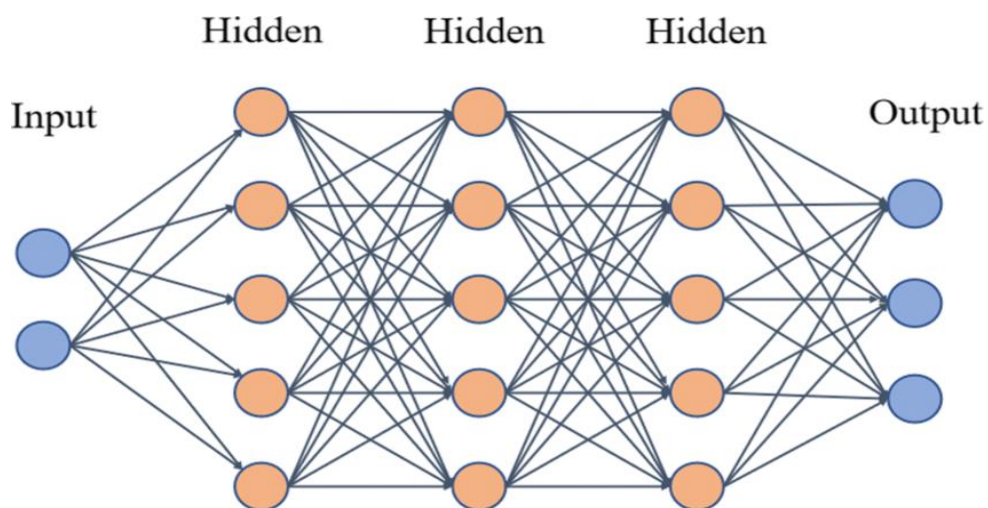
Các hàm tổng hợp phổ biến nhất có thể được áp dụng là:

- Max pooling: là một loại phép gộp được sử dụng trong mạng nơ-ron tích chập (CNN) và các kiến trúc liên quan. Nó thường được áp dụng sau lớp tích chập để giảm kích thước của đầu vào và tạo ra các phiên bản đã tạo ra của đặc trưng cục bộ.
- Sum pooling: là một phương pháp trong xử lý ảnh và mạng học sâu, được sử dụng để giảm kích thước của feature maps trong quá trình trích xuất thông tin từ hình ảnh. Tương tự như Average Pooling và Max Pooling, Sum Pooling cũng là một loại phép gộp thông tin trong mạng neural.
- Average pooling : là một phép tích chập thường được sử dụng trong mạng học sâu và xử lý ảnh. Nó là một phần quan trọng của các kiến trúc mạng thần kinh như Convolutional Neural Networks (CNNs) để giảm kích thước của đặc trưng (feature maps) trong quá trình trích xuất thông tin từ hình ảnh.



c, Fully connected layers

Fully connected layer là lớp cuối cùng trong mạng CNN có chức năng thực hiện việc phân loại hoặc dự đoán các giá trị đầu ra của mô hình. Convolutional layer và pooling layer đã thực hiện các phép tích chập và giảm kích thước của dữ liệu đầu vào, tạo ra các đặc trưng cục bộ cho ảnh. Tuy nhiên thì chúng chỉ tạo ra các đặc trưng cục bộ và không liên kết các đặc trưng đó để đưa ra một dự đoán cuối cùng. Để phân loại chính xác hơn, chúng ta cần kết nối tất cả các đặc trưng đó với nhau và tính toán xác suất cho các lớp đầu ra. Fully connected layer trong mạng CNN được thiết kế có một số nơ-ron kết nối với tất cả các nơ-ron trong lớp trước đó (có thể là một lớp convolutional layer hoặc pooling layer) để nhận đầu vào là tất cả các đặc trưng từ lớp trước đó. Fully connected layer lúc này có nhiệm vụ kết nối các đặc trưng cục bộ đã được trích xuất từ các lớp convolutional layer và pooling layer để đưa ra một dự đoán cuối cùng.



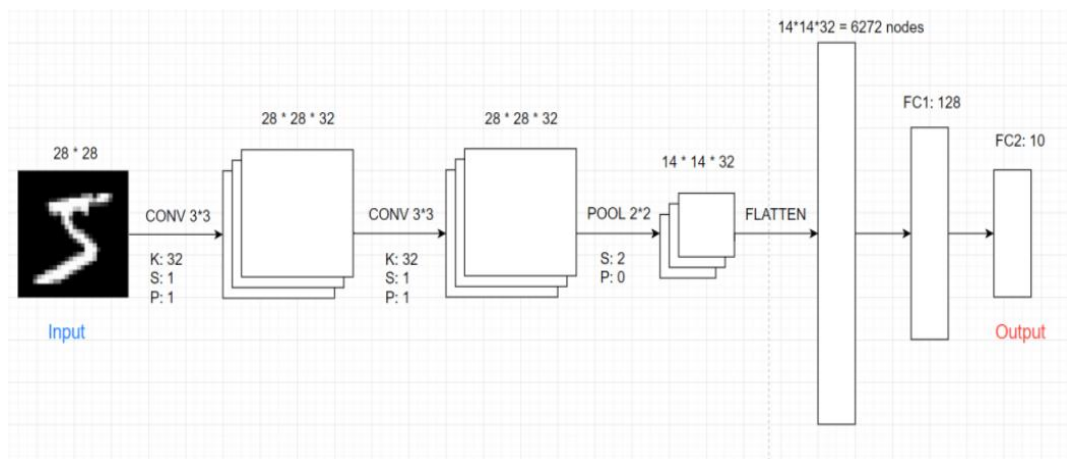
Fully connected layer with ReLU output

Hình 2.10: Minh họa lớp kết nối đầy đủ với đầu ra ReLU

Fully connected layer giúp mạng CNN học các mối quan hệ phức tạp giữa các đặc trưng đầu vào và tạo ra các phân loại chính xác hơn. Tuy nhiên khi sử dụng quá nhiều fully connected layer trong một mạng CNN sẽ dẫn đến hiện tượng overfitting, khi mô hình chỉ học nhớ dữ liệu huấn luyện mà không có khả năng dự đoán đúng trên các dữ liệu mới. Do đó số lượng fully connected layer cần được cân nhắc để tối ưu hóa hiệu suất của mạng CNN.

2.5.3 Ví dụ minh họa mô hình CNN

Mô hình chung bài toán CNN: Input image \rightarrow Convolutional layer (Conv) + Pooling layer (Pool) \rightarrow Fully connected layer (FC) \rightarrow Output.



Các lớp của mô hình được mô tả ,hoạt động theo thứ tự:

- Ảnh đầu vào là ảnh màu, kích thước 28x28
- Lớp tích chập thứ nhất sử dụng 32 mặt nạ tích chập, padding=1, stride=1. Mặt nạ tích chập kích thước 3x3. Thu được ảnh đầu ra gồm 32 ảnh có kích thước 28x28 sau đó dùng hàm kích hoạt Rectified Linear Unit (ReLU)
- Lớp tích chập thứ hai sử dụng 32 mặt nạ tích chập, padding=1, stride=1. Mặt nạ tích chập kích thước 3x3. Thu được ảnh đầu ra gồm 32 ảnh có

kích thước 28x28 sau đó dùng hàm kích hoạt Rectified Linear Unit (ReLU)

- Lớp max pooling thứ nhất kích thước 2x2 cho đầu ra là 32 ảnh kích thước 14x14
- Tiếp theo dùng hàm FLATTEN để làm phẳng dữ liệu cho trước khi đến lớp Fully Connected , đặc trưng trong 32 ảnh tạo thành 1 vector có độ dài $14*14*32 = 6272$ (nodes)
- Lớp full connect 1 (FC1) gồm 128 neural sau khi được phân lớp từ 6272 nodes dữ liệu đầu vào
- Lớp FC2 gồm 10 neural tương ứng đầu ra, các neural sẽ cho biết xác suất tương ứng với các lớp đối tượng của dữ liệu đầu vào

Ưu điểm

- Trích xuất đặc trưng tự động: CNN tự động học cách trích xuất các đặc trưng quan trọng từ dữ liệu ảnh mà không cần phải xác định rõ ràng các đặc trưng cụ thể. Điều này giúp giảm thiểu công việc thủ công và tốn thời gian để thiết lập các đặc trưng thủ công.
- Xử lý dữ liệu không gian: CNN được thiết kế đặc biệt để xử lý dữ liệu không gian như ảnh và âm thanh. Các lớp tích chập và gộp cho phép nó hiệu quả trong việc trích xuất thông tin từ các cấu trúc không gian này.
- Phân loại và nhận dạng tốt: CNN đã đạt được hiệu suất ấn tượng trong các nhiệm vụ phân loại và nhận dạng. Điều này bao gồm nhận biết khuôn mặt, phân loại đối tượng, dự đoán từ ảnh, và nhiều nhiệm vụ khác.
- Tích hợp thông tin cục bộ: Các lớp tích chập trong CNN cho phép mạng tập trung vào việc xử lý thông tin cục bộ trong các phần nhỏ

của dữ liệu ảnh. Điều này giúp mạng nhận biết các đặc trưng cục bộ và tạo ra biểu diễn chất lượng cao.

- Khả năng học tập sâu: CNN có thể được thiết kế với nhiều lớp và kích thước khác nhau, cho phép nó học các đặc trưng ngày càng phức tạp và sâu hơn. Điều này giúp mạng học được các biểu diễn tổng hợp và trừu tượng từ dữ liệu ảnh.
- Hiệu suất cao: Các mô hình CNN tiên tiến có khả năng đạt được hiệu suất rất cao trong các thách thức như các cuộc thi nhận dạng hình ảnh (ImageNet) và các ứng dụng thực tiễn như xe tự hành.

Nhược điểm

- Số lượng tham số và tài nguyên tính toán: Các mạng CNN tiên tiến thường có hàng triệu hoặc thậm chí hàng tỷ tham số. Điều này đòi hỏi tài nguyên tính toán lớn để huấn luyện và triển khai, đặc biệt là trên các thiết bị có tài nguyên hạn chế như điện thoại di động.
- Overfitting: Trong mạng CNN lớn, có nguy cơ mô hình bị overfitting, tức là mô hình học quá nhiều từ dữ liệu huấn luyện và không tổng quát hóa tốt trên dữ liệu mới. Điều này có thể dẫn đến hiệu suất kém trên dữ liệu không nhìn thấy trước.
- Dữ liệu huấn luyện hạn chế: Mạng CNN yêu cầu một lượng lớn dữ liệu huấn luyện để học hiệu quả. Nếu dữ liệu không đủ đa dạng hoặc không đại diện, mô hình có thể không học được các biểu diễn tổng quát.
- Chuẩn bị dữ liệu: Trước khi áp dụng CNN, dữ liệu thường cần phải được chuẩn bị và tiền xử lý một cách cẩn thận, bao gồm việc thay đổi kích thước ảnh, chuẩn hóa dữ liệu và loại bỏ nhiễu.
- Khả năng biểu diễn không tốt cho dữ liệu không gian tương đối dài: Mạng CNN không thể hiệu quả biểu diễn thông tin về thứ tự và tương quan không gian dài hơn, chẳng hạn như các dãy thời gian dài.

CHƯƠNG 3: XÂY DỰNG HỆ QUYẾT ĐỊNH SỬ DỤNG MÔ HÌNH CNN TRONG BÀI TOÁN NHẬN DIỆN CHỮ SỐ VIẾT TAY

3.1 Thuật toán áp dụng

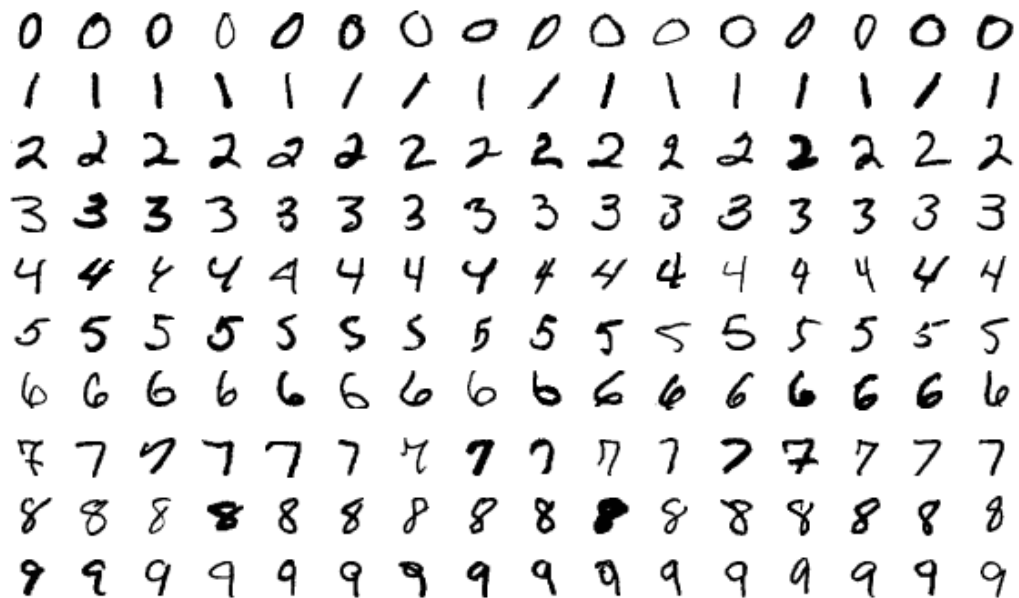
Trong bài toán nhận dạng chữ số viết tay, việc chọn lựa thuật toán phù hợp đóng vai trò quan trọng để đạt được hiệu suất tốt nhất. Trong nghiên cứu này, nhóm chúng em quyết định sử dụng thuật toán Convolutional Neural Network (CNN) với những lý do chủ yếu sau đây:

- CNN giữ nguyên hiệu suất phân loại khi dữ liệu trải qua các biến đổi như dịch chuyển và xoay. Điều này làm cho nó thích hợp cho việc nhận dạng chữ số viết tay trong các tình huống thực tế, nơi vị trí và góc nhìn của chữ số có thể thay đổi.
- CNN được thiết kế để tự động học các đặc trưng từ dữ liệu mà không yêu cầu sự giám sát cụ thể. Trong trường hợp nhận dạng chữ số viết tay, có nhiều biến thể về viết tay và đặc điểm của chữ số. CNN có khả năng học và ứng dụng các đặc trưng phức tạp này, giúp nó hiệu quả trong việc nhận dạng các biểu thức viết tay đa dạng.
- Một ưu điểm quan trọng của CNN là khả năng chia sẻ trọng số trong các lớp convolutional. Điều này giúp giảm lượng tham số cần học, đặc biệt là khi có sẵn một lượng dữ liệu lớn về chữ số viết tay. Chia sẻ trọng số giúp tăng khả năng tổng quát hóa của mô hình và giảm nguy cơ quá mức học từ dữ liệu huấn luyện.

Với những đặc điểm trên khiến cho CNN trở thành một lựa chọn lý tưởng để giải quyết bài toán nhận dạng chữ số viết tay, và sự chọn lựa này sẽ đạt được kết quả chính xác và đáng tin cậy trong mô hình nhận dạng chữ số viết tay.

3.2 Dữ liệu đầu vào

Nhóm sử dụng bộ dữ liệu MNIST để huấn luyện và kiểm thử mô hình Convolutional Neural Network (CNN) cho bài toán nhận dạng chữ số viết tay. Bộ dữ liệu MNIST là một trong những bộ dữ liệu phổ biến nhất trong lĩnh vực học máy và thị giác máy tính, được sử dụng rộng rãi để đánh giá hiệu suất của các mô hình trong việc nhận dạng và phân loại chữ số từ 0 đến 9.



Hình 2.11. Một vài ảnh mẫu từ tập dữ liệu thử nghiệm MNIST

Bộ dữ liệu MNIST gồm:

- Bộ dữ liệu huấn luyện (training.csv):
 - + Bộ dữ liệu huấn luyện gồm các hình ảnh chữ số viết tay từ 0 đến 9.
 - + Mỗi hình ảnh có kích thước là 28x28 pixel.
 - + Tổng cộng có 42000 hình ảnh trong tập huấn luyện.
- Bộ dữ liệu kiểm thử (test.csv):
 - + Bộ dữ liệu kiểm thử cũng chứa các hình ảnh chữ số tương ứng với tập huấn luyện.
 - + Mỗi hình ảnh có định dạng và kích thước giống như tập huấn luyện.

+ Số lượng hình ảnh trong tập kiểm thử là 28000.

Bộ dữ liệu MNIST được chọn vì tính đa dạng và phổ biến của nó trong lĩnh vực nhận dạng chữ số. Việc sử dụng bộ dữ liệu này giúp đánh giá khả năng tổng quát hóa và hiệu suất của mô hình CNN trong việc nhận diện chữ số từ dữ liệu mới, chưa được mô hình thấy trước đó.

3.3 Cài đặt

3.3.1 Khai báo thư viện

```
[3] from keras.datasets import mnist
import tensorflow as tf
import numpy as np
import pandas as pd
from keras.utils import to_categorical
from tensorflow.keras import layers, models
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D
import matplotlib.pyplot as plt
```

3.3.2 Lấy cơ sở dữ liệu MNIST

```
[4] # x: images, y: labels
(train_x, train_y), (test_x, test_y) = mnist.load_data()
# train_images là dữ liệu đã train

print(train_x.shape)
# 60.000 bức ảnh với kích thước 28*28 pixel
print(train_y.shape)

print(test_x.shape)
# 10.000 bức ảnh với kích thước 28*28 pixel
print(test_y.shape)
```

3.3.3 Tập kiểm thử - Validation

```
[5] # lấy một phần con của dữ liệu huấn luyện gốc (train_x là dữ liệu đầu vào và train_y là nhãn tương ứng) để kiểm thử
    val_x, val_y = train_x[50000:60000, :], train_y[50000:60000] # từ hàng 50000 đến 59999
    train_x, train_y = train_x[:50000, :], train_y[:50000] # từ đầu đến hàng 49999
    print(val_x.shape)
    print(val_y.shape)
    print(train_x.shape)
    print(train_y.shape)
```

```
(10000, 28, 28)
(10000,)
(50000, 28, 28)
(50000,)
```

3.3.4 Chuẩn hóa dữ liệu

```
[6] so_kenh_mau = 1 # Điều này có nghĩa là hình ảnh sẽ được coi là ảnh grayscale, có một kênh màu duy nhất.
    cao_pixel = 28
    rong_pixel = 28

    # train_images.shape là một tuple chứa thông tin về kích thước của X_train. Trong trường hợp này, .shape[0] trả về số lượng mẫu.
    # In ra số lượng mẫu hình ảnh trong tập huấn luyện
    print(train_x.shape[0])

    # reshape tập train và test, dữ liệu hình ảnh sẽ được biểu diễn dưới dạng một tensor 4D
    # Chuyển đổi kiểu dữ liệu của mỗi phần tử trong tensor thành kiểu dữ liệu float32
    # Sau đó, dữ liệu được chuẩn hóa bằng cách chia cho 255.0 để đưa giá trị pixel về khoảng [0, 1]
    train_x = train_x.reshape((train_x.shape[0], cao_pixel, rong_pixel, so_kenh_mau)).astype('float32') / 255.0
    val_x = val_x.reshape((val_x.shape[0], cao_pixel, rong_pixel, so_kenh_mau)).astype('float32') / 255.0
    test_x = test_x.reshape((test_x.shape[0], cao_pixel, rong_pixel, so_kenh_mau)).astype('float32') / 255.0

    print("Shape of train_x: {}".format(train_x.shape))
    print("Shape of val_x: {}".format(val_x.shape))
    print("Shape of test_x: {}".format(test_x.shape))
```

```
50000
Shape of train_x: (50000, 28, 28, 1)
Shape of val_x: (10000, 28, 28, 1)
Shape of test_x: (10000, 28, 28, 1)
```

```
[7] # Thực hiện one-hot encoding cho nhãn trong tập train_labels và test_labels với 10 là số lượng lớp.
    # One-hot encoding là một quá trình biến đổi nhãn thành vector có độ dài bằng số lượng lớp,
    # trong đó chỉ có một phần tử có giá trị 1, và phần còn lại có giá trị 0

    train_Y = to_categorical(train_y, 10)
    val_Y = to_categorical(val_y, 10)
    test_Y = to_categorical(test_y, 10)

    print('Dữ liệu labels ban đầu: ', train_y[100])
    print('Dữ liệu label sau one-hot encoding: ', train_Y[100])
```

```
Dữ liệu labels ban đầu: 5
Dữ liệu label sau one-hot encoding: [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
```

3.3.5 Xây dựng mô hình CNN

```
[8] # Khởi tạo mô hình tuần tự (sequential model) - Mô hình tuần tự là một chồng các lớp tuyến tính trong Keras
model = models.Sequential()

#layer 1
# Lớp Convolutional đầu tiên với 32 bộ lọc kích thước (3, 3) và hàm kích hoạt ReLU (Rectified Linear Unit)
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
# Lớp MaxPooling để giảm kích thước ảnh
model.add(layers.MaxPooling2D((2, 2)))

#layer 2
# Lớp Convolutional thứ hai với 64 bộ lọc kích thước (3, 3) và hàm kích hoạt ReLU
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
# Lớp MaxPooling để giảm kích thước ảnh
model.add(layers.MaxPooling2D((2, 2)))

#layer 3
# Lớp Convolutional thứ ba với 64 bộ lọc kích thước (3, 3) và hàm kích hoạt ReLU
model.add(layers.Conv2D(64, (3, 3), activation='relu'))

#FCN layer
# Lớp Flatten để chuyển từ tensor 3D thành vector 1D
model.add(layers.Flatten())
# Lớp fully connected (Dense) với 64 đơn vị và hàm kích hoạt ReLU
model.add(layers.Dense(64, activation='relu'))
# Lớp fully connected (Dense) với 10 đơn vị (số lượng classes) và hàm kích hoạt softmax (Output Layer)
model.add(layers.Dense(10, activation='softmax'))

#tóm tắt mô hình
model.summary()
```

meet.google.com is sharing your screen. [Stop sharing](#) [Help](#)

3.3.6 Huấn luyện mô hình

```
# Sử dụng optimizer Adam, hàm loss là categorical_crossentropy cho bài toán phân loại nhiều classes
EPOCHS = 20 # số lần lặp trên toàn bộ dữ liệu
BATCH_SIZE = 32 # số mẫu được sử dụng mỗi lần

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Huấn luyện mô hình trên dữ liệu huấn luyện với số epoch là 20 và batch size là 32
history = model.fit(train_x, train_y, validation_data = (val_x, val_y), epochs = EPOCHS, batch_size = BATCH_SIZE)
```

3.3.7 Đánh giá và lưu mô hình huấn luyện

```
[ ] # Lấy thông tin về độ chính xác và độ mất mát từ quá trình huấn luyện
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']

# Vẽ đồ thị độ chính xác qua các epoch
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(range(1, len(acc) + 1), acc, label='Training Accuracy')
plt.plot(range(1, len(val_acc) + 1), val_acc, label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

# Vẽ đồ thị độ mất mát qua các epoch
plt.subplot(1, 2, 2)
plt.plot(range(1, len(loss) + 1), loss, label='Training Loss')
plt.plot(range(1, len(val_loss) + 1), val_loss, label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

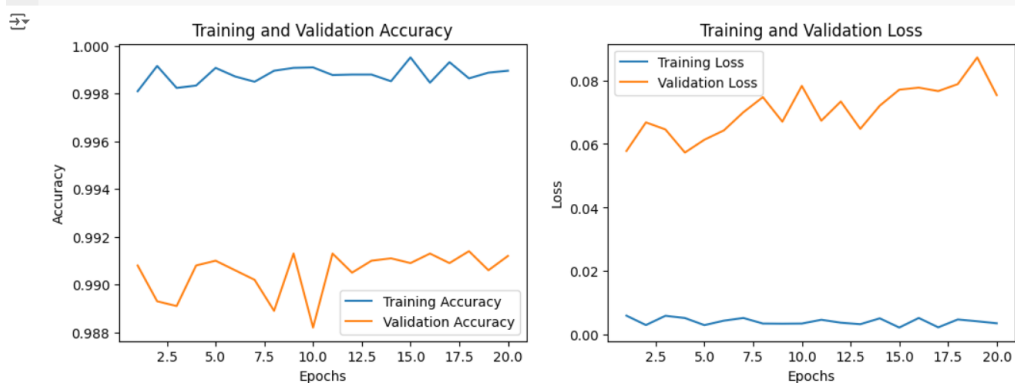
# Hiển thị đồ thị
plt.show()
```

Lưu model

```
[ ] # save a trained neural network model to a file
model.save('reco_digit.keras')
```

3.3.8 Biểu đồ đánh giá

```
# Hiển thị đồ thị
plt.show()
```



3.3.9 Thử nghiệm mô hình hình ảnh

Chọn ảnh cần nhận dạng

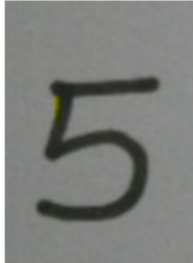
```
[41] from google.colab import files
      uploaded = files.upload()
      filename = next(iter(uploaded))

      # Hình ảnh được tải lên với kích thước ban đầu
      from IPython.display import Image
      Image(filename) #show image with its original size
```



Choose Files so5.jpg

- **so5.jpg**(image/jpeg) - 3178 bytes, last modified: 8/26/2024 - 100% done
Saving so5.jpg to so5.jpg



```
[42] # make a prediction for a new image.
      from keras.preprocessing.image import load_img
      from keras.preprocessing.image import img_to_array
      from keras.models import load_model

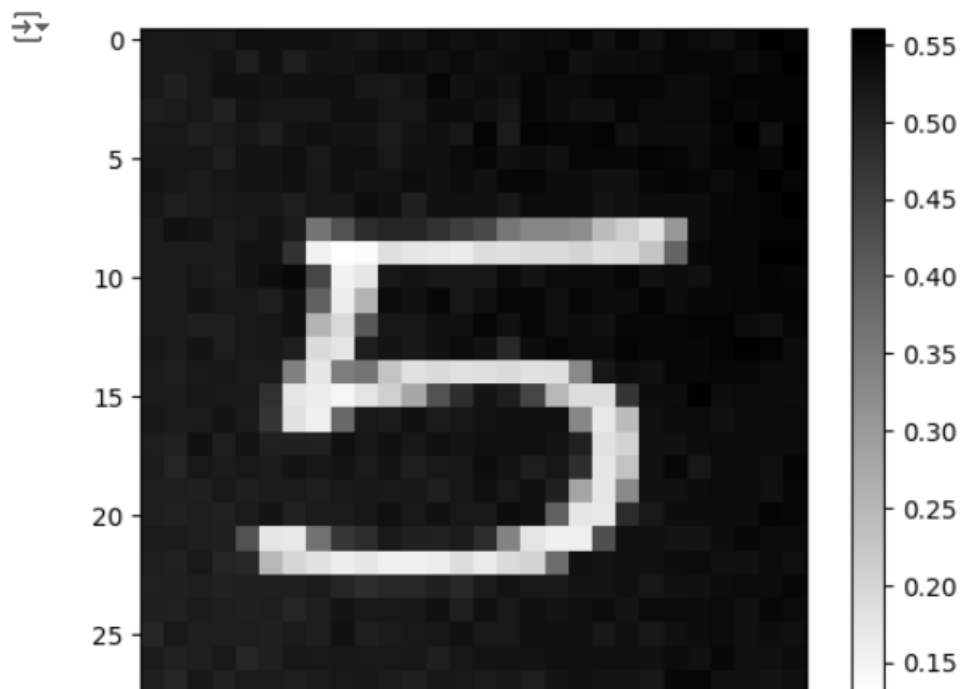
      # Tải hình ảnh
      img = load_img(filename, color_mode="grayscale", target_size=(28, 28,1))
      # Chuyển đổi thành mảng
      img = img_to_array(img)
      # reshape into a single sample with 1 channel
      img = img.reshape(28, 28, 1)
      # Chuẩn bị dữ liệu pixel cho hình ảnh
      img = img.astype('float32')
      img = img / 255.0
      # Chuyển đổi hình ảnh về dạng
      type(img)
      tf.shape(img)
      print(tf.shape(img))
```



tf.Tensor([28 28 1], shape=(3,), dtype=int32)

```
[43] # Take 1 image and reshaping
x = np.array([img])
image = x
image = image.reshape(28, 28)

# Plot the image - voila a piece of fashion clothing
plt.figure()
plt.imshow(image, cmap=plt.cm.binary)
plt.colorbar()
plt.grid(False)
plt.show()
```



```
[44] predict = model.predict(x)
label = np.argmax(predict,axis = 1)
print(f"Số nhận dạng được: {label}")
```

1/1 ————— 0s 25ms/step
Số nhận dạng được: [5]

Kết luận: Chương ba đã trình bày quá trình xây dựng hệ quyết định cho bài toán nhận diện chữ số viết tay sử dụng mô hình Convolutional Neural Network (CNN). Qua nghiên cứu và thực hiện, chúng em đã đạt được những kết quả đáng chú ý về hiệu suất của hệ thống. Mô hình CNN đã được chọn lựa và tinh chỉnh để đáp ứng đặc điểm riêng của bài toán nhận diện chữ số viết tay. Việc này bao gồm việc xử lý đối với độ phức tạp của các biểu diễn hình ảnh, đồng thời tối ưu hóa khả năng trích xuất đặc trưng từ vùng quan trọng của hình ảnh chữ số. Tóm lại, nội dung chương đã đặt nền tảng cho việc triển khai một hệ thống nhận diện chữ số hiệu quả dựa trên mô hình CNN. Các kết quả đạt được không chỉ là bước quan trọng trong nghiên cứu của chúng em mà còn mang lại những triển vọng tích cực cho ứng dụng thực tế trong lĩnh vực xử lý và nhận diện chữ viết.

TỔNG KẾT

Trong quá trình nghiên cứu và thực hiện đề tài về mạng nơ-ron tích chập (Convolutional Neural Network - CNN), chúng ta đã đi sâu vào hiểu và khám phá một trong những công cụ quan trọng nhất trong lĩnh vực trí tuệ nhân tạo và xử lý hình ảnh. CNN đã chứng minh sức mạnh của mình trong việc xử lý dữ liệu hình ảnh phức tạp. Thông qua các lớp tích chập và lớp tổng hợp, CNN có khả năng trích xuất và hiểu các đặc trưng quan trọng trong dữ liệu hình ảnh, từ cạnh, góc đến các hình dạng phức tạp.

Các ứng dụng của CNN rất đa dạng và ảnh hưởng đến nhiều lĩnh vực. Với khả năng phân loại ảnh và nhận dạng đối tượng, CNN đã đóng góp vào việc xây dựng các hệ thống tự động như xe tự hành và nhận dạng khuôn mặt. Ngoài ra, CNN cũng được sử dụng trong y học để phân tích hình ảnh y tế và hỗ trợ chẩn đoán.

Tuy nhiên, việc xây dựng và huấn luyện CNN cũng đòi hỏi sự tốn kém về tài nguyên tính toán và dữ liệu. Để đạt được kết quả tốt, chúng ta cần có một tập dữ liệu đủ lớn và đa dạng, cùng với quy trình huấn luyện và tinh chỉnh thích hợp.

Trên thực tế, việc phát triển CNN vẫn đang tiếp phát triển mạnh mẽ trong tương lai. Các nghiên cứu tiếp theo có thể tập trung vào việc tối ưu hóa cấu trúc CNN, kết hợp nó với các mô hình học máy khác và nghiên cứu các phương pháp mới để tăng cường hiệu suất và độ chính xác.

Tóm lại, đề tài này không chỉ mở ra cơ hội để khám phá và áp dụng mô hình CNN, mà còn thúc đẩy chúng ta suy ngẫm về tiềm năng của trí tuệ nhân tạo và học máy trong việc cải thiện trải nghiệm người dùng.

TÀI LIỆU THAM KHẢO

- [1]. Lancini, Stefano; Lazzari, Marco; Masera, Alberto; Salvaneschi, Paolo (1997). "Diagnosing Ancient Monuments with Expert Software" (PDF). *Structural Engineering International*. 7 (4): 288–291. doi:10.2749/101686697780494392.
- [2]. Lazzari, M.; Salvaneschi, P. (1999). "Embedding a Geographic Information System in a Decision Support System for Landslide Hazard Monitoring" (PDF). *Natural Hazards*. 20 (2–3): 185–195. doi:10.1023/A:1008187024768. S2CID 1746570.
- [3]. Haettenschwiler, P. (1999). *Neues anwenderfreundliches Konzept der Entscheidungsunterstützung. Gutes Entscheiden in Wirtschaft, Politik und Gesellschaft*. Zurich, vdf Hochschulverlag AG: 189-208.
- [4]. Lam Anh (2019). Hệ hỗ trợ quyết định (Decision Support System - DSS) là gì?.