**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

BÀI TẬP LỚN

HỆ HỖ TRỢ QUYẾT ĐỊNH

**Đề tài : Deep Learing- Xây dựng hệ hỗ trợ phân loại chữ số sử dụng mạng CNN**

|  |  |
| --- | --- |
| **GVHD:** | **Ths. Trần Thanh Hùng** |
| **Nhóm:** | **14** |
| **Lớp:** | **20221IT6058001** |
| **Thành viên:** | **Phạm Đức Minh - 2019603324    Đinh Nguyễn Tùng Anh - 2019605055    Trần Hải Đăng - 2019605387    Nguyễn Văn Nhuận - 2019605161** |

Mục Lục

[**Chương 1. Tổng quan** 5](#_Toc123144390)

[**1.1 Hệ hỗ trợ ra quyết định** 5](#_Toc123144391)

[1.1.1 Hệ hỗ trợ quyết định là gì? 5](#_Toc123144392)

[1.1.2 Tại sao cần xây dựng hệ hỗ trợ ra quyết định? 5](#_Toc123144393)

[1.1.3 Bản chất của hệ hỗ trợ quyết định 6](#_Toc123144394)

[1.1.4 Các yếu tố ảnh hưởng đến hệ hỗ trợ quyết định 6](#_Toc123144395)

[**1.2 Mô hình ra quyết định** 6](#_Toc123144396)

[1.2.1 Khái niệm Deep learning 6](#_Toc123144397)

[1.2.2 Mục đích xây dựng 8](#_Toc123144398)

[1.2.3 Tương lai của Deep Learning 8](#_Toc123144399)

[1.2.4 Một số ứng dụng của Deep Learning 9](#_Toc123144400)

[**1.3. Giới thiệu về mạng CNN** 10](#_Toc123144401)

[1.3.1. Khái niệm 10](#_Toc123144402)

[1.3.2. Các lớp cơ bản của CNN 11](#_Toc123144403)

[1.3.3. Các hàm kích hoạt thường gặp 14](#_Toc123144404)

[1.3.4. Cách chọn tham số cho CNN 15](#_Toc123144405)

[**Chương 2. Phân tích đề tài** 16](#_Toc123144406)

[**2.1 Tính cấp thiết của việc xây dựng đề tài** 16](#_Toc123144407)

[**2.2 Dữ liệu xây dựng đề tài** 17](#_Toc123144408)

[**2.3 Mô hình luồng bài toán** 18](#_Toc123144409)

[**Chương 3. Xây dựng chương trình** 19](#_Toc123144410)

[**3.1 Ngôn ngữ lập trình** 19](#_Toc123144411)

[**3.2 Môi trường huấn luyện** 19](#_Toc123144412)

[**3.3 Thuật toán sử dụng** 20](#_Toc123144413)

[**3.4 Xây dựng mô hình hệ hỗ trợ.** 20](#_Toc123144414)

[**Tổng kết** 24](#_Toc123144415)

[**TÀI LIỆU THAM KHẢO** 25](#_Toc123144416)

**Lời nói đầu**

Ngày nay, cùng với sự phát triển của khoa học kỹ thuật, chất lượng cuộc sống của con người ngày càng được nâng cao. Nền công nghiệp ví như là một "cánh tay phải" đắc lực, hỗ trợ con người trong nhiều mặt của đời sống: Liên lạc, giải trí, làm việc... Đi cùng với xu thế hội nhập đó, việc nhận dạng ngôn ngữ nói chung hay nhận diện chữ số nói riêng là một trong những vấn đề vô cùng cấp thiết và có thể ứng dụng rộng rãi trong đời sống thông qua các ví dụ điển hình như mô hình nhận dạng biển số xe, nhận dạng văn bản, nhận dạng chứng minh thư…vv. Nhận thấy sự bất cập đó, chúng em đã nhận đề tài làm về hệ thống nhận dạng chữ số, giúp cho hệ thống thông minh có thể nhận diện chữ số thông qua dữ liệu hình ảnh đầu vào bằng cách nhận dạng và chuyển chúng thành các mẫu số thông qua các nét cũng như kiểu viết đầu vào của chữ số.

Chúng em rất lấy lòng cảm ơn thầy Trần Thanh Hùng đã giúp đỡ, hỗ trợ nhiệt tình trong quá trình nhóm chúng em hoàn thành đề tài này. Dù vẫn còn nhiều thiếu sót, một số bất cập có thể sinh ra trong quá trình chạy, vì thế chúng em sẽ cố gắng cải thiện phần mềm này một cách tốt nhất trong tương lai.

*Chúng em xin chân thành cảm ơn Thầy!*

# Tổng quan

## Hệ hỗ trợ ra quyết định

### Hệ hỗ trợ quyết định là gì?

- Hệ hỗ trợ quyết định trong tiếng Anh gọi là Decision Support System, viết tắt là DSS.

- Hệ hỗ trợ quyết định (DSS) là một chương trình vi tính được sử dụng để hỗ trợ đưa ra các quyết định, phán đoán và chiều hướng hành động của một tổ chức hoặc một doanh nghiệp. DSS sẽ sàng lọc và phân tích lượng dữ liệu khổng lồ, tổng hợp thông tin một cách toàn diện mà có thể được sử dụng để giải quyết các vấn đề và trong quá trình ra quyết định.

- Thông tin thường được sử dụng bởi DSS gồm doanh thu mục tiêu, số liệu bán hàng từ các khoảng thời gian khác nhau và dữ liệu kiểm kê hoặc những hoạt động liên quan khác.

- Một DSS sẽ tập hợp và phân tích dữ liệu, tổng hợp nó để tạo ra các báo cáo thông tin tổng quát. Theo cách này, là một ứng dụng thông tin, DSS khác với những ứng dụng hoạt động thông thường chỉ có chức năng là thu thập dữ liệu.

DSS có thể được máy tính hóa hoàn toàn hoặc được điều khiển bởi con người. - Trong một số trường hợp, nó có thể kết hợp cả hai. Các hệ thống lí tưởng sẽ phân tích thông tin và thực sự đưa ra quyết định cho người dùng. Ít nhất chúng cho phép người dùng đưa ra quyết định sáng suốt hơn với tốc độ nhanh hơn.

### Tại sao cần xây dựng hệ hỗ trợ ra quyết định?

* 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.
* Nhu cầu hỗ trợ ra quyết định.
* Ra quyết định luôn cần xử lý kiến thức.
* Kiến thức là nguyên liệu và thành phẩm của ra quyết định, cần được sở hữu hoặc tích lũy bởi người ra quyết định.
* Giới hạn về nhận thức (trí nhớ có hạn ..)
* Giới hạn về kinh tế (chi phí nhân lực ..)
* Giới hạn về thời gian.

### Bản chất của hệ hỗ trợ quyết định

• Cung cấp thông tin, tri thức

• Có thể thể qua tương tác người – máy, qua mô phỏng

### Các yếu tố ảnh hưởng đến hệ hỗ trợ quyết định

• Công nghệ - thông tin - máy tính

• Tính cạnh tranh – phức tạp cấu trúc

• Các thay đổi biến động

## Mô hình ra quyết định

### Khái niệm Deep learning

- Khái niệm: Deep learning đã và đang là một chủ đề AI được bàn luận sôi nổi. Là một phạm trù nhỏ của machine learning, deep learning tập trung giải quyết các vấn đề liên quan đến mạng thần kinh nhân tạo nhằm nâng cấp các công nghệ như nhận diện giọng nói, tầm nhìn máy tính và xử lý ngôn ngữ tự nhiên.

- Deep learning đang trở thành một trong những lĩnh vực hot nhất trong khoa học máy tính, deep learning đã thúc đẩy tiến bộ trong đa dạng các lĩnh vực như nhận thức sự vật (object perception), dịch tự động (machine translation), nhận diện giọng nói,… - những vấn đề từng rất khó khăn với các nhà nghiên cứu trí tuệ nhân tạo.

- Để hiểu hơn về deep learning, hãy nhìn lại một số khái niệm cơ bản về trí tuệ nhân tạo: có thể được hiểu đơn giản là được cấu thành từ các lớp xếp chồng lên nhau, trong đó mạng thần kinh nhân tạo nằm ở dưới đáy, machine learning nằm ở tầng tiếp theo và deep learning nằm ở tầng trên cùng.

- Các ưu điểm của Deep Learning:

* Kiến trúc mạng nơ-ron linh hoạt, có thể dễ dàng thay đổi để phù hợp với nhiều vấn đề khác nhau.
* Có khả năng giải quyết nhiều bài toán phức tạp với độ chính xác rất cao.
* Tính tự động hóa cao, cố khả năng tự điều chỉnh và tự tối ưu.
* Có khả năng thực hiện tính toán song song, hiệu năng tốt, xử lý được lượng dữ liệu lớn.

-Nhược điểm của Deep Learning:

* Cần có khối lượng dữ liệu rất lớn để tận dụng tối đa khả năng của Deep Learning.
* Chi phí tính toán cao vì phải xử lý nhiều mô hình phức tạp.
* Chưa có nền tảng lý thuyết mạnh mẽ để lựa chọn các công cụ tối ưu cho Deep Learning.

### Mục đích xây dựng

- Deep learning là một trong những phương pháp của Machine learning, với mạng nơ-ron nhân tạo được xây dựng nhằm mục đích mô phỏng khả năng tư duy logic của bộ não con người.

- Mạng nơ-ron gồm có nhiều lớp (layer) khác nhau. Layer có số lượng càng nhiều thì mạng càng “deep” (sâu) hơn. Bên trong mỗi layer sẽ có các nút mạng (node), được liên kết với các lớp liền kề khác. Mỗi một kết nối giữa các node với nhau sẽ có một trọng số tương ứng, trọng số càng cao thì kết nối càng có ảnh hưởng lớn đối với mạng nơ-ron.

- Ngoài ra, mỗi nơ-ron sẽ có một hàm kích hoạt nhằm chuẩn hóa đầu ra. Khi người dùng đưa dữ liệu vào mạng nơ-ron, dữ liệu đó sẽ truyền qua các layer và trả về kết quả tại layer cuối cùng, gọi là output layer.

- Trong quá trình huấn luyện mô hình mạng nơ-ron, các trọng số sẽ được thay đổi. Khi đó, mô hình sẽ có trách nhiệm tìm ra bộ giá trị của trọng số để phán đoán là tốt nhất.

- Các hệ thống Deep learning sẽ cần phải có phần cứng mạnh để xử lý lượng lớn dữ liệu, đồng thời thực hiện các phép tính phức tạp. Đối với một số mô hình Deep learning, có thể sẽ mất vài tuần hoặc thậm chí vài tháng để có thể triển khai trên những phần cứng tiên tiến hiện nay.

### Tương lai của Deep Learning

- Deep learning đang ngày càng cho thấy một tương lai đầy hứa hẹn với ứng dụng vào điều khiển xe tự lái hay robot quản gia. Mặc dù các sản phẩm này vẫn còn nhiều hạn chế nhưng những thứ chúng làm được hiện nay thực sự rất khó tưởng tượng nổi chỉ vài năm trước đây; tốc độ nâng cấp cũng cao chưa từng thấy.

- Khả năng phân tích dữ liệu lớn và sử dụng deep learning vào các hệ thống máy tính có thể tự thích nghi với những gì chúng tiếp nhận mà không cần đến bàn tay lập trình của con người sẽ nhanh chóng mở đường cho nhiều đột phá trong tương lai.

- Những đột phá này có thể là việc thiết kế ra những trợ lý ảo, các hệ thống xe tự lái hay sử dụng vào thiết kế đồ họa, sáng tác nhạc, cho đến phát triển các nguyên liệu mới giúp robot thấu hiểu thế giới xung quanh hơn. Chính vì tính thương mại cao mà các công ty lớn, đặc biệt là Google, luôn ưu tiên các startup về robot và deep learning trong danh sách thâu tóm của mình.

### Một số ứng dụng của Deep Learning

- Ứng dụng trong ngành công nghệ cao:

+ Ứng dụng vượt bậc của Deep Learning không thể không kể đến đó chính là chế tạo Robot. Hiện nay, các phiên bản Robot giống con người với khả năng cảm nhận, phản ứng với môi trường đang dần được ra đời.

* Hiện nay, Robot còn có thể hợp tác với các hoạt động của con người và chúng có thể thực hiện được các nhiệm vụ riêng biệt phù hợp với thế mạnh của chúng. Robot đang góp phần thay thế con người trong việc thực hiện các công việc khó khăn hơn. Đây chính là phát minh tuyệt vời nhờ ứng dụng Deep Learning.

- Ứng dụng trong nông nghiệp:

* Hiện nay, nhờ có Deep Learning mà người nông dân có thể triển khai những thiết bị có khả năng phân biệt được cỏ dại với cây trồng. Từ đó, các loại máy móc phun thuốc diệt cỏ có thể chọn lọc phun lên cỏ dại để đảm bảo cho cây trồng không bị ảnh hưởng.
* Ngoài vai trò loại bỏ cỏ dại bằng thuốc diệt cỏ thì nhờ Deep Learning mà sản lượng nông nghiệp ngày càng được cải thiện. Bên cạnh đó, Deep Learning đang được mở rộng hơn nữa sang các hoạt động như: thu hoạch, tưới tiêu, bón phân, gieo trồng,...

- Ứng dụng trong điện tử:

* Deep Learning được ứng dụng cho công việc nhận diện không mặt trong các thiết bị điện tử như điện thoại, các phần mềm như facebook, messenger,... Thêm vào đó, Deep Learning còn được dùng trong nghe dịch và nói tự động mà trang web google dịch đang sử dụng. Ngoài ra, một số thiết bị trợ giúp tại nhà cũng được ứng dụng Deep Learning để đáp ứng giọng nói và hiểu được sở thích của bản (đây chính là ứng dụng học sâu).

## 1.3. Giới thiệu về mạng CNN

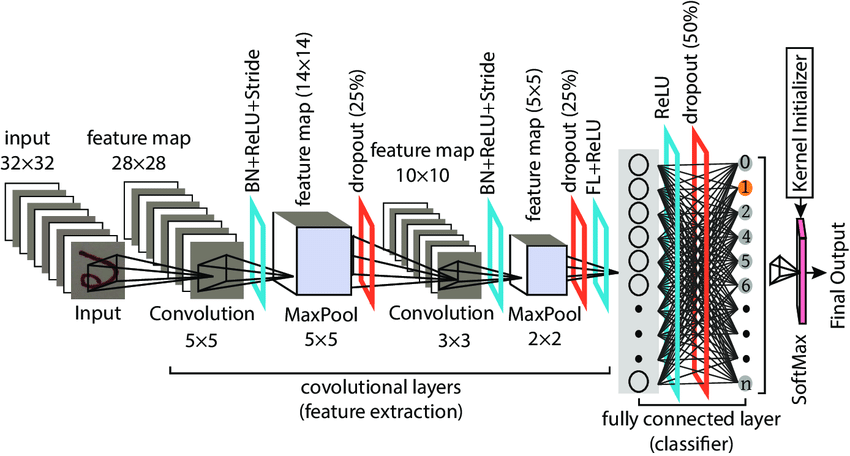
### 1.3.1. Khái niệm

- Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay.

- CNNs được chia thành 3 chiều: rộng, cao, sâu. Các Nơ-ron trong mạng không liên kết hoàn toàn với toàn bộ Nơ-ron kế đến mà chỉ liên kết tới một vùng nhỏ. Cuối cùng, một tầng đầu ra được tối giản thành vec-tơ của giá trị xác suất.

- CNNs gồm 2 thành phần:

* Phần tầng ấn hay phần rút trích đặc trưng: Trong phần này, mạng sẽ tiếng hành tính toán hàng loạt phép tích chập (Convolutional layer) và phép hợp nhất (pooling) để phát hiện các đặc trưng.
* Phần phân lớp: Tại phần này, một số lớp các liên kết đầy đủ (Fully Connected) sẽ đóng vai trò như một bộ phân lớp các đặc trưng đã rút trích trước đó.Tầng này sẽ đưa ra xác suất của một đối tượng trong hình.



### CS 230 - Mạng nơ ron tích chập cheatsheet1.3.2. Các lớp cơ bản của CNN

*Minh họa các tầng của CNN*

* **Convolutional Layer**

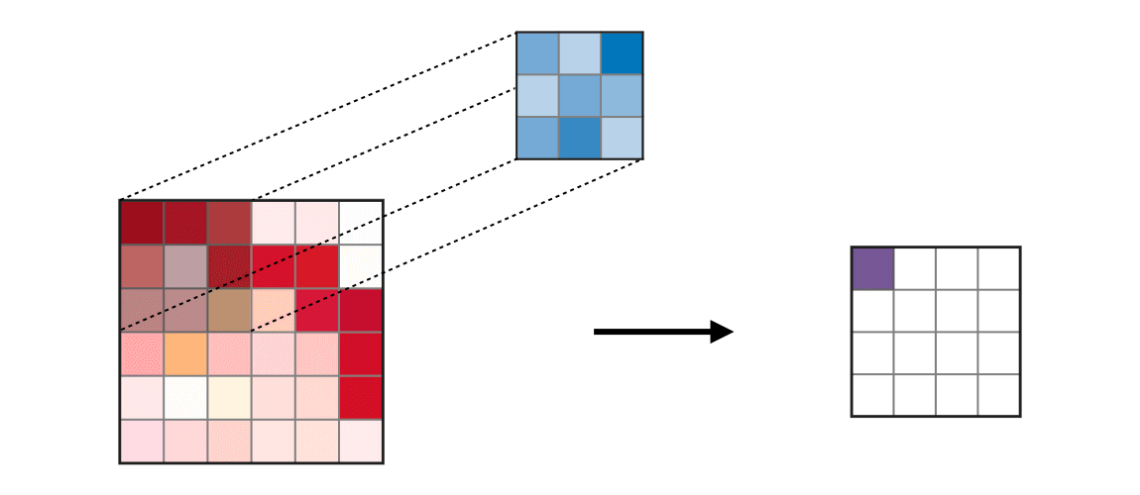
- Trong ba lớp của Convolutional Neural Network, Convolutional Layer được xem là lớp có vai trò quan trọng nhất. Bởi vì Convolutional Layer sẽ đại diện CNN thực hiện mọi phép toán.

+ **Filter Map**: CNN được sử dụng những Filter để áp vào các vùng của hình ảnh. Những Filter Map này có thể xem là một ma trận 3 chiều, bao gồm những con số và các con số chính là Parameter.

+ **Stride**: Trong CNN, Stride được hiểu là khi chúng ta dịch chuyển Filter Map theo Pixel và dựa vào giá trị từ trái sang phải. Stride đơn giản là biểu thị sự dịch chuyển này.

+ **Padding**: Chính là những giá trị 0 được thêm vào lớp Input.

+ **Feature Map**: Đây là kết quả hiển thị sau mỗi lần Filter Map quét qua Input. Cứ mỗi lần quét như vậy, bạn sẽ thấy sự xuất hiện của quá trình tính toán được xảy ra



* **Pooling Layer**

- Khi đầu vào quá lớn, các lớp Pooling Layer sẽ được dịch chuyển vào giữa những lớp Convolutional Layer nhằm giảm các Parameter.

- Pooling Layer được biết đến với hai loại phổ biến là: Max Pooling và Average Pooling.

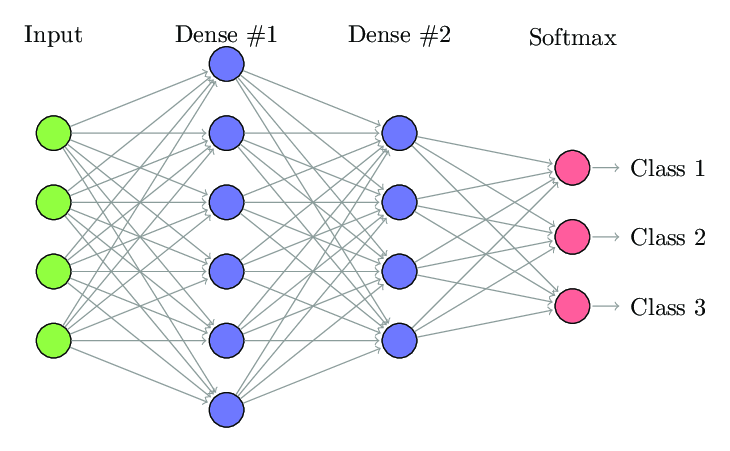
- Tại Pooling Layer, khi bạn sử dụng lớp Max Pooling thì số lượng Parameter có thể sẽ giảm đi. Vì vậy, Convolutional Neural Network sẽ xuất hiện nhiều lớp Filter Map, mỗi Filter Map đó sẽ cho ra một Max Pooling khác nhau.



*Pooling layer*

* **Fully Connected Layer**

- Tầng kết nối đầy đủ (FC) nhận đầu vào là các dữ liệu đã được làm phẳng, mà mỗi đầu vào đó được kết nối đến tất cả neuron. Trong mô hình mạng CNNs, các tầng kết nối đầy đủ thường được tìm thấy ở cuối mạng và được dùng để tối ưu hóa mục tiêu của mạng ví dụ như độ chính xác của lớp.

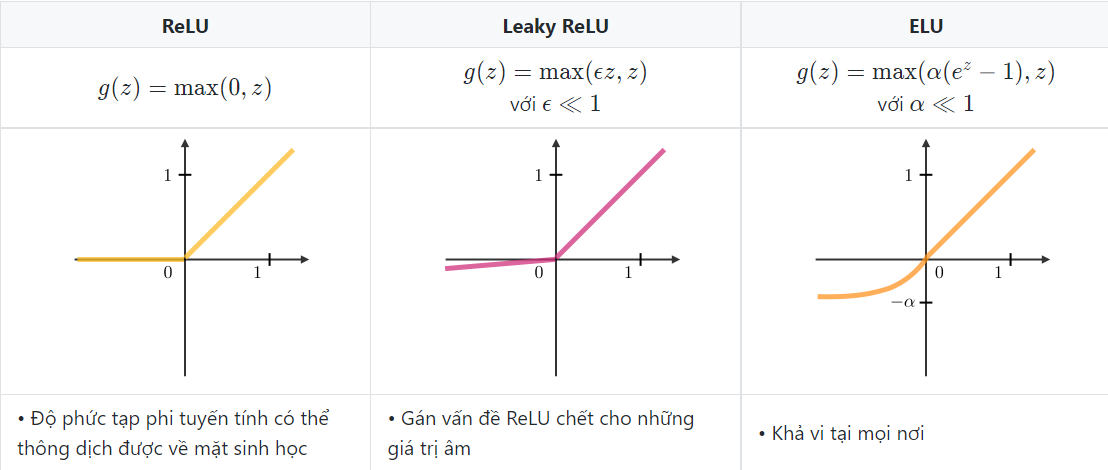


*Full Connected Layer*

### 1.3.3. Các hàm kích hoạt thường gặp

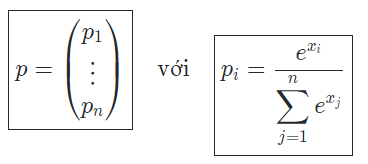
* **Rectified Linear Unit - ReLU**

Là một hàm kích hoạt *g* được sử dụng trên tất cả các thành phần. Mục đích của nó là tăng tính phi tuyến tính cho mạng. Những biến thể khác của ReLU được tổng hợp ở bảng dưới:



* **Softmax**

Bước softmax có thể được coi là một hàm logistic tổng quát lấy đầu vào là một vector chứa các giá trị x\in\mathbb{R}^n*x*∈R*n* và cho ra là một vector gồm các xác suất p\in\mathbb{R}^n*p*∈R*n* thông qua một hàm softmax ở cuối kiến trúc. Nó được định nghĩa như sau:



### 1.3.4. Cách chọn tham số cho CNN

- Số các convolution layer: càng nhiều các convolution layer thì performance càng được cải thiện. Sau khoảng 3 hoặc 4 layer, các tác động được giảm một cách đáng kể

- Filter size: thường filter theo size 5×5 hoặc 3×3

- Pooling size: thường là 2×2 hoặc 4×4 cho ảnh đầu vào lớn

- Cách cuối cùng là thực hiện nhiều lần việc train test để chọn ra được param tốt nhất.

# Phân tích đề tài

## Tính cấp thiết của việc xây dựng đề tài

- Hiện nay, vấn đề nhận dạng chữ số rất cần thiết, có nhiều ứng dụng rộng rãi trong đời sống xã hội như nhận dạng bảng điểm, nhận dạng bảng số xe, nhận dạng phiếu hàng hóa,… Vấn đề nhận dạng chữ số là một thách thức lớn đối với các nhà nghiên cứu. Mỗi người có một cách viết chữ số khác nhau, chúng ta không thể xác định cách duy nhất để nhận dạng chữ số. Do vậy, xây dựng hệ thống nhận dạng chữ số một cách đáng tin cậy để có có thể nhận dạng bất cứ ký tự số nào là điều không dễ dàng.

- Nhận dạng chữ số được thực hiện qua hai hình thức đó là nhận dạng online và nhận dạng offline. Nhận dạng online có nghĩa là máy tính sẽ nhận dạng các chữ được viết lên màn hình ngay khi nó được viết. Đối với những hệ nhận dạng chữ số, máy tính sẽ lưu lại các thông tin về nét chữ như thứ tự nét viết, hướng và tốc độ của nét viết trong khi nó đang được viết. Còn nhận dạng offline tức là việc nhận dạng được thực hiện sau khi chữ số đã được viết hay in lên giấy rồi, lúc đó thông tin đầu vào là hình ảnh văn bản hoặc ký tự cần nhận dạng.

- Đối với bài toán nhóm sử dụng phương pháp nhận dạng offline với bộ dữ liệu test có sãn của DeepLearning là MNIST nhằm traning dữ liệu hiểu quả cũng như cho ra kết quả ít sai sót nhất.

* **Những khó khăn trong việc nhận dạng chữ số:**

- Chữ số có nhiều cách viết, kiểu viết khác nhau như viết bằng công cụ máy tính, viết bằng trình soạn thảo văn bản, viết bằng tay…vv. Điều này vô hình chung tạo ra vô vàn kiểu font cũng như định dạng chữ số khác nhau.

- Chữ số viết tay được viết bởi những người khác nhau, ở những trạng thái khác nhau nên đôi khi một số chữ số viết bị nhòe hoặc mờ, bề mặt ký tự có thể bị mờ hoặc che khuất bởi một đối tượng khác, dẫn đến đọc ký tự sai.

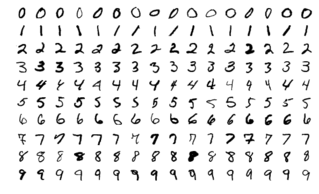
- Trong một số trường hợp, người viết có thể viết các chữ số dính liền nhau dẫn đến việc phân tích ký tự chưa được chính xác, dẫn đến đọc ký tự sai.

- Một số chữ số viết nằm giao nhau, cắt nhau hoặc chữ số này chứa trong chữ số kia gây khó khăn trong việc nhận dạng, dẫn đến đọc ký tự sai.

- Giới hạn về cơ sở dữ liệu ảnh: các chữ số 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

## Dữ liệu xây dựng đề tài

- Hệ thống ban đầu sẽ được training theo bộ dữ liệu đầu **MNIST** là bộ dữ liệu là các con số viết tay từ 0 đến 9. Bộ dữ liệu này bao gồm 60.000 mẫu cho huấn luyện và 10.000 mẫu để kiểm thử. Các mẫu dữ liệu trong bộ MNIST đã được chuẩn hóa về kích thước: căn chỉnh chính giữa dữ liệu, và mỗi mẫu sẽ là một ảnh có kích thước (shape) 28×28 pixel có nhãn là giá trị số của nó (0 đến 9).



*Bộ dữ liệu MNIST*

## Mô hình luồng bài toán

- Nhận dạng chữ số sẽ bao gồm luồng dữ liệu được đi qua 5 giai đoạn theo trình tự tuyến tính của mô hình thác nước:

* Tiền xử lý (preprocessing): giảm nhiễu cho các lỗi trong quá trình quét ảnh định dạng các chữ số, chuẩn hóa dữ liệu và nén dữ liệu.
* Tách chữ số (segmentation): chia nhỏ hình ảnh chữ số thành những thành phần nhỏ hơn, tách các nét của chữ số ra thành nhiều phần.
* Trích chọn đặc trưng (representation): giai đoạn đóng vai trò quan trọng nhất trong nhận dạng chữ viết số. Để tránh những phức tạp của chữ viết số cũng như tăng cường độ chính xác, ta cần phải biểu diễn thông tin chữ số dưới những dạng đặc biệt hơn và cô đọng hơn, rút trích các đặc điểm riêng nhằm phân biệt các ký tự khác nhau.
* Huấn luyện và nhận dạng (training and recognition): huấn luyện dữ liệu bằng bộ dữ liệu đầu vào là folder ảnh gồm nhiều chữ số theo nhiều cách viết cũng như là định dạng, kích thước khác nhau, hệ hỗ trợ sẽ học máy nhằm tạo hệ nhận dạng kiểu dữ liệu, giai đoạn này vô cùng quan trọng vì sẽ tạo bộ nhận dạng cho hệ hỗ trợ giúp kết quả đạt được như yêu cầu.
* Hậu xử lý (postprocessing): Dữ liệu sau khi được huấn luyện sẽ được lưu cũng như truyền tải lên màn hình hiển thị khi có yêu cầu truy xuất và sử lý, nhận dạng dữ liệu nhằm truy xuất thông tin nhanh chóng cũng như sử dụng vào mục đích làm dữ liệu nhận dạng đầu vào cho chu trình tiếp theo.

# Xây dựng chương trình

## Ngôn ngữ lập trình

- Hệ hỗ trợ của nhóm sử dụng ngôn ngữ lập trình Python do ngôn ngữ lập trình này có lối truy xuất cũng như đọc dữ liệu mang khuynh hướng toán học phù hợp với AI và Machine Learning kèm theo đó Python cũng có nhiều Framework cũng như Thư viện nhằm mở rộng giúp đơn giản hóa việc thực hiện các chức năng khác nhau của hệ thống.

- Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991.

- Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ.

- Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển trí tuệ nhân tạo.

- Python có một kho công nghệ phong phú bao gồm rất nhiều thư viện cho trí tuệ nhân tạo và học máy. Dưới đây là một số thư viện, framework phổ biến:

* Keras, TensorFlow, and Scikit-learn cho Học máy
* NumPy cho phân tích dữ liệu và tính toán khoa học hiệu năng cao
* SciPy cho advanced computing
* Pandas để phân tích dữ liệu mục đích chung
* Seaborn cho trực quan hóa dữ liệu (Data Visualization)

## Môi trường huấn luyện

- Nhóm sử dụng môi trường huấn luyện bằng bộ công cụ lập trình tiện ích của Google: Google Colaboratory.

- Colaboratory hay còn gọi là Google Colab, là một sản phẩm từ Google Research, nó cho phép chạy các dòng code python thông qua trình duyệt, đặc biệt phù hợp với Data analysis, machine learning và giáo dục. Colab không cần yêu cầu cài đặt hay cấu hình máy tính, mọi thứ có thể chạy thông qua trình duyệt, bạn có thể sử dụng tài nguyên máy tính từ CPU tốc độ cao và cả GPUs và cả TPUs đều được cung cấp cho bạn.

- Colab cung cấp nhiều loại GPU, thường là Nvidia K80s, T4s, P4s and P100s, tuy nhiên người dùng không thể chọn loại GPU trong Colab, GPU trong Colab thay đổi theo thời gian. Vì là dịch vụ miễn phí, nên Colab sẽ có những thứ tự ưu tiên trong việc sử dụng tài nguyên hệ thống, cũng như giới hạn thời gian sử dụng, thời gian sử dụng tối đa lên tới 12 giờ.

## Thuật toán sử dụng

- Nhóm sử dụng mạng CNN là kiến trúc lý tưởng khi giải quyết vấn đề dữ liệu hình ảnh, một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh phù hợp với đề tài nhận dạng chữ số mà nhóm triển khai.

- CNN có khả năng ghi lại sự phụ thuộc không gian của hình ảnh kể từ khi nó xử lý chúng dưới dạng ma trận và phân tích toàn bộ các phần của một hình ảnh tại một thời điểm, tùy thuộc vào kích thước của bộ lọc. Mỗi phần của hình ảnh được cung cấp một tập hợp các tham số (chiều rộng và độ lệch) sẽ tham chiếu mức độ liên quan của tập hợp pixel đó với toàn bộ hình ảnh, tùy thuộc vào bộ lọc. Theo điều này, bằng cách giảm số lượng các tham số và bằng cách phân tích hình ảnh theo từng phần, CNN có thể hiển thị đại diện tốt hơn của hình ảnh.

## Xây dựng mô hình hệ hỗ trợ.

* 1. Nhập những thư viện cần thiết

import tensorflow as tf

import numpy as np

from keras.models import Sequential

from keras.layers import Dense, Dropout, Activation, Flatten

from keras.layers import Conv2D, MaxPooling2D

from keras.utils import np\_utils

from keras.datasets import mnist

* 1. Đọc dữ liệu

mnist = tf.keras.datasets.mnist

(training\_images, training\_labels), (test\_images, test\_labels) = mnist.load\_data()

* 1. Tạo model và thêm các lớp tích chập

# Khởi tạo model

model = Sequential()

# Thêm một lớp tích chập + lớp Activation: relu

model.add(Conv2D(32,(3,3), activation='relu', input\_shape=(28,28,1)))

#model.add(Conv2D(32,(3,3), input\_shape=(28,28,1)))

#model.add(Activation('relu'))

# Thêm một lớp Max Pooling

model.add(MaxPooling2D(2,2))

# Thêm một lớp duỗi

model.add(Flatten())

# Kiểm tra model

model.summary()

#Thêm một lớp Dense 128 + lớp Activation: relu

model.add(Dense(128, activation='relu'))

# Thêm một lớp Dense 10 + lớp Activation: Softmax

model.add(Dense(10, activation="softmax"))

* 1. Xử lý và chuẩn hóa dữ liệu

# Xử lý dữ liệu ảnh

training\_images.shape

# Chuẩn hóa data đưa vào các giá trị từ 0-1

training\_images = training\_images / 255.0

# Đưa vào data 4 chiều(số lượng ảnh, kích cỡ chiều dài, kích cỡ chiều rộng, số lớp của 1 ảnh )

training\_images = training\_images.reshape(60000,28,28,1) #Ảnh xám nên chỉ có 1 lớp

#Chuẩn hóa tập Validation

#Kiểm tra chiều đầu tiên

test\_images.shape

# Chuẩn hóa hình dạng ảnh

test\_images = test\_images / 255.0

# Rest lại

test\_images = test\_images.reshape(10000, 28, 28, 1)

* 1. Load và Reshape lại dữ liệu theo kích thước hàm keras

#Load dữ liệu MNIST

(X\_train, y\_train), (X\_test, y\_test) = mnist.load\_data()

X\_val, y\_val = X\_train[50000:60000,:], y\_train[50000:60000]

X\_train, y\_train = X\_train[:50000,:], y\_train[:50000]

print(X\_train.shape)

# Reshape lại dữ liệu cho đúng kích thước mà keras yêu cầu

X\_train = X\_train.reshape(X\_train.shape[0], 28, 28, 1)

X\_val = X\_val.reshape (X\_val.shape[0], 28, 28, 1)

X\_test = X\_test.reshape(X\_test.shape[0], 28, 28, 1)

#One hot encoding label (Y)

Y\_train = np\_utils.to\_categorical(y\_train, 10)

Y\_val = np\_utils.to\_categorical (y\_val, 10)

Y\_test = np\_utils.to\_categorical (y\_test, 10)

* 1. Trainning dữ liệu

# Thêm một hàm tối ưu cho model

model.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics='acc') #Đây là hàm loss có thể làm việc luôn vs nhãn chính là 1 số

# Tiến hành training dữ liệu

model.fit(training\_images, training\_labels, epochs=10, validation\_data=(test\_images, test\_labels))

* 1. Xây dựng bộ mô hình huấn luyện

model.add(Conv2D(32, (3, 3), activation='sigmoid', input\_shape=(28,28,1)))

# Thêm Convolutional layer

model.add(Conv2D(32, (3, 3), activation='sigmoid'))

# Thêm Max pooling layer

model.add(MaxPooling2D(pool\_size=(2, 2)))

# Flatten layer chuyển từ tensor sang vector

model.add(Flatten())

#Thêm Fully Connected layer với 128 nodes và dùng hàm sigmoid

model.add(Dense(128, activation='sigmoid'))

#Output layer với 10 node và dùng softmax function để chuyển sang xác suất

model.add(Dense(10, activation='softmax'))

#Compile model, chỉ rõ hàm loss\_function nào được sử dụng, phương thức

# đùng để tối ưu hàm loss function.

model.compile(loss='categorical\_crossentropy',optimizer='adam',metrics=['accuracy'])

* 1. Dự đoán trên bộ test

#Dự đoán ảnh

plt.imshow(X\_test[155].reshape(28,28), cmap='gray')

y\_predict = model.predict(X\_test[155].reshape(1,28,28,1))

print('Giá trị dự đoán: Số ',np.argmax(y\_predict))

### Tổng kết

Trong quá trình tìm hiểu và nghiên cứu về đề tài: “**Deep Learing- Xây dựng hệ hỗ trợ phân loại chữ số sử dụng mạng CNN**”, chúng em đã học hỏi được rất nhiều kiến thức chuyên môn cũng như kỹ năng bổ ích giúp nâng cao trình độ và phát triển bản thân. Sau khi hoàn thành đề tài chúng em đã thu được một số kiến thức: Nắm bắt kiến thức cơ bản về các kỹ thuật phân tích và những yếu tố cần thiết để xây dựng một hệ thống hiệu quả chuẩn xác. Biết được một số kỹ thuật nhận diện khác nhau. Ứng dụng mạng CNN vào trong phân tích hình ảnh. Tuy nhiên trong quá trình nghiên cứu, tìm hiểu và thực hiện nhóm chúng em vẫn còn một vài vấn đề cần cải thiện như: Việc nhận dạng còn dựa trên chất lượng ảnh cũng như ánh sáng của ảnh đầu vào. Điều đó cho thấy mô hình huấn luyện xuất hiện hiện tượng overfitting và cần được cải thiện trong tương lai để đạt hiệu quả cao hơn.

Trong thời gian tới, chúng em sẽ cố gắng tiếp tục tìm hiểu nhiều hơn về các công nghệ kỹ thuật nhận diện và phân tích để tiến tới hoàn thiện hệ thống. Qua đó ứng dụng có thể được phát triển và áp dụng trong môi trường làm việc thực tế và giảm bớt được các công việc trong cuộc sống hiện nay ở nước ta.

# TÀI LIỆU THAM KHẢO

Website https://www.miai.vn/

Website <https://viblo.asia/>

<https://www.kaggle.com/uciml/mushroom-classification>

<https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/>