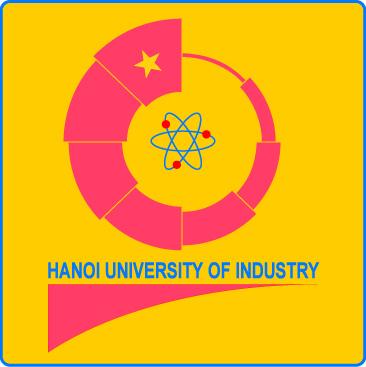
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-----------------------------------------**

**BÁO CÁO BÀI TẬP LỚN**

Học phần: Trí Tuệ Nhân Tạo

**ĐỀ TÀI:**

**Phân Loại Biển Báo Giao Thông Sử Dụng Mạng CNN**

|  |  |
| --- | --- |
| **Giảng Viên Hướng Dẫn** | **TS. Trần Hùng Cường** |
| **Nhóm** | **5** |
| **Sinh Viên** | **Trần Thu Hằng**  **Nguyễn Văn Hoàng**  **Nguyễn Văn Trọng Khá**  **Nguyễn Ngọc Khánh**  **Bùi Văn Mạnh** |
| **Lớp: KHMT2** | **Khóa: 14** |

**Hà Nội. 5/2022**

**MỤC LỤC**

Contents

[**LỜI MỞ ĐẦU** 5](#_Toc105800979)

[**CHƯƠNG 1: TỔNG QUAN** 6](#_Toc105800980)

[**1.1.** **Trí tuệ nhân tạo là gì** 6](#_Toc105800981)

[**1.2.** **Artificial Neural Network – ANN** 6](#_Toc105800982)

[1.2.1. Khái niệm 6](#_Toc105800983)

[1.2.2. Đặc điểm 7](#_Toc105800984)

[1.2.3. Các hàm kích hoạt của mạng ANN 8](#_Toc105800985)

[**1.3.** **Convolutional Neural Network - CNN** 13](#_Toc105800986)

[1.3.1. Convolutional layer 13](#_Toc105800987)

[1.3.2. ReLU layer 14](#_Toc105800988)

[1.3.3. Pooling layer 15](#_Toc105800989)

[1.3.4. Fully connected layer 15](#_Toc105800990)

[**CHƯƠNG 2: XÂY DỰNG VÀ HUẤN LUYỆN MÔ HÌNH** 17](#_Toc105800991)

[**2.1. Tổng quan chương trình** 17](#_Toc105800992)

[2.1.1. Công cụ sử dụng 17](#_Toc105800993)

[2.1.2. Ngôn ngữ lập trình 17](#_Toc105800994)

[2.1.3. Các thư viện và kỹ thuật sử dụng trong chương trình 17](#_Toc105800995)

[**2.2. Chương trình** 18](#_Toc105800996)

[2.2.1 Bộ dữ liệu(dataset) 18](#_Toc105800997)

[2.2.2. Cấu trúc mạng CNN cho mô hình 19](#_Toc105800998)

[2.2.3. Xây dựng chương trình 20](#_Toc105800999)

[**CHƯƠNG 3: KẾT QUẢ** 23](#_Toc105801000)

[**CHƯƠNG 4: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM** 24](#_Toc105801001)

[**4.1. Kết luận** 24](#_Toc105801002)

[4.1.1. Tính khả thi 24](#_Toc105801003)

[4.1.2. Thuận lợi 24](#_Toc105801004)

[4.1.3. Khó khăn 24](#_Toc105801005)

[**4.2. Bài học kinh nghiệm** 24](#_Toc105801006)

[4.2.1. Kiến thức 24](#_Toc105801007)

[4.2.2. Kỹ năng 24](#_Toc105801008)

[**TÀI LIỆU THAM KHẢO** 25](#_Toc105801009)

**DANH MỤC BẢNG BIỂU**

**DANH MỤC HÌNH ẢNH**

[Hình 1. 1: Mô tả cấu trúc của một neral sinh học 6](#_Toc105014573)

[Hình 1. 2: Cấu trúc mạng ANN 7](#_Toc105014574)

[Hình 1. 3: Đồ thị Sigmoid 8](#_Toc105014575)

[Hình 1. 4: Đồ thị hàm Tanh 9](#_Toc105014576)

[Hình 1. 5: Đồ thị hàm ReLU 10](#_Toc105014577)

[Hình 1. 6: Đồ thị hàm Leaky ReLU 11](file:///D:\HK6\TTNT\Báo%20Cáo%20NMLTND%20Main%20Main.docx#_Toc105014578)

[Hình 1. 7: Cấu trúc mạng CNN 12](#_Toc105014579)

[Hình 1. 8: Lớp tích chập 13](#_Toc105014580)

[Hình 1. 9: Hàm kích hoạt ReLU 14](#_Toc105014581)

[Hình 1. 10: Lớp pooling 14](#_Toc105014582)

[Hình 2. 1: Cấu trúc bộ dữ liệu 18](#_Toc105801010)

[Hình 2. 2: Các loại biển báo có trong bộ dữ liệu 18](#_Toc105801011)

[Hình 2. 3: Các ảnh trong thư mục Test 19](#_Toc105801012)

[Hình 2. 4: Cấu trúc mạng CNN của mô hình 19](#_Toc105801013)

[Hình 2. 5: Tạo thư mục và download dữ liệu 20](#_Toc105801014)

[Hình 2. 6: Load và phân chia dữ liệu 20](#_Toc105801015)

[Hình 2. 7: Convert nhãn sang ma trận nhị phân 21](#_Toc105801016)

[Hình 2. 8: Xây dựng mô hình sử dụng Sequential 21](#_Toc105801017)

[Hình 2. 9: Huấn luyện mô hình 21](#_Toc105801018)

[Hình 2. 10: Kết quả huấn luyện mô hình 22](#_Toc105801019)

[Hình 2. 11: Lưu trữ mô hình 22](#_Toc105801020)

# **LỜI MỞ ĐẦU**

Việt Nam là một đất nước có hệ thống chính trị ổn định bậc nhất trong khu vực và trên thế giới, tình trạng khủng bố gần như là không xảy ra. Thế nhưng tình trạng thiệt mạng do tai nạn giao thông thì lại rất cao, với hàng chục nghìn vụ tai nạn xảy ra hằng năm và hàng nghìn người thương vong, bên cạnh thiệt hại về người là thiệt hại về kinh tế ước tính lên đến tỷ USD. Qua thống kê về lượng người chết và thiệt hại về kinh tế, ta thấy sự mất mát rất lớn và rất nhiều người đã là nạn nhân trực tiếp hoặc là hệ lụy, ảnh hưởng của tai nạn giao thông. Từ tình hình thực tế giao thông tại Việt Nam, có nhiều nguyên nhân gây ra các vụ tai nạn giao thông đường bộ nhưng phần lớn là do tài xế không làm chủ tốc độ, thiếu quan sát hoặc không kịp nhận ra các loại biển báo và tín hiệu giao thông. Những mối nguy hiểm này đã được cảnh báo trước bởi hệ thống biển báo giao thông nhưng tai nạn vẫn thường xuyên xảy ra, gây không ít thiệt hại về tính mạng, tài sản của người tham gia giao thông và đặc biệt lại để lại hậu quả nặng nề cho xã hội.

Mặc dù công nghệ đã được áp dụng rất nhiều trong các lĩnh vực của đời sống, tuy nhiên những công nghệ hay nghiên cứu này chưa thực sự hiệu quả đối với giao thông đường bộ tại Việt Nam, trong khi đó tình hình nghiên cứu biển báo giao thông tại Việt Nam vẫn còn nhiều hạn chế và chưa đầy đủ.

Nhằm nghiên cứu về các nền tảng công nghệ, lý thuyết cơ bản trong bài toán xử lý ảnh, học máy và sử dụng vào việc phân loại biển báo giao thông nhằm áp dụng xây dựng chương trình nhắc nhở tài xế, nhóm chúng em thực hiện đề tài *“Phân loại biển báo giao thông sử dụng mạng CNN”*, trong quá trình thực hiện đề tài sẽ không thể tránh khỏi những thiếu sót, vì vậy chúng em mong nhận được đóng góp của thầy và các bạn để có thể hoàn thiện đề tài hơn.

Chúng em xin chân thành cảm ơn!

*Nhóm sinh viên thực hiện*

# **CHƯƠNG 1: TỔNG QUAN**

## **Trí tuệ nhân tạo là gì**

Trí tuệ nhân tạo (Artificial Intelligence) là một ngành thuộc lĩnh vực khoa học máy tính do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người.

Có 2 loại:

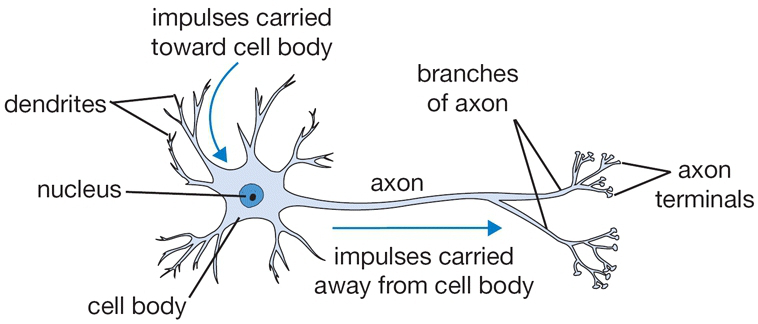
* Strong AI: Có thể tạo ra thiết bị có trí thông minh và các chương trình máy tính thông minh hơn người.
* Weak AI: Chương trình máy tính có thể mô phỏng các hành vi thông minh của con người.

|  |  |
| --- | --- |
| **Lập trình hệ thống** | **Lập trình AI** |
| Dữ liệu + Thuật toán = Chương trình | Tri thức + Điểu khiển = Chương trình |
| Xử lý dữ liệu | Xử lý dữ liệu định tính |
| Dữ liệu trong bộ nhớ được đánh địa chỉ số | Xử lý dựa trên tri thức cho phép dùng các thuật giải Heuristic, các cơ chế suy diễn |
| Xử lý theo các thuật toán | Tri thức được cấu trúc hóa, để trong bộ nhớ và làm việc theo ký hiệu |
| Định hướng xử lý theo các đại lượng, định lượng số | Định hướng xử lý các đại lượng định tính, các ký hiệu tượng trưng và danh sách |
| Xử lý tuần tự theo mẻ | Xử lý theo chế độ tương tác |
| Không giải thích trong quá trình thực hiện | Có giải thích hành vi của hệ thống trong quá trình thực hiện |
| Kết quả chính xác, không được mắc lỗi | Kết quả tốt, cho phép mắc lỗi |

## **Artificial Neural Network – ANN**

### Khái niệm

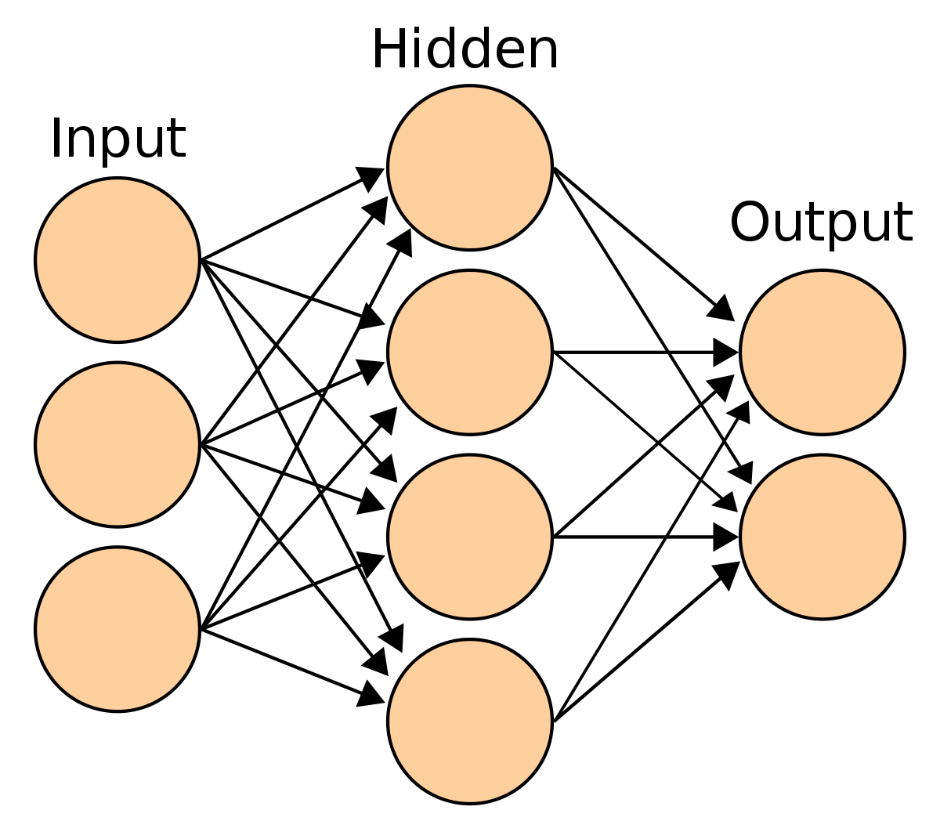
* Mạng nơron nhân tạo là mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thống thần kinh của sinh vật, bao gồm số lượng lớn các nơron được gắn kết để xử lý thông tin. Mạng nơron nhân tạo giống như bộ não con người, được học bởi kinh nghiệm (thông qua huấn luyện), có khả năng lưu giữ những kinh nghiệm hiểu biết (tri thức) và sử dụng những tri thức đó trong việc dự đoán các dữ liệu chưa biết (unseen data).



Hình 1. 1: Mô tả cấu trúc của một neral sinh học

### Đặc điểm

* Mạng nơron nhân tạo có thể hoạt động như mạng nơ ron của con người. Mỗi một nơ ron thần kinh trong nơ ron nhân tạo là hàm toán học với chức năng thu thập và phân loại các thông tin dựa theo cấu trúc cụ thể.
* Mạng Neural Network là sự kết hợp của những tập perceptron hay còn gọi là perceptron đa tầng. Và mỗi một mạng Neural Network thường bao gồm 3 kiểu tầng là:
* *Tầng input layer (tầng vào)*: Tầng này nằm bên trái cùng của mạng, thể hiện cho các đầu vào của mạng.
* *Tầng output layer (tầng ra)*: Là tầng bên phải cùng và nó thể hiện cho những đầu ra của mạng.
* *Tầng hidden layer (tầng ẩn)*: Tầng này nằm giữa tầng vào và tầng ra nó thể hiện cho quá trình suy luận logic của mạng



*Hình 1. SEQ Hình\_1. \\* ARABIC 2: Kiến trúc của mạng nơ ron nhân tạo*

Hình 1. 2: Cấu trúc mạng ANN

* Với mạng Neural Network thì mỗi nút mạng là một sigmoid nơron nhưng chúng lại có hàm kích hoạt khác nhau. Thực tế, người ta thường sử dụng có cùng loại với nhau để việc tính toán thuận lợi hơn.
* Tại mỗi tầng, số lượng nút mạng có thể khác nhau còn tùy vào bài toán hoặc cách giải quyết.
* Tuy nhiên, khi làm việc người ta sẽ để các tầng ẩn số với số lượng nơron khác nhau. Ngoài ra, những nơron nằm ở tầng thường sẽ liên kết đôi với nhau để tạo thành mạng kết nối đầy đủ nhất. Khi đó, người dùng có thể tính toán được kích cỡ của mạng dựa vào tầng và số lượng nơron.

### Các hàm kích hoạt của mạng ANN

* Hàm kích hoạt (activation function) mô phỏng tỷ lệ truyền xung qua axon của một neuron thần kinh. Trong một mạng nơ-ron nhân tạo, hàm kích hoạt đóng vai trò là thành phần phi tuyến tại output của các nơ-ron.

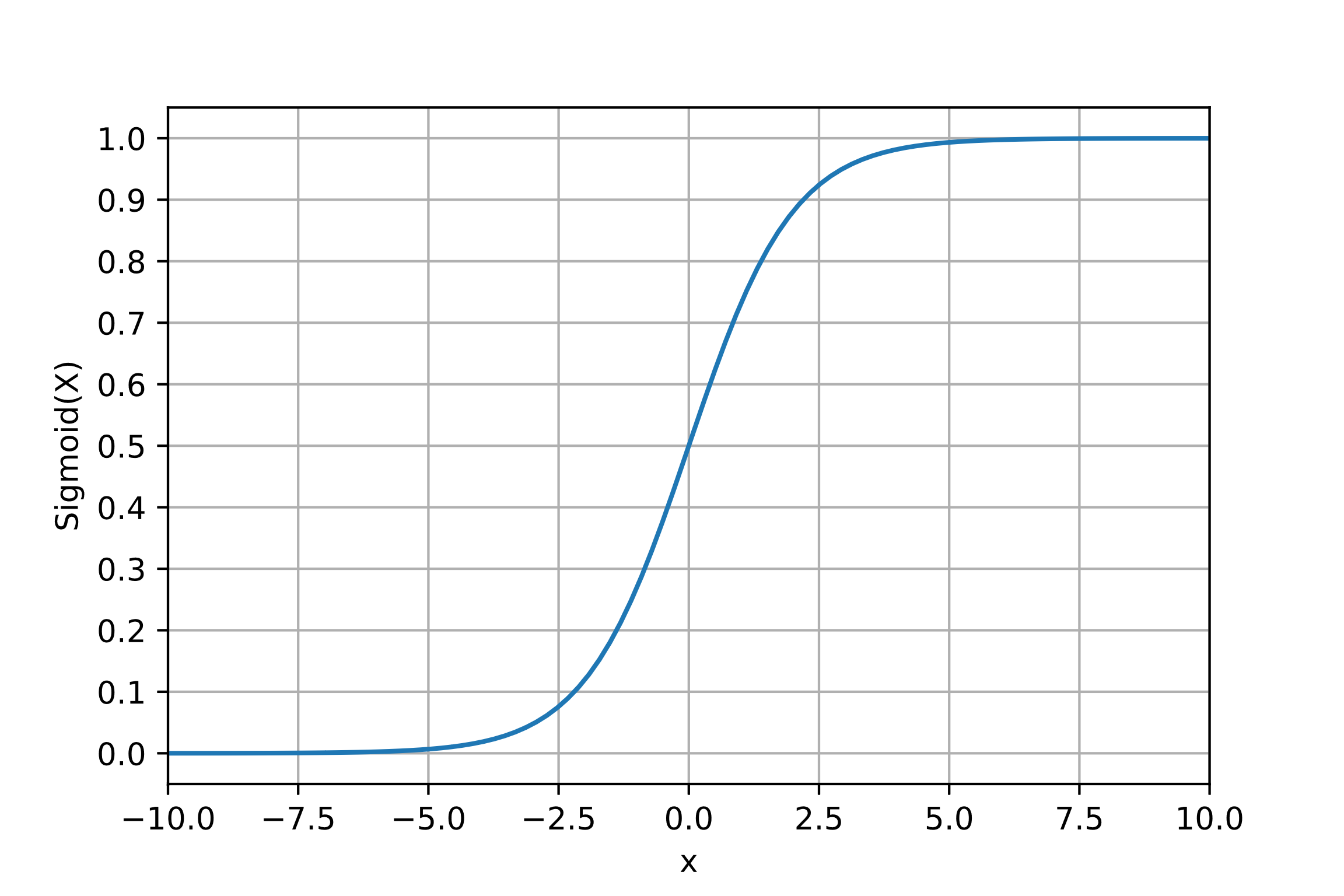
1. **Hàm Sigmoid**

* *Công thức:*

A picture containing text, clock

Description automatically generated

* *Phân tích:*



Hình 1. 3: Đồ thị Sigmoid

* Hàm Sigmoid nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (0;1). Đầu vào là số thực âm rất nhỏ sẽ cho đầu ra tiệm cận với 0, ngược lại, nếu đầu vào là một số thực dương lớn sẽ cho đầu ra là một số tiệm cận với 1. Trong quá khứ hàm Sigmoid hay được dùng vì có đạo hàm rất đẹp. Tuy nhiên hiện nay hàm Sigmoid rất ít được dùng vì những nhược điểm sau:
* Hàm Sigmoid bão hoà và triệt tiêu gradient.
* Hàm Sigmoid không có trung tâm là 0 gây khó khăn cho việc hội tụ.

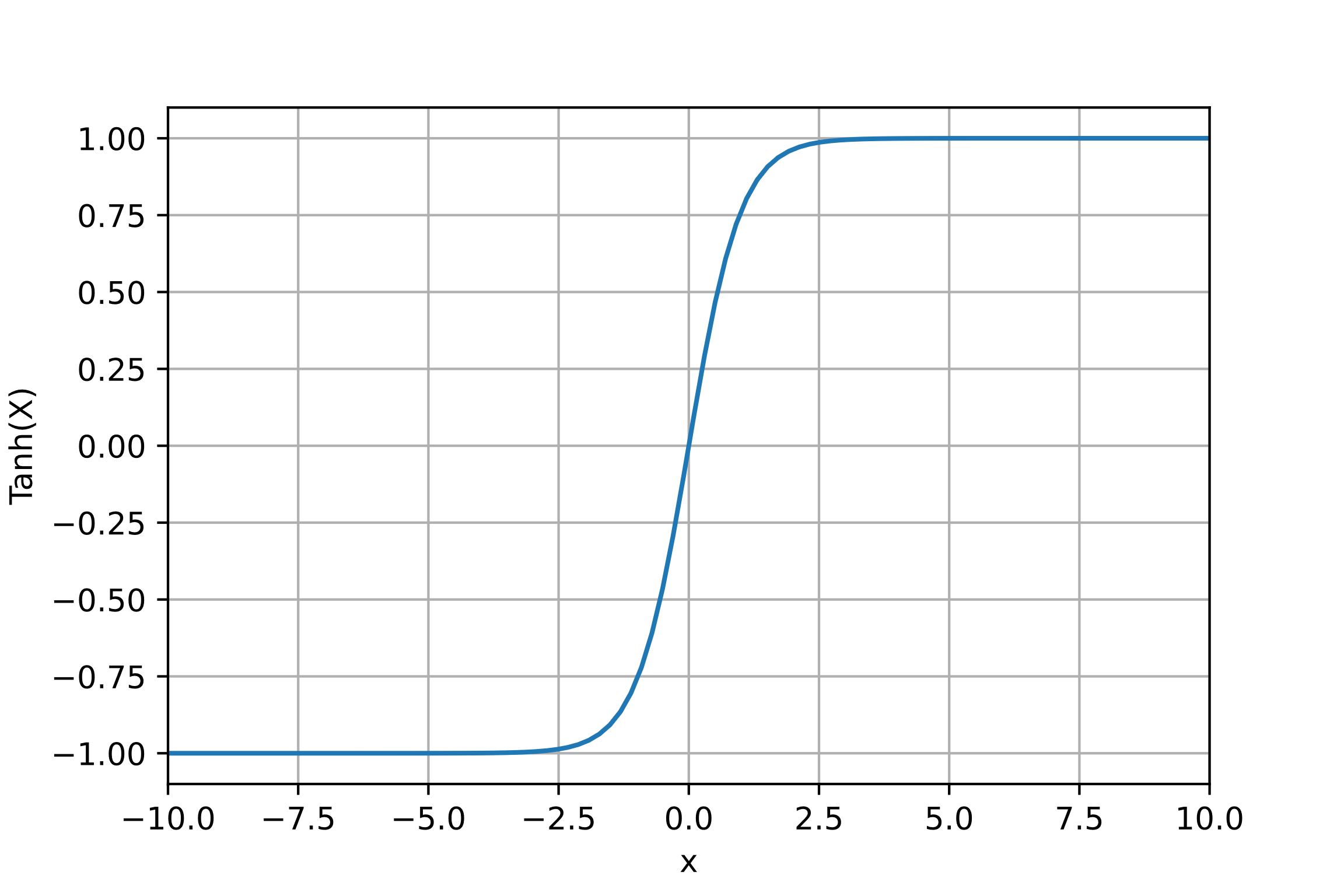
1. **Hàm Tanh**

* *Công thức:*

A picture containing chart

Description automatically generated

* *Phân tích*



Hình 1. 4: Đồ thị hàm Tanh

* Hàm nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (-1; 1). Cũng như Sigmoid, hàm Tanh bị bão hoà ở 2 đầu (gradient thay đổi rất ít ở 2 đầu). Tuy nhiên hàm Tanh lại đối xứng qua 0 nên khắc phục được một nhược điểm của Sigmoid.

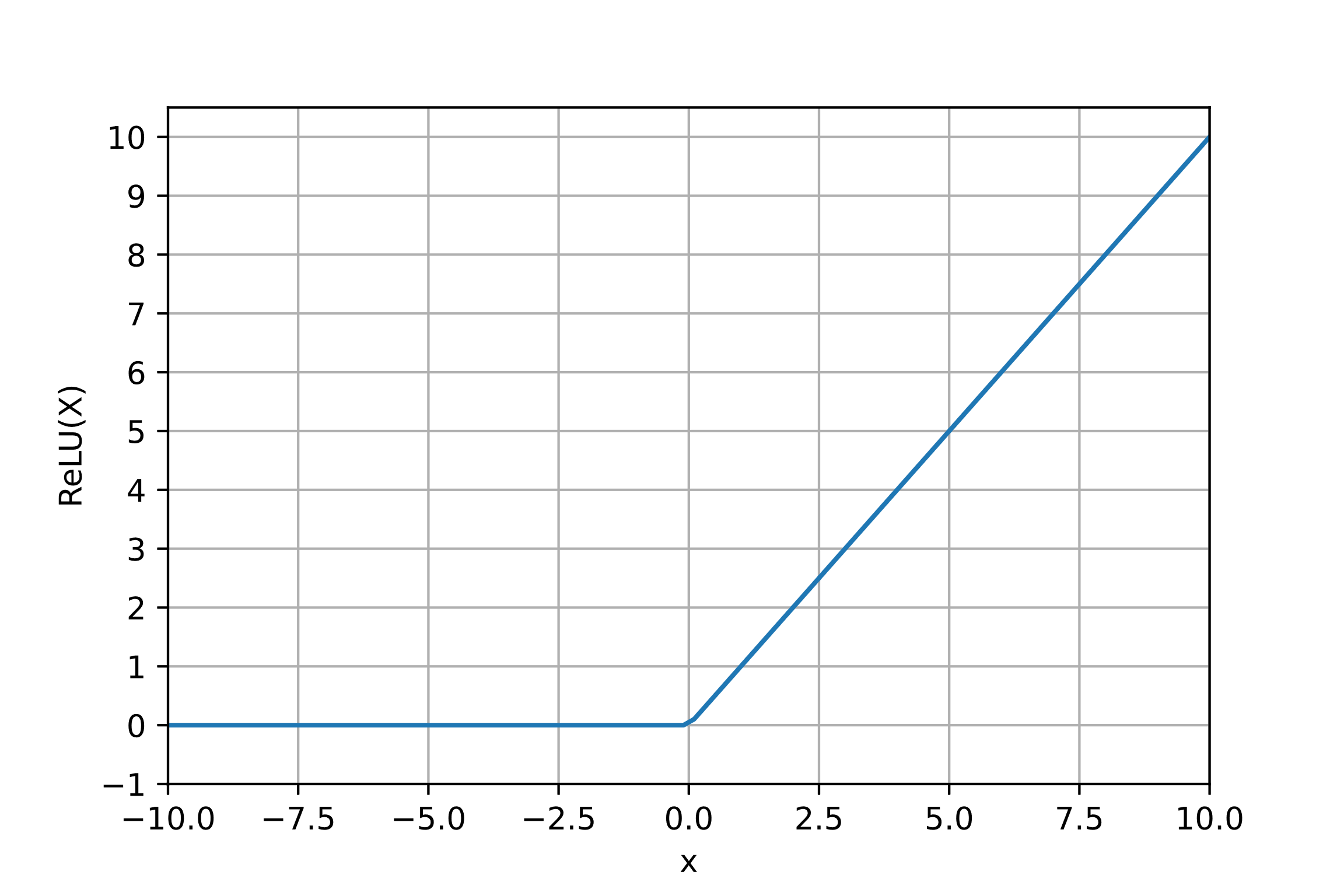
1. **ReLU**

* *Công thức:*

Text

Description automatically generated with low confidence

* *Phân tích:*



Hình 1. 5: Đồ thị hàm ReLU

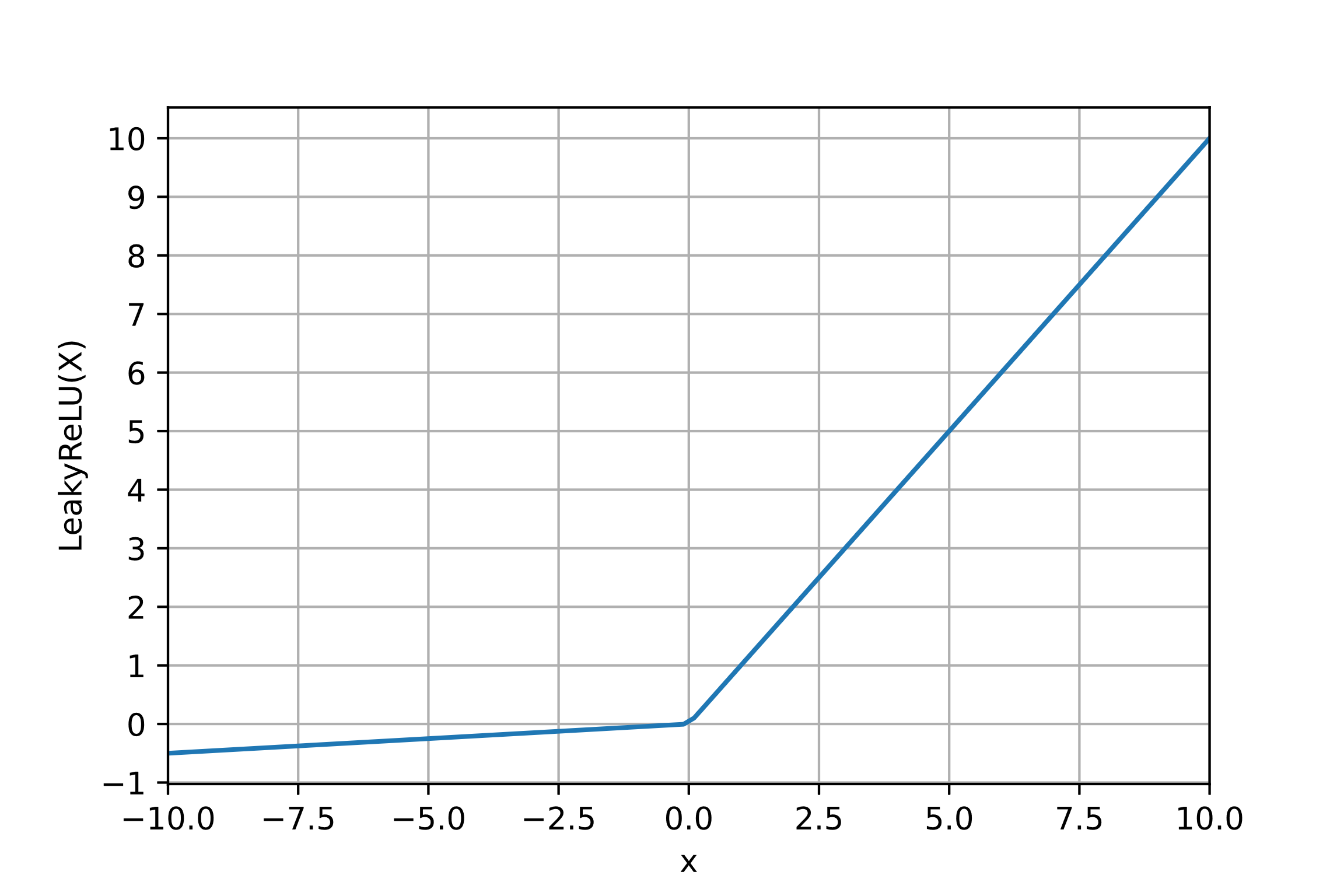
* Hàm ReLU đang được sử dụng khá nhiều trong những năm gần đây khi huấn luyện các mạng neuron. ReLU đơn giản lọc các giá trị < 0. Nhìn vào công thức chúng ta dễ dàng hiểu được cách hoạt động của nó. Một số ưu điểm khá vượt trội của nó so với Sigmoid và Tanh:
* Tốc độ hội tụ nhanh hơn hẳn. ReLU có tốc độ hội tụ nhanh gấp 6 lần Tanh (Krizhevsky et al.). Điều này có thể do ReLU không bị bão hoà ở 2 đầu như Sigmoid và Tanh.
* Tính toán nhanh hơn. Tanh và Sigmoid sử dụng hàm exp và công thức phức tạp hơn ReLU rất nhiều do vậy sẽ tốn nhiều chi phí hơn để tính toán.
* Tuy nhiên ReLU cũng có một nhược điểm: Với các node có giá trị nhỏ hơn 0, qua ReLU activation sẽ thành 0, hiện tượng đấy gọi là “Dying ReLU”. Nếu các node bị chuyển thành 0 thì sẽ không có ý nghĩa với bước linear activation ở lớp tiếp theo và các hệ số tương ứng từ node đấy cũng không được cập nhật với gradient descent.
* Khi learning rate lớn, các trọng số (weights) có thể thay đổi theo cách làm tất cả neuron dừng việc cập nhật.

1. **Leaky ReLU**

* *Công thức:*

A picture containing text

Description automatically generated*với α là hằng số nhỏ*

* *Phân tích:* 

Hình 1. 6: Đồ thị hàm Leaky ReLU

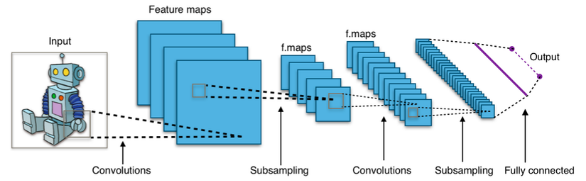
* Leaky ReLU là một cố gắng trong việc loại bỏ "dying ReLU". Thay vì trả về giá trị 0 với các đầu vào < 0 thì Leaky ReLU tạo ra một đường xiên có độ dốc nhỏ. Có nhiều báo cáo về việc hiệu Leaky ReLU có hiệu quả tốt hơn ReLU, nhưng hiệu quả này vẫn chưa rõ ràng và nhất quán.
* Ngoài Leaky ReLU có một biến thể cũng khá nổi tiếng của ReLU là PReLU. PReLU tương tự Leaky ReLU nhưng cho phép neuron tự động chọn hệ số α tốt nhất.

1. **Maxout**

* Khi đến với Maxout, chúng ta sẽ không sử dụng công thức dạng **f (** nữa. Một dạng khá phổ biến là Maxout neuron (giới thiệu bởi Goodfellow et al.)) được tính bằng công thức: **max (.** Leaky ReLU và ReLU là các dạng đặc biệt của công thức này (Với ReLU, áp dụng công thức với ). Maxout có tất cả ưu điểm của Leaky ReLU, ReLU. Tuy vậy, nó khiến mạng phải sử dụng gấp đôi số tham số (parameter) cho mỗi neuron, vì thế làm tăng đáng kể chi phí cả về bộ nhớ và tính toán - một điều cần suy xét khi huấn luyện mạng deep learning ở hiện tại.

## **Convolutional Neural Network - CNN**

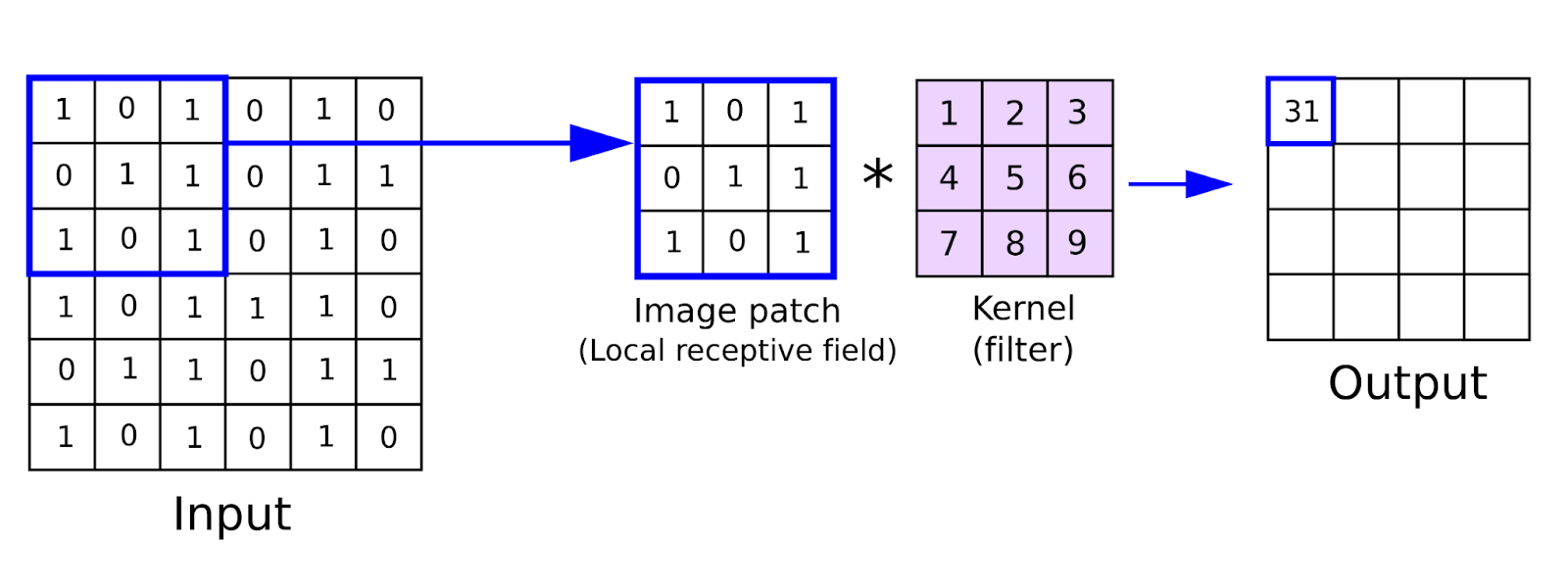
CNN hay mạng nơron tích chập là một trong những mô hình Deep Learning vô cùng tiên tiến, cho phép xây dựng các hệ thống thông minh với độ chính xác vô cùng cao.



Hình 1. 7: Cấu trúc mạng CNN

### Convolutional layer

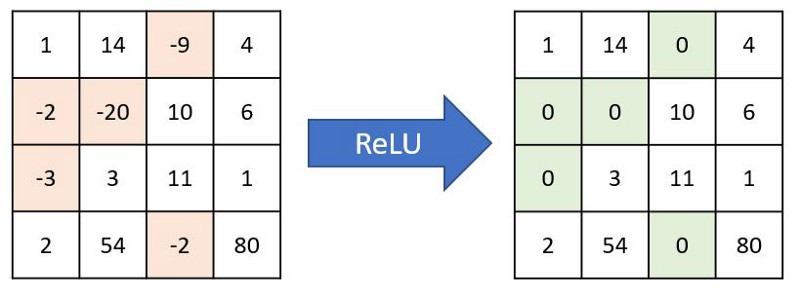
* Đây là lớp quan trọng nhất của CNN, có nhiệm vụ thực hiện mọi tính toán.
* Những yếu tố quan trọng của một convolutional layer là: stride, padding, filter map, feature map.
* CNN sử dụng các filter để áp dụng vào vùng của hình ảnh. Những filter map này được gọi là ma trận 3 chiều, mà bên trong nó là các con số và chúng là parameter.
* Stride có nghĩa là khi bạn dịch chuyển filter map theo pixel dựa vào giá trị trừ trái sang phải. Và sự chuyển dịch này chính là Stride.
* Padding: Là các giá trị 0 được thêm vào với lớp input.
* Feature map: Nó thể hiện kết quả của mỗi lần filter map quét qua input. Sau mỗi lần quét sẽ xảy ra quá trình tính toán.



Hình 1. 8: Lớp tích chập

### ReLU layer

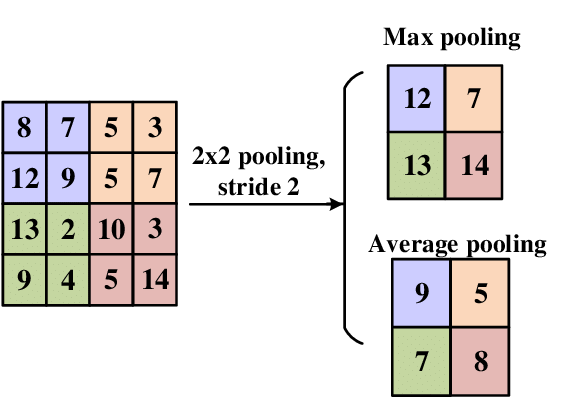
* Là hàm kích hoạt trong neural network và được gọi là activation function. Tác dụng của hàm kích hoạt là mô phỏng các nơron có tỷ lệ truyền xung qua axon. Trong activation function thì nó còn có hàm nghĩa là: Relu, Leaky, Tanh, Sigmoid, Maxout… Hiện nay hàm Relu được dùng phổ biến và vô cùng thông dụng.
* Nó được sử dụng nhiều cho các nhu cầu huấn luyện mạng nơron thì Relu mang lại nhiều ưu điểm nổi bật như: việc tính toán sẽ trở nên nhanh hơn... Quá trình sử dụng Relu, cần lưu ý đến vấn đề tùy chỉnh các learning rate và theo dõi dead unit. Những lớp Relu layer đã được sử dụng sau khi filter map được tính ra và áp dụng hàm Relu lên những giá trị của filter map.



Hình 1. 9: Hàm kích hoạt ReLU

### Pooling layer

* Khi đầu vào quá lớn, những lớp pooling layer sẽ được xếp vào giữa giữa những lớp Convolutional layer để làm giảm parameter. Hiện nay, pooling layer có 2 loại chủ yếu là: max pooling và average.



Hình 1. 10: Lớp pooling

### Fully connected layer

* Lớp này có nhiệm vụ đưa ra kết quả sau khi lớp convolutional layer và pooling layer đã nhận được ảnh truyền. Lúc này, ta thu được kết quả là model đã đọc được thông tin của ảnh và để liên kết chúng cũng như cho ra nhiều output hơn thì ta sử dụng fully connected layer.
* Ngoài ra, nếu như fully connected layer có được giữ liệu hình ảnh thì chúng sẽ chuyển nó thành mục chưa được phân chia chất lượng. Cái này khá giống với phiếu bầu rồi chúng sẽ đánh giá để bầu chọn ra hình ảnh có chất lượng cao nhất.

# **CHƯƠNG 2: XÂY DỰNG VÀ HUẤN LUYỆN MÔ HÌNH**

## **2.1. Tổng quan chương trình**

### 2.1.1. Công cụ sử dụng

Google Colab hay Google Colaboratory là một sản phẩm của Google Research. Colab dựa trên Jupyter Notebook, người dùng có thể viết và thực thi đoạn mã python thông qua trình duyệt.

Ưu điểm:

* Không yêu cầu cài đặt hay cấu hình máy tính
* Cung cấp tài nguyên máy tính như CPU, GPUs, TPUs giúp train mô hình nhanh hơn.

### 2.1.2. Ngôn ngữ lập trình

* Ngôn ngữ lập trình Python và những thư viện liên quan tới xây dựng mô hình AI như Keras, Tensorflow. Thư viện giúp xử lý dữ liệu như Numpy, zipfile.
* OpenCV (Open-Source Computer Vision Library) một thư viện thuộc về Thị giác máy tính (Computer Vision).

### 2.1.3. Các thư viện và kỹ thuật sử dụng trong chương trình

* Keras: là một mã nguồn mở cho Neural Network được viết bởi ngôn ngữ Python. Là thư viện được phát triển vào năm 2005 bởi Francois Chollet, là một kỹ sư nghiên cứu Deep Learning.
* Matplotlib: một thư viện vẽ đồ thị cho ngôn ngữ lập trình Python và phần mở rộng toán học số NumPy của nó. Nó cung cấp một API hướng đối tượng để nhúng các lô vào ứng dụng bằng cách sử dụng các bộ công cụ GUI có mục đích chung như Tkinter, wxPython, Qt hoặc GTK
* To\_categorical: chuyển các nhãn dưới dạng số nguyên thành ma trận nhị phân
* Dropout: dropout là việc chúng ta sẽ bỏ qua một vài unit trong suốt quá trình train trong mô hình với mục đích để tránh over-fitting những unit bị bỏ qua được lựa chọn ngẫu nhiên.
* Hàm mất mát categorical\_crossentropy
* Hàm tối ưu Adam

## **2.2. Chương trình**

* Hướng tiếp cận bài toán: mô hình sau khi được huấn luyện và kiểm tra độ chính xác sẽ được lưu lại, sau đó được load lại, tải lên một vài bức ảnh từ máy tính sau đó đưa vào mô hình để dự đoán biển báo đó thuộc loại nào trong 43 loại trong bộ dữ liệu

### 2.2.1 Bộ dữ liệu(dataset)

* Bộ dữ liệu được sử dụng là German Traffic Sign – một bộ dữ liệu nổi tiếng trong lĩnh vực AI về biển báo giao thông
* Bộ dữ liệu được chia thành 3 thư mục và các file csv

Text

Description automatically generated with medium confidence

Hình 2. 1: Cấu trúc bộ dữ liệu

* Dữ liệu trong thư mục Train gồm có 39,209 ảnh với các kích thước khác nhau, chia thành 43 thư mục con tương ứng với 43 loại biển báo, ví dụ: *Speed Limit 20km/h, Stop, General Caution, …*

A picture containing shape

Description automatically generated

Hình 2. 2: Các loại biển báo có trong bộ dữ liệu

* Thư mục Test gồm các ảnh của các loại biển báo, các ảnh này được sử dụng để kiểm tra độ chính xác của mô hình sau khi huấn luyện đối với dữ liệu chưa nhìn thấy

A screenshot of a computer

Description automatically generated with low confidence

Hình 2. 3: Các ảnh trong thư mục Test

* Các file .csv được sử dụng để lưu thông tin của các ảnh trong thư mục bao gồm kích thước width, height, classId, path, …

### 2.2.2. Cấu trúc mạng CNN cho mô hình

Diagram

Description automatically generated

Hình 2. 4: Cấu trúc mạng CNN của mô hình

* Sau mỗi lớp MaxPooling sẽ sử dụng kỹ thuật Dropout với p = 0.25 và sau lớp FC1 với p = 0.5 để giảm sự phụ thuộc giữa các nơron và tránh over-fitting

### 2.2.3. Xây dựng chương trình

* Tạo thư mục để chứa dữ liệu trên Colab sau đó download file nén chứa dữ liệu của mô hình và giải nén

Graphical user interface, text, application

Description automatically generated

Hình 2. 5: Tạo thư mục và download dữ liệu

* Load data từ thư mục Train và thực hiện phân chia dữ liệu train và test với tỉ lệ 8 – 2

Graphical user interface, text, application

Description automatically generated

Hình 2. 6: Load và phân chia dữ liệu

* Các ảnh được đưa về kích thước 30x30 trước khi đưa vào mảng, kết quả cho thấy có 39202 ảnh trong *data* cùng với các nhãn tương ứng được lưu vào *labels*, được phân chia thành 2 phần sử dụng cho train và validation trong quá trình huấn luyện
* Chuyển các label từ số nguyên (1...43) về dạng ma trận nhị phân

Text

Description automatically generated

Hình 2. 7: Convert nhãn sang ma trận nhị phân

* Xây dựng mô hình

Text

Description automatically generated

Hình 2. 8: Xây dựng mô hình sử dụng Sequential

* Huấn luyện mô hình: thực hiện huấn luyện mô hình với 15 epochs và batch\_size = 32

Text

Description automatically generated with low confidence

Hình 2. 9: Huấn luyện mô hình

Chart, line chart

Description automatically generated

Hình 2. 10: Kết quả huấn luyện mô hình

* Mô hình cho kết quả với độ chính xác trên tập train là 95.53% và trên tập test là 99.06%
* Lưu trữ mô hình đã được huấn luyện

**Graphical user interface, application, website

Description automatically generated**

Hình 2. 11: Lưu trữ mô hình

# **CHƯƠNG 3: KẾT QUẢ**

## **3.1. Kết quả huấn luyện mô hình**

* Kiểm tra độ chính xác của mô hình trên dữ liệu chưa nhìn thấy bằng cách đọc file Test.csv để lấy dữ liệu sau đó đưa vào mô hình để dự đoán

**Text

Description automatically generated**

Hình 3. 1: Đọc dữ liệu từ Test.csv và đưa vào mô hình

* Tính toán độ chính xác sử dụng hàm accuracy\_score của thư viện sklearn, kết quả cho thấy mô hình đạt độ chính xác là xấp xỉ 96% trên tập dữ liệu chưa nhìn thấy

Graphical user interface, text, application

Description automatically generated

Hình 3. 2: Tính toán độ chính xác từ labels và pred

**3.2. Một số kết quả khác**

* Thực hiện load loại mô hình đã được lưu

Text

Description automatically generated

Hình 3. 3: Load mô hình đã được lưu

* Upload các ảnh từ máy tính và đưa vào mô hình để dự đoán

**A picture containing application

Description automatically generated**

Hình 3. 4: Các ảnh và nhãn được dự đoán từ mô hình

# **CHƯƠNG 4: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM**

## **4.1. Kết luận**

### 4.1.1. Tính khả thi

* Sau thời gian tìm hiểu và nghiên cứu, nhóm chúng em đã xây dựng thành công chương trình “Phân loại chó mèo”. Tuy nhiên, nhận thấy tính khả thi khi áp dụng vào thực tế là chưa cao, bởi độ chính xác còn thấp và yêu cầu dữ liệu lớn hơn để nâng cao độ chính xác.

### 4.1.2. Thuận lợi

* Có nhiều tài liệu để tìm hiểu và nghiên cứu.
* Sự góp ý, hướng dẫn nhiệt tình của giảng viên.

### 4.1.3. Khó khăn

* Đề tài đòi hỏi nhiều kiến thức chuyên sâu, chưa có nhiều kinh nghiệm trong việc xây dựng ứng dụng AI nên việc tìm hiểu, nghiên cứu và xây dựng còn chậm.
* Do lịch học khác nhau nên các thành viên khó khăn trong việc liên lạc và trao đổi công việc.

## **4.2. Bài học kinh nghiệm**

Trong quá trình thực hiện đề tài, nhóm chúng em đã học được rất nhiều kiến thức chuyên ngành cũng như kỹ năng mềm.

### 4.2.1. Kiến thức

* Mạng nơron nhân tạo
* Mạng tích chập
* Hàm kích hoạt, hàm tối ưu
* Xây dựng và huấn luyện mô hình

### 4.2.2. Kỹ năng

* Kỹ năng làm việc nhóm
* Kỹ năng giao tiếp
* Kỹ năng quản lý thời gian hiệu quả
* Sử dụng công cụ hỗ trợ như Word, PowerPoint. Sử dụng Google Colab…

# **TÀI LIỆU THAM KHẢO**