

LẬP TRÌNH

AI

CHO NGƯỜI

MỚI BẮT ĐẦU

ĐỂ KHÔNG BỎ LỠ  
XU THẾ CỦA THỜI ĐẠI

## LẬP TRÌNH AI CHO NGƯỜI MỚI BẮT ĐẦU

Để không bỏ lỡ xu thế của thời đại

Bản quyền tiếng Việt© 2024 thuộc  
Công ty TNHH Sách và Truyền thông Việt Nam

Bất cứ sự sao chép nào không được sự đồng ý của Công ty TNHH Sách và Truyền thông Việt Nam đều là bất hợp pháp, vi phạm Luật Xuất bản Việt Nam và Công ước Berne về sở hữu trí tuệ.

Copyright© 2024 VIETNAM MEDIA AND BOOKS Co., Ltd  
All rights reserved.

*Các tác giả viết sách có nhu cầu xuất bản xin vui lòng liên hệ với chúng tôi qua:*

🌐 trithuctrebooks@gmail.com

☎ 024.62932066 - 090 607 0779

✉ trithuctrebooks.com

📍 Số 28, Lô A, Khu 7,2ha Vĩnh Phúc, P. Vĩnh Phúc, Q. Ba Đình, Hà Nội

TS. LƯƠNG ANH VŨ - TẠ VĂN DŨNG



ĐỂ KHÔNG BỎ LỠ  
XU THẾ CỦA THỜI ĐẠI



NHÀ XUẤT BẢN THANH NIÊN

### **Hai tác giả:**

Tiến sĩ Lương Anh Vũ: Tốt nghiệp chuyên ngành Học máy (Machine Learning) tại Đại học Griffith, Úc; đồng sáng lập tổ chức mã nguồn mở Young Monkeys.

Tạ Văn Dũng: Đồng sáng lập tổ chức mã nguồn mở Young Monkeys.



## LỜI NÓI ĐẦU

LẬP TRÌNH AI CHO NGƯỜI MỚI BẮT ĐẦU – ĐỂ KHÔNG BỎ LỠ XU THẾ CỦA THỜI ĐẠI - Cuốn sách dành cho những người mới bắt đầu tìm hiểu về lập trình AI.

Sự ra đời của ChatGPT đã thúc đẩy sự phát triển vượt bậc của ngành trí tuệ nhân tạo, khi hàng trăm tỉ đô la đã được đầu tư vào lĩnh vực này bởi các công ty khởi nghiệp và các tập đoàn công nghệ trên toàn cầu, khiến AI trở thành xu thế không thể đảo ngược. Sự ra đời của ChatGPT nói riêng và các công nghệ AI để hỗ trợ lập trình nói chung đã và đang làm cho công việc lập trình giờ đây đơn giản hơn rất nhiều. Một lập trình viên có thể làm được nhiều việc hơn, tạo ra sản phẩm nhanh hơn với chất lượng tốt hơn, cho dù chỉ bằng số năm kinh nghiệm ít ỏi. Điều này đặt áp lực lên các doanh nghiệp khi họ cần phải tối ưu nguồn nhân lực, thực hiện những ý tưởng với độ khó cao trong thời gian ngắn. Vậy nên ngày nay, lập trình viên dù ít dù nhiều cũng cần trang bị cho bản thân các kiến thức cơ bản về AI để sử dụng công cụ này một cách hiệu quả hơn trong công việc, đảm bảo lợi thế cạnh tranh của mình.

Cũng chính vì có các công cụ tra cứu thông tin bằng AI có sẵn, chúng tôi nghĩ rằng không khó để bạn tìm ra được các định nghĩa cơ bản cho các phần kiến thức trong cuốn sách này. Vì vậy chúng tôi sẽ rút gọn phần mô tả và giải thích cho các thuật ngữ chúng tôi cho là phổ biến để tập trung vào các kiến thức cốt lõi, giúp cuốn sách này gọn nhẹ hơn và cũng thuận tiện hơn cho bạn khi mang theo bên mình.

Mục tiêu của chúng tôi là giúp bạn thấy rằng lập trình AI không thực sự phức tạp như bạn nghĩ khi ngày nay đã có sự hỗ trợ đến từ các thư viện lập trình. Chúng tôi cũng hiểu rằng toán học không phải là một môn dễ dàng, vậy nên nội dung sẽ sử dụng tối thiểu các công thức toán học để bạn có thể tập trung vào lập trình và hoàn thành các ví dụ dựa trên các thư viện lập trình có sẵn.

TS. LƯƠNG ANH VŨ, TẠ VĂN DŨNG

# MỤC LỤC

Các thuật ngữ.....	9
<b>CHƯƠNG 1: GIỚI THIỆU VỀ AI</b> .....	<b>21</b>
AI là gì? .....	22
Sự phát triển của AI.....	24
Mô hình (model) là gì? .....	27
Ngôn ngữ lập trình Python.....	29
Thư viện Matplotlib .....	32
Thư viện numpy .....	34
Một số hàm dành cho xác suất thống kê .....	35
Một số hàm dành cho giải tích .....	36
<b>CHƯƠNG 2: BÀI TOÁN TRÍ TUỆ NHÂN TẠO</b> .....	<b>40</b>
Phát biểu bài toán .....	41
Giới thiệu hai bộ dữ liệu.....	42
Bộ dữ liệu chó mèo .....	42
Bộ dữ liệu Fashion MNIST .....	43
Sử dụng mô hình xử lý ảnh truyền thống.....	54
Lý thuyết cơ sở.....	55
Vectơ là gì?.....	55
Khoảng cách Bhattacharyya.....	57
Phân loại chó mèo.....	59
Phân loại mặt hàng thời trang.....	62
Tổng kết.....	68
Sử dụng mô hình học máy truyền thống.....	69

Support Vector Machine (SVM).....	70
Decision Trees (Cây quyết định).....	94
Random Forest.....	111
Naive Bayes .....	121
K-nearest neighbors (KNN) .....	134
Sử dụng mạng nơ-ron .....	149
Lý thuyết mạng nơ-ron.....	150
Giới thiệu về thư viện lập trình PyTorch .....	154
Giới thiệu cấu trúc dữ liệu tensor.....	155
Đọc dữ liệu với Dataset và DataLoader .....	167
Xây dựng mô hình mạng nơ-ron kết nối đầy đủ .....	180
Giải thích cơ chế đạo hàm tự động.....	197
Tối ưu tham số mô hình với Gradient Descent.....	215
Mã nguồn hoàn chỉnh .....	224
Cải tiến mô hình.....	239
Tổng kết.....	246
Sử dụng mô hình mạng nơ-ron tích chập.....	247
Lý thuyết cơ bản .....	248
Tìm hiểu một số hàm .....	250
Mã nguồn cài đặt .....	261
Sự khác biệt đến từ đâu?.....	267
Tổng kết.....	268
Tổng hợp kết quả.....	270
<b>CHƯƠNG 3: ỨNG DỤNG VÀO THỰC TẾ.....</b>	<b>272</b>
Một chút thiết kế hệ thống.....	273
Cài đặt máy chủ AI.....	275
Đọc tham số và tạo mô hình.....	275

Tạo hàm đoán nhãn của ảnh.....	277
Tạo API.....	278
Mã nguồn đầy đủ.....	280
Khởi chạy máy chủ web.....	284
Gọi API.....	285
<b>CHƯƠNG 4: CHATGPT .....</b>	<b>286</b>
<b>CHƯƠNG 5: ĐỊNH HƯỚNG NGHỀ NGHIỆP .....</b>	<b>289</b>
Nhà phát triển ứng dụng AI.....	290
Chuyên gia phân tích dữ liệu.....	291
Kỹ sư AI.....	291
Nhà nghiên cứu AI.....	293
Thị trường việc làm tại Việt Nam.....	293
Nhận định về tương lai nghề nghiệp.....	296
<b>KẾT LUẬN.....</b>	<b>297</b>
<b>ĐỊNH HƯỚNG TIẾP THEO CHO CUỐN SÁCH.....</b>	<b>298</b>
<b>PHỤ LỤC.....</b>	<b>299</b>
PL1. Mã nguồn đồ thị hóa việc phân lớp kết quả được dự đoán.....	299
PL2. Cài đặt môi trường.....	300
<b>THAM KHẢO.....</b>	<b>302</b>

# Các thuật ngữ

## A

**ANN:** Artificial Neural Network - mạng nơ-ron nhân tạo.

**API:** Application Programming Interface - giao diện lập trình ứng dụng. Có thể hiểu là các lối vào mà một chương trình máy tính mở ra, cho phép các chương trình khác kết nối đến để trao đổi thông tin.

**argmax:** Viết tắt của cụm từ "argument of the maximum" (đối số của giá trị lớn nhất). Trong toán học, argmax của một hàm là đầu vào (hoặc đối số) mà làm cho hàm đạt được giá trị lớn nhất. Ví dụ, nếu bạn có một hàm số  $f(x)$  đạt giá trị lớn nhất tại  $x = x_0$ , thì  $x_0$  đó chính là argmax của  $f(x)$ .

## B

**Batch:** Bó. Có thể hiểu là một tập con của bộ dữ liệu dùng để huấn luyện.

**Bias:** Hệ số tự do.

## C

**Channel:** Kênh của dữ liệu.

**Chiều dữ liệu:** Dùng để mô tả các thuộc tính hoặc đặc trưng của một tập hợp dữ liệu. Nó hỗ trợ việc phân loại, sắp xếp và phân tích dữ liệu. Ví dụ, một bức ảnh màu kích thước 28x28 (pixel) được đặc trưng bởi 3 yếu tố:

1. Chiều rộng: 28.

2. Chiều cao: 28.

3. Số lượng kênh màu: 3 (RGB).

Số chiều dữ liệu là:  $28 * 28 * 3 = 2,352$ .

## Đ

**Định lý xấp xỉ phổ quát:** Tiếng Anh là Universal Approximation Theorem. Định lý này nói rằng một mạng nơ-ron nhân tạo với ít nhất một tầng ẩn và sử dụng các hàm kích hoạt phi tuyến tính có khả năng xấp xỉ bất kỳ hàm liên tục nào trên một tập hợp đóng trong không gian nhiều chiều với độ chính xác mong muốn, miễn là có đủ số lượng nơ-ron trong tầng ẩn.

**Đặc trưng:** Là một thuộc tính của dữ liệu thể hiện được tính chất cụ thể của đối tượng. Ví dụ như hình dạng, màu sắc, chiều dài, chiều rộng... đều là các đặc trưng của một đối tượng.

## E

**Epoch:** Trong lĩnh vực học máy và đặc biệt là học sâu, epoch là một khái niệm quan trọng, đánh dấu một vòng lặp hoàn toàn của một mô hình học máy qua toàn bộ tập dữ liệu huấn luyện. Trong quá trình huấn luyện mô hình học máy, dữ liệu huấn luyện thường được chia thành các batch (bó) để tăng tốc tính toán và cập nhật trọng số của mô hình. Mỗi batch thường được sử dụng để cập nhật trọng số một lần trong mỗi epoch.

Một epoch bao gồm việc duyệt qua tất cả các batch trong tập dữ liệu huấn luyện một lần. Mục tiêu của việc lặp lại qua các epoch là để mô hình học được toàn bộ dữ liệu huấn luyện nhiều lần, từ đó cải thiện khả năng tổng quát hóa và hiệu suất của mô hình.

Sau mỗi epoch, các độ đo như độ chính xác trên tập dữ liệu kiểm tra thường được tính toán để đánh giá hiệu suất của mô hình trên dữ liệu mà nó chưa từng thấy (dữ liệu kiểm tra). Điều này giúp đánh giá xem liệu mô hình đã học được cách tổng quát hóa từ dữ liệu huấn luyện hay không.

**exp:** Công thức tính hàm số mũ tự nhiên:  $\exp(x) = e^x$ . Trong đó:

1.  $e$  là một hằng số, được gọi là số Euler, và có giá trị xấp xỉ là 2.71828.
2.  $x$  là số mũ.

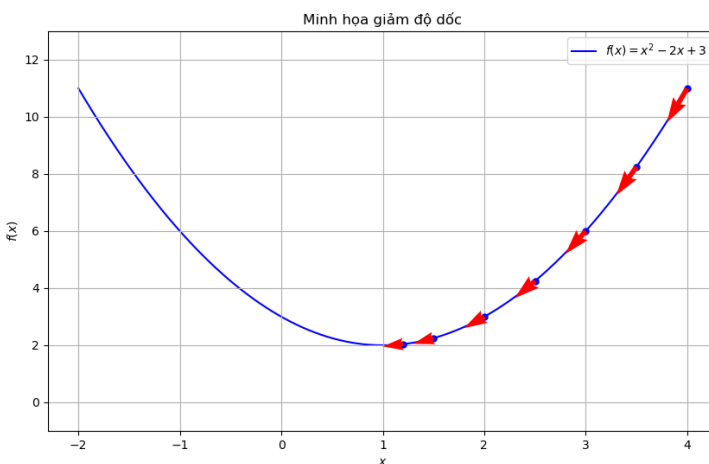
## F

**Fine-tune:** Là việc huấn luyện tiếp một mô hình đã được huấn luyện trước đó (thường là mạng nơ-ron) trên một bộ dữ liệu mới. Về mặt toán học thì nó sẽ làm thay đổi trọng số của mạng nơ-ron để phù hợp với bộ dữ liệu mới nhưng vẫn tận dụng được những gì mà mô hình đã được huấn luyện trước đó.

**Fully connected layer:** Tầng được kết nối đầy đủ.

## G

**Gradient:** Có thể dịch ra tiếng Việt là “độ dốc”, hay trong toán học thì nó là một vectơ chứa các đạo hàm bậc nhất của một hàm số theo các biến số của nó. Ví dụ, trong hình dưới đây, mỗi vectơ màu đỏ biểu diễn hướng của một gradient.



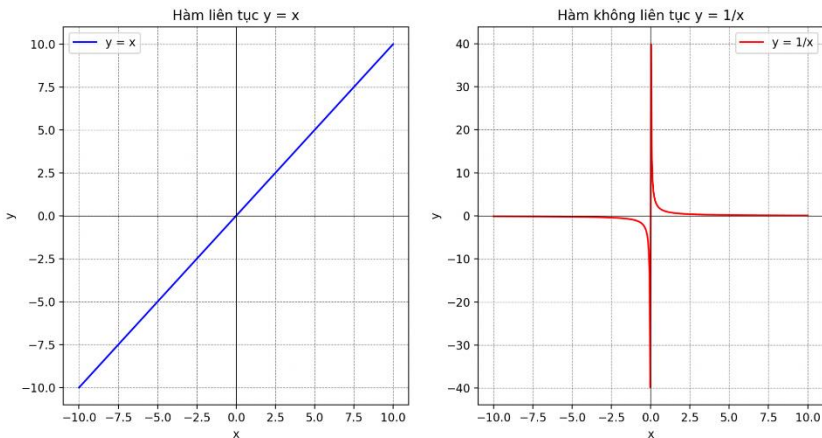
**Gradient descent:** Giảm độ dốc. Đây à một phương pháp tối ưu sử dụng đạo hàm để tìm giá trị lớn nhất hoặc nhỏ nhất cho một hàm số. Ví dụ, công thức lặp được sử dụng cho hàm số một biến:  $x^{(k+1)} = x^{(k)} - \eta \times f'(x^{(k)})$

Hình phía trên cũng minh họa cho việc suy giảm độ dốc từ điểm (4, 11) xuống về điểm (1, 2).

## H

**Hàm kích hoạt:** Tiếng Anh là “activation function”, lấy cảm hứng từ hoạt động dẫn truyền xung thần kinh trong mạng nơ-ron sinh học. Bằng cách áp dụng một phép biến đổi phi tuyến tính lên đầu ra của một tầng trong mạng nơ-ron, hàm kích hoạt sẽ giúp cho mạng nơ-ron có khả năng học hỏi và biểu diễn các mô hình phi tuyến tính phức tạp.

**Hàm liên tục:** Là một hàm số không có sự thay đổi đột ngột trong giá trị của nó. Ví dụ, hàm  $y = x$  là liên tục còn hàm  $y = \frac{1}{x}$  là không liên tục, bị gián đoạn ở điểm  $x = 0$  do không có số nào có thể chia cho 0. Bạn có thể quan sát hình dưới đây để hiểu rõ hơn.



**Hệ số tự do:** Trong toán học, ví dụ, ta có hàm số  $y = ax + b$  thì  $b$  chính là hệ số tự do. Nếu không có  $b$ , chỉ có  $y = ax$  là một đường thẳng luôn đi qua gốc tọa độ thì trong bài toán phân lớp, khi cần dịch chuyển



đường thẳng để chia cắt được nhiều đối tượng nhất có thể, chúng ta chỉ thực hiện được phép xoay đường thẳng quanh gốc tọa độ và điều này làm hạn chế khả năng phân lớp. Nhưng với  $b$  chúng ta có thể dịch chuyển được đường thẳng đến bất kỳ đâu chúng ta muốn, kết hợp với phép xoay, chúng ta sẽ có khả năng tìm ra đường thẳng phân lớp tốt nhất. Đây chính là ý nghĩa của từ “tự do”.

**Hidden layer:** Tầng ẩn.

**Histogram:** Biểu đồ tần suất. Histogram được đề cập trong cuốn sách này là một đặc trưng của ảnh dạng biểu đồ biểu diễn số lượng điểm ảnh tương ứng với mức độ sáng tối của bức ảnh sau khi chụp. Trong không gian màu RGB (Red, Green, Blue) phổ biến, histogram màu sẽ đo lường tần suất xuất hiện của các giá trị màu đỏ, xanh lá cây và xanh dương trong ảnh. Mỗi kênh màu trong không gian màu RGB có giá trị từ 0 đến 255, biểu thị cường độ của màu tương ứng. Histogram màu thể hiện mức độ phân bố của các giá trị màu này trong bức ảnh. Chính vì điều này nên chúng ta có thể sử dụng histogram để làm đặc trưng phân biệt giữa ảnh này và ảnh kia.

**Huấn luyện mô hình:** Là quá trình trong đó mô hình học máy được “học” từ dữ liệu. Bản chất của quá trình này là việc cập nhật các tham số của mô hình để nó có thể đưa ra các dự đoán chính xác hơn trên dữ liệu mới trong tương lai.

Khi bắt đầu, các tham số của mô hình học máy thường được khởi tạo ngẫu nhiên hoặc theo một cách nào đó. Trong quá trình huấn luyện, mô hình sử dụng một thuật toán tối ưu, ví dụ như Gradient Descent để điều chỉnh các tham số này, nhằm giảm thiểu sai số giữa dự đoán của mô hình và giá trị thực tế trong tập dữ liệu huấn luyện.

## I

**Iterable:** Là một cách thức để duyệt qua các phần tử trong một mảng hoặc một tập hợp. Nó có thể được biểu diễn với mã nguồn Python như sau:

```
class MyIterable:
    def __init__(self, n):
        self.n = n
        self.current = 1

    def __iter__(self):
        return self

    def __next__(self):
        if self.current <= self.n:
            result = self.current
            self.current += 1
            return result
        else:
            raise StopIteration
```

# Ví dụ sử dụng iterable lặp qua các số tự nhiên từ 1 đến 5:

```
iterable = MyIterable(5)
for number in iterable:
    print(number)
```

**Kết quả chạy:**

```
1
2
3
4
5
```

## K

**Kênh của dữ liệu:** Là các thành phần riêng lẻ mà một đơn vị dữ liệu (như một pixel trong hình ảnh) được chia ra, đại diện cho một

chiều hoặc một giá trị khác nhau của dữ liệu đó. Ví dụ, mỗi pixel thường được biểu diễn bởi một vectơ có ba giá trị (RGB) nên số kênh sẽ là 3. Trong trường hợp ảnh đen trắng thì số kênh sẽ là 1.

**Không gian đặc trưng:** Tiếng Anh là “feature space”, là không gian mà mỗi điểm dữ liệu trong tập dữ liệu được biểu diễn dưới dạng một vectơ đặc trưng. Mỗi chiều của không gian này tương ứng với một đặc trưng cụ thể của dữ liệu.

**Không gian Euclid:** Được lấy theo tên nhà toán học Hy Lạp cổ đại Euclid, là không gian của tất cả các bộ  $n$  số thực  $(x_1, x_2, \dots, x_n)$ . Các bộ  $n$  số thực này thường được gọi là các điểm, và không gian  $n$ -chiều này thường được ký hiệu là  $\mathbb{R}^n$ .

**Kỳ vọng (xác suất):** Là giá trị mong đợi hoặc trung bình của một biến ngẫu nhiên. Nó được tính bằng tổng các tích giữa xác suất xảy ra của mỗi giá trị có thể của biến với giá trị đó. Ví dụ, chúng ta chơi trò đoán xúc xắc, nếu đoán trúng số chấm mỗi lần chúng ta sẽ nhận được gấp 6 lần số tiền đặt cược. Chúng ta chỉ có một cơ hội thắng được 6 đồng nhưng có đến 5 khả năng bị thua mất 1 đồng nên giá trị kỳ vọng khoản lợi thu được từ việc đặt cược 1 đồng là:

$$(6 \times \frac{1}{6}) + (-1 \times \frac{5}{6}) \approx 0.17$$

## L

**Label:** Nhãn.

## M

**Mạng nơ-ron:** Hay mạng nơ-ron nhân tạo là một mô hình học máy lấy cảm hứng từ cấu trúc của bộ não con người, bao gồm các tầng nơ-ron kết nối với nhau để nhận dữ liệu đầu vào, xử lý thông qua các trọng số và hàm kích hoạt, rồi xuất kết quả ra. Nó có khả năng học từ dữ liệu thông qua quá trình huấn luyện mô hình.

**Miền xác định:** Hay tập xác định của một hàm số là miền chứa những giá trị của biến số làm cho hàm số đó có nghĩa. Ví dụ, hàm số  $f(x) = \frac{1}{x}$  có miền xác định là tất cả các số thực  $x$  khác 0.

**Model:** Trong cuốn sách này là viết gọn của mô hình học máy.

**Mô hình học máy:** Là tập hợp các thành phần được sử dụng để biểu diễn một thuật toán học máy.

## N

**Nhãn:** Nhãn hay nhãn lớp là tên danh mục để gom các dữ liệu cùng loại.

## O

**One-hot vector:** Trong học máy và học sâu, “one-hot vector” (hay vector one-hot) là một cách biểu diễn số học của một biến phân loại (categorical variable). Một vector one-hot là một vector mà tất cả các phần tử đều bằng 0, ngoại trừ một phần tử có giá trị là 1, thể hiện một trong các trạng thái hoặc lớp cụ thể của biến phân loại.

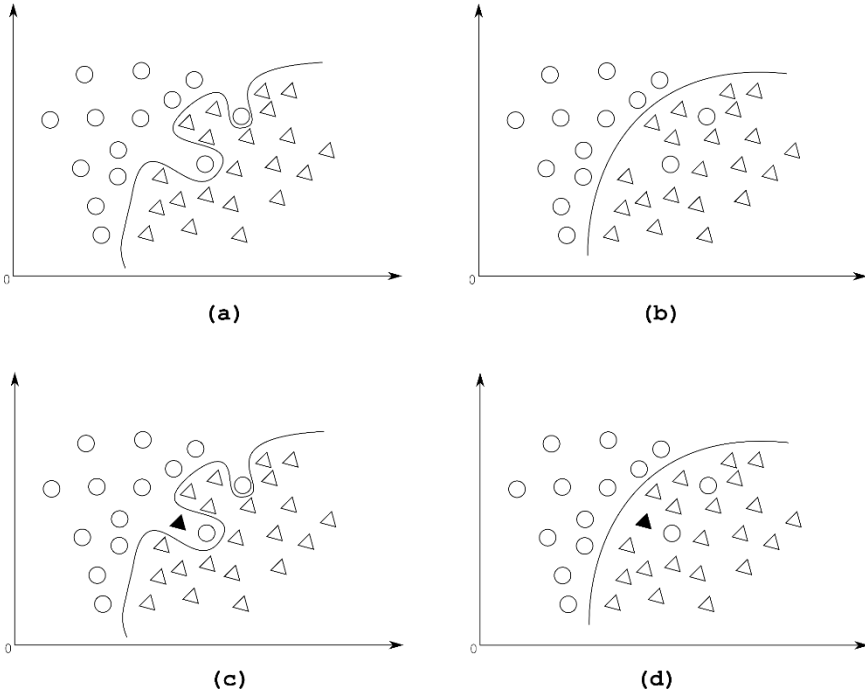
Ví dụ, giả sử chúng ta có một biến phân loại có 3 lớp: “apple”, “banana”, và “orange”. Mỗi lớp có thể được biểu diễn bằng một vector one-hot như sau:

- “apple”: [1, 0, 0]
- “banana”: [0, 1, 0]
- “orange”: [0, 0, 1]

Trong các vector one-hot, chỉ có một phần tử có giá trị là 1, đại diện cho lớp tương ứng, và tất cả các phần tử khác có giá trị là 0. Điều này giúp cho việc biểu diễn các biến phân loại trong các mô hình máy học trở nên dễ dàng và hiệu quả.

**OpenCV:** Một thư viện lập trình xử lý ảnh.

**Overfit:** Quá khớp. Thuật ngữ này mô tả một mô hình học máy quá khớp với dữ liệu huấn luyện hay nói cách khác là quá nhạy cảm với các nhiễu ngẫu nhiên, dẫn đến việc nó trở nên kém hiệu quả khi dự đoán với dữ liệu mới được truyền vào. Hãy lấy một ví dụ với hình bên dưới:



Đường cong ở hình (a) quá nhạy cảm với các hình tròn nằm xen kẽ với các hình tam giác, dẫn đến việc nó có thể phân lớp được toàn bộ các hình của hai lớp. Tuy nhiên, khi có một hình tam giác đen mới cần phân lớp ở hình (c), đường cong này sẽ phân nó vào lớp hình tròn, mặc dù hình này nên được phân vào lớp hình tam giác. Đường phân lớp chúng ta nên tìm là đường cong trơn ở hình (b) và (d), mặc dù nó bỏ qua hai hình tròn tại hình (b) tuy nhiên lại có thể phân lớp tốt hơn với dữ liệu mới tại hình (d).

## P

**Phi tuyến tính:** Là một khái niệm trong toán học dùng để mô tả mối quan hệ giữa các biến trong một hàm số không thể biểu diễn bằng một đường thẳng trong đồ thị hai chiều hoặc bằng một siêu phẳng trong không gian nhiều chiều. Ví dụ  $y = f(x) = x^2$ ,  $y = f(x) = \sin(x)$  và  $y = f(x) = x - 2 \times$  phần nguyên của  $\left(\frac{x+1}{2}\right)$  đều là

các hàm số phi tuyến tính vì đồ thị của chúng đều là các đường cong hoặc zigzag.

**Phương sai:** Trong lý thuyết xác suất thống kê, phương sai của một biến ngẫu nhiên là một thước đo sự phân tán của biến đó. Phương sai được tính bằng giá trị trung bình của các bình phương của chênh lệch giữa giá trị cụ thể và giá trị trung bình. Công thức của nó là:

$$\sigma^2 = \frac{\sum_{i=1}^N (x_i - \mu)^2}{N}$$

Trong đó:

1.  $N$  là số lượng các mẫu trong tập dữ liệu.
2.  $x_i$ : Là giá trị của mẫu thứ  $i$ .
3.  $\mu$ : Là giá trị trung bình của tập dữ liệu.

Phương sai cũng hàm ý các giá trị của biến thường cách giá trị kỳ vọng bao xa.

Giả sử chúng ta có một tập hợp dữ liệu là điểm số của 5 học sinh ( $N = 5$ ) trong một bài kiểm tra:  $\{8, 7, 6, 9, 10\}$ . Để tính phương sai của tập hợp này, ta thực hiện các bước sau:

1. Tính giá trị điểm trung bình  $\mu = \frac{8+7+6+9+10}{5} = 8$
2. Tính độ lệch của mỗi giá trị so với trung bình  $x_i - \mu$ , ta có:  $(8 - 8), (7 - 8), (6 - 8), (9 - 8), (10 - 8) = 0, -1, -2, 1, 2$ .
3. Bình phương các độ lệch  $(x_i - \mu)^2$ , ta có:  $0^2, (-1)^2, (-2)^2, 1^2, 2^2 = 0, 1, 4, 1, 4$
4. Tính tổng các bình phương độ lệch ta có:  $\sum_{i=1}^N (x_i - \mu)^2 = 0 + 1 + 4 + 1 + 4 = 10$ .
5. Chia tổng các bình phương độ lệch cho  $N = 5$ , ta có phương sai:  $\sigma^2 = \frac{10}{5} = 2$ .

Thông qua phương sai bằng 2, chúng ta có thể nói rằng điểm của học sinh có sự phân tán nhưng không vượt quá xa so với kỳ vọng điểm trung bình là 8.

**Postman:** Một phần mềm hỗ trợ gọi API.

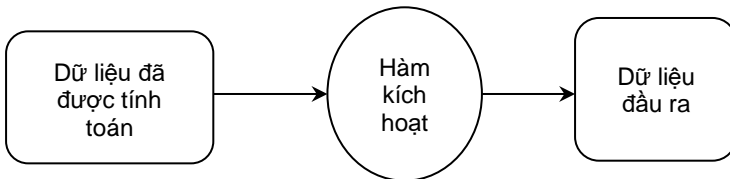
**S**

**Siêu phẳng:** Là một không gian con  $n - 1$  chia không gian  $n$  chiều thành hai nửa. Ví dụ, một đường thẳng chia một mặt phẳng thành hai nửa, một mặt phẳng chia một không gian ba chiều thành hai nửa không gian.

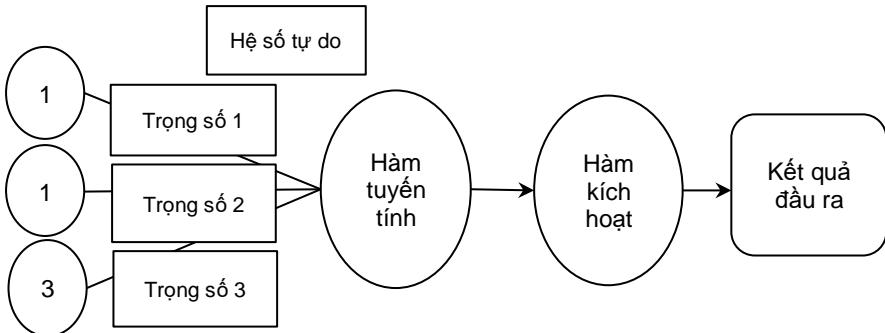
**T**

**Training:** Huấn luyện mô hình.

**Tầng ẩn:** Nó là tầng nằm giữa tầng đầu vào và tầng đầu ra trong cấu trúc của một mạng nơ-ron. Đặc điểm chính của tầng ẩn là nó không tiếp xúc trực tiếp với dữ liệu đầu vào hoặc đầu ra cuối cùng mà thay vào đó, nó xử lý dữ liệu và tạo ra các đặc trưng trung gian (intermediate features) mà tầng đầu ra sau đó sử dụng để tạo ra kết quả cuối cùng.



**Tầng kết nối đầy đủ:** Là một tầng mà mỗi nút trong tầng này được kết nối với mỗi nút trong tầng liền trước và liền sau. Điều này có nghĩa là mỗi đầu ra của tầng trước sẽ trở thành đầu vào cho tất cả các nút trong tầng này, và đầu ra của tầng này sẽ là đầu vào cho tầng liền sau. Một tầng kết nối đầy đủ có thể được minh họa như sau:



**Trọng số:** Là độ quan trọng của một đặc trưng của dữ liệu. Ví dụ, trong một ứng dụng hẹn hò, người tham gia sẽ quan tâm đến giới tính, mức thu nhập và khoảng cách địa lý thì chúng ta có thể đặt giới tính có độ quan trọng là 1, mức thu nhập là 0.8 và khoảng cách địa lý là 0.5. Trong toán học và học máy, với hàm  $f(x, y) = ax + by$  có 2 biến là  $x$  và  $y$  thì trọng số ở đây là  $a$  và  $b$ . Trong các công thức được đề cập trong sách như  $f(x) = w_1x^{(1)} + w_2x^{(2)}$  thì trọng số là  $w_1$  và  $w_2$ .

**Tuyến tính:** Là một khái niệm trong toán học mô tả mối quan hệ giữa các biến là một hàm  $f(x)$  có thể biểu diễn trên đồ thị hai chiều là một đường thẳng hoặc trong không gian là siêu phẳng. Ví dụ hàm  $y = f(x) = ax + b$  là một hàm tuyến tính.

## V

**Vector:** Hay “Vector”. Trong toán học, vector là một đoạn thẳng có phương hướng và độ lớn, trong giải tích thì vector thường được biểu diễn bằng một bộ số thực  $(x_1, x_2, \dots, x_n)$ , trong đó  $n$  là số chiều của vector.



# CHƯƠNG 1

## GIỚI THIỆU VỀ AI

Trong chương này chúng ta sẽ cùng nhau tìm hiểu về:

1. AI là gì?
2. Sự phát triển của AI.
3. Mô hình (model) là gì?
4. Ngôn ngữ lập trình Python.
5. Thư viện Matplotlib.
6. Thư viện NumPy.

# AI là gì?

Nếu bạn nghĩ AI là một cái gì đó giống như các robot có khả năng biến hình, xuyên không về quá khứ để bảo vệ các nhân vật quan trọng thì hiện tại, nó chỉ có trong các bộ phim khoa học viễn tưởng mà thôi.

AI là viết tắt của “artificial intelligence” hay “trí tuệ nhân tạo”. Đây là một khái niệm tương đối rộng và trừu tượng. Về cơ bản, nó thường được sử dụng để mô tả các phần mềm, máy tính hoặc máy móc có khả năng thực hiện một cách tự động các tác vụ mà thông thường cần đến trí tuệ con người. Các tác vụ phổ biến có thể kể đến là lái xe tự động, nghe giọng nói, hiểu ngôn ngữ tự nhiên, phân tích dữ liệu, chẩn đoán bệnh...

Lĩnh vực AI trong khoa học máy tính tập trung nghiên cứu và phát triển các thuật toán giúp máy tính có thể thay thế hoặc hỗ trợ con người nhận diện mẫu, dự đoán và ra quyết định, nhằm tăng hiệu suất trong các công việc khác nhau. Các thuật toán này được áp dụng rộng rãi ở rất nhiều lĩnh vực từ công nghiệp, y tế, tài chính cho đến giáo dục và giải trí.

Hai khái niệm chính trong lập trình AI mà chúng ta cần quan tâm đó là:

1. Học máy - Machine Learning.
2. Học sâu - Deep Learning.

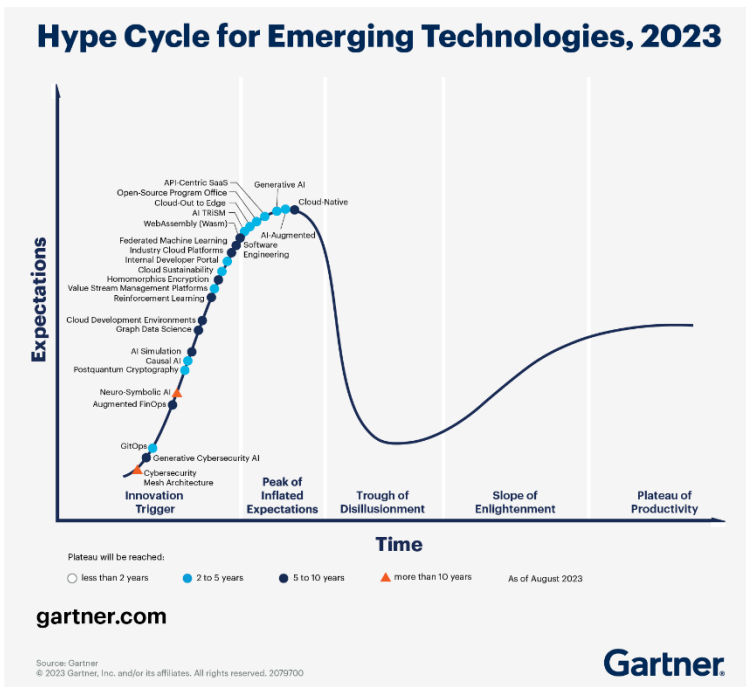
Học máy bao gồm các thuật toán cho phép máy tính giải các bài toán bằng kinh nghiệm (dữ liệu). Các thuật toán này học từ dữ liệu để tìm ra các mẫu và quy luật, từ đó thực hiện các dự đoán hoặc quyết định mà không cần lập trình logic chi tiết cho từng nhiệm vụ cụ thể.

Học sâu là một nhánh của học máy, sử dụng các mô hình nhiều tầng, chủ yếu là các mạng nơ-ron (Neural Networks) để xử lý dữ liệu. Các mô hình học sâu có khả năng tự động trích xuất đặc trưng từ dữ liệu phức tạp, giúp giải quyết các bài toán khó như xử lý ngôn ngữ tự nhiên, nhận diện hình ảnh và giọng nói.

# Sự phát triển của AI

Thời điểm trước cuối năm 2022, trước khi ChatGPT ra đời, phần lớn chúng ta vẫn còn khá mơ hồ về AI và vẫn chủ yếu sử dụng Google để tìm kiếm thông tin. Tuy nhiên ở thời điểm hiện tại, thời gian con người sử dụng Google đã giảm đi đáng kể và thay vào đó, chúng ta sử dụng ChatGPT để tìm kiếm câu trả lời cho các câu hỏi của mình. Điều đó có nghĩa là AI đã thực sự đi vào đời sống của con người.

Chúng ta có thể sử dụng biểu đồ của Gartner để hiểu về giai đoạn phát triển và nhận thức các công nghệ mới của của con người và cụ thể là về AI ở năm 2023.



Trong biểu đồ này, trục đứng thể hiện sự kỳ vọng vào sự phát triển các công nghệ mới của con người, còn trục ngang thể hiện các giai đoạn nhận thức, trong đó:

1. Innovation Trigger (Sự đổi mới/Khởi đầu): Đây là giai đoạn đầu tiên trong biểu đồ, khi công nghệ hoặc khái niệm mới bắt đầu thu hút sự chú ý của các nhà nghiên cứu, các nhà phát triển, và đôi khi cả truyền thông. Ở giai đoạn này, công nghệ còn đang trong giai đoạn nghiên cứu và phát triển, có ít sản phẩm hoặc ứng dụng thực tế. Tuy nhiên, tiềm năng của công nghệ thường là đề tài để bàn tán và tranh luận sôi nổi. Hiện tại, nhiều nhánh của AI đã vượt qua được giai đoạn này.
2. Peak of Inflated Expectations (Đỉnh của những kỳ vọng phóng đại): Trong giai đoạn này, công nghệ mới thu hút được nhiều sự quan tâm và thậm chí còn được quảng bá mạnh mẽ. Các câu chuyện thành công và tiềm năng của công nghệ thường được phóng đại, dẫn đến việc kỳ vọng về công nghệ này tăng cao. Người ta thường tin rằng công nghệ sẽ giải quyết được nhiều vấn đề lớn và có tác động đáng kể. Tuy nhiên, thực tế thường không như mong đợi. Hiện tại ở năm 2024, AI đang ở giai đoạn này với những thông tin liên tục được cập nhật và phóng đại.
3. Trough of Disillusionment (Hố sâu của sự vỡ mộng): Đây là giai đoạn mà những kỳ vọng cao trước đó bắt đầu giảm đi khi các hạn chế, khó khăn và thất bại trở nên rõ ràng. Người tiêu dùng, những nhà đầu tư công chúng... có thể trở nên thất vọng với công nghệ và sự quan tâm bắt đầu giảm. Những vấn đề như chi phí, kỹ thuật hoặc thực tiễn có thể gây ra sự vỡ mộng. Cá nhân chúng tôi nhận thấy giai đoạn này đang dần hiện hữu khi các ứng dụng của AI ngoài ChatGPT phục vụ chủ yếu cho mục tiêu cá nhân là thành công thì các ứng dụng khác được áp dụng để phục vụ các dự án thực tế như chăm sóc khách hàng lại có kết quả khá tiêu cực khi AI vẫn chưa thực sự thân thiện để khách hàng cảm thấy được sự tôn trọng và hữu ích với mình so với nhân viên là con người.

4. Slope of Enlightenment (Đốc của sự khai sáng): Giai đoạn này đánh dấu sự phục hồi của công nghệ. Các công ty và nhà phát triển bắt đầu hiểu rõ hơn về các ứng dụng thực tế của công nghệ và tìm cách giải quyết các vấn đề gặp phải trong giai đoạn trước. Sự nhận thức và thông hiểu về công nghệ dần dần cải thiện, nhiều ứng dụng thực tiễn bắt đầu xuất hiện.

5. Plateau of Productivity (Cao nguyên của sự hiệu quả): Đây là giai đoạn cuối cùng của biểu đồ, khi công nghệ đã được chấp nhận rộng rãi và bắt đầu mang lại giá trị thực sự. Các sản phẩm và dịch vụ dựa trên công nghệ này đã trở nên phổ biến và được triển khai trong nhiều lĩnh vực khác nhau. Lúc này, công nghệ đã được kiểm chứng và trở thành một phần của cuộc sống hàng ngày với nhiều ứng dụng thực tế và thành công.

# Mô hình (model) là gì?

Mô hình AI là một thuật toán hoặc một tập hợp các thuật toán được sử dụng để mô phỏng, đại diện hoặc giải quyết một nhiệm vụ cụ thể trong lĩnh vực trí tuệ nhân tạo (AI). Mô hình AI thường được xây dựng dựa trên dữ liệu và có khả năng học từ dữ liệu đó để tự điều chỉnh và cải thiện hiệu suất.

Có nhiều loại mô hình AI khác nhau, một số tiêu biểu bao gồm:

1. Các mô hình sử dụng xác suất thống kê và các quy tắc “nếu-thì” được xác định bởi chuyên gia.

2. Các mô hình học máy truyền thống, ví dụ như các thuật toán học có giám sát: Support Vector Machine, Cây quyết định, Random Forest... và các thuật toán học không giám sát: K-means Clustering, Hierarchical Clustering, DBSCAN...

3. Mạng nơ-ron nhân tạo (Artificial Neural Networks - ANN): Đây là một loại mô hình được lấy cảm hứng từ cấu trúc của não người. ANN thường được sử dụng trong các nhiệm vụ như nhận dạng hình ảnh, dịch ngôn ngữ và dự đoán chuỗi thời gian.

4. Mô hình học sâu (Deep Learning Models): Đây là một dạng của ANN với nhiều tầng ẩn, cho phép mô hình học được các đặc trưng phức tạp và trừu tượng từ dữ liệu đầu vào.

5. Mô hình học tăng cường (Reinforcement Learning Models): Loại mô hình này tập trung vào việc học từ trải nghiệm thông qua việc tương tác với một môi trường và nhận phản hồi dựa trên hành động của mô hình. Các ứng dụng của mô hình học tăng cường bao gồm việc điều khiển robot, chơi trò chơi và quản lý tài nguyên.

6. Các dạng mô hình khác như mô hình học biểu diễn (Representation Learning), mô hình học thời gian thực (Online Learning Models), mô hình học đồ thị (Graphical Models), hệ thống logic mờ (Fuzzy System), hệ thống lai (kết hợp nhiều phương pháp AI).

Các mô hình này được sử dụng để giải quyết các nhiệm vụ dự đoán như phân lớp dữ liệu, đề xuất dữ liệu tương đồng, nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, phát hiện gian lận...

Mỗi loại mô hình AI có ưu điểm và hạn chế riêng và việc lựa chọn mô hình phù hợp phụ thuộc vào yêu cầu cụ thể của bài toán và đặc điểm của dữ liệu cụ thể.



# Ngôn ngữ lập trình Python

Chúng tôi mong muốn bạn sẽ tìm hiểu ngôn ngữ Python vì nó là một trong những ngôn ngữ phổ biến nhất trong lĩnh vực AI. Trong cuốn sách này chúng ta cũng sẽ sử dụng Python.

Để có thể lập trình bằng Python, trước tiên chúng ta sẽ cần cài đặt nó. Phiên bản Python 2 đã quá cũ, vậy nên bạn sẽ cần cài Python 3 theo đường liên kết này: <https://pythontest.com/python/installing-python-3-11/>.

Tiếp theo bạn sẽ cần lựa chọn một công cụ lập trình, ở đây chúng tôi đề nghị bạn sử dụng công cụ Visual Studio Code (VSC), bạn có thể tải xuống tại đây: <https://code.visualstudio.com/> và cài đặt.

Python là một ngôn ngữ thông dịch tương đối dễ học với cú pháp cũng tương đối đơn giản. Chúng ta hãy bắt đầu với chương trình “Hello World”. Đầu tiên bạn hãy tạo ra một thư mục làm việc để lưu trữ các mã nguồn Python, (hãy nói bạn đặt tên nó là AI), sau đó bạn hãy mở VSC và tìm đến thư mục AI, tạo một tập tin có tên *hello-world.py*, sau đó thêm mã nguồn vào tập tin này như sau:

```
print("Hello World")
```

Tiếp theo, bạn có thể di chuyển đến phần trình thực thi (Trên hệ điều hành Mac và Linux là “terminal” và trên Windows là “cmd”) ở phía dưới VSC và gõ lệnh:

```
python3 hello-world.py
```

Bạn sẽ thấy kết quả hiện ra:

```
Hello World
```

Chúc mừng! Bạn đã cài đặt môi trường phát triển Python thành công và chạy được chương trình đầu tiên.

Tiếp theo, chúng ta có thể khởi tạo một hàm tính tổng. Hãy trở lại VSC và tạo một tập tin có tên *sum.py* với mã nguồn như sau:

```
# Định nghĩa hàm tính tổng
def sum(a, b):
    return a + b

# Tính và in ra kết quả
print(sum(1, 2))
```

Hãy thực thi lệnh *python3 sum.py* tại trình thực thi và chúng ta sẽ nhận được kết quả là 3.

Thông qua mã nguồn, chúng ta có thể thấy một hàm của Python được bắt đầu bằng từ khóa *def* và không cần phải quy định kiểu trả về.

Ngoài lập trình hướng hàm thì Python cũng hỗ trợ lập trình hướng đối tượng và chúng ta cũng có thể tạo ra các lớp để “đóng gói” lại các thuộc tính và hàm nếu cần. Giả sử chúng ta cần tạo ra một lớp mô tả một người, có thể tạo một lớp *Person* như sau:

```
class Person:
    def __init__(self, name="", age=0):
        self.name = name
        self.age = age

    def to_string(self):
        return "name: " + self.name + ", age: " +
str(self.age)
```

Một lớp trong Python cũng được bắt đầu bằng từ khóa *class* tương tự như các ngôn ngữ khác, hàm tạo được mặc định là *\_\_init\_\_* với tham số *self* cũng giống như con trỏ *this* trong C++, Java hay C# và các tham số khác có thể mặc định giá trị. Lớp *Person* này có hai thuộc tính là *name* và *age* đại diện cho tên và tuổi, một hàm *to\_string* để trả về thông tin người dùng dưới dạng chuỗi.