|  |  |
| --- | --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG THÔN VÀ PTNT** |
| **TRƯỜNG ĐẠI HỌC THỦY LỢI** | |



LÊ CÔNG MINH

**TÌM HIỂU PHƯƠNG PHÁP LIFELONG MACHINE LEARNING VÀ ỨNG DỤNG CHO BÀI TOÁN PHÁT HIỆN Ý ĐỊNH MUA BÁN**

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2023

|  |  |
| --- | --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG THÔN VÀ PTNT** |
| **TRƯỜNG ĐẠI HỌC THỦY LỢI** | |

LÊ CÔNG MINH

**TÌM HIỂU PHƯƠNG PHÁP LIFELONG MACHINE LEARNING VÀ ỨNG DỤNG CHO BÀI TOÁN PHÁT HIỆN Ý ĐỊNH MUA BÁN**

|  |  |
| --- | --- |
| Ngành: | Công nghệ thông tin |
| Mã số: | 7480401 |

|  |  |
| --- | --- |
| GIẢNG VIÊN HƯỚNG DẪN: | PGS.TS LÊ ĐỨC HẬU |

HÀ NỘI, NĂM 2023

**GÁY BÌA ĐỒ ÁN TỐT NGHIỆP, KHÓA LUẬN TỐT NGHIỆP**

**HỌ VÀ TÊN: LÊ CÔNG MINH ĐỒ ÁN/KL TỐT NGHIỆP HÀ NỘI, NĂM 2023**

**LỜI CAM ĐOAN**

Em tên là Lê Công Minh, em xin cam đoan đây là Đồ án tốt nghiệp / Khóa luận tốt nghiệp do chính bản thân em làm. Các kết quả trong Đồ án tốt nghiệp / Khóa luận tốt nghiệp này mà em làm là trung thực, không sao chép toàn bộ từ bất cứ một nguồn nào, dưới mọi hình thức nào. Việc tham khảo các nguồn tài liệu (nếu có) đã được em thực hiện trích dẫn và ghi nguồn tài liệu tham khảo một cách rõ ràng, đúng quy định.

|  |  |
| --- | --- |
|  | **Tác giả ĐATN / KLTN** |
|  |  |
|  | **Lê Công Minh** |

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn tới các thầy cô trường Đại học Thủy Lợi và những người bạn đã đồng hành cùng em trong suốt quá trình học tập và rèn luyện trong những năm tháng đại học. Em xin chân thành cảm ơn PGS.TS Lê Đức Hậu đã giúp đỡ rất nhiều để em có thể hoàn thành đồ án, giúp em bổ sung kiến thức về bài toán xử lý ngôn ngữ tự nhiên trong lĩnh vực trí tuệ nhân tạo.

Trong quá trình thực hiện đồ án tốt nghiệp do kiến thức còn hạn chế và thiếu kinh nghiệm thực tế nên không thể tránh khỏi những sai sót. Em rất mong nhận được những ý kiến đóng góp của các thầy cô để đồ án được hoàn thiện một cách tốt nhất.

Em xin chân thành cảm ơn !

|  |  |
| --- | --- |
| Description: Logo-WRU | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập - Tự do - Hạnh phúc**  ----------★----------  **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP** |

|  |  |
| --- | --- |
| Họ tên sinh viên: **Lê Công Minh** | Hệ đào tạo: **Đại học chính quy** |
| Lớp: **61TH3** | Ngành: **Công nghệ thông tin** |
| Khoa: **Công nghệ thông tin** |  |

1. TÊN ĐỀ TÀI:

**TÌM HIỂU PHƯƠNG PHÁP LIFELONG MACHINE LEARNING VÀ ỨNG DỤNG CHO BÀI TOÁN PHÁT HIỆN Ý ĐỊNH MUA BÁN**

1. CÁC TÀI LIỆU CƠ BẢN:
2. T. Nguyễn, "Deep Learning cơ bản," 30 March 2019. [Online]. Available: https://nttuan8.com/gioi-thieu-ve-deep-learning/. [Accessed June 2023].
3. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, 1998.
4. J. Brownlee, "Handwritten Digit Recognition using Convolutional Neural in Python with Keras," 27 June 2016. [Online]. Available: <https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/>. [Accessed June 2023].
5. Kasun LLC, Zhou H, Huang GB, Vong CM, Representational learning with extreme learning machine for big data, 2013.
6. Tapson J, de Chazal P, van Schaik A, Explicit computation of input weights in extreme learning machines, 2014.
7. Tapson J, van Schaik A, Learning the pseudoinverse solution to network weights, 2013.
8. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, 2015.
9. L. T. T. HẰNG, NGHIÊN CỨU VỀ MẠNG NEURAL TÍCH CHẬP VÀ ỨNG DỤNG CHO BÀI TOÁN NHẬN DẠNG BIỂN SỐ XE, Hà Nội, 2016.
10. NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN:

|  |  |
| --- | --- |
| **Nội dung các phần** | **Tỷ lệ** |
| Chương 1. Giới thiệu bài toán | 20% |
| Chương 2. Cơ sở lý thuyết | 20% |
| Chương 3. Ứng dụng phương pháp cho bài toán | 50% |
| Chương 4. Thực nghiệm | 10% |
| Chương 5. Thực nghiệm |  |

1. GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN

|  |  |
| --- | --- |
| **Phần** | **Họ tên giáo viên hướng dẫn** |
| Chương 2. Cơ sở lý thuyết | PGS. TS Nguyễn Ngọc Quỳnh Châu |
| Chương 3. Ứng dụng phương pháp cho bài toán | PGS. TS Nguyễn Ngọc Quỳnh Châu |

1. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày .... tháng …. năm 2023.

|  |  |  |
| --- | --- | --- |
| **Trưởng Bộ môn**  *(Ký và ghi rõ Họ tên)* |  | **Giáo viên hướng dẫn chính**  *(Ký và ghi rõ Họ tên)* |

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua.

|  |
| --- |
| Ngày …. tháng …. năm 2023.  **Chủ tịch Hội đồng**  *(Ký và ghi rõ Họ tên)* |

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi.

|  |
| --- |
| Ngày .... tháng …. năm 2023.  **Sinh viên làm Đồ án tốt nghiệp**  *(Ký và ghi rõ Họ tên)* |

|  |  |
| --- | --- |
| logo | TRƯỜNG ĐẠI HỌC THUỶ LỢI  **KHOA CÔNG NGHỆ THÔNG TIN**  BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP |

**Tên đề tài: Tìm hiểu phương pháp LIFELONG MACHINE LEARNING và ứng dụng cho bài toán phát hiện ý định mua bán**

*Sinh viên thực hiện*: **Lê Công Minh**

*Lớp*: **61TH3**

*Mã sinh viên:* **1951060862**

*Số điện thoại:* **0365829001**

*Email:* **1951060862@e.tlu.edu.vn**

*Giáo viên hướng dẫn*: **PGS.TS Lê Đức Hậu**

**TÓM TẮT ĐỀ TÀI**

Hiện nay các phương pháp học máy thông thường hầu hết đều là các phương pháp học có giới hạn: chúng ta đưa đầu vào là một tập dữ liệu huấn luyện, tiếp đó thực hiện giải thuật học máy trên bộ dữ liệu đó để tạo ra mô hình học máy ứng dụng cho thực tế. Những phương pháp học máy này thường bị giới hạn bởi việc không tích lũy tri thức đã học được. Trong những trường hợp cụ thể các phương pháp học máy thông thường có thể đạt được kết quả cao nhưng yêu cầu tập dữ liệu huấn luyện là rất lớn và mục tiêu cho bài toán phải được xác định cụ thể.

Phương pháp học máy LIFELONG MACHINE LEARNING (học cả đời) là một phương pháp học máy nâng cao có thể học liên tục, tích lũy những tri thức học được từ quá khứ để sử dụng cho các nhiệm vụ trong tương lai. Càng học sẽ càng tích lũy được nhiều tri thức, giống với cách học của con người chúng ta. Điều này là rất quan trọng với sự bùng nổ dữ liệu trong thời đại công nghệ hiện nay.

Đồ án này sẽ trình bày những kiến thức cơ bản về LIFELONG MACHINE LEARNING và áp dựng cùng phương pháp Naïve Bayes với cho bài toán phát hiện ý định mua bán thông qua bình luận trên các diễn đàn sử dụng bộ dữ liệu NAACL2013-Chen-Intention cho việc nghiên cứu. Bên cạnh các lý do nêu trên, việc thực hiện đồ án này còn cho phép sinh viên được tự mình tìm hiểu và thực nghiệm, tích lũy tri thức cho con đường sự nghiệp tương lai.

**CÁC MỤC TIÊU CHÍNH**

* Tìm hiểu cơ bản về phương pháp LIFELONG MACHINE LEARNING.
* Áp dụng phương pháp LIFELONG MACHINE LEARNING cho bài toán phát hiện ý định mua bán sử dụng giải thuật Naïve Bayes, Gradient Descent, trích chọn đặc trưng dựa vào độ lợi thông tin và sử dụng độ đo Precision, Recall, F1 cho các thực nghiệm.

**KẾT QUẢ DỰ KIẾN**

* Nắm được các vấn đề cơ bản về phương pháp LIFELONG MACHINE LEARNING.
* Xây dựng chương trình hoàn chỉnh.
* Hoàn thiện luận văn tốt nghiệp.

**TÀI LIỆU THAM KHẢO**

* Cross Domain Intention Detection in Discussion Forums – N.X. Bach, L.C. Linh, T.M. Phuong

<https://www.researchgate.net/publication/321737605_Cross-Domain_Intention_Detection_in_Discussion_Forums/link/5cab5cc0a6fdcca26d06adf6/download>

* Identifying Intention Posts in Discussion Forums – Zhiyuan Chen, Bing Liu, Meichun Hsu, Malu Castellanos, Riddhiman Ghosh

<https://aclanthology.org/N13-1124/>

* LIFELONG MACHINE LEARNING

<https://www.cs.uic.edu/~liub/2016-Lifelong-Machine-Learning-final.pdf>

**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN 1](#_Toc154153905)

[1.1 Lý do chọn đề tài 1](#_Toc154153906)

[1.2 Mục tiêu nghiên cứu. 1](#_Toc154153907)

[1.3 Đối tượng nghiên cứu 2](#_Toc154153908)

[1.4 Phạm vi nghiên cứu 2](#_Toc154153909)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 4](#_Toc154153910)

[2.1 Tổng quan về học máy 4](#_Toc154153911)

[2.2 Tổng quan về phương pháp Lifelong machine learning 5](#_Toc154153912)

[2.2.1 Lịch sử hình thành của phương pháp Lifelong machine learning 5](#_Toc154153913)

[2.2.2 Phương pháp Lifelong machine learning 6](#_Toc154153914)

[2.3 Giải thuật phân loại Naive Bayes (NB) 11](#_Toc154153915)

[2.4 Phương pháp trích chọn đặc trưng Information Gain (IG) 14](#_Toc154153916)

[2.5 Phương pháp Stochastic gradient descent (SGD) 16](#_Toc154153917)

[2.6 Các thang đo đánh giá mô hình học máy 18](#_Toc154153918)

[2.7 Cách áp dụng Lifelong machine learning cho đồ án 18](#_Toc154153919)

[2.7.1 Tổng quan về cách áp dụng 18](#_Toc154153920)

[2.7.2 Chi tiết về cách áp dụng 19](#_Toc154153921)

[CHƯƠNG 3: ỨNG DỤNG PHƯƠNG PHÁP CHO BÀI TOÁN 20](#_Toc154153922)

[3.1 Phân tích bài toán 20](#_Toc154153923)

[3.2 Ngôn ngữ và thư viện 20](#_Toc154153924)

[3.2.1 Ngôn ngữ Python 20](#_Toc154153925)

[3.2.2 Thư viện 22](#_Toc154153926)

[3.3 Mô hình 26](#_Toc154153927)

[3.4 Cơ sở dữ liệu 29](#_Toc154153928)

[3.4.1 Bộ cơ sở dữ liệu MNIST 29](#_Toc154153929)

[3.4.2 Bộ cơ sở dữ liệu A\_Z handwritten 29](#_Toc154153930)

[CHƯƠNG 4: THỰC NGHIỆM 31](#_Toc154153931)

[4.1 Các bước thực hiện 31](#_Toc154153932)

[4.2 Thực nghiệm 32](#_Toc154153933)

[4.2.1 Đưa dữ liệu vào 32](#_Toc154153934)

[4.2.2 Xây dựng mô hình 34](#_Toc154153935)

[4.2.3 Tiền xử lý dữ liệu 36](#_Toc154153936)

[4.2.4 Đánh giá mô hình 40](#_Toc154153937)

[CHƯƠNG 5: Kết luận 48](#_Toc154153938)

[CHƯƠNG 6: TÀI LIỆU THAM KHẢO 50](#_Toc154153939)

**DANH MỤC CÁC HÌNH ẢNH**

[Hình 2.1 Sơ đồ một hệ thống học máy truyền thống [1] 5](#_Toc154093034)

[Hình 2.2 Kiến trúc một hệ thống Lifelong machine learning [1] 7](#_Toc154093035)

[Hình 2.3 Sơ đồ hệ thống Transfer learning [4] 9](#_Toc154093036)

[Hình 2.4 Sơ đồ hệ thống Multi-task learning [6] 10](#_Toc154093037)

[Hình 2.5 Sơ đồ áp dụng phương pháp Lifelong machine learning cho bài toán [9] 17](#_Toc154093038)

# GIỚI THIỆU BÀI TOÁN

## Lý do chọn đề tài

Các phương pháp học máy hiện nay hầu hết là các phương pháp học có giới hạn với các bước: đầu vào là một tập dữ liệu huấn luyện, thực hiện các thuật toán học máy trên bộ dữ liệu đó để tạo ra mô hình phù hợp rồi sử dụng cho mục đích cụ thể. Điều này gây ra việc không tích lũy được tri thức đã học được trong quá trình học. Bởi vậy những phương pháp học máy thông thường nếu muốn đạt được kết quả cao thường sẽ yêu cầu tập dữ liệu huấn luyện rất lớn và chỉ phù hợp cho mục đích cụ thể.

Lifelong machine learning là một phương pháp học máy nâng cao có thể học liên tục dựa trên phương pháp học của con người mà không bị giới hạn như các phương pháp học máy truyền thống. Những tri thức học được trong quả khứ sẽ được duy trì và tích lũy cho các nhiệm vụ học trong tương lai.

Từ những đặc điểm đó, em tin rằng phương pháp Lifelong machine learning sẽ giúp ích rất nhiều cho lĩnh vực trí tuệ nhân tạo trong kỷ nguyên bùng nổ công nghệ hiện nay. Một hệ thống học máy sẽ không thể thực sự thông minh nếu không có sự tích lũy tri thức trong quá trình học.

Hiện nay các bài toán về xử lý ngôn ngữ tự nhiên đang rất được quan tâm vì mỗi ngày có đến hàng nghìn đến vài chục nghìn các bài đăng trên các diễn đàn mạng xã hội. Với các doanh nghiệp thì việc quản lý các bài đăng này và tìm ra những bài có ý định mua bán để đưa ra chiến lược tiếp thị là vô cùng quan trọng. Nếu thực hiện việc đó thủ công sẽ tốn rất nhiều thời gian và công sức. Việc áp dụng phương pháp học máy này sẽ mang lại nhiều lợi ích cho các doanh nghiệp có nhu cầu.

Từ những lý do trên, đồ án này sẽ trình bày những kiến thức cơ bản về phương pháp Lifelong machine learning và áp dụng cho đề tài xử lý ngôn ngữ tự nhiên với bài toán phát hiện ý định mua bán sử dụng giải thuật Naive Bayes.

## Mục tiêu nghiên cứu.

Dưới đây là một số mục tiêu cụ thể mà tôi đặt ra cho đề tài này:

* Mục tiêu 1: Tìm hiểu về phương pháp Lifelong machine learning
* Mục tiêu 2: Áp dụng phương pháp Lifelong machine learning cho đề tài
* Mục tiêu 3: Đánh giá mô hình áp dụng phương pháp này
* Mục tiêu 4: Đưa ra hướng phát triển của mô hình

## Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài "Tìm hiểu phương pháp Lifelong Machine Learning và ứng dụng cho bài toán phát hiện ý định mua bán" gồm các đối tượng sau:

Phương pháp Lifelong machine learning: Đối tượng chính của nghiên cứu là phương pháp Lifelong machine learning với mục tiêu là giới thiệu các thành phần cơ bản của phương pháp này và một số phương pháp học máy khác có liên quan đồng thời đưa ra phương pháp đánh giá một hệ thống học máy Lifelong machine learning.

Bài toán phát hiện ý định mua bán: Đây là một bài toán cụ thể trong lĩnh vực xử lý ngôn ngữ tự nhiên. Mục tiêu là xử lý các bài đăng và đưa ra kết luận có hay không ý định mua bán hàng hóa.

Dữ liệu huấn luyện và kiểm tra: Để phát triển mô hình cho bài toán này, đối tượng nghiên cứu cũng bao gồm các bộ dữ liệu huấn luyện và kiểm tra. Đồ án sử dụng bộ dữ liệu NAACL2013-Chen-Intention để huấn luyện, đánh giá hiệu suất và độ chính xác của mô hình.

## Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài "Tìm hiểu phương pháp Lifelong Machine Learning và ứng dụng cho bài toán phát hiện ý định mua bán" gồm các phạm vi sau:

Nghiên cứu về phương pháp Lifelong machine learning và một số phương pháp phương pháp học máy có liên quan như Transfer learning và Multi-task learning.

Nghiên cứu về các phương pháp xử lý các bài đăng mạng xã hội. Mục tiêu là tìm ra các phương pháp để tiền xử lý dữ liệu chuẩn bị cho quá trình huấn luyện.

Hiện nay có rất nhiều phương pháp trích chọn đặc trưng có thể nói đến như TF-IDF, Bag-of-words, Information Gain,… nhưng với phạm vi của đồ án nhỏ hẹp nên sẽ chỉ trình bày về phương pháp Information Gain để trích chọn đặc trưng.

Về xử lý ngôn ngữ tự nhiên cụ thể là bài toán phân lớp có nhiều phương pháp để thực hiện như SVM, Decision Tree, Naive Bayes,… nhưng với phạm vi nhỏ hẹp của đồ án nên sẽ chỉ trình bày về phương pháp Multinomial Naïve Bayes áp dụng Lifelong Machine Learning để phục vụ cho việc nghiên cứu.

Nghiên cứu về các phương pháp và chỉ số để đánh giá hiệu suất của mô hình nhận diện các bài đăng có ý định mua bán. Mục tiêu là đo lường độ chính xác, độ nhạy, độ đặc hiệu và tìm cách cải thiện mô hình thông qua tối ưu hóa các thuật toán, thay đổi các tham số hay tăng cường tập dữ liệu huấn luyện.

Phạm vi nghiên cứu có thể được điều chỉnh và mở rộng để phù hợp với khả năng và quy mô của đồ án "Tìm hiểu phương pháp Lifelong Machine Learning và ứng dụng cho bài toán phát hiện ý định mua bán".

# 

# CƠ SỞ LÝ THUYẾT

## Tổng quan về học máy

Trong những năm gần đây, lĩnh vực trí tuệ nhân tạo và cụ thể là học máy đang nổi lên rất mạnh mẽ và xuất hiện trong nhiều lĩnh vực cuộc sống của chúng ta. Học máy tập trung vào việc phát triển các thuật toán để máy tính có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình một cách rõ ràng.

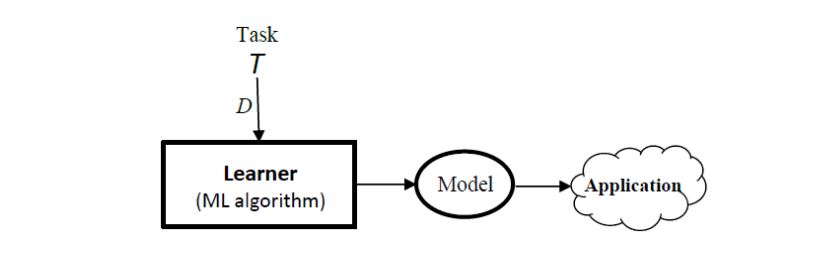
Các phương pháp học máy chủ yếu được chia làm hai loại chính: học có giám sát (Supervised Learning), học không giám sát (Unsupervised learning). Với học có giám sát, máy tính sẽ được huấn luyện từ các dữ liệu đầu vào đã được gán nhãn và dự đoán đầu ra của một dữ liệu mới tương ứng. Học có giám sát còn được tiếp tục chia nhỏ ra thành hai bài toán chính là phân lớp (Classification) và hồi quy (Regression). Một bài toán được xem là phân lớp nếu các nhãn của dữ liệu đầu vào được chia thành một số hữu hạn nhóm, ví dụ như bài toán xác định xem liệu một email có phải là spam hay không. Còn nếu nhãn của các dữ liệu đầu vào không được chia thành các nhóm mà là một giá trị thực cụ thể thì sẽ là bài toán hồi quy, ví dụ như bài toán dự đoán giá nhà.

Trong khi đó với học không giám sát thì chúng ta không biết được đầu ra hay nhãn mà chỉ có dữ liệu đầu vào. Mục tiêu của học không giám sát là dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân cụm (Clustering) hoặc giảm số chiều của dữ liệu (Dimension Reduction) để thuận tiện trong việc lưu trữ và tính toán.

Ngày nay các ứng dụng của học máy đã đạt được nhiều thành công rực rỡ trong các lĩnh vực như xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói, thị giác máy tính và nhiều lĩnh vực khác. Học máy đang đóng vai trò quan trọng trong cuộc sống, mang lại tiềm năng lớn cho việc tự động hóa và tối ưu hóa các nhiệm vụ phức tạp.

Để áp dụng học máy cho một vấn đề cụ thể yêu cầu dữ liệu phải đủ lớn, phải hiểu biết về việc lựa chọn thuật toán, tinh chỉnh tham số, khả năng đánh giá và giải thích kết quả mô hình. Sự thành công của học máy cũng phụ thuộc rất nhiều vào độ chính xác, độ tin cậy và mức độ phức tạp của mô hình.

Tuy nhiên, việc các phương pháp học máy theo kiểu truyền thống chỉ đơn thuần sử dụng một tập dữ liệu huấn luyện có sẵn để sinh ra mô hình học máy thì sẽ không có sự tích lũy tri thức để phục vụ cho các bài toán trong tương lai. Mỗi nhiệm vụ cần học yêu cầu chúng ta phải chuẩn bị tập dữ liệu rất lớn để đảm bảo cho sự chính xác của mô hình, gây ra rất nhiều giới hạn và thường cho ra độ chính xác không đảm bảo. Cho dù thời điểm hiện nay đã có rất nhiều những phương pháp để cải tiến độ chính xác nhưng các phương pháp đó vẫn tập trung thực hiện một nhiệm vụ duy nhất, vẫn gặp nhiều vấn đề vì không tích lũy được tri thức đã học.



Hình 2.1 Sơ đồ một hệ thống học máy truyền thống [1]

Những năm gần đây học máy đã phát triển vượt bậc nhờ vào lĩnh vực mới là học sâu (Deep Learning) mang lại rất nhiều lợi ích trong xử lý những bài toán phức tạp. Nhưng do để phù hợp với nội dung của đồ án, em xin phép không trình bày về phần này.

## Tổng quan về phương pháp Lifelong machine learning

### Lịch sử hình thành của phương pháp Lifelong machine learning

Phương pháp Lifelong machine learning hay Lifelong learning được hai nhà nghiên cứu Thrun và Mitchell đề xuất lần đầu tiên vào năm 1995 để khắc phục nhược điểm của những phương pháp học máy đã nêu ra trong phần 2.1 ở phía trên. Sau đó phương pháp này được tiến hành nghiên cứu trong bốn lĩnh vực chính: Lifelong Supervised Learning, Lifelong Unsupervised Learning, Lifelong Semi-Supervised Learning và Lifelong Reinforcement Learning. Sau đây là phần trình bày về một vài nghiên cứu cụ thể của từng lĩnh vực.

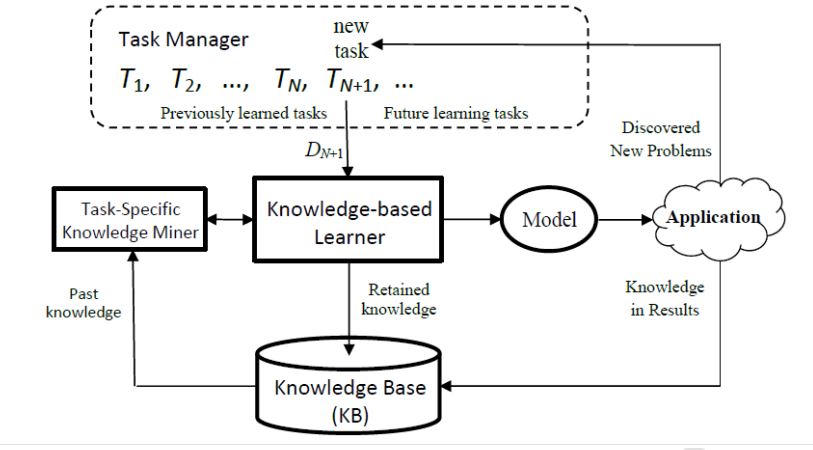
* **Lifelong Supervised Learning**: Vào năm 1996, Thrun lần đầu tiên nghiên cứu phương pháp Lifelong machine learning khi từng nhiệm vụ trong quá khứ trước đó hay nhiệm vụ mới đều nhằm nhận diện một khái niệm hoặc một lớp cụ thể bằng cách sử dụng phân lớp nhị phân (binary classification). Năm 2016, Fei et al [2] đã mở rộng cấu trúc của Lifelong machine learning với mục đích khiến phương pháp này có thể tích lũy kiến thức trong quá trình học, tức là khi gặp một lớp hay một nhiệm vụ mới thì sẽ xây dựng một bộ phân loại mới mà có thể phân loại cả những lớp cũ và mới bằng cách cập nhật lại bộ phân loại cũ dựa vào lớp mới. Năm 2015, Chen et al [3] đã đưa ra phương pháp Lifelong machine learning sử dụng giải thuật phân loại Naïve Bayesian. Năm 2014, lý thuyết về Lifelong machine learning được hoàn thiện bới Pentina và Lampert trong PAC-learning framework. [1]
* **Lifelong Unsupervised Learning**: Những bài báo về lĩnh vực này chủ yếu là về mô hình hóa chủ đề (Topic Modeling) và trích xuất thông tin trong Lifelong machine learning. Trong những năm 2014 và 2016, Zhiyuan Chen, Bing Liu và Shuai Wang đã cung cấp rất nhiều phương pháp về mô hình hóa chủ đề mà kho tri thức được xây dựng từ những nhiệm vụ trong quá khứ và sử dụng để tạo ra những chủ đề tốt hơn trong những nhiệm vụ mới. Cũng trong năm 2016, Lei Shu đã cung cấp phương pháp lifelong relaxation để giải quyết một bài toán phân loại trong chủ đề học không giám sát. [1]
* **Lifelong Semi-Supervised Learning**: Những nghiên cứu trong lĩnh vực này được biểu diễn bởi hệ thống NELL (Never-Ending Language Learner). Hệ thống này liên tục đọc dữ liệu trên Web phục vụ cho công việc trích xuất thông tin từ những năm 2010, hiện tại đã đạt tới hàng triệu thực thể. [1]
* **Lifelong Reinforcement Learning**: Năm 1995, Thurn và Mitchell đã đề xuất một vài phương pháp Lifelong machine learning về việc học cho robot. Năm 1998, Mark B Ring đã cung cấp một phương pháp học liên tục nhằm giải quyết những nhiệm vụ khó bằng cách xử lý những nhiệm vụ dễ trước. Năm 2007, Aaron Wilson cùng các nhà nghiên cứu khác đã cung cấp phương pháp quy trình quyết định Markov (Markov Decision Process). Từ đó đến nay đã có rất nhiều nghiên cứu về lĩnh vực này. [1]

### Phương pháp Lifelong machine learning

#### Định nghĩa và các thành phần cơ bản

Định nghĩa về Lifelong machine learning được Thrun đề cập vào năm 1996 như sau: Một hệ thống đã thực hiện việc học qua các nhiệm vụ từ 1 đến N. Khi đối mặt với việc học nhiệm vụ thứ N + 1, hệ thống sẽ sử dụng tri thức đã được tích lũy từ việc học N nhiệm vụ trước đó để phục vụ cho việc học nhiệm vụ thứ N + 1.

Dựa vào định nghĩa đó, chúng ta có thể nhận ra được ba đặc điểm chính của phương pháp Lifelong machine learning: (1) học một cách liên tục, (2) các tri thức được khai phá từ các nhiệm vụ trong quá khứ sẽ được lưu trữ trong cơ sở tri thức (Knowledge Base), (3) khả năng sử dụng những tri thức đã được tích lũy đó để giúp đỡ các nhiệm vụ cần học trong tương lai. Từ đó, phương pháp Lifelong machine learning sẽ học một chuỗi các nhiệm vụ, có thể sẽ không bao giờ ngừng lại quá trình học và từ đó hệ thống sẽ trở nên ngày càng có nhiều tri thức, giúp việc học trở nên tốt hơn. Những đặc điểm đó khiến cho phương pháp Lifelong machine learning trở nên khác biệt với những phương pháp có vẻ tương đồng như Transfer learning hay Multi-task learning khi nhưng phương pháp đó không có một hay những đặc điểm trên. [1]



Hình 2.2 Kiến trúc một hệ thống Lifelong machine learning [1]

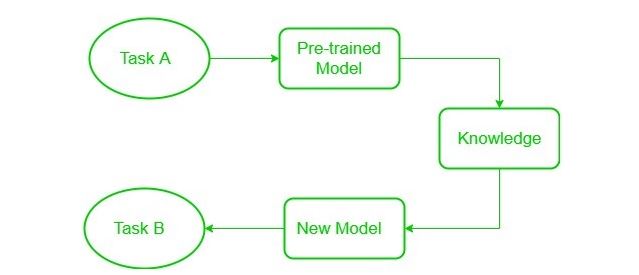
Một hệ thống Lifelong machine learning thường có bốn thành phần chính sau:

1. **Nơi lưu trữ thông tin quá khứ (Past Information Store: PIS):** Đây là nơi lưu trữ những thông tin có được từ những nhiệm vụ học trước đó. Nơi này có thể có nhiều phần nhỏ để lưu trữ những thông tin như dữ liệu gốc, các kết quả trung gian hay các mô hình đã được xây dựng trong quá khứ. [3]
2. **Cơ sở tri thức (Knowledge Base: KB):** Đây là nơi lưu trữ tri thức tích lũy được từ việc khai phá trong PIS. Trong một hệ thống cụ thể, người dùng cần quyết định một cách biểu diễn tri thức sao cho phù hợp với bài toán cần dùng. [3]
3. **Bộ khai phá tri thức (Knowledge Miner: KM)**: Đây là nơi sẽ khai phá tri thức trong PIS và những tri thức có được sẽ được chuyển tới lưu trữ trong KB. [3]
4. **Bộ học dựa trên cơ sở tri thức (Knowledge-Based Learner: KBL):** Dựa vào những tri thức tích lũy được trong KB và những thông tin trong PIS, bộ học này tiến hành việc học những nhiệm vụ mới. [3]

Từ kiến trúc được mô tả như hình 2.2, ta có thể phát biểu cách thức hoạt động của phương pháp Lifelong machine learning như sau: Khi gặp một nhiệm vụ mới TN+1 với bộ dữ liệu DN+1, Task Manager (TM) sẽ đưa nhiệm vụ đó tới Knowledge-based Learner (KBL) . KBL sẽ thực hiện việc học nhiệm vụ mới TN+1 với sự trợ giúp từ tri thức đã được tích lũy từ các nhiệm vụ T1 đến TN trước đó trong Knowledge Base (KB) sẽ tạo đầu ra là một Model để sử dụng cho các ứng dụng thực tế. Đồng thời các tri thức mới vừa được học sẽ được chuyển tới KB để phục vụ cho các nhiệm vụ khác sau này. Ứng dụng được áp dụng Model đó có thể tìm ra các nhiệm vụ mới và đồng thời tiếp tục cung cấp những tri thức được khai phá cho KB để nâng cao chất lượng Model.

#### Các phương pháp học máy tương tự khác

Như đã đề cập phía trên, một hệ thống Lifelong machine learning gồm ba đặc điểm chính là khả năng học liên tục, tri thức được tích lũy, khả năng áp dụng những tri thức đó cho các nhiệm vụ trong tương lai. Hiện nay có khá nhiều những phương pháp học máy khác có những đặc điểm tương tự như vậy. Sau đây em xin trình bày về hai phương pháp có nét tương đồng nhất là Transfer learning và Multi-task learning.



Hình 2.3 Sơ đồ hệ thống Transfer learning [4]

Transfer learning là một chủ đề nổi bật trong lĩnh vực học máy và khai phá dữ liệu trong những năm gần đây. Phương pháp này còn được biết đến như là thích ứng miền (Domain adaptation) trong lĩnh vực xử lý ngôn ngữ tự nhiên. [1]

Phương pháp Transfer learning sử dụng source domain đã được gán nhãn để phục vụ cho quá trình học target domain. Không giống như Lifelong machine learning, Transfer learning thì học không liên tục và không có sự lưu giữ tri thức đã được học. Không chỉ vậy, source domain phải có sự tương tự như target domain thì mới áp dụng được Transfer learning. Luồng hoạt động chỉ là một chiều vì chỉ có source domain hỗ trợ cho target domain và không có chiều ngược lại. [5]



Hình 2.4 Sơ đồ hệ thống Multi-task learning [6]

Multi-task learning là phương pháp có thể học nhiều nhiệm vụ cùng một thời điểm. Mặc dù phương pháp này có khả năng học liên tục nhưng lại không có sự tích lũy tri thức trong quá trình học. Điều này gây ra sự khó khăn khi gặp phải nhiệm vụ mới nếu số lượng nhiệm vụ là rất lớn vì không có tri thức tích lũy được. [5]

Với những so sánh đó, chúng ta có thể nhận thấy rằng phương pháp Lifelong machine learning mang những đặc điểm tốt hơn so với hai phương pháp kể trên. Hiện này còn có rất nhiều những phương pháp khác mang tính chất tương tự nhưng để phù hợp với nội dung của đồ án, em xin không trình bày thêm về phần này.

#### Phương pháp đánh giá hệ thống

Khi tiến hành đánh giá những phương pháp học máy truyền thống, chúng ta đánh giá dựa vào quá trình huấn luyện và kiểm thử dữ liệu được lấy ở cùng một miền. Với Lifelong machine learning, chúng ta cần sử dụng một phương pháp đánh giá khác bởi nó thực hiện việc hoc trên nhiều miền khác nhau và cần phải đánh giá kết quả của cách học đó. Thông thường các bước đánh giá phương pháp Lifelong machine learning được thực hiện như sau:

1. ***Chạy trên dữ liệu từ những nhiệm vụ trong quá khứ:*** Đầu tiên chúng ta tiến hành chạy giải thuật học máy trên tập dữ liệu từ một tập các nhiệm vụ đã được học từ trước và tri thức thu được sẽ được lưu trữ trong KB. Tùy theo từng giải thuật khác nhau chúng ta sẽ có các phương pháp lưu trữ tri thức khác nhau. [1]
2. ***Chạy trên dữ liệu từ nhiệm vụ mới:*** Tiếp theo chúng ta tiến hành chạy giải thuật học máy trên dữ liệu từ nhiệm vụ mới bằng cách sử dụng tri thức tích lũy được trong KB. [1]
3. ***Chạy giải thuật thực nghiệm:*** Để tiến hành so sánh, chúng ta sẽ tiến hành chạy một vài giải thuật thực nghiệm. Thông thường sẽ có hai loại thực nghiệm. Loại đầu tiên là sử dụng thuật toán theo phương pháp truyền thống trên dữ liệu mới mà không áp dụng những tri thức tích lũy được trong quá khứ. Thực nghiệm thứ hai là sử dụng giải thuật Lifelong machine learning mà chúng ta đã tạo ra. [1]
4. ***Phân tích kết quả:*** Chúng ta tiến hành so sánh kết quả từ bước hai và bước ba. [1]

#### Những khó khăn của phương pháp

Lifelong machine learning là phương pháp mang lại nhiều thách thức như cần phải lựa chọn tri thức nào phù hợp để giữ lại, làm sao để sử dụng những tri thức đã tích lũy được cũng như làm sao để duy trì KB. Sau đầy là 2 thách thức tiềm ẩn nhưng rất quan trọng trong việc sử dụng phương pháp này:

* *Tính chính xác của tri thức*: Việc sử dụng tri thức không chính xác sẽ gây ra những hậu quả nghiêm trọng cho việc học các nhiệm vụ mới. Những tri thức được tích lũy không chính xác từ nhiệm vụ mới sẽ càng khiến cho hệ thống gặp nhiều lỗi hơn. [1]
* Khả năng ứng dụng tri thức: Có thể một phần của tri thức áp dụng chính xác cho một vài nhiệm vụ trong quá khứ nhưng không có nghĩa nó sẽ tương thích với các nhiệm vụ trong tương lai. Nếu nhiệm vụ mới được áp dụng những tri thức không phù hợp thì cũng sẽ gây ra hậu quả như trường hợp kể trên. [1]

## Giải thuật phân loại Naive Bayes (NB)

Naive Bayes là một phương pháp đơn giản phân loại dựa vào xác suất. Phương pháp này được sử dụng rộng rãi trong lĩnh vực học máy, phổ biến trong nhiều lĩnh vực như các công cụ tìm kiếm, các bộ lọc mail nói riêng và phân loại văn bản nói chung. Ý tưởng cơ bản của cách tiếp cận Naive Bayes là sử dụng xác suất có điều kiện giữa từ và chủ đề để dự đoán xác suất chủ đề của một văn bản cần phân loại. Ưu điểm của giải thuật này là dễ cài đặt và thời gian huấn luyện mô hình nhanh.

Giải thuật này sử dụng tới định lý Bayes nên tiếp theo đây sẽ nêu ra ngắn gọn công thức của định lý Bayes. Công thức đó được phát biểu như sau:

|  |  |
| --- | --- |
|  | (2‑1) |

Trong đó:

* P(Y|X): Xác suất có điều kiện sự kiện Y xảy ra nếu biết rằng sự kiện X đã xảy ra. [7]
* P(X|Y): Xác suất có điều kiện sự kiện X xảy ra nếu biết rằng sự kiện Y đã xảy ra. [7]
* P(Y): Xác suất sự kiện Y xảy ra. [7]
* P(X): Xác suất sự kiện X xảy ra. [7]

Xét bài toán phân lớp có C lớp 1,2,…,C. Giả sử có một điểm dữ liệu x được biểu diễn dưới dạng một vector có d chiều (x Rd). Khi đó xác suất để điểm dữ liệu này rơi vào lớp c được tính bằng: [8]

|  |  |
| --- | --- |
|  | (2‑2) |

Biểu thức này, nếu tính được, sẽ giúp chúng ta xác định được xác suất để điểm dữ liệu rơi vào mỗi class. Từ đó có thể giúp xác định class của điểm dữ liệu đó bằng cách chọn ra class có xác suất cao nhất: [8]

|  |  |
| --- | --- |
|  | (2‑3) |

Biểu thức (2‑3) thường khó tính được trực tiếp, thay vào đó quy tắc Bayes (2‑1) thường được sử dụng nên công thức này sẽ được thay bằng: [8]

|  |  |
| --- | --- |
|  | (2‑4) |

Vì mẫu số P(x) trong công thức (2‑4) không phụ thuộc vào c nên ta có thể rút gọn công thức đó thành: [8]

|  |  |
| --- | --- |
|  | (2‑5) |

Lúc này, chúng ta cần tính hai giá trị là P(x|c) và P(c). Với P(c) chúng ta có thể tính theo công thức:

|  |  |
| --- | --- |
|  | (2‑6) |

Trong đó Nc là số lượng dữ liệu trong tập dữ liệu huấn luyện có nhãn là c và N là tổng số dữ liệu trong tập dữ liệu huấn luyện.

Với P(x|c) tức phân phối của các điểm dữ liệu trong class c thường rất khó tính toán vì x là một biến ngẫu nhiên nhiều chiều, cần rất rất nhiều dữ liệu training để có thể xây dựng được phân phối đó. Để giúp cho việc tính toán được đơn giản, người ta thường giả sử một cách đơn giản nhất rằng các thành phần của biến ngẫu nhiên x là [độc lập với nhau](https://machinelearningcoban.com/2017/07/09/prob/#-independence), nếu biết c. Tức là: [8]

|  |  |
| --- | --- |
|  | (2‑7) |

Sau khi xác định được P(c) và P(x|c) thì lúc này công thức (2‑4) sẽ được phát biểu thành:

|  |  |
| --- | --- |
|  | (2‑8) |

Lúc này chúng ta cần xác định P(xi|c) để tìm ra điểm dữ liệu đầu vào sẽ thuộc vào lớp nào. Việc tính toán P(xi|c) phụ thuộc vào kiểu dữ liệu và có ba loại được sử dụng phổ biến là Gaussian Naive Bayes, Multinomial Naive Bayes và Bernoulli Naive Bayes. Mô hình Gaussian Naive Bayes được sử dụng chủ yếu trong loại dữ liệu mà các thành phần là các biến liên tục, Multinomial Naive Bayes được sử dụng chủ yếu trong phân loại văn bản còn Bernoulli Naive Bayes được áp dụng cho các dữ liệu mà mỗi thành phần là một giá trị binary (bằng 0 hoặc bằng 1). [8] Do đó để phù hợp với nội dung của đồ án là phân loại những bài đăng mạng xã hội thì em sử dụng Multinomail Naive Bayes. Khi đó P(xi|c) sẽ được tính như sau:

|  |  |
| --- | --- |
|  | (2‑9) |

Trong đó Nci là tổng số lần xuất hiện đặc trưng i trong các văn bản của lớp c và Nc là tổng số từ xuất hiện trong lớp c. Cách tính này có một hạn chế là nếu gặp một từ chưa bao giờ xuất hiện trong lớp c bất kì dẫn đến P(xi|c) = 0. Kéo theo kết quả cần tìm của biểu thức (2‑8) sẽ luôn có giá trị bằng 0, điều này làm cho kết quả là không chính xác. Để giải quyết trường hợp này một kĩ thuật được gọi là Laplace smoothing được áp dụng như sau:

|  |  |
| --- | --- |
|  | (2‑10) |

Trong đó k là hằng số được thêm vào để tránh xác suất bằng 0 và thường được gán bằng 1. |V| là số lượng từ vựng khác nhau có trong các bộ dữ liệu huấn luyện.

Lúc này vẫn còn một vấn đề với biểu thức (2‑8) khi d lớn thì vế phải của biểu thức sẽ là một số rất nhỏ và gần với giá trị 0 khiến cho việc tính toán sẽ gặp sai số. Để giải quyết vấn đề này thì biểu thức được tính lại bằng cách lấy log của vế phải, điều này không gây ảnh hưởng tới kết quả bởi log là một hàm đồng biến trên tập các số dương. Lúc này biểu thức sẽ được phát biểu thành: [8]

|  |  |
| --- | --- |
|  | (2‑11) |

## Phương pháp trích chọn đặc trưng Information Gain (IG)

Trong thời điểm hiện nay dữ liệu mà chúng ta thu thập thường ở dạng thô và đến từ nhiều nguồn khác nhau trên mạng internet. Khi đó thì dữ liệu chỉ có một số lượng đặc trưng nhất định là có ích cho bài toán còn phần lớn là thông tin nhiễu gây ra những ảnh hưởng không tốt tới bài toán. Dữ liệu này cần phải được xử lý để chọn ra những đặc trưng tốt thì những phương pháp học máy chúng ta xây dựng mới đạt được hiệu quả cao. Với dữ liệu là văn bản như trong đồ án, có thể tồn tại nhiều dạng khác nhau như chữ cái thường, in hoa, dấu câu hay những kí tự đặc biệt.

Có rất nhiều những phương pháp trích chọn đặc trưng như Bag-of-words, TF-IDF, Word2vec, Information Gain hay Skip-gram được sử dụng trong xử lý văn bản. Với phạm vi của đồ án, bài toán phát hiện ý định trong văng bản thường sẽ chỉ có một hoặc hai câu thể hiện ý định còn phần lớn sẽ không liên quan trực tiếp đến việc thể hiện ý định. Bởi vậy trích chọn đặc trưng là rất quan trọng với đồ án này để giảm đi những thông tin nhiễu. Để phù hợp với phạm vi nhỏ hẹp của đồ án này, em sẽ sử dụng phương pháp IG để tìm ra độ lợi thông tin.

Với mỗi đặc trưng f thì độ lợi thông tin sẽ được tính như sau: [9]

|  |  |
| --- | --- |
|  | (2‑12) |

Trong đó không phụ thuộc vào đặc trưng f nên công thức (2‑12) sẽ được rút ngọn như sau: [9]

|  |  |
| --- | --- |
|  | (2‑13) |

Với bài toán chỉ có hai nhãn là positive (+) và negative (-), công thức (2‑13) sẽ được biến đổi thành: [9]

|  |  |
| --- | --- |
|  | (2‑14) |

Trong đó:

* : Xác suất của một mẫu chứa đặc trưng f. [9]
* : Xác suất của một mẫu không chứa đặc trưng f. [9]
* và : Xác suất của một mẫu mà có nhãn là positive (+) và negative (-) khi có chứa đặc trưng f. [9]
* và : Xác suất của một mẫu mà có nhãn là positive (+) và negative (-) khi không chứa đặc trưng f. [9]

Để dễ dàng cho việc thực hiện tính toán, em gọi một số giá trị như sau:

* Df : Số lượng mẫu chứa đặc trưng f trong bộ dữ liệu huấn luyện
* D: Tổng số mẫu của tập giá trị huấn luyện
* và : Số lượng mẫu có nhãn là positive (+) và negative (-) mà chứa đặc trưng f trong tập dữ liệu huấn luyện
* và : Số lượng mẫu có nhãn là positive (+) và negative (-) mà không chứa đặc trưng f trong tập dữ liệu huấn luyện

Khi đó ta sẽ có những công thức sau:

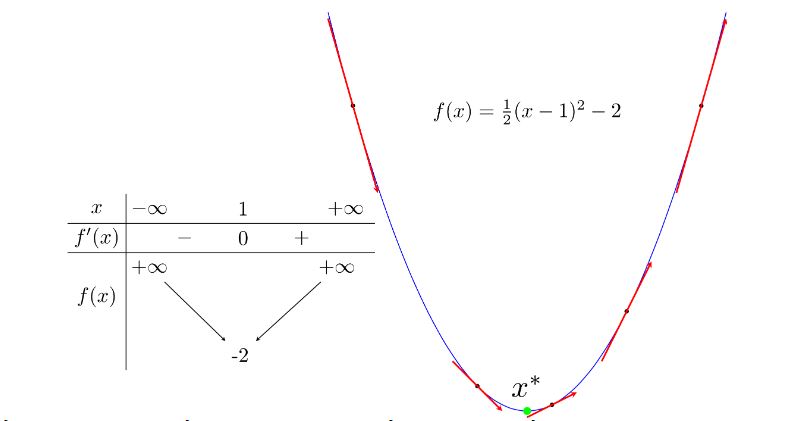
|  |  |
| --- | --- |
|  | (2‑15) |
|  | (2‑16) |
|  | (2‑17) |
|  | (2‑18) |
|  | (2‑19) |
|  | (2‑20) | |

Lúc này khi đã tính được IG cho từng đặc trưng của bộ dữ liệu huấn luyện, ta tiến hành sắp xếp các giá trị IG đó theo thứ tự giảm dần và chọn những đặc trưng có IG cao nhất mà chúng ta cần thiết.

## Phương pháp Stochastic gradient descent (SGD)

Trong học máy nói riêng và toán tối ưu nói chung, chúng ta thường xuyên phải tìm giá trị nhỏ nhất (hoặc đôi khi là lớn nhất) của một hàm số nào đó. Nhìn chung, việc tìm giá trị nhỏ nhất (global minimum) của các hàm mất mát trong học máy là rất phức tạp nên thay vào đó người ta thường cố gắng tìm các điểm cực tiểu địa phương (local minimum) và ở một mức độ nào đó, coi đó là nghiệm cần tìm của bài toán. [10] Khi cực tiểu địa phương chính là giá trị nhỏ nhất của hàm số thì đó là trường hợp tốt nhất mà chúng ta hướng đến.

Các điểm cực tiểu địa phương là nghiệm của phương trình đạo hàm bằng 0. Nếu bằng một cách nào đó có thể tìm được toàn bộ (hữu hạn) các điểm cực tiểu, ta chỉ cần thay từng điểm cực tiểu địa phương đó vào hàm số rồi tìm điểm làm cho hàm có giá trị nhỏ nhất. Tuy nhiên, trong hầu hết các trường hợp, việc giải phương trình đạo hàm bằng 0 là bất khả thi. Nguyên nhân có thể đến từ sự phức tạp của dạng của đạo hàm, từ việc các điểm dữ liệu có số chiều lớn, hoặc từ việc có quá nhiều điểm dữ liệu. [10]



Hình 2.5 Minh họa về điểm cực tiểu [10]

Hướng tiếp cận phổ biến nhất là xuất phát từ một điểm mà chúng ta coi là gần với nghiệm của bài toán, sau đó dùng một phép toán lặp để tiến dần đến điểm cần tìm, tức đến khi đạo hàm gần với 0. Gradient Descent (GD) và các biến thể của nó là một trong những phương pháp được dùng nhiều nhất. [10]

Trước tiên chúng ta sẽ xem xét trường hợp hàm số chỉ phụ thuộc vào một biến duy nhất với quan sát hình 2.5. Giả sử xt là điểm ta tìm được sau vòng lặp thứ t. Ta cần tìm một thuật toán để đưa xt về càng gần x\* càng tốt. [10]

Chúng ta có thể quan sát được như sau:

Nếu đạo hàm của hàm số tại xt: f′(xt) > 0 thì xt nằm về bên phải so với x\* (và ngược lại). Để điểm tiếp theo xt+1 gần với x\* hơn, chúng ta cần di chuyển xt về phía bên trái, tức về phía âm. Nói các khác, **chúng ta cần di chuyển ngược dấu với đạo hàm**: [10]

|  |  |
| --- | --- |
|  | (2‑21) |

Trong đó là một đại lượng ngược dấu với đạo hàm f′(xt). [10]

Nếu xt càng xa x\* về phía bên phải thì f′(xt) càng lớn hơn 0 (và ngược lại). Vậy, lượng di chuyển Δ, một cách trực quan nhất, là tỉ lệ thuận với −f′(xt). [10]

Với những nhận xét đó chúng ta có thể viết lại công thức (2‑21) như sau:

|  |  |
| --- | --- |
|  | (2‑22) |

Trong đó  là một số dương được gọi là tốc độ học (learning rate). Dấu trừ thể hiện việc chúng ta phải đi ngược với đạo hàm (Đây cũng chính là lý do phương pháp này được gọi là Gradient Descent - descent nghĩa là đi ngược). Các quan sát đơn giản phía trên, mặc dù không phải đúng cho tất cả các bài toán, là nền tảng cho rất nhiều phương pháp tối ưu nói chung và thuật toán Machine Learning nói riêng. [10]

Việc lựa chọn tốc độ học là rất quan trọng trong các bài toán thực tế. Việc lựa chọn giá trị này phụ thuộc vào từng bài toán cụ thể và phải làm một vài thí nghiệm để chọn ra giá trị tốt nhất. Nếu tốc độ học quá nhỏ thì tốc độ hội tụ sẽ chậm, ảnh hưởng tới tốc độ của phương pháp học máy và thậm chí không thể tới được đích. Ngược lại nếu tốc độ học quá lớn thì có thể thuật toán sẽ nhanh hơn nhưng cũng có thể không hội tụ được nếu bước nhảy là quá lớn gây ra sự quẩn quanh ở đích. [10]

Với trường hợp sử dụng GD cho hàm nhiều biến, chúng ta có thể tổng quát như sau đây. Giả sử ta cần tìm cực tiểu địa phương cho hàm f() trong đó (theta) là một vector, thường được dùng để ký hiệu tập hợp các tham số của một mô hình cần tối ưu. Đạo hàm của hàm số đó tại một điểm  bất kỳ được ký hiệu là . Tương tự như hàm một biến, thuật toán GD cho hàm nhiều biến cũng bắt đầu bằng một điểm dự đoán , sau đó, ở vòng lặp thứ t, quy tắc cập nhật là: [10]

|  |  |
| --- | --- |
|  | (2‑23) |

Trong đó là đạo hàm của hàm mất mát tại .

Thuật toán GD có rất nhiều biến thể giúp gia tăng tốc độ chạy hay sự hội tụ của thuật toán của thuật toán như Batch Gradient Descent, Mini-batch Gradient Descent hay Stochastic gradient descent. Ở đây em sử dụng Stochastic gradient descent (SGD) vì sự đơn giản áp dụng cho đồ án.

Khác với 2 thuật toán biến thể của GD nêu ở trên thì SGD chỉ thực hiện tính đạo hàm của hàm mất mát dựa trên chỉ một điểm dữ liệu xi rồi cập nhật θ dựa trên đạo hàm này. Việc này được thực hiện với từng điểm trên toàn bộ dữ liệu, sau đó lặp lại quá trình trên. Thuật toán rất đơn giản này trên thực tế lại làm việc rất hiệu quả. [10]

Mỗi lần duyệt một lượt qua tất cả các điểm trên toàn bộ dữ liệu được gọi là một epoch. Với GD thông thường thì mỗi epoch ứng với 1 lần cập nhật θ còn với SGD thì mỗi epoch ứng với N lần cập nhật θ với N là số điểm dữ liệu. Nhìn vào một mặt, việc cập nhật từng điểm một như thế này có thể làm giảm đi tốc độ thực hiện 1 epoch. Nhưng nhìn vào một mặt khác, SGD chỉ yêu cầu một lượng epoch rất nhỏ (thường là 10 cho lần đầu tiên, sau đó khi có dữ liệu mới thì chỉ cần chạy dưới một epoch là đã có nghiệm tốt). Vì vậy SGD phù hợp với các bài toán có lượng cơ sở dữ liệu lớn và cần tốc độ cao. [10]

Một điểm cần lưu ý đó là: sau mỗi epoch, chúng ta cần shuffle (xáo trộn) thứ tự của các dữ liệu để đảm bảo tính ngẫu nhiên. Việc này cũng ảnh hưởng tới hiệu năng của SGD. [10]

Một cách toán học, quy tắc cập nhật của SGD là:

|  |  |
| --- | --- |
|  | (2‑24) |

Trong đó là hàm mất mát với chỉ 1 điểm dữ liệu (input,label) là (xi;yi). [10]

Điều kiện dừng của SGD có thể được áp dụng theo nhiều cách như giới hạn số vòng lặp, so sánh giá trị đạo hàm giữa hai lần liên tiếp nếu độ chênh lệch ở một ngưỡng chấp nhận được hay so sánh giá trị hàm mất mát giữa hai lần liên tiếp cũng tương tự như vậy.

## Các thang đo đánh giá mô hình học máy

Khi chúng ta đã xây dựng một mô hình học máy và huấn luyện nó trên một tập dữ liệu, điều tiếp theo cần làm là đánh giá hiệu năng của mô hình trên tập dữ liệu mới. Việc đánh giá mô hình giúp chúng ta trả lời những câu hỏi như liệu mô hình đã được huấn luyện thành công hay chưa và mức độ thành công của mô hình tốt đến đâu. [11]

Đánh giá một mô hình có tốt hay không thường được thực hiện trên dữ liệu mà mô hình chưa được huấn luyện. Tỷ lệ thường thấy của một tập dữ liệu huấn luyện so với tập dữ liệu thử nghiệm là 70% và 30%. [11]

Sau đây em xin trình bày về 4 thang đo được sử dụng rỗng rãi trong lĩnh vực học máy cụ thể là trong bài toán phân lớp là Accuracy, Precision, Recall và F1.

Khi thực hiện bài toán phân loại, có 4 trường hợp của dự đoán có thể xảy ra:

* True Positive (TP): đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Positive (dự đoán đúng). [11]
* True Negative (TN): đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Negative (dự đoán đúng). [11]
* False Positive (FP): đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Positive (dự đoán sai). [11]
* False Negative (FN): đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Negative (dự đoán sai). [11]

Thang đo Accuracy được định nghĩa là tỷ lệ phần trăm dự đoán đúng cho dữ liệu thử nghiệm. Nó có thể được tính toán dễ dàng bằng cách chia số lần dự đoán đúng cho tổng số lần dự đoán. [11]

|  |  |
| --- | --- |
|  | (2‑25) |

Trong đó Ntrue là số lần dự đoán đúng và N là tổng số dữ liệu trong bộ dữ liệu kiểm thử.

Thang đo Precision được định nghĩa là phần nhỏ của các ví dụ có liên quan (tích cực thực sự) trong số tất cả các ví dụ được dự đoán là thuộc một lớp nhất định.

|  |  |
| --- | --- |
|  | (2‑26) |

Thang đo Recall được định nghĩa là phần nhỏ của các ví dụ được dự đoán thuộc về một lớp so với tất cả các ví dụ thực sự thuộc về lớp đó.

|  |  |
| --- | --- |
|  | (2‑27) |

Precision và Recall rất hữu ích trong trường hợp các lớp không được phân bổ đồng đều. Ví dụ như phát triển một thuật toán phân loại dự đoán xem ai đó có mắc bệnh hay không. Giả sử, tỷ lệ dân số mắc bệnh là 1%, công cụ dự đoán đưa ra kết quả là tất cả mọi người đều không mắc bệnh. Vậy công cụ vẫn đạt tỷ lệ chính xác là 99%. Tuy nhiên, chúng ta sẽ không tìm được ai là người mắc bệnh. Vậy độ hữu ích của công cụ là 0%. [11]

Vì vậy, điều quan trọng là phải đánh giá cả Precision và Recall của một mô hình để xác định ra được người bị bệnh hay không mà không có nhầm lẫn. Chúng ta có công thức tính F1 kết hợp giữa Precision và Recall được thể hiện như sau:

|  |  |
| --- | --- |
|  | (2‑28) |

Thang đo F1 có giá trị lớn khi cả 2 giá trị Precision và Recall đều lớn. Ngược lại, chỉ cần 1 giá trị nhỏ sẽ làm cho thang đo F1 nhỏ.

## Cách áp dụng Lifelong machine learning cho đồ án

### Tổng quan về cách áp dụng

Với bài toán phát hiện ý định mua bán được giới thiệu chung tại chương 1, để phù hợp với mục tiêu để ra thì đồ án này sẽ cung cấp phương pháp Lifelong machine learning áp dụng cùng giải thuật phân loại Naive Bayes với mục đích tìm ra những bài đăng có mang ý định mua bán hay không. Với ý tưởng sử dụng những tri thức đã được tích lũy trong quá khứ để giúp đỡ cho các nhiệm vụ học sau này của phương pháp Lifelong machine learning, các bước áp dụng cho bài toán sẽ được trình bày như sau:



Hình 2.6 Sơ đồ áp dụng phương pháp Lifelong machine learning cho bài toán [9]

Phương pháp này được chia làm ba phần chính:

* **Trộn dữ liệu:** Trong phần này sẽ tiến hành trích xuất tri thức từ nhiều source domain và lưu trữ trong KB. [9]
* **Tối ưu hóa:** Phần này sẽ sử dụng tri thức tích lũy được trong KB để tối ưu hóa những tham số sẽ sử dụng trong mô hình phân loại. [9]
* **Phân loại:** Phần này sẽ sử dụng những tham số đã được tối ưu để xây dựng giải thuật phân loại Naive Bayes. [9]

Từ sơ đồ áp dụng hình 2.5, chúng ta có thể phát biểu cụ thể các bước tiến hành như sau: Với các dữ liệu từ các source domain, chúng ta tiến hành trộn các dữ liệu đó và thực hiện việc trích chọn tri thức để lưu vào KB. Bước tiếp theo từ dữ liệu target domain của nhiệm vụ học mới, ta kết hợp cùng những tri thức đã được tích lũy trong KB rồi thực hiện tối ưu hóa sử dụng Stochastic Gradient Descent (SGD). Chúng ta sử dụng SGD bởi khi tiến hành trộn dữ liệu có thể xảy ra việc nhiễu dữ liệu. Tiếp đó chúng ta tiến hành chọn ra các đặc trưng quan trọng sau khi đã tối ưu bằng SGD để phục vụ cho việc phân loại sử dụng giải thuật Naive Bayes tạo ra mô hình.

### Chi tiết về cách áp dụng

#### Các thành phần trong bài toán

Dựa vào những thông tin đã được đề cập trong phần 2.2.2.1, các thành phần để áp dụng phương pháp Lifelong machine learning cho đồ án sẽ được phát biểu như sau:

* **PIS**: Với từng nhiệm vụ đã học trong quá khứ, chúng ta tiến hành lưu trữ 2 phần:

1. và cho mỗi từ w có trong nhiệm vụ NB . [3]
2. Số lần xuất hiện của từ w trong các dữ liệu positive (+) và số lần xuất hiện của từ w trong các dữ liệu negative (-) . trong mỗi nhiệm vụ . [3]

* **KB**: Cơ sở tri thức sẽ lưu trữ số lần từ w xuất hiện trong các dữ liệu mang nhãn positive (+) và negative (-) của tất cả các nhiệm vụ trước đó: [3]

|  |  |  |
| --- | --- | --- |
|  | (2‑29) | |
|  | | (2‑30) |

* **KM:** Bộ khai phá tri thức ở đây sẽ tiến hành tổng hợp số lượng những từ w được đề cập phía trên theo 2 công thức (2‑29), (2‑30) và được lưu trữ trong KB. [3]
* **KBL**: Bộ học dựa trên cơ sở tri thức sẽ tiến hành thực hiện việc học những nhiệm vụ tiếp theo nhờ những tri thức và thông tin thu được. [3]

#### Thực hiện tối ưu hóa

# ỨNG DỤNG PHƯƠNG PHÁP CHO BÀI TOÁN

## Phân tích bài toán

Trong những năm gần đây, đã có nhiều tiến bộ đáng kể trong lĩnh vực xử lý ảnh. Các bài toán nhận dạng đang được áp dụng rộng rãi trong thực tế tập trung vào nhận dạng mẫu, nhận dạng tiếng nói và nhận dạng chữ viết. Trong số đó, việc nhận dạng chữ viết tay là một bài toán được quan tâm đặc biệt, vì nó đáp ứng nhu cầu trong nhiều ứng dụng thực tế. Công nghệ nhận dạng chữ viết tay đã và đang được áp dụng trong đời sống hàng ngày, như tự động hóa việc đọc tài liệu, tăng tốc độ và hiệu quả khi nhập thông tin vào máy tính. Nhận dạng chữ viết tay cũng có thể hỗ trợ cho các ứng dụng đọc và xử lý chứng từ, hóa đơn, phiếu ghi và các bản viết tay khác.

Nhận diện chữ viết tay là một bài toán trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo, có mục tiêu nhận diện và phân loại các ký tự chữ viết tay từ hình ảnh đầu vào. Mục đích là xây dựng một hệ thống tự động có khả năng nhận biết các ký tự viết tay và gán nhãn chính xác cho chúng.

Chính vì vậy bài toán nhận diện chữ viết tay là rất cần thiết trong cuộc sống hiện tại và tương lai.

## Ngôn ngữ và thư viện

### Ngôn ngữ Python

Python là một ngôn ngữ lập trình hướng đối tượng bậc cao, mạnh mẽ được đưa ra bởi Guido van Rossum. Nó rất dễ học và nhanh chóng trở thành một trong những ngôn ngữ lập trình nhập môn tối ưu nhất cho người mới bắt đầu. Python được sử dụng bộ nhớ cấp pháp động và tạo kiểu động. Python là ngôn ngữ lập trình bậc cao nhưng lại có cách tiếp cận đơn giản, hiệu quả để lập trình hướng đối tượng. Với cú pháp đơn giản của python mà vẫn linh hoạt, rõ ràng, dễ hiểu để viết kịch bản và phát triển ứng dụng trong nhiều ngành nghề, lĩnh vực của cuộc sống.



Hình 20.3.2.1. Ngôn ngữ Python

Python là một ngôn ngữ lập trình nổi bật với các tính năng:

* Ngôn ngữ bậc cao nhưng lại đơn giản và dễ hiểu: Python với cú pháp đơn giản và gọn gàng. Nó dễ học và lập trình hơn nhiều so với nhiều ngôn ngữ lập trình khác như C++, Java, C#. Ngôn ngữ python giúp cho việc lập trình trở nên thú vị, cho phép người lập trình có những giải pháp tốt hơn trong các lĩnh vực của cuộc sống.
* Mã nguồn mở là miễn phí: Ngôn ngữ được sử dụng tự do không tính phí, ngay cả cho mục đích thương mại. Bởi vì nó là mã nguồn mở, nên có thể sửa đổi một cách đơn giản. Ngôn ngữ lập trình python có một cộng đồng lớn đang liên tục được cải thiện sau mỗi bản cập nhật.
* Khả năng lập trình trên nhiều hệ điều hành: Các mã chương trình python có thể di chuyển từ nền tảng này sang nền tảng khác mà không thay đổi cấu trúc của chương trình. Nó chạy được hầu hết trên các hệ điều hành phổ biến như Windows, macOs, Linux.
* Khả năng nhúng và mở rộng của chương trình: Python dễ dàng kết hợp các đoạn mã C, C++ và những ngôn ngữ lập trình khác vào đoạn mã python. Điều này giúp ứng dụng có thêm nhiều tính năng tốt hơn mà những ngôn ngữ lập trình đơn lẻ khác khó có thể làm được.
* Ngôn ngữ thông dịch cấp cao: Khác với C/C++, python không khó khăn trong việc quản lý bộ nhớ. Vì khi chạy mã nó tự động chuyển mã lập trình thành ngôn ngữ máy giúp hoàn thành công việc. Bởi vì khi lập trình với nhiều
* Thư viện tiêu chuẩn lớn và phổ biến: Python có một lượng lớn các thư viện tiêu chuẩn giúp việc lập trình đơn giản hơn rất nhiều vì không trực tiếp cần phải tự viết tất cả các đoạn mã. Mà chỉ việc hiểu và ứng dụng các thư viện có sẵn được liên kết với python.
* Hướng đối tượng: Là ngôn ngữ lập trình hướng đối tượng (OOP) giúp giải quyết nhiều vấn đề phức tạp một cách trực quan. Với các vẫn đề phức tạp OPP giúp chia nhỏ tạo ra các đối tượng.

Hiện nay python vẫn là ngôn ngữ lập trình được sử dụng phổ biến trong các lĩnh vực của đời sống như: Lập trình ứng dụng web, khoa học tính toán, dự báo thiên nhiên, chẩn đoán trong y học, phát triển trò chơi.

Ngoài những tính năng nổi bật như trên python còn có một số hạn chế đáng kể:

* Tốc độ tính toán, thực thi chậm.
* Sử dụng nhiều không gian bộ nhớ.
* Hạn chế trong việc phát triển ứng dụng trên thiết bị di động.
* Hạn chế trong việc thiết kế ứng dụng.

### Thư viện

#### Thư viện Keras

Keras mà một mã nguồn mở được sử dụng cho mạng nơ-ron được viết bởi ngôn ngữ python. Nó là một thư viện được tạo ra bởi Francois Chollet, là một kỹ sư chuyên gia nghiên cứu về deep learning. Keras có thể sử dụng chung với nhiều thư viện nổi tiếng: Tensorflow, CNTK, Theano. Keras nổi bật với những ưu điểm dưới đây:

* Sử dụng đơn giản, xây dựng mô hình một cách nhanh chóng.
* Chạy được trên cả CPU lẫn GPU.Hỗ trợ xây dựng ANN, RFR, CNN và nhiều mô hình khác.

#### Thư viện pandas

Pandas là một thư viện mã nguồn mở được xây dựng và phát triển bởi McKinney vào năm 2008. Pandas được sử dụng chủ yếu để phân tích và dọn dẹp dữ liệu. Pandas cung cấp cho việc lập trình những cấu trúc dữ liệu cũng như các phép toán trong việc xử lý dữ liệu chuỗi thời gian (time series). Pandas nổi bật với các tính năng sau:

* Phân tích dữ liệu nhanh và hiệu quả.
* Có thể sử dụng từ nhiều nguồn dữ liệu khác nhau.
* Dễ dàng xử lý những dữ liệu bị thiếu ví dụ như NaN.Cung cấp các chức năng chuỗi dữ liệu (time-series).

#### Thư viện OpenCV

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở sử dụng phổ biến trong 2 lĩnh vực chính là : thị giác máy tính và xử lý hình ảnh. Nó được phát triển bởi Intel và được duy trì bởi một nhóm các nhà phát triển đóng góp từ khắp nơi trên thế giới. OpenCV được viết bằng ngôn ngữ C++, nhưng cung cấp các giao diện lập trình cho nhiều ngôn ngữ như Python, Java, và MATLAB.

OpenCV cung cấp một loạt các chức năng và công cụ mạnh mẽ để phân tích, xử lý và trích xuất thông tin từ hình ảnh và video. Dưới đây là một số tính năng chính của OpenCV:

* Xử lý ảnh: OpenCV cung cấp các chức năng để đọc, ghi, và xử lý hình ảnh. Bạn có thể thực hiện các thao tác như lọc ảnh, biến đổi hình ảnh, phát hiện cạnh, phát hiện đặc trưng, phân đoạn ảnh và nhiều hơn nữa.
* Xử lý video: OpenCV hỗ trợ xử lý video và phân tích video trực tiếp. Bạn có thể thực hiện các thao tác như ghi video, đọc video, trích xuất khung hình, xử lý khung hình, theo dõi đối tượng, và nhận dạng chuyển động.
* Machine learning: OpenCV tích hợp các thuật toán học máy và học sâu. Bạn có thể sử dụng các mô hình đã được huấn luyện trước hoặc huấn luyện các mô hình mới để phân loại, nhận dạng, và dự đoán đối tượng trong hình ảnh hoặc video.
* Xử lý thời gian thực: OpenCV hỗ trợ xử lý hình ảnh và video trong thời gian thực. Bạn có thể tạo các ứng dụng thị giác máy tính có khả năng xử lý hình ảnh và video theo thời gian thực, chẳng hạn như xử lý hình ảnh từ camera trực tiếp.
* Tương tác với phần cứng: OpenCV cho phép tương tác với các thiết bị phần cứng như camera và cảm biến. Bạn có thể đọc dữ liệu từ camera, điều khiển camera và thực hiện các thao tác khác với các thiết bị ngoại vi.
* Đa nền tảng: OpenCV hỗ trợ nhiều nền tảng, bao gồm Windows, Linux, macOS, iOS và Android. Bạn có thể phát triển ứng dụng sử dụng OpenCV trên nhiều nền tảng khác nhau mà không cần thay đổi mã nguồn.

OpenCV là một thư viện mạnh mẽ và linh hoạt, được sử dụng trong nhiều ứng dụng như xử lý ảnh y tế, nhận dạng khuôn mặt, tự động hóa công nghiệp, xe tự hành, và nhiều lĩnh vực khác. Nó cung cấp một tập hợp các công cụ và thuật toán tiện lợi để giúp bạn xây dựng ứng dụng thị giác máy tính mạnh mẽ.

#### Thư viện Numpy

Thư viện NumPy (Numerical Python) là một thư viện mã nguồn mở cho ngôn ngữ lập trình Python, cung cấp các công cụ và chức năng để làm việc với mảng và ma trận đa chiều. NumPy là một thư viện quan trọng trong cộng đồng khoa học dữ liệu và tính toán số, và nó là một phần không thể thiếu trong nhiều ứng dụng Python.

Các tính năng chính của thư viện NumPy áp dụng vào bài toán :

* Mảng đa chiều: NumPy tạo ra một cấu trúc dữ liệu gọi là ndarray (n-dimensional array) để lưu trữ mảng đa chiều. Mảng ndarray trong NumPy cho phép lưu trữ và xử lý dữ liệu nhanh chóng, hiệu quả và linh hoạt. Mảng này có thể có nhiều chiều, cho phép bạn thực hiện các phép toán trên các phần tử mảng một cách dễ dàng.
* Phép toán số học và logic: NumPy cung cấp các hàm số học và logic để thực hiện các phép toán cơ bản như cộng, trừ, nhân, chia, luỹ thừa, căn bậc hai, phép so sánh, và logic. Các phép toán này có thể được áp dụng cho mảng ndarray và được thực hiện nhanh chóng và hiệu quả.
* Truy cập và cắt mảng: NumPy cho phép truy cập và cắt mảng dễ dàng thông qua các chỉ số và đánh dấu vị trí. Bạn có thể truy cập các phần tử, hàng, cột hoặc các khối con của mảng một cách linh hoạt và hiệu quả.
* Hỗ trợ cho ma trận và tính toán ma trận: NumPy cung cấp các chức năng mạnh mẽ để làm việc với ma trận, bao gồm các phép nhân ma trận, chuyển vị ma trận, tính định thức, tính giá trị riêng và nhiều hơn nữa. Điều này rất hữu ích trong các ứng dụng liên quan đến đại số tuyến tính và lý thuyết đồ thị.
* Thao tác và xử lý dữ liệu: NumPy cung cấp nhiều công cụ và chức năng để thao tác và xử lý dữ liệu trong mảng ndarray. Bạn có thể sắp xếp mảng, ghép mảng, tìm kiếm phần tử, thống kê dữ liệu và thực hiện nhiều thao tác dữ liệu khác một cách thuận tiện.

NumPy là một thư viện rất mạnh mẽ và quan trọng cho tính toán số và xử lý dữ liệu trong Python. Nó cung cấp một cơ sở chắc chắn cho việc phân tích dữ liệu, tính toán khoa học, và xử lý hình ảnh, và là một công cụ quan trọng trong lĩnh vực khoa học dữ liệu và machine learning.

#### Thư viện Imutils

Thư viện imutils là một thư viện Python được phát triển bởi Adrian Rosebrock, tập trung vào việc thực hiện các thao tác xử lý hình ảnh cơ bản một cách dễ dàng và thuận tiện. Imutils cung cấp một tập hợp các chức năng hỗ trợ cho việc xử lý hình ảnh trong OpenCV, giúp đơn giản hóa mã nguồn và tăng tốc quá trình phát triển ứng dụng thị giác máy tính.

Đối với bài toán nhận diện biển số xe tập trung vào một số vấn đề như :

* Resize: Chức năng resize trong imutils giúp thay đổi kích thước của hình ảnh một cách dễ dàng. Nó hỗ trợ nhiều phương pháp thay đổi kích thước, bao gồm giữ nguyên tỷ lệ khung hình, thay đổi chiều rộng hoặc chiều cao cố định và thay đổi theo tỷ lệ tùy chỉnh.
* Đọc và ghi hình ảnh: Imutils cung cấp các chức năng imread và imwrite để đọc và ghi hình ảnh một cách thuận tiện. Bạn có thể đọc hình ảnh từ tệp tin hoặc từ URL và lưu hình ảnh thành tệp tin ở định dạng khác nhau.

Thư viện imutils là một công cụ hữu ích để giảm thời gian và công sức trong quá trình phát triển ứng dụng thị giác máy tính với OpenCV. Nó giúp đơn giản hóa việc xử lý hình ảnh và cung cấp các chức năng cơ bản mà thường xuyên được sử dụng trong các dự án liên quan đến thị giác máy tính.

## Mô hình

ResNet (Residual Network) là một kiến trúc mạng nơ-ron sâu (deep neural network) được đề xuất bởi Kaiming He và đồng nghiệp vào năm 2015. Nó đã giúp giải quyết vấn đề mất mát thông tin (vanishing gradient) trong việc huấn luyện mạng nơ-ron sâu và đạt được hiệu suất tốt trong các bài toán nhận dạng hình ảnh và xử lý ảnh.

Diagram

Description automatically generated

Hình 21.3.3a. Mô hình Resnet

Ký hiệu đầu vào là x. Giả sử ánh xạ lý tưởng muốn học được là f(x), và được dùng làm đầu vào của hàm kích hoạt. Phần nằm trong viền nét đứt bên trái phải khớp trực tiếp với ánh xạ f(x). Điều này có thể không đơn giản nếu chúng ta không cần khối đó và muốn giữ lại đầu vào x. Khi đó, phần nằm trong viền nét đứt bên phải chỉ cần tham số hoá độ lệch khỏi giá trị x, bởi vì ta đã trả về x + f(x). Trên thực tế, ánh xạ phần dư thường dễ tối ưu hơn, vì chỉ cần đặt f(x) = 0. Nửa bên phải mô tả khối phần dư cơ bản của ResNet.

Diagram

Description automatically generated

Hình 22.3.3b: Sự khác biệt giữa một khối thông thường (trái) và một khối phần dư (phải). Trong khối phần dư, ta có thể nối tắt các tích chập.

Nếu muốn thay đổi số lượng kênh hoặc sải bước trong khối phần dư, cần thêm một tầng tích chập 1 × 1 để thay đổi kích thước đầu vào tương ứng ở nhánh ngoài. Có thể thay đổi số kênh và các khối phần dư trong môđun để tạo ra các mô hình ResNet khác nhau.

Dưới đây là một số thông tin chi tiết về kiến trúc mô hình ResNet:

Residual Learning:

* Đặc trưng chính của ResNet là khái niệm của residual learning (học có dư). Thay vì học các hàm trực tiếp từ các tầng đầu vào đến các tầng đầu ra, ResNet sử dụng các kết nối dư (residual connections) để truyền dữ liệu từ đầu vào trực tiếp đến các tầng đầu ra.
* Các kết nối dư cho phép thông tin được truyền qua mô hình mà không bị biến đổi quá nhiều, giúp tránh hiện tượng mất mát thông tin và cho phép xây dựng mạng nơ-ron sâu hơn mà vẫn có khả năng học tốt.

Residual Block:

* ResNet được xây dựng dựa trên khối (block) cơ bản gọi là residual block.
* Mỗi residual block bao gồm một chuỗi các tầng convolutional và tầng kết nối dư.
* Tầng convolutional thường sử dụng các bộ lọc (filter) có kích thước nhỏ (thường là 3x3) để học các đặc trưng ảnh.
* Sau mỗi tầng convolutional, kết quả được cộng với đầu vào ban đầu thông qua kết nối dư.
* Một số block cũng có thể sử dụng tầng pooling hoặc tầng batch normalization để cải thiện hiệu suất mô hình.

Kiến trúc chính của ResNet

* ResNet có một kiến trúc chính với sự kết hợp của các residual block.
* Trong kiến trúc gốc ResNet, có 5 giai đoạn (stage), mỗi giai đoạn bao gồm một số lượng residual block.
* Giai đoạn đầu tiên (stage 1) sử dụng một tầng convolutional và một tầng pooling để giảm kích thước ảnh.
* Các giai đoạn tiếp theo (stage 2 đến stage 5) sử dụng các residual block với số lượng tầng tùy thuộc vào biến thể của ResNet.
* Cuối cùng, một tầng pooling và một tầng fully connected được sử dụng để phân loại đầu ra.

Biến thể của ResNet:

* Có nhiều biến thể của ResNet như ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152 và nhiều hơn nữa.
* Số trong tên biến thể thường chỉ số lượng tầng trong mạng.
* Số lượng tầng và độ sâu của các biến thể khác nhau cho phép tùy chỉnh mô hình cho các yêu cầu cụ thể của bài toán.

Ứng dụng của ResNet: ResNet đã được áp dụng rộng rãi trong các bài toán nhận dạng hình ảnh, như phân loại ảnh, nhận dạng đối tượng, nhận diện khuôn mặt, phân đoạn ảnh và nhiều ứng dụng khác trong lĩnh vực thị giác máy tính.

## Cơ sở dữ liệu

### Bộ cơ sở dữ liệu MNIST

Cơ sở dữ liệu MNIST (Modified National Institute of Standards and Technology) là một trong những cơ sở dữ liệu phổ biến và được sử dụng rộng rãi trong lĩnh vực học máy và thị giác máy tính. Nó được phát triển từ cơ sở dữ liệu của Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ.

MNIST bao gồm một tập hợp các hình ảnh chữ số viết tay từ 0 đến 9, với tổng cộng 70.000 mẫu. Cơ sở dữ liệu này được chia thành hai phần: một tập huấn luyện gồm 60.000 hình ảnh và một tập kiểm tra gồm 10.000 hình ảnh. Mỗi hình ảnh trong MNIST có kích thước 28x28 pixel và được biểu diễn dưới dạng một ma trận gồm các giá trị pixel từ 0 đến 255, thể hiện mức độ sáng tương ứng.

Việc sử dụng MNIST không chỉ giúp nghiên cứu và phát triển thuật toán học máy, mà còn cung cấp một cơ sở để so sánh và đánh giá độ chính xác của các mô hình và phương pháp khác nhau. MNIST đã trở thành một thước đo phổ biến để so sánh các công trình nghiên cứu và đánh giá sự tiến bộ trong lĩnh vực nhận dạng chữ số

A picture containing text

Description automatically generated

Hình 23.3.4.1. Một số hình ảnh ví dụ trích xuất từ cơ sở dữ liệu MNIST.

### Bộ cơ sở dữ liệu A\_Z handwritten

Bộ cơ sở dữ liệu A-Z Handwritten (A\_Z Handwritten) là một tập dữ liệu được sử dụng phổ biến trong lĩnh vực nhận dạng chữ viết tay. Bộ dữ liệu này chứa các hình ảnh của các chữ cái từ A đến Z viết tay trên giấy. Bộ dữ liệu A-Z Handwritten Alphabets Dataset được tạo ra bởi Nishant Sinha và được phát hành dưới giấy phép CC BY-SA 4.0.

Dữ liệu A\_Z Handwritten bao gồm 26 chữ cái in hoa tiếng Anh viết tay (A-Z), tổng cộng 26,000 hình ảnh grayscale (ảnh xám) với 372,450 mẫu có kích thước 28x28 pixel. Mỗi hình ảnh tương ứng với một chữ cái trong bảng chữ cái tiếng Anh từ A đến Z. Các hình ảnh này đã được chuẩn hóa và điều chỉnh kích thước để phù hợp với việc sử dụng trong các thuật toán học máy và mô hình nhận dạng.



Hình 24.3.4.2. Ví dụ trích xuất từ cơ sở dữ liệu A-Z Handwritten.

# THỰC NGHIỆM

## Các bước thực hiện

Dưới đây là các bước chính để thực hiện quy trình nhận dạng chữ viết tay:

Chuẩn bị dữ liệu

* Tải xuống bộ dữ liệu MNIST và A-Z Handwritten. Bộ dữ liệu MNIST chứa ảnh chữ viết tay các chữ số từ 0 đến 9, trong khi bộ dữ liệu A-Z Handwritten chứa ảnh chữ viết tay các chữ cái từ A đến Z.
* Tiền xử lý dữ liệu: Chuyển đổi định dạng ảnh thành định dạng phù hợp với mô hình, thường là kích thước ảnh đồng nhất và chuẩn hóa giá trị pixel.

Xây dựng mô hình CNN

* Định nghĩa mô hình: Sử dụng kiến trúc CNN để xây dựng mô hình nhận diện chữ viết tay. Sử dụng các lớp tích chập (convolutional layers), lớp gộp (pooling layers), và các lớp kết nối đầy đủ (fully connected layers).
* Thiết lập các tham số: Xác định số lượng lớp, số lượng bộ lọc, kích thước cửa sổ bộ lọc, hệ số học (learning rate), và các tham số khác cho mô hình của bạn.

Xây dựng mô hình Resnet

* Sử dụng các kết nối "shortcut" trong các khối dư (residual block).
* Xây dựng 1 khối residual gồm có các lớp BatchNormalization, lớp Activation, lớp Conv2D.
* Dựa vào giá trị của stages (số khối residual) và filters (số filters của từng khối) tiến hành lặp qua từng stages.

Tiến hành huấn luyện

* Tiến hành huấn luyện mô hình bằng cách đưa dữ liệu huấn luyện qua mô hình và điều chỉnh các trọng số của mô hình để giảm thiểu sai số.
* Đánh giá mô hình bằng cách sử dụng tập kiểm tra và tính toán độ chính xác (accuracy), độ mất mát (loss).

Đánh giá và dự đoán

* Sử dụng tập kiểm tra hoặc tập dữ liệu mới, đưa dữ liệu qua mô hình đã được huấn luyện.
* Đánh giá độ chính xác hoặc các độ đo khác để đánh giá hiệu suất của mô hình.
* Đưa ra biểu đồ so sánh
* Dự đoán chữ viết tay bằng cách sử dụng mô hình đã huấn luyện trên các ảnh chữ viết tay mới.

Chart, diagram

Description automatically generated

Hình 25.4.1. Sơ đồ các bước thực hiện

## Thực nghiệm

### Đưa dữ liệu vào

Graphical user interface, application, table, Excel

Description automatically generated

Hình 26.4.2.1a. Dữ liệu A-Z handwritten.csv

Đọc dữ liệu A-Z handwritten

Text

Description automatically generated

Hình 27.4.2.1b. Đọc dữ liệu A-Z handwritten

Đọc dữ liệu Mnist database

Text, letter

Description automatically generated

Hình 28.4.2.1c. Đọc dữ liệu Mnist database

Text

Description automatically generated

Hình 294.2.1d. Kết quả ra kích cỡ file dữ liệu

Kích thước của bộ dữ liệu A-Z là (372450, 28, 28), có tổng cộng 372,450 mẫu với kích thước 28x28. Các nhãn tương ứng được hiển thị trong mảng az\_labels.

Kích thước của bộ dữ liệu MNIST là (70000, 28, 28), bao gồm 70,000 mẫu với kích thước 28x28. Các nhãn tương ứng được hiển thị trong mảng mnist\_labels.

### Xây dựng mô hình

Triển khai một module Resnet để huấn luyện mô hình mạng nơron tích chập (CNN). Mỗi module ResNet được tạo ra bằng cách xếp chồng các residual block (bao gồm nhiều residual module) và kết thúc bằng một lớp fully connected layer. Mô hình sử dụng các kết nối skip (shortcut connection) để giúp giảm vanishing gradient và cải thiện hiệu suất mô hình, để giữ lại thông tin quan trọng của đầu vào. Khi số lượng các residual block tăng lên, mô hình có thể học được các feature phức tạp hơn và độ chính xác cũng được cải thiện kết nối trực tiếp giữa đầu vào và đầu ra của các layer) để giúp mô hình dễ dàng học được các feature phức tạp hơn.

A picture containing text

Description automatically generated

Hình 30.4.2.2a. Xây dựng một khối Residual

Xây dựng một phương thức tĩnh `build` trong một lớp có tên là `ResNet`. Phương thức này được sử dụng để xây dựng một kiến trúc mô hình ResNet để phân loại hình ảnh dựa trên khối residual module đã được định nghĩa.

Text

Description automatically generated

Hình 31.4.2.2b. Xây dựng model

### Tiền xử lý dữ liệu

Bước 1: Chuẩn bị ảnh màu

Ảnh màu là một hình ảnh được tái hiện với sự sử dụng của ba kênh màu chính: đỏ (R), xanh lá cây (G) và xanh dương (B). Mỗi pixel trong một ảnh màu được biểu diễn bằng một giá trị kết hợp của ba kênh màu này, cho phép tái tạo một loạt màu sắc khác nhau.

Một ảnh màu có thể được lưu trữ và truyền tải dưới dạng các định dạng phổ biến như JPEG, PNG hoặc TIFF. Các phần mềm xử lý ảnh và trình duyệt hình ảnh thường hỗ trợ hiển thị và chỉnh sửa ảnh màu.

Text, letter

Description automatically generated

Hình 32.4.2.3a. Ảnh màu chữ viết tay

Bước 2: Đưa về ảnh xám

Ảnh xám, còn được gọi là ảnh đen trắng, là một loại ảnh mà chỉ có các mức độ xám khác nhau mà không có màu sắc. Mỗi pixel trong ảnh xám chỉ cần biểu diễn bằng một giá trị nguyên trong khoảng từ [0,255] thay vì (r,g,b) như trong ảnh màu. Giá trị 0 là màu đen, 255 là màu trắng và giá trị pixel càng gần 0 thì càng tối và càng gần 255 thì càng sáng.

Chuyển hệ màu trong ảnh:

* Mỗi pixel trong ảnh màu được biểu diễn bằng 3 giá trị (r,g,b) còn trong ảnh xám chỉ cần 1 giá trị x để biểu diễn.
* Khi chuyển từ ảnh màu sang ảnh xám ta có thể dùng công thức:

x = r \* 0.299 + g \* 0.587 + b \* 0.114.

Các trọng số trong công thức thể hiện sự khác biệt trong việc quan sát màu sắc của mắt người. Sự nhạy cảm của mắt người với màu sắc không đồng đều và phụ thuộc vào bước sóng của ánh sáng. Mắt người nhạy cảm nhất với màu xanh lá cây (green) sau đó đến màu đỏ (red) và nhạy cảm ít nhất với màu xanh dương (blue).

Điều này có nghĩa là mắt người nhìn thấy màu xanh lá cây (green) rõ ràng và nhạy cảm hơn so với các màu khác, trong khi màu xanh dương (blue) ít được nhạy cảm. Điều này cũng phản ánh trong việc chọn trọng số cho các kênh màu khi chuyển đổi sang ảnh xám

Text, letter

Description automatically generated

Hình 33.4.2.3b. Ảnh xám

Bước 3: Lọc nhiễu

Lọc nhiễu là quá trình loại bỏ hoặc giảm thiểu các thành phần nhiễu không mong muốn từ một tín hiệu hoặc tập dữ liệu. Có nhiều phương pháp lọc nhiễu khác nhau, bao gồm lọc thông thấp, lọc thông cao, lọc thông dải, lọc thông dải qua, lọc tuyến tính và phi tuyến, cũng như lọc Kalman. Sự lựa chọn phương pháp phụ thuộc vào loại nhiễu và yêu cầu của hệ thống cụ thể.



Graphical user interface, text, chat or text message

Description automatically generated

Hình 34.4.2.3c. Làm mờ ảnh xám

Bước 4: Phát hiện cạnh

Phát hiện cạnh là quá trình xác định và đánh dấu các điểm chuyển đổi sắc thái hoặc độ dốc trong một hình ảnh. Nó giúp xác định ranh giới và biên cạnh của các đối tượng trong hình ảnh. Các phương pháp phổ biến để phát hiện cạnh bao gồm việc sử dụng các bộ lọc gradient, như Sobel hoặc Prewitt, và các thuật toán như Canny và Laplacian của Gaussian. Các kỹ thuật phát hiện cạnh rất hữu ích trong xử lý ảnh, thị giác máy tính, và các ứng dụng liên quan đến trí tuệ nhân tạo và thị giác máy tính.



Text

Description automatically generated

Hình 35.4.2.3d. Ảnh phát hiện biên

### Đánh giá mô hình

Để đánh giá hiệu suất của mô hình đã xây dựng, tôi sử dụng tập dữ liệu kiểm tra như đã được trình bày trong mục 3.3. Chúng tôi đánh giá mô hình với các giá trị epoch là 5 và batch size là 256.

Text

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Khi thực thi chương trình, độ chính xác của mô hình huấn luyện sẽ được in ra theo từng epoch và ở cuối lỗi phân loại cũng được in ra. Bảng 1 tóm tắt độ chính xác phân loại, tỷ lệ lỗi và thời gian trung bình khi thực thi mô hình với các cấu hình mạng khác nhau.

Bảng 2.4.2.4a. Các kết quả thực nghiệm với các cấu hình mạng khác nhau

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cấu hình mạng | | | Kết quả phân loại | | Thời gian trung bình thực hiện 1 epoch (giây) |
| Số filter trong mỗi Convolutional Layer | Hàm tối ưu hóa | Mức học | Độ chính xác cao nhất (%) | Tỷ lệ lỗi (%) |
| 64,64,128,256 | sgd | 0,0005 | 91,49 | 0,99 | 4767,8 |
| 64,128,256,512 | sgd | 0,0005 | 88.32 | 6,96 | 7968,8 |
| 64,64,128,256 | adam | 0,0005 | 90,51 | 4,88 | 7666,6 |
| 64,128,256,512 | sgd | 0,001 | 89,54 | 3,85 | 4194,4 |

Chart, line chart

Description automatically generated

Hình 4.2.4a1. Biểu đồ 64,64,128,256-sgd-0.0005

Chart, line chart

Description automatically generated

Hình 4.2.4a2. Biểu đồ 64,64,128,256-adam-0.0005

Tôi cũng thực hiện so sánh kết quả nghiên cứu của tôi với các kết quả nghiên cứu trước đó trên cùng tập dữ liệu huấn luyện và kiểm tra.

Bảng 3.4.2.4b. So sánh với các kết quả nghiên cứu khác

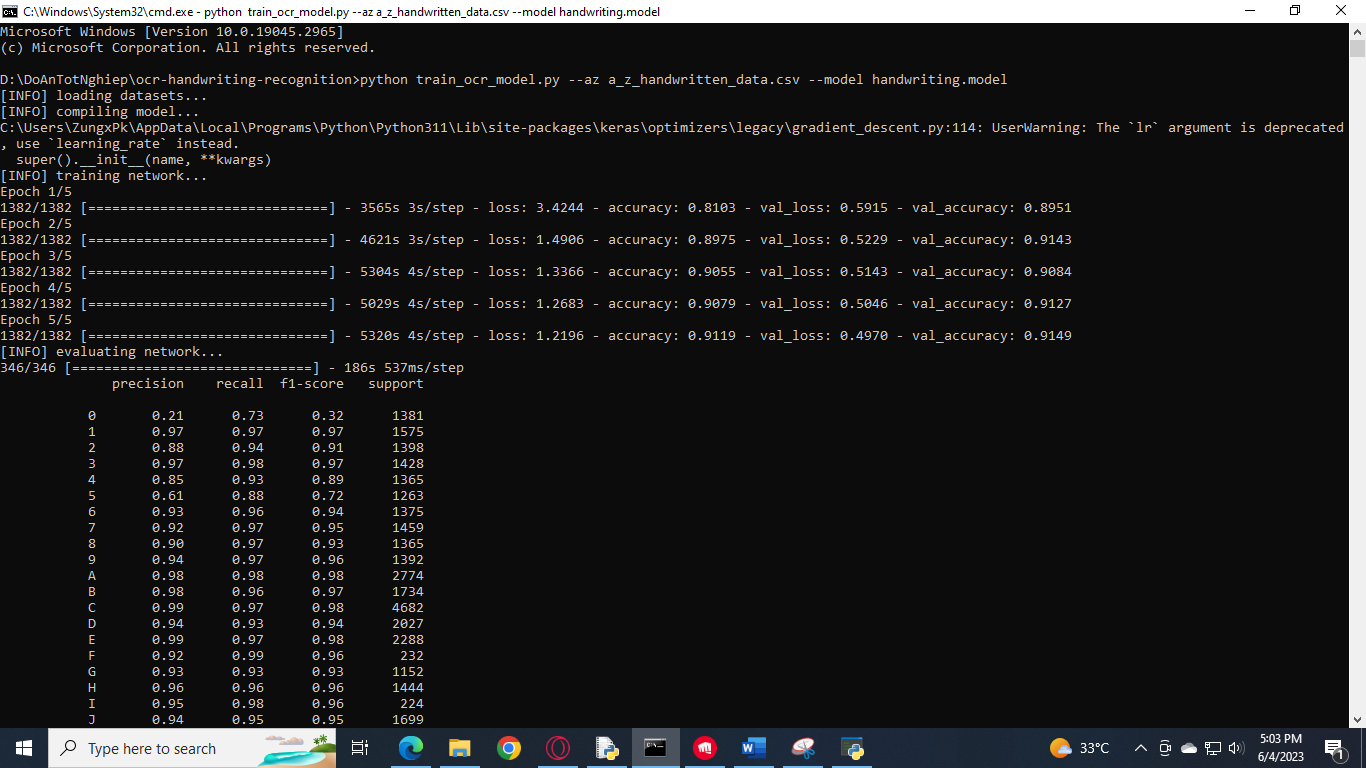
|  |  |  |
| --- | --- | --- |
| Nghiên cứu của tác giả | Độ chính xác cao nhất (%) | Tỷ lệ lỗi (%) |
| LeCun và các tác giả | 99,01 | 0,95 |
| Jason Brownlee | 99,31 | 0,82 |
| Kasun và các tác giả | 99,03 | 0,97 |
| Tapson và các tác giả | 96,00 | 1,52 |
| Tapson và các tác giả | 90,00 | 2,75 |
| Nghiên cứu của tôi | 91,49 | 8,61 |

Nguyên nhân dẫn đến nhận diện sai:

* Mô hình cần nhiều thời gian huấn luyện để tìm ra kết quả tốt nhất
* Dữ liệu viết tay có độ phức tạp cao hơn so với dữ liệu chữ in. Các chữ cái viết tay có thể khác nhau đáng kể về kích thước, hình dạng và phong cách viết. Điều này tạo ra những thách thức cho mô hình trong việc nhận diện các chữ cái khác nhau đúng xác suất cao.
* Số lượng dữ liệu huấn luyện hạn chế: bộ dữ liệu huấn luyện không đại diện đúng cho toàn bộ phân phối của chữ viết tay. Điều này có thể làm cho mô hình không thể học được tất cả các biến thể của các chữ cái và dẫn đến hiện tượng nhận diện sai.
* Nhiễu và biến dạng: Dữ liệu chữ viết tay thường bị nhiễu và biến dạng. Có thể có những vết mờ, nét viết không đều, nét viết lấn lên nhau hoặc bị thừa.

Giải pháp:

* Tăng kích thước bộ dữ liệu huấn luyện và đảm bảo đại diện đúng cho toàn bộ phân phối của chữ viết tay.
* Áp dụng các kỹ thuật tiền xử lý dữ liệu như chuẩn hóa, cân bằng lớp dữ liệu và loại bỏ nhiễu và biến dạng.
* Xem xét sử dụng các kiến trúc mạng nơ-ron khác nhau hoặc tinh chỉnh các siêu tham số của mạng hiện tại để cải thiện hiệu suất.
* Sử dụng các kỹ thuật tăng cường dữ liệu để tạo thêm dữ liệu huấn luyện bằng cách áp dụng các biến dạng và nhiễu khác nhau lên các chữ cái hiện có



Hình 36.4.2.4b1. Kết quả 64,64,128,256-sgd-0.0005

A picture containing text, electronics

Description automatically generated

Hình 37.4.2.4b2. Kết quả 64,64,128,256-sgd-0.0005

A screenshot of a computer

Description automatically generated with medium confidence

Hình 38.4.2.4c1. Kết quả 64,128,256,512-sgd-0.0005

A picture containing text, electronics

Description automatically generated

Hình 39.4.2.4c2. Kết quả 64,128,256,512-sgd-0.0005

A screenshot of a computer

Description automatically generated with medium confidence

Hình 40.4.2.4d1. Kết quả 64,64,128,256-adam-0.0005

A screenshot of a computer

Description automatically generated with medium confidence

Hình 41.4.2.4d2. Kết quả 64,64,128,256-adam-0.0005

A screenshot of a computer

Description automatically generated with medium confidence

Hình 42.4.2.4e1. Kết quả 64,128,256,512-sgd-0.0005-epo=10,bs=128

A picture containing text, electronics, computer

Description automatically generated

Hình 43.4.2.4e2. Kết quả 64,128,256,512-sgd-0.0005-epo=10,bs=128

Text, letter, whiteboard

Description automatically generated

Hình 44.4.2.4f. Kết quả nhận diện ảnh chụp chữ viết tay

# Kết luận

Demo nhận dạng với phương pháp trình bày trong đồ án này đã đạt được một số điểm như:

* Nắm được các vấn đề cơ bản của ảnh số và xử lý ảnh số
* Sử dụng tương đối tốt thư viện OpenCV cho Python để xử lý ảnh.
* Nắm được các đặc điểm của một bài toán nhận dạng nói chung và nhận dạng chữ cái viết tay nói riêng
* Demo và Test thử thành công trên các bộ dữ liệu mẫu với tỉ lệ nhận dạng chính xác 91%.

**Bên cạnh đó, còn tồn tại một số điểm hạn chế chưa giải quyết được như:**

* Chưa nhận diện được chữ cái tiếng việt mà chỉ nhận diện được chữ cái tiếng anh
* Tỉ lệ nhận dạng còn phụ thuộc nhiều vào điều kiện ánh sáng, phản chiếu hoặc che bóng.

**Hướng phát triển của bài toán**

Transfer learning cho phép chia sẻ tri thức đã học từ các nhiệm vụ khác để cải thiện hiệu suất nhận diện chữ viết tay. Thay vì huấn luyện mô hình từ đầu, ta có thể sử dụng mô hình đã được huấn luyện trước trên dữ liệu lớn (ví dụ: ImageNet) và điều chỉnh lại một số lớp cuối cùng hoặc thêm các lớp mới để phù hợp với bài toán nhận diện chữ viết tay cụ thể.

Một hướng phát triển tiềm năng khác là kết hợp các phương pháp nhận diện chữ viết tay khác nhau để tạo ra một hệ thống mạnh mẽ hơn. Ví dụ, kết hợp phương pháp dựa trên cấu trúc và phương pháp dựa trên biểu đồ có thể cải thiện khả năng nhận diện và độ chính xác.

Đối với những tình huống phức tạp hơn như nhận diện chữ viết tay trong các văn bản, có thể sử dụng thông tin ngữ cảnh bổ sung để cải thiện độ chính xác. Ví dụ, việc sử dụng thông tin về từ điển, ngữ pháp, ngữ nghĩa hoặc bối cảnh xung quanh từ cần nhận diện có thể giúp loại bỏ các giả định không mong muốn và tăng cường khả năng nhận diện.

Bên cạnh việc cải thiện hiệu suất của các phương pháp nhận diện chữ viết tay, mục tiêu phát triển tiếp theo là tạo ra các ứng dụng thực tế và tiện ích cao. Ví dụ, việc nhận diện chữ viết tay trên điện thoại di động, máy tính bảng hoặc bảng trắng thông minh có thể mang lại lợi ích lớn cho việc ghi chú, tìm kiếm thông tin và tương tác người dùng.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Z. Chen and B. Liu, "Lifelong Machine Learning," in *Synthesis Lectures on Artificial Intelligence and Machine Learning*, Morgan and Claypool, 2016. |
| [2] | G. Fei et al, "Learning Cumulatively to Become More Knowledgeable," *KDD,* 2016. |
| [3] | Z. Chen et al, "Lifelong Learning for Sentiment Classification," *ACL,* pp. 750-756, 2015. |
| [4] | "What is Transfer learning ?," Geeksforgeeks, 29 11 2023. [Online]. Available: https://www.geeksforgeeks.org/ml-introduction-to-transfer-learning/. |
| [5] | Z. Chen and B. Liu, "Lifelong and Continual Learning," [Online]. Available: https://www.cs.uic.edu/~liub/lifelong-learning.html. |
| [6] | "Difference between single task learning and multitask learning," ResearchGate, [Online]. Available: https://www.researchgate.net/figure/Difference-between-single-task-learning-and-multitask-learning-a-single-task-learning\_fig2\_307622018. |
| [7] | N. T. Trung and T. M. Tuấn, "Phân tích dữ liệu lớn: Phần 2," Đại học Thủy Lợi, [Online]. Available: http://tailieuso.tlu.edu.vn/handle/DHTL/8914. |
| [8] | V. H. Tiệp, "Bài 32: Naive Bayes Classifier," 8 8 2017. [Online]. Available: https://machinelearningcoban.com/2017/08/08/nbc/. |
| [9] | N. X. Bách et al, "Cross-Domain Intention Detection in Discussion Forums," *SoICT,* pp. 173-180, 2017. |
| [10] | V. H. Tiệp, "Bài 8: Gradient Descent (phần 2/2)," 16 1 2017. [Online]. Available: https://machinelearningcoban.com/2017/01/16/gradientdescent2/. |
| [11] | "Các phương pháp đánh giá mô hình học máy, học sâu (Machine learning & Deep learning)," Rabiloo, [Online]. Available: https://rabiloo.com/vi/blog/cac-phuong-phap-danh-gia-mo-hinh-machine-learning-va-deep-learning. |