

**BỘ GIÁO DỤC VÀ ĐÀO TẠO    BỘ NÔNG NGHIỆP VÀ PTNT**  
**TRƯỜNG ĐẠI HỌC THỦY LỢI**



**ĐINH NGỌC ANH**

**ỨNG DỤNG MỘT SỐ MÔ HÌNH HỌC MÁY ĐỂ CHẨN ĐOÁN  
UNG THƯ VÚ**

**ĐỒ ÁN TỐT NGHIỆP**

**HÀ NỘI, NĂM 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO      BỘ NÔNG NGHIỆP VÀ PTNT**  
**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**ĐINH NGỌC ANH**

**ỨNG DỤNG MỘT SỐ MÔ HÌNH HỌC MÁY ĐỂ CHẨN ĐOÁN  
UNG THU VÚ**

Ngành : Công nghệ thông tin

Mã số: 7480201

NGƯỜI HƯỚNG DẪN: TS. Lê Nguyễn Tuấn Thành

HÀ NỘI, NĂM 2024

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

**ĐINH NGỌC ANH**

**ĐỒ ÁN TỐT NGHIỆP**

**HÀ NỘI, NĂM 2024**



**TRƯỜNG ĐẠI HỌC THỦY LỢI  
KHOA CÔNG NGHỆ THÔNG TIN**

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ tên sinh viên: Đinh Ngọc Anh

Hệ đào tạo: Đại học Chính quy

Lớp: 61TH4

Ngành: Công nghệ thông tin

Khoa: Công nghệ thông tin.

**1. TÊN ĐỀ TÀI:**

Ứng dụng một số mô hình học máy để chẩn đoán ung thư vú.

**2. CÁC TÀI LIỆU CƠ BẢN:**

- [1] N. T. Trung. [Trực tuyến]. Available: <https://tailieu.vn/doc/luan-van-thac-si-tai-chinh-ngan-hang-ung-dung-ky-thuat-hoc-may-trong-xay-dung-mo-hinh-du-bao-tai-ch-2338549.html>.
- [2] [Online]. Available: <https://techie.vn/hoc-tang-cuong-reinforcement-learning-giai-dap-a-z/>.
- [3] [Online]. Available: <https://rabiloo.com/vi/blog/cac-phuong-phap-danh-gia-mo-hinh-machine-learning-va-deep-learning>.
- [4] [Online]. Available: [https://phamdinhhkhanh.github.io/deepai-book/ch\\_ml/classification.ht](https://phamdinhhkhanh.github.io/deepai-book/ch_ml/classification.ht).
- [5] [Online]. Available: <https://plus.vtc.edu.vn/machine-learning-la-gi>.
- [6] [Online]. Available: <https://plus.vtc.edu.vn/machine-learning-la-gi>.
- [7] <https://machinelearningcoban.com/2017/04/09/smv/>.

**3. NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN:**

Nội dung các phần thuyết minh và tính toán	Tỉ lệ %
Chương 1: Cơ sở lý thuyết	30%
Chương 2: Mô hình học máy trong việc dự đoán	40%
Chương 3: Thực nghiệm	30%

**4. GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN**

Phần	Họ tên giáo viên hướng dẫn
Chương 1: Cơ sở lý thuyết	TS. Lê Nguyễn Tuấn Thành
Chương 2: Mô hình học máy trong việc dự đoán	TS. Lê Nguyễn Tuấn Thành
Chương 3: Thực nghiệm	TS. Lê Nguyễn Tuấn Thành

## 5. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày ..... tháng ..... năm 2024

**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 2024

**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 2024.

**Sinh viên làm Đồ án tốt nghiệp**

*(Ký và ghi rõ Họ tên)*



Đinh Ngọc Anh



**TRƯỜNG ĐẠI HỌC THUY LỢI  
KHOA CÔNG NGHỆ THÔNG TIN**

**BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP**

**Tên đề tài: Ứng dụng một số mô hình học máy để chẩn đoán ung thư vú**

*Sinh viên thực hiện: Đinh Ngọc Anh*

*Lớp: 61TH4*

*Mã sinh viên: 1951060505*

*Số điện thoại: 0941103889*

*Email: 1951060505@e.tlu.edu.vn*

*Giáo viên hướng dẫn: TS. Lê Nguyễn Tuấn Thành*

**TÓM TẮT ĐỀ TÀI**

Mỗi năm số ca tử vong ngày càng tăng cao vì ung thư vú. Đây là loại thường xuyên nhất trong tất cả ung thư và nguyên nhân chính gây tử vong ở phụ nữ trên toàn thế giới. Bất kỳ sự phát triển nào để dự đoán và chẩn đoán ung thư bệnh tật là vốn quan trọng cho một cuộc sống khỏe mạnh. Do đó, độ chính xác cao trong dự đoán ung thư là rất quan trọng đối với cập nhật khía cạnh điều trị và tiêu chuẩn sống sót của bệnh nhân. Kỹ thuật học máy có thể mang lại lợi ích lớn góp phần vào quá trình dự đoán và chẩn đoán sớm ung thư vú, trở thành điểm nóng nghiên cứu và được chứng minh là một kỹ thuật mạnh mẽ. Trong nghiên cứu này, em đã áp dụng ba thuật toán học máy: Máy Vector hỗ trợ (SVM), hồi quy logistic và Mạng nơ-ron nhân tạo (ANN) trên Bộ dữ liệu Chẩn đoán Ung thư vú Wisconsin, sau khi có kết quả sẽ tiến hành đánh giá và so sánh hiệu suất được thực hiện giữa các phân loại khác nhau.

## **CÁC MỤC TIÊU CHÍNH**

- Phát triển một hệ thống chẩn đoán ung thư vú hiệu quả và chính xác dựa trên dữ liệu Wisconsin.
- Huấn luyện các mô hình học máy: Huấn luyện các mô hình SVM, hồi quy logistic và ANN cho nhiệm vụ phân loại ung thư vú.
- Tiền xử lý dữ liệu bằng cách xử lý các giá trị thiếu, chuẩn hóa dữ liệu và chia dữ liệu thành tập huấn luyện và tập thử nghiệm.
- Huấn luyện các mô hình học máy: Huấn luyện các mô hình SVM, hồi quy logistic và ANN cho nhiệm vụ phân loại ung thư vú.
- Tối ưu hóa các tham số của các mô hình học máy bằng phương pháp điều chỉnh siêu tham số.
- Đánh giá hiệu suất mô hình: Đánh giá hiệu suất của các mô hình học máy trên tập thử nghiệm bằng các chỉ số như độ chính xác, độ nhạy, độ đặc hiệu và F1-score.
- Phân tích các yếu tố ảnh hưởng đến hiệu suất của mô hình học máy.

## **KẾT QUẢ DỰ KIẾN**

- Các mô hình học máy SVM, hồi quy logistic, ANN sẽ được xây dựng và huấn luyện trên dữ liệu Wisconsin.
- Xây dựng thành công ba mô hình học máy (SVM, hồi quy logistic, ANN) để phân loại các trường hợp ung thư vú.
- Đánh giá hiệu suất của từng mô hình bằng các chỉ số như độ chính xác, độ nhạy, độ đặc hiệu. So sánh hiệu suất của các mô hình học máy với nhau và mô hình học máy có hiệu suất tốt nhất sẽ được lựa chọn cho việc chẩn đoán ung thư vú.
- Trong tương lai, nghiên cứu có thể mở rộng bằng cách thử nghiệm các phương pháp tiền xử lý dữ liệu khác nhau, tăng kích thước mẫu dữ liệu, hoặc thử nghiệm các kiểu mô hình học máy khác nhau để tìm ra phương pháp dự đoán tối ưu nhất cho bài toán phân loại ung thư vú.
- Đề xuất các hướng nghiên cứu tiếp theo để cải thiện hiệu quả chẩn đoán ung thư vú bằng phương pháp học.



## **LỜI CAM ĐOAN**

Em xin cam đoan đây là Đồ án tốt nghiệp của bản thân em. Các kết quả trong Đồ án tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu (nếu có) đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

**Tác giả ĐATN**



**Đinh Ngọc Anh**

## LỜI CẢM ƠN

Trong suốt quá trình hoàn thiện đề tài đồ án “Ứng dụng một số mô hình học máy để chẩn đoán ung thư vú”, em đã nhận được sự hướng dẫn và giúp đỡ nhiệt tình từ phía nhà trường, thầy cô và bạn bè.

Lời đầu tiên, em xin gửi lời cảm ơn chân thành tới thầy cô trường Đại học Thủy Lợi nói chung, thầy cô khoa Công nghệ thông tin nói riêng đã tận tình giảng dạy, truyền đạt những kiến thức quý giá cho em cũng như các bạn sinh viên khác trong quá trình học tập tại trường, từ đó làm cơ sở cho chúng em có kiến thức để thực hiện đồ án này.

Em xin gửi lời cảm ơn sâu sắc tới TS. Lê Nguyễn Tuấn Thành, giảng viên khoa Công nghệ thông tin đã tận tình hướng dẫn, chỉ bảo và góp ý cho em trong suốt quá trình thực hiện đồ án.

Với kinh nghiệm cũng như thời gian có hạn nên đồ án của em không thể không tránh khỏi những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của quý thầy cô để đồ án của em được hoàn thiện tốt hơn.

Em xin chân thành cảm ơn!

## MỤC LỤC

LỜI CAM ĐOAN .....	1
LỜI CẢM ƠN .....	2
MỤC LỤC .....	3
DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ .....	6
CHƯƠNG 1 CƠ SỞ LÝ THUYẾT .....	9
1.1    Lịch sử ra đời .....	9
1.2    Khái niệm .....	12
1.3    Cách hoạt động .....	13
1.4    Phân loại .....	15
1.4.1    Học có giám sát (Supervised Learning) .....	15
1.4.2    Học không giám sát (Unsupervised Learning) .....	16
1.4.3    Học bán giám sát (Semi – Supervisor Learning) .....	18
1.4.4    Học tăng cường (Rainforcement Learning) .....	20
1.5    Ưu và nhược điểm .....	21
1.5.1    Ưu điểm .....	21
1.5.2    Nhược điểm .....	22
1.6    Các ứng dụng .....	22
1.7    Môi trường cài đặt .....	23
1.7.1    Ngôn ngữ lập trình python .....	23
1.7.2    Các thư viện .....	23
1.7.3    Visual Studio Code .....	26
CHƯƠNG 2 MÔ HÌNH HỌC MÁY TRONG VIỆC DỰ ĐOÁN .....	27
2.1    Mô hình hồi quy tuyến tính Logistic .....	27
2.1.1    Hồi quy Logistic là gì ? .....	27
2.1.2    Hàm Sigmoid .....	28
2.1.3    Đường phân chia của hàm Sigmoid .....	29
2.1.4    Ứng dụng hồi quy Logistic .....	30
2.1.4    Ví dụ đơn giản của hồi quy Logistic .....	30
2.1.6    Mã giả của hồi quy Logistic .....	31
2.2    Mô hình Support Vector Machine (SVM) .....	32

2.2.1	Support Vector Machine là gì ?	32
2.2.2	Thuật toán SVM	33
2.2.3	Ứng dụng của thuật toán SVM	35
2.2.4	Bài toán ví dụ về SVM	36
2.2.5	Mã giả của SVM	41
2.3	Mô hình mạng thần kinh nhân tạo (ANN)	42
2.3.1	Mạng thần kinh nhân tạo là gì ?	42
2.3.2	Thuật toán ANN	45
2.3.2.	Ứng dụng của ANN	47
2.3.3	Bài toán ví dụ của ANN	47
2.3.4	Mã giả của ANN	52
2.4	Các chỉ số đánh giá	54
CHƯƠNG 3 THỰC NGHIỆM		56
3.1	Tổng quan về tập dữ liệu	56
3.2	Các bước thực nghiệm	58
3.2.1	Import thư viện	58
3.2.2	Đọc dữ liệu từ tệp	58
3.2.3	Tóm tắt dữ liệu	59
3.2.4	Mức độ tương quan giữa các biến	59
3.2.5	Tiền xử lý dữ liệu	61
3.2.6	Huấn luyện và kiểm thử	62
KẾT LUẬN		68
TÀI LIỆU THAM KHẢO		69

## DANH MỤC HÌNH ẢNH

Hình 1. 1: Tổng quan về học máy. ....	12
Hình 1. 2: Quá trình tự học của học máy. ....	14
Hình 1. 3: Quá trình tự phân cụm của học không giám sát.....	16
Hình 1. 4: Mô hình tương tác giữa các thuật ngữ trong học tăng cường. ....	20
Hình 2. 1: Đồ thị hàm Sigmoid trong trục Oxy.....	28
Hình 2. 2: Các mặt phân lớp hai class. ....	34
Hình 2. 3: Mặt phẳng phân chia và các điểm xanh đỏ thuộc hai lớp khác nhau.....	35
Hình 2. 4: Cấu trúc cơ bản của một ANN. ....	43
Hình 2. 5: Mô hình ANN.....	44
Hình 3. 1: Cấu trúc tập dữ liệu Breast Cancer Wisconsin.....	58
Hình 3. 2: Import các thư viện.....	58
Hình 3. 3: Đọc dữ liệu từ file.....	58
Hình 3. 4: Tóm tắt toàn bộ dữ liệu. ....	59
Hình 3. 5: Biểu đồ heatmap mức độ tương quan giữa các biến. ....	60
Hình 3. 6: Số lượng bản ghi cho hai lớp dự đoán. 357 ác tính và 212 lành tính.....	61
Hình 3. 7: Chia dữ liệu thành hai tập huấn luyện và kiểm thử.....	61
Hình 3. 8: Chuẩn hóa dữ liệu.....	62
Hình 3. 9: Huấn luyện và kiểm thử với logistic regression. ....	62
Hình 3. 10: Confusion Matrix của mô hình logistic regression. ....	63
Hình 3. 11: Chênh lệch giữa giá trị thực tế và giá trị dự đoán của mô hình hồi quy tuyến tính. ....	63
Hình 3. 12: Huấn luyện và kiểm thử trên mô hình SVM. ....	64
Hình 3. 13: Confusion matrix của mô hình SVM. ....	64
Hình 3. 14: Chênh lệch giữa giá trị thực tế và giá trị dự đoán của mô hình SVM.....	65
Hình 3. 15: Model ANN ba lớp.....	65

Hình 3. 16: Huấn luyện ANN với 150 epoch. ....	66
Hình 3. 17: Giá trị hàm mất mát và tỉ lệ chính xác sau 150 epoch. ....	66
Hình 3. 18: Confusion matrix của ANN.....	66

## DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ

Kí hiệu	Diễn giải	Ý nghĩa
AI	Artificial Intelligence	Trí tuệ nhân tạo
ANN	Artificial Neural Network	Mạng nơ-ron nhân tạo
SVM	Support vector machine	Máy vector hỗ trợ
ML	Machine Learning	Học Máy
HAC	Hierarchical Clustering	Phân tích cụm phân cấp

## GIỚI THIỆU

Vài năm trở lại đây, sự phát triển của khoa học công nghệ đã khai phá tiềm năng trong việc phân tích và vận dụng trí tuệ nhân tạo học máy trong y học. Đặc biệt, sự phát triển vượt bậc trong công nghệ hình ảnh y tế đã tạo ra một nguồn thông tin lớn về hình ảnh lâm sàng, từ cách thu thập dữ liệu cho đến cách phân tích chúng. Cùng với đó, sự tiến bộ trong công nghệ tế bào đơn giản hóa đã giúp thu thập và phân tích dữ liệu đa chiều một cách hiệu quả, bao gồm dữ liệu về hệ thống gen, dữ liệu về hệ thống protein của sinh vật và con người, và dữ liệu về trao đổi chất ở con người.

**Lí do chọn đề tài:** Ung thư vú là một trong những bệnh phổ biến nhất ở phụ nữ từ 40 đến 59 tuổi với nhiều yếu tố nguy cơ liên quan như yếu tố di truyền, môi trường và hành vi bởi sự tăng sinh rối loạn kèm theo sự phát triển không ngừng của các tế bào trong cơ quan này. Ung thư vú dạng viêm rất hiếm gặp và thường biểu hiện nặng, làm tổn thương toàn bộ vú, khiến vú bị sưng và sung huyết. Triệu chứng ban đầu là một nốt nhỏ ở vú, thường không đau và có thể phát triển chậm hoặc nhanh tùy thuộc vào khả năng gây ung thư của nó. Nó phải có đường kính khoảng một centimet thì ung thư vú mới có thể sờ thấy được. Và phải mất nhiều năm để nó đạt được kích thước này, vì vậy việc chẩn đoán sớm thậm chí còn khó khăn hơn, bởi 80% bệnh ung thư biểu hiện dưới dạng khối u không đau, trong đó chỉ có một thiểu số, 10% bệnh nhân phàn nàn về cơn đau mà không nhận thức được khối u. Có rất nhiều sự khác biệt về diễn biến lâm sàng của bệnh ung thư vú cũng như thời gian sống mà bệnh nhân có thể mong đợi. Một loạt các cơ chế vẫn chưa được biết đến, chẳng hạn như tình trạng miễn dịch, nội tiết tố và dinh dưỡng của bệnh nhân, ảnh hưởng đến sự khác biệt này, chẳng hạn như sự khác biệt về tốc độ nhân đôi của khối u và khả năng di căn của khối u.

**Mục tiêu:** Phát triển một mô hình học máy có thể dự đoán liệu một khối u vú là lành tính (benign) hay ác tính (malignant) dựa trên các đặc trưng hình ảnh kỹ thuật số của các mẫu sinh thiết tế bào khối u và bằng cách sử dụng các đặc trưng này, mô hình học máy có thể được huấn luyện để phân loại các khối u vú một cách chính xác dựa trên các đặc trưng của chúng.

**Đối tượng và phạm vi nghiên cứu:** bộ dữ liệu "Breast Cancer Wisconsin (Diagnostic)".

**Phương pháp nghiên cứu:** Phương pháp nghiên cứu sử dụng trong suốt quá trình thực hiện đồ án tốt nghiệp là phương pháp nghiên cứu lý thuyết các mô hình để thực hiện với bài toán cụ thể. Phương pháp thực nghiệm dựa trên tập dữ liệu thực tế để dự báo và có độ đo để xác định sự chính xác của mô hình.



## CHƯƠNG 1 CƠ SỞ LÝ THUYẾT

Trí Tuệ Nhân Tạo (AI) hay chi tiết hơn là Học Máy (Machine Learning) đã trở thành bằng chứng rõ ràng hơn bao giờ hết cho cuộc cách mạng lần thứ tư những năm qua. AI đang thâm nhập vào mọi khía cạnh của cuộc sống, đôi khi chúng ta thậm chí đã không nhận ra. Các ứng dụng tiêu biểu của Trí Tuệ Nhân tạo/ Học máy như xe tự hành của Tesla và Google, nhận diện khuôn mặt trên Facebook, trợ lý kỹ thuật số của Apple, hệ thống gợi ý âm nhạc của Spotify, hệ thống đề xuất phim của Netflix và máy chơi cờ vây AlphaGo của Google DeepMind và đây chỉ là một số ít trong rất nhiều ứng dụng của AI trong thực tế.

### 1.1 Lịch sử ra đời

Thật thú vị nếu như ta đặt câu hỏi rằng liệu giấc mơ về người máy của con người có thể thành hiện thực ? Thế nhưng, sự thực là khoa học đã tiến bộ đến giải đoạn mà chúng bắt đầu hòa lẫn với khoa học viễn tưởng. Hiện tại tuy ta chưa có các người máy tự động có khả năng đối đầu với con người nhưng ta đang ngày càng tiến gần tới những gì được gọi là "trí tuệ nhân tạo."

Hiện nay, các thuật toán học máy cho phép máy tính có khả năng tương tác với con người, lái xe tự động, viết và xuất bản báo cáo về các trận đấu thể thao, phát hiện các nghi phạm liên quan đến hoạt động khủng bố. Em tin chắc rằng machine learning sẽ tác động một cách sâu sắc đến mọi ngành công nghiệp và các công việc liên quan đến chúng, đó là lý do tại sao mọi nhà quản lý cần phải có ít nhất một số kiến thức về machine learning.

Trong phần này của nghiên cứu sẽ cung cấp một biên niên sử phát triển của machine learning và các sự kiện quan trọng gần đây nhất:

- Năm 1950 : Nhà bác học Alan Turing đã tạo ra "Turing Test (phép thử Turing)" để xác định xem liệu một máy tính có trí thông minh thực sự hay không. Để vượt qua bài kiểm tra đó, một máy tính phải có khả năng đánh lừa một con người tin rằng nó cũng là con người. [1]
- Năm 1952 : Arthur Samuel đã viết ra chương trình học máy (computer learning) đầu tiên. Chương trình này là trò chơi cờ đam, và hãng máy tính IBM đã cải tiến trò chơi

này để nó có thể tự học và tổ chức những nước đi trong chiến lược để giành chiến thắng. [1]

- Năm 1957 : Frank Rosenblatt đã thiết kế mạng noron (neural network) đầu tiên cho máy tính, trong đó mô phỏng quá trình suy nghĩ của bộ não con người. [1]
- Năm 1967 : Thuật toán "nearest neighbor" đã được viết, cho phép các máy tính bắt đầu sử dụng những mẫu nhận dạng (pattern recognition) rất cơ bản. Nó được sử dụng để vẽ ra lộ trình cho một người bán hàng có thể bắt đầu đi từ một thành phố ngẫu nhiên nhưng đảm bảo anh ta sẽ đi qua tất cả các thành phố khác theo một quãng đường ngắn nhất. [1]
- Năm 1979: Sinh viên tại trường đại học Stanford đã phát minh ra giỏ hàng "Stanford Cart" có thể điều hướng để tránh các chướng ngại vật trong một căn phòng. [1]
- Năm 1981: Gerald Dejong giới thiệu về khái niệm Explanation Based Learning (EBL), trong đó một máy tính phân tích dữ liệu huấn luyện và tạo ra một quy tắc chung để nó có thể làm theo bằng cách loại bỏ đi những dữ liệu không quan trọng. [1]
- Năm 1985: Terry Sejnowski đã phát minh ra NetTalk, nó có thể học cách phát âm các từ giống như cách một đứa trẻ tập nói. [1]
- Năm 1990: Machine Learning đã dịch chuyển từ cách tiếp cận hướng kiến thức (knowledge driven) sang cách tiếp cận hướng dữ liệu (data-driven). Các nhà khoa học bắt đầu tạo ra các chương trình cho máy tính để phân tích một lượng lớn dữ liệu và rút ra các kết luận - hay là "học" từ các kết quả đó. [1]
- Năm 1997: Deep Blue của hãng IBM đã đánh bại nhà vô địch cờ vua thế giới. [1]
- Năm 2006: Geoffrey Hinton đã đưa ra một thuật ngữ "deep learning" để giải thích các thuật toán mới cho phép máy tính "nhìn thấy" và phân biệt các đối tượng và văn bản trong các hình ảnh và video. [1]
- Năm 2010: Microsoft Kinect có thể theo dõi 20 hành vi của con người ở một tốc độ 30 lần mỗi giây, cho phép con người tương tác với máy tính thông qua các hành động và cử chỉ. [1]
- Năm 2011: Máy tính Watson của hãng IBM đã đánh bại các đối thủ là con người tại Jeopardy. [1]

- Năm 2011: Google Brain đã được phát triển, và mạng deep neuron (deep neural network) của nó có thể học để phát hiện và phân loại nhiều đối tượng theo cách mà một con mèo thực hiện. [1]
- Năm 2012: X Lab của Google phát triển một thuật toán machine learning có khả năng tự động duyệt qua các video trên YouTube để xác định xem video nào có chứa những con mèo. [1]
- Năm 2014: Facebook phát triển DeepFace, một phần mềm thuật toán có thể nhận dạng hoặc xác minh các cá nhân dựa vào hình ảnh ở mức độ giống như con người có thể. [1]
- Năm 2015: Amazon ra mắt nền tảng machine learning riêng của mình. [1]
- Năm 2015: Microsoft tạo ra Distributed Machine Learning Toolkit, trong đó cho phép phân phối hiệu quả các vấn đề machine learning trên nhiều máy tính. [1]
- Năm 2015: Hơn 3.000 nhà nghiên cứu AI và Robotics, được sự ủng hộ bởi những nhà khoa học nổi tiếng như Stephen Hawking, Elon Musk và Steve Wozniak (và nhiều người khác), đã ký vào một bức thư ngỏ để cảnh báo về sự nguy hiểm của vũ khí tự động trong việc lựa chọn và tham gia vào các mục tiêu mà không có sự can thiệp của con người. [1]
- Năm 2016: Thuật toán trí tuệ nhân tạo của Google đã đánh bại nhà vô địch trò chơi Cờ Vây, được cho là trò chơi phức tạp nhất thế giới (khó hơn trò chơi cờ vua rất nhiều). [1]

Vậy thì ta đã tiến gần hơn đến trí tuệ nhân tạo hay chưa? Một số nhà khoa học cho rằng đó thực sự là một câu hỏi vô lý.

Bởi người ta tin rằng máy tính sẽ không thể "nghĩ" được theo phương thức của não bộ loài người. Suy nghĩ này cũng giống như việc so sánh quả ổi và quả lê vậy do việc đặt lên bàn cân giữa trình độ phân tích tính toán và thuật toán của máy tính với trí tuệ con người là hoàn toàn khác nhau.

Bất kể điều đó, tiềm năng của máy tính trong việc quan sát, nhận thức và giao tiếp mới môi trường xung quanh đang phát triển nhanh chóng. Với lượng dữ liệu chúng ta tạo ra tiếp tục tăng lên theo cấp số nhân, khả năng của máy tính trong xử lý, phân tích và học từ những dữ liệu này cũng ngày càng phát triển và mở rộng.

## 1.2 Khái niệm

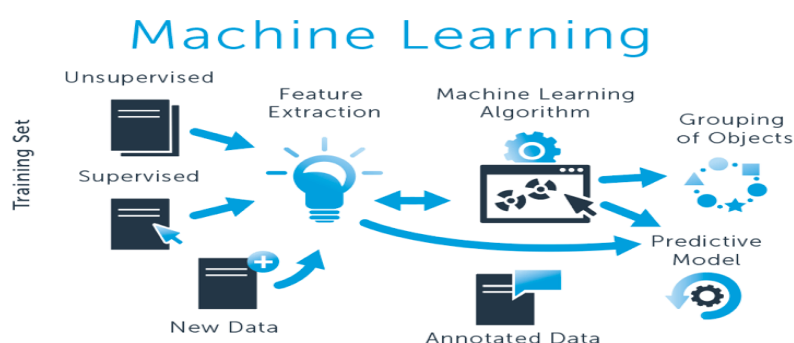
Có nhiều lĩnh vực trong khoa học máy tính và trí tuệ nhân tạo. Học máy, cách gọi khác là machine learning là một trong số đó, chúng chú trọng đến việc sử dụng dữ liệu và thuật toán nhằm mục đích mô phỏng và cải thiện khả năng học hỏi tự động mà không cần chỉ dẫn trực tiếp.

Ngoài ra, học máy cũng có vai trò quan trọng trong lĩnh vực khoa học dữ liệu đang phát triển. Chúng được huấn luyện để thực hiện việc phân loại, dự đoán và khai thác dữ liệu nhờ áp dụng những kỹ thuật thống kê, những thuật toán.

Thông qua việc khám phá các thông tin chi tiết, machine learning đóng vai trò quan trọng trong việc hỗ trợ ra quyết định cho các ứng dụng và doanh nghiệp, có ảnh hưởng sâu rộng đến các chỉ số tăng trưởng. Với sự gia tăng không ngừng của dữ liệu lớn cũng sẽ kéo theo sự gia tăng về nhu cầu tuyển dụng các nhà khoa học dữ liệu. Các chuyên gia này sẽ được yêu cầu định rõ các câu hỏi kinh doanh quan trọng nhất và thu thập dữ liệu cần thiết để giải quyết chúng.

Theo một định nghĩa khác mang tính toán học hơn, học máy sử dụng dữ liệu và các thuật toán đầu vào để có thể tự động giải quyết các vấn đề và không ngừng cải tiến nhằm phát triển các kỹ thuật xử lý mới và tối ưu hơn, giống như cách mà não bộ con người tự học.

Trong sử dụng học máy, có hai mục tiêu chính đó là dự đoán và thống kê do đó hệ thống này được xây dựng với khả năng tự học và tự cải thiện dựa vào các nguyên lý cơ bản của lập trình. Trong một vài tình huống, học máy có khả năng tự đưa ra các giải pháp tối ưu mà không cần lập trình sẵn. Có thể nói, học máy giống như một người lao động có khả năng tự học hỏi, hoàn thiện và tích lũy kinh nghiệm theo thời gian.



Hình 1. 1 Tổng quan về học máy

Những yếu tố quan trọng trong quá trình học máy bao gồm:

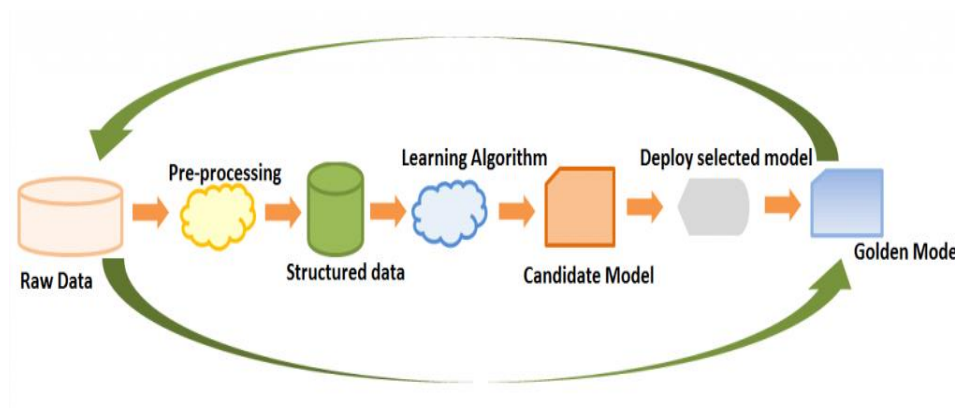
- *Trích xuất đặc trưng (Feature Extraction)*: Đây là quá trình lựa chọn và biến đổi dữ liệu đầu vào thành các đặc trưng thích hợp để sử dụng trong mô hình máy tính. Ví dụ, trong xử lý ngôn ngữ tự nhiên, việc chuyển đổi văn bản thành các biểu đồ từ khóa có thể là một bước trích xuất đặc trưng.
- *Thuật toán Học máy (Machine Learning Algorithm)*: Đây là phần cốt lõi của quá trình học máy, bao gồm các thuật toán và phương pháp để xây dựng mô hình từ dữ liệu. Ví dụ, Support Vector Machines (SVM), Decision Trees, hoặc Neural Networks.
- *Nhóm các đối tượng (Group of Objects)*: Học máy thường được dùng để phân loại hoặc phân cụm các đối tượng thông qua các đặc trưng của chính nó. Ví dụ, trong phân loại hình ảnh, các đối tượng có thể được phân loại thành các lớp khác nhau như "mèo," "chó," "xe hơi," v.v.
- *Mô hình dự đoán (Predictive Model)*: Mô hình học máy học từ dữ liệu huấn luyện và được sử dụng để dự báo kết quả cho dữ liệu mới. Ví dụ, một mô hình dự đoán giá cổ phiếu dựa trên dữ liệu lịch sử.
- *Dữ liệu có nhãn (Annotated Data)*: Dữ liệu có nhãn là tập hợp dữ liệu mà mỗi ví dụ đã được gán nhãn để sử dụng trong việc huấn luyện và kiểm tra mô hình. Ví dụ, các đánh giá sản phẩm trực tuyến được gán nhãn là "tốt" hoặc "xấu" để xây dựng một mô hình đánh giá sản phẩm.

Học máy đã thay đổi cách ta giải quyết nhiều vấn đề phức tạp và cũng đã có những ứng dụng rất quan trọng trong thế giới thực. Ví dụ như hệ thống phát hiện gian lận trong giao dịch tài chính, chatbot trả lời tự động trong dịch vụ khách hàng, và các hệ thống tự động lái xe là các ứng dụng tiêu biểu của học máy.

### **1.3 Cách hoạt động.**

Như đã đề cập trong phần 1.1.2, có thể hình dung học máy giống như một người lao động có khả năng tự học hỏi, phát triển và trở nên thành thạo hơn theo thời gian. Vậy cụ thể thì quá trình hoạt động của học máy diễn ra như thế nào và cơ chế nào đứng sau khả năng ưu việt đó?

Tiến trình này bao gồm định nghĩa vấn đề, đối chiếu các phương án từ tập dữ liệu đầu vào và đưa ra kết quả thích hợp. Sau mỗi lần thực hiện, học máy có khả năng đưa ra đánh giá và tích lũy kinh nghiệm để làm phong phú tập dữ liệu gốc. Quá trình học tập của ML được mô tả như hình 1.2.



*Hình 1. 2 Quá trình tự học của học máy*

Quá trình này là quá trình được khép kín, tự vận hành và tự chuyển hóa liên tục. Các bước quan trọng của nó sẽ bao gồm:

- Tiền xử lý dữ liệu: Dữ liệu đầu vào sẽ được sàng lọc để loại bỏ các giá trị không cần thiết và phân loại các nhóm theo cấu trúc nhất định. Dữ liệu thô trong từng bối cảnh khác nhau có thể rất lộn xộn, nhưng học máy có thể tự động xử lý và sắp xếp lại nhờ cơ chế sàng lọc. Cách thức này hỗ trợ nhận diện dữ liệu cần thiết, tập hợp chúng lại và loại bỏ những giá trị dư thừa nhằm giảm chi phí “chế biến”.
- Áp dụng thuật toán để tìm ra các giải pháp tối ưu nhất: Các dữ liệu đã được phân loại sẽ được chuyển đến quá trình xử lý để phân tích và tổng hợp lại nhằm tạo ra các phương pháp mà học máy cho là thích hợp nhất đối với vấn đề hiện tại. Nói đơn giản, đây là giai đoạn hệ thống tổ hợp các dữ liệu đầu vào và sử dụng các thuật toán để hình thành, đánh giá, và lựa chọn các phương án khả thi nhất. Nói ngắn gọn, đây là giai đoạn hệ thống tổng hợp những dữ liệu đầu vào và sử dụng các thuật toán để xây dựng, đánh giá và lựa chọn các giải pháp tốt nhất.

Các giải pháp khả thi đó sẽ được kiểm tra thông qua những điều kiện giả định của các yêu cầu đầu vào sau khi đã thực hiện xong quá trình đánh giá. Những phương án bộc lộ yếu điểm sẽ bị loại bỏ. Kết quả cuối cùng là phương án được thử nghiệm và kiểm chứng

nhiều lần, mà học máy cho là phù hợp nhất với bài toán đang giải.

- Thực hiện giải pháp hiệu quả nhất và không ngừng cải thiện: Giải pháp sau cùng sẽ được áp dụng vào trong thực tiễn và trong lúc áp dụng thì học máy sẽ liên tục ghi chép, theo dõi các sự cố nảy sinh, bao gồm cả điểm mạnh và điểm yếu chưa được phát hiện trong quá trình đánh giá. Tiếp đến, hệ thống sẽ cập nhật cho bộ dữ liệu ban đầu cách giải quyết mới nhằm tạo ra những giải pháp chính xác hơn theo thời gian.

Sau khi trải qua nhiều lần cập nhật, học máy sẽ thiết lập các tiêu chuẩn cho các phương án đầu ra nhằm đạt được hiệu quả với tốc độ và độ chính xác cao hơn, ít rủi ro hơn. Đó gọi là sự tăng cường của mô hình học máy

## **1.4 Phân loại**

Dựa vào mục đích, cấu trúc và thuật toán tạo nên chúng thì machine learning có thể chia thành bốn loại như sau: học có giám sát, học không giám sát, học bán giám sát và học tăng cường.

### **1.4.1 Học có giám sát (*Supervised Learning*)**

Học máy có giám sát là một kỹ thuật mà trong đó các thuật toán được huấn luyện trên các tập dữ liệu đã được gắn nhãn, cho phép chúng phân loại dữ liệu hoặc dự đoán kết quả chính xác.

Trong quá trình học có giám sát, hệ thống sẽ được cung cấp một lượng dữ liệu lớn có gắn nhãn, chẳng hạn như các hình ảnh về các số viết bằng tay được chú thích rõ ràng để chỉ ra con số này tương ứng với hình nào. Sau khi thu thập đủ dữ liệu ví dụ, hệ thống học có giám sát sẽ học cách nhận diện các mẫu điểm ảnh và hình dạng liên quan đến mỗi con số, cuối cùng có thể phân biệt chính xác các con số như 1 và 5, hoặc 3 và 7.

Thế nhưng bởi vì một vài hệ thống cần đến hàng triệu ví dụ để thành thạo một tác vụ cụ thể nên việc huấn luyện các hệ thống này yêu cầu một lượng dữ liệu gắn nhãn cực kỳ lớn. Do đó, bộ dữ liệu dùng để huấn luyện các hệ thống này có thể sẽ rất khổng lồ (điển hình như cơ sở dữ liệu của IMDb chứa hơn 7 triệu thông tin của phim và chương trình truyền hình, trang Wikipedia hiện tại đã có hơn 6 triệu bài viết trên nhiều ngôn ngữ khác nhau và Amazon bán hơn 350 triệu sản phẩm của họ trên toàn cầu)

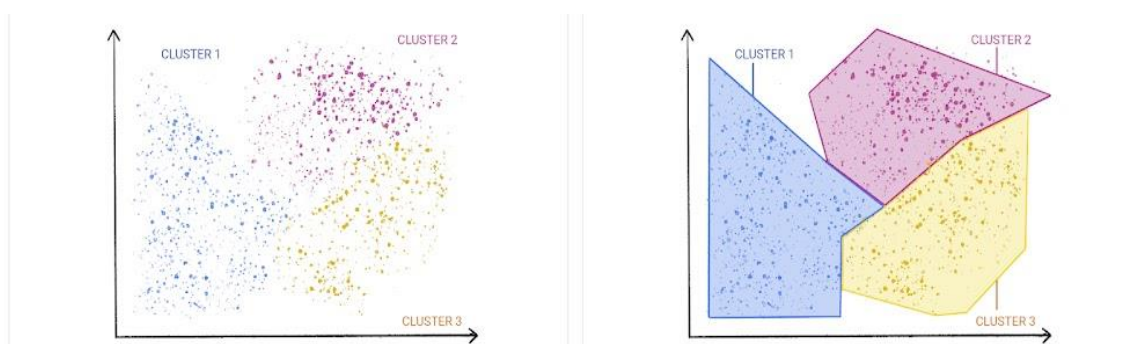
Quá trình gắn nhãn cho các bộ dữ liệu sử dụng trong học có giám sát thường được thực hiện thông qua các dịch vụ tuyển dụng tự do, như CrowdFlower (Figure Eight), Clickworker và Amazon Mechanical Turk. Điển hình như chỉ trong hơn 2 năm với sự đóng góp của gần 50.000 người, phần lớn trong số họ được tuyển dụng qua Amazon Mechanical Turk mà ImageNet đã được xây dựng.

#### 1.4.2 Học không giám sát (*Unsupervised Learning*)

Trái ngược lại với học có giám sát, khi muốn phân lớp thì con người phải cung cấp nhãn để máy tính hiểu thì học không giám sát không yêu cầu điều đó. Học máy không giám sát giao nhiệm vụ cho các thuật toán để xác định các mẫu trong tập dữ liệu, tìm kiếm những điểm tương đồng giữa chúng và tự động phân loại dữ liệu có điểm tương đồng vào các nhóm tương ứng.

Chẳng hạn, AirBNB phân loại các nhà cho thuê theo khu vực, hoặc GoogleNews tập hợp các bài viết có chủ đề tương tự nhau mỗi ngày.

Thế nhưng những kỹ thuật học không giám sát thì lại không thiết kế với mục đích xác định những loại dữ liệu cụ thể, tường minh mà với mục đích tìm ra các đặc trưng có thể được nhóm lại với nhau dựa trên sự tương đồng hoặc phát hiện những điểm dữ liệu bất thường. Mặc dù bản thân thuật toán không hiểu các mẫu này dựa trên bất kỳ thông tin nào con người đã cung cấp trước đó, nhưng sau đó ta có thể xem qua các nhóm dữ liệu và cố gắng phân loại chúng dựa trên sự hiểu biết của bạn về tập dữ liệu.



Hình 1. 3 Quá trình tự phân cụm của học không giám sát

Học không giám sát chủ yếu sẽ đảm nhận các nhiệm vụ sau: phân cụm, quy tắc kết hợp và giảm kích thước dữ liệu.

Phân cụm là một kỹ thuật để khám phá dữ liệu thô, chưa được gắn nhãn và chia nó thành



các nhóm (hoặc cụm) dựa trên những điểm tương đồng hoặc khác biệt. Được áp dụng rộng rãi trong nhiều lĩnh vực, bao gồm phân khúc khách hàng, phát hiện gian lận và phân tích hình ảnh. Các thuật toán phân cụm chia dữ liệu thành các nhóm tự nhiên bằng cách tìm các cấu trúc hoặc mẫu tương tự trong dữ liệu chưa được phân loại.

### **Phân cụm**

Phân cụm là một trong những kỹ thuật học không giám sát phổ biến nhất. Có một số loại thuật toán học không giám sát được sử dụng để phân cụm, bao gồm độc quyền, chồng chéo, phân cấp và xác suất.

*Phân cụm độc quyền:* Dữ liệu được nhóm theo cách mà một điểm dữ liệu duy nhất chỉ có thể tồn tại trong một cụm. Điều này còn được gọi là phân cụm "cứng". Một ví dụ phổ biến về phân cụm độc quyền là thuật toán phân cụm K-means, nó phân chia các điểm dữ liệu thành số cụm K do người dùng xác định.

*Phân cụm chồng chéo:* Dữ liệu được nhóm theo cách mà một điểm dữ liệu có thể tồn tại trong hai hoặc nhiều cụm với mức độ thành viên khác nhau. Điều này còn được gọi là phân cụm "mềm".

*Phân cụm theo cấp bậc:* Dữ liệu được chia thành các cụm riêng biệt dựa trên những điểm tương đồng, sau đó được hợp nhất và sắp xếp nhiều lần dựa trên mối quan hệ phân cấp của chúng. Có hai loại phân cụm theo cấp bậc chính: phân cụm kết tụ và phân cụm phân chia. Phương pháp này còn được gọi là HAC - phân tích cụm phân cấp.

*Phân cụm theo xác suất:* Dữ liệu được nhóm thành các cụm dựa trên xác suất của từng điểm dữ liệu thuộc mỗi cụm. Cách tiếp cận này khác với các phương pháp khác, nhóm các điểm dữ liệu dựa trên điểm tương đồng của chúng với các điểm khác trong một cụm.

### **Sự kết hợp**

Khai thác quy tắc kết hợp là một phương pháp dựa trên quy tắc để khám phá mối quan hệ thú vị giữa các điểm dữ liệu trong một tập dữ liệu lớn. Các thuật toán học không giám sát tìm kiếm các liên kết “nếu”-“thì” để khám phá các mối tương quan và sự xuất hiện đồng thời trong dữ liệu cũng như các kết nối khác nhau giữa các đối tượng dữ liệu.

Nó được sử dụng phổ biến nhất để phân tích giỏ bán lẻ hoặc bộ dữ liệu giao dịch để thể hiện tần suất mua một số mặt hàng nhất định cùng nhau. Các thuật toán này khám phá các mô hình mua hàng của khách hàng và các mối quan hệ ẩn giấu trước đây giữa các sản phẩm, giúp cung cấp thông tin cho các công cụ đề xuất hoặc các cơ hội bán kèm khác. Ta có thể quen thuộc nhất với những quy tắc này từ phần “Thường xuyên mua cùng nhau” và “Những người đã mua mặt hàng này cũng đã mua” trên cửa hàng bán lẻ trực tuyến yêu thích của mình.

Các luật kết hợp cũng thường được sử dụng để tổ chức các tập dữ liệu y tế cho chẩn đoán lâm sàng. Việc sử dụng các quy tắc kết hợp và học máy không giám sát có thể giúp bác sĩ xác định xác suất chẩn đoán cụ thể bằng cách so sánh mối quan hệ giữa các triệu chứng từ các trường hợp bệnh nhân trong quá khứ.

Thông thường, thuật toán Apriori được sử dụng rộng rãi nhất cho việc học quy tắc kết hợp để xác định các bộ sưu tập vật phẩm hoặc bộ vật phẩm có liên quan. Tuy nhiên, các loại khác được sử dụng, chẳng hạn như thuật toán tăng trưởng Eclat và FP.

### **Giảm kích thước**

Giảm kích thước là một kỹ thuật học máy không giám sát nhằm giảm số lượng tính năng hoặc kích thước của tập dữ liệu. Nhiều dữ liệu hơn thường tốt hơn cho machine learning nhưng nó cũng có thể khiến việc trực quan hóa dữ liệu trở nên khó khăn hơn.

Giảm kích thước trích xuất các tính năng quan trọng từ tập dữ liệu, giảm số lượng các tính năng ngẫu nhiên hoặc không liên quan hiện có. Phương pháp này sử dụng thuật toán phân tích thành phần nguyên tắc (PCA) và phân rã giá trị số ít (SVD) để giảm số lượng dữ liệu đầu vào mà không ảnh hưởng đến tính toàn vẹn của các thuộc tính trong dữ liệu gốc.

#### ***1.4.3 Học bán giám sát (Semi – Supervisor Learning).***

Các bài toán trong Semi-Supervised Learning xảy ra khi có một lượng dữ liệu lớn, nhưng chỉ có một phần nhỏ trong số đó được gắn nhãn. Đây là nhóm bài toán nằm giữa hai loại học máy được đề cập ở trên.

Học có giám sát có một số hạn chế, đó là:

- Khá chậm chạp vì yêu cầu người là chuyên gia để dán nhãn thủ công cho từng dữ liệu huấn luyện.
- Tốn kém chi phí vì phải huấn luyện mô hình trên khối lượng lớn dữ liệu được gán nhãn thủ công để đạt được độ chính xác cao.

Bên cạnh đó, học không giám sát là cách tiết kiệm chi phí hơn so với nó nhưng không hẳn là giải pháp hiệu quả. Bởi vì nó có phạm vi ứng dụng hạn chế, thường là dùng cho bài toán phân cụm và cho kết quả kém chính xác hơn.

Học bán giám sát là một phương pháp nằm giữa học có giám sát và học không giám sát và nó giải quyết những thách thức chính của cả hai. Với học bán giám sát, chúng ta có thể huấn luyện mô hình ban đầu với dữ liệu được gán nhãn, sau đó áp dụng mô hình này để xử lý các dữ liệu lớn hơn chưa được gán nhãn.

- Khác với học không giám sát, học bán giám sát giải quyết được nhiều nhiệm vụ khác nhau từ phân loại, hồi quy, phân cụm và cả kết hợp cả ba.
- Phương pháp này sử dụng một lượng nhỏ dữ liệu được gán nhãn cùng với một lượng lớn dữ liệu chưa được gán nhãn, giúp giảm thời gian và chi phí gán nhãn thủ công so với học có giám sát.

Vì dữ liệu không được gán nhãn rất phong phú, cho nên không khó để lấy được chúng và nếu có phí thì chúng ở mức rất rẻ. Hãy xem xét một ví dụ sau: giả sử, một công ty A có 10 triệu người dùng. Và họ đã phân tích 5% tất cả giao dịch của 10 triệu người dùng đó để phân loại chúng vào lừa đảo hay không lừa đảo, trong khi đó phần giao dịch còn lại không được gán nhãn là lừa đảo hay không. Trong trường hợp này, học bán giám sát cho phép duyệt qua tất cả toàn bộ giao dịch để xác minh giao dịch bất hợp lý mà không cần đội ngũ nhân viên gán nhãn hoặc giảm đi độ chính xác của mô hình.

Một ví dụ điển hình khác trong nhóm này là khi chỉ một phần nhỏ ảnh hoặc văn bản được gán nhãn (ví dụ như ảnh của con người, động vật, hoặc các văn bản về khoa học, chính trị), trong khi phần lớn ảnh/ văn bản khác chưa có nhãn được thu thập từ internet. Thực tế cho thấy nhiều bài toán Machine Learning thuộc nhóm này do việc thu thập dữ liệu có nhãn tốn rất nhiều thời gian và chi phí. Nhiều loại dữ liệu đòi hỏi sự can thiệp của chuyên gia để gán nhãn (ví dụ như ảnh y học). Ngược lại, dữ liệu chưa có nhãn có

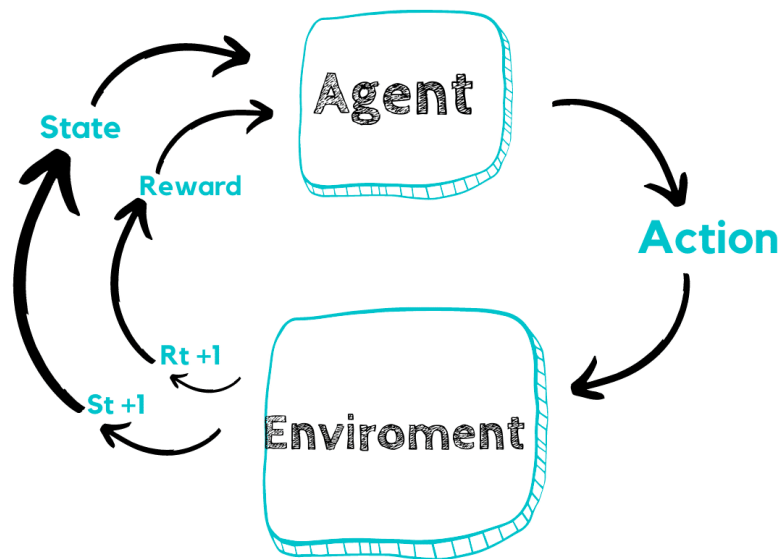
thể được thu thập từ internet với chi phí thấp hơn.

#### 1.4.4 Học tăng cường (Rainforcement Learning)

Học tăng cường là một phương pháp học máy mà trong đó hệ thống tự động tiếp thu và nâng cao hành vi bằng cách tương tác với môi trường. Quá trình này căn cứ vào nguyên lý học từ phản hồi (feedback) và thưởng (reward) để tối ưu hóa một hàm phần thưởng đã được định nghĩa sẵn.

Học tăng cường thực hiện như một chỉ báo cho các hành động xấu và tốt. Mục đích duy nhất của học tăng cường là phát triển ra một mô hình hành vi hiệu quả nhằm tối đa hóa tổng phần thưởng tích lũy của đại lý. Phương pháp này cho phép tác nhân đưa ra các quyết định để đạt được mục tiêu mà không cần sự can thiệp trực tiếp từ con người hoặc lập trình chi tiết.

Mô hình tương tác trong học tăng cường như sau:



Hình 1. 4 Mô hình tương tác giữa các thuật ngữ trong học tăng cường

Trong đó:

- Agent (tác nhân): người ra quyết định và là người học duy nhất.
- Action (hành động): danh sách các hành động của tác nhân.
- Enviroment (môi trường): một thế giới môi trường nơi tác nhân tìm hiểu và quyết định hành động.
- State (trạng thái): tình hình hiện tại của tác nhân trong môi trường.

- **Reward (phần thưởng):** Đây là giá trị mà môi trường cung cấp cho tác nhân sau mỗi hành động, thường là một số vô hướng, để tác nhân có thể học và cải thiện hành vi.
- **Chính sách (policy):** Là chiến lược của tác nhân để ra quyết định và ứng xử trong môi trường, với mục tiêu tối đa hóa phần thưởng hoặc hiệu suất.
- **Hàm giá trị (value function):** Được sử dụng để đánh giá mức độ tốt của một hành động hoặc trạng thái cụ thể trong môi trường, giúp tác nhân đưa ra các quyết định dựa trên những ước tính về giá trị tương lai.
- **Mô hình (model):** Biểu diễn của môi trường mà tác nhân tương tác để học và dự đoán. Mô hình có thể là một phần của hệ thống mà tác nhân cần phải hiểu để đưa ra các hành động phù hợp và tối ưu.

Trong học tăng cường, các nhà phát triển đã phát triển một phương pháp để khen thưởng các hành vi tốt và trừng phạt các hành vi xấu. Phương pháp này thường gán các giá trị dương cho các hành động được khuyến khích để tác nhân có xu hướng thực hiện chúng, và gán các giá trị âm cho các hành vi xấu để giảm thiểu chúng. Quá trình này giúp tác nhân tối ưu hóa tổng lượng phần thưởng để đạt được một giải pháp tối ưu. [2]

Theo thời gian, tác nhân học cách tránh các hành vi tiêu cực và tìm kiếm những hành vi tích cực hơn. Phương pháp học này đã được áp dụng trong trí tuệ nhân tạo (AI) như một phương pháp để hướng dẫn học máy không giám sát thông qua phần thưởng và hình phạt. [2]

Ví dụ sau đây sẽ giúp ta có cái nhìn rõ ràng, tổng quát hơn về cách hoạt động của học tăng cường: Ta đang huấn luyện một chú chó ngồi xuống theo hiệu lệnh, khi mà chú chó thực hiện đúng theo hiệu lệnh thì ta thưởng cho nó đồ ăn mà nó thích (đây chính là reward). Từ đó chú chó sẽ hiểu được rằng, mỗi khi làm theo hiệu lệnh thì nó sẽ được thưởng.

## **1.5 Ưu và nhược điểm**

### **1.5.1 Ưu điểm**

Machine learning là một lĩnh vực quan trọng trong việc phát triển khoa học và công nghệ nhờ vào những ưu điểm sau:

- Có khả năng nhận diện xu hướng và mẫu dữ liệu mà con người có thể bỏ qua.
- Có thể hoạt động tự động sau khi thiết lập, ví dụ như trong phần mềm an ninh mạng, máy học có thể liên tục giám sát và phát hiện các hoạt động bất thường trong lưu lượng mạng mà không cần sự can thiệp của người quản trị.
- Kết quả của máy học có thể trở nên chính xác hơn theo thời gian.
- Có thể thao tác với nhiều kiểu dữ liệu khác nhau trên các nền tảng dữ liệu linh động, có quy mô lớn và rắc rối.

### **1.5.2 Nhược điểm**

Song song với những ưu điểm tiên tiến mà ML mang lại cho chúng ta, bên cạnh đó vẫn còn tồn tại một vài nhược điểm:

- Sự thổi phồng quá lớn từ báo chí khiến nhiều người có suy nghĩ sai lầm là AI có thể làm được mọi thứ. Điều đó dẫn đến sự thất bại của nhiều dự án và hao tổn tài nguyên.
- Đòi hỏi trình độ kỹ thuật viên khá cao để tạo ra các mô hình có tính chính xác và áp dụng cao.
- Yêu cầu phần cứng đủ nhanh.
- Thời gian đào tạo dành cho các bộ dữ liệu lớn khá lâu.

## **1.6 Các ứng dụng**

Học máy còn là lực lượng nòng cốt trong mô hình kinh doanh của một số công ty, ví dụ như các gợi ý của Facebook hoặc công cụ dịch tự động của Google. Dưới đây là một số lĩnh vực chính đang ứng dụng Machine Learning:

*Các thuật toán:* Machine Learning được ứng dụng để tạo ra các thuật toán đề xuất. Đó là những thứ mà bạn thấy hàng ngày trên internet như gợi ý của Netflix và YouTube, thông tin trên nguồn cấp dữ liệu của Facebook, gợi ý mua hàng... Các thuật toán đang cố gắng tìm hiểu sở thích của người dùng để hiển thị chính xác những nội dung trên màn hình.

*Phân tích hình ảnh và phát hiện đối tượng:* Học máy có thể phân tích hình ảnh để trích xuất nhiều loại thông tin khác nhau, chẳng hạn như nhận diện và phân biệt giữa các cá nhân, hay đếm số lượng ô tô trong bãi đậu xe.

*Phát hiện gian lận:* Machine Learning có thể phân tích các mẫu, chẳng hạn như cách ai đó thường chi tiêu hoặc nơi họ thường mua sắm để xác định các giao dịch thẻ tín dụng có khả năng gian lận, các nỗ lực đăng nhập hoặc spam email.

*Tổng đài hỗ trợ tự động hoặc chatbot:* Nhiều công ty đang sử dụng chatbot trực tuyến, cho phép khách hàng tương tác với máy thay vì con người. Các thuật toán này sử dụng học máy để phân tích các bản ghi cuộc trò chuyện trước đây và đưa ra các phản hồi phù hợp.

*Xe ô tô tự lái:* Công nghệ đằng sau ô tô tự lái chủ yếu dựa trên học máy, đặc biệt là học sâu (Deep Learning).

*Chẩn đoán hình ảnh trong y tế:* Các chương trình học máy có thể được huấn luyện để kiểm tra và phân tích các hình ảnh y tế hoặc các thông tin khác nhằm hỗ trợ chẩn đoán. Thông qua sự so sánh với các dữ liệu trong quá khứ, nó sẽ tìm kiếm các dấu hiệu bệnh tật nhất định. Đây là một trong những công cụ có thể dự đoán nguy cơ ung thư nhanh chóng.

## **1.7 Môi trường cài đặt**

### **1.7.1 Ngôn ngữ lập trình python**

Ngôn ngữ lập trình máy tính bậc cao – Python, thường được sử dụng để xây dựng trang web và phát triển phần mềm, thực hiện phân tích dữ liệu và tự động hóa các nhiệm vụ. Nó được Guido van Rossum phát triển vào cuối những năm 1980 tại Viện Nghiên cứu Quốc gia về Toán học và Khoa học Máy tính ở Hà Lan. Python là ngôn ngữ có mục đích chung, có khả năng tạo ra nhiều chương trình khác nhau và không chuyên biệt cho bất kỳ vấn đề cụ thể nào.

Cho tới nay, trải qua hàng chục năm phát triển thì Python đã trở thành một trong những ngôn ngữ lập trình thông dụng nhất trên toàn thế giới. Python còn là ngôn ngữ chính thức tại Google. Phiên bản hiện tại của Python là 3.12.

### **1.7.2 Các thư viện**

#### ***Numpy***

NumPy (Numeric Python) là một thư viện toán học rất phổ biến và mạnh mẽ trong

Python. Thư viện này được trang bị các hàm số đã được tối ưu, cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là xử lý dữ liệu ma trận và mảng lớn với tốc độ nhanh hơn nhiều so với việc sử dụng Python đơn thuần. NumPy là một trong những thư viện quan trọng của Python, đặc biệt là trong nghiên cứu về các phép toán số học.

NumPy được phát triển bởi Jim Hugunin. Phiên bản ban đầu của NumPy là Numarray, được phát triển với một số tính năng bổ sung. Năm 2005, Travis Oliphant đã tạo ra thư viện NumPy bằng cách hợp nhất các tính năng của Numarray và thư viện Numeric.

Sử dụng NumPy, lập trình viên có thể thực hiện các thao tác sau:

- Các phép toán toán học và logic trên mảng.
- Các biến đổi Fourier và các quy trình để thao tác shape.
- Các phép toán liên quan đến đại số tuyến tính. NumPy tích hợp sẵn các hàm cho đại số tuyến tính và tạo số ngẫu nhiên.

NumPy thường được sử dụng kết hợp với các gói như SciPy (Python Scientific) và Matplotlib (thư viện vẽ đồ thị). Sự kết hợp này rất phổ biến để thay thế cho MatLab, một nền tảng phổ biến trong tính toán kỹ thuật. Tuy nhiên, Python đã thay thế MatLab như một ngôn ngữ lập trình hiện đại và hoàn thiện hơn. Điều quan trọng là NumPy là một thư viện mã nguồn mở và miễn phí, trong khi MatLab là một thư viện mã nguồn đóng và yêu cầu phải trả phí.

### ***Pandas***

Thư viện pandas trong Python là một thư viện mã nguồn mở mạnh mẽ, được sử dụng rộng rãi để xử lý và phân tích dữ liệu. Đây là một công cụ thiết yếu trong ngôn ngữ lập trình Python, được áp dụng rộng rãi trong cả nghiên cứu và phát triển các ứng dụng về lĩnh vực khoa học dữ liệu. Thư viện này sử dụng cấu trúc dữ liệu chính là DataFrame và cung cấp nhiều chức năng để xử lý và làm việc trên cấu trúc này. Chính nhờ sự linh hoạt và hiệu quả, pandas đã trở nên phổ biến và được sử dụng rộng rãi.

### ***Tensorflow***

TensorFlow là một thư viện mã nguồn mở end-to-end được thiết kế chủ yếu cho các ứng



dụng machine learning. Đó là một thư viện toán học ký hiệu áp dụng luồng dữ liệu và lập trình có khả năng phân biệt để triển khai các nhiệm vụ đa dạng, chú trọng vào đào tạo và suy luận các mạng nơ-ron sâu (deep neural network). Các nhà phát triển có thể xây dựng các ứng dụng học máy bằng cách sử dụng các công cụ, thư viện và tài nguyên từ nhiều cộng đồng khác nhau.

TensorFlow của Google hiện tại là thư viện học sâu được biết đến rộng rãi nhất trên toàn cầu. Google tích hợp công nghệ máy học này vào các sản phẩm của mình nhằm tối ưu hóa công cụ tìm kiếm, dịch thuật, chú thích hình ảnh và các đề xuất khác.

Người dùng Google có thể trải nghiệm tính năng tìm kiếm thông minh hơn và nhanh chóng hơn nhờ vào trí tuệ nhân tạo. Khi nhập từ khóa vào thanh tìm kiếm, Google sẽ đề xuất các từ khóa liên quan và có thể tiếp theo để giúp người dùng tìm kiếm hiệu quả hơn.

Google muốn áp dụng công nghệ máy học để khai thác bộ dữ liệu khổng lồ của họ nhằm mang đến trải nghiệm tối ưu cho người dùng. Ba nhóm người sử dụng chủ yếu công nghệ này bao gồm các nhà nghiên cứu, nhà khoa học dữ liệu và lập trình viên. Tất cả đều có thể sử dụng cùng một bộ công cụ để hợp tác và tối đa hóa hiệu quả công việc của mình.

Google không chỉ sở hữu dữ liệu mà còn là chủ sở hữu của máy tính lớn nhất thế giới. Do đó, họ đã xây dựng TensorFlow để mở rộng quy mô. TensorFlow là một thư viện do Nhóm Google Brain phát triển nhằm gia tăng tốc độ trong lĩnh vực học máy và nghiên cứu mạng nơ-ron sâu.

Kiến trúc Tensorflow hoạt động trong ba phần:

- Xử lý trước dữ liệu
- Xây dựng mô hình
- Đào tạo và ước tính mô hình

TensorFlow được đặt tên là như vậy vì nó nhận đầu vào dưới dạng một mảng đa chiều, được gọi là Tensor. Bạn có thể xây dựng một lưu đồ hoạt động (hay còn gọi là đồ thị -

Graph) để thực hiện các hoạt động trên đầu vào đó. Đầu vào được đưa vào một đầu của đồ thị và sau đó chảy qua một chuỗi các hoạt động trước khi đi ra ở đầu kia dưới dạng đầu ra. Đây là lý do vì sao TensorFlow được đặt tên như thế, vì Tensor đi qua nó và chảy qua một danh sách các hoạt động trước khi ra phía bên kia.

Mô hình có thể được đào tạo và sử dụng trên cả GPU và CPU. GPU ban đầu được thiết kế cho trò chơi điện tử. Vào cuối năm 2010, các nhà nghiên cứu tại Stanford đã phát hiện rằng GPU rất giỏi trong các phép toán ma trận và đại số, giúp thực hiện các loại tính toán này một cách nhanh chóng. Học sâu dựa trên rất nhiều phép nhân ma trận. TensorFlow thực hiện các phép nhân ma trận nhanh chóng nhờ việc sử dụng C++. Ngoài ra, TensorFlow có thể được truy cập và điều khiển bởi nhiều ngôn ngữ khác nhau, chủ yếu là Python.

### ***Keras***

Keras là một thư viện mạng nơ-ron sâu trong Python, hỗ trợ đa dạng các kiểu mạng nơ-ron và có khả năng xuất sắc trong xử lý dữ liệu, trực quan hóa và các tính năng khác. Thư viện này được cấu trúc mô-đun, giúp dễ dàng phát triển các ứng dụng sáng tạo.

### ***1.7.3 Visual Studio Code***

Visual Studio Code là một ứng dụng để biên tập và soạn thảo mã nguồn, hỗ trợ trong quá trình xây dựng và thiết kế dự án. Viết tắt là VS Code, nó hoạt động trên các nền tảng như Windows, macOS và Linux. Ngoài ra, trình soạn thảo này cũng tương thích với các thiết bị có cấu hình trung bình, giúp người dùng sử dụng một cách dễ dàng.

Visual Studio Code cung cấp nhiều tính năng hỗ trợ debug đa dạng, tích hợp với Git, có tô màu cú pháp (Syntax Highlighting). Đặc biệt, nó cung cấp tính năng tự động hoàn thành mã thông minh (IntelliSense), Snippets và khả năng cải tiến mã nguồn. Visual Studio Code cho phép lập trình viên thay đổi giao diện (Theme), phím tắt và nhiều tùy chọn khác nhờ vào khả năng tùy chỉnh của nó.

## CHƯƠNG 2 MÔ HÌNH HỌC MÁY TRONG VIỆC DỰ ĐOÁN

### 2.1 Mô hình hồi quy tuyến tính Logistic

#### 2.1.1 Hồi quy Logistic là gì ?

Phân tích hồi quy là một trong những kỹ thuật thống kê trong học máy được dùng trong việc phân tích mối tương quan giữa một biến đầu vào (nhãn) với một hoặc nhiều biến đầu ra (đặc trưng). Do đó, dự báo hoặc diễn tả biến đầu ra dựa trên các biến đầu vào là mục đích chủ yếu của phương pháp này.

Trong phân tích hồi quy thì biến đầu ra là biến mà ta muốn ước tính hay bao quát hơn. Còn các biến đầu vào là những nhân tố mà ta cho là có tác động đến biến đầu ra. Kỹ thuật này hay được triển khai bằng cách xác định một mô hình hồi quy để diễn giải mối tương quan giữa biến phụ thuộc và biến độc lập.

Ví dụ về phân tích hồi quy: Với bài toán "chẩn đoán ung thư vú", thì "chẩn đoán ung thư vú" sẽ là biến phụ thuộc. Các yếu tố độc lập có thể bao gồm diện tích khối u, chu vi khối u, kết cấu khối u, bán kính khối u,...

Hồi quy Logistic là một thuật toán học máy có giám sát nhằm hoàn thành các nhiệm vụ phân loại nhị phân bằng cách dự đoán xác suất của một kết quả, sự kiện hoặc quan sát mà một mẫu dữ liệu thuộc về một lớp nhất định. Nó thực hiện điều này bằng cách sử dụng một hàm Sigmoid để ánh xạ mọi giá trị đầu vào thành một giá trị xác suất trong khoảng từ 0 đến 1. Mô hình đưa ra kết quả nhị phân hoặc phân đôi được giới hạn ở hai kết quả có thể xảy ra: có/không, 0/1 hoặc đúng/sai.

Trong hồi quy tuyến tính ta dựa vào một hàm hồi quy giả thuyết  $h_W(X) = W^T X$  để dự báo biến mục tiêu  $y$ . Do giá trị của biến phụ thuộc trong hồi quy Logistic có thể không nằm trong khoảng  $[0,1]$ , ta cần một hàm số để ánh xạ giá trị dự báo vào không gian xác suất  $[0,1]$ . Đồng thời, hàm này cũng giúp tạo tính phi tuyến cho mô hình hồi quy, cải thiện khả năng phân loại giữa hai nhóm. Đó gọi là hàm Sigmoid.

Hồi quy Logistic phân tích mối quan hệ giữa một hoặc nhiều biến độc lập và phân loại

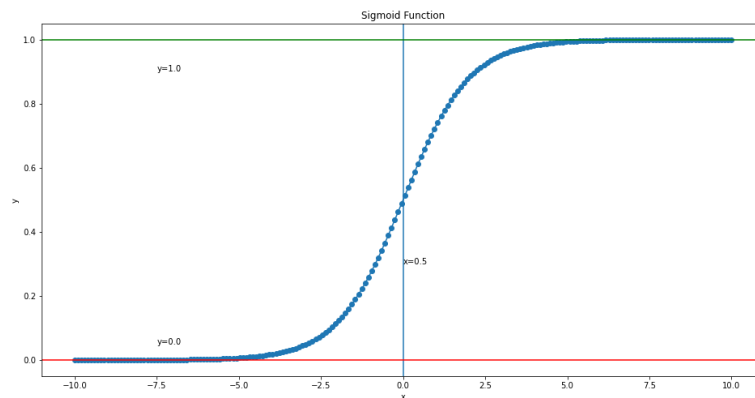
dữ liệu thành các nhóm rời rạc. Nó được sử dụng rộng rãi trong mô hình dự đoán, trong đó mô hình ước tính xác suất toán học về việc một thể hiện có thuộc một danh mục cụ thể hay không.

### 2.1.2 Hàm Sigmoid

Mô hình hồi quy Logistic mở rộng ý tưởng từ hồi quy tuyến tính sang các bài toán phân loại. Đầu ra từ hàm tuyến tính được đưa qua hàm Sigmoid để tính toán phân phối xác suất của dữ liệu. Hàm Sigmoid được sử dụng đặc biệt trong các bài toán phân loại nhị phân, có công thức như sau:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2-1)$$

Với đồ thị của nó trong tọa độ Oxy như sau:



Hình 2. 1 Đồ thị hàm Sigmoid trong trục Oxy

Ta thấy rằng hàm *Sigmoid* có dạng đường cong chữ S và là một hàm đơn điệu tăng. Chính vì điều này, nó còn được biết đến với tên gọi khác là hàm chữ S. Nhiều tài liệu cũng gọi hàm này là hàm *Logistic*, đại diện cho mô hình hồi quy *Logistic*.

Ngoài ra, dễ dàng nhận thấy rằng:

$$\lim_{x \rightarrow +\infty} \sigma(x) = \lim_{x \rightarrow +\infty} \frac{1}{1 + e^{-x}} = 1 \quad (2-2)$$

$$\lim_{x \rightarrow -\infty} \sigma(x) = \lim_{x \rightarrow -\infty} \frac{1}{1 + e^{-x}} = 0 \quad (2-3)$$

Do đó, hàm Sigmoid rất thích hợp để sử dụng trong các bài toán phân loại hoặc nhận dạng. Với hai biến đầu vào là  $x = (x_1, x_2)$  ta thu được một hàm hồi quy sau:

$$\hat{y} = g(x) = w_0 + w_1 + w_2 = W^T x \quad (2-4)$$

Xác suất để sự kiện  $y = 1$  xảy ra với đầu vào  $X$  và trọng số  $W$  được tính theo công thức:

$$P(y = 1 | x; W) = \sigma(W^T x) = \frac{1}{1 + e^{-W^T x}} \quad (2-5)$$

### 2.1.3 Đường phân chia của hàm Sigmoid

Trong bài toán phân loại nhị phân, ta thường lựa chọn một ngưỡng xác suất để quyết định dự báo nhãn cho một quan sát. Giả sử chúng ta chọn ngưỡng xác suất là 0.5. Khi đó, dự báo nhãn sẽ được quyết định như sau: nếu xác suất dự báo (tính bằng hàm Sigmoid) lớn hơn hoặc bằng 0.5, thì quan sát đó sẽ được dự báo là nhãn positive (1); ngược lại, nếu xác suất nhỏ hơn 0.5, thì quan sát đó sẽ được dự báo là nhãn negative (0).

$$\begin{cases} 0 & \text{nếu } P(y = 1 | x; W) \leq 0.5 \\ 1 & \text{nếu } P(y = 1 | x; W) > 0.5 \end{cases} \quad (2-6)$$

Trong trường hợp  $y = 1$ :

$$\begin{aligned} h_W(x) &> 0.5 \\ \Leftrightarrow \frac{1}{1 + e^{-W^T x}} &> 0.5 \\ \Leftrightarrow e^{-W^T x} &< 1 \\ \Leftrightarrow W^T x &> 0 \end{aligned}$$

Trong trường hợp  $y = 0$ :

$$\begin{aligned} h_W(x) &\leq 0.5 \\ \Leftrightarrow \frac{1}{1 + e^{-W^T x}} &\leq 0.5 \\ \Leftrightarrow e^{-W^T x} &\leq 1 \\ \Leftrightarrow W^T x &\leq 0 \end{aligned}$$

Như vậy ta có thể nhận ra những điểm thuộc về nhãn 1 sẽ nằm bên phải đường biên phân chia  $W^T x$  trong khi những điểm thuộc về nhãn 0 sẽ nằm bên trái. Đồng thời đường biên phân chia hai nhãn 0 và 1 cũng là một phương trình tuyến tính.

#### **2.1.4 Ứng dụng hồi quy Logistic**

- Y tế: Hồi quy logistic được sử dụng để dự đoán nguy cơ mắc các bệnh như bệnh tim, ung thư, và nhiều bệnh khác. Nó cũng góp phần vào việc chẩn đoán bệnh thông qua các triệu chứng và kết quả xét nghiệm.
- Tài chính: Hồi quy Logistic được sử dụng để dự đoán khả năng vỡ nợ của người vay tiền, giá cổ phiếu và các biến động thị trường khác. Nó cũng được sử dụng để phát hiện gian lận thẻ tín dụng và bảo hiểm.
- Tiếp thị: Hồi quy Logistic được sử dụng để phân loại khách hàng tiềm năng, dự đoán tỷ lệ mua hàng và nhắm mục tiêu quảng cáo. Nó cũng được khai thác để tùy chỉnh trải nghiệm cho khách hàng.
- Sản xuất: Hồi quy Logistic được áp dụng để dự đoán các sự cố máy móc, quản lý chất lượng và tối ưu hóa quy trình sản xuất. Nó cũng được sử dụng để dự báo nhu cầu và quản lý hàng tồn kho.

#### **2.1.4 Ví dụ đơn giản của hồi quy Logistic**

Dự báo xác suất để một người không trả được nợ

- Gọi biến phụ thuộc “vỡ nợ” là  $y$ :
  - $y = 1$ : Người đó vỡ nợ
  - $y = 0$ : Người đó không vỡ nợ
- Dự báo  $P(y = 1)$  : Ta muốn tính toán xác suất để một người sẽ vỡ nợ.
- $P(y = 0) = 1 - P(y = 1)$ : Nếu biết xác suất để một người vỡ nợ, thì xác suất để người đó không vỡ nợ bằng 1 trừ đi xác suất vỡ nợ.
- Các biến độc lập  $x_1, x_2, \dots, x_k$ , đây là các yếu tố đầu vào của người mà ta đang xem xét. Ví dụ:
  - Thu nhập hàng tháng
  - Tuổi
  - Trình độ học vấn

- Tài sản đảm bảo,...

- Sử dụng hàm logistic:  $P(y = 1) = \frac{1}{1+e^{-z}}$

Trong đó:  $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$

- Dự báo  $y = 1$  nếu  $P(y = 1) \geq 0,5$  :

$$\frac{1}{1+e^{-z}} \geq 0,5 \Leftrightarrow z \geq 0 \Leftrightarrow \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k \geq 0$$

Dự báo  $y = 0$  nếu  $P(y = 1) < 0,5$  :

$$\frac{1}{1+e^{-z}} < 0,5 \Leftrightarrow z < 0 \Leftrightarrow \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k < 0$$

- Giả sử mô hình hồi quy logistic với hai biến  $x_1, x_2$  có hệ số ước lượng:

$$\beta_0=3, \beta_1=1, \beta_2=1$$

- Dự báo  $y = 1$  nếu  $-3 + x_1 + x_2 \geq 0$

- Dự báo  $y = 0$  nếu  $-3 + x_1 + x_2 < 0$

Ví dụ:

- Với điểm (1,1):  $-3 + 1 + 1 = -1 < 0 \Leftrightarrow$  Dự báo  $y = 0$

- Với điểm (3,1):  $-3 + 3 + 1 = 1 \geq 0 \Leftrightarrow$  Dự báo  $y = 1$

### 2.1.6 Mã giả của hồi quy Logistic

- 1: **Input:** Training data
- 2: **Begin**
- 3: For  $i = 1$  to  $k$
- 4: For each training data instance  $d_i$ .
- 5: Set the target value for the regression to  $z_i = \frac{y_i - P(1|d_j)}{[P(1|d_j)(1 - P(1|d_j))]}$
- 6: Initialize the weight of instance  $d_j$  to  $[P(1|d_j)(1 - P(1|d_j))]$
- 7: Finalize a  $f(j)$  to the data with class value ( $Z_j$ ) and weight ( $w_j$ )
- 8: **Classical label decision**
- 9: Assign (class label: 1) if  $P_{id} > 0.5$ , otherwise (class label: 2)
- 10: **End**

Hình 2. 2 Mã giả hồi quy Logistic

1. Dữ liệu đầu vào của thuật toán là tập dữ liệu huấn luyện.
2. Đánh dấu sự bắt đầu thuật toán.
3. Vòng lặp bên ngoài chạy từ 1 đến  $k$  lần, trong đó  $k$  là số lần lặp (số epoch hoặc số lần huấn luyện).
4. Vòng lặp bên trong chạy qua từng mẫu dữ liệu huấn luyện  $d_i$ .
5. Thiết lập giá trị mục tiêu cho hồi quy, được kí hiệu là  $z_i$ . Trong đó:

- $y_i$ : Nhãn thực của mẫu  $d_i$ .
  - $P(1|d_j)$ : Xác suất dự đoán lớp 1 của mẫu  $d_i$ .
6. Khởi tạo trọng số của mẫu  $d_j$  với giá trị  $[P(1|d_j)(1 - P(1|d_j))]$ . Đây là cách tính trọng số cho mẫu dựa trên xác suất dự đoán và xác suất còn lại (1 - Xác suất dự đoán).
  7. Hoàn thiện một hàm hồi quy  $f(j)$  dựa trên dữ liệu với giá trị lớp ( $Z_j$ ) và trọng số ( $w_j$ ).
  8. Quyết định nhãn lớp theo cách cổ điển.
  9. Gán nhãn lớp cho dữ liệu mới. Nếu xác suất dự đoán  $P_{id}$  lớn hơn 0,5 thì dữ liệu được gán nhãn lớp 1. Ngược lại, nếu xác suất dự đoán  $P_{id}$  nhỏ hơn hoặc bằng 0,5 thì dữ liệu được gán nhãn lớp 2.
  10. Kết thúc thuật toán.

## 2.2 Mô hình Support Vector Machine (SVM)

### 2.2.1 Support Vector Machine là gì ?

Trong lĩnh vực học máy, Support Vector Machine (SVM), hay còn gọi là máy vector hỗ trợ, là một mô hình giám sát được thiết kế để tối đa hóa lợi nhuận và sử dụng các thuật toán liên quan để phân tích và phân loại dữ liệu. Được phát triển tại Phòng thí nghiệm AT&T Bell bởi Vladimir Vapnik cùng các đồng nghiệp (Boser và cộng sự, 1992, Guyon và cộng sự, 1993, Cortes và Vapnik, 1995, Vapnik và cộng sự, 1997). SVM là một trong những SVM phổ biến nhất. Các mô hình được nghiên cứu, dựa trên khung học tập thống kê hoặc lý thuyết VC do Vapnik (1982, 1995) và Chervonenkis (1974) đề xuất.

Ngoài việc thực hiện phân loại tuyến tính, SVM có thể thực hiện phân loại phi tuyến tính một cách hiệu quả bằng cách sử dụng những gì được gọi là "thủ thuật hạt nhân". Thủ thuật này giúp ánh xạ các đầu vào ban đầu vào các không gian đặc trưng có số chiều cao hơn. SVM cũng có thể được sử dụng cho các nhiệm vụ hồi quy.

Ý tưởng chính của SVM là tìm ra một siêu phẳng (hyperplane) trong không gian đa chiều (các đặc trưng) sao cho nó có thể phân tách các điểm dữ liệu của các lớp khác nhau một cách hiệu quả nhất. Khoảng cách từ siêu phẳng đến các điểm dữ liệu gần nhất của mỗi lớp được gọi là biên an toàn (margin). SVM cố gắng tối đa hóa biên này. Các điểm dữ liệu nằm trên biên hoặc bên trong biên được gọi là support vectors.



### 2.2.2 Thuật toán SVM

Trong không gian hai chiều, khoảng cách từ một điểm có tọa độ  $(x_0, y_0)$  tới đường thẳng có phương trình  $w_1x + w_2y + b = 0$  được xác định bởi công thức:

$$\frac{|w_1x_0 + w_2y_0 + b|}{\sqrt{w_1^2 + w_2^2}} \quad (2-7)$$

Trong không gian ba chiều, khoảng cách từ một điểm có tọa độ  $(x_0, y_0, z_0)$  tới đường thẳng có phương trình  $w_1x + w_2y + w_3z + b = 0$  được xác định bởi công thức:

$$\frac{|w_1x + w_2y + w_3z + b|}{\sqrt{w_1^2 + w_2^2 + w_3^2}} \quad (2-8)$$

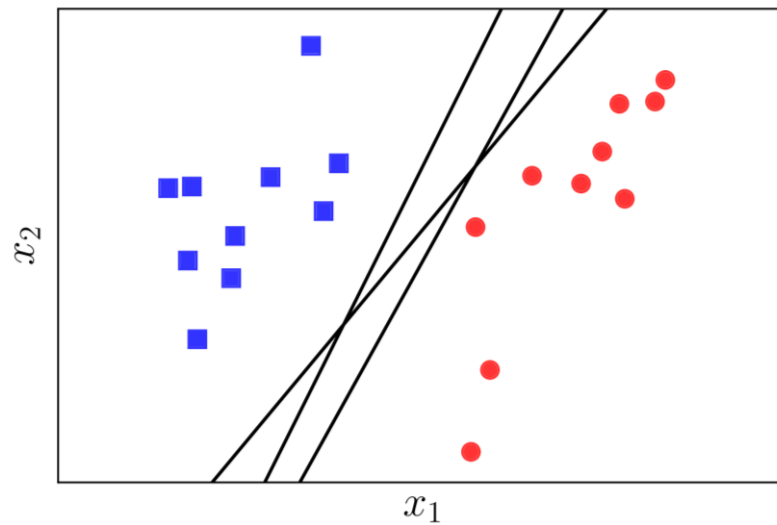
Ngoài ra, nếu loại bỏ dấu trị tuyệt đối ở mẫu số, chúng ta có thể xác định vị trí của điểm đó so với đường thẳng hoặc mặt phẳng. Những điểm làm cho biểu thức trong dấu giá trị tuyệt đối mang giá trị dương nằm về cùng một phía (gọi là phía dương của đường thẳng), những điểm làm cho biểu thức mang giá trị âm nằm về phía ngược lại (gọi là phía âm). Các điểm nằm trên đường thẳng hoặc mặt phẳng làm cho tử số bằng 0, tức là khoảng cách từ điểm đó đến đường thẳng/mặt phẳng là 0.

Việc này có thể được tổng quát lên trên không gian nhiều chiều: khoảng cách từ một điểm có tọa độ  $x_0$  tới một siêu mặt phẳng có phương trình  $W^T x_0 + b = 0$  được xác định bởi:

$$\frac{|W^T x_0 + b|}{||W||_2} \quad (2-9)$$

Với  $||W||_2 = \sqrt{\sum_{i=1}^d w_i^2}$  với  $d$  là số chiều của siêu mặt phẳng. Giả sử có hai lớp khác nhau được miêu tả bởi các điểm trong không gian nhiều chiều và hai lớp này là linearly separable, có nghĩa là tồn tại một siêu mặt phẳng có thể chia chính xác hai lớp đó. Để tìm một siêu mặt phẳng chia hai lớp này, ta cần tìm một mặt phẳng sao cho tất cả các điểm của một lớp nằm về cùng một phía của mặt phẳng đó và tất cả các điểm của lớp còn lại nằm về phía ngược lại của mặt phẳng. Chúng ta đã biết rằng thuật toán PLA

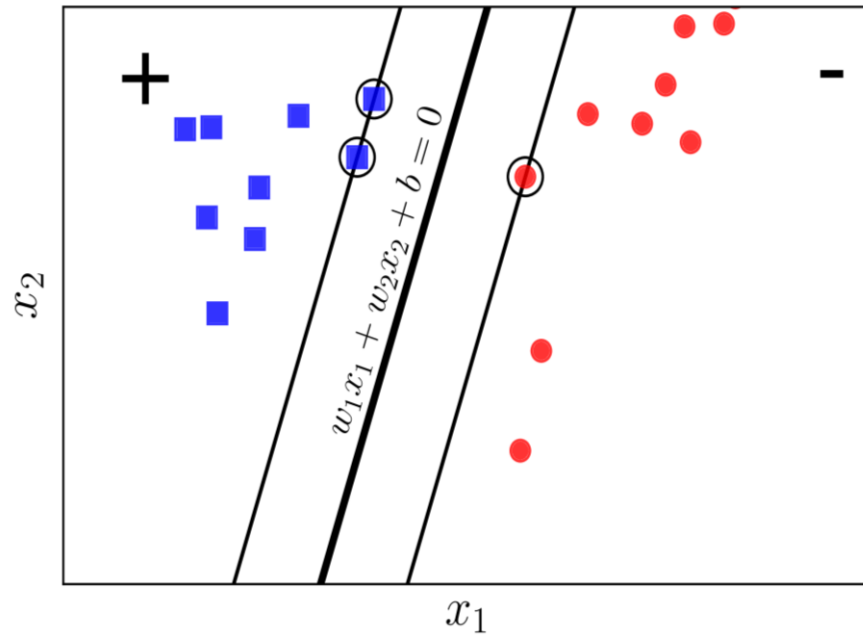
(Perceptron Learning Algorithm) có thể giúp chúng ta tìm được mặt phân chia nhưng có thể dẫn đến vô số nghiệm khác nhau như được minh họa trong hình dưới đây:



Hình 2. 3 Các mặt phân lớp hai class

Điều cần phải xét đến là : trong số vô vàn các mặt phân chia có thể, mặt phân chia nào được coi là tốt nhất theo một tiêu chuẩn cụ thể? Trên hình 2.3 ở trên, có ba đường thẳng minh họa, trong đó có hai đường thẳng lệch khá nhiều về phía của lớp hình tròn màu đỏ. Điều này có thể khiến cho lớp màu đỏ cảm thấy không hài lòng vì cảm giác rằng lãnh thổ của họ bị chiếm dụng nhiều quá. Vậy có phương pháp nào để tìm ra một đường phân chia mà cả hai lớp đều cho là công bằng và đảm bảo 'hài lòng' nhất không? Đó chính là mục tiêu của bài toán SVM.

Giả sử rằng tập training set có dạng  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  với vector  $x_i \in \mathbb{R}^d$  để thể hiện đầu vào của một điểm dữ liệu,  $d$  là số chiều của dữ liệu và  $N$  là số dữ liệu. Gọi các điểm vuông xanh thuộc class 1, các điểm tròn đỏ thuộc class -1 và mặt  $W^T x + b = w_1 x + w_2 y + b = 0$  là mặt phân chia giữa 2 classes. Hình minh họa 2.3.



Hình 2.4 Mặt phẳng phân chia và các điểm xanh đỏ thuộc hai lớp khác nhau

Gọi *margin* là khoảng cách gần nhất từ một điểm tới mặt phẳng, bất kì điểm nào trong cả hai classes.

$$margin = \min_n \frac{y_n(W^T x_n + b)}{\|W\|_2} \quad (2-10)$$

Do đó, bài toán tối ưu của SVM chính là bài toán tìm trọng số  $w$  và độ lệch  $b$  sao cho *margin* đạt giá trị lớn nhất. Và để xác định class của một điểm dữ liệu mới thì sẽ được xác định bằng công thức:

$$class(x) = \text{sgn}(W^T x + b) \quad (2-11)$$

Trong đó *sgn* được định nghĩa là hàm xác định dấu, nhận giá trị 1 nếu đối số là không âm và nhận giá trị -1 nếu đối số là âm.

### 2.2.3 Ứng dụng của thuật toán SVM

#### Phân loại:

- Phân loại văn bản: Phân loại email rác, phân loại chủ đề tài liệu, phân tích tình cảm.
- Xử lý ảnh: Nhận diện đối tượng, phân loại ảnh, phân tích ảnh y tế.
- Sinh học: Phân loại protein, dự đoán cấu trúc protein, phân tích dữ liệu microarray.
- Tài chính: Phân tích rủi ro tín dụng, phát hiện gian lận, dự đoán thị trường chứng khoán.

- Tiếp thị: Phân khúc khách hàng, nhắm mục tiêu quảng cáo, cá nhân hóa trải nghiệm khách hàng.

#### **Hỏi quy:**

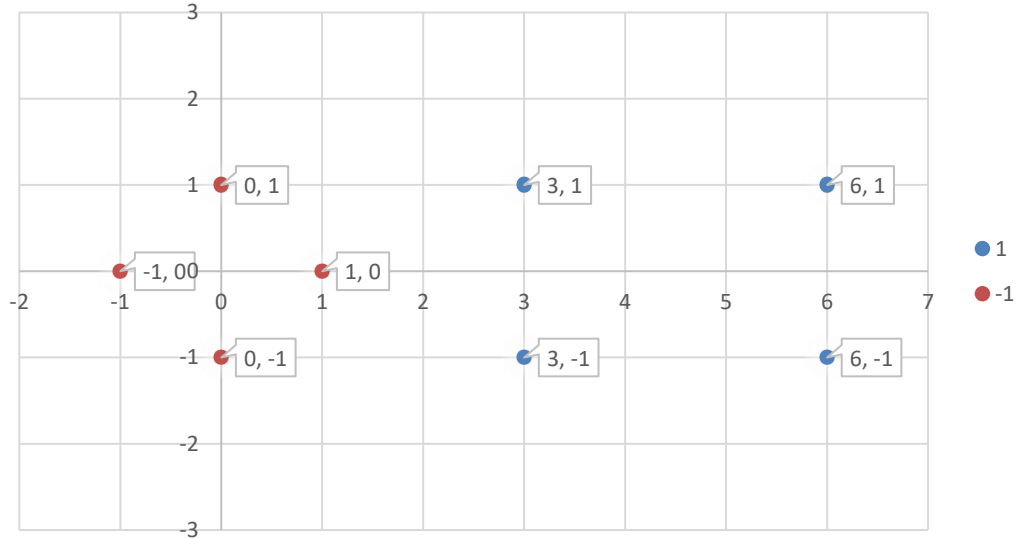
- Dự báo giá cả: Dự báo giá cổ phiếu, giá bất động sản, giá hàng hóa.
- Dự đoán nhu cầu: Dự đoán nhu cầu cho sản phẩm, dịch vụ.
- Phân tích chuỗi thời gian: Dự đoán xu hướng thị trường, dự báo doanh thu.

#### **2.2.4 Bài toán ví dụ về SVM**

Giả sử chúng ta có một bộ dữ liệu đơn giản với hai lớp (dương tính là +1 và âm tính là -1) trong không gian hai chiều:

<b>Điểm dữ liệu</b>	<b>x1</b>	<b>x2</b>	<b>Class</b>
Điểm 1	3	1	+1
Điểm 2	3	-1	+1
Điểm 3	6	1	+1
Điểm 4	6	-1	+1
Điểm 5	1	0	-1
Điểm 6	0	1	-1
Điểm 7	0	-1	-1
Điểm 8	-1	0	-1

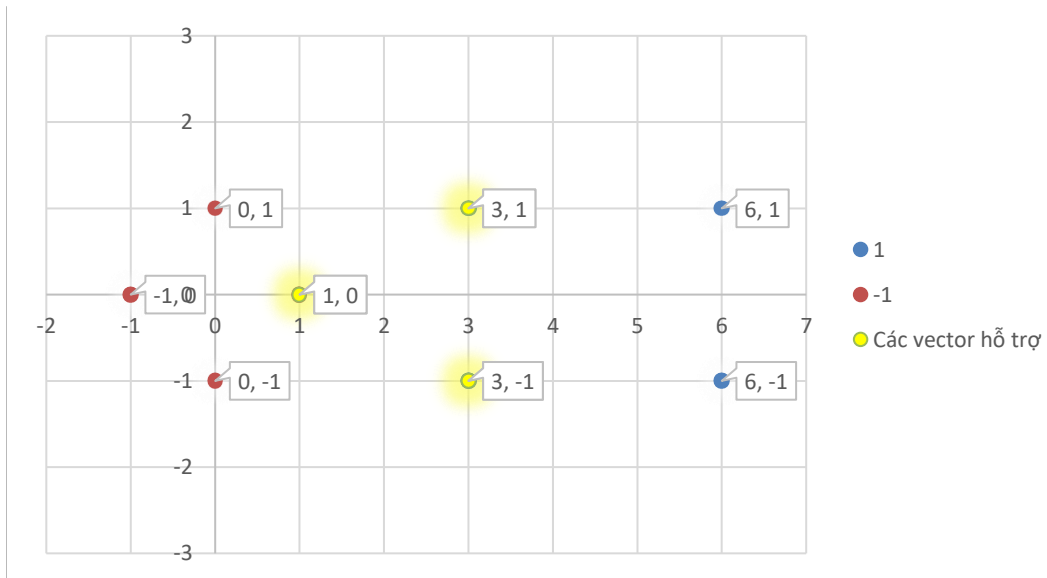
*Bảng 2.4 Dữ liệu ví dụ*



Hình 2.5 Mô phỏng các điểm dữ liệu

Nhìn vào biểu đồ có thể dễ dàng nhìn thấy có ba vector hỗ trợ gần với hyperplane nhất và nằm trên đường biên là:

$$\{s_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, s_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, s_3 = \begin{pmatrix} 3 \\ -1 \end{pmatrix}\}$$



Hình 2.6 Mô phỏng các vector hỗ trợ

Mỗi vector đầu vào được mở rộng bằng cách thêm một hằng số (thường là 1) để tính đến độ lệch  $b$  trong hàm quyết định. Điều này được thực hiện bằng cách thêm 1 chiều phụ vào vector đầu vào, ta sẽ phân biệt các vector này bằng dấu ngã :

$$s_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow \tilde{s}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

$$s_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow \tilde{s}_2 = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix}$$

$$s_3 = \begin{pmatrix} 3 \\ -1 \end{pmatrix} \Rightarrow \tilde{s}_3 = \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$$

Để tìm được trọng số  $w$  và độ lệch  $b$  cần phải tính được  $\vec{a}$  đại diện cho các điểm dữ liệu đầu vào, có tất cả 3 điểm dữ liệu đầu vào là vector hỗ trợ, do đó:

$$\vec{a} = (\alpha_1, \alpha_2, \alpha_3)$$

Giờ ta cần tìm ba giá trị  $\alpha_1, \alpha_2, \alpha_3$  dựa trên ba phương trình đường thẳng:

$$\Leftrightarrow \begin{cases} \alpha_1 \tilde{s}_1 \cdot \tilde{s}_1 + \alpha_2 \tilde{s}_2 \cdot \tilde{s}_1 + \alpha_3 \tilde{s}_3 \cdot \tilde{s}_1 = -1 \\ \alpha_1 \tilde{s}_1 \cdot \tilde{s}_2 + \alpha_2 \tilde{s}_2 \cdot \tilde{s}_2 + \alpha_3 \tilde{s}_3 \cdot \tilde{s}_2 = +1 \\ \alpha_1 \tilde{s}_1 \cdot \tilde{s}_3 + \alpha_2 \tilde{s}_2 \cdot \tilde{s}_3 + \alpha_3 \tilde{s}_3 \cdot \tilde{s}_3 = +1 \end{cases}$$

$$\Leftrightarrow \begin{cases} \alpha_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = -1 \\ \alpha_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} = +1 \\ \alpha_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} = +1 \end{cases}$$

$$\Leftrightarrow \begin{cases} \alpha_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix} = -1 \\ \alpha_1 \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 9 \\ 1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 9 \\ -1 \\ 1 \end{pmatrix} = +1 \\ \alpha_1 \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 9 \\ -1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 9 \\ 1 \\ 1 \end{pmatrix} = +1 \end{cases}$$

$$\Leftrightarrow \begin{cases} \alpha_1 (1 + 0 + 1) + \alpha_2 (3 + 0 + 1) + \alpha_3 (3 + 0 + 1) = -1 \\ \alpha_1 (3 + 0 + 1) + \alpha_2 (9 + 0 + 1) + \alpha_3 (9 - 1 + 1) = +1 \\ \alpha_1 (3 + 0 + 1) + \alpha_2 (9 - 1 + 1) + \alpha_3 (9 + 1 + 1) = +1 \end{cases}$$

Sau khi đơn giản hóa, ta có hệ phương trình:

$$\Leftrightarrow \begin{cases} 2\alpha_1 + 4\alpha_2 + 4\alpha_3 = -1 \\ 4\alpha_1 + 11\alpha_2 + 9\alpha_3 = +1 \\ 4\alpha_1 + 9\alpha_2 + 11\alpha_3 = +1 \end{cases}$$

Rút gọn ba phương trình đồng thời, ta có kết quả:

$$\Leftrightarrow \begin{cases} \alpha_1 = -3,5 \\ \alpha_2 = 0,75 \\ \alpha_3 = 0,75 \end{cases}$$

Siêu phẳng phân biệt lớp tích cực với tích cực được cho bởi:

$$\tilde{w} = \sum_i \alpha_i \tilde{s}_i \quad (2-12)$$

Trong đó:  $\tilde{w}$  là trọng số được thêm chiều phụ.

Thay giá trị vừa tìm được:

$$\tilde{w} = \alpha_1 \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \cdot \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + \alpha_3 \cdot \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$$

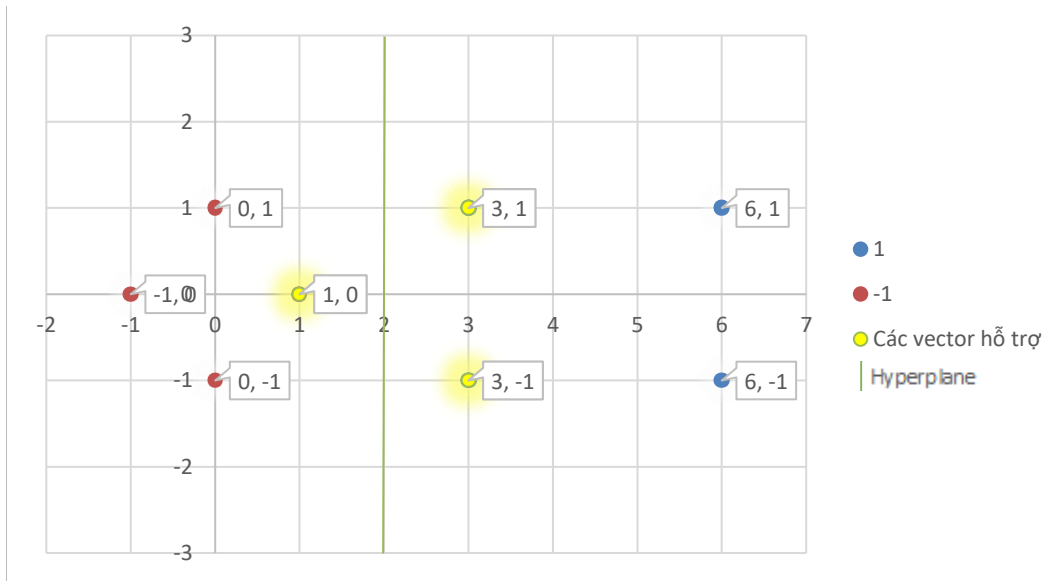
$$\tilde{w} = -3,5 \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + 0,75 \cdot \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + 0,75 \cdot \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$$

$$\tilde{w} = \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}$$

Các vector đã được bổ sung với một độ lệch  $b$ , do đó chúng ta có thể coi các giá trị trong  $\tilde{w}$  như là một siêu phẳng với độ lệch  $b$ . Phương trình siêu phẳng có dạng:

$$y = wx + b \text{ với } w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, b = 2$$

Như vậy, phương trình siêu phẳng phân cách trở thành:  $y = \begin{pmatrix} 1 \\ 0 \end{pmatrix} x + (-2)$  tức  $y = x - 2$ .



Hình 2.6 Mô phỏng Hyperplane

Khi đã tìm được phương trình siêu phẳng, chúng ta có thể tính biên độ (margin) bằng công thức sau:

$$margin = \frac{2}{||w||}$$

Trong đó,  $||w||$  là độ dài của vector trọng số. Để tính độ dài của vector trọng số  $||w||$  sử dụng công thức:

$$||w|| = \sqrt{w_1^2 + w_2^2} = \sqrt{1^2 + 0^2} = \sqrt{1} = 1$$

Do đó, biên độ (margin) của siêu phẳng được tính như sau:

$$margin = \frac{2}{1} = 2$$

Vậy, biên độ cho siêu phẳng với phương trình  $y = x - 2$  là 2.



### 2.2.5 Mã giả của SVM

```

Training Model for SVM


---


Input: D=[X,Y]; X(array of input with m features), Y(array of class labels)
        Y=array(C) // Class label
Output: Find the performance of the system
function train_svm(X,Y, number_of_runs)
    initialize: learning_rate=Math.random();
    for learning_rate in number_of_runs
        error=0;
        for i in X
            if (Y[i] *(X[i]*w))<1 then
                update : w=w + learning_rate * ((X[i]*Y[i])*(-2*(1/number_of_runs)*w)
            else
                update: w=w+learning_rate *(-2*(1/number_of_runs)*w)
            end if
        end
    end

```

Hình 2.7 Mã giả của thuật toán SVM

#### 1. **Input: D = [X, Y]; X (array of input with m features), Y (array of class labels)**

Dữ liệu đầu vào của thuật toán là 1 tập dữ liệu D bao gồm 2 mảng X và Y. Trong đó:

- X: Mảng dữ liệu đầu vào với m đặc trưng.
- Y: Mảng nhãn lớp tương ứng với dữ liệu đầu vào.

#### 2. **Y = array(C) // Class label**

Mảng các nhãn lớp, với mỗi nhãn lớp là một phần tử của mảng C. Các giá trị nhãn lớp thường là -1 hoặc 1 trong mô hình SVM.

#### 3. **Output: Find the performance of the system**

Đầu ra cho thuật toán SVM là hiệu suất của hệ thống. Hiệu suất của hệ thống thường được đo bằng độ chính xác hoặc các chỉ số khác.

#### 4. **function train\_svm(X, Y, number\_of\_runs)**

Khai báo một hàm có tên là train\_svm với ba tham số: X, Y và number\_of\_runs. Trong đó:

- X: Mảng dữ liệu đầu vào với m đặc trưng.
- Y: Mảng nhãn lớp tương ứng với dữ liệu đầu vào.
- number\_of\_runs: Số lần lặp lại thuật toán (số lần mô hình SVM sẽ được cập nhật dựa trên dữ liệu huấn luyện. Số lần lặp càng nhiều, mô hình càng có khả năng học tập tốt hơn).

#### 5. **initialize: learning\_rate = Math.random();**

Khởi tạo tốc độ học (`learning_rate`) với 1 giá trị ngẫu nhiên. Tốc độ học này quyết định mức độ điều chỉnh trọng số  $w$  trong mỗi lần cập nhật.

**6. `error = 0;`**

Khởi tạo biến lỗi bằng 0. Biến lỗi được sử dụng để theo dõi số lượng ví dụ dữ liệu được phân loại sai trong mỗi lần lặp.

**7. `for i in X`**

Bắt đầu một vòng lặp qua từng phần tử  $i$  trong mảng  $X$ .

**8. `if (Y[i] * (X[i] * w) < 1) then`**

Kiểm tra điều kiện phân loại. Điều kiện này kiểm tra xem điểm dữ liệu thứ  $i$  có được phân loại chính xác hay không. Nếu tích của nhãn lớp  $Y[i]$  và tích vector điểm dữ liệu  $X[i]$  với vector trọng số  $w$  mà nhỏ hơn 1 thì điểm dữ liệu  $i$  được coi là được phân loại sai.

**9. `update: w = w + learning_rate * ((X[i] * Y[i]) * (-2 * (1 / number_of_runs) * w))`**

Cập nhật trọng số  $w$  trong trường hợp điều kiện đúng bằng cách cộng thêm một lượng điều chỉnh, thành phần điều chỉnh phụ thuộc vào trọng số hiện tại và số lần huấn luyện.

**10. `else update: w = w + learning_rate * (-2 * (1 / number_of_runs) * w)`**

Nếu điều kiện là sai thì cập nhật trọng số  $w$  bằng cách cộng thêm một lượng điều chỉnh nhỏ hơn, thành phần điều chỉnh phụ thuộc vào trọng số hiện tại và số lần huấn luyện.

## **2.3 Mô hình mạng thần kinh nhân tạo (ANN)**

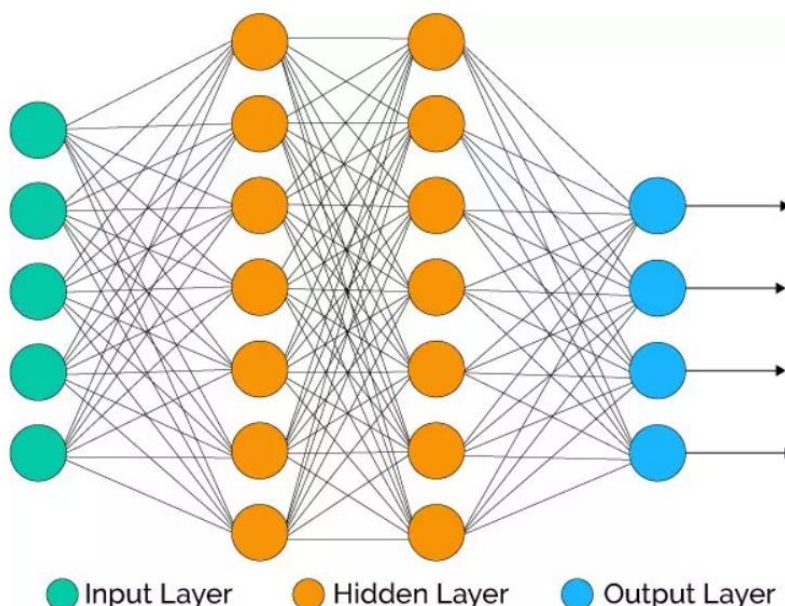
### **2.3.1 *Mạng thần kinh nhân tạo là gì ?***

Trong học máy, mạng thần kinh nhân tạo (ANN) – hay còn được gọi là mạng nơ-ron nhân tạo, là các thuật toán dựa trên chức năng não và được sử dụng để mô hình hóa các mô hình phức tạp và dự báo các vấn đề. Mạng thần kinh nhân tạo (ANN) là một phương pháp học sâu xuất phát từ khái niệm Mạng thần kinh sinh học não người. Sự phát triển của ANN là kết quả của nỗ lực tái tạo hoạt động của bộ não con người. Hoạt động của ANN cực kỳ giống với hoạt động của mạng lưới thần kinh sinh học, mặc dù chúng không giống nhau. Thuật toán ANN chỉ chấp nhận dữ liệu số và có cấu trúc.

Ý tưởng chính của ANN là xây dựng một hệ thống mô phỏng nơ-ron sinh học để học từ dữ liệu. Mô hình gồm nhiều lớp nơ-ron, thực hiện các phép tính và điều chỉnh trọng số thông qua các quá trình lan truyền tiến và lan truyền ngược, để tối ưu hóa khả năng dự đoán hoặc phân loại của nó dựa trên dữ liệu huấn luyện.

Một ANN điển hình gồm 3 thành phần chính: Lớp đầu vào nhận diện dữ liệu từ môi trường bên ngoài và lớp đầu ra chỉ gồm 1 layer cung cấp kết quả của mạng, lớp ẩn có thể có 1 hay nhiều layer nhằm xử lý và trích xuất các đặc trưng từ dữ liệu đầu vào.

Trong ANN, trừ lớp đầu vào thì tất cả các node thuộc các layer khác đều hoàn toàn được kết nối với các node thuộc layer trước nó. Mỗi node thuộc lớp ẩn nhận vào ma trận đầu vào từ lớp trước và kết hợp với trọng số để ra được kết quả.



*Hình 2. 8 Cấu trúc cơ bản của một ANN*

Với báo cáo này, dựa vào cấu trúc của ANN và các cấu trúc mạng liên quan, em xây dựng một mạng ANN đơn giản với cấu trúc mô hình mạng nơ-ron truyền thẳng tuần tự có tên gọi là sequential neural network. Các lớp của mạng như sau:

Model: "sequential"		
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 16)	496
dense_1 (Dense)	(None, 8)	136
dense_2 (Dense)	(None, 1)	9
Total params: 641 (2.50 KB)		
Trainable params: 641 (2.50 KB)		
Non-trainable params: 0 (0.00 Byte)		

Hình 2. 9 Mô hình ANN

Mô hình gồm có 3 layer với chi tiết thông số gồm:

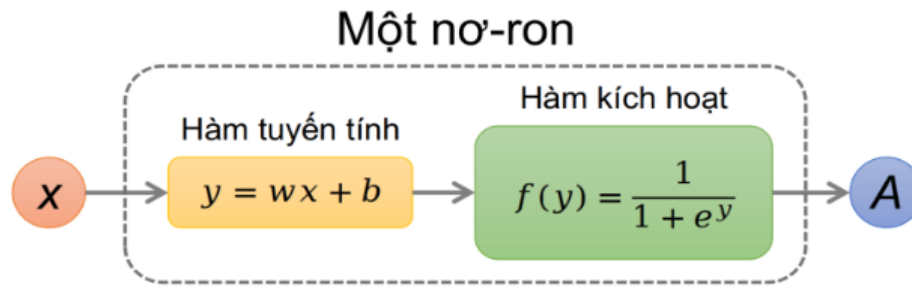
- Layer một: thuộc loại Dense với hàm kích hoạt ReLU. Lớp này có 16 nơ-ron và mỗi nơ-ron được kết nối với toàn bộ nơ-ron của lớp trước đó. Đầu vào của lớp này là kích thước bằng 30 (input\_dim = 30), đồng nghĩa với việc số lượng biến đặc trưng của nó là 30. Đầu ra của lớp này sẽ là một vector có kích thước (None, 16) với None là số lượng mẫu đầu vào và 16 là số lượng nơ-ron trong lớp. Tổng số tham số của lớp này sẽ là 496.
- Layer hai: có số lượng nơ-ron là 8 và hàm kích hoạt ReLU. Lớp này không cần chỉ định kích thước đầu vào vì layer một đã xác định được kích thước đầu vào. Lớp này có tổng tham số là 136.
- Layer ba: có số lượng nơ-ron là 1 và hàm kích hoạt Sigmoid. Đây là lớp cuối cùng của mô hình và nhận tham số đầu vào là từ lớp hai trả về. Tổng tham số cho lớp cuối cùng này là 9.

Sau khi xây dựng xong mô hình với ba lớp trên, ta sẽ đưa dữ liệu đặc trưng và dữ liệu mục tiêu vào để tiến hành đào tạo mô hình nhằm dự đoán bệnh dựa trên các đặc điểm trước đó.

### 2.3.2 Thuật toán ANN

#### Hàm kích hoạt

Hàm kích hoạt (activation function) hoặc hàm truyền được sử dụng nhằm thay đổi các tham số đầu vào sang một khoảng giá trị khác. Để mạng có thể đưa ra lựa chọn giữa truyền tiếp dữ liệu hay không truyền tiếp dữ liệu và cường độ truyền là bao nhiêu thì mạng cần có hàm kích hoạt.



Hình 2. 10 Cấu trúc của một nơ-ron bao gồm một hàm tuyến tính và một hàm kích hoạt

- **Hàm tuyến tính:**

Công thức của hàm:

$$f(x) = x \quad (2-13)$$

Đạo hàm của hàm tuyến tính:

$$\frac{df}{dx} = 1 \quad (2-14)$$

Giá trị đầu ra phụ thuộc vào phạm vi giá trị của đầu vào khi được xử lý qua hàm. Do bài toán giải quyết của nghiên cứu là xây dựng ANN để chẩn đoán ung thư vú vì vậy hàm được sử dụng ở lớp đầu ra là hàm tuyến tính.

- **Hàm sigmoid**

ANN sử dụng nhiều hàm kích hoạt cho từng nơ-ron trong toàn bộ cấu trúc mạng của mình, một trong số đó là hàm kích hoạt Sigmoid đã được đề cập đến trên 2.1.2.

Đạo hàm của hàm Sigmoid:

$$\frac{df}{dx} = f(x) \cdot (1 - f(x)) \quad (2-15)$$

### Thuật toán suy giảm độ dốc và tốc độ học

Suy giảm độ dốc (Gradient Descent) là một phương pháp tối ưu được sử dụng nhằm mục đích tìm ra giá trị cực tiểu của hàm số. Ban đầu, phương pháp này sẽ bắt đầu từ một điểm ngẫu nhiên trên hàm số và sau đó dịch chuyển điểm này theo hướng giảm dần của đạo hàm cho đến khi đến được điểm cực tiểu. Gradient Descent thường được sử dụng với mục đích cập nhật trọng số  $w$  và độ lệch  $b$  của mỗi lớp qua công thức:

$$w_{new} = w_{old} - \eta \cdot \frac{\partial cost}{\partial w_{old}} \quad (2-16)$$

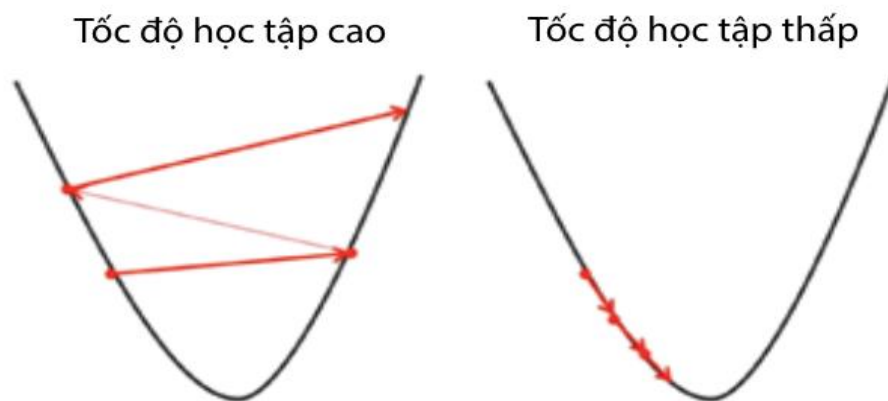
$$b_{new} = b_{old} - \eta \cdot \frac{\partial cost}{\partial b_{old}} \quad (2-17)$$

Trong đó:

$w_{new}$ : Trọng số  $w$  khi đã cập nhật;  $w_{old}$ : Trọng số  $w$  cũ khi chưa cập nhật;  $b_{new}$ : Độ lệch  $b$  mới khi đã cập nhật;  $b_{old}$ : Độ lệch  $b$  cũ khi chưa cập nhật;  $\eta$ : Tốc độ học;  $\frac{\partial cost}{\partial w_{old}}$ :

Đạo hàm của hàm mất mát theo trọng số  $w$  cũ;  $\frac{\partial cost}{\partial b_{old}}$ : Đạo hàm của hàm mất mát theo độ lệch  $b$  cũ.

Tốc độ học hay còn gọi là Learning rate, là một chỉ số quan trọng nhằm quản lý số lần lặp trong suy giảm độ dốc. Nếu tốc độ học tập nhỏ thì thuật toán sẽ phải thực hiện nhiều bước lặp để hàm số đến được điểm cực tiểu. Trái lại, khi chỉ số này lớn thì thuật toán sẽ yêu cầu ít vòng lặp hơn, thế nhưng điều đó có thể dẫn đến trường hợp hàm số không thể đạt được sự hội tụ và bỏ qua điểm cực tiểu.



Hình 2.11 So sánh tốc độ học

### 2.3.2 Ứng dụng của ANN

- Công nghiệp: Chẩn đoán và phát hiện sự cố, kiểm soát chất lượng, phân tích mối liên hệ giữa dữ liệu từ các cảm biến và tín hiệu khác.
- Tài chính: Xây dựng mô hình và dự báo thị trường (ví dụ như thị trường tiền tệ), phân bổ ngân quỹ, đưa ra quyết định đầu tư.
- Máy tính và viễn thông: Giảm nhiễu, xử lý tín hiệu, nhận diện (hình ảnh, giọng nói).
- Môi trường: Dự đoán và phân tích khí hậu, đánh giá nguy cơ, quản lý nguồn tài nguyên, phân tích hóa học.

### 2.3.3 Bài toán ví dụ của ANN

#### Khởi tạo các trọng số và độ lệch:

- Trọng số từ lớp đầu vào đến lớp ẩn:

$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$
0,15	0,20	0,30	0,05	0,01	-0,2

- Trọng số từ lớp ẩn đến lớp đầu ra:

$w_7$	$w_8$
0,8	0,6

- Độ lệch từ lớp đầu vào đến lớp ẩn:

$b_1$	$b_2$
0,35	0,35

- Độ lệch từ lớp đầu ẩn đến lớp đầu ra:

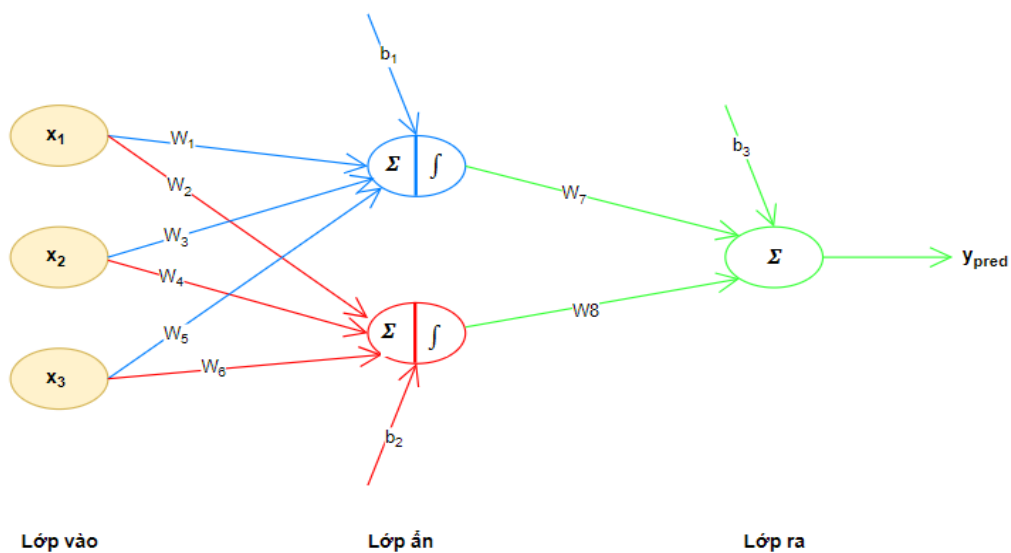
$b_3$
0,16

- Input đầu vào:

$x_1$	$x_2$	$x_3$
0,1	0,2	0,3
0,5	0,6	0,7
0,8	0,9	1

- Mục tiêu:

$y$
0,4



Hình 2. 12 Mô hình hóa các giá trị giữa các lớp

### Theo lan truyền xuôi (Forward Propagation)

Tính toán đầu ra của mạng bằng cách truyền các giá trị đầu vào qua các lớp của mạng. Mục đích là xác định đầu ra dự đoán của mạng dựa trên trọng số  $w$  và độ lệch  $b$ .

**Bước 1:** Tính toán giá trị lớp ẩn

$$\begin{aligned}
 z_1 &= w_1x_1 + w_3x_2 + w_5x_3 + b_1 \\
 &= 0,12 \cdot 0,1 + 0,30 \cdot 0,2 + 0,01 \cdot 0,3 + 0,35 \\
 &= 0,425
 \end{aligned}$$

$$h_1 = \text{sigmoid}(z_{h1})$$

$$= \frac{1}{1+e^{-z_1}}$$



$$= \frac{1}{1+e^{-0,425}}$$

$$= 0,60$$

$$z_2 = w_2x_1 + w_4x_2 + w_6x_3 + b_2$$

$$= 0,20 \cdot 0,1 + 0,05 \cdot 0,2 + (-0,2) \cdot 0,3 + 0,35$$

$$= 0,32$$

$$h_2 = \text{sigmoid}(z_{h2})$$

$$= \frac{1}{1+e^{-z_2}}$$

$$= \frac{1}{1+e^{0,32}}$$

$$= 0,42$$

**Bước 2:** Tính giá trị lớp đầu ra

$$z_y = w_7h_1 + w_8h_2 + b_3$$

$$= 0,8 \cdot 0,60 + 0,6 \cdot 0,42 + 0,16$$

$$= 0,892$$

$$y_{pred} = \text{sigmoid}(z_y)$$

$$= \frac{1}{1+e^{-z_y}}$$

$$= \frac{1}{1+e^{-0,892}}$$

$$= 0,71$$

### Theo lan truyền ngược (Back Propagation)

Mục đích của lan truyền ngược là cập nhật trọng số  $w$  và độ lệch  $b$ . Ta sẽ cập nhật liên tục cho đến khi lỗi giữa giá trị đầu ra dự đoán và giá trị đầu ra thực tế là nhỏ nhất. Để tối ưu các trọng số ta thực hiện Gradient Descent, để làm được điều này ta cần tính được đạo hàm của trọng số  $w$  và độ lệch  $b$ .

Đầu tiên ta sẽ chọn một hàm lỗi hoặc hàm chi phí để tính toán sai số giữa giá trị đầu ra dự đoán và đầu ra thực tế:

$$\text{cost function} = (y_{pred} - y_{act})^2 \quad (2-18)$$

Đạo hàm riêng:

$$\frac{\partial \text{cost}}{\partial w_7} = \frac{\partial \text{cost}}{\partial y_{pred}} \cdot \frac{\partial y_{pred}}{\partial w_7} = 2(y_{pred} - y_{act}) \cdot h_1 = 2 \cdot (0,71 - 0,4) \cdot 0,60 = 0,372$$

$$\frac{\partial cost}{\partial w_8} = \frac{\partial cost}{\partial y_{pred}} \cdot \frac{\partial y_{pred}}{\partial w_8} = 2(y_{pred} - y_{act}) \cdot h_2 = 2 \cdot (0,71 - 0,4) \cdot 0,42 = 0,2604$$

$$\frac{\partial cost}{\partial b_3} = \frac{\partial cost}{\partial y_{pred}} \cdot \frac{\partial y_{pred}}{\partial b_3} = 2(y_{pred} - y_{act}) \cdot b_3 = 2 \cdot (0,71 - 0,4) \cdot 0,16 = 0,9902$$

$$\begin{aligned} \frac{\partial cost}{\partial w_1} &= \frac{\partial cost}{\partial y_{pred}} \cdot \frac{\partial y_{pred}}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_1} = 2(y_{pred} - y_{act}) \cdot w_7 \cdot \left[ \left( \frac{1}{1+e^{-z_1}} \right) \cdot \left( 1 - \frac{1}{1+e^{-z_1}} \right) \right] \cdot x_1 \\ &= 2 \cdot (0,71 - 0,4) \cdot 0,8 \cdot [0,60 \cdot (1 - 0,60)] \cdot 0,1 \\ &= -0,017 \end{aligned}$$

$$\begin{aligned} \frac{\partial cost}{\partial w_2} &= \frac{\partial cost}{\partial y_{pred}} \cdot \frac{\partial y_{pred}}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial w_2} = 2(y_{pred} - y_{act}) \cdot w_8 \cdot \left[ \left( \frac{1}{1+e^{-z_2}} \right) \cdot \left( 1 - \frac{1}{1+e^{-z_2}} \right) \right] \cdot x_1 \\ &= 2 \cdot (0,71 - 0,4) \cdot 0,6 \cdot [0,58 \cdot (1 - 0,58)] \cdot 0,1 \\ &= 9,06 \end{aligned}$$

$$\begin{aligned} \frac{\partial cost}{\partial w_3} &= \frac{\partial cost}{\partial y_{pred}} \cdot \frac{\partial y_{pred}}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_3} = 2(y_{pred} - y_{act}) \cdot w_7 \cdot \left[ \left( \frac{1}{1+e^{-z_1}} \right) \cdot \left( 1 - \frac{1}{1+e^{-z_1}} \right) \right] \cdot x_2 \\ &= 2 \cdot (0,71 - 0,4) \cdot 0,8 \cdot [0,60 \cdot (1 - 0,60)] \cdot 0,2 \\ &= 0,023 \end{aligned}$$

$$\begin{aligned} \frac{\partial cost}{\partial w_4} &= \frac{\partial cost}{\partial y_{pred}} \cdot \frac{\partial y_{pred}}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial w_4} = 2(y_{pred} - y_{act}) \cdot w_8 \cdot \left[ \left( \frac{1}{1+e^{-z_2}} \right) \cdot \left( 1 - \frac{1}{1+e^{-z_2}} \right) \right] \cdot x_2 \\ &= 2 \cdot (0,71 - 0,4) \cdot 0,6 \cdot [0,58 \cdot (1 - 0,58)] \cdot 0,2 \\ &= 0,018 \end{aligned}$$

$$\begin{aligned} \frac{\partial cost}{\partial w_5} &= \frac{\partial cost}{\partial y_{pred}} \cdot \frac{\partial y_{pred}}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_5} = 2(y_{pred} - y_{act}) \cdot w_7 \cdot \left[ \left( \frac{1}{1+e^{-z_1}} \right) \cdot \left( 1 - \frac{1}{1+e^{-z_1}} \right) \right] \cdot x_3 \\ &= 2 \cdot (0,71 - 0,4) \cdot 0,8 \cdot [0,60 \cdot (1 - 0,60)] \cdot 0,3 \\ &= 0,035 \end{aligned}$$

$$\begin{aligned} \frac{\partial cost}{\partial w_6} &= \frac{\partial cost}{\partial y_{pred}} \cdot \frac{\partial y_{pred}}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial w_6} = 2(y_{pred} - y_{act}) \cdot w_8 \cdot \left[ \left( \frac{1}{1+e^{-z_2}} \right) \cdot \left( 1 - \frac{1}{1+e^{-z_2}} \right) \right] \cdot x_3 \\ &= 2 \cdot (0,71 - 0,4) \cdot 0,6 \cdot [0,58 \cdot (1 - 0,58)] \cdot 0,3 \\ &= 0,027 \end{aligned}$$

$$\begin{aligned} \frac{\partial cost}{\partial b_1} &= \frac{\partial cost}{\partial y_{pred}} \cdot \frac{\partial y_{pred}}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial b_1} = 2(y_{pred} - y_{act}) \cdot w_7 \cdot \left[ \left( \frac{1}{1+e^{-z_1}} \right) \cdot \left( 1 - \frac{1}{1+e^{-z_1}} \right) \right] \cdot 1 \\ &= 2 \cdot (0,71 - 0,4) \cdot 0,8 \cdot [0,60 \cdot (1 - 0,60)] \cdot 1 \\ &= 0,119 \end{aligned}$$

$$\begin{aligned} \frac{\partial cost}{\partial b_2} &= \frac{\partial cost}{\partial y_{pred}} \cdot \frac{\partial y_{pred}}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial b_2} = 2(y_{pred} - y_{act}) \cdot w_8 \cdot \left[ \left( \frac{1}{1+e^{-z_2}} \right) \cdot \left( 1 - \frac{1}{1+e^{-z_2}} \right) \right] \cdot 1 \\ &= 2 \cdot (0,71 - 0,4) \cdot 0,6 \cdot [0,58 \cdot (1 - 0,58)] \cdot 1 \\ &= 0,090 \end{aligned}$$

**Bước 2:** Cập nhật trọng số  $w$  và độ lệch  $b$ 

Ta có đạo hàm riêng cho trọng số  $w$  và độ lệch  $b$  của lớp đầu ra, bây giờ sẽ cập nhật trọng số  $w$  và độ lệch  $b$  theo phương pháp gradient descent.

$$w_{1\text{ new}} = w_{1\text{ old}} - \eta \cdot \frac{\partial \text{cost}}{\partial w_1} = 0,15 - 0,01 \cdot (-0,017) = 0,15017$$

$$w_{2\text{ new}} = w_{2\text{ old}} - \eta \cdot \frac{\partial \text{cost}}{\partial w_2} = 0,20 - 0,01 \cdot 9,06 = 0,1094$$

$$w_{3\text{ new}} = w_{3\text{ old}} - \eta \cdot \frac{\partial \text{cost}}{\partial w_3} = 0,30 - 0,01 \cdot 0,023 = 0,29977$$

$$w_{4\text{ new}} = w_{4\text{ old}} - \eta \cdot \frac{\partial \text{cost}}{\partial w_4} = 0,05 - 0,01 \cdot 0,018 = 0,04982$$

$$w_{5\text{ new}} = w_{5\text{ old}} - \eta \cdot \frac{\partial \text{cost}}{\partial w_5} = 0,01 - 0,01 \cdot 0,035 = 9,65$$

$$w_{6\text{ new}} = w_{6\text{ old}} - \eta \cdot \frac{\partial \text{cost}}{\partial w_6} = -0,2 - 0,01 \cdot 0,027 = -0,20027$$

$$w_{7\text{ new}} = w_{7\text{ old}} - \eta \cdot \frac{\partial \text{cost}}{\partial w_7} = 0,8 - 0,01 \cdot 0,372 = 0,79628$$

$$w_{8\text{ new}} = w_{8\text{ old}} - \eta \cdot \frac{\partial \text{cost}}{\partial w_8} = 0,6 - 0,01 \cdot 0,2604 = 0,597396$$

$$b_{1\text{ new}} = b_{1\text{ old}} - \eta \cdot \frac{\partial \text{cost}}{\partial b_1} = 0,35 - 0,01 \cdot 0,119 = 0,3881$$

$$b_{2\text{ new}} = b_{2\text{ old}} - \eta \cdot \frac{\partial \text{cost}}{\partial b_2} = 0,35 - 0,01 \cdot 0,090 = 0,3491$$

$$b_{3\text{ new}} = b_{3\text{ old}} - \eta \cdot \frac{\partial \text{cost}}{\partial b_3} = 0,16 - 0,01 \cdot 0,9902 = 0,150098$$

Ta đã hoàn thành một lượt của lan truyền xuôi và lan truyền ngược bằng cách sử dụng một ví dụ huấn luyện từ tập dữ liệu. Đây còn được gọi là giảm độ dốc ngẫu nhiên, tức là cập nhật các trọng số bằng cách sử dụng một ví dụ huấn luyện tại một thời điểm.

Lặp lại quá trình lan truyền xuôi bằng cách sử dụng các trọng số và độ lệch được cập nhật từ lần lan truyền ngược trước đó cho ví dụ huấn luyện thứ 2. Làm tương tự với ví dụ huấn luyện thứ 3.

### 2.3.4 Mã giả của ANN

#### Algorithm 1 ANN Training pseudo-code

---

```

1:  $W1 \leftarrow$  Weight vector of input layer to hidden layer
2:  $W2 \leftarrow$  Weight vector of hidden layer to output layer
3:  $NumberCorrect = 0$ 
4: for  $i < \text{Number of Training Iterations}$  do
5:   for  $j < \text{Size(Train set)}$  do
6:      $Input \leftarrow \text{Trainset}(j)$ 
7:      $HiddenOutput \leftarrow f(\text{Bias}; W1, Input)$ 
8:      $Output \leftarrow g(\text{Bias}; W2, HiddenOutput)$ 
9:      $Prediction \leftarrow \text{argmax}(Output)$ 
10:    if  $Prediction = \text{Train label}(j)$  then
11:       $NumberCorrect++ = 1$ 
12:    end if
13:     $\delta_1 \leftarrow (Output - \text{trainlabel}(j)) * (1 - Output^2)$ 
14:     $\delta_2 \leftarrow (W2 * \delta_1) * (1 - HiddenOutput^2)$ 
15:     $W1 \leftarrow W1 - \alpha * (Input * \delta_2')$ 
16:     $W2 \leftarrow W2 - \alpha * (hiddenoutput * \delta_1')$ 
17:  end for
18:   $accuracy \leftarrow NumberCorrect / N$ 
19: end for

```

---

Hình 2. 13 Mã giả của ANN theo lan truyền ngược

#### 1. $W1 \leftarrow$ Weight vector of input layer to hidden layer

Khởi tạo vector trọng số  $w_1$  kết nối lớp đầu vào với lớp ẩn.

#### 2. $W2 \leftarrow$ Weight vector of hidden layer to output layer

Khởi tạo vector trọng số  $w_2$  kết nối lớp ẩn với lớp đầu ra.

#### 3. $NumberCorrect = 0$

Khởi tạo biến đếm số lượng dự đoán đúng là 0.

#### 4. **for** $i < \text{Number of Training Iterations}$ **do**

Vòng lặp này được sử dụng để huấn luyện mô hình trong nhiều lần lặp.

#### 5. **for** $j < \text{Size(Train set)}$ **do**

Vòng lặp này lặp qua toàn bộ tập dữ liệu huấn luyện.

#### 6. $Input \leftarrow \text{Train set}(j)$

Dữ liệu đầu vào lấy từ tập huấn luyện tại vị trí  $j$ .

#### 7. $HiddenOutput \leftarrow f(\text{Bias}; W1, Input)$

Tính toán đầu ra của lớp ẩn bằng cách áp dụng hàm kích hoạt  $f$  với độ lệch (Bias), trọng

số  $w_1$  và đầu vào.

**8. Output  $\leftarrow g(\text{Bias}; W2, \text{HiddenOutput})$**

Tính toán đầu ra của lớp đầu ra bằng cách áp dụng hàm kích hoạt  $g$  với độ lệch (Bias), trọng số  $w_2$ , đầu ra của lớp ẩn.

**9. Prediction  $\leftarrow \text{argmax}(\text{Output})$**

Lấy nhãn dự đoán bằng cách chọn chỉ số có giá trị lớn nhất từ đầu ra.

**10. if Prediction = Train label(j) then**

Nếu nhãn dự đoán trùng với nhãn thực tế của mẫu dữ liệu thứ  $j$  trong tập huấn luyện

**11. NumberCorrect += 1**

Tăng biến đếm số lượng dự đoán đúng lên 1.

**12. end if**

Kết thúc kiểm tra điều kiện.

**13.  $\text{delta}_1 \leftarrow (\text{Output} - \text{Train label}(j)) * (1 - \text{Output}^2)$**

Tính toán  $\text{delta}$  cho lớp đầu ra (sai số giữa đầu ra và nhãn thực tế) và nhân với đạo hàm của hàm kích hoạt đầu ra.

**14.  $\text{delta}_2 \leftarrow (W2 * \text{delta}_1) * (1 - \text{HiddenOutput}^2)$**

Tính toán  $\text{delta}$  cho lớp ẩn bằng cách nhân trọng số  $w_2$  với  $\text{delta}$  của lớp đầu ra và đạo hàm của hàm kích hoạt lớp ẩn.

**15.  $W1 \leftarrow W1 - \alpha * (\text{Input} * \text{delta}'_2)$**

Cập nhật trọng số  $w_1$  bằng cách trừ đi một lượng tỉ lệ với đầu vào và  $\text{delta}$  của lớp ẩn, nhân với hệ số học  $\alpha$ .

**16.  $W2 \leftarrow W2 - \alpha * (\text{HiddenOutput} * \text{delta}'_1)$**

Cập nhật trọng số  $w_2$  bằng cách trừ đi một lượng tỉ lệ với đầu ra của lớp ẩn và  $\text{delta}$  của lớp đầu ra, nhân với hệ số học  $\alpha$ .

**17. end for**

Kết thúc vòng lặp qua tập huấn luyện.

$$18. \text{accuracy} = \frac{\text{NumberCorrect}}{N}$$

Tính toán độ chính xác của mô hình bằng cách chia số lượng dự đoán đúng cho tổng số mẫu  $N$ .

#### 19. end for

Kết thúc vòng lặp qua các lần huấn luyện.

### 2.4 Các chỉ số đánh giá

Sau khi đã xây dựng được những mô hình dự đoán, việc tiếp theo là kiểm định hiệu suất dự đoán của mô hình, và từ đó mới kết luận xem mô hình nào có hiệu năng tốt hơn. Có bốn trường hợp xảy ra khi mô hình dự đoán được huấn luyện đó là:

- True Positive (TP): Đối tượng thuộc lớp Positive và mô hình dự đoán là Positive.
- True Negative (TN): Đối tượng ở lớp Negative và mô hình dự đoán Negative.
- False Positive (FP): Đối tượng ở lớp Negative, mô hình dự đoán Positive.
- False Negative (FN): Đối tượng ở lớp Positive, mô hình dự đoán Negative.

Bốn loại trường hợp trên thường được biểu diễn dưới dạng ma trận hỗn loạn, hay còn gọi là confusion matrix. Trong thực tế, có ba chỉ số để đánh giá một mô hình theo dạng dự đoán đó là: accuracy, precision và recall. Trong đó:

- Accuracy được định nghĩa là tỉ lệ phần trăm dự đoán đúng trên tổng số lượng dữ liệu thử nghiệm. Để tính toán accuracy, ta chia số lần dự đoán đúng cho tổng số lượng dự đoán:

$$acc = \frac{\text{số lần dự đoán đúng}}{\text{tổng số lượng dự đoán}}$$

- Precision được định nghĩa là phần nhỏ của các ví dụ có liên quan trong số tất cả các ví dụ được dự đoán thuộc một lớp nhất định. Được tính theo công thức :

$$\frac{\text{true positive}}{\text{true positive} + \text{false positive}}$$

- Recall được định nghĩa là phần nhỏ của các ví dụ được dự đoán thuộc một lớp so với tất cả ví dụ thực sự thuộc lớp đó. Recall xác định bằng công thức :

$$\frac{\text{true positive}}{\text{true positive} + \text{false negative}}$$

## CHƯƠNG 3 THỰC NGHIỆM

### 3.1 Tổng quan về tập dữ liệu

Nghiên cứu này sử dụng tập dữ liệu có tên “Breast Cancer Wisconsin” được phát hành bởi các tác giả Wolberg, William, Mangasarian, Olvi, Street, Nick, and Street,.. vào năm 1995 và hiện đang có sẵn tại kho dữ liệu trực tuyến UC Irvine Machine Learning Repository. Tập dữ liệu chứa 569 bản ghi về các tế bào của khối u lành tính và ác tính. Là một bộ dữ liệu nổi tiếng dùng trong học máy để phân loại các khối u vú là lành tính hoặc ác tính.

Mỗi hàng đại diện cho một khối u, và mỗi cột đại diện cho một đặc trưng cụ thể. Dưới đây là giải thích chi tiết về các cột dữ liệu trong bộ dữ liệu này:

1. ID: Mã số định danh duy nhất cho mỗi mẫu bệnh nhân.
2. Diagnosis: Chẩn đoán khối u, với hai giá trị: "M" (Malignant - ác tính) và "B" (Benign - lành tính).

Mỗi thuộc tính trên đều có 3 giá trị thống kê được tính toán:

- mean (trung bình) : Giá trị trung bình của các đặc trưng tế bào học.
- se (standard error – sai số chuẩn) : Sai số chuẩn của các đặc trưng về tế bào học.
- worst: Giá trị lớn nhất (hoặc tệ nhất) của các đặc trưng về tế bào học.

Vì vậy, trong bộ dữ liệu có tổng cộng 30 thuộc tính (10 thuộc tính chính, mỗi thuộc tính có 3 giá trị). Tên các cột tương ứng là:

1. Radius (Bán kính)
  - radius\_mean: Bán kính trung bình
  - radius\_se: Sai số chuẩn của bán kính
  - radius\_worst: Bán kính lớn nhất
2. Texture (Kết cấu)
  - texture\_mean: Kết cấu trung bình (độ lệch chuẩn của giá trị xám)
  - texture\_se: Sai số chuẩn của kết cấu
  - texture\_worst: Kết cấu lớn nhất
3. Perimeter (Chu vi)



- `perimeter_mean`: Chu vi trung bình
  - `perimeter_se`: Sai số chuẩn của chu vi
  - `perimeter_worst`: Chu vi lớn nhất
4. Area (Diện tích)
- `area_mean`: Diện tích trung bình
  - `area_se`: Sai số chuẩn của diện tích
  - `area_worst`: Diện tích lớn nhất
5. Smoothness (Độ mịn)
- `smoothness_mean`: Độ mịn trung bình (sự biến đổi cục bộ của bán kính)
  - `smoothness_se`: Sai số chuẩn của độ mịn
  - `smoothness_worst`: Độ mịn lớn nhất
6. Compactness (Độ đặc)
- `compactness_mean`: Độ đặc trung bình ( $\frac{chu\ vi^2}{diện\ tích} - 1.0$ )
  - `compactness_se`: Sai số chuẩn của độ đặc
  - `compactness_worst`: Độ đặc lớn nhất
7. Concavity (Độ lõm)
- `concavity_mean`: Độ lõm trung bình (mức độ lõm của các phần của ranh giới)
  - `concavity_se`: Sai số chuẩn của độ lõm
  - `concavity_worst`: Độ lõm lớn nhất
8. Concave Points (Các điểm lõm)
- `concave_points_mean`: Số lượng điểm lõm trung bình trên ranh giới
  - `concave_points_se`: Sai số chuẩn của số lượng điểm lõm
  - `concave_points_worst`: Số lượng điểm lõm lớn nhất
9. Symmetry (Độ đối xứng)
- `symmetry_mean`: Độ đối xứng trung bình
  - `symmetry_se`: Sai số chuẩn của độ đối xứng
  - `symmetry_worst`: Độ đối xứng lớn nhất
10. Fractal Dimension (Độ phân dạng)
- `fractal_dimension_mean`: Độ phân dạng trung bình ( $\frac{“coastlineapproximation”}{diện\ tích}$ )

- fractal\_dimension\_se: Sai số chuẩn của độ phân dạng
- fractal\_dimension\_worst: Độ phân dạng lớn nhất

Cấu trúc file dữ liệu như sau:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980

Hình 3. 1 Cấu trúc tập dữ liệu Breast Cancer Wisconsin

## 3.2 Các bước thực nghiệm

### 3.2.1 Import thư viện

Cài đặt các thư viện cần thiết cho chương trình.

```
import pandas as pd
import seaborn as sns
import numpy as np
from skimpy import skim
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
from sklearn.svm import SVC
```

Hình 3. 2 Import các thư viện

### 3.2.2 Đọc dữ liệu từ tập

Sử dụng thư viện pandas đọc dữ liệu từ file dữ liệu từ cung cấp vào dataframe để xử lý:

```
data = pd.read_csv(r"C:\Users\Hi\Desktop\data.csv")
data.head()
```

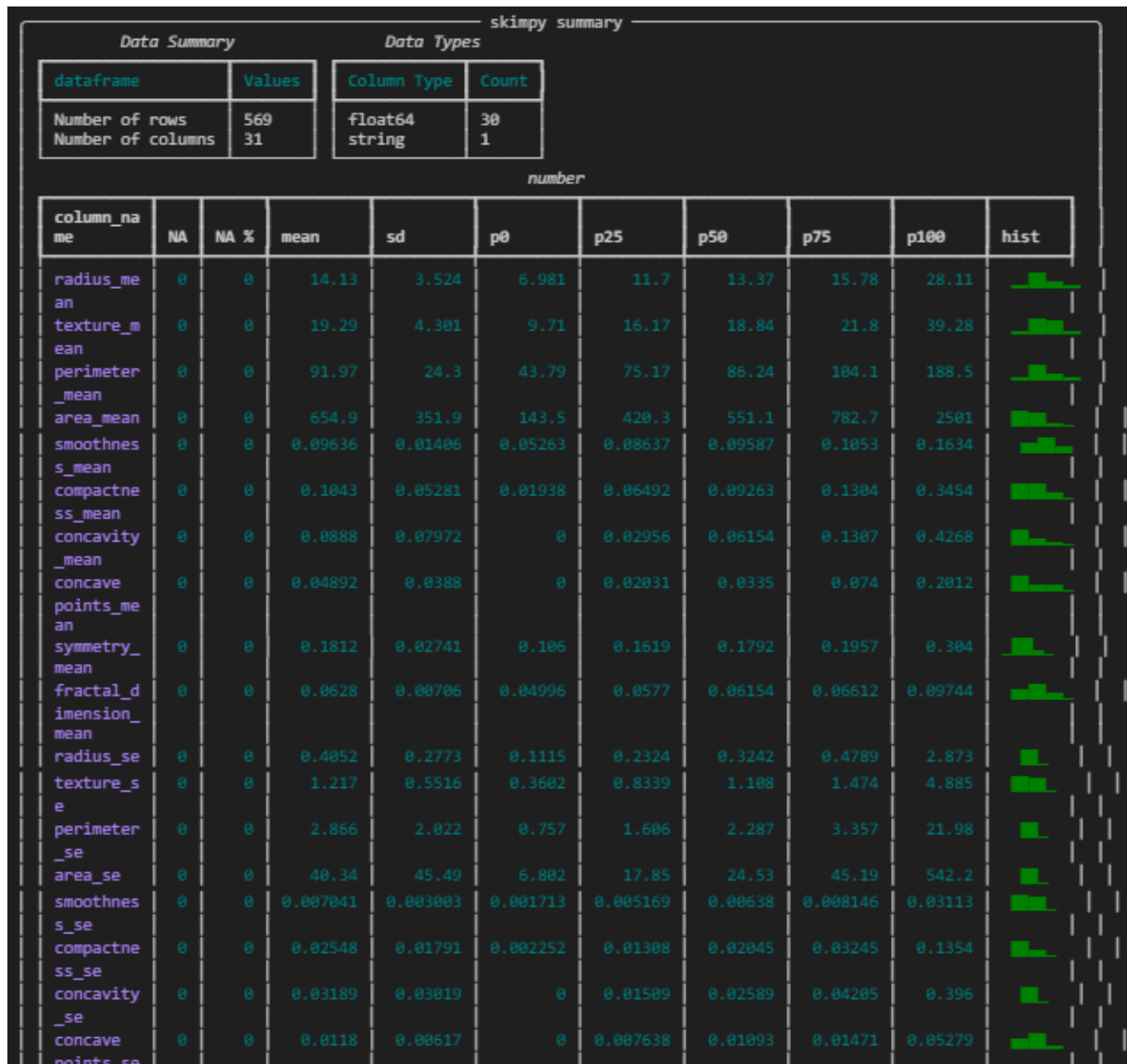
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst	...
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	25.38	...
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	24.99	...
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	23.57	...
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	14.91	...
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	22.54	...

5 rows × 32 columns

Hình 3. 3 Đọc dữ liệu từ file

### 3.2.3 Tóm tắt dữ liệu

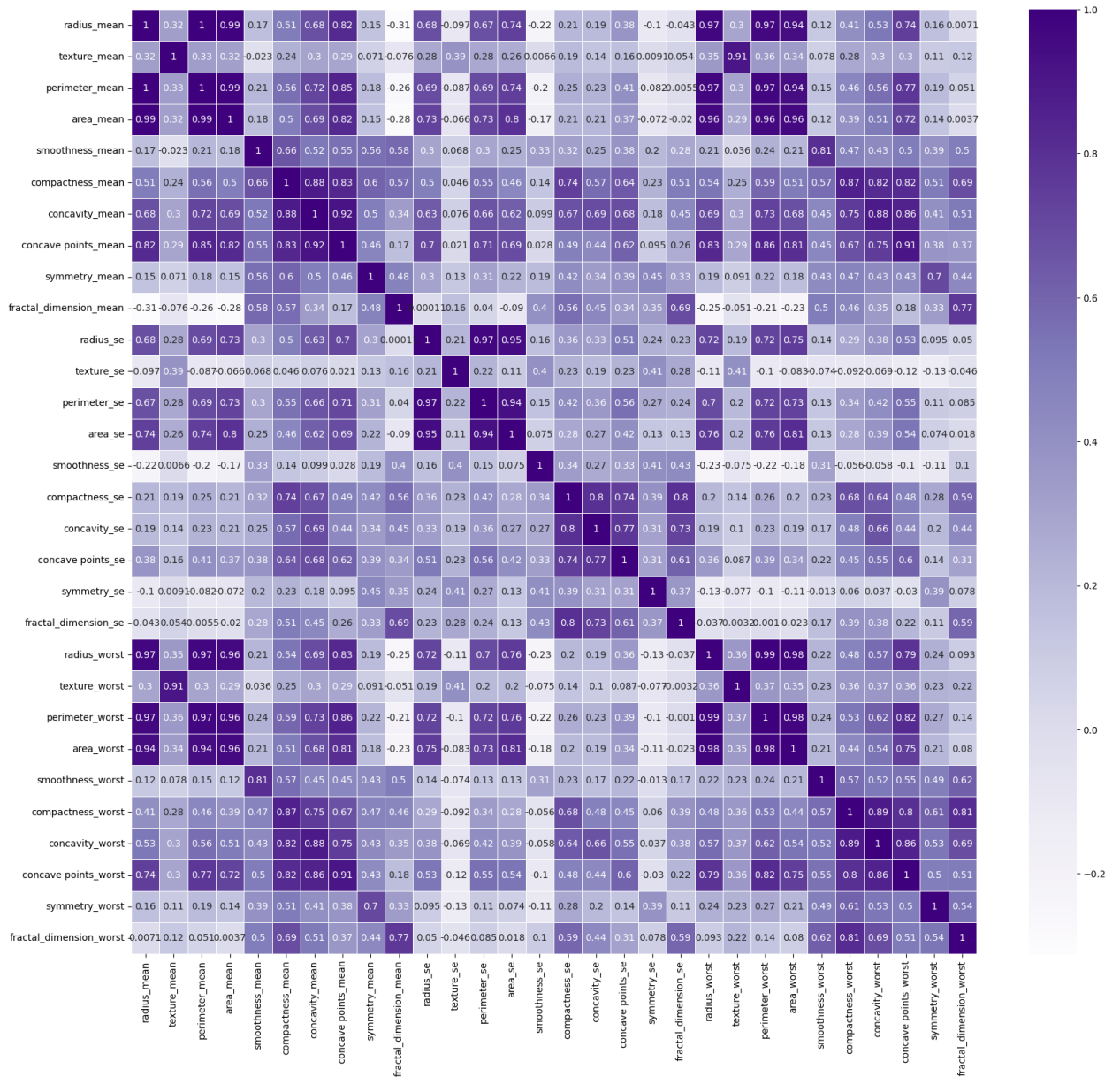
Sử dụng thư viện skimpy để tóm tắt toàn bộ dữ liệu.



Hình 3. 4 Tóm tắt toàn bộ dữ liệu

### 3.2.4 Mức độ tương quan giữa các biến

Hiện thị biểu đồ heat map thể hiện mức độ tương quan giữa các biến trong toàn bộ tập dữ liệu. Trong đó các giá trị tương quan cao gần với 1 sẽ có màu sáng, và các giá trị tương quan thấp gần với 0 sẽ có màu tối.



Hình 3. 5 Biểu đồ heatmap mức độ tương quan giữa các biến

### 3.2.5 Tiền xử lý dữ liệu

Để có thể có kết quả tốt nhất trong việc phân loại và dự đoán, một điều không thể thiếu đó chính là tiền xử lý và chia tập dữ liệu cho việc huấn luyện và kiểm tra. Tuy nhiên, với tập dữ liệu hiện có trong hình 3.4, hoàn toàn không có giá trị trống hoặc bằng 0 nào. Và các giá trị thống kê của các chỉ số cũng hoàn toàn được chọn lọc kỹ càng nên các bước tiền xử lý cơ bản sẽ không áp dụng cho tập này ngoại trừ bước xóa cột ID ra khỏi vì nó không cần thiết cho quá trình huấn luyện và dự đoán. Chuẩn hóa giá trị trong cột *diagnosis* thành số: B là 0, M là 1.

Số lượng bản ghi cho từng dự đoán lành tính và ác tính như sau:

```
data['diagnosis'].value_counts()
✓ 0.0s

diagnosis
0    357
1    212
Name: count, dtype: int64
```

Hình 3. 6 Số lượng bản ghi cho hai lớp dự đoán. 357 ác tính và 212 lành tính

Tiến hành chia tập dữ liệu thành hai tập dữ liệu con sử dụng thư viện *scikit-learn*: các biến mục tiêu là cột *diagnosis*, các biến thuộc tính đặc trưng là các cột còn lại. Tỷ lệ là 80% dùng cho việc huấn luyện và 20% dành cho việc kiểm thử.

```
X_train, X_test, y_train, y_test = train_test_split(
    data.drop('diagnosis', axis=1),
    data['diagnosis'],
    test_size=0.2,
    random_state=42)

print("Shape of training set:", X_train.shape)
print("Shape of test set:", X_test.shape)
✓ 0.0s

Shape of training set: (455, 30)
Shape of test set: (114, 30)
```

Hình 3. 7 Chia dữ liệu thành hai tập huấn luyện và kiểm thử

Sau đó, sử dụng thư viện *scaler* để chuẩn hóa dữ liệu. Cụ thể, thực hiện chuẩn hóa để cho giá trị trung bình gần bằng 0, và độ lệch chuẩn gần bằng 1. Điều này nhằm mục đích loại bỏ các giá trị ngoại lai còn tồn tại và đảm bảo dữ liệu luôn cùng một thang đo.

```
scaler = StandardScaler()  
X_train = scaler.fit_transform(X_train)  
X_test = scaler.fit_transform(X_test)  
✓ 0.0s
```

Hình 3. 8 Chuẩn hóa dữ liệu

### 3.2.6 Huấn luyện và kiểm thử

#### *Logistic Regression*

Tạo và huấn luyện dữ liệu với mô hình hồi quy logistic trên tập huấn luyện. Sau đó sử dụng tập kiểm thử để kiểm tra hiệu suất của mô hình.

```
USING LOGISTIC  
  
logreg = LogisticRegression()  
logreg.fit(X_train, y_train)  
✓ 0.0s  
  
▼ LogisticRegression ⓘ ?  
LogisticRegression()  
  
y_pred = logreg.predict(X_test)  
✓ 0.0s
```

Hình 3. 9 Huấn luyện và kiểm thử với logistic regression

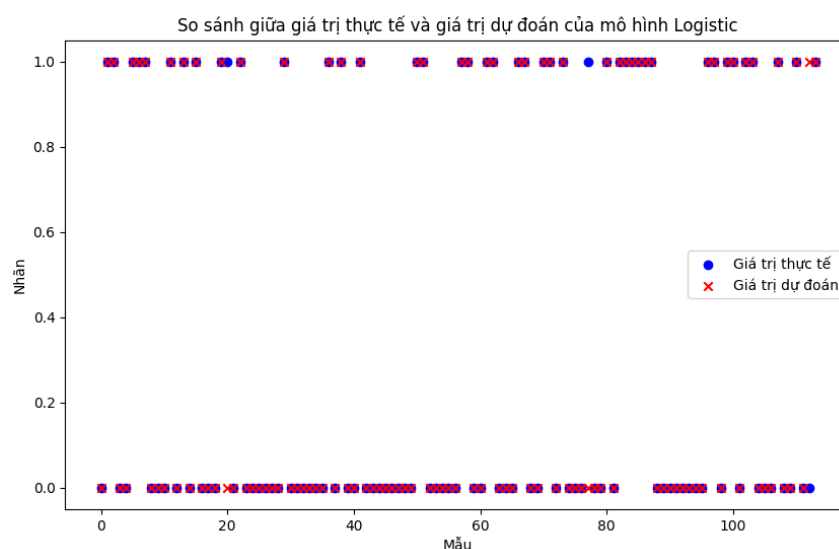
Kết quả đạt được với độ chính xác tương đối cao, với 97,36%. Với confusion matrix.

Confusion Matrix:

```
[[71  0]
 [ 2 41]]
```

		precision	recall	f1-score	support
	0	0.97	1.00	0.99	71
	1	1.00	0.95	0.98	43
	accuracy			0.98	114
	macro avg	0.99	0.98	0.98	114
	weighted avg	0.98	0.98	0.98	114

Hình 3. 10 Confusion Matrix của mô hình logistic regression



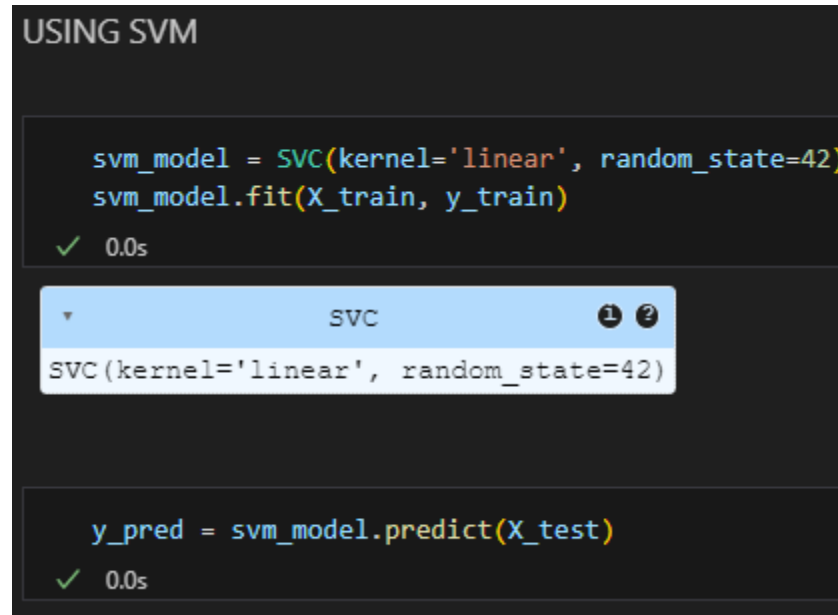
Hình 3. 11 Chênh lệch giữa giá trị thực tế và giá trị dự đoán của mô hình hồi quy tuyến tính

Từ kết quả trên, ta thấy:

- Có 71 mẫu âm tính được dự đoán đúng, 0 mẫu âm tính bị phân loại sai thành dương tính. 2 mẫu dương tính bị phân loại thành âm tính, 41 mẫu dương tính được phân loại đúng.
- Bên cạnh đó, các chỉ số accuracy, precision, recall có các giá trị tương đối cao cho thấy mô hình có hiệu suất tốt và có khả năng dự đoán dữ liệu một cách chính xác.

## Support Vector Machine

Tương tự với mô hình hồi quy, tạo và huấn luyện dữ liệu với mô hình SVM. Sau đó dùng tập kiểm thử để kiểm tra hiệu suất mô hình.



```
USING SVM

svm_model = SVC(kernel='linear', random_state=42)
svm_model.fit(X_train, y_train)

✓ 0.0s

SVC
SVC(kernel='linear', random_state=42)

y_pred = svm_model.predict(X_test)

✓ 0.0s
```

Hình 3. 12 Huấn luyện và kiểm thử trên mô hình SVM

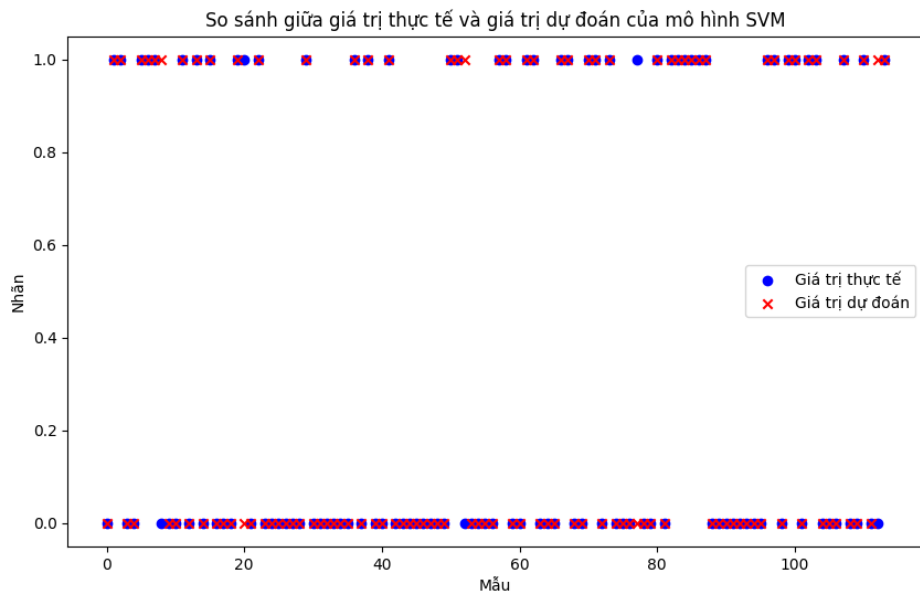
Mô hình đạt kết quả 97,36 tỉ lệ chính xác với kết quả từ confusion matrix như sau:

Confusion matrix:				
[[71  0]				
[ 3 40]]				
	precision	recall	f1-score	support
0	0.96	1.00	0.98	71
1	1.00	0.93	0.96	43
accuracy			0.97	114
macro avg	0.98	0.97	0.97	114
weighted avg	0.97	0.97	0.97	114

Hình 3. 13 Confusion matrix của mô hình SVM

- Có 71 mẫu âm tính được dự đoán đúng, 0 mẫu âm tính bị phân loại sai thành dương tính. 3 mẫu dương tính bị phân loại thành âm tính, 40 mẫu dương tính được phân loại đúng.





Hình 3. 14 Chênh lệch giữa giá trị thực tế và giá trị dự đoán của mô hình SVM

## ANN

Tiến hành xây dựng mạng nơ-ron ẩn với cấu trúc mô tả tại 2.3 với tham số tối ưu hóa Adam như sau:

```
model = Sequential([
    Dense(16, activation='relu', input_dim=30),
    Dense(8, activation='relu'),
    Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.summary()
```

✓ 0.0s

Model: "sequential\_3"

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 16)	496
dense_10 (Dense)	(None, 8)	136
dense_11 (Dense)	(None, 1)	9

Total params: 641 (2.50 KB)

Trainable params: 641 (2.50 KB)

Non-trainable params: 0 (0.00 B)

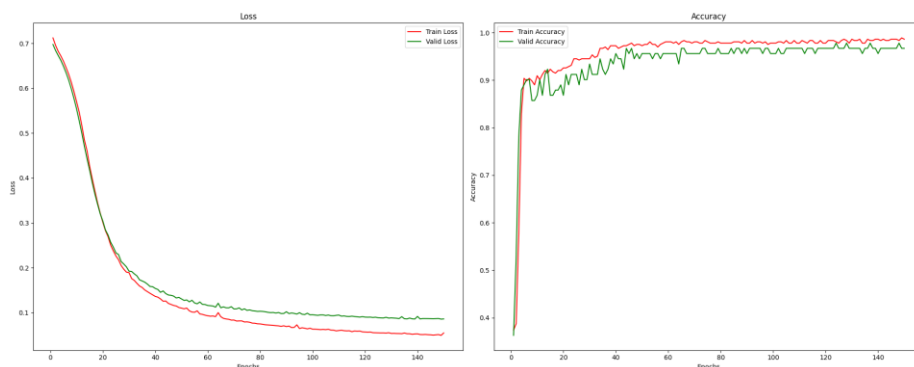
Hình 3. 15 Model ANN ba lớp

Sau đó huấn luyện mô hình với 150 epoch đạt tỉ lệ chính xác lên tới 98,26.

```
Epoch 1/150
12/12 ----- 2s 25ms/step - accuracy: 0.3922 - loss: 0.6838 - val_accuracy: 0.3956 - val_loss: 0.6766
Epoch 2/150
12/12 ----- 0s 7ms/step - accuracy: 0.4325 - loss: 0.6661 - val_accuracy: 0.4615 - val_loss: 0.6619
Epoch 3/150
12/12 ----- 0s 6ms/step - accuracy: 0.5065 - loss: 0.6553 - val_accuracy: 0.6264 - val_loss: 0.6444
Epoch 4/150
12/12 ----- 0s 5ms/step - accuracy: 0.6814 - loss: 0.6349 - val_accuracy: 0.8132 - val_loss: 0.6234
Epoch 5/150
12/12 ----- 0s 6ms/step - accuracy: 0.8101 - loss: 0.6098 - val_accuracy: 0.8132 - val_loss: 0.5981
Epoch 6/150
12/12 ----- 0s 5ms/step - accuracy: 0.8674 - loss: 0.5840 - val_accuracy: 0.8462 - val_loss: 0.5713
Epoch 7/150
12/12 ----- 0s 5ms/step - accuracy: 0.8550 - loss: 0.5586 - val_accuracy: 0.8681 - val_loss: 0.5380
Epoch 8/150
12/12 ----- 0s 5ms/step - accuracy: 0.8862 - loss: 0.5205 - val_accuracy: 0.8791 - val_loss: 0.5028
Epoch 9/150
12/12 ----- 0s 5ms/step - accuracy: 0.8945 - loss: 0.4836 - val_accuracy: 0.8791 - val_loss: 0.4648
Epoch 10/150
12/12 ----- 0s 6ms/step - accuracy: 0.8979 - loss: 0.4489 - val_accuracy: 0.8901 - val_loss: 0.4277
Epoch 11/150
12/12 ----- 0s 6ms/step - accuracy: 0.9290 - loss: 0.4065 - val_accuracy: 0.8901 - val_loss: 0.3936
Epoch 12/150
12/12 ----- 0s 6ms/step - accuracy: 0.9181 - loss: 0.3834 - val_accuracy: 0.8901 - val_loss: 0.3639
Epoch 13/150
...
Epoch 149/150
12/12 ----- 0s 5ms/step - accuracy: 0.9892 - loss: 0.0408 - val_accuracy: 0.9560 - val_loss: 0.0899
Epoch 150/150
12/12 ----- 0s 6ms/step - accuracy: 0.9866 - loss: 0.0400 - val_accuracy: 0.9670 - val_loss: 0.0920
```

Hình 3. 16 Huấn luyện ANN với 150 epoch

Kết quả giá trị hàm mất mát và tỉ lệ chính xác sau 150 epoch như sau:



Hình 3. 17: Giá trị hàm mất mát và tỉ lệ chính xác sau 150 epoch.

Và confusion matrix:

[[70 1]				
[ 2 41]]				
	precision	recall	f1-score	support
0	0.97	0.99	0.98	71
1	0.98	0.95	0.96	43
accuracy			0.97	114
macro avg	0.97	0.97	0.97	114
weighted avg	0.97	0.97	0.97	114

Hình 3. 18 Confusion matrix của ANN

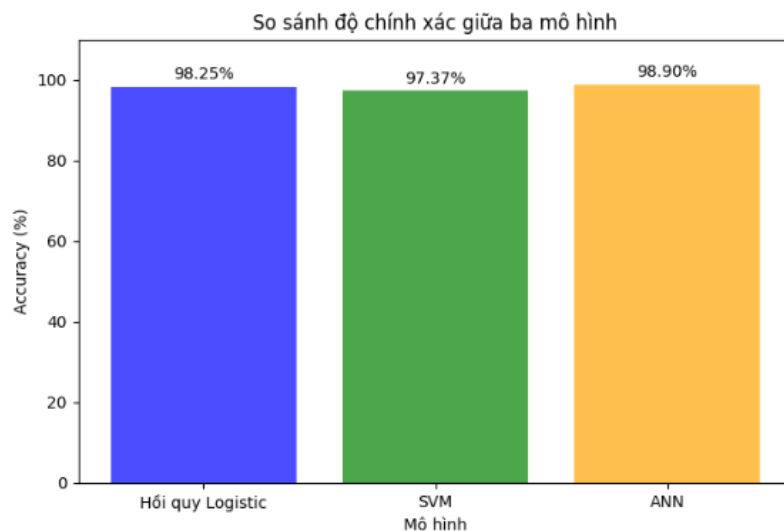
Từ các kết quả trên, thấy được rằng:

- Có 70 mẫu âm tính được dự đoán đúng, 1 mẫu âm tính bị phân loại sai thành dương tính. 2 mẫu dương tính bị phân loại thành âm tính, 41 mẫu dương tính được phân loại đúng
- Tỷ lệ chính xác ngày càng tăng theo từng epoch. Hàm mất mát cũng giảm theo từng epoch. Điều này đồng nghĩa với việc mô hình hoạt động tốt, huấn luyện càng lâu thì kết quả chính xác sẽ càng cao và mất mát sẽ càng ít. Nó cũng cho thấy rằng mô hình có hiệu suất rất tốt trong việc dự đoán.

### ***So sánh kết quả:***

Dựa trên kết quả từ biểu đồ cột với các giá trị accuracy đã cập nhật, ta có thể rút ra một số kết luận quan trọng như sau:

- Hiệu suất cao của các mô hình: Tất cả ba mô hình (hồi quy Logistic, SVM, và ANN) đều cho thấy khả năng dự đoán hiệu quả, với các giá trị accuracy đáng chú ý là 98.25%, 97.37%, và 98.90%. Điều này cho thấy các mô hình đã được huấn luyện và đánh giá một cách chính xác và có thể đưa ra dự đoán chính xác trên dữ liệu mới.
- So sánh giữa các mô hình: Dựa trên kết quả, mô hình ANN có độ chính xác cao nhất (98.90%), theo sau là hồi quy logistic (98.25%) và SVM có độ chính xác thấp hơn một chút (97.37%). Điều này có thể gợi ý rằng mô hình ANN phù hợp hơn cho bài toán này so với SVM và hồi quy logistic.



*Hình 3. 12 Biểu đồ cột so sánh độ chính xác giữa ba mô hình*

## KẾT LUẬN

Trong nghiên cứu này, em đã tiến hành dự đoán ung thư vú dựa trên tập dữ liệu Breast Cancer Wisconsin bằng cách sử dụng ba mô hình học máy khác nhau: hồi quy logistic, máy vector hỗ trợ (SVM), và mạng nơ-ron nhân tạo (ANN). Mục tiêu của em là xác định một mô hình dự đoán chính xác nhất có thể để phân loại ung thư vú là ác tính hoặc lành tính.

Mặc dù mô hình ANN đạt được kết quả tốt nhất, nhưng không nên bỏ qua khả năng của các mô hình khác như hồi quy logistic và SVM. Việc sử dụng một loạt các mô hình có thể giúp chúng ta hiểu rõ hơn về dữ liệu và có thể cải thiện hiệu suất dự đoán.

Với các độ chính xác cao như vậy, các mô hình có thể được áp dụng hiệu quả trong các ứng dụng thực tế như dự đoán, phân loại, hay nhận diện. Điều này mang lại giá trị thực tiễn và khả năng áp dụng rộng rãi của các kết quả nghiên cứu hoặc thử nghiệm này.

Trong tương lai, nghiên cứu có thể mở rộng bằng cách thử nghiệm các phương pháp tiền xử lý dữ liệu khác nhau, tăng kích thước mẫu dữ liệu, hoặc thử nghiệm các kiểu mô hình học máy khác nhau để tìm ra phương pháp dự đoán tối ưu nhất cho bài toán phân loại ung thư vú.

## TÀI LIỆU THAM KHẢO

- [1] N. T. Trung. [Trực tuyến]. Available: <https://tailieu.vn/doc/luan-van-thac-si-tai-chinh-ngan-hang-ung-dung-ky-thuat-hoc-may-trong-xay-dung-mo-hinh-du-bao-tai-ch-2338549.html>.
- [2] Karabatak, M., & Ince, M. C. (2011). An expert system for detection of breast cancer based on association rules and neural network. *Expert Systems with Applications*, 38(7), 9010-9016.
- [3] Al-Masni, M. A., Al-Azawi, R. A., & Al-Qerem, A. H. (2015). Classification of breast cancer data using artificial neural network. *International Journal of Computer Science and Information Security*, 13(7), 1-5.
- [4] Eltoukhy, M. M., & Salama, M. M. A. (2020). Automated classification of breast cancer histopathology images using deep learning. *Neural Computing and Applications*, 32, 18923-18934.
- [5] Mishra, N., Prakash, O., & Sinha, A. (2020). Feature selection and classification of breast cancer data using logistic regression. *Journal of King Saud University - Computer and Information Sciences*, 32(6), 731-736
- [6] Rajkovic, Vladislav. (1997). Nursery. UCI Machine Learning Repository. <https://doi.org/10.24432/C5P88W>.
- [7] Sousa, M. O., Carvalho, A. R., & Marques, M. A. (2019). Logistic regression applied to breast cancer risk prediction. *Journal of Biomedical Informatics*, 92, 103144.
- [8] Baser, E., & Morgül, İ. (2019). A novel ensemble model for breast cancer diagnosis. *Neural Computing and Applications*, 31(12), 8561-8571