TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

——————o0o——————

ĐỒ ÁN 2

Đề tài

HỆ THỐNG SUY LUẬN THẦN KINH MỜ THÍCH NGHI - ANFIS TRONG DỰ ĐOÁN BỆNH TIỂU ĐƯỜNG

NGUYỄN THỊ QUÝ

quy.nt185396@sis.hust.edu.vn

Ngành Toán Tin

Giảng viên hướng dẫn: **TS.Lê Chí Ngọc**

Bộ môn: **Toán Tin**

Viện: **Toán ứng dụng và tin học**

Hà Nội, 2/2022

Mục lục

Danh sách hình vẽ

Danh sách bảng

Danh sách thuật ngữ

Thuật ngữ	Ý nghĩa
ANFIS	Adaptive neuro-fuzzy inference system
	Hệ suy luận thần kinh thích ứng mờ
ANN	Artifical Neural Network
	Mạng nơron nhân tạo.
DTC	Decision Tree Algorithm
	Thuật toán cây quyết định
FNN	Feed – forward neural network
	Mạng truyền thẳng
KNN	K-nearest neighbor
LR	Logistic Regression
	Hồi quy phi tuyến
MF	Membership function
	Hàm thành viên
NBC	Naive Bayes classifier
	Phân loại Bayes
PE	Processing element
	Thành phần xử lí
PSO	Particle Swarm Optimization
	Thuật toán tối ưu bầy đàn
RFC	Random Forest Classifier
	Phân loại rừng ngẫu nhiên
RNN	Recurrent neural network
	Mạng tái tạo
SVM	Support Vector Machine

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

1. Mục đích và nội dung của đồ án:

2. Kết quả đạt được:

3. Ý thức làm việc của sinh viên:

Hà nội, ngày tháng năm 2022

Giảng viên hướng dẫn
(*ký và ghi rõ họ tên*)

Lê Chí Ngọc

LỜI CẢM ƠN

Trong suốt quá trình học tập và nghiên cứu đồ án này, dẫu cho trong tình hình dịch bệnh phức tạp, em luôn được sự quan tâm, hướng dẫn và giúp đỡ tận tình của các thầy cô giáo trong Viện Toán ứng dụng và Tin học, cùng với sự động viên giúp đỡ của bạn bè và gia đình.

Đặc biệt, để có thể hoàn thành đồ án này, em đặc biệt được bày tỏ lòng biết ơn chân thành sâu sắc tới thầy giáo TS.Lê Chí Ngọc đã trực tiếp giúp đỡ, hướng dẫn em hoàn thành đồ án này.

Tuy nhiên trong quá trình nghiên cứu đề tài, do kiến thức chuyên ngành còn hạn chế nên em vẫn còn thiếu xót khi tìm hiểu, đánh giá và trình bày đề tài. Em rất mong nhận được sự quan tâm, góp ý của các thầy/cô để đề tài em được đầy đủ, cũng như cho các dự án phát triển lên sau này.

Em xin chân thành cảm ơn!

MỞ ĐẦU

Cùng với sự phát triển của khoa học và công nghệ, thuật ngữ Data mining (khai phá dữ liệu) đã không còn là xa lạ. Nó là một lĩnh vực sáng tạo nhất của khoa học máy tính sử dụng các phương pháp thống kê khác nhau: phân lớp, phân cụm, hồi quy, nhận dạng mẫu. Phương pháp luận nằm trong khả năng tìm kiếm các mẫu và các mối quan hệ. Nó ứng dụng trong mọi khía cạnh, lĩnh vực của cuộc sống và mang lại tiềm năng ứng dụng rộng lớn. Ngày nay, ứng dụng trong lĩnh vực y học ngày càng được các nhà khoa học quan tâm. Trong hầu hết các lĩnh vực y học, data mining đã chứng minh kết quả thu được với nhiều phương pháp luận đã cải thiện độ chính xác và hiệu suất cao.

Một vấn đề ngày nay vẫn luôn đáng lo ngại là vấn đề bệnh tiểu đường thường xảy ra ở người trung niên. Một cách khái quát thì bệnh tiểu đường là một bệnh mãn tính với sự gia tăng glucozo (đường) trong máu luôn cao hơn mức bình thường do cơ thể thiếu hụt hoặc đề kháng với insulin, dẫn đến rối loạn lượng đường trong máu. Năm 2021 đánh dấu 100 năm kể từ khi phát hiện insulin , một loại thuốc ức chế bệnh trong cuộc chiến chống lại bệnh tiểu đường. Tuy nhiên, những ca mắc tiểu đường không giảm đi mà có xu hướng tăng lên, một số chuyên gia cho rằng Covid - 19 có thể là một trong những nguyên nhân.

Cứ 10 người trưởng thành có một người mắc bệnh tiểu đường

Theo số liệu do Liên đoàn Tiểu đường quốc tế (IDF)[1], hiện trên thế giới có khoảng 537 triệu người mắc bệnh tiểu đường, tỉ lệ này hiện là 1/10, nghĩa là cứ 10 người trưởng thành thì sẽ có 1 người mắc bệnh tiểu đường. IDF dự đoán đến năm 2024, tỉ lệ này sẽ tăng 1/8.

Một nghiên cứu được công bố vào tháng 2.2021[2] cho thấy tiểu đường làm tăng nguy cơ tử vong của bệnh nhân mắc Covid-19. Theo số liệu thống kê, có tới 40% bệnh nhân tử vong do Covid là người mắc bệnh tiểu đường.

Hơn nữa, IDF cũng ước tính rằng trong số 537 triệu người trưởng thành sống chung với bệnh tiểu đường trên khắp thế giới, gần một nửa (44,7%) chưa được chuẩn đoán bệnh.

Vì thế, để giảm bớt các ca bệnh tiểu đường ngày càng gia tăng, bệnh nhân cần được chuẩn đoán mắc bệnh sớm hơn, nhất là giai đoạn tiền tiểu đường bởi đây là giai đoạn trước khi cơ thể tổn thương do lượng đường trong máy không đều, và trong giai đoạn này người bệnh cũng dễ dàng thay đổi lối sống hơn.

Lý do chọn đề tài

Một trong những khía cạnh đang được tập trung nghiên cứu trong lĩnh vực y học là áp dụng data mining trong dự đoán khả năng mắc các bệnh mãn tính dựa trên một số đặc điểm.

Phân tích dự đoán khả năng mắc bệnh tiểu đường cũng đã được nghiên cứu và công bố kết quả dựa trên những thuật toán học máy hay học sâu nhưng đều mang lại kết quả chưa khả quan.

Hệ thống suy luận thần kinh mờ thích ứng - ANFIS là một hướng nghiên cứu hoàn toàn mới mẻ, với sự kết hợp sức mạnh của hai mô hình logic mờ và nơron nhân tạo đã mang đến những kết quả mong đợi vượt trội về độ chính xác.

Mục đích thực hiện đề tài

Áp dụng hệ thống ANFIS trong phân loại bệnh tiểu đường, cải thiện một số thuật toán ML truyền thống với mong muốn đạt được một hệ thống phân loại tối ưu, hỗ trợ cho các quyết định y tế.

Đối tượng và phạm vi nghiên cứu

Tập dữ liệu được sử dụng trong nghiên cứu được lấy từ cơ sở dữ liệu học tập UCI (kho lưu trữ tài liệu học Máy từ Khoa Thông tin và Khoa học Máy tính, Đại học California).

Kết quả đạt được

Độ chính xác thu được của hệ thống xây dựng là 79% Đây là một kết quả khả thi và hứa hẹn mang đến những hướng nghiên cứu xa hơn trong lĩnh vực này.

Bố cục đồ án

Chương 1: Cơ sở lí thuyết

Chương 2: Mô hình mạng ANFIS

Chương 3: Mô hình ứng dụng

Từ khóa: ANFIS, mạng nơron nhân tạo, logic mờ, bệnh tiểu đường.

Chương 1

CƠ SỞ LÍ THUYẾT

1.1 Một số thuật toán học máy

Song song với sự bùng nổ và ngày càng lớn dần của dữ liệu, các thuật toán máy học ngày càng đóng vai trò quan trọng trong khám phá tri thức, len lỏi và áp dụng trong mọi lĩnh vực. Thuật ngữ máy học dần trở nên gần với thuật ngữ "trí tuệ nhân tạo", vì khả năng học hỏi là đặc điểm chính của một thực thể. Mục đích chính của học máy là việc xây dựng các hệ thống máy tính có thể thích ứng và học hỏi kinh nghiệm của chúng.

Khám phá tri thức trong cơ sở dữ liệu là một lĩnh vực bao gồm các lý thuyết, phương pháp và kỹ thuật, cố gắng hiểu dữ liệu và trích xuất kiến thức hữu ích từ chúng. Nó được coi là một quá trình nhiều bước (lựa chọn, tiền xử lý, chuyển đổi, khai thác dữ liệu, giải thích-đánh giá). Bước quan trọng nhất trong quy trình này là khai thác dữ liệu.

Cho đến hiện tại, có rất nhiều các thuật toán học máy đã được nghiên cứu và công bố cũng như độ hiệu quả của nó. Trong nghiên cứu này, sẽ sử dụng một số thuật toán học máy thông dụng và nền tảng phù hợp với mục tiêu bài toán.

Logistic Regression - LR

Hồi quy logicstic cũng là một kĩ thuật học có giám sát, mặc dù tên của nó, nhưng nó là một mô hình phân loại chứ không phải hồi quy. Hồi quy logistic là một phương pháp đơn giản và hiệu quả hơn cho các bài toán phân loại nhị phân và tuyến tính. Đây là một mô hình phân loại, rất dễ nhận ra và đạt được hiệu suất rất tốt với các lớp có thể phân tách tuyến tính.

Hồi quy logistic về cơ bản sử dụng một hàm logistic được định nghĩa bên dưới để lập mô hình biến đầu ra nhị phân. Phạm vi hồi quy giới hạn từ 0 đến 1 và không yêu cầu mối quan hệ tuyến tính đối với đầu vào và đầu ra do thường áp dụng một phép biến đổi log phi tuyến:

$$Logisticfunction = \frac{1}{1 + e^{-x}} \tag{1.1}$$

Đối với hồi quy Logistic, dữ liệu hai class không nhất thiết là phân tách tuyến tính , tuy nhiên phân cách tìm được có dạng tuyến tính, nên mô hình này phù hợp với dữ liệu gần như có dạng phân cách tuyến tính . Một nhược điểm của phương pháp này là không làm việc với dữ liệu mà một class chứa các điểm nằm trong một vòng tròn , class kia chứa điểm nằm bên ngoài đường tròn đó. Một hạn chế nữa của nó là nó yêu cầu các điểm dữ liệu tạo ra độc lập với nhau.

Support Vector Machine - SVM

SVM là kĩ thuật học máy không tham số, ý tưởng ban đầu của thuật toán là nhằm mục đích giải quyết các vấn đề nhị phân. Nó dựa trên khái niệm giảm thiểu rủi ro, tối đa hóa và phân tách siêu phẳng và các điểm dữ liệu. Nếu ta định nghĩa mức độ phù hợp của 1 class tỉ lệ thuận với khoảng cách gần nhất từ một điểm của class đó tới đường/mặt phân chia, khoảng cách gần nhất của 1 điểm trên class tới đường/mặt phân chia được gọi là margin. Thuật toán SVM với chức năng chính là tìm ra các ranh giới tối ưu, phân chia các lớp.

Việc margin rộng hơn sẽ mang lại hiệu ứng phân lớp tốt hơn vì sự phân chia giữa hai classes là rạch ròi hơn. Việc này là một điểm khá quan trọng giúp Support Vector Machine mang lại kết quả phân loại tốt hơn so với Neural Network với 1 layer, tức Perceptron Learning Algorithm. Bài toán tối ưu trong Support Vector Machine (SVM) chính là bài toán đi tìm đường phân chia sao cho margin là lớn nhất. Đây cũng là lý do vì sao SVM còn được gọi là Maximum Margin Classifier.

K-nearest neighbor - KNN

Đây được coi như là một thuật toán học có giám sát đơn giản nhất (nhưng khá hiệu quả trong nhiều trường hợp) trong học máy. Nó cũng có thể được goi là thuật toán "lười" vì nó không học gì trong quá trình train cả, mọi tính toán đểu được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. Sử dụng linh hoạt cho cả hai bài toán: phân loại và hồi quy.

Với KNN, trong bài toán phân loại, nhãn của một điểm dữ liệu mới sẽ được suy ra trực tiếp từ k điểm dữ liệu gần nhất trong tập dữ liệu đào tạo. Việc quyết định nhãn của dữ liệu có thể quyết định bằng bầu chọn theo số phiếu giữa các điểm gần nhất, hoặc cũng có thể suy ra bằng đánh các trọng số khác nhau.

Việc bầu chọn hay đánh trọng số đều dựa trên tiêu chí khoảng cách giữa 2 điểm khác nhau. Có nhiều phép đo khoảng cách được sử dụng, thông thường nhất vẫn là norm 1 và norm 2.

Ưu điểm của KNN là hầu như độ tính toán trong quá trình đào tạo là bằng 0, việc dự đoán kết quả mới rất đơn giản, không quan tâm quá nhiều đến mức độ phân phối của các class. Tuy nhiên, một nhược điểm lớn của thuật toán này là khá nhạy cảm với nhiễu, hay việc tăng k quá lớn cũng ảnh hưởng đến độ phức tạp tính toán.

Naive Bayes classifier - NBC

Bộ phân loại Naïve Bayes (NBC) là một kỹ thuật học Bayes mang tính thực tiễn cao. Bộ phân loại này sử dụng quy tắc Bayes giả định sự độc lập giữa các yếu tố dự đoán. Nói một cách đơn giản, NBC định đề rằng sự hiện diện của một thuộc tính trong một lớp không liên quan đến sự hiện diện của bất kỳ thuộc tính nào khác. Các quy tắc Bayes dựa trên tính toán xác suất có điều kiện với thể hiện toán học như sau:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \tag{1.2}$$

trong đó
$$\begin{cases} P(B|A) : \text{là xác suất xảy ra B khi có A} \\ P(A|B) : \text{xác xuất xảy ra A khi có B} \\ P(A) : \text{xác suất xảy ra sự kiện A} \\ P(B) : \text{xác suất xảy ra sự kiện B} \end{cases}$$

Bằng cách sử dụng kỹ thuật này, tất cả các đặc trưng được cho là độc lập theo định lý Bayes. có nghĩa là không có sự phụ thuộc giữa giá trị thuộc tính trên một lớp nhất định và các thuộc tính khác. NBC hoạt động bằng cách tính xác suất có điều kiện của lớp. Vì vậy, nó giả lập các thuộc tính là độc lập có điều kiện, với biểu thức toán học:

$$P(B|A = a) = \prod_{i=1}^{d} P(B_i|A = a) \tag{1.3}$$

Mỗi tập thuộc tính B: $\{B_1, B_2, ..., B_d\}$ bao gồm d tính năng thuộc tính. Để thực hiện phân loại trên tập dữ liệu, NBC hoạt động bằng cách tính xác suất sau của mỗi lớp A bằng cách sử dụng:

$$P(A|B) = \frac{P(A)\prod_{i=1}^{d} P(B_i|A)}{P(B)} \tag{1.4}$$

Decision Tree Algorithm - DTC

Thuật toán cây quyết định cũng là một thuật toán học máy có giám sát, được sử dụng cho cả bài toán phân loại và hồi quy. Mục tiêu của thuật toán này là tạo ra một mô hình dự đoán giá trị của một biến mục tiêu, mà cây quyết định sử dụng biểu diễn cây để giải quyết vấn đề trong đó nút lá tương ứng với nhãn lớp và các thuộc tính được biểu diễn trên nút bên trong. của cây.

Quyết định phân chia chiến lược ảnh hưởng rất nhiều đến độ chính xác của cây. Các tiêu chí quyết định khác nhau đối với cây phân loại và cây hồi quy.

Cây quyết định sử dụng nhiều thuật toán để quyết định chia một nút thành hai hoặc nhiều

nút con. Việc tạo ra các nút con làm tăng tính đồng nhất của các nút con kết quả. Nói cách khác, chúng ta có thể nói rằng độ tinh khiết của nút tăng lên so với biến đích. Cây quyết định chia các nút trên tất cả các biến có sẵn và sau đó chọn phép tách dẫn đến hầu hết các nút con đồng nhất. Một số thuật toán được sửu dụng phổ biến trong cây quyết định như: ID3 (phần mở rộng của D3), C4.5 (kế thừa của ID3), CART (Cây phân loại và hồi quy), CHAID (Phát hiện tương tác tự động chi-square Thực hiện tách nhiều cấp khi tính toán cây phân loại), MARS (các đường hồi quy thích ứng đa biến)

Nếu tập dữ liệu bao gồm N thuộc tính thì việc quyết định thuộc tính nào để đặt ở gốc hoặc ở các cấp khác nhau của cây làm các nút bên trong là một bước phức tạp. Bằng cách chỉ chọn ngẫu nhiên bất kỳ nút nào làm gốc không thể giải quyết được vấn đề. Nếu chúng ta làm theo cách tiếp cận ngẫu nhiên, nó có thể cho chúng ta kết quả không tốt với độ chính xác thấp. Để giải quyết vấn đề này, một số giải pháp được đưa chẳng hạn như xem xét đánh giá các tiêu chí: Entropy, tăng thông tin, chỉ số Gini, tỷ lệ tăng, giảm phương sai, Chi-square. Các tiêu chí này được tính toán trên mọi thuộc tính và sắp xếp đặt trong cây theo thứ tự.

Random Forest Classifier - RFC

Là một thuật toán học có giám sát, sở dĩ được gọi là rừng, vì nó được xây dựng là một tập hợp các cây quyết định, với ý tưởng đóng gói sự kết hợp của các mô hình học tập để có được sự dự đoán chính xác và ổn định hơn.

Rừng ngẫu nhiên có các siêu tham số gần giống như cây quyết định và bổ sung thêm tính ngẫu nhiên cho mô hình, cây. Thay vì tìm kiếm tính năng quan trọng nhất trong khi tách một nút, nó tìm kiếm tính năng tốt nhất trong số một tập hợp con ngẫu nhiên của các tính năng. Điều này dẫn đến sự đa dạng rộng rãi và thường dẫn đến một mô hình tốt hơn.

Khi xây dựng rừng ngẫu nhiên, người ta thường quan tâm đến một số siêu tham số như: số cây trong rừng (n_estimators) khi lấy phiều bầu tối đa hay lấy giá trị trung bình của các dự đoán. Số lượng cây lớn, có thể mang lại kết quả dự đoán tốt, nhưng bù lại chi phí tính toán lớn. Một siêu tham số quan trọng khác là max_features, là số lượng tối đa các đối tượng mà rừng ngẫu nhiên xem xét để tách 1 nút. Siêu tham số quan trọng cuối cùng là min_sample_leaf. Điều này xác định số lượng lá tối thiểu cần thiết để tách một nút bên trong.

Thuật toán rừng ngẫu nhiên có những ưu điểm lớn vì tính linh hoạt, thường mang đến kết quả dự đoán tốt. Một trong những các vấn đề quan trọng của học máy là trang bị quá nhiều, nhưng hầu hết thời gian điều này sẽ không xảy ra nhờ bộ phân loại rừng ngẫu nhiên. Nếu có đủ cây trong rừng, bộ phân loại sẽ không trang bị quá nhiều cho mô hình.

AdaBoots

Boost là một thuật toán học tăng cường, đây là một kĩ thuật phổ biến để giải quyết các vấn đề phân loại nhị phân. Thuật toán này cải thiện khả năng dự đoán bằng cách chuyển đổi một

số người học yếu thành người học mạnh.

Nguyên tắc đằng sau các thuật toán tăng cường là đầu tiên chúng tôi xây dựng một mô hình trên tập dữ liệu đào tạo, sau đó mô hình thứ hai được xây dựng để sửa chữa các lỗi có trong mô hình đầu tiên. Quy trình này được tiếp tục cho đến khi và trừ khi các lỗi được giảm thiểu và tập dữ liệu được dự đoán chính xác.

AdaBoost còn được gọi là Tăng cường thích ứng là một kỹ thuật trong Học máy được sử dụng như một Phương pháp kết hợp. Thuật toán phổ biến nhất được sử dụng với AdaBoost là cây quyết định có một mức nghĩa là cây quyết định chỉ có 1 lần phân chia. Những cây này còn được gọi là Cây Quyết Định. Những gì thuật toán này làm là nó xây dựng một mô hình và đưa ra trọng số bằng nhau cho tất cả các điểm dữ liệu. Sau đó, nó chỉ định các trọng số cao hơn cho các điểm được phân loại sai. Bây giờ tất cả các điểm có trọng số cao hơn được coi trọng hơn trong mô hình tiếp theo. Nó sẽ giữ các mô hình đào tạo cho đến khi và trừ khi nhận được lỗi ghi nợ.

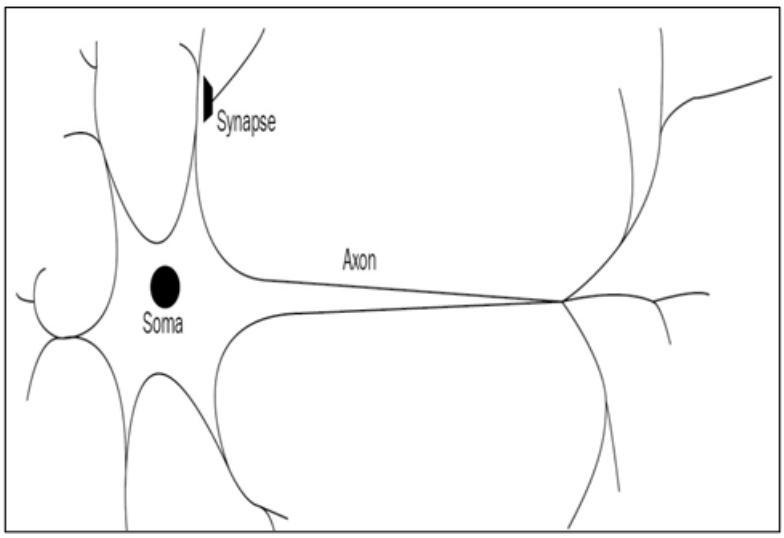
1.2 Mạng nơron nhân tạo

1.2.1 Cấu trúc và mô hình chung

Mô hình một nơron sinh học

Nơron sinh học là những tế bào thần kinh chính thức có chức năng truyền dẫn xung điện, là đơn vị cơ bản cấu tạo nên hệ thần kinh và cũng là phần quan trọng nhất của bộ não.

Về nghiên cứu về bộ não con người, toàn bộ hệ thần kinh có số lượng khoảng 100 tỉ nơron thần kinh. Một phần mô não, có kích thước bằng hạt cát, chứa tới 1 tỉ khớp thần kinh và 100.000 tế bào thần kinh, chúng có quan hệ mật thiết với nhau.



Hình 1.1: Cấu trúc của một nơron sinh học điển hình.

Mỗi noron sinh học (hình 1.1) gồm 3 thành phần cơ bản:

- Các nhánh vào hình cây (dendrites): chịu trách nhiệm nhận kí hiệu từ các noron khác.
- Thân tế bào (cell body hay soma).
- Sợi trục (axon): truyền các tín hiệu tạo ra bởi tế bào thần kinh đến một tế bào thần kinh khác kết nối với nó.

Cơ chế hoạt động: các nhánh hình cây nhận tín hiệu đầu vào đến thân tế bào. Thân tế bào tổng hợp và xử lí tín hiệu và truyền ra. Sợi trục truyền tín hiệu từ tế bào này sang nơron khác. Điểm liên kết giữa sợi trục của nơron này với nhánh hình cây của nơron khác gọi là synapse. Một số cấu trúc của nơron được xác định trước lúc sinh ra. Một số cấu trúc được phát triển thông qua quá trình học. Trong cuộc đời của các cá thể, một số liên kết mới được hình thành, một số khác bị hủy bỏ.

Như vậy, có thể thấy nơron sinh học hoạt động theo cách thức: nhận tín hiệu đầu vào, xử lí các tín hiệu này và cho ra một tín hiệu output. Tín hiệu output này sau đó sẽ được truyền đi và làm tín hiệu input cho các nơron khác.

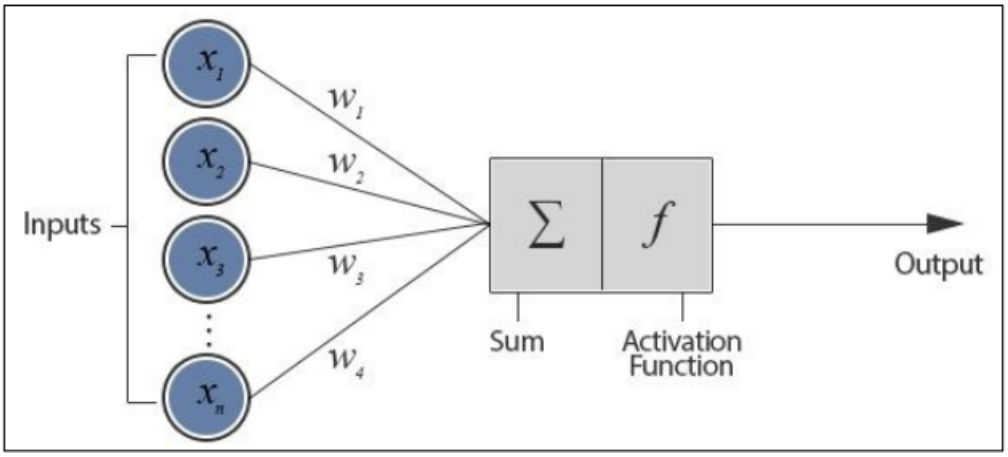
Dựa trên những hiểu biết của nơron sinh học, con người xây dựng nơron nhân tạo với hi vọng tạo nên một mô hình có sức mạnh như bộ não để ứng dụng cho những bài toán, vấn đề trong thực tế.

Nơron nhân tạo

Ngoài vai trò là nguồn gốc của trí thông minh tự nhiên, bộ não con người có thể xử lý thông tin không đầy đủ thu được do nhận thức với tốc độ rất nhanh. Lấy đặc tính sinh học này của nơron thần kinh, các nhà nghiên cứu đã cố gắng mô hình hóa bộ não con người, dẫn đến sự phát triển của **mạng nơron nhân tạo**. Ở đây, bộ não được mô hình hóa như một hệ thống phi tuyến động thời gian liên tục với một kiến trúc kết nối. Trong kiến trúc này, các nơron hay các đơn vị xử lí được kết nối với nhau qua các trọng số (weights) với mong muốn bắt chước khả năng xử lí của bộ não con người.

Mô hình toán học đầu tiên của mạng nơron sinh học được đề xuất bởi nhà thần kinh học McCulloch cùng với nhà toán học trẻ tuổi Pitts [3], thường được gọi là nơron M-P, hay còn gọi là phần tự xử lí và được kí hiệu là PE (Processing element).

Quá trình xử lí của một PE thường có cấu trúc:



Hình 1.2: Cấu trúc của một PE

Gồm các thành phần cơ bản chính:

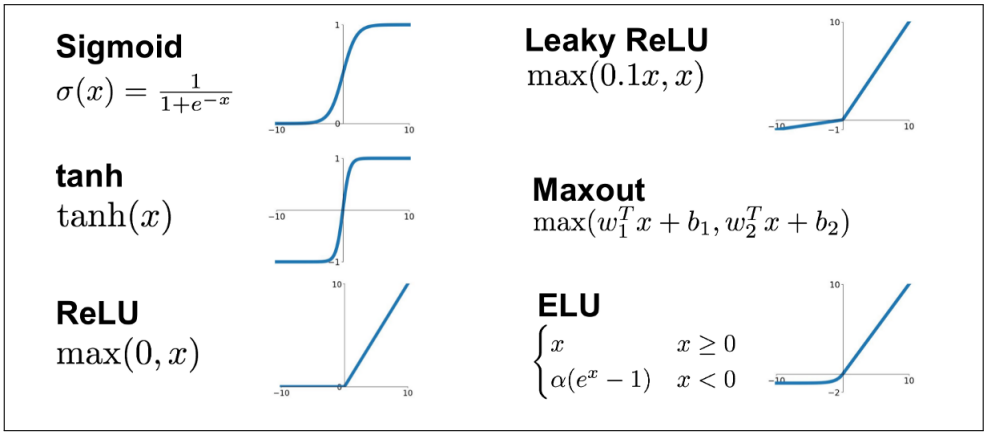
- Inputs(dữ liệu vào): Mỗi input tương ứng với 1 thuộc tính (atrribute) của dữ liệu.
- Output(kết quả): Kết quả là giải phát cho một vấn đề, hoặc cũng có thể cung cấp số liệu cho một kết nối noron khác.
- Conection Weights (trọng số liên kết): Đây là một thành phần quan trọng của một nơron, nó thể hiện mức độ quan trọng (độ mạnh) của dữ liệu đầu vào đối với quá trình xử lí thông tin (quá trình chuyển đổi dữ liệu từ nơron này sang nơron khác). Quá trình học nơron thực sự là một quá trình điều chỉnh trọng số của các Input data để được kết quả mong muốn.
- Summation Funtion (Hàm tổng): Tính tổng trọng số của tất cả các input được đưa vào mỗi nơron. Hàm tổng của một nơron đối với n input được tính theo công thức sau:

$$Y = \sum_{i=1}^{n} X_i W_i \tag{1.5}$$

- Activation funtion (hàm kích hoạt): đóng vai trò là thành phần phi tuyến tại output. Hàm này giới hạn phạm vi đầu ra của mỗi nơron. Nó nhận đầu vào là kết quả của hàm tổng của hàm kết hợp và ngưỡng đã cho.Thông thường phạm vi đầu ra của mỗi nơron được giới hạn trong khoảng [0,1] hoặc [-1,1]. Các hàm kích hoạt rất đa dạng, có thể là hàm tuyến tính hoặc phi tuyến. Việc lựa chọn hàm kích hoạt rất quan trọng có tác động lớn đến kết quả của bài toán. Thực tế, việc chọn hàm truyền sao cho phù hợp phụ thuộc vào yêu cầu đầu ra của bài toán và kinh nghiệm của người thiết lập mô hình.

Các hàm kích hoạt có thể là hàm tuyến tính hay phi tuyến, tuy nhiên các hàm kích hoạt phi

tuyến lại có vai trò rất quan trọng. Nếu không có các hàm kích hoạt phi tuyến, thì mạng nơron dù nhiều lớp nhưng vẫn có hiệu quả như một lớp tuyến tính mà thôi. Tuy nhiên các hàm kích hoạt tuyến tính vẫn được sử dụng. Áp dụng chủ yếu trong các bài toán regression, kết quả đầu Y là một số thực, hàm kích hoạt ngay phía trước Y có thể là một hàm tuyến tính. Dù thế, các hàm kích hoạt ở các lớp ẩn (hidden layer) bắt buộc phải có các yếu tố phi tuyến.



Hình 1.3: Đồ thị các hàm activation funtion

Hình 1.3 cho ta thấy công thức toán học cũng như biểu diễn hình học của một số hàm kích hoạt thông dụng.

- Hàm Sigmoid nhận giá trị trong khoảng $(0, 1)$ hay còn được gọi là hàm chuẩn hóa. Hàm tuy có đạo hàm đẹp nhưng lại bão hòa và triệt tiêu gradient, hàm Sigmoid không có trung tâm là 0 nên rất khó hội tụ. Vì thế hiện nay rất ít được ứng dụng.
- Hàm Tanh nhận đầu vào là một số thực và chuyển đổi thành một giá trị trong khoảng (-1,1). Hàm Tanh bão hòa ở hai đầu, tuy nhiên hàm Tanh đối xứng qua 0 nên khắc phục được nhược điểm của khó hội tụ như hàm Sigmoid.
- Hàm ReLU được sử dụng phổ biến. ReLU lọc các giá trị < 0. Nó nổi bật với nhiều ưu điểm như:do không bị bão hòa ở hai đầu tốc độ hội tụ nhanh hơn Tanh và Sigmoid [4], chi phí tính toán ít. Tuy nhiên nó cũng có nhược điểm với các node có giá trị nhỏ hơn 0, qua ReLU sẽ thành 0, gọi là "Dying ReLU". Điều này sẽ không có ý nghĩa với các bước linear activation kế tiếp, và các hệ số tương ứng ở node đó cũng không được cập nhập.
- Leaky ReLU là một cố gắng trong việc loại bỏ "dying ReLU". Thay vì trả về giá trị 0 với các đầu vào < 0 thì Leaky ReLU tạo ra một đường xiên có độ dốc nhỏ (hình 1.3). Có nhiều báo cáo về việc liệu Leaky ReLU có hiệu quả tốt hơn ReLU [5], nhưng hiệu quả này vẫn chưa rõ ràng và nhất quán.

- Maxout có đầy đủ ưu điểm của ReLU và LeakyReLU, tuy nhiên chi phí tính toán, bộ nhớ là lớn do sử dụng gấp đôi tham số cho mối nơron.
- ELU cũng là một hàm đề xuất để giải quyết nhược điểm của ReLU, ELU có các giá trị âm cho phép chúng đẩy các giá trị trung bình về gần 0 cùng với độ tính toán thấp. Có nghĩa là hàm có xu hướng hội tụ về phía 0 nhanh và tạo kết quả chính xác.

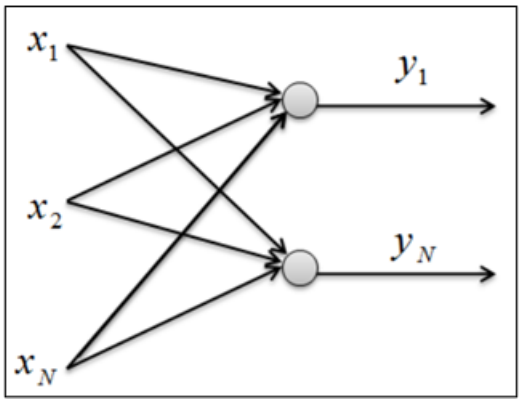
1.2.2 Phân loại và phương thức làm việc của mạng nơron nhân tạo

Mạng nơron nhân tạo (Artifical Neural Network – ANN) là một mô hình xử lí thông tin mô phỏng dựa trên hoạt động của hệ thống thần kinh của sinh vật, bao gồm số lượng lớn các nơron được gắn kết chặt chẽ với nhau hoạt động song song để xử lí thông tin. Các liên kết giữa các nơron quyết định chức năng của mạng.

Phương thức hoạt động của mạng nơron phụ thuộc vào cấu trúc mạng, trọng số liên kết giữa các nơron và quá trình xử lí bên trong nơron. Cấu trúc chung của mạng ANN bao gồm một hay nhiều tầng (layer) hay lớp. Mỗi tầng bao gồm nhiều nơron có cùng một chức năng trong mạng. Vì thế, dựa vào số tầng hay sự liên kết giữa các lớp trong mạng mà người ta phân ANN thành các nhóm khác nhau:

Phân loại theo số tầng

Mạng một tầng là cấu trúc đơn giản nhất cấu thành từ một tầng nơron, nó vừa đảm nhận chức năng là tầng vào, vừa là tầng ra.



Hình 1.4: Cấu trúc mạng 1 tầng

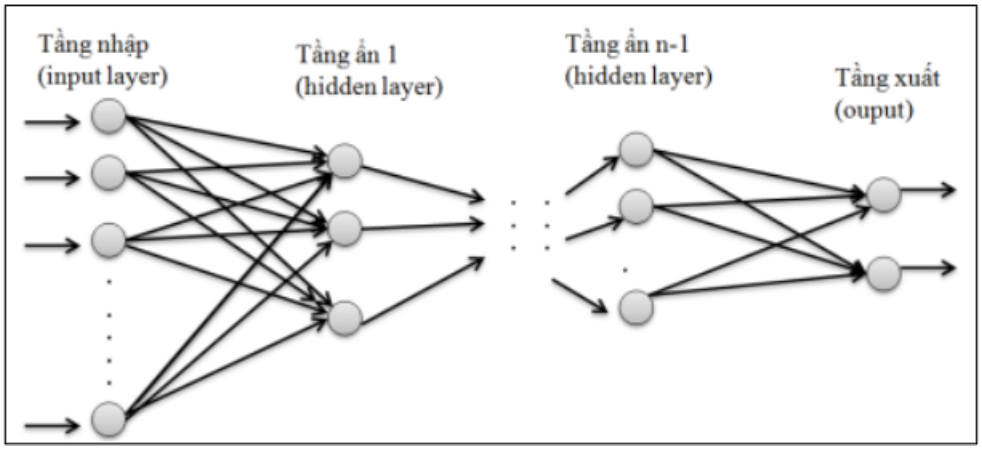
Với mỗi giá trị đầu vào $x = [x_1, x_2, ..., x_N]^T$, qua quá trình xử lí của mạng ta thu được một bộ các giá trị tương ứng $y = [y_1, y_2, ..., y_M]^T$ thông qua hàm:

$$y_i = f_i(\sum_{j-1}^{N} w_{ij}x_j - \theta_j), \qquad i = \overline{1, M} \tag{1.6}$$

Trong đó :

+ N: số tín hiệu input
+ M: số tín hiệu output
+ $W_i^T = [w_{i1}, w_{i2}, ..., w_{im}]^T$ là vector trọng số của nơron thứ i item [+] f_i là hàm kích hoạt của nơron thứ i
+ θ_i: là ngưỡng của nơron thứ i

Mạng nhiều tầng: với mạng có $n > 2$ tầng: trong đó gồm tầng nhận tín hiệu đầu vào gọi là tầng input, tầng cuối là đầu ra của dữ liệu là output. Các tầng nằm giữa tầng vào và tầng ra được gọi là tầng ẩn, có n-1 tầng ẩn hay còn gọi là các hidden layer.



Hình 1.5: Cấu trúc mạng nhiều tầng

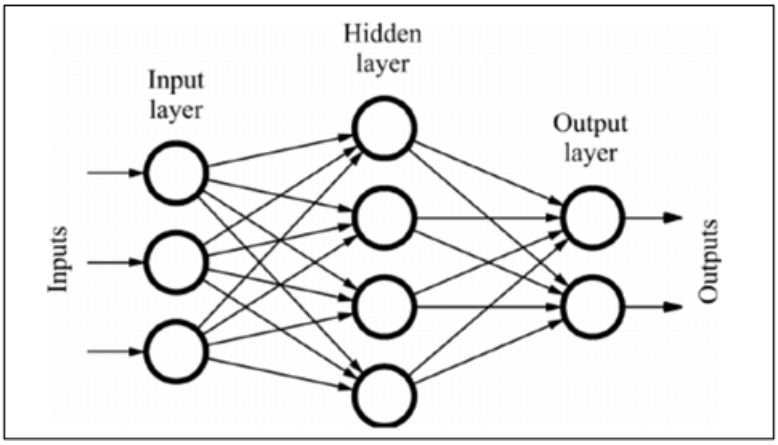
Phân loại theo cách thức liên kết

Mạng truyền thẳng (Feed – forward neural network - FNN)

Đây là cấu trúc mạng thần kinh nhân tạo đầu tiên và đơn giản nhất được phát minh ra. Trong mạng này thông tin di chuyển chỉ một chiều – chuyển tiếp- từ các nút đầu vào, qua các lớp ẩn (nếu có) và đến các nút đầu ra. Mạng chuyển tiếp thẳng không có chu kì hoặc vòng lặp. Mạng truyền thẳng ở dạng đơn giản nhất (hình 1.6) là một perceptron 1 lớp. Trong mô hình này, một loạt các đầu vào đi vào lớp và được nhân với trọng số. Mỗi giá trị sau đó được cộng lại với nhau để có được tổng các giá trị đầu vào có trọng số.

Perceptron lớp đơn là một mô hình quan trọng của mạng nơ-ron chuyển tiếp và thường được sử dụng trong các nhiệm vụ phân loại, điều chỉnh trọng số thông qua quá trình đào tạo để tạo ra giá trị đầu ra chính xác hơn. Quá trình đào tạo và học tập này tạo ra một hình thức giảm độ dốc.

Trong các perceptron nhiều lớp, quá trình cập nhật trọng số gần như tương tự, tuy nhiên

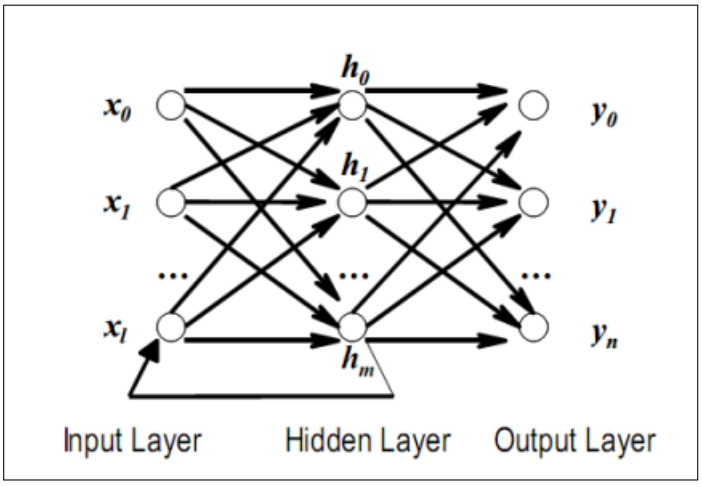


Hình 1.6: Cấu trúc 1 mạng FNN có 2 layer

quá trình này được định nghĩa cụ thể hơn là lan truyền ngược. Trong những trường hợp như vậy, mỗi lớp ẩn trong mạng được điều chỉnh theo các giá trị đầu ra do lớp cuối cùng tạo ra.

Mạng tái tạo(Recurrent neural network-RNN):

Khác với mạng nơron truyền thẳng, mạng nơron tái tạo chúng chia sẻ tham số trên mỗi lớp của mạng. Trong khi mạng truyền thẳng có trọng số khác nhau trên mỗi nút, các mạng thần kinh tái tạo chia sẻ cùng một tham số trong mỗi lớp của mạng. Những trọng số này vẫn được điều chỉnh thông qua quá trình nhân giống ngược và giảm độ dốc để tạo điều kiện cho việc học củng cố.



Hình 1.7: Cấu trúc mạng nơron tái tạo (Recurrent neural network)

1.2.3 Các luật học

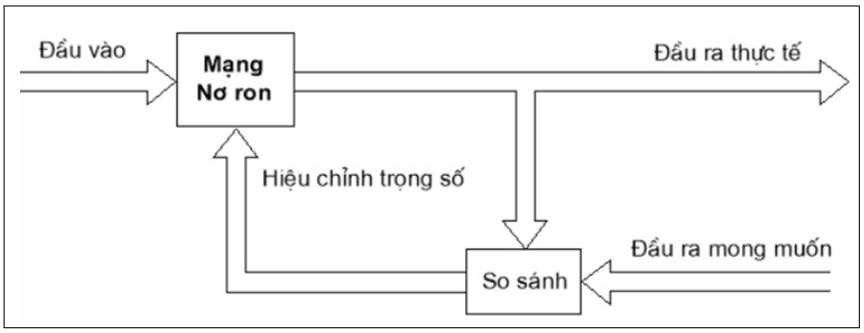
Mạng nơron hình thành chưa có tri thức, tri thức của mạng hình thành sau mỗi lần học. Mạng nơron được đào tạo và học tập bằng cách đưa vào những đầu vào kích thích và mạng

hình thành những đáp ứng, những đáp ứng phù hợp với từng loại kích thích lưu trữ. Các kĩ thuật học nhằm vào việc hiệu chỉnh các trọng số vì việc điều chỉnh, sửa đổi cấu trúc của mạng như số lớp, số nơron, kiểu và các lớp liên kết với nhau là cố định trong suốt quá trình huấn luyện... Quá trình hiểu chỉnh các trọng số để mạng "nhận biết" hay huấn luyện (training). Và dựa theo điều chỉnh việc học của mạng, có rất nhiều thuật toán sinh ra để tìm ra tập trọng số tối ưu cho việc học. Các thuật toán đó có thể chia thành ba nhóm chính: học có giám sát, học không giám sát và học củng cố.

Học có giám sát (Supervised learning)

Mạng học có giám sát hay học "có thầy" được định nghĩa bằng cách sử dụng các tập dữ liệu được gắn nhãn để huấn luyện các thuật toán phân loại dữ liệu hoặc dự đoán kết quả một cách chính xác. Khi dữ liệu đầu vào được đưa vào mô hình, nó sẽ điều chỉnh trọng số của nó cho đến khi mô hình được lắp một cách thích hợp, điều này xảy ra như một phần của quá trình xác nhận chéo (validtion). Học tập có giám sát giúp các tổ chức giải quyết nhiều vấn đề trong thế giới thực trên quy mô lớn, chẳng hạn như phân loại thư rác trong một thư mục riêng biệt từ hộp thư đến của bạn, nhận dạng hình ảnh, đối tượng, phân tích dự đoán, phân tích tâm lí khách hàng,...

Thuật toán đo độ chính xác của nó thông qua hàm mất mát, điều chỉnh cho đến khi sai số được giảm thiểu đến mức tối thiểu.



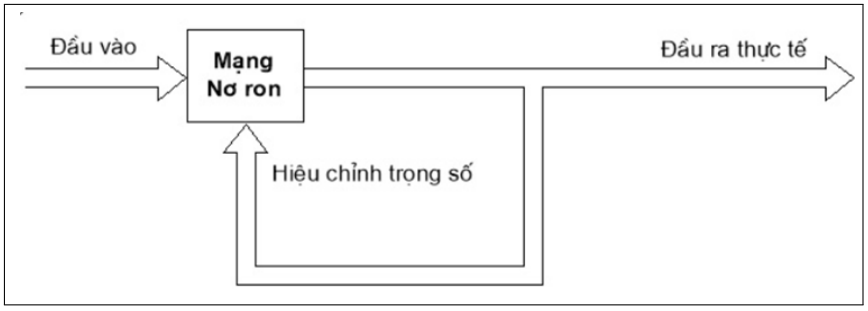
Hình 1.8: Mô hình học giám sát (Supervised learning)

Học không giám sát (Unsupervised Learning)

Trong học không giám sát thì không có bất kỳ một thông tin phản hồi từ môi trường. Mạng chỉ được cung cấp các dữ liệu đầu vào mà không có dữ liệu chính xác đầu ra. Mạng phải tự tìm ra các đặc tính, quy luật, tương quan trong dữ liệu đầu vào và tập hợp lại để tạo đầu ra. Khi tự tìm ra các đặc điểm này, mạng đã trải qua các thay đổi về tham số của nó. Quá trình này được gọi là tự tổ chức. Mạng nơron điển hình được huấn luyện theo kiểu tự tổ chức.

Với mô hình học này thường sử dụng các thuật toán học máy để phân tích và phân cụm dữ liệu không được gắn nhãn. Các thuật toán này phát hiện ra các mẫu hoặc nhóm dữ liệu ẩn

mà không cần sự can thiệp của con người.



Hình 1.9: Mô hình mạng học không giám sát (Unsupervised Learning)

Chương 2

MÔ HÌNH MẠNG ANFIS

2.1 Hệ mờ và mạng nơ ron

2.1.1 Một số khái niệm cơ bản

Logic mờ là một cách tiếp cận tính toán dựa trên "mức độ chân lý" chứ không phải là logic Boolean "đúng hoặc sai" (1 hoặc 0) thông thường mà máy tính hiện đại dựa trên đó.

Ý tưởng về logic mờ đầu tiên được Lotfi Zadenh của Đại học California tại Berkelry phát triển vào những năm 1960.

Logic mờ được sử dụng để bắt chước suy luận và nhận thức của con người. Thay vì các trường hợp chân lý nhị phân hoàn toàn, logic mờ bao gồm 0 và 1 như các trường hợp cực trị của chân lý nhưng với các mức độ chân lý trung gian khác nhau.

Biến ngôn ngữ tự nhiên (Linguistic variables): là các biến mang giá trị ngôn ngữ tự nhiên, trong đó các từ tập trung bởi các tập mờ.

Các quy tắc If-then mờ có thể mô hình hóa các khía cạnh định tính của tri thức con người và các quá trình lí luận mà không cần sử dụng các định lượng chính xác.

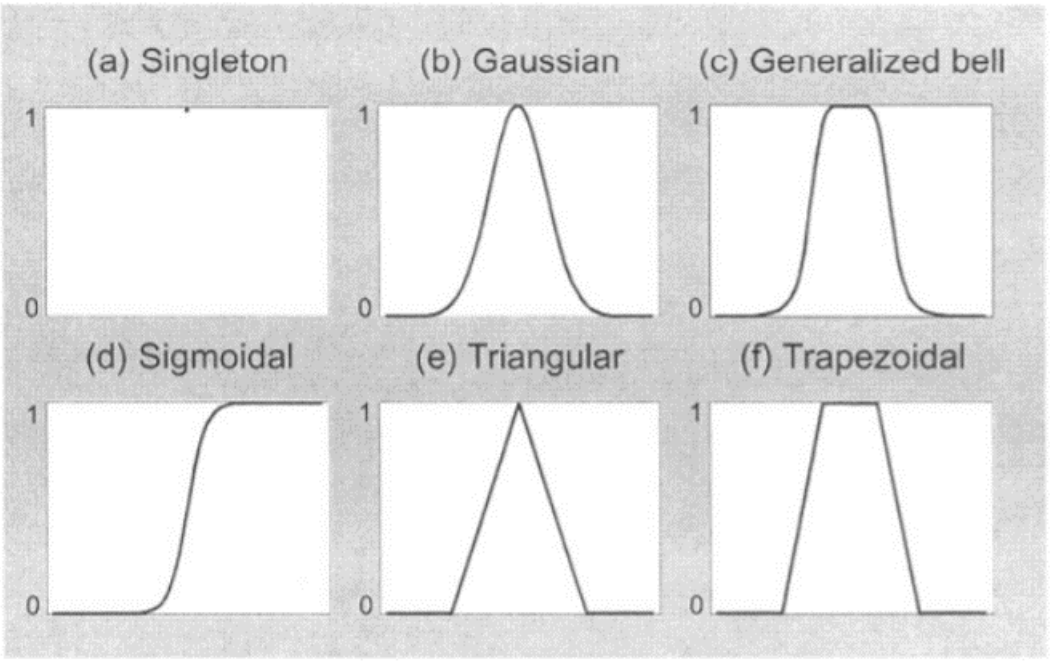
Hàm thành viên (membership funtion): hàm đại diện cho mức độ của sự thật trong một tập mờ.

Tập mờ(fuzzy set): là tập đặc trưng bởi các hàm thành viên. Mức độ thành viên nằm trong khoảng $[0, 1]$. Nếu nó là 0 thì giá trị không thuộc tập mờ đã cho, còn nếu là 1 thì giá trị đó hoàn toàn thuộc tập mờ. Bất kỳ giá trị nào từ 0 đến 1 thể hiện mức độ không chắc chắn của giá trị đó trong tập hợp. Các tập mờ này thường được mô tả bằng từ ngữ, và do đó, bằng cách gán đầu vào hệ thống cho các tập mờ, chúng ta có thể suy luận với nó theo cách tự nhiên về mặt ngôn ngữ.

2.1.2 Suy luận mờ

Hệ thống suy luận mờ gồm 1 tập hợp các quy tắc IF-then mờ hoặc câu lệnh điều kiện mờ là biểu thức có dạng If A then B, trong đó A và B là các nhãn của tập mờ được đặc trưng bởi các hàm thành viên (membership funtion MF).

Một số hàm thành viên thông dụng:



Hình 2.1: Một số hàm MF thông dụng

Với công thức toán học:

+ Gausian MF

  Hàm thành viên Gaussian thường được viết là $Gausian(x, c, s)$, trong đó c, s lần lượt là trung bình và độ lệch chuẩn.

$$\mu_A(x, c, s, m) = e^{-\frac{1}{2}(|\frac{x-c}{s}|)^m} \tag{2.1}$$

  ở đó, c là giá trị trung tâm, s là chiều rộng, m là nhân tố mờ.

+ Generalized bell MF

  Hàm thành viên Generalized bell có 3 tham số: a đại diện chiều rộng, c đại diện cho giá trị trung tâm, b đại diện cho độ dốc.

$$gbellmf(x, a, b, c) = \frac{1}{1 + |\frac{x-c}{b}|^{2b}} \tag{2.2}$$

+ Sigmoid MF

Hàm thành viên Sigmoid có 2 tham số: a đại diện cho độ dốc tại điểm giao nhau $x = c$

$$sigmf(x, a, c) = \frac{1}{1 + e^{-a(x-c)}} \tag{2.3}$$

+ Triangular MF

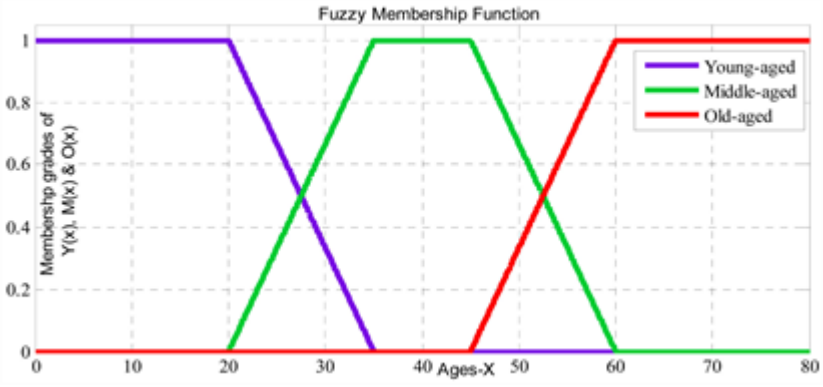
$$\mu_A(x, a, b, c) = \begin{cases} 0 & if \quad x \leq a \\ \frac{x-a}{b-a} & if \quad a \leq x \leq b \\ \frac{c-x}{c-b} & if \quad b \leq x \leq c \\ 0 & if \quad x \geq c \end{cases} \tag{2.4}$$

+ Trapezoidal MF

$$\mu_A(x, a, b, c, d) = max(min(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}), 0) \tag{2.5}$$

Các quy tắc If-then thường được sử dụng để nắm bắt các phương thức lập luận mơ hồ, không rõ ràng đóng vai trò thiết yếu trong khả năng đưa ra quyết định của con người trong một môi trường không chắc chắn hoặc không chính xác.

Ví dụ 2.1. Xét 3 tập mờ : $Y : X \to [0, 1]; M : X \to [0, 1], O \to [0, 1]$ đại diện cho các đặc điểm: trẻ, trung niên, già. Với hàm thành viên hình thang, ta có mức độ giá trị của các thành viên được xác định như sau:



Hình 2.2: Đồ thị phân bố mức độ thành viên

Với công thức toán học như sau:

Young_aged (trẻ), $Y(x) = \begin{cases} 1; & x \leq 20 \\ \frac{35-x}{15}; & 20 < x < 35 \\ 0; & x \geq 35 \end{cases}$

Middle_age(trung niên), $M(x) = \begin{cases} 0, & x \leq 20 \\ \frac{x-20}{15}; & 20 < x < 35 \\ 1; & 35 \geq x \leq 45 \\ \frac{60-x}{15}; & 45 < x < 60 \\ 0; & x \geq 60 \end{cases}$

Old_aged (già), $O(x) = \begin{cases} 1; & x \leq 45 \\ \frac{x-45}{15}; & 45 < x < 60 \\ 1; & x \geq 60 \end{cases}$

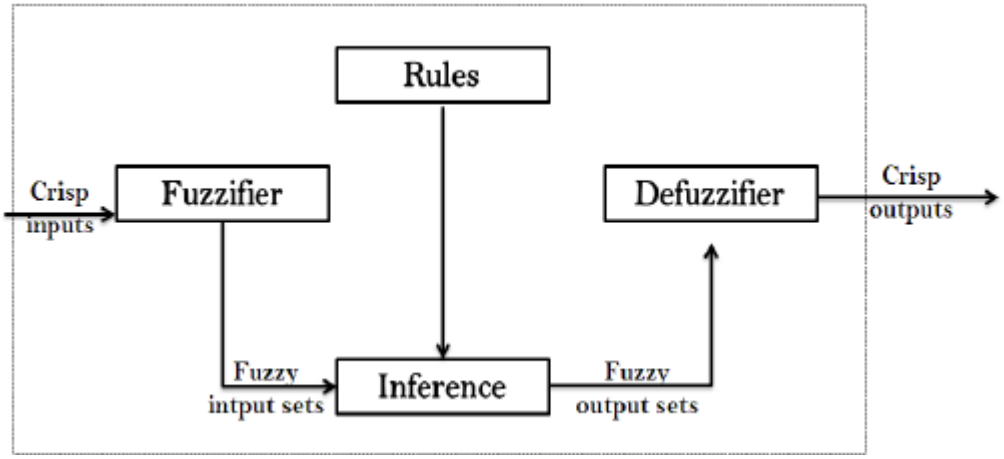
2.1.3 Giải mờ

Giải mờ (Defuzzification) là quá trình định lượng số cho các logic mờ, các tập mờ đã cho và các cấp độ thành viên tương ứng. Đây là quá trình ánh xạ một tập hợp mờ thành một tập sắc nét: gồm các quyết định cụ thể hoặc giá trị thực.

Một số phương pháp giải mờ thông dụng:

+ Max-member: Lấy các gí trị đầu ra đỉnh.
+ Phương pháp Centroid
+ Phương pháp trung bình trọng số: Áp dụng cho các hàm thành viên đối xứng.
+ Phương pháp trung bình cực đại.
+ Phương pháp lấy trung tâm của các tổng.
+ Phương pháp lấy tâm của vùng lớn nhất.

2.1.4 Cấu trúc hệ thống



Hình 2.3: Cấu trúc của một hệ thống logic mờ

Cấu trúc hệ thống suy luận mờ (hình 2.3) hay còn gọi là hệ thống dựa trên các quy tắc mờ, mô hình mờ, bộ nhớ kết hợp mờ, hoặc bộ điều khiển mờ khi được sử dụng làm bộ điều khiển. Về cơ bản, cấu trúc hệ thống suy luận mờ gồm 5 khối chức năng chính:

- A rule base: bao gồm 1 tập hợp các quy tắc If-then mờ
- A database: một cơ sở dữ liệu xác định các hàm thành viên của các tập mờ được sử dụng trong các luật mờ.
- A decision-making: các phép toán suy luận về các quy tắc.
- A fuzzication interface: biến các biến đầu vào sắc nét thành các mức độ phù hợp với các biến ngôn ngữ.
- A defuzzification interface: Biến đổi các kết quả mờ của suy luận thành một đầu ra rõ ràng.

Tùy thuộc vào mục đích và loại suy luận và các quy tắc IF-THEN được sử dụng, hầu hết các hệ thống suy luận mờ có thể phân loại thành 3 loại:

- Type 1: đầu ra tổng thể là giá trị trung bình có trọng số của đầu ra sắc nét của mỗi quy tắc do độ kích hoạt của quy tắc gây ra và các hàm thành viên đầu ra. Các hàm liên thuộc (membership funtions) được sử dụng trong lược đồ này phải đơn điệu không giảm.
- Type 2: kết quả đầu ra mờ tổng thể được suy ra bằng cách áp dụng hoạt động tối đa cho các đầu ra mờ đủ điều kiện. Một số đầu ra sắc nét hay được sử dụng dựa trên đầu ra mờ tổng thể: Tâm của diện tích (COA), đường phân giác của diện tích (BOA), trung

bình của cực đại (MeOM), ...

- Type 3: quy tắc IF-THEN mờ của Takagi và Sugeni với đầu ra của mỗi quy tắc là sự kết hợp tuyến tính của các biến đầu vào cộng với một số hạng không đổi và đầu ra cuối cùng là giá trị trung bình có trọng số của mỗi quy tắc.

2.2 Mô hình mạng ANFIS

2.2.1 Giới thiệu chung

Khi khảo sát mạng nơron va logic mờ, ta thấy mỗi loại đều có điểm mạnh, điểu yếu riêng của nó. Đối với logic mờ, ta dễ dàng thiết kế một hệ thống mong muốn chỉ bằng các luật IF-THEN gần với việc xử lí của con người. Với đa số ứng dụng thì điều này cho phép tạo lời giải đơn giản, trong khoảng thời gian ngắn hơn. Thêm nữa, ta dễ dàng sử dụng những hiểu biết của mình về đối tượng để tối ưu hệ thống một cách trực tiếp.

Tuy nhiên, đi đôi với các ưu điểm hệ điều khiển mờ, còn tồn tại một số khuyết điểm như việc thiết kế và tối ưu hóa hệ logic mờ đòi hỏi phải có một số kinh nghiệm về điều khiển đối tượng, đối với những người thiết kế lần đầu đó hoàn toàn không đơn giản. Mặt khác còn hàng loạt các câu hỏi khác đặt ra cho người thiết kế mà nếu chỉ dừng lại ở tư duy logic mờ thì hầu như chưa có lời giải. Ví dụ: số tập mờ trong mỗi biến ngôn ngữ cần bao nhiêu là tối ưu? Hình dạng các tập mờ như thế nào? Vị trí mỗi tập mờ ở đâu? Việc kết hợp các tập mờ như thế nào? Trọng số của các luật điều khiển bằng bao nhiêu?

Đối với mạng nơron, chúng có ưu điểm như xử lí song song nên tốc độ xử lý rất nhanh, mạng nơron có khả năng học hỏi, ta có thể huấn luyện mạng để xấp xỉ một hàm phi tuyến bất kì, đặc biệt khi đã biết một tập dữ liệu vào/ra... Song nhược điểm của mạng nơron là khó giải thích rõ ràng hoạt động của mạng nơron như thế nào. Do đó việc chỉnh sửa trong mạng rất khó khăn.

Hai tiêu chí cơ bản trợ giúp cho người thiết kế ở logic mờ và ở mạng nơron thể hiện trái ngược nhau:

Tiêu chí	Mạng nơron	Logic mờ
Thể hiện tri thức	Không tường minh, khó giải thích sửa đổi.	Tường minh, dễ kiểm chứng hoạt động và dễ sửa đổi.
Khả năng học	Có khả năng học thông qua các tập dữ liệu.	Không có khả năng học, người thiết kế phải tự định nghĩa và thiết kế tất cả.

Bảng 2.1: So sánh mạng nơron và logic mờ

Từ những phân tích trên, ta thấy nếu kết hợp logic mờ và mạng nơron, ta có một hệ lai với ưu điểm của cả hai: logic mờ cho phép thiết kế hệ dễ dàng, tường minh trong khi mạng

nơron cho phép học những gì mà ta yêu cầu về bộ điều khiển. Nó sửa đổi các hàm phụ thuộc về hình dạng, vị trí và sự kết hợp,... hoàn toàn là tự động. Điều này làm giảm bớt chi phí khi phát triển hệ thống.

Từ đó mô hình ANFIS ra đời đại diện cho sự kết hợp của cả 2 cách tiếp cận, mang đến những ưu điểm và hiệu quả vượt bậc.

2.2.2 Hệ thống suy luận thần kinh thích ứng mờ-ANFIS

Mô hình hệ thống mạng thần kinh thích ứng mờ - ANFIS là một loại mạng nơron nhân tạo dựa trên hệ thống suy luận mờ Takagi – Sugeno. Kỹ thuật này phát triển vào đầu những năm 1990.

Hệ thống này hình thành tích hợp cả mạng nơron và các quy tắc logic mờ, có tiềm năng kết hợp các ưu điểm của hai phương pháp trong một mô hình duy nhất.

Hệ thống suy luận của nó tương ứng với một tập hợp các quy tắc IF-THEN mờ có khả năng học để tính gần đúng các hàm phi tuyến.

Là kiểu mạng có cấu trúc tương tự mạng nơron, nó ánh xạ các đầu vào qua các hàm thành viên vào với các thông số tương ứng và sau đó là thông qua các hàm ra với các tham số tương ứng tạo nên các đầu ra có thể được sử dụng để giải thích ánh xạ vào/ra. Các thông số tương ứng với hàm thành viên sẽ thay đổi thông qua quá trình học. Việc tính toán các tham số này (hoặc việc điều chỉnh chúng) thực hiện dễ dàng bằng véc tơ gradient nó đưa ra giới hạn theo cách tốt cho hệ thống suy diễn mờ được mô hình hoá dữ liệu vào/ra theo tập các tham số nhất định. Ta đã biết, véc tơ gradient được áp dụng cho một vài thủ tục tối ưu cốt để điều chỉnh các tham số sao cho giá trị sai số là nhỏ nhất (thường được định nghĩa bằng tổng bình phương sai lệch giữa đầu ra hiện thời và đầu ra mong muốn). Anfis sử dụng điều đó theo giải thuật lan truyền ngược hoặc kết hợp sự ước lượng bình phương cực tiểu và sự lan truyền ngược cho sự ước lượng tham số hàm thành viên.

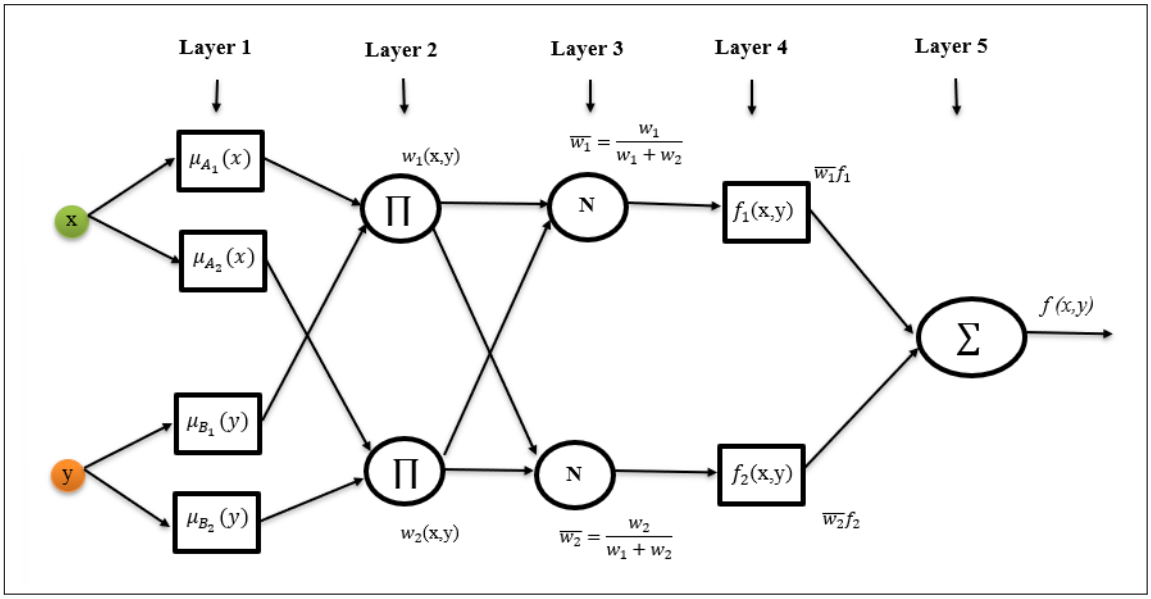
2.2.3 Cấu trúc ANFIS

Kiến trúc hệ thống ANFIS [6](hinh 2.4) gồm 5 lớp: lớp đầu vào và đầu ra, ba lớp ẩn đại diện cho hàm thành viên và luật mờ.

Để đơn giản, ta xét hệ thống suy luận mờ kiểu Takagi-Sugeno[7] có hai đầu vào x, y và đầu ra z. Đầu vào x đại diện bởi tập mờ A_1, A_2 , đầu vào y đại diện bằng các tập mờ B_1, B_2 và đầu ra z của tập mờ W_1, W_2. Ở đó luật R_K có thể được trình bày:

$$R_k : \text{IF}\mu_{A_i}\text{AND}\mu_{B_i}\text{THEN}f = p_k x + q_k y + r_k$$

Trong đó: k là số luật, μ là các hàm MF, p_k, q_k, r_k là các tham số tuyến tính tương ứng với từng luật.



Hình 2.4: Cấu trúc mô hình ANFIS

Layer 1: Lớp mờ (Fuzzification layer) Lớp nhận các giá trị đầu vào và xác định các hàm thành viên thuộc về nó.
Mỗi node i trong layer là một node thích nghi của 1 node hàm hành viên.

$$O_i^1 = \mu_{A_i}(x), \qquad i = 1, 2, ... \tag{2.6}$$
$$O_i^1 = \mu_{B_i}(x), \qquad i = 1, 2, ... \tag{2.7}$$

Với x, y là các input tới node i, A_i, B_i là các biến ngôn ngữ tự nhiên.
Layer 2: Lớp các quy tắc (Rule layer) tính toán cường độ kích hoạt của mỗi quy tắc mờ thông qua toán tử $\prod$.

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}, \quad i = 1, 2, ... \tag{2.8}$$

Layer 3 Tính toán chuẩn cường độ kích hoạt của quy tắc từ layer trước nó.

$$O_i^3 = \overline{W_i} = \frac{w_i}{\sum w_i}, \quad i = 1, 2, ... \tag{2.9}$$

Layer 4: Lớp giải mờ Mỗi node đại diện cho một phần hệ quả của luật mờ, các hệ quả của hệ số tuyến tính của luật có thể được huấn luyện:

$$O_i^4 = \overline{w_i} \cdot f_i = \overline{w_i} \cdot p_k x + q_k y + r_k, i = 1, 2, ... \tag{2.10}$$

trong đó p_k, q_k, r_k là các tham số tuyến tính.

Layer 5: Lớp kết quả đầu ra Tính tổng đầu ra của tất cả các tín hiệu đến và đưa ra kết quả sắc nét:

$$O_i^5 = \sum_{i=1}^{n} \overline{w_i} \cdot f_i = \sum_{i=1}^{n} \overline{w_i} \cdot (p_k x + q_k y + r_k) \tag{2.11}$$

2.2.4 Thuật toán huấn luyện mô hình ANFIS

Thuật toán truyền thống

Từ lớp kết quả đầu ra :

$$O_i^5 = \sum_{i=1}^{n} \overline{w_i} \cdot f_i = \sum_{i=1}^{n} \overline{w_i} \cdot (p_k x + q_k y + r_k)$$

ta có biểu diễn:

$$O_i^5 = \overline{w_1} x p_1 + \overline{w_1} y q_1 + \overline{w_1} r_1 + \overline{w_2} x p_2 + \overline{w_2} y q_2 + \overline{w_2} r_2$$

là một tổ hợp tuyến tính của các tham số hệ quả $p_1, q_1, r_1, p_2, q_2, r_2$.

Phương pháp bình phương nhỏ nhất có thể được sử dụng để xác định giá trị tối ưu của các tham số này một cách dễ dàng. Tuy nhiên, khi tham số khởi tạo ban đầu không được cố định, không gian tìm kiếm trở nên lớn hơn và quá trình hội tụ trở nên chậm hơn.

Thuật toán học lai với sự kết hợp của phương pháp bình phương nhỏ nhất và phương pháp gradient descent áp dụng và giải quyết hiệu quả vấn đề này. Thuật toán kết hợp bao gồm đường chuyển tiếp tiến và một đường chuyển ngược. Phương pháp bình phương nhỏ nhất (chuyển tiếp) được sử dụng để tối ưu hóa các thông số kết quả với các tham số khởi tạo ban đầu. Một khi tìm thấy các tham số hệ quả tối ưu, phương thức chuyển ngược gradient descent được sử dụng để điều chỉnh các tham số khởi tạo tương ứng với các tập mờ một cách tối ưu.

Lỗi đầu ra được sử dụng để điều chỉnh các tham số khởi tạo bằng thuật toán lan truyền ngược[8].

Thuật toán học lai này đã được chứng minh về độ hiệu quả trong việc học đào tạo ANFIS (Jang,1993 [6]).

Thuật toán tối ưu bầy đàn (PSO)

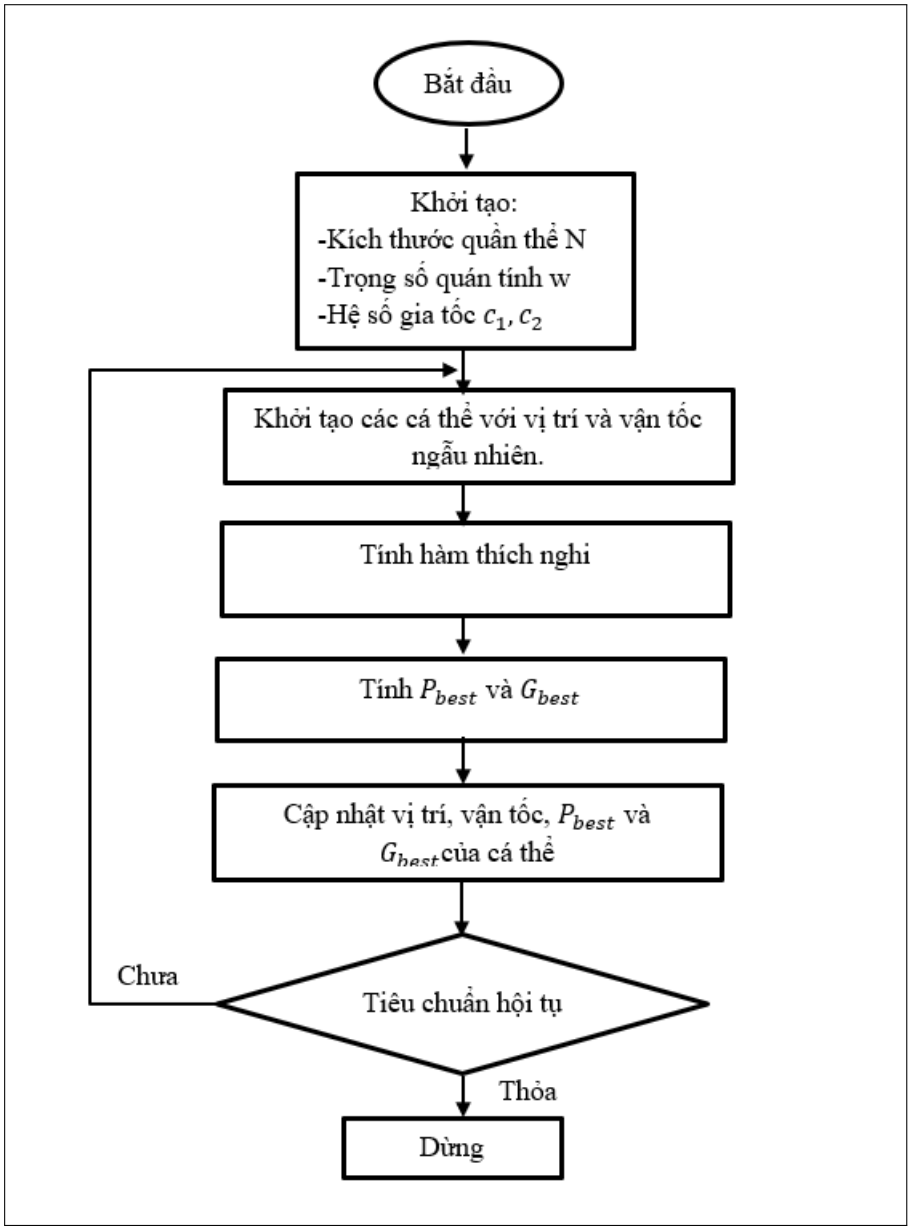
Dựa trên khái niệm trí tuệ bầy đàn như chim, cá để trong quá trình tìm kiếm thức ăn để áp dụng tìm kiếm lời giải cho các bài toán tối ưu hóa trên một không gian tìm kiếm nào đó. Đây được coi như là một thuật toán tiến hóa quần thể, với sự tương tác giữa các các thể trong một quần thể để khám phá một không gian tìm kiếm. Thuật toán được giới thiệu đầu tiên

vào năm 1995 bởi James Kennedy và Russell C.Eberhart từ một nghiên cứu mô tả hành vi xã hội của các loài sinh vật sống bầy đàn và nhận ra tiềm năng của nó trong các bài kiểm tra tối ưu hóa. Từ đó, một thuật toán tối ưu hóa mới được ra đời và công bố [9] và dần trở thành một trong những thuật toán hữu ích và phổ biến nhất để giải quyết các vấn đề tối ưu áp dụng trong nhiều lĩnh vực.

Từ quan điểm toán học, các trọng số đặc trưng có thể xem như một vấn đề tối ưu không lồi phi tuyến với tối thiểu đa cục bộ địa phương. Kỹ thuật tối ưu bầy đàn (Particle Swarm Optimization - PSO) có thể tìm ra giá trị tối ưu toàn cục với thiết lập điều kiện khởi tạo đơn giản. Với việc sử dụng các phép toán nguyên thủy do đó tiết kiệm chi phí tính toán về bộ nhớ lưu trữ và tốc độ xử lí.

PSO được khởi tạo bằng một nhóm cá thể ngẫu nhiên và sau đó tìm nghiệm tối ưu bằng cách cập nhật các thế hệ. Trong mỗi thế hệ, mỗi cá thể được cập nhật theo hai vị trí tốt nhất. Vị trí đầu tiên là vị trí tốt nhất mà mỗi cá thể đạt được cho tới thời điểm hiện tại, gọi là P_{best}, vị trí thứ hai là nghiệm tối ưu toàn cục G_{best} là vị trí tốt nhất trong tất cả các quá trình tìm kiếm của cả quần thể từ trước tới thời điểm hiện tại. Nói cách khác, mỗi cá thể trong quần thể cập nhật vị trí của nó theo vị trí tốt nhất của nó và của cả quần thể tính tới thời điểm hiện tại.

Sơ đồ thuật toán PSO có thể được biểu diễn như sau:



Hình 2.5: Sơ đồ thuật toán PSO

Với các bước cơ bản:

- Bước 1: Khởi tạo quần thể: Chọn ngẫu nhiên 1 quần thể có N cá thể có các vị trí và vân tốc ban đầu:$\{x_1, x_2, ..., x_n\}, \{v_1, v_2, ..., v_n\}$. Khi đó mỗi cá thể tương ứng với một hàm mục tiêu $f(x_0)$ và quần thể tương ứng với tập giá trị hàm mục tiêu: $\{f(x_0), f(x_1), ..., f(x_n)\}$, di chuyển đến bước tiếp theo.
- Bước 2: Tìm vị trí tốt nhất của các cá thể và của cả quần thể: trong quá trình tìm kiếm, mỗi cá thể chịu tác động của 2 thông tin đó là vị trí tốt nhất của chính cá thể đó trong quá khứ P_{best} và vị trí tốt nhất mà cả bầy đàn đạt được trong quá khứ G_{best}.

- Bước 3: Cập nhật vị trí và vận tốc của các cá thể: Mỗi cá thể sẽ điều chỉnh vận tốc và vị trí của mình theo các cá thể có giá trị thích nghi tốt nhất với:

$$X_{i,j}^{k+1} = X_{i,j}^{k} + V_{i,j}^{k+1} \tag{2.12}$$

$$V_{i,j}^{k+1} = w \times V_{i,j}^{k} + c_1 r_1((P_{best})_{i,j}^{k} - X_{i,j}^{k}) + c_2 r_2((G_{best})_{j}^{k} - X_{i,j}^{k}) \tag{2.13}$$

Với $i = 1, 2, ..., N; j = 1, 2, .., n$, c_1, c_2 là các hệ số gia tốc, t là biến vòng lặp, R_1, R_2 là các biến ngẫu nhiên trong khoảng $[0, 1]$, w là trọng số quán tính.

- Bước 4: Sau mỗi vòng lặp, các cá thể được cập nhật vị trí và đánh giá giá trị hàm mục tiêu, giá trị tốt nhất cá thể đạt được cũng được cập nhật. Giá trị vị trí tốt nhất mới của cá thể i trong vòng lặp thứ $t + 1$ được định nghĩa:

$$P_{best}^{i}(t+1) = \begin{cases} x_i(t+1), \text{if} f(x_{t+1}^{i} \leq f(P_{best}^{i}(t)) \\ p_i(t+1), \text{if} f(x_{t+1}^{i} > f(P_{best}^{i}(t)) \end{cases} \tag{2.14}$$

Sau khi cập nhật các giá trị vị trí trong tập hợp P, việc xác định chỉ số g cho vị trí có giá trị hàm mục tiêu tốt nhất sẽ hoàn thành một vòng lặp cho giải thuật PSO.

2.3 Một số ứng dụng của mạng ANFIS

Logic mờ nói chung và mạng ANFIS nói riêng đang ngày càng được áp dụng phổ biến, len lỏi phát triển trong mọi lĩnh vực công nghệ tiên tiến.

Ứng dụng nổi bật của ANFIS là trong lĩnh vực điều khiển, tự động hóa thông minh: chuyển động học nghịch đảo của robot[10][11], hệ thống điện tự động[12],...

Bên cạnh đó, nó còn được áp dụng rộng rãi trong các bài toán nhận dạng mẫu: kí tự in, phân loại,... hay các hệ thống hồi quy phi tuyến tính, hệ thống xác thực không tuyến tính. Mô hình ANFIS dự đoán hồi quy được áp dụng nghiên cứu rỗng rãi trong mọi lĩnh vực của cuộc sống. Điển hình trong lĩnh vực y học, ANFIS cho thấy tiềm năng tối ưu của mình cùng với các kinh nghiệm của chuyên gia trong các mô hình dự đoán, chuẩn đoán bệnh: dự đoán bệnh tiểu đường và chuẩn đoán ung thư[13], phân đoạn khối u não[14],...Từ đó ANFIS với công cụ tính toán thông minh giúp ích rất lớn cho các chuyên gia trong việc chuẩn đoán bệnh một cách nhanh chóng và chính xác. Hay việc chuẩn đoán bệnh một cách nhanh chóng với độ chính xác cao mang sẽ có lợi ích lớn trong phòng tránh, hay các biện pháp giảm khả năng mắc bệnh hay chữa trị.

Hay với dữ liệu chuỗi thời gian, ANFIS cũng cho thấy sự hiệu quả lớn khi kết hợp với các mô hình dự đoán chuỗi thời gian truyền thống: AR-ANFIS[15], ANFIS dự đoán chứng khoán[16], dự đoán thời tiết[17],...

Chương 3

MÔ HÌNH ỨNG DỤNG

3.1 Dữ liệu

Tập dữ liệu được sử dụng trong nghiên cứu được lấy từ cơ sở dữ liệu học tập UCI (kho lưu trữ tài liệu học Máy từ Khoa Thông tin và Khoa học Máy tính, Đại học California). Dữ liệu được thu thập lấy từ Viện Quốc gia về bệnh tiểu đường và các bệnh tiêu hóa và thận. Tất cả những bệnh nhân trong đây đều là những người phụ nữ ở Pima Ấn Độ từ 21 tuổi. Mục tiêu của bộ dữ liệu là chuẩn đoán bệnh nhân có mắc bệnh tiểu đường hay không dựa trên các phép đo chuẩn đoán nhất định trong tập dữ liệu.

Tập dữ liệu có một biến dự đoán mục tiêu nhận 2 giá trị là "0 và 1", trong đó "0" là không mắc bệnh, "1" là mắc bệnh. Có 268 (34.9%) trường hợp là "1" và 500 (65,1%) trường hợp là "0". Có 8 phát hiện lâm sàng (8 features): Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age. Thống kê ngắn gọn được thể hiện trong bảng sau:

Thuộc tính	Min,Max	Trung bình	Độ lệch chuẩn	Mô tả
Pregnancies	0/17	3.8	3.4	Số lần mang thai
Glucose	0/199	120.9	32	Nồng độ đường trong máu trong 2 giờ.
BloodPressure	0/122	69.1	19.4	Huyết áp tâm trương(mm Hg)
SkinThickness	0/99	20.5	16.0	Độ dày của da (mm)
Insulin	0/846	79.8	115.2	Insulin trong máu trong 2h (mu U/ml)
BMI	0/67.1	32.0	7.9	Chỉ số khối cơ thể
DiabetesPedigreeFunction	0.078/2.42	0.5	0.3	Phả hệ bệnh tiểu đường
Age	21/81	33.2	11.8	Tuổi

Bảng 3.1: Thống kê vắn tắt các thuộc tính của dữ liệu

3.2 Đưa ra bài toán

Với tập dữ liệu này bài toán đặt ra là dựa vào những phát hiện lâm sàng có thể dự đoán được khả năng mắc bệnh tiểu đường của bệnh nhân.
Bài toán đặt ra gồm một số yêu cầu sau:

+ Do dữ liệu là thô, thiếu nhiều giá trị, yêu cầu về tiền xử lí, tối ưu hóa dữ liệu.
+ Phân tích, lựa chọn các feature phù hợp, giúp tối ưu mô hình.
+ Xây dựng hệ thống ANFIS dự đoán khả năng mắc bệnh.
+ Xây dựng một số bộ phân loại ML thông dụng nhằm đánh giá so sánh.
+ So sánh với một số nghiên cứu đã được công bố trước đó.

3.3 Tiền xử lí dữ liệu

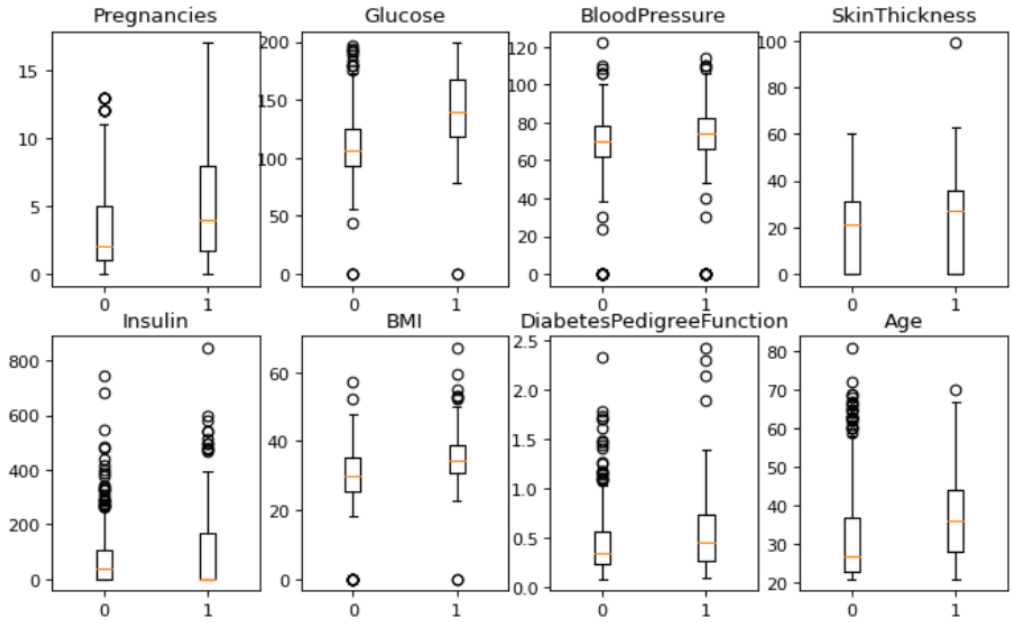
Trong các mô hình máy học, việc dữ liệu bị khuyết ảnh hưởng khá lớn đến sự thành công của mô hình. Xử lí dữ liệu khuyết là một công việc khá khó khăn, việc điền bổ sung, hay xóa đều phải quan tâm đến các mối liên hệ với các dữ liệu đó. Câu hỏi đặt ra là: Liệu xóa các dữ liệu khuyết đó có tối ưu, hay việc điền giá trị khuyết đó như thế nào cho phù hợp?

Với tập dữ liệu sử dụng trong nghiên cứu này, dữ liệu ban đầu có những tập hồ sơ chứa giá trị 0, tuy nhiên chúng ta chưa thể khẳng định được đó là giá trị thiếu hay là giá trị thực của nó. Trước hết, ta có một bảng thống kê ngắn gọn như sau:

Thuộc tính	**Số ô có giá trị 0**	**Phán đoán**
Pregnancies	111	Có thể là giá trị thực, những người phụ nữ chưa mang thai bao giờ.
Glucose	5	Giá trị thiếu
BloodPressure	35	Giá trị thiếu
SkinThickness	227	Giá trị thiếu
Insulin	374	Giá trị thiếu
BMI	11	Giá trị thiếu
DiabetesPedigreeFunction	0	
Age	0	

Bảng 3.2: Thống kê giá trị thiếu trong các thuộc tính

Để có một cái nhìn trực quan hơn về độ dàn trải dữ liệu, ta có biểu đồ boxplot của các thuộc tính theo thuộc tính phân loại như sau:



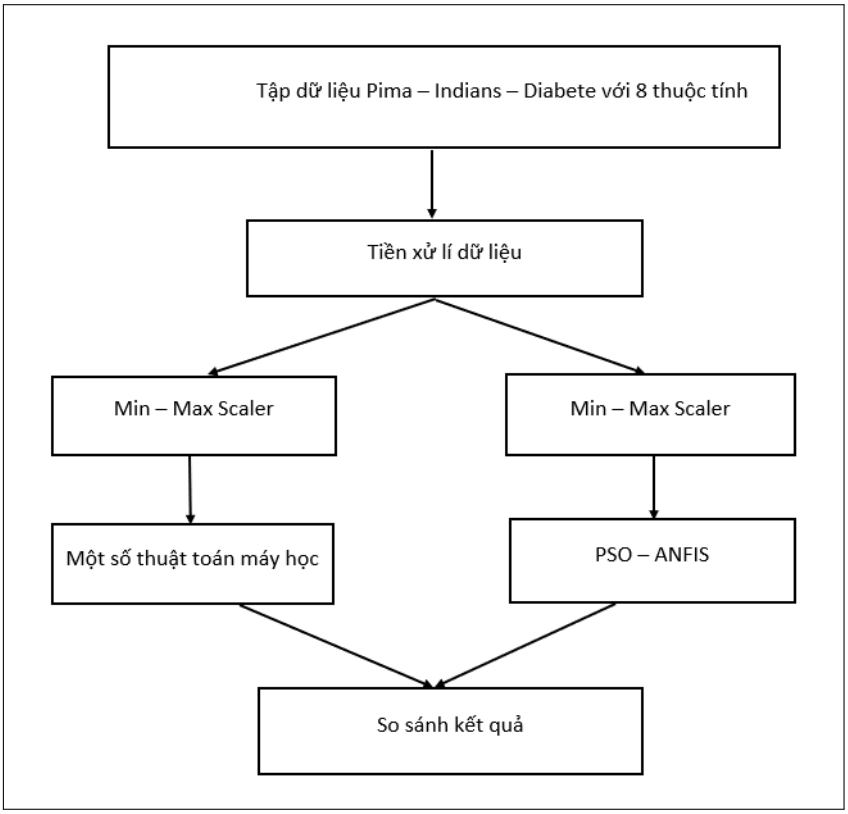
Hình 3.1: Biểu đồ phân phối dữ liệu theo biến dự báo của từng feature

Từ đó ta rút ra một số nhận xét sau:

- Xóa bỏ những thuộc tính Insulin, SkinThickness do thiếu nhiều dữ liệu (gần 50%).
- Các thuộc tính BMI, glucozo, BloodPressure các ô dữ liệu thiếu sẽ được điền bổ sung bằng giá trị trung bình.
- Với các thuật toán ML tiến hành chuẩn hóa Min_Max dữ liệu nhằm đạt hiệu quả cao cho thuật toán.
- Với các mô hình ANFIS thì dữ liệu sẽ được chuẩn hóa Min_Max trước khi đưa vào mô hình, để giảm tải khối lượng tính toán với mong muốn đạt được hiệu quả tính toán tốt nhất.

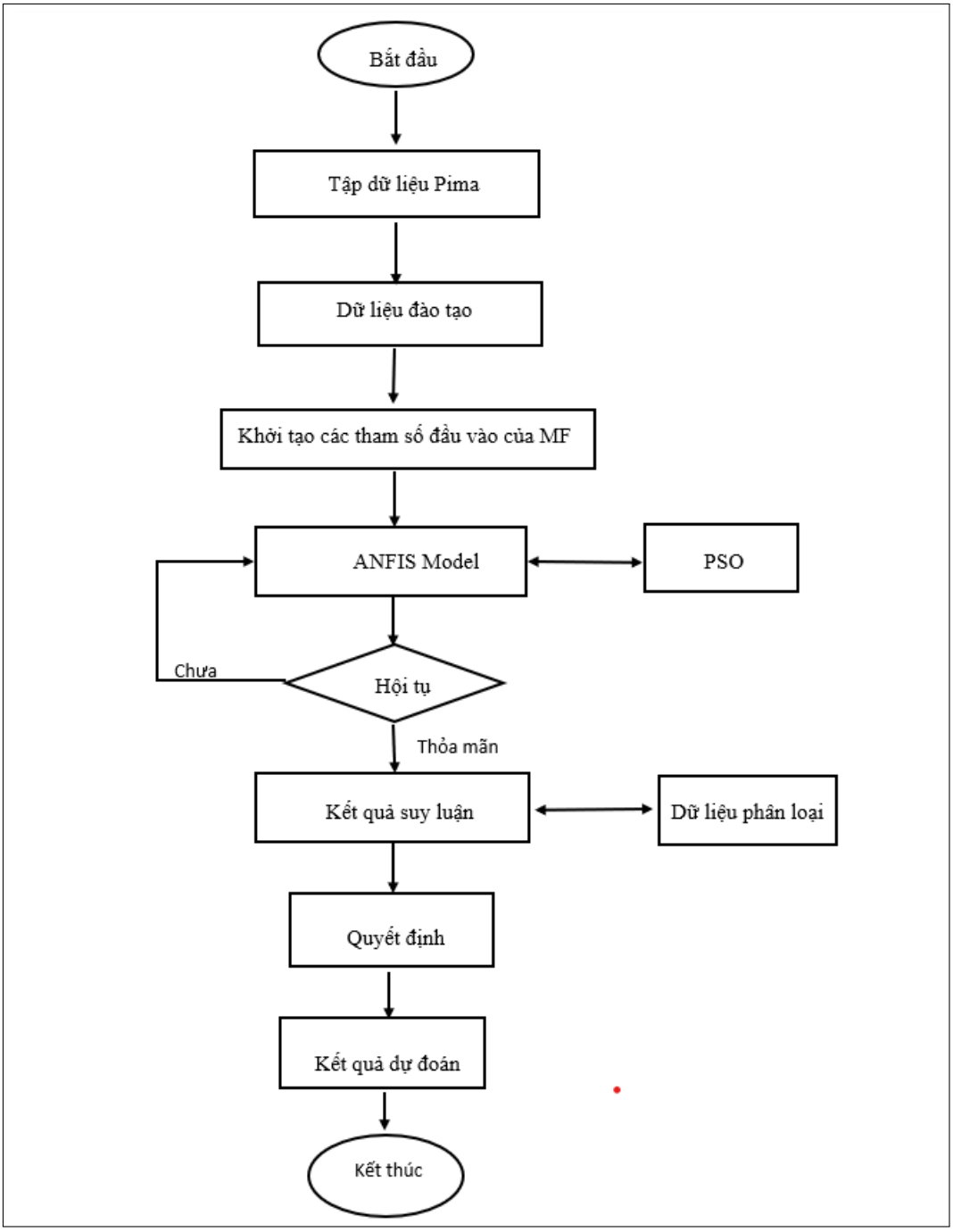
3.4 Phân tích cấu trúc chương trình

Trong thực nghiệm của mình, mô hình lai PSO-ANFIS được sử dụng trong mô hình phân loại. Trong đó thuật toán PSO được sử dụng để tối ưu các tham số cho mô hình ANFIS với mong muốn nâng cao tối ưu các hệ số hàm MF và cải thiện độ chính xác phân loại. Kết quả của thực nghiệm sẽ được so sánh với kết quả của các phương pháp truyền thống.



Hình 3.2: Sơ đồ khối của hệ thống đề xuất

Mô hình PSO-ANFIS: Sơ đồ khối phương pháp:



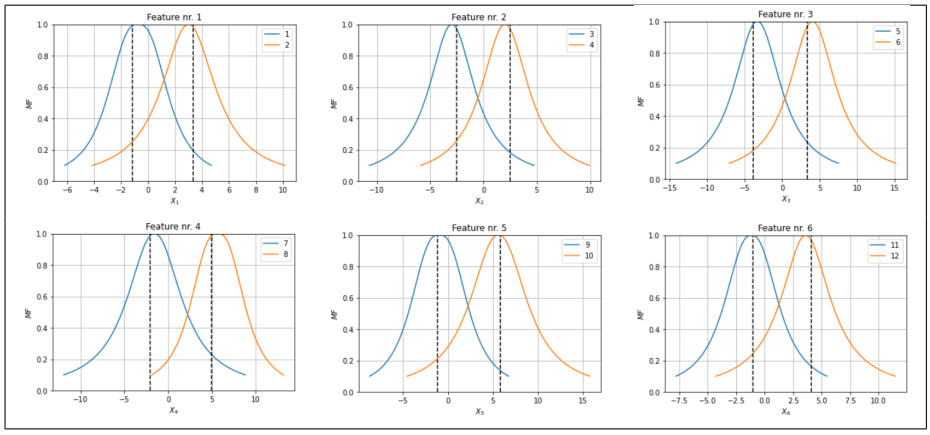
Hình 3.3: Sơ đồ thuật toán PSO-ANFIS

Sử dụng tập dữ liệu sau khi được tiền xử lí, gồm 6 thuộc tính.

- Dữ liệu được chia theo tỉ lệ 0.75 tức 75% dữ liệu dùng để train, 25% dữ liệu dùng để test.

- Số hàm MF sử dụng tương ứng cho mỗi feature là (2,2,2,2,2,2).
- Hàm thành viên sử dụng cho các tập mờ là hàm gBell, với μ, c, s là khởi tạo ngẫu nhiên
- Tổng số luật mờ sử dụng trong mô hình là 64.
- Thuật toán học, tối ưu là : PSO.
- Số epochs: 500

Hình dạng hàm các MF của mỗi feature.



Bảng 3.3: Hình dạng hàm MF của mỗi feature

3.5 Độ đo đánh giá thực nghiệm

Độ đo lỗi

Với một tập hợp các trọng số khởi tạo ngẫu nhiên, các kết quả đầu ra của mạng có thể khác so với các phân loại mong muốn. Khi mạng được đào tạo, trọng số của hệ thống liên tục được điều chỉnh để giảm sự khác nhau giữa đầu ra của hệ thống và kết quả mong muốn. Sự khác biệt này được gọi là sai số và thường đo bằng nhiều cách khác nhau. Một phép đo phổ biến là MSE. MSE là giá trị trung bình của các bình phương sai số giữa đầu ra hệ thống và đầu ra mong muốn.

Độ chính xác của phân loại

Ma trận nhầm lẫn (confusion matrix) chứa thông tin về phân loại thực tế và dự đoán của hệ thống. Đây là độ đo phổ biến phổ biến hay được sử dụng nhất trong các bài toán phân loại.

Thực tế	Dự đoán	
	Dương tính	Âm tính
Dương tính	TP	FP
Âm tính	FN	TN

Bảng 3.4: Ma trận nhầm lẫn (confusion matrix)

Trong đó:

- Condition positive (P): Tổng số ca mắc bệnh tiểu đường (dương tính) thực tế.
- Condition Negative (N): Tổng số ca không mắc bệnh tiểu đường (âm tính) thực tế.
- True positive (TP): Số các ca dự đoán dương tính đúng hay dương tính thật.
- True negative (TN): Số các ca dự đoán âm tính đúng hay âm tính thật.
- False positive (FP): Số các ca dự đoán dương tính sai hay dương tính giả.
- False negative (FN): Số các ca dự đoán âm tính sai hay âm tính giả.

Từ đó, ta có các độ đo đánh giá:

$$\textbf{Accuracy(T)} = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + FP + TN + FN}(\%) \tag{3.1}$$

$$\textbf{Precision} = \frac{TP}{TP + FP}(\%) \tag{3.2}$$

$$\textbf{Recall} = \frac{TP}{TP + FN}(\%) \tag{3.3}$$

$$\textbf{F1 Score} = \frac{2precision \times recall}{precision + recall}(\%) \tag{3.4}$$

3.6 Kết quả thực nghiệm

Sau khi tiến thành train và test trên bộ dữ liệu với các thuật toán ML, và PSO-ANFIS, kết quả thu được được trên ập train là 77,6%, kết quả thu được trên tập test là 79.7%. Cụ thể, kết quả được thể hiện dưới dạng ma trận nhầm lẫn như sau:

Thực tế	Dự đoán	
	Dương tính	Âm tính
Dương tính	129	81
Âm tính	48	318

Tập train

Thực tế	Dự đoán	
	Dương tính	Âm tính
Dương tính	41	17
Âm tính	22	112

Tập test

Bảng 3.5: Ma trẫn nhầm lẫn

Với các độ đo đánh giá cho việc chuẩn đoán 1 người có mắc bệnh tiểu đường như sau:

Độ đo	Tập train	Tập test
Accuracy	0.77	0.79
Precision	0.72	0.65
Recall	0.61	0.71
F-score	0.67	0.68

Bảng 3.6: Kết quả đánh giá mô hình

So sánh kết quả với một số phương pháp đã được nghiên cứu và công bố trước đó[18]:

Phương pháp	Độ chính xác(%)	Nguồn tham khảo
PSO-ANFIS	79.68	Nghiên cứu này
Logistic Regression	78.64	Nghiên cứu này
SVM	77.60	Nghiên cứu này
KNN(k =22)	78.64	Nghiên cứu này
Naive_bayes	77.60	Nghiên cứu này
Decesion Tree Classifier	70.31	Nghiên cứu này
Random Forest Classifier	76.56	Nghiên cứu này
Adaboost	70.3	Nghiên cứu này
Logdisc	77.7	Statlog
SMART	76.8	Statlog
MLP+BP	76.4	Ster & Dobnikar
MLP+BP)	76.47	Ster & Dobnikar
LVQ	75.8	Ster & Dobnikar
RBF	75.7	Statlog
kNN, k=22, Manh	75.5	Karol Grudziński
C4.5 DT	73.0	Stalog
CART	72.8	Ster & Dobnikar
Kohonen	72.7	Statlog
ID3	71.7±6.6	Zarndt
OCN2	65.1±1.1	Zarndt
QDA	59.5	Ster, Dobnikar

Bảng 3.7: Độ chính xác phân loại của mô hình ANFIS so với một số mô hình khác

Nhận xét:

- Qua một số bước tiền xử lí dữ liệu, các thuật toán ML được sử dụng trong nghiên cứu này đã mang lại kết quả khả quan hơn so với những phương pháp đã được đề xuất và công bố trước đó.
- Mô hình ANFIS với độ chính xác 79% tuy nhiên giá trị precision và recall chuẩn đoán một người có bị mắc bệnh ở mức trung bình. Giá trị này không quá tốt. Tuy nhiên, với thực nghiệm này, tiềm năng mà mô hình ANFIS mang lại trong dự đoán lĩnh vực y học dự kiến sẽ đạt hiệu quả cao hơn.

KẾT LUẬN

Với những cải tiến trong hệ thống chuyên gia và công cụ ML, tác động của những đổi mới này ngày càng được ứng dụng phổ biến trong mọi lĩnh vực và y tế là một trong số đó. Việc ra quyết định trong lĩnh vực y tế đôi khi có thể là một rắc tối lớn. Vì vậy, các hệ thống phân loại sẽ hỗ trợ quá trình ra quyết định nhằm rút ngắn thời gian và thủ tục chi tiết.

Hệ thống ANFIS với sự kết hợp của lí thuyết logic mờ và mạng nơron nhân tạo hứa hẹn sẽ mang lại những hệ thống phân loại hiệu quả hỗ trợ các quyết định y tế. Mục tiêu chung của khoa học là sự cải tiến và không ngừng đi lên, và hệ thống ANFIS được mong chờ sẽ ngày càng được cải tiến, hay đạt được kết quả ngày càng tốt hơn khi khám phá thêm dữ liệu và cải tiến thuật toán ngày càng tối ưu.

Qua quá trình nghiên cứu đề tài, đã đạt được một số kết quả nhất định như sau:

- Tổng hợp, nghiên cứu về mạng mơ ron, lí thuyết hệ thống logic mờ.
- Nghiên cứu thuật toán ANFIS và các ứng dụng của nó.
- Xây dựng bộ phân loại ANFIS cho vấn đề dự đoán bệnh tiểu đường và cải thiện đưa ra kết quả của một số thuật toán ML trên tập dữ liệu.

Kết quả thu được từ nghiên cứu và cải thiện các thuật toán có cải thiện nhưng chưa thật sự tối ưu. Xa hơn, hệ thống ANFIS có thể cải thiện, tối ưu hơn bằng các thuật toán tối ưu khác nhau, hay việc lựa chọn tối ưu các hàm MF vẫn chưa được nghiên cứu trong đồ án này. Từ đó, đề tài sẽ là nền tảng, hứa hẹn sẽ mang lại kết quả đáng mong đợi cho các nghiên cứu sâu hơn không chỉ với bệnh tiểu đường, mà trong rất nhiều lĩnh vực y học. ANFIS hứa hẹn sẽ mang lại một hệ thống kết hợp với các phần mềm trong chuẩn đoán, ra quyết định y tế. Lợi ích của hệ thống là giúp bác sĩ đưa ra kết quả cuối cùng mà không hề e ngại.

Tài Liệu Tham Khảo

[1] "International Diabetes Federation," *IDF Diabetes Atlas, 10th edn*, 2021. [Online]. Available: `https://diabetesatlas.org/`.

[2] G. Tornese, R. Schiaffini, E. Mozzillo, R. Franceschi, A. P. Frongia, and A. Scaramuzza, "The effect of the covid-19 pandemic on telemedicine in pediatric diabetes centers in italy: Results from a longitudinal survey," *Diabetes Research and Clinical Practice*, vol. 179, p. 109 030, 2021, ISSN: 0168-8227.

[3] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[4] X. Li, Z. Hu, and X. Huang, "Combine relu with tanh," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, vol. 1, 2020, pp. 51–55.

[5] A. K. Dubey and V. Jain, "Comparative study of convolution neural network's relu and leaky-relu activation functions," in *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*, Springer, 2019, pp. 873–880.

[6] J.-S. Jang, "Anfis: Adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[7] T. Takagi and M. Sugeno, "Derivation of fuzzy control rules from human operator's control actions," *IFAC Proceedings Volumes*, vol. 16, no. 13, pp. 55–60, 1983, ISSN: 1474-6670.

[8] R. Rojas, "The backpropagation algorithm," in *Neural networks*, Springer, 1996, pp. 149–182.

[9] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proceedings of the IEEE international conference on neural networks*, Citeseer, vol. 4, 1995, pp. 1942–1948.

[10] S. Alavandar and M. J. Nigam, "Inverse kinematics solution of 3dof planar robot using anfis," *Int. J. of Computers, Communications & Control*, vol. 3, pp. 150–155, 2008.

[11] T. P. Tho, N. T. Thinh, N. T. Tuan, and M. N. T. Nhan, "Solving inverse kinematics of delta robot using anfis," in *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, IEEE, 2015, pp. 790–795.

[12] S. R. Khuntia and S. Panda, "Simulation study for automatic generation control of a multi-area power system by anfis approach," *Applied soft computing*, vol. 12, no. 1, pp. 333–341, 2012.

[13] C. Kalaiselvi and G. Nasira, "A new approach for diagnosis of diabetes and prediction of cancer using anfis," in *2014 World Congress on Computing and Communication Technologies*, IEEE, 2014, pp. 188–190.

[14] M. Sharma and S. Mukharjee, "Brain tumor segmentation using genetic algorithm and artificial neural network fuzzy inference system (anfis)," in *Advances in computing and information technology*, Springer, 2013, pp. 329–339.

[15] B. Sarıca, E. Eğrioğlu, and B. Aşıkgil, "A new hybrid method for time series forecasting: Ar–anfis," *Neural Computing and Applications*, vol. 29, no. 3, pp. 749–760, 2018.

[16] L.-Y. Wei, "A hybrid anfis model based on empirical mode decomposition for stock time series forecasting," *Applied Soft Computing*, vol. 42, pp. 368–376, 2016.

[17] M. Tektaş, "Weather forecasting using anfis and arima models," *Environmental Research, Engineering and Management*, vol. 51, no. 1, pp. 5–10, 2010.

[18] "Datasets used for classification: Comparison of results." [Online]. Available: `http://fizyka.umk.pl/kis-old/projects/datasets.html#Diabetes`.

[19] D. LES CRITÈRES BIOLOGIQUES and D. SUCRÉ, "Définition et classification du diabète," *Médecine Nucléaire-Imagerie fonctionnelle et métabolique*, vol. 25, no. 2, p. 91, 2001.

[20] S. Chen, S. Zhang, J. Shang, B. Chen, and N. Zheng, "Brain-inspired cognitive model with attention for self-driving cars," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 1, pp. 13–25, 2017.

[21] S. R. Granter, A. H. Beck, and D. J. Papke Jr, "Alphago, deep learning, and the future of the human microscopist," *Archives of pathology & laboratory medicine*, vol. 141, no. 5, pp. 619–621, 2017.

Chỉ mục