**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC SƯ PHẠM ĐÀ NẴNG**

**KHOA TIN HỌC**

**\*\*\*\*\***

**BÁO CÁO**

**TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI: ANN - ARTIFICIAL NEURAL NETWORK**

GVHD:TS. Nguyễn Thị Ngọc Anh

SVTH: Lê Thành Tiến

Lớp: 17CNTT3

MSV: 3120217181

**Đà Nẵng, tháng 7/2020**

**NHẬN XÉT CỦA GIẢNG VIÊN**

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

Đà Nẵng, ngày ...... tháng ...... năm 2020

**Giảng viên**

**MỤC LỤC**

NHẬN XÉT CỦA GIẢNG VIÊN................................................2

MỤC LỤC....................................................................................3

NỘI DUNG..................................................................................4

1. MACHINE LEARNING.......................................................4
2. TRÍ TUỆ NHÂN TẠO (Artifical Intelligent)........................5
3. CLASSIFICATION...............................................................6
4. KHÁI NIỆM.......................................................................................6
5. QUY TRÌNH TỔNG QUAN CHO VIỆC PHÂN LỚP.....................6
6. CÁC THUẬT NGỮ LIÊN QUAN.....................................................8
7. THUẬT TOÁN ANN (ARTIFICIAL NEURAL NETWORKS).................................................................10
8. KHÁI NIỆM.....................................................................................10
9. CẤU TRÚC TỔNG QUAN MÔ HÌNH ANN.................................10
10. CÁC THÀNH PHẦN CỦA ANN....................................................11
11. MÔ HÌNH PERCEPTRON(SINGLE-LAYER NEURAL NETWORK).....................................................................................12
12. MULTILAYER PERCEPTRON......................................................15
13. APPLICATION: APPLY ANN ON IRIS DATASET FOR CLASSIFICATION TASK....................................17
14. DATA DESCRIPTION (MÔ TẢ DỮ LIỆU).................................. 17
    1. CÁCH PHÂN CHIA DỮ LIỆU TRAIN/TEST................................ 17
    2. FEATURE SELECTION...................................................................17
15. ỨNG DỤNG ANN CHO BÀI TOÁN PHÂN LỚP TẬP DỮ LIỆU IRIS...................................................................................................25
    1. SỬ DỤNG WEKA.............................................................................25
    2. SỬ DỤNG C++..................................................................................29

TÀI LIỆU THAM KHẢO...........................................................38

**NỘI DUNG**

**I. MACHINE LEARNING:**

Machine learing là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể.

Cụ thể hơn, machine learning đề cập tới bất kỳ hệ thống mà hiệu suất của máy tính khi thực hiện một nhiệm vụ sẽ trở nên tốt hơn sau khi hoàn thành nhiệm vụ đó nhiều lần. Hay nói cách khác, khả năng cơ bản nhất của machine learning là sử dụng thuật toán để phân tích những thông tin có sẵn, học hỏi từ nó rồi đưa ra quyết định hoặc dự đoán về một thứ gì đó có liên quan. Thay vì tạo ra một phần mềm với những hành động, hướng dẫn chi tiết để thực hiện một nhiệm vụ cụ thể, máy tính được “huấn luyện” bằng cách sử dụng lượng dữ liệu và các thuật toán để học cách thực hiện nhiệm vụ.

Nếu không có machine learning, AI hiện tại sẽ bị hạn chế khá nhiều bởi nó mang lại cho máy tính sức mạnh để tìm ra mọi thứ mà không được lập trình rõ ràng. Ví dụ về một loại machine learning, giả sử bạn muốn một chương trình có thể xác định được mèo trong các bức ảnh:

- Đầu tiên, bạn cung cấp cho AI một tập hợp các đặc điểm của loài mèo để máy nhận dạng, ví dụ như màu sắc lông, hình dáng cơ thể, kích thước…

- Tiếp theo, bạn cung cấp một số hình ảnh cho AI, trong đó một số hoặc tất cả các hình ảnh có thể được dán nhãn "mèo" để máy có thể chọn hiệu quả hơn các chi tiết, đặc điểm có liên quan đến mèo.

- Sau khi máy đã nhận được đủ dữ liệu cần thiết về mèo, nó phải biết cách tìm một con mèo trong một bức tranh - “Nếu trong hình ảnh có chứa các chi tiết X, Y, hoặc Z nào đó, thì 95% khả năng đó là một con mèo”.

Nhìn chung, ứng dụng của machine learning ngày nay là vô cùng phổ biến và độ hữu ích thì không phải bàn cãi nhiều nữa.

**II. TRÍ TUỆ NHÂN TẠO (Artifical Intelligent):**

Trong khoa học máy tính, trí tuệ nhân tạo hay AI (tiếng Anh: Artificial Intelligence), đôi khi được gọi là trí thông minh nhân tạo, là trí thông minh được thể hiện bằng máy móc, trái ngược với trí thông minh tự nhiên được con người thể hiện. Thông thường, thuật ngữ "trí tuệ nhân tạo" thường được sử dụng để mô tả các máy móc (hoặc máy tính) bắt chước các chức năng "nhận thức" mà con người liên kết với tâm trí con người, như "học tập" và "giải quyết vấn đề".

Trí tuệ nhân tạo có thể được phân thành ba loại hệ thống khác nhau: trí tuệ nhân tạo phân tích, lấy cảm hứng từ con người và nhân tạo. AI phân tích chỉ có các đặc điểm phù hợp với trí tuệ nhận thức; tạo ra một đại diện nhận thức về thế giới và sử dụng học tập dựa trên kinh nghiệm trong quá khứ để thông báo các quyết định trong tương lai. AI lấy cảm hứng từ con người có các yếu tố từ trí tuệ nhận thức và cảm xúc; hiểu cảm xúc của con người, ngoài các yếu tố nhận thức và xem xét chúng trong việc ra quyết định. AI nhân cách hóa cho thấy các đặc điểm của tất cả các loại năng lực (nghĩa là trí tuệ nhận thức, cảm xúc và xã hội), có khả năng tự ý thức và tự nhận thức được trong các tương tác.

Trí tuệ nhân tạo được thành lập như một môn học thuật vào năm 1956, và trong những năm sau đó đã trải qua nhiều làn sóng lạc quan, sau đó là sự thất vọng và mất kinh phí (được gọi là " mùa đông AI "), tiếp theo là cách tiếp cận mới, thành công và tài trợ mới.

Lĩnh vực này được thành lập dựa trên tuyên bố rằng trí thông minh của con người "có thể được mô tả chính xác đến mức một cỗ máy có thể được chế tạo để mô phỏng nó". Điều này làm dấy lên những tranh luận triết học về bản chất của tâm trí và đạo đức khi tạo ra những sinh vật nhân tạo có trí thông minh giống con người, đó là những vấn đề đã được thần thoại, viễn tưởng và triết học từ thời cổ đại đề cập tới. Một số người cũng coi AI là mối nguy hiểm cho nhân loại nếu tiến triển của nó không suy giảm. Những người khác tin rằng AI, không giống như các cuộc cách mạng công nghệ trước đây, sẽ tạo ra nguy cơ thất nghiệp hàng loạt

Trong thế kỷ 21, các kỹ thuật AI đã trải qua sự hồi sinh sau những tiến bộ đồng thời về sức mạnh máy tính, dữ liệu lớn và hiểu biết lý thuyết; và kỹ thuật AI đã trở thành một phần thiết yếu của ngành công nghệ, giúp giải quyết nhiều vấn đề thách thức trong học máy, công nghệ phần mềm và nghiên cứu vận hành.

**III. CLASSIFICATION:**

1. **Khái niệm:**

Phân lớp là dạng phân tích dữ liệu nhằm rút trích các mô hình mô tả các lớp dữ liệu hoặc dự đoán xu hướng dữ liệu.

Mục đích: để dự đoán những nhãn phân lớp cho các bộ dữ liệu/mẫu mới.

Tiến trình phân lớp: gồm 2 bước

* Xây dựng mô hình từ tập huấn luyện.

*+ Mỗi bộ/mẫu dữ liệu được phân vào một lớp được xác định trước.*

*+ Lớp của một bộ/mẫu dữ liệu được xác định bởi thuộc tính gán nhãn lớp.*

*+ Tập các bộ/mẫu dữ liệu huấn luyện – tập huấn luyện – được dùng để xây dựng mô hình.*

*+ Mô hình được biểu diễn bởi các luật phân lớp, các cây quyết định hoặc các công thức toán học.*

* Sử dụng mô hình – kiểm tra tính đúng đắng của mô hình và dùng nó để phân lớp dữ liệu mới.

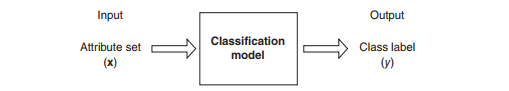
*+ Phân lớp cho những đối tượng mới hoặc chưa được phân lớp.*

*+ Đánh giá độ chính xác của mô hình.*

*Lớp biết trước của một mẫu/bộ dữ liệu đem kiểm tra được so sánh với kết quả thu được từ mô hình.*

*Tỉ lệ chính xác bằng phần trăm các mẫu/bộ dữ liệu được phân lớp đúng bởi mô hình trong số các lần kiểm tra.*

1. **Quy trình tổng quan cho việc phân lớp:**

****

Hình 1: Sơ đồ của 1 nhiệm vụ phân loại

- Đầu vào: một tập các mẫu dữ liệu huấn luyện, với một nhãn phân lớp cho mỗi mẫu dữ liệu.

- Đầu ra: mô hình (bộ phân lớp) dựa trên tập huấn luyện và những nhãn phân lớp.

- Ý tưởng: Yêu cầu phân loại một tập dữ liệu các trường hợp. Mỗi cá thể được đặc trưng bởi tuple(x,y). Trong đó x là tập hợp thuộc tính các giá trị mô tả (có thể chứa các tuộc tính bất thuộc loại nào), y là nhãn lớp (phải được phân loại).

- Mô hình phân loại là một đại diện trừu tượng của mối quan hệ giữa tập thuộc tính và nhãn lớp. Chúng ta có thể biểu diễn nó một cách toán học như một hàm mục tiêu f dùng làm đầu vào tạo thuộc tính x và tạo ra một đầu ra tương ứng với lớp dự đoán nhãn.

- Mô hình được cho là phân loại chính xác một thể hiện (x,y) nếu f(x) = y.

- VD:

+Lọc thư rác, x (bộ thuộc tính) là tập hợp các tính năng được trích xuất từ thông điệp email tiêu đề và nội dung, y (nhãn lớp) thư rác hoặc không phải thư rác (phân loại nhị phân).

+Phân loại thiên hà, x (bộ thuộc tính) là cacs tính năng được trích xuất từ hình ảnh kính viễn vọng, y (nhãn lớp) hình eclip, xoẵn ốc, hoặc hình dạng không dều (phân loại đa lớp – lớn hơn 2 nhãn lớp).

-Một mô hình phân loại phục vụ 2 vai trò trong khai thác dữ liệu: predictive model (mô hình phân loại), descriptive model (mô hình miêu tả).

+Mô hình phân loại: để phân loại nững trường hợp chưa được gắn nhãn lớp trước đó.Một nô hình paan loại tốt là phải cung cấp các dự đoán chính xác với thời gian đáp ứng nhanh.

+Mô hình mô tả: để xác định các đặc điểm phân biệt các trường hợp từ các lớp khác nhau. Điều này rất có ích cho các ứng dụng quan trọng, vd: chuẩn đoán y tế, nơi không đủ có một mô hình đưa ra dự đoán mà không cần chứng minh là thế nào nó đạt được như vậy một quyết định.

- Nhãn phải là loại danh nghĩa điều này phân biệt phân loại từ các nhiệm vụ mô hình dự đoán như hồi quy, trong đó giá trị dự đoán thường là định lượng chứ không phải khi nào cũng là định tính.

- Không phải tất cả các thuộc tính đều có liên quan đến nhiệm vụ phân loại. Vì vậy mà phát hiện sự kết hợp tối ưu của các thuộc tính phân biệt tốt nhất các trường hợp từ các lớp khác nhau là thách thức chính trong việc xây dựng các mô hình phân loại.

1. **Các thuật ngữ liên quan:**

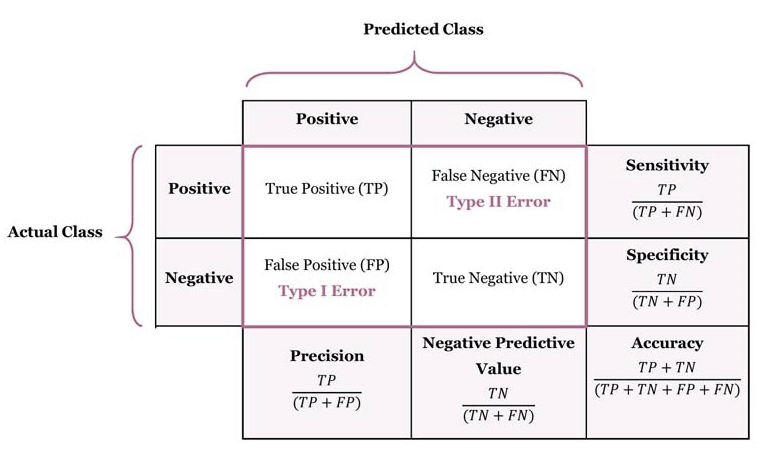
-Feature: các thuộc tính của bộ dữ liệu, giúp miêu tả các đặc trưng của bộ dữ liệu đó.

- Sample: nhãn lớp, thể hiện mục tiêu để phân loại.

- Training dataset: là tập dữ liệu chưa các giá trị cống nạp cũng như nhãn lớp cho từng trường hợp máy dùng để học và rút trích được những đặt điểm quan trọng để ghi nhớ lại.

- Testing dataset: là dữ liệu được dùng để sử dụng để đánh giá độ hiệu quả của mô hình, mức độ chính xác trong việc phân loại dữ liệu.

- Confusion matrix là một ma trận tổng quát thể hiện kết quả phân loại chính xác và kết quả phân loại sai được tạo ra bởi mô hình phân loại bằng cách so sánh với giá trị thật của biến mục tiêu của dữ liệu test.



Hình 2: Confusion matrix

-Trong đó :

+TP: tỉ lệ các trường hợp positive đúng.

+FP: tỉ lệ các trường hợp positive sai.

+TN: tỉ lệ các trường hợp negative đúng.

+FN: tỉ lệ các trường hợp negative sai.

- Feature Selection: là quá trình chọn một tập hợp con các tính năng có liên quan (biến, dự đoán) để sử dụng trong xây dựng mô hình.

- Feature Reduction (giảm kích thước): là sự chuyển đổi dữ liệu từ không gian chiều cao sang không gian chiều thấp để biểu diễn chiều thấp giữ lại một số tính chất có ý nghĩa của dữ liệu gốc, lý tưởng gần với kích thước nội tại của nó.

- Validation: được sử dụng để đánh giá hiệu suất của mô hình được xây dựng trong giai đoạn huấn luyện, hỗ trợ thử nghiệm để tinh chỉnh các tham số mô hình và chọn mô hình hoạt động tốt nhất.

- K-fold cross validation: một kĩ thuật phân chia tập dữ liệu ban đầu thành training data được sử dụng để huấn luyện mô hình và một tập dữ liệu độc lập được sử dụng để đánh giá.

- Visualization: là việc mô tả dữ liệu một cách đơn giản nhất dưới dạng các hình ảnh trực quan như bảng, biểu đồ, đồ thị...

- Learning algorithm: : là cách tiếp cân có hệ thống để học một mô hình phân loại được cung cấp một training set.

- Induction: : là quá trình sử dụng learning algorithm để xây dựng một mô hình phân loại từ một training set.

- Deduction: là quá trình áp dụng mô hình phân loại trên các trường hợp kiểm tra không nhìn thấy để dự đoán class labels.

**IV. THUẬT TOÁN ANN (ARTIFICIAL NEURAL NETWORKS):**

1. **Khái niệm:**

Mạng neuron nhân tạo (Artificial Neural Netwok – ANN): là một tập hợp các bộ xử lý rất đơn giản – neuron – và nối với nhau.

Cấu trúc và phương thức hoạt động của ANN mô phỏng tương tự mạng neuron sinh học.

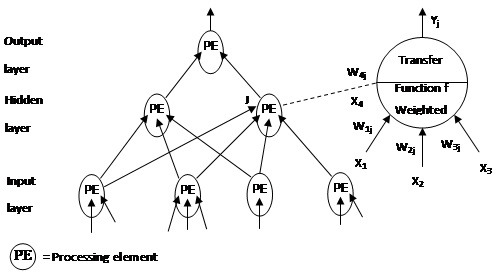
Các neuron được bởi các liên kết với các trọng số tương ứng. Các trọng số ứng với bộ nhớ dài hạn của ANN.

ANN “học” bằng cách điều chỉnh từ từ các trọng số này qua quá trình tương tác với môi trường (huấn luyện).

Một số mạng nơron tiêu biểu:

* Mạng neuron nhiều lớp (Multilayer Perceptron): được ứng dụng rộng rãi trong các bài toán phân lớp (nhận dạng), hồi quy (dự đoán)…
* Mạng Hopfield: một loại bộ nhớ nội dung có thể đánh địa chỉ (content-addressable memory), dùng để lưu trữ dữ liệu
* Ánh xạ tự tổ chức (Self Organising Maps) – Mạng Kohonen: dùng trong học bán giám sát, dùng để gom nhóm dữ liệu

1. **Cấu trúc tổng quan mô hình ANN:**



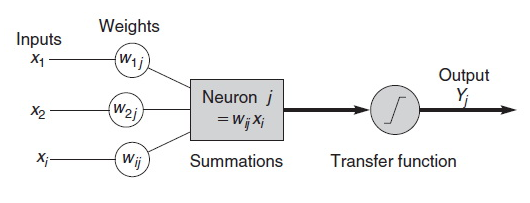
Hình 3: Mô hình tổng quát của ANN

Kiến trúc chung của một mạng nơron nhân tạo (ANN) gồm 3 thành phần đó là: Input Layer, Hidden Layer và Output Layer (Hình 3)

Trong đó, lớp ẩn (Hidden Layer) gồm các Nơron nhận dữ liệu input từ các Nơron ở lớp (Layer) trước đó và chuyển đổi các input này cho các lớp xử lý tiếp theo. Trong một ANN có thể có nhiều lớp ẩn.

Trong đó các Processing Elements (PE) của ANN gọi là Nơron, mỗi Nơron nhận các dữ liệu vào (Inputs) xử lý chúng và cho ra một kết quả (Output) duy nhất. Kết quả xử lý của một Nơron có thể làm Input cho các Nơron khác.

1. **Các thành phần của ANN:**



Hình 4: Quá trình xử lý thông tin của một ANN

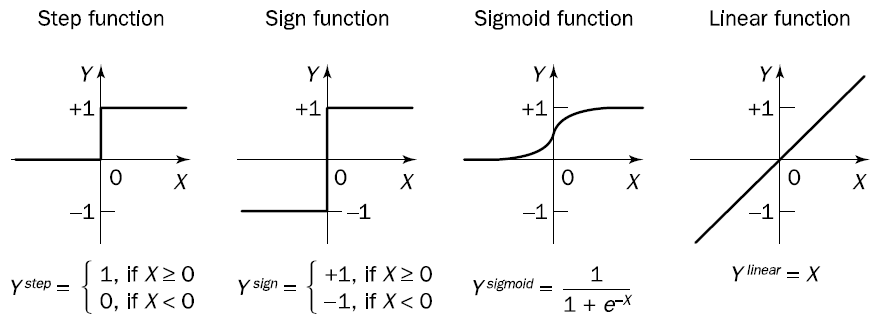
Inputs (dữ liệu vào): Mỗi Input tương ứng với 1 thuộc tính (attribute) của dữ liệu (patterns).

Output (kết quả): Kết quả của một ANN là một giải pháp cho một vấn đề.

Connection Weights (Trọng số liên kết) : Đây là thành phần rất quan trọng của một ANN, nó thể hiện mức độ quan trọng (độ mạnh) của dữ liệu đầu vào đối với quá trình xử lý thông tin (quá trình chuyển đổi dữ liệu từ Layer này sang layer khác). Quá trình học (Learning Processing) của ANN thực ra là quá trình điều chỉnh các trọng số (Weight) của các input data để có được kết quả mong muốn.

Summation Function (Hàm tổng): Tính tổng trọng số của tất cả các input được đưa vào mỗi Nơron (phần tử xử lý PE). Hàm tổng của một Nơron đối với n input được tính theo công thức sau:

Transfer Function (Hàm chuyển đổi): Hàm tổng (Summation Function) của một Nơron cho biết khả năng kích hoạt (Activation) của Nơron đó còn gọi là kích hoạt bên trong (internal activation). Các Nơron này có thể sinh ra một output hoặc không trong ANN (nói cách khác rằng có thể output của 1 Nơron có thể được chuyển đến layer tiếp trong mạng Nơron hoặc không). Mối quan hệ giữa Internal Activation và kết quả (output) được thể hiện bằng hàm chuyển đổi (Transfer Function).

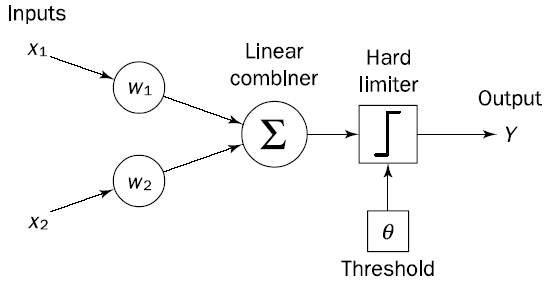


Hình 5: Bốn hàm truyền có ứng dụng thực tế

Hàm sign và step được gọi là hàm giới hạn cứng.

1. **MÔ HÌNH PERCEPTRON (SINGLE-LAYER NEURAL NETWORK):**

Perceptron là mạng neuron nhân tạo đơn giản nhất gồm một neuron với các trọng số có thể điều chỉnh được và một giới hạn cứng.



Hình 6: Mô hình Perceptron

Perceptron phân loại các input thành hai lớp A1 và A2 bị chia cách bởi một siêu phẳng cho bởi hàm phân biệt tuyến tính

Luật học perceptron có thể được viết: *wi (p+1)* = *wi (p)* + \* *xi (p)* \* *e(p)*

Với là tốc độ học

Một perceptron học bằng cách điều chỉnh siêu phẳng phân biệt theo sai số

e(p) = Yd(p) – Y(p) với Yd/Y(p) là kết xuất thực/mong muốn tại bước p.

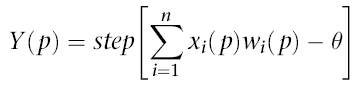
Nếu e(p) > 0 🡪 tăng *Y(p),* ngược lại giảm *Y(p).*

Nếu xi(p) > 0, tăng wi(p) sẽ làm tăng Y(p). Ngược lại nếu xi(p) < 0, tăng wi(p) sẽ làm giảm Y(p).

-Thuật toán:

Khởi tạo: (w1, w2,…, wn, t) ∈ [-0.5, 0.5]

Kích hoạt: tính kết xuất thực sự tại lần lặp p = 1



Cập nhật trọng số:



Trong đó luật học delta :



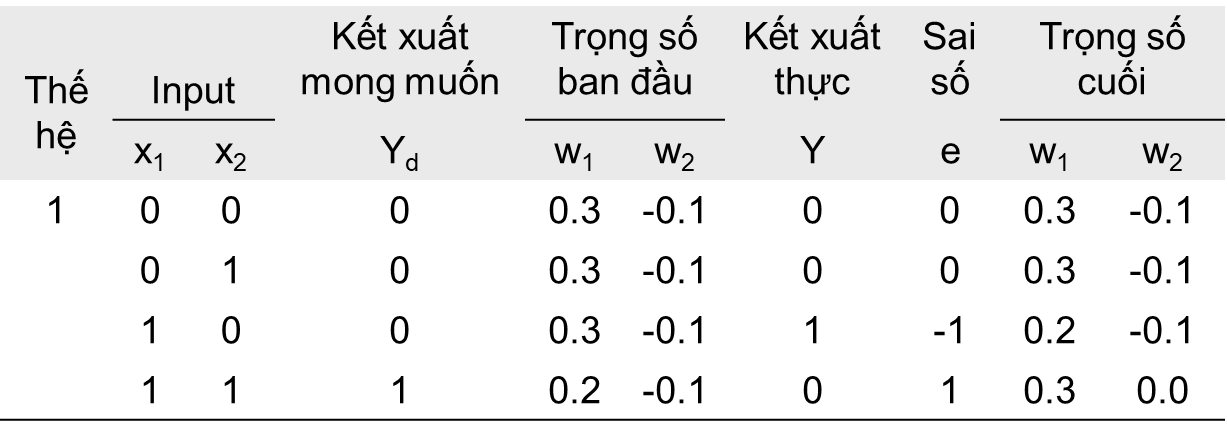
Lặp: tăng p lên 1 và lặp lại bước 2.

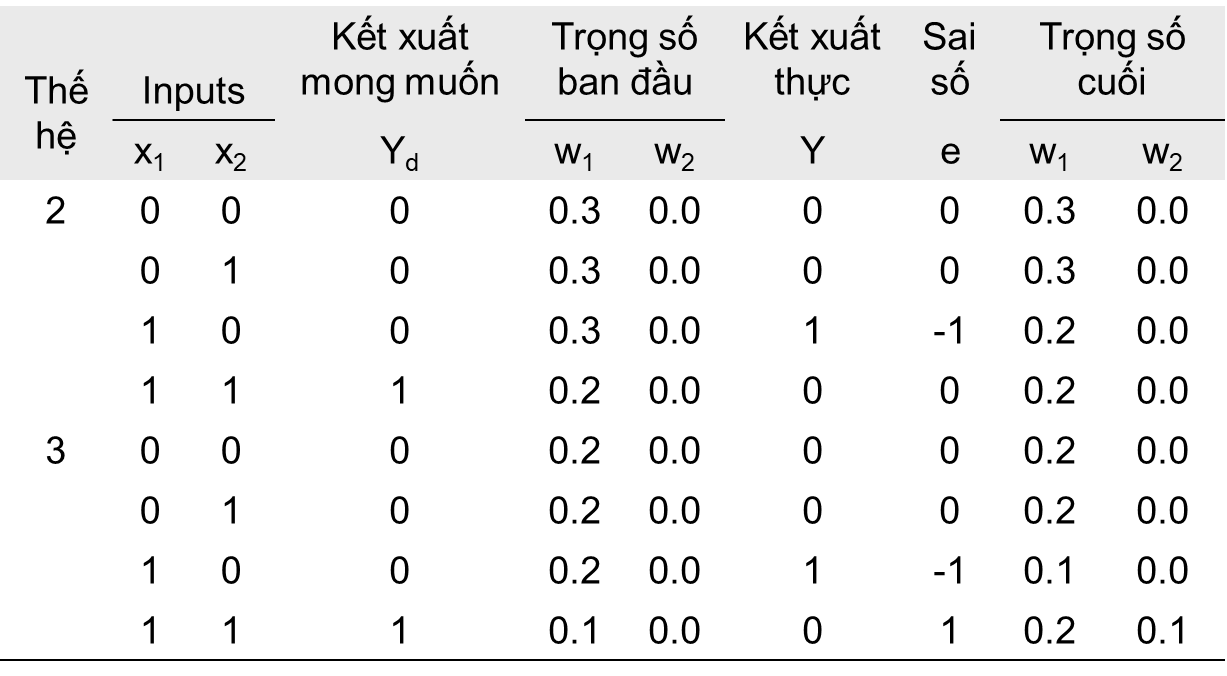
* Ví dụ:

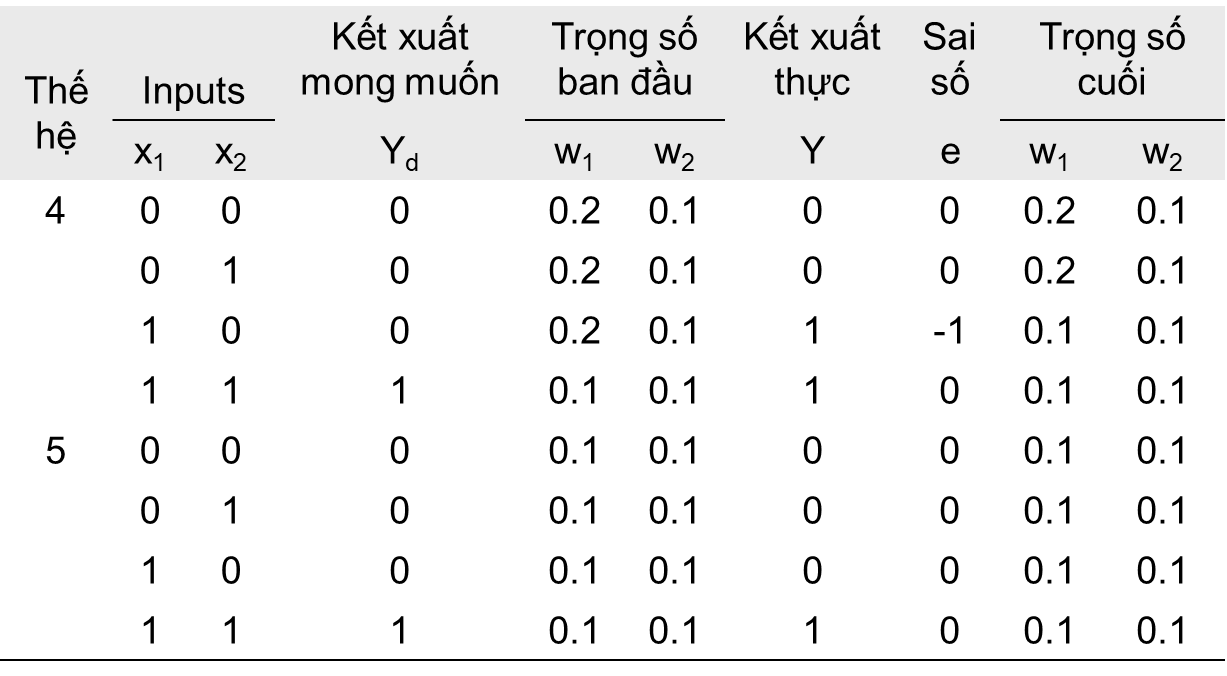
|  |
| --- |
| x1 x2 x1 AND x2 |
| 0 0 0  0 1 0  1 0 0  1 1 1 |

Khởi tạo trọng số: w1 = 0.3, w2 = -0.1,  = 0.2, tốc độ học  = 0.1

Lặp:





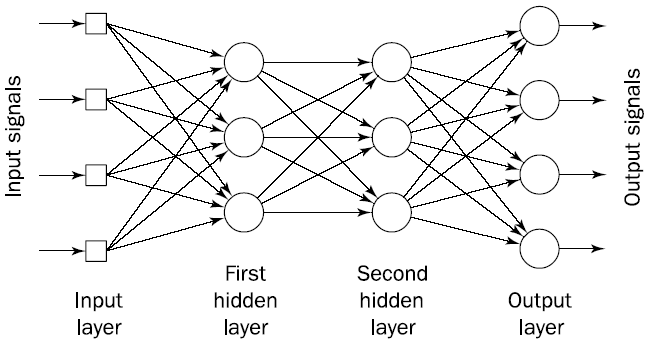


Nhược điểm của mô hình này:

* Kết xuất của perceptron là một đường thẳng (mặt phẳng) phân biệt tuyến tính: x1 \*w1 + x2 \*w2 =
* Do đó, perceptron có thể học được hàm AND, OR nhưng không được hàm XOR.

1. **Multilayer Perceptron:**

Một mạng neuron lan truyền tiến gồm một lớp nhập, ít nhất một lớp ẩn, và một lớp xuất.



Hình 7: Mô hình Multilayer perceptron

Lớp nhập: nhận các input và phân phối chúng cho tất cả neuron trong lớp ẩn

Lớp xuất: biểu diễn kết quả của toàn mạng

Lớp ẩn:

*Dò tìm các đặc trưng.*

*Các neuron trong lớp này “ẩn” các kết xuất mong muốn của chúng.*

*Mạng một lớp ẩn có thể biểu diễn bất kỳ hàm liên tục nào.*

*Mạng hai lớp ẩn có thể biểu diễn các hàm không liên tục.*

Luật học trong MLP:

* Tại nút xuất, tính sai số



với là kết xuất mong muốn ở nút k tại vòng lặp thứ p

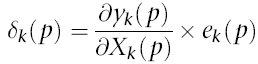
* Cập nhật sai số tương tự luật học perceptron



với

gradient sai số tại neuron p của lớp xuất

Gradient sai số = đạo hàm hàm kích hoạt nhân với sai số tại neuron xuất:



Tính đạo hàm ta được



Trong đó

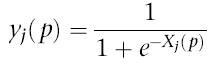


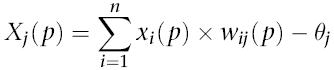
* Tại nút ẩn, ta có thể áp dụng cùng công thức với lớp xuất:



* trong đó  là gradient sai số tại neuron *j* trong lớp ẩn

với l là số neuron của nút xuất và





với n là số neuron của lớp nhập

**V. APPLICATION: APPLY ANN ON IRIS DATASET FOR CLASSIFICATION TASK:**

**1. Data Description (Mô tả dữ liệu):**

1.1 Cách phân chia dữ liệu Train/Test:

Phân chia training set/validation set là một trong những bước quan trọng nhất của một dự án machine learning. Để làm điều này chúng ta thường sử dụng các thư việc có sẵn để phân chia ngẫu nhiên 2 tập dữ liệu này dựa trên một tỉ lệ nào đó.

Chia data thành 2 phần, Training Data và Test Data. Tiến hành dùng Training Data để tạo model, dùng Test Data để dự đoán rồi xác định tỷ lệ đoán xịt, đoán trúng. Thông thường tỷ lệ khi chia data Training:Test = 70:30

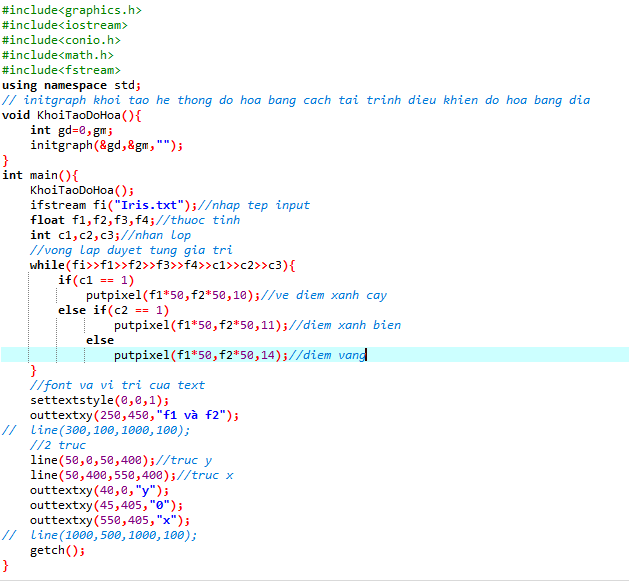
1.2. Feature selection:

Dùng 2 features:

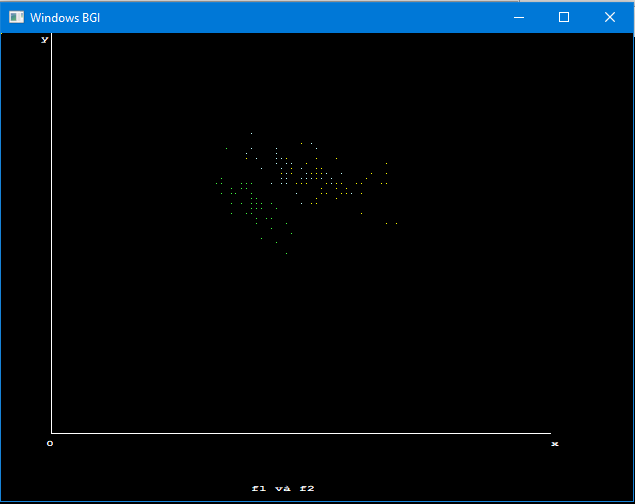
Ví dụ: sử dụng tập dữ liệu Iris:

Kết hợp từng thuộc tính với nhau lần lượt là f1 và f2, f1,và f3, f1 và f4, f2 và f3, f2 và f4, f3 và f4.

Sử dụng ngôn ngữ c++, cùng graphics để biểu diễn dữ liệu kết hợp trên:

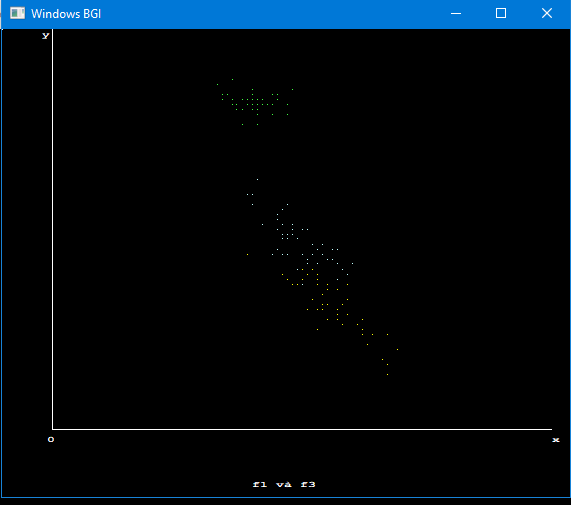


Hình 8:Chương trình f1 và f2

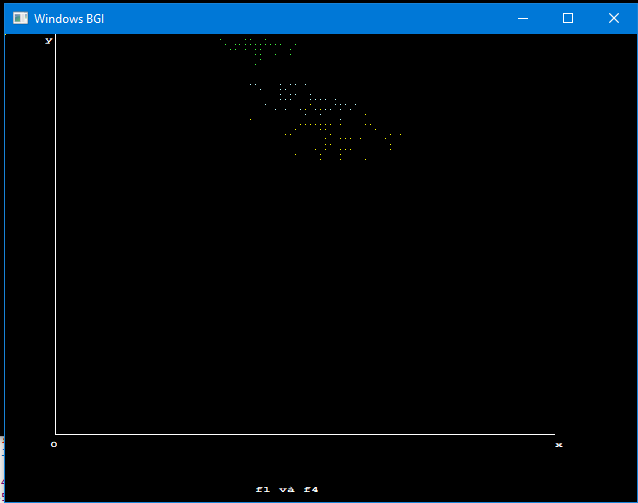


Hình 9: Hình biểu diễn dữ liệu f1 và f2

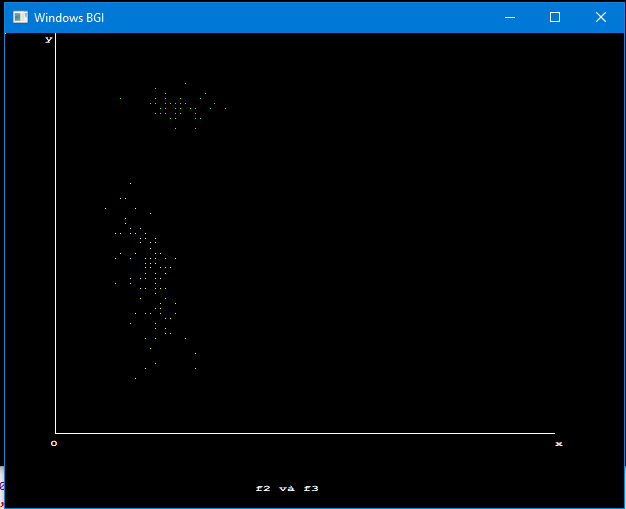
Chương trình tương tự và chúng ta có được các kết quả lần lượt như sau:



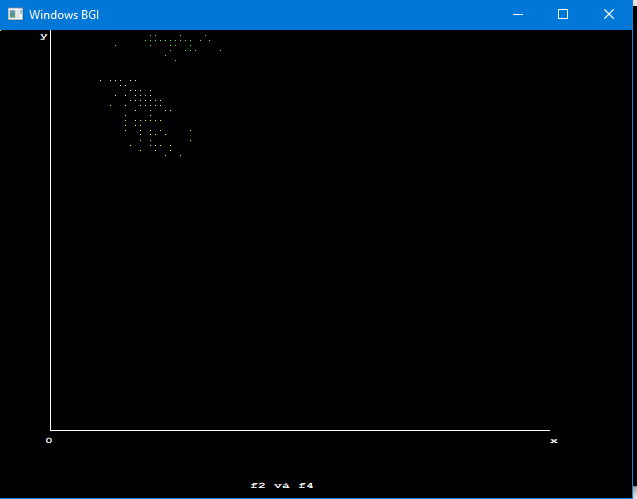
Hình 10: Hình biểu diễn dữ liệu f1 và f3



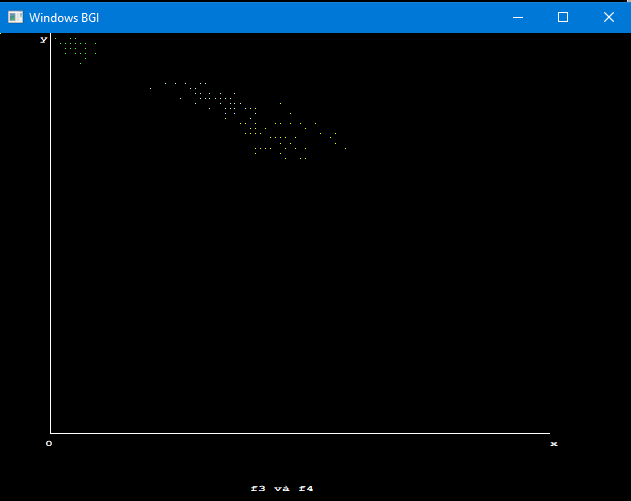
Hình 11: Hình biểu diễn dữ liệu f1 và f4



Hình 12: Hình biểu diễn dữ liệu f2 và f3



Hình 13: Hình biểu diễn dữ liệu f2 và f4



Hình 14: Hình biểu diễn dữ liệu f3 và f4

**2. Ứng dụng ANN cho bài toán phân lớp tập dữ liệu Iris:**

Hướng dẫn sử xây dựng mạng neural theo mô hình Multilayer perceptron trong ứng dụng weka và trong c++

2.1 Sử dụng Weka:

- Xử lý tẹp dữ liệu:

Khi xử lý dữ liệu trong tệp để sử dụng trong weka ta cần chú ý:

+Chỉ nhận và xử lý dữ liệu với đuôi .arff.

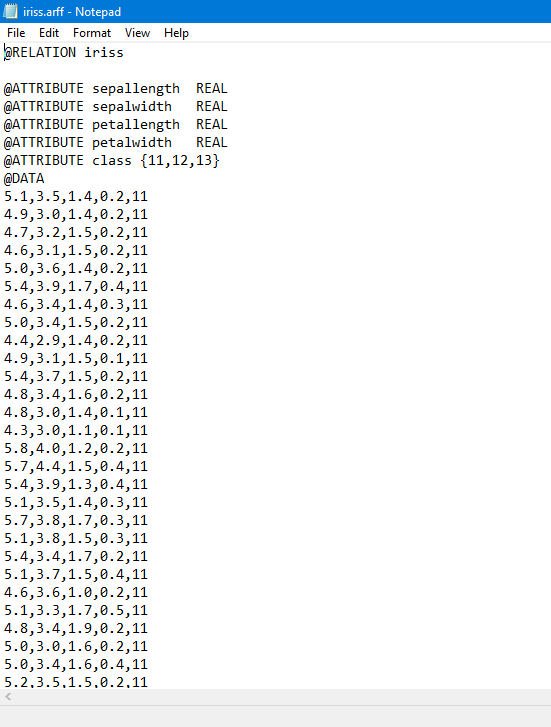
+Trong tệp cần có :

• Tên của dữ liệu

• Khai báo thuộc tính (gồm tên và kiểu dữ liệu)

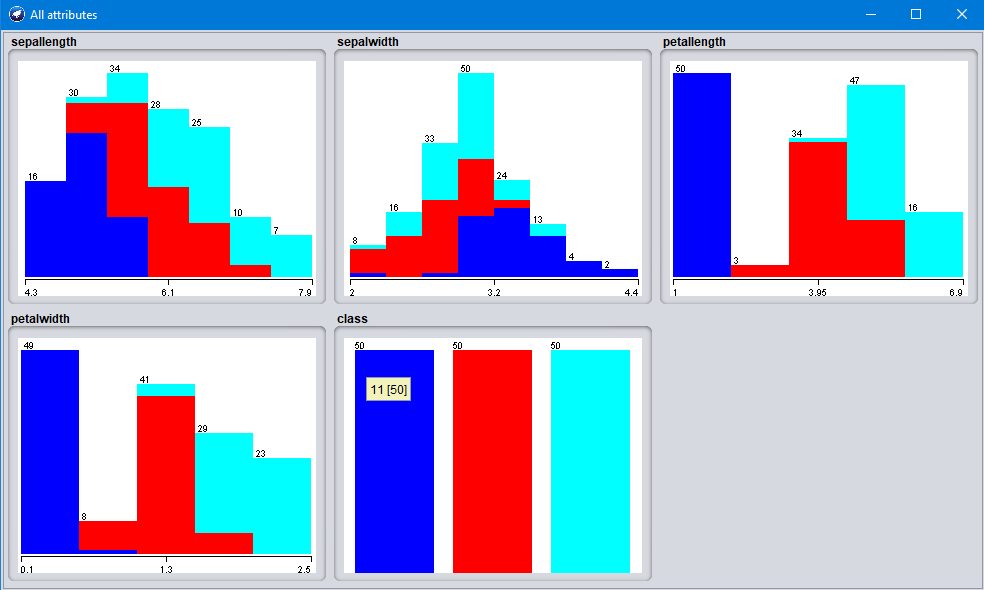
• Khai báo nhãn lớp (gồm tên và các giá trị của nhãn)

• Truyền dữ liệu



Hình 15: biểu diển tệp dữ liệu sử dụng trong weka

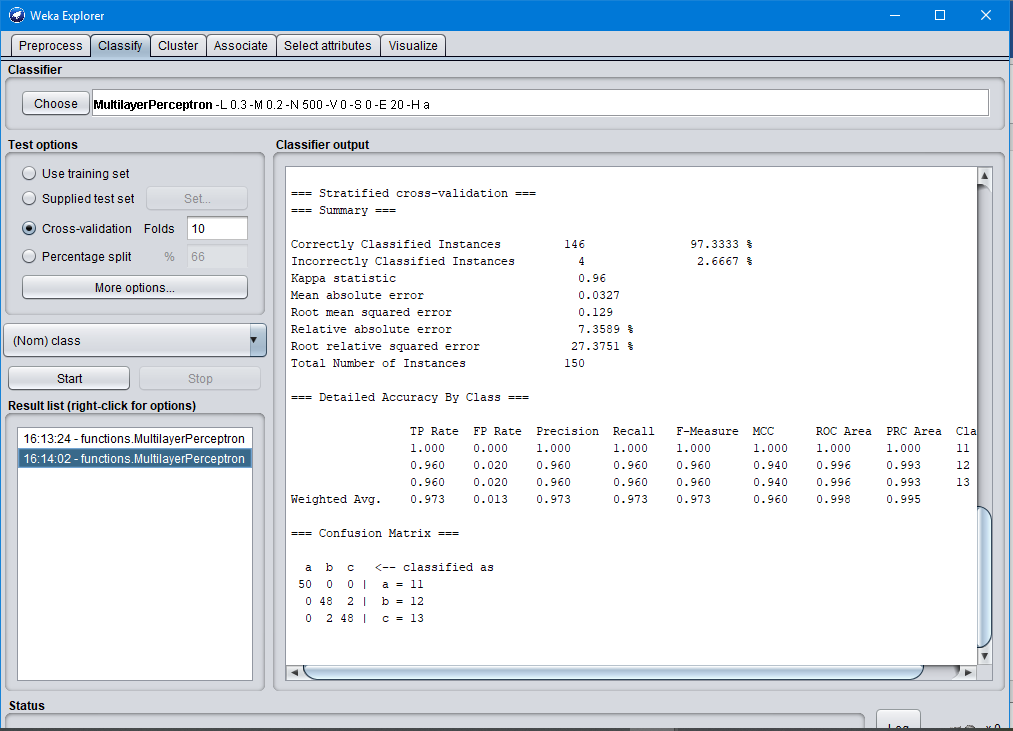
-Sau khi add tệp vào weka thì ta có tất cả các attributes:



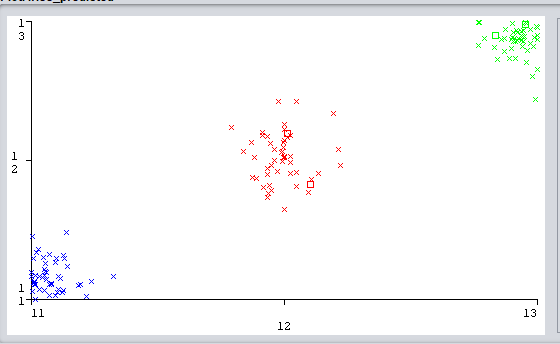
Hình 16: All attributes

-Vào classify:

+Chọn vào cross-validation: Chúng ta có kết quả là 97,33%

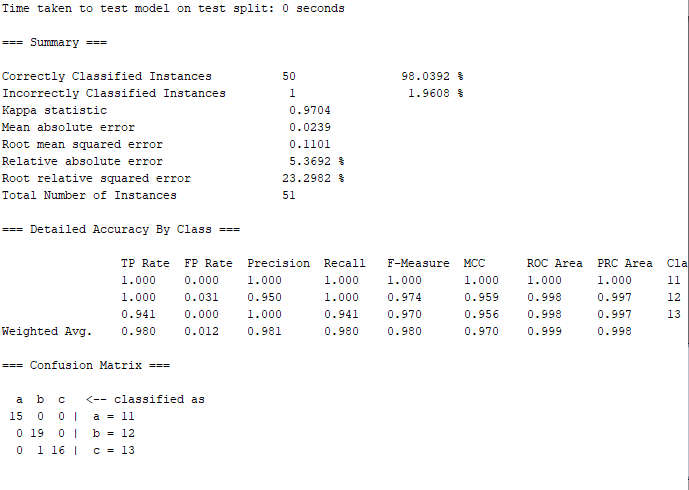


Hình 17: Cross-validation

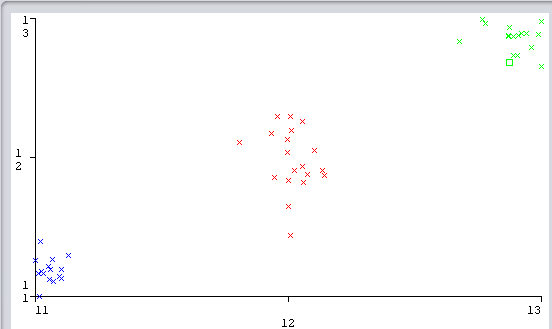


Hình 18: Sơ đồ biểu diễn

-Khi chọn Percentage split: tỉ lệ chính xác thành 98,0392%

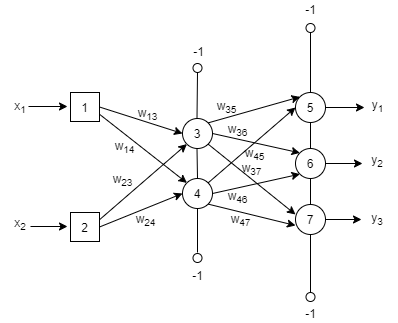


Hình 19: Percentage split



Hình 20: Sơ đồ biểu diễn

2.2 Sử dụng c++:



Hình 21: Mô hình multilayer perceptron của bài toán

Mô tả thuật toán:

- Khởi tạo: Đặt giá trị ngẫu nhiên cho các trọng số và ngưỡng của mạng

float w13 = 0.5;

float w14 = 0.9;

float w23 = 0.4;

float w24 = 1.0;

float w35 = -1.2;

float w45 = 1.1;

float w36 = 1.0;

float w46 = 1.3;

float w37 = 0.6;

float w47 = 1.0;

float theta3 = 0.8;

float theta4 = -0.1;

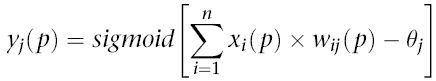
float theta5 = 0.3;

float theta6 = 0.7;

float theta7 = 0.5;

-Kích hoạt:

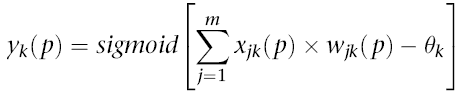
Tính kết xuất thực tại nút ẩn



float y3 = 1 / ( 1 + pow(e,-( X[j][0] \*w13+X[j][1]\*w23-theta3)));

float y4 = 1 / ( 1 + pow(e,-( X[j][0] \*w14+X[j][1]\*w24-theta4)));

Tính kết xuất thực tại nút xuất



float y5 = 1 / ( 1 + pow(e,-(y3\*w35+y4\*w45-theta5)));

float y6 = 1 / ( 1 + pow(e,-(y3\*w36+y4\*w46-theta6)));

float y7 = 1 / ( 1 + pow(e,-(y3\*w37+y4\*w47-theta7)));

Tính sai số

error5 = Y[j][0] - y5;

error6 = Y[j][1] - y6;

error7 = Y[j][2] - y7;

-Huấn luyện trọng số:

Tính gradient sai số tại các neuron



gradient5 = y5\*(1-y5)\*error5;

gradient6 = y6\*(1-y6)\*error6;

gradient7 = y7\*(1-y7)\*error7;

Tính các giá trị điều chỉnh



deltaw35 = Alpha \* y3 \* gradient5;

deltaw45 = Alpha \* y4 \* gradient5;

deltatheta5 = Alpha \* (-1) \* gradient5;

deltaw36 = Alpha \* y3 \* gradient6;

deltaw46 = Alpha \* y4 \* gradient6;

deltatheta6 = Alpha \* (-1) \* gradient6;

deltaw37 = Alpha \* y3 \* gradient7;

deltaw47 = Alpha \* y4 \* gradient7;

deltatheta7 = Alpha \* (-1) \* gradient7;

Cập nhật lại trọng số



w35 = w35 + deltaw35;

w45 = w45 + deltaw45;

theta5 = theta5 + deltatheta5;

w36 = w36 + deltaw36;

w46 = w46 + deltaw46;

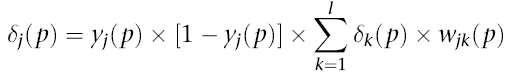
theta6 = theta6 + deltatheta6;

w37 = w37 + deltaw37;

w47 = w47 + deltaw47;

theta7 = theta7 + deltatheta7;

Tính gradien sai số và cập nhật trọng số lớp ẩn







gradient3 = y3\*(1-y3)\*(gradient5\*w35+gradient6\*w36+gradient7\*w37);

gradient4 = y4\*(1-y4)\*(gradient5\*w45+gradient6\*w46+gradient7\*w47);

deltaw13 = Alpha \* X[j][0]\*gradient3;

deltaw23 = Alpha \* X[j][1]\*gradient3;

deltatheta3 = Alpha \* (-1) \* gradient3;

deltaw14 = Alpha \* X[j][0]\*gradient4;

deltaw24 = Alpha \* X[j][1]\*gradient4;

deltatheta4 = Alpha \* (-1) \* gradient4;

w13 = w13 + deltaw13;

w23 = w23 + deltaw23;

theta3 = theta3 + deltatheta3;

w14 = w14 + deltaw14;

w24 = w24 + deltaw24;

theta4 = theta4 + deltatheta4;

-Chương trình hoàn chỉnh

#include <iostream>

#include <fstream>

#include <math.h>

#include <vector>

# define e 2.718281828

using namespace std;

ifstream fi("iris.txt");

ifstream fii("iris.txt");

ifstream fiii("iristest.txt");

class Multilayer{

  private:

    float Alpha;

    int Lo;

    float W13;

    float W14;

    float W23;

    float W24;

    float W35;

    float W45;

    float W36;

    float W46;

    float W37;

    float W47;

    float Theta3;

    float Theta4;

    float Theta5;

    float Theta6;

    float Theta7;

  public:

    Multilayer(float alpha, int lo){

      Alpha = alpha;

      Lo = lo;

    }

    void training(vector<vector<float>> X, vector<vector<float>> Y){

      //khoi tao cho trong so & nguong

      float w13 = 0.5;

      float w14 = 0.9;

      float w23 = 0.4;

      float w24 = 1.0;

      float w35 = -1.2;

      float w45 = 1.1;

      float w36 = 1.0;

      float w46 = 1.3;

      float w37 = 0.6;

      float w47 = 1.0;

      float theta3 = 0.8;

      float theta4 = -0.1;

      float theta5 = 0.3;

      float theta6 = 0.7;

      float theta7 = 0.5;

      float error5;

      float error6;

      float error7;

      float gradient3;

      float gradient4;

      float gradient5;

      float gradient6;

      float gradient7;

      float deltaw13;

      float deltaw14;

      float deltaw23;

      float deltaw24;

      float deltaw35;

      float deltaw45;

      float deltaw36;

      float deltaw46;

      float deltaw37;

      float deltaw47;

      float deltatheta3;

      float deltatheta4;

      float deltatheta5;

      float deltatheta6;

      float deltatheta7;

      for(int i=0; i<Lo; i++){

        for(int j=0; j< X.size(); j++){

          //tinh ket xuat thuc cua nut an

          float y3 = 1 / ( 1 + pow(e,-( X[j][0] \*w13+X[j][1]\*w23-theta3)));

          float y4 = 1 / ( 1 + pow(e,-( X[j][0] \*w14+X[j][1]\*w24-theta4)));

          //ket xuat tai nut xuat

          float y5 = 1 / ( 1 + pow(e,-(y3\*w35+y4\*w45-theta5)));

          float y6 = 1 / ( 1 + pow(e,-(y3\*w36+y4\*w46-theta6)));

          float y7 = 1 / ( 1 + pow(e,-(y3\*w37+y4\*w47-theta7)));

          //tai lop xuat

          //sai so

          error5 = Y[j][0] - y5;

          error6 = Y[j][1] - y6;

          error7 = Y[j][2] - y7;

          //tinh gradient sai so tai cac neuron lop xuat

          gradient5 = y5\*(1-y5)\*error5;

          gradient6 = y6\*(1-y6)\*error6;

          gradient7 = y7\*(1-y7)\*error7;

          //tinh cac gia tri dieu chinh trong so

          deltaw35 = Alpha \* y3 \* gradient5;

          deltaw45 = Alpha \* y4 \* gradient5;

          deltatheta5 = Alpha \* (-1) \* gradient5;

          deltaw36 = Alpha \* y3 \* gradient6;

          deltaw46 = Alpha \* y4 \* gradient6;

          deltatheta6 = Alpha \* (-1) \* gradient6;

          deltaw37 = Alpha \* y3 \* gradient7;

          deltaw47 = Alpha \* y4 \* gradient7;

          deltatheta7 = Alpha \* (-1) \* gradient7;

          //cap nhat trong so

          w35 = w35 + deltaw35;

          w45 = w45 + deltaw45;

          theta5 = theta5 + deltatheta5;

          w36 = w36 + deltaw36;

          w46 = w46 + deltaw46;

          theta6 = theta6 + deltatheta6;

          w37 = w37 + deltaw37;

          w47 = w47 + deltaw47;

          theta7 = theta7 + deltatheta7;

          //tuong tu, tai lop an

          gradient3 = y3\*(1-y3)\*(gradient5\*w35+gradient6\*w36+gradient7\*w37);

          gradient4 = y4\*(1-y4)\*(gradient5\*w45+gradient6\*w46+gradient7\*w47);

          deltaw13 = Alpha \* X[j][0]\*gradient3;

          deltaw23 = Alpha \* X[j][1]\*gradient3;

          deltatheta3 = Alpha \* (-1) \* gradient3;

          deltaw14 = Alpha \* X[j][0]\*gradient4;

          deltaw24 = Alpha \* X[j][1]\*gradient4;

          deltatheta4 = Alpha \* (-1) \* gradient4;

          w13 = w13 + deltaw13;

          w23 = w23 + deltaw23;

          theta3 = theta3 + deltatheta3;

          w14 = w14 + deltaw14;

          w24 = w24 + deltaw24;

          theta4 = theta4 + deltatheta4;

        }

      }

cout<<"Phan training: \n";

      cout<<" w13: "<<w13<<"\n w14: "<<w14<<"\n w53: "<<w35<<"\n w45: "<<w45<<"\n w36: "<<w36<<"\n w46: "<<w46<<"\n w37: "<<w37<<"\n w47: "<<w47<<endl;

      cout<<" theta3: "<<theta3<<"\n theta4: "<<theta4<<"\n theta5: "<<theta5<<"\n theta6: "<<theta6<<"\n theta7: "<<theta7<<endl;

      W13 =w13;

      W14 =w14;

      W23 =w23;

      W24 =w24;

      W35 =w35;

      W45 =w45;

      W36 =w36;

      W46 =w46;

      W37 =w37;

      W47 =w47;

      Theta3 = theta3;

      Theta4 = theta4;

      Theta5 = theta5;

      Theta6 = theta6;

      Theta7 = theta7;

    }

    void testing(vector<vector<float>> X){

cout<<"\nPhan testing: \n";

      for(int i = 0 ; i< X.size(); i++){

        float y3 = 1 / ( 1 + pow(e,-(X[i][0]\*W13+X[i][1]\*W23-Theta3)));

        float y4 = 1 / ( 1 + pow(e,-(X[i][0]\*W14+X[i][1]\*W24-Theta4)));

        float y5 = 1 / ( 1 + pow(e,-(y3\*W35+y4\*W45-Theta5)));

        float y6 = 1 / ( 1 + pow(e,-(y3\*W36+y4\*W46-Theta6)));

        float y7 = 1 / ( 1 + pow(e,-(y3\*W37+y4\*W47-Theta7)));

        cout<<X[i][0]<<"\t"<<X[i][1]<<endl;

        cout<<y5<<"\t"<<y6<<"\t"<<y7<<endl;

      }

    }

};

vector<vector<float>> getIrisX(){

  vector< vector<float> > X;

  float f1,f2,f3,f4;

  float c1, c2, c3;

  while(fi>>f1>>f2>>f3>>f4>>c1>>c2>>c3){

    X.push\_back(vector < float > {c1, c2, c3});

  }

  return X;

}

vector<vector<float>> getIrisY(){

  vector< vector<float> > Y;

  float f1,f2,f3,f4;

  float c1, c2, c3;

  while(fii>>f1>>f2>>f3>>f4>>c1>>c2>>c3){

    Y.push\_back(vector < float > {c1, c2, c3});

  }

  return Y;

}

vector<vector<float>> getIrisTest(){

  vector< vector<float> > T;

  float f1,f2,f3,f4;

  float c1, c2, c3;

  while(fiii>>f1>>f2>>f3>>f4>>c1>>c2>>c3){

    T.push\_back(vector < float > {f1, f2});

  }

  return T;

}

int main() {

  vector<vector<float>> X = getIrisX();

  vector<vector<float>> Y = getIrisY();

 vector<vector<float>> T = getIrisTest();

 Multilayer iris(0.8,8000);

 iris.training(X,Y);

 iris.testing(T);

  return 0;

}

-Kết quả của chương trình khi chúng ta lấy 20% dữ liệu test

|  |  |
| --- | --- |
|  |  |
|  |  |

Hình 22: Kết quả của chương trình demo

**TÀI LIỆU THAM KHẢO**

1. **Tài liệu sách:**

[1] 02-ANN\_VNese\_Lecture.

[2] 04\_classification\_textbook\_E.

[3] weka\_guide.

1. **Website:**

[1]Website.<http://nawapi.gov.vn/index.php?option=com_content&view=article&id=3238%3Agii-thiu-tng-quan-v-mng-nron-nhan-to-artificial-neural-network-ann&catid=70%3Anhim-v-chuyen-mon-ang-thc-hin&Itemid=135&lang=vi>.

[2] Website. [https://vi.wikipedia.org/wiki/ Trí\_tuệ\_nhân\_tạo](https://vi.wikipedia.org/wiki/%20Trí_tuệ_nhân_tạo).

[3] Website. <https://bigdatauni.com/vi/tin-tuc/phuong-phap-danh-gia-mo-hinh-phan-loai-classification-model-evalutation.html>.

[4] Website. <https://machinelearningcoban.com/2017/01/21/perceptron/>