

**TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN LẬP TRÌNH TÍNH TOÁN**

**Đề tài 503: Bài toán dự đoán doanh số bán  
hàng sử dụng mạng neural**

**Người hướng dẫn: PGS. TS. NGUYỄN TẤN KHÔI**

**Sinh viên thực hiện:**

**Tên sinh viên 1: Nguyễn Trí Hoài Thương**

**Tên sinh viên 2: Hoàng Thị Hồng Thắm**

**LỚP: 21T-DT2**

**NHÓM: 21Nh10**

**Đà Nẵng, 04/2020**

# MỤC LỤC

MỞ ĐẦU .....	i
1.1. Lý do chọn đề tài .....	i
1.2. Mục tiêu và nhiệm vụ chính .....	i
1.3. Đối tượng và phạm vi nghiên cứu .....	i
1.4. Phương pháp nghiên cứu .....	i
1.5. Ý nghĩa khoa học và thực tiễn của đề tài .....	i
1.6. Bố cục đồ án môn học .....	i
1. TỔNG QUAN ĐỀ TÀI .....	1
2. CƠ SỞ LÝ THUYẾT .....	1
2.1. Ý tưởng.....	1
2.2. Cơ sở lý thuyết.....	1
2.2.1. Mô hình dự báo: .....	1
2.2.2. Mạng nơ-ron nhân tạo:.....	2
3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN.....	13
3.1. Phát biểu bài toán .....	13
3.2. Thiết kế mạng nơron .....	14
3.2.1. Số lớp nơron.....	14
3.2.2. Cấu trúc mạng .....	14
3.2.3. Hàm tương tác đầu ra.....	14
3.3. Một số khái niệm cần biết: .....	14
3.3.1. Đầu tiên là bước lan truyền tiến: .....	15
3.3.2. Lan truyền ngược: .....	18
3.3. Cấu trúc dữ liệu .....	21
3.4. Thuật toán.....	21

---

3.4.1.	Hàm sigmoid .....	21
3.4.2.	Đạo hàm của sigmoid.....	21
3.4.3.	Hàm tính đầu ra cho các Nơ-ron .....	22
3.4.4.	Hàm tính lại trọng số weights .....	22
3.4.5.	Hàm tính tín hiệu lỗi .....	22
3.4.6.	Hàm lan truyền tín hiệu lỗi .....	22
4.	CHƯƠNG TRÌNH VÀ KẾT QUẢ .....	23
4.3.	Tổ chức chương trình .....	23
4.4.	Cài đặt chương trình .....	23
4.5.	Kết quả thực hiện.....	24
4.5.1.	Giao diện chính của chương trình .....	24
4.5.2.	Các kết quả thực thi của chương trình .....	24
4.5.3.	Nhận xét đánh giá kết quả.....	24
5.	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	25
5.3.	Kết luận .....	25
5.4.	Hướng phát triển.....	25

---

## DANH MỤC HÌNH VẼ

Hình 1. Neural Networks.....	1
Hình 2 Mô hình mạng nơron sinh học.....	2
Hình 3 Mô hình của một mạng nơron nhân tạo.....	4
Hình 4. Mô hình cấu trúc của mạng nơron nhân tạo .....	5
Hình 5. Mạng nơron truyền thẳng .....	6
Hình 6. Mạng nơron hồi quy .....	6
Hình 7. Học có giám sát. ....	7
Hình 8. Học tăng cường. ....	8
Hình 9. Học không giám sát.....	8
Hình 10. Mô hình mạng nowrowrron nhân tạo. ....	9
Hình 11. Mô hình phi tuyến thứ hai của một nơron. ....	10
Hình 12. (a): Hàm ngưỡng, (b): Hàm vùng tuyến tính, (c): Hàm sigma với tham số độ dốc a thay đổi. ....	11

## DANH MỤC BẢNG BIỂU

Bảng 3-1. Bảng dữ liệu doanh số bán hàng của Đà Nẵng Computer từ 1/1/2022 đến 25/1/2022 .....	13
-----------------------------------------------------------------------------------------------	----

## MỞ ĐẦU

### 1.1. Lý do chọn đề tài

Hiện nay công tác dự báo được ứng dụng rộng rãi ở hầu hết các lĩnh vực trong xã hội. Công tác dự báo có nhiệm vụ dự báo trước sự thay đổi của đối tượng dựa trên cơ sở nghiên cứu các quy luật của chúng. Đề tài tập trung nghiên cứu kỹ thuật sử dụng mạng nơron để đưa ra dự đoán giúp người dùng trong việc đưa ra quyết định. Trên cơ sở đó, đề tài nghiên cứu xây dựng chương trình dự báo doanh số bán hàng sử dụng mạng nơron lan truyền ngược.

### 1.2. Mục tiêu và nhiệm vụ chính

Nghiên cứu ứng dụng mạng nơron nhân tạo vào lớp bài toán dự đoán và phân loại.

Xây dựng chương trình “dự đoán doanh số bán hàng” nhằm phân tích, dự báo, hỗ trợ cho những người quan tâm.

### 1.3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu là lớp bài toán dự đoán và phân loại, sử dụng mạng nơron nhân tạo truyền thẳng huấn luyện bằng thuật toán lan truyền ngược.

Phạm vi nghiên cứu là lý thuyết ứng dụng mạng nơron nhân tạo cho bài toán dự đoán và phân loại.

### 1.4. Phương pháp nghiên cứu

Nghiên cứu lý thuyết để nắm rõ các mô hình mạng nơron, các phương pháp dự báo, lý thuyết về mạng nơron nhân tạo, mạng nơron lan truyền thẳng và thuật toán lan truyền ngược.

### 1.5. Ý nghĩa khoa học và thực tiễn của đề tài

Đề tài làm rõ khả năng ứng dụng của mạng nơron trong bài toán dự đoán và phân loại. Xây dựng thành quy trình với các bước thực hiện cụ thể cho việc giải bài toán dự đoán và phân loại bằng mạng nơron.

### 1.6. Bố cục đồ án môn học

Toàn bộ nội dung đồ án được trình bày trong 5 chương:

Chương 1: Tổng quan đề tài.

Chương 2: Cơ sở lý thuyết.

Chương 3: Tổ chức cấu trúc dữ liệu và thuật toán.

Chương 4: Chương trình và kết quả.

Chương 5: Kết luận và hướng phát triển.

## 1. TỔNG QUAN ĐỀ TÀI

Tìm hiểu và nắm vững cơ sở lý thuyết về kỹ thuật mạng Noron nhân tạo, thuật giải di truyền.

Vận dụng xây dựng chương trình “Dự báo doanh số bán hàng”. Công cụ nhằm phân tích, dự báo, hỗ trợ cho những người quan tâm và cơ quan có chức năng.

## 2. CƠ SỞ LÝ THUYẾT

### 2.1. Ý tưởng

Tìm hiểu các đặc trưng của mạng nơ-ron nhân tạo, khả năng và các nguyên tắc để ứng dụng thành công mạng nơ-ron nhân tạo trong thực tế. Nghiên cứu ứng dụng mạng nơ-ron nhân tạo vào lớp bài toán dự đoán và phân loại.

Xây dựng chương trình “dự đoán doanh số bán hàng” nhằm phân tích, dự báo, hỗ trợ cho những người quan tâm.



Hình 1. Neural Networks

### 2.2. Cơ sở lý thuyết

#### 2.2.1. Mô hình dự báo:

- **Mô hình ứng dụng mạng nơron:** Là mô hình có khả năng “học” từ các dữ liệu quá khứ, có thể cập nhật các tham số. Nếu lựa chọn được các tham số tối ưu thì

đó là mô hình xấp xỉ rất tốt đường cong dịch chuyển của đối tượng cần dự báo. Kết quả cũng có độ lệch chính xác cao.

### **2.2.2. Mạng nơ-ron nhân tạo:**



*Hình 2 Mô hình mạng nơron sinh học*

Mạng nơ-ron sinh học là một mạng lưới (plexus) các nơ-ron có kết nối hoặc có liên quan về mặt chức năng trực thuộc hệ thần kinh ngoại biên (peripheral nervous system) hay hệ thần kinh trung ương (central nervous system). Trong ngành thần kinh học (neuroscience), nó thường được dùng để chỉ một nhóm nơ-ron thuộc hệ thần kinh là đối tượng của một nghiên cứu khoa học nhất định.

Hệ thần kinh con người có khoảng  $10^{10}$  tế bào thần kinh được gọi là các nơron, mỗi nơron có thể liên kết với  $10^4$  nơron khác thông qua khớp nối.

Mỗi nơron gồm có 3 phần: thân nơron có nhiệm vụ tiếp nhận hay phát ra các xung thần kinh, bên trong có nhân (Soma), hệ thống dây thần kinh vào (dendrites – còn gọi là các nhánh thụ giác) và một đầu dây thần kinh ra (sợi trục axon – nhánh trục giác) để dẫn truyền các xung thần kinh. Các đầu dây thần kinh vào nhận tín hiệu từ các nơron khác, nhân nơron sẽ sinh ra tín hiệu ở đầu ra của nơron và truyền tới các nơron khác được kết nối với đầu ra qua trục.

#### **2.2.2.1 Quá trình phát triển, mô hình và quá trình xử lý trong mạng nơron nhân tạo:**

##### **2.2.2.1.1 Quá trình phát triển:**

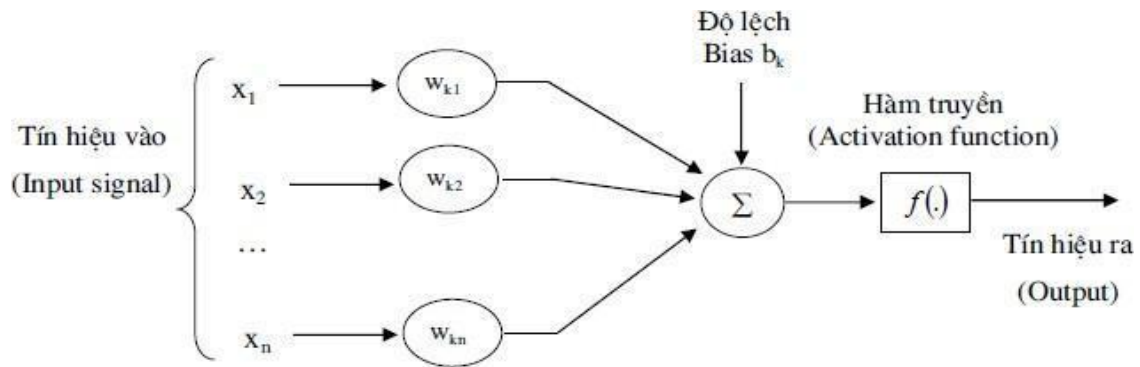


- Năm 1943, McCulloch và Pitts đã đưa ra khả năng liên kết và một số liên kết cơ bản của mạng noron.
- Năm 1949, Hebb đã đưa ra các luật thích nghi trong mạng noron.
- Năm 1958, Rosenblatt đưa ra cấu trúc Perception.
- Năm 1969, Minsky và Papert phân tích sự đúng đắn của Perception.
- Năm 1976, Grossberg dựa vào tính chất sinh học đã đưa ra một số cấu trúc của hệ động học phi tuyến với các tính chất mới.
- Cho đến những năm 1980 việc nghiên cứu mạng noron phát triển mạnh mẽ cùng với sự phát triển vượt bậc của PC. Có hai khái niệm mới liên quan tới sự hồi sinh này:
  - o Việc sử dụng các phương pháp thống kê để giải thích hoạt động của một lớp các mạng hồi quy (recurrent network) có thể được dùng như bộ nhớ liên hợp (associative memory) trong công trình nghiên cứu nhà vật lý học John Hopfield.
  - o Sự ra đời của thuật toán lan truyền ngược (back - propagation) để luyện các mạng nhiều lớp.
- Năm 1982, Hopfield đã đưa ra mạng học phi tuyến với các tính chất mới và Rumelhart đưa ra mô hình song song (Parallel Distributer Processing – PDS) và một số kết quả, thuật toán.

#### **2.2.2.1.2 Mô hình và quá trình xử lý trong mạng noron nhân tạo:**

Để mô phỏng các tế bào thần kinh và các khớp nối thần kinh của bộ não con người, trong mạng noron nhân tạo cũng có các thành phần có vai trò tương tự là các noron nhân tạo và kết nối giữa chúng( kết nối này gọi là weights).

Noron là một đơn vị tính toán có nhiều đầu vào và một đầu ra, mỗi đầu vào đến từ một synapse. Đặc trưng của noron là một hàm kích hoạt phi tuyến chuyển đổi một tổ hợp tuyến tính của tất cả các tín hiệu đầu vào thành tín hiệu đầu ra.



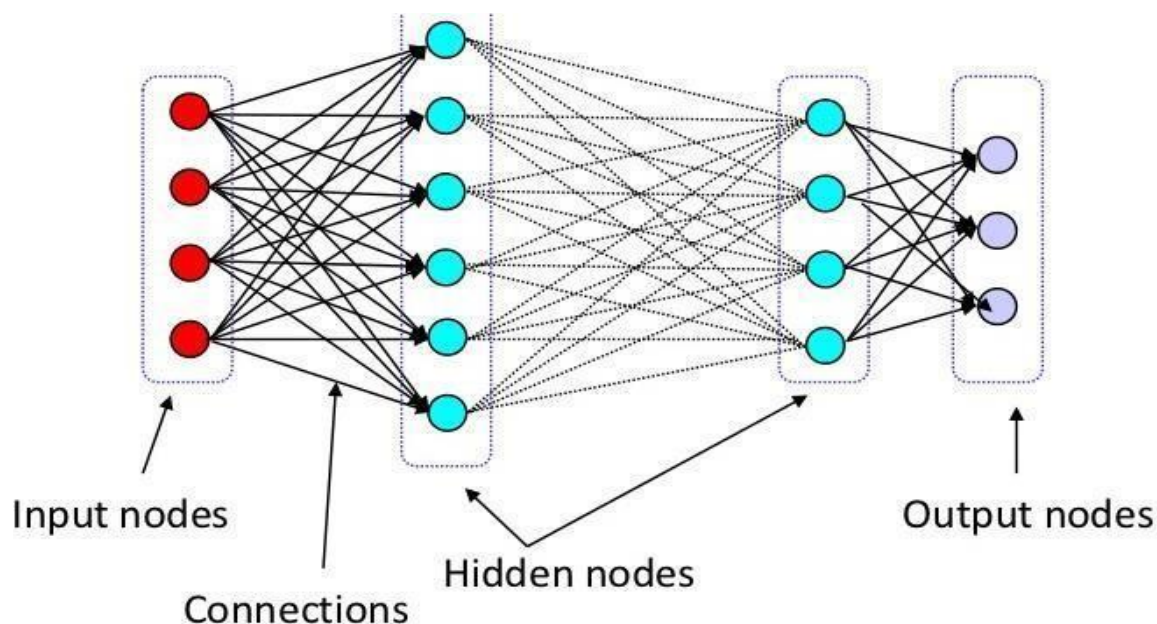
Hình 3 Mô hình của một mạng nơron nhân tạo

- Đầu vào của nơron gồm  $n$  tín hiệu  $x = (x_1, x_2, \dots, x_n)$ , đầu ra tín hiệu  $y = (y_1, y_2, \dots, y_m)$ .
- Một tập hợp các khớp nối và trọng số tương ứng với  $w_i$ .
- Một bộ cộng  $\Sigma$  thực hiện trên các trọng số các khớp nối thường được gọi là bộ kết hợp tính tuyến.
- Một hàm chuẩn không chế giá trị đầu ra của mạng nơron được gọi là hàm truyền hay hàm kích hoạt. Thông thường, tín hiệu đầu ra của một nơron trong khoảng  $[0, 1]$  hoặc  $[-1, 1]$ .
- Trạng thái bên trong của nơron được xác định qua bộ tổng các đầu vào có trọng số  $w$  ( $i = 1, 2, \dots, n$ ). Đầu ra  $y$  được xác định qua hàm phi tuyến  $f$ .

### 2.2.2.2 Cấu trúc và phân loại mạng nơron nhân tạo:

#### 2.2.2.2.1 Cấu trúc của mạng nơron nhân tạo:

Trong mô hình mạng nơron nhân tạo, các nơron được nối với nhau bằng các liên kết nơron, mỗi liên kết có một trọng số đặc trưng cho đặc tính kích hoạt hay ức chế giữa các nơron. Đồng thời, các nơron được nhóm lại với nhau theo cấu trúc phân lớp, bao gồm: lớp vào (input layer), lớp ra (output layer) và lớp ẩn (hidden layer).



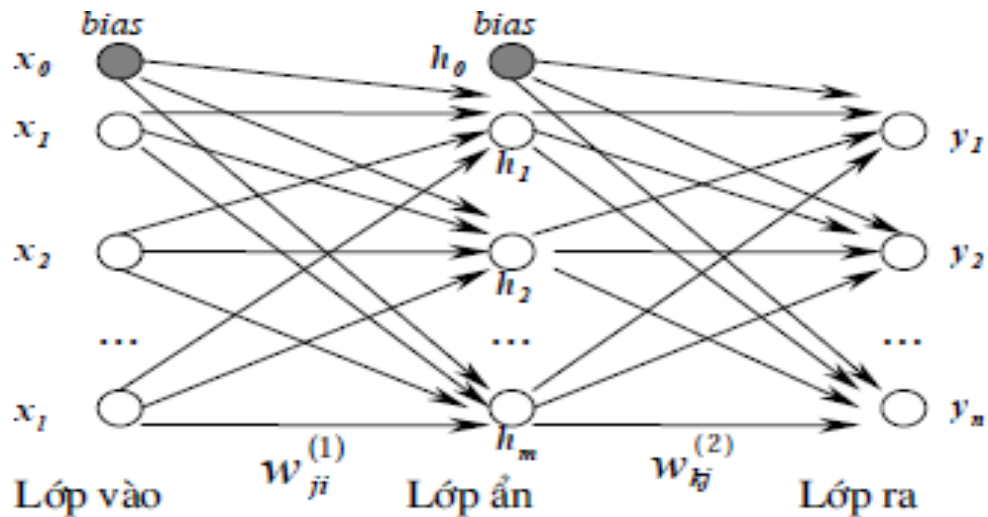
Hình 4. Mô hình cấu trúc của mạng nơron nhân tạo

- **Lớp vào:** Các nút trong lớp vào được gọi là các nút vào, chúng sẽ được mã hóa để đưa vào mạng xử lý. Các nơron vào không xử lý thông tin mà chỉ phát tán thông tin cho các nút khác.
- **Lớp ẩn:** Các nơron trong lớp ẩn chúng tạo thành các mô hình toán học phi tuyến cho mạng và chúng không thể nhìn thấy trực tiếp.
- **Lớp ra:** Các nơron này được gọi là các nút ra, nhiệm vụ của chúng là đưa thông tin ra thích nghi mẫu mã nguồn sử dụng cần.

#### 2.2.2.2.2 Phân loại mạng nơron:

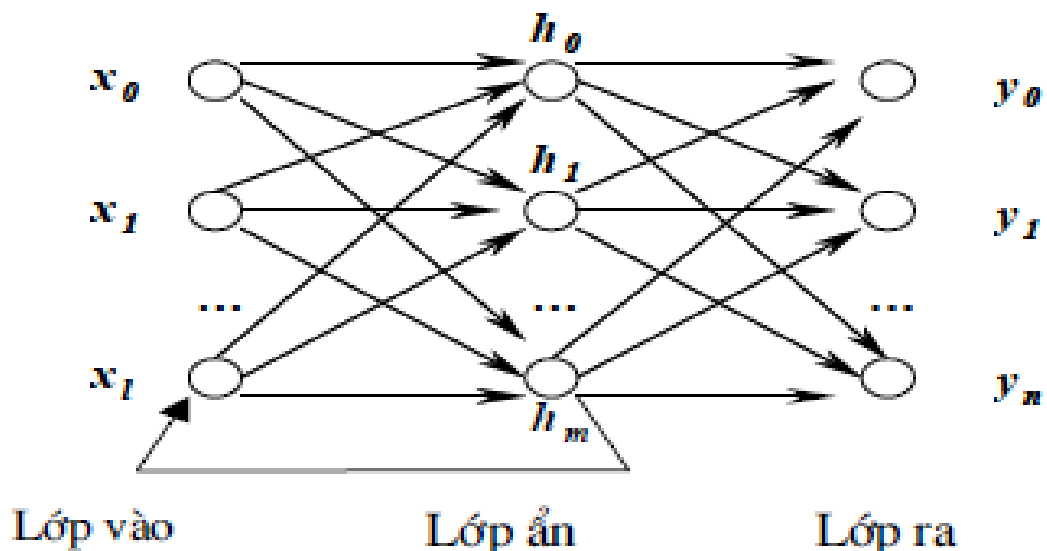
##### 2.2.2.2.2.1 Phân loại theo liên kết:

- **Mạng nơron truyền thẳng (feed – forward Neural Network):** Trong mạng nơron này, các liên kết nơron chỉ truyền theo một hướng từ lớp vào cho tới lớp ra, không tạo thành các chu trình với các đỉnh là các nút nơron, các cung là các liên kết giữa chúng.



Hình 5. Mạng nơron truyền thẳng

- **Mạng nơron hồi quy:** Cho phép các liên kết nơron tạo thành chu trình, có thông tin xử lý hai chiều. Mạng hồi quy có khả năng lưu trữ trạng thái trong của nó dưới dạng các ngưỡng kích hoạt ngoài các trọng số liên kết nơron vì các thông tin ra của các nơron được truyền lại cho các nơron đã góp phần kích hoạt chúng.



Hình 6. Mạng nơron hồi quy

- **Mạng nơron đối xứng và bất đối xứng:** Nếu thỏa mãn yêu cầu có một đường nối từ nút i đến nút j thì cũng có một đường nối từ nút j đến nút i và trọng số tương ứng với hai đường nối này bằng nhau:  $w_{ji} = w_{ij}$ . Mạng không thỏa mãn điều kiện trên là mạng bất đối xứng.

#### 2.2.2.2.2 Phân loại theo số lớp:

Mạng chỉ gồm một lớp vào và một lớp ra được gọi là mạng đơn hay mạng một lớp.

Mạng có từ một lớp ẩn trở lên được gọi là mạng đa lớp hay mạng nhiều lớp.

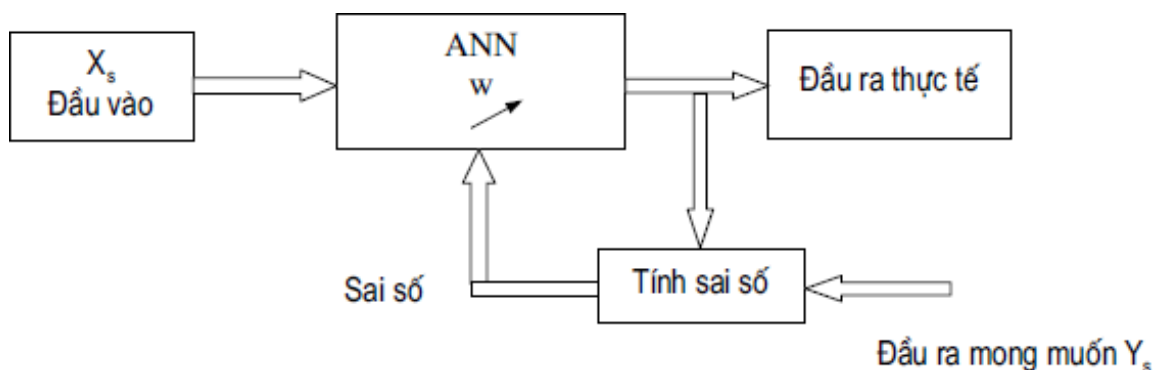
Trong mô hình mạng đa lớp, đầu ra của các phần tử tính toán tại một lớp là đầu vào của lớp tiếp theo. Không cho phép các liên kết giữa các neuron trong cùng một lớp và các liên kết neuron nhảy qua một lớp trở lên.

#### 2.2.2.3 Học và lan truyền trong mạng neuron nhân tạo

- Mạng neuron thực hiện hai chức chính là học và tổng quát hóa.
  - o Học là quá trình hiệu chỉnh các tham số và trọng số liên kết trong mạng để tối thiểu hóa sai số với vector đầu cho trước. Quá trình học dừng lại khi thỏa mãn một điều kiện nào đó ví dụ như khi các trọng số của mạng tạo ra lỗi đủ nhỏ giữa đầu ra mong đợi và kết quả đầu ra của mạng với đầu vào cho trước.
  - o Tổng quát hóa là quá trình đưa vào một vector đầu vào mới và sản sinh ra quyết định dựa trên vector đầu ra tính được từ mạng.
- Có hai kiểu học đó là:
  - **Học có tham số:** là các tham số về trọng số cập nhật kết nối giữa các neuron.
  - **Học cấu trúc:** trọng tâm là sự biến đổi cấu trúc của các mạng neuron gồm số lượng nút và các loại liên kết.

##### 2.2.2.3.1 Học và tổng quát hóa

- **Học có giám sát:**

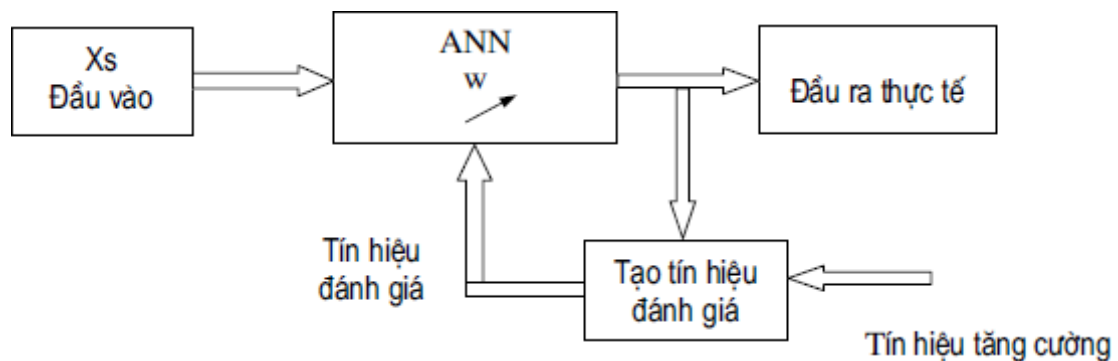


Hình 7. Học có giám sát.

Học có giám sát là mạng được cung cấp một tập mẫu học  $\{(X_s, Y_s)\}$  theo đó  $X_s$  là tập tín hiệu đầu vào và kết quả đúng của hệ phải là  $Y_s$ . Ở mỗi lần học vector tín hiệu  $X_s$  được đưa vào mạng sao đó so sánh sự sai khác giữa các kết quả ra đúng  $Y_s$  với kết quả tính toán qua mạng. Quá trình tiếp tục cho đến khi thỏa mãn một yêu cầu nào đó.

- **Học tăng cường**

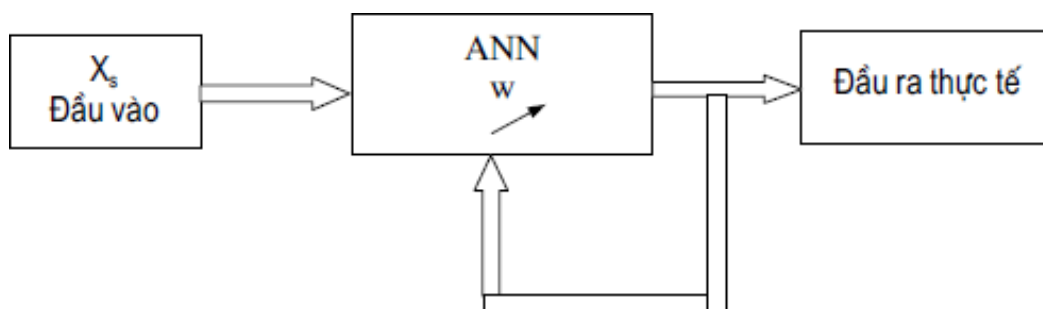
Cũng như kỹ thuật học giám sát, các vector đầu ra được biết chính xác. Nhưng trong trường hợp ít thông tin như output quá lớn hoặc chỉ đúng khoảng 40%. Khi đó một tín hiệu đánh giá là “True” hay “False” quay lại mạng.



Hình 8. Học tăng cường.

- **Học không giám sát**

Trong mô hình học này, đầu ra mong muốn của mạng sẽ không được cho trước và mạng được trang bị khả năng tự tổ chức. Mạng không sử dụng mối quan hệ lớp của các mẫu học mà dùng thông tin kết hợp với nhóm các nơon để thay đổi các tham số cục bộ sao cho hợp nhất.

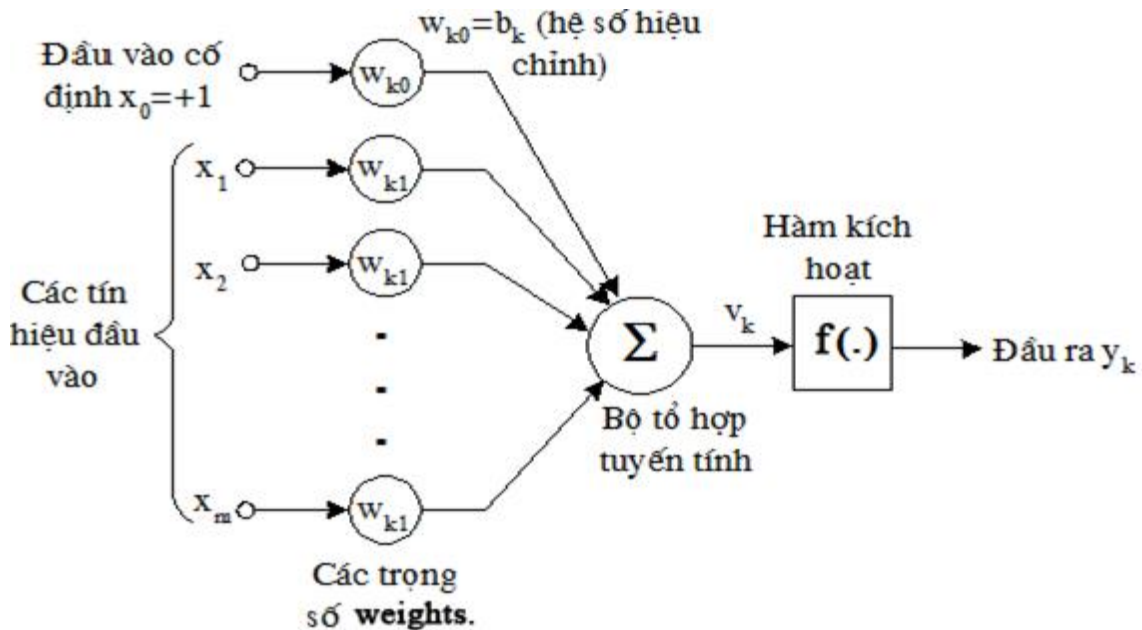


Hình 9. Học không giám sát

### 2.2.2.3.2 Lan truyền trong mạng nơon nhân tạo

### 2.2.2.3.2.1 Các định nghĩa và công thức

Ở đây, chúng ta xác định ba thành phần cơ bản của một mô hình neuron:



Hình 10. Mô hình phi tuyến thứ hai của một neuron.

Một tập hợp các *weights* hay *các kết nối*, mà mỗi một trong chúng được đặc trưng bởi một trọng số của riêng nó. Tức là một tín hiệu  $x_j$  tại đầu vào của weights  $j$  nối với neuron  $k$  sẽ được nhân với trọng số weights  $w_{kj}$ . Ở đó  $k$  là chỉ số của neuron tại đầu ra của weights đang xét, còn  $j$  chỉ điểm đầu vào của weights. Các trọng số weights của một neuron nhân tạo có thể nhận cả các giá trị âm và các giá trị dương.

Một bộ cộng để tính tổng các tín hiệu đầu vào của neuron, đã được nhân với các trọng số weights tương ứng; phép toán được mô tả ở đây tạo nên một bộ tổ hợp tuyến tính.

Một hàm kích hoạt (*activation function*) để giới hạn biên độ đầu ra của neuron. Hàm kích hoạt cũng được xem xét như là một hàm nén; nó nén (giới hạn) phạm vi biên độ cho phép của tín hiệu đầu ra trong một khoảng giá trị hữu hạn. Mô hình neuron trong hình 10 còn bao gồm một hệ số hiệu chỉnh tác động từ bên ngoài,  $b_k$ . Hệ số hiệu chỉnh  $b_k$  có tác dụng tăng lên hoặc giảm đi đầu vào thực của hàm kích hoạt, tùy theo nó dương hay âm. Có nhiều loại hàm kích hoạt.

**2.2.2.3.2.2 Các kiểu hàm kích hoạt**

Hàm kích hoạt, ký hiệu bởi  $f(v)$ , xác định đầu ra của nơron. Dưới đây là các kiểu hàm kích hoạt cơ bản:

**1.Hàm ngưỡng:** Đối với loại hàm này, chúng ta có:

$$f(v) = \begin{cases} 1 & \text{nếu } v \geq 0 \\ 0 & \text{nếu } v < 0 \end{cases}$$

(1.7)

Trong các tài liệu kỹ thuật, dạng hàm ngưỡng này thường được gọi là hàm Heaviside. Đầu ra của nơron k sử dụng hàm ngưỡng sẽ như sau:

- $y_k = 1$  nếu  $v_k \geq 0$
- $y_k = 0$  nếu  $v_k < 0$

(1.8)

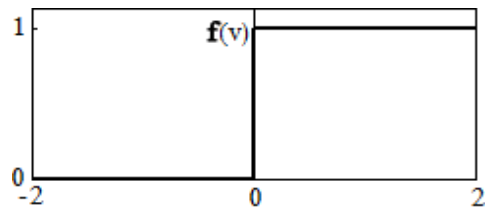
ở đó  $v_k$  là đầu ra của bộ tổ hợp tuyến tính, có nghĩa là:

$$v_k = \sum_{j=1}^m w_{kj} x_j + b_k$$

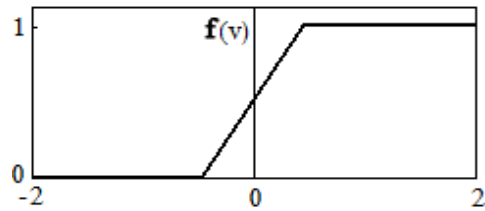
(1.9)



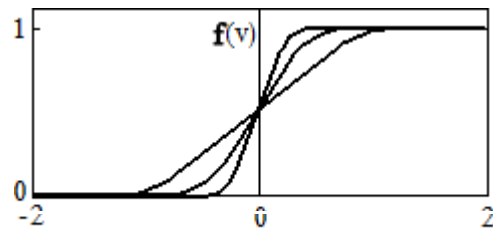
Một nơon như vậy thường được gọi là mô hình *McCulloch-Pitts*.



(a)



(b)



(c)

Hình 11. (a): Hàm ngưỡng, (b): Hàm vùng tuyến tính, (c): Hàm sigma với tham số độ dốc  $a$  thay đổi.

**2. Hàm vùng tuyến tính:** Đối với loại hàm này:

$$f(v) = \begin{cases} 1, & v \geq +\frac{1}{2} \\ v, & +\frac{1}{2} \geq v > -\frac{1}{2} \\ 0, & v \leq -\frac{1}{2} \end{cases}$$

(1.10)

Dạng hàm này có thể được xem như một xấp xỉ của một bộ khuếch đại phi tuyến.

### 3. Hàm sigmoid

Hàm sigmoid là dạng chung nhất của hàm kích hoạt được sử dụng trong cấu trúc mạng nơon nhân tạo. Nó là một hàm tăng và nó thể hiện một sự trung gian

giữa tuyến tính và phi tuyến. Một ví dụ của hàm này là *hàm logistics*, xác định như sau:

$$f(v) = \frac{1}{1 + \exp(-av)}$$

(1.11)

Ở đó  $a$  là tham số độ dốc của hàm sigmoid. Bằng việc biến đổi tham số  $a$ , chúng ta thu được các hàm sigmoid với các độ dốc khác nhau, như được minh họa trong hình 11.c. Thực tế, hệ số góc tại  $v=0$  là  $a/4$ . Khi tham số hệ số góc tiến tới không xác định, hàm sigmoid trở **thành** một hàm ngưỡng đơn giản. Trong khi một hàm ngưỡng chỉ có giá trị là 0 hoặc 1, thì một hàm sigmoid nhận các giá trị từ 0 tới 1. Cũng phải ghi nhận rằng hàm sigmoid là hàm phân biệt, trong khi hàm ngưỡng thì không (Tính phân biệt của hàm là một đặc tính quan trọng trong lý thuyết mạng nơron).

Các hàm kích hoạt được định nghĩa trong các công thức (1.7), (1.10), (1.11) đều trong phạm vi từ 0 đến 1. Đôi khi có những yêu cầu xây dựng hàm kích hoạt trong phạm vi từ -1 đến 1, trong trường hợp này hàm kích hoạt được giả định có dạng đối xứng qua gốc toạ độ (hay có thể gọi là dạng phản đối xứng); nghĩa là hàm kích hoạt là một hàm lẻ. Ví dụ, hàm ngưỡng trong công thức (1.7) bây giờ được xác định như sau:

$$f(v) = \begin{cases} 1 & v > 0 \\ 0 & v = 0 \\ -1 & v < 0 \end{cases}$$

( 1.12)

Hàm này thường được gọi là *hàm signum*.

Mạng nơron lan truyền thông tin từ lớp vào đến lớp ra. Khi việc lan truyền kết thúc, thông tin tại lớp ra chính là kết quả lan truyền.

### 3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

#### 3.1. Phát biểu bài toán

BẢNG DỮ LIỆU DOANH SỐ BÁN HÀNG CỦA ĐÀ NẴNG COMPUTER					
TT	Ngày	Doanh thu	TT	Ngày	Doanh thu
1	1/1/2022	864279406	16	16/1/2022	488002961
2	2/1/2022	597185430	17	17/1/2022	634140258
3	3/1/2022	838533277	18	18/1/2022	701094200
4	4/1/2022	230731875	19	19/1/2022	295762813
5	5/1/2022	402787269	20	20/1/2022	206457195
6	6/1/2022	889245469	21	21/1/2022	888056835
7	7/1/2022	777522086	22	22/1/2022	193231321
8	8/1/2022	276224191	23	23/1/2022	879919553
9	9/1/2022	414855208	24	24/1/2022	670697341
10	10/1/2022	828899226	25	25/1/2022	403240659
11	11/1/2022	575276329			
12	12/1/2022	266946930			
13	13/1/2022	801216595			
14	14/1/2022	824990097			
15	15/1/2022	253248750			

Bảng 3-1. Bảng dữ liệu doanh số bán hàng của Đà Nẵng Computer từ 1/1/2022 đến 25/1/2022

Từ bảng dữ liệu 3.1, bài toán được xây dựng bài toán như sau:

- Lấy 4 ngày đầu của tháng 1 năm 2022 làm dữ liệu input đầu vào để dự báo cho ngày thứ 5.
- Sau đó lại lấy tiến lên một ngày từ ngày thứ 2 đến ngày thứ 5 để dự báo cho ngày thứ 6.
- Cứ tiếp tục như vậy cho đến hết dữ liệu.

Bài toán sẽ gồm 2 pha như sau:

- **Pha 1:** Pha học.
  - Dữ liệu dùng để học sẽ lấy từ ngày 1 đến 24 như vậy ta có 20 mẫu, mỗi mẫu gồm 4 đầu vào là doanh số của 4 ngày.
  - Mẫu 1 là doanh số của ngày 1 đến ngày 4 và đầu ra là ngày 5.
  - Mẫu 2 lấy tiến lên 1 ngày là doanh số từ ngày 2 đến ngày 5 và đầu ra là doanh số ngày 6, cứ như vậy cho tới mẫu dữ liệu 20.
- **Pha 2:** Pha chạy.

- Từ kết quả của pha 1 ta tìm được  $W = [w_{ij}]$ .
- Cho tập dữ liệu ngày 21 đến 24.
- Dự đoán doanh số của ngày 25.
- So sánh độ chính xác với doanh số thực.

### 3.2. Thiết kế mạng nơron

Trong bài toán, ta xác định cặp đầu vào  $(x(k), d(k))$  trong đó  $k=1 \dots 20$  là doanh thu của tháng 1 năm 2022.

#### 3.2.1. Số lớp nơron

Bài toán lựa chọn mô hình mạng nơron với 3 lớp (1 lớp vào, 1 lớp ẩn, 1 lớp ra).

#### 3.2.2. Cấu trúc mạng

Số nơron lớp vào: 4 nơron.

Số nơron lớp ẩn: 4 nơron.

Số nơron lớp ra: Có 1 đầu ra.

Hằng số học: Chọn bằng phương pháp thực nghiệm.

#### 3.2.3. Hàm tương tác đầu ra

Hàm kích hoạt được sử dụng trong lớp vào và lớp ẩn là hàm Log-sigmoid,

### 3.3. Một số khái niệm cần biết:

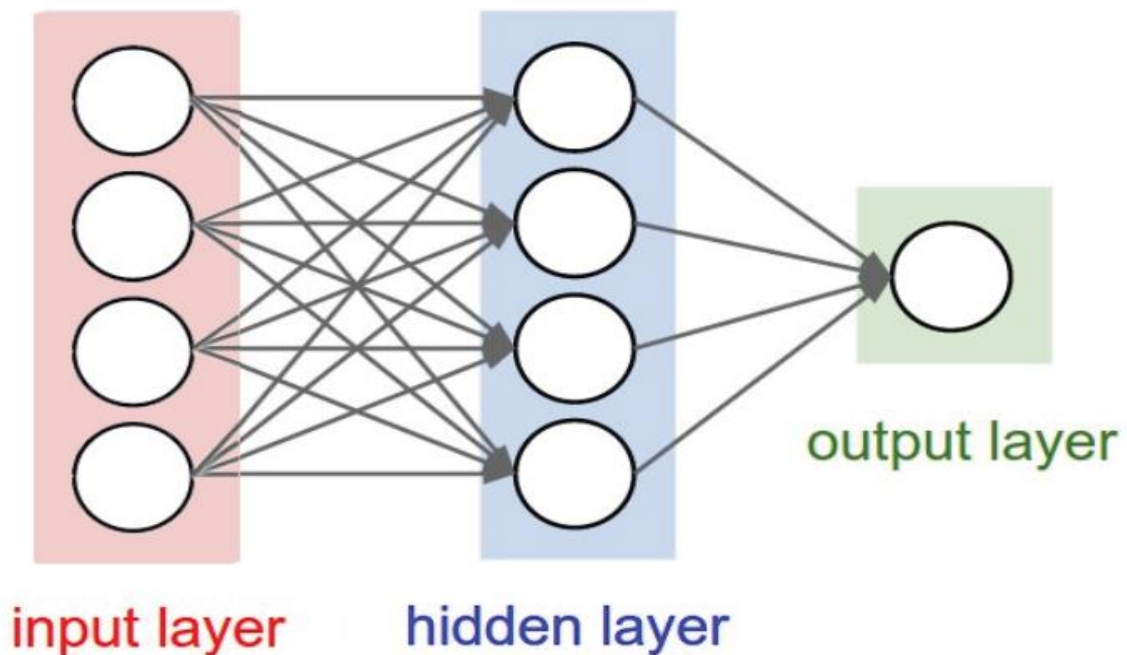
Học trong mạng nơ-ron là quá trình thay đổi giá trị của  $w$  để mạng có thể đưa ra được kết quả mong muốn tối ưu. Sự khác nhau giữa giá trị thật của mẫu và kết quả dự đoán của mạng gọi là lỗi. Quá trình học sẽ tìm các  $w_i^*$  để lỗi nhỏ nhất.

#### Giải thuật:

Giải thuật Backpropagation gồm 2 giai đoạn:

- Lan truyền tiến (tính output của các nơ-ron).
- Lan truyền ngược.

Để minh họa cho giải thuật, chúng ta sẽ xem xét giải thuật Back Propagation hiệu chỉnh các weights của một mạng nơ-ron gồm 4 lớp với 4 đầu vào và một đầu ra. Như hình dưới:



Mỗi nơ-ron sẽ gồm hai thành phần chính. Thứ nhất là các trọng số weights và các trị số đầu vào  $x$ . Phần thứ 2 là hàm kích hoạt  $f$ . Và  $y = f(e)$  là đầu ra của một nơ-ron.

Để dạy cho mạng chúng ta cần phải có các mẫu. Mỗi mẫu bao gồm hai thành phần đó là:

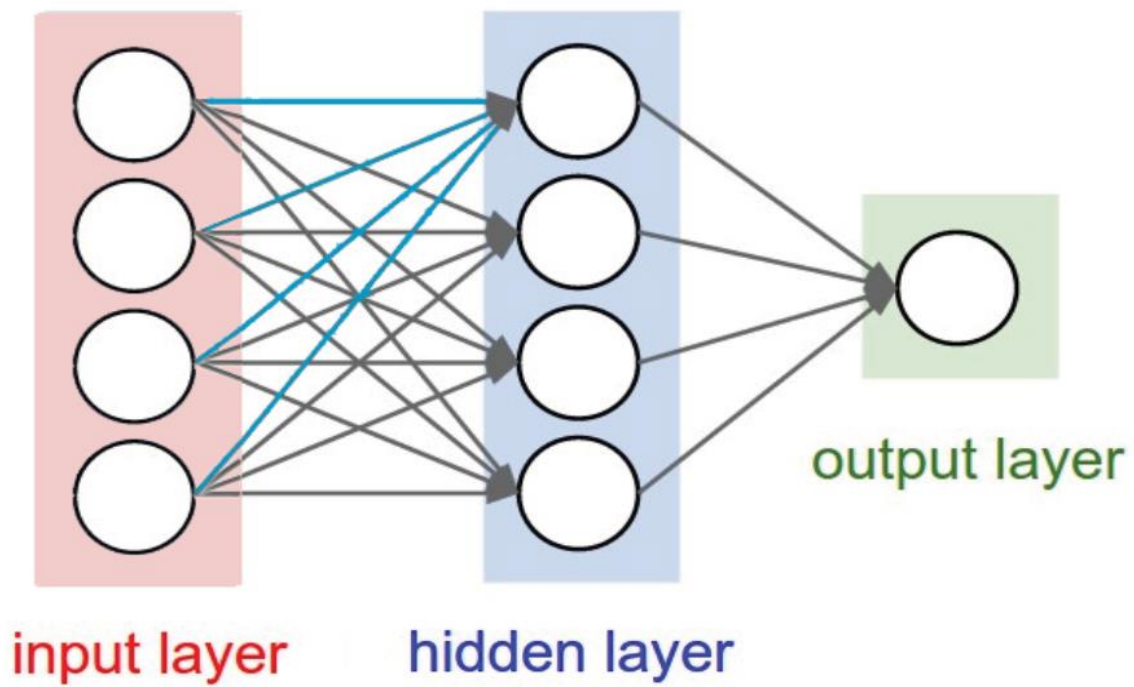
- Đầu vào  $x_1, x_2, x_3, x_4$ ,
- Đầu ra  $z$  là kết quả của mẫu.

### 3.3.1. Đầu tiên là bước lan truyền tiến:

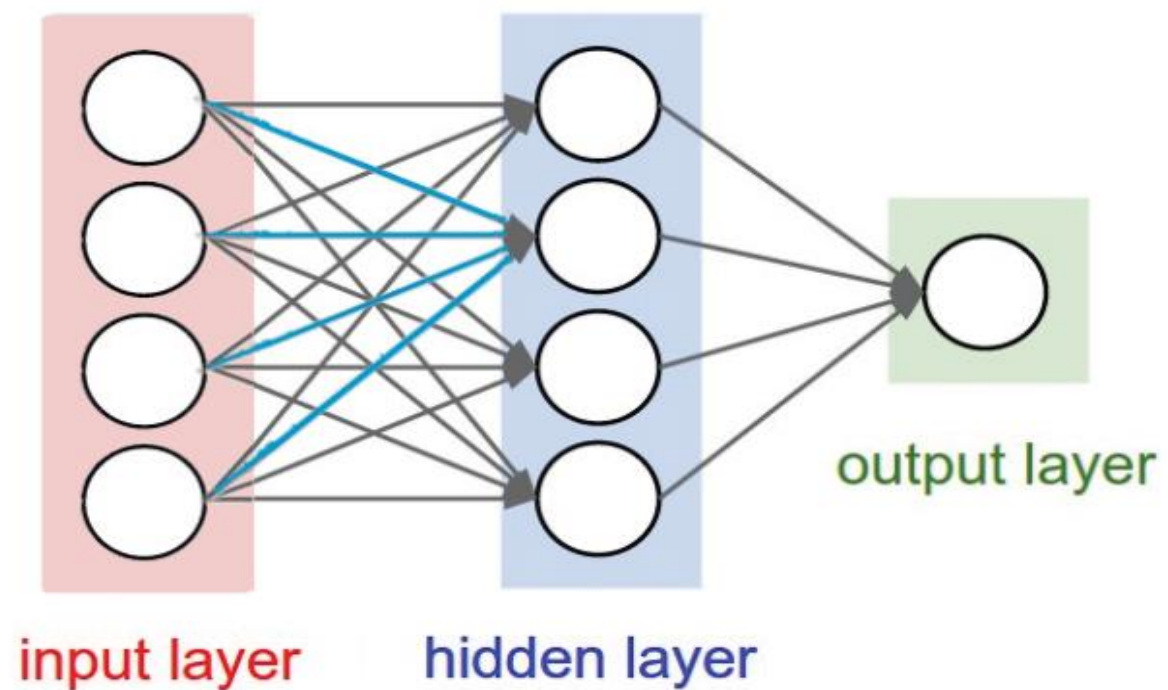
Chúng ta sẽ bắt đầu với 4 đầu vào  $x_1, x_2, x_3, x_4$ . Từ đó chúng ta có thể tính được đầu ra của mỗi nơ-ron trong mạng. Hình dưới minh họa cho sự lan truyền các giá trị trong mạng.

#### Trong đó:

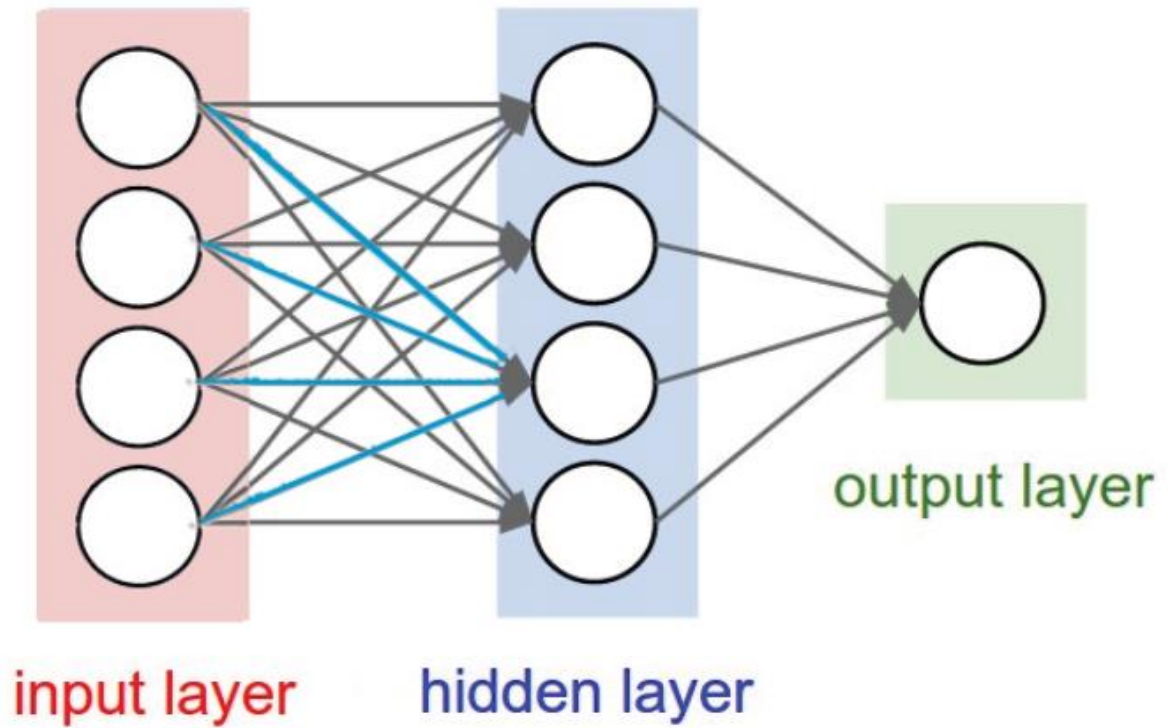
- $W_{(i,j)}$ : tượng trưng cho các trọng số weights của kết nối giữa đầu vào  $x_i$  và nơ-ron  $j$  trong tầng input của mạng.
- $Y_n$  tượng trưng cho đầu ra của nơ-ron  $n$ .



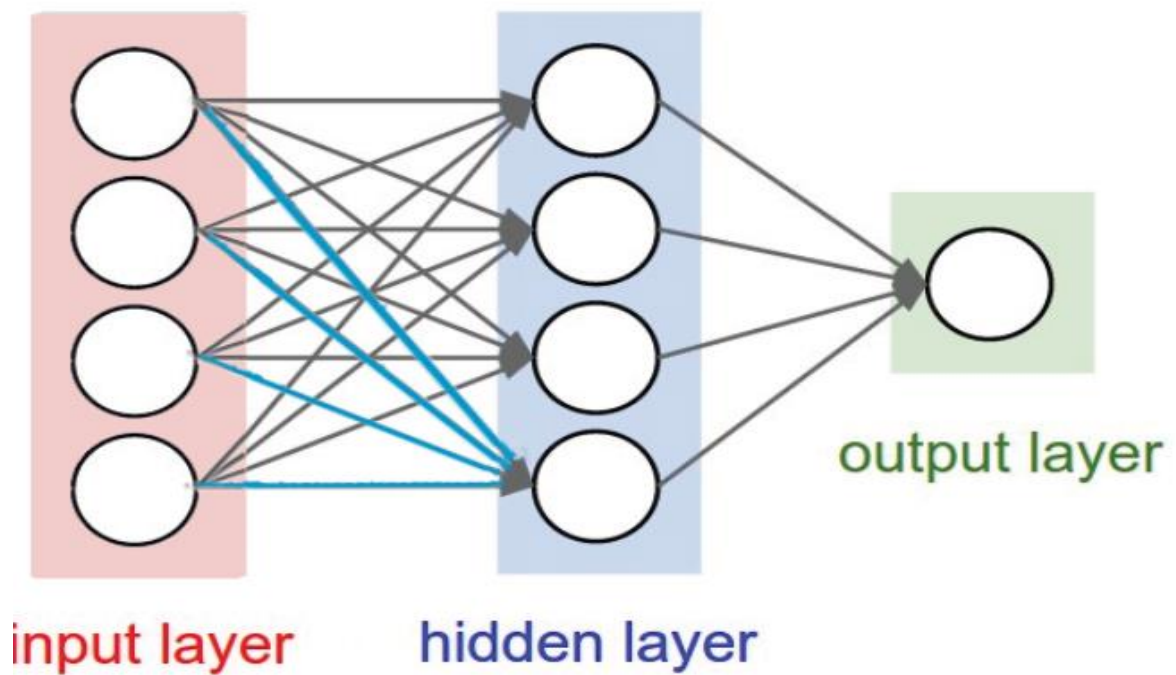
$$y_1 = f_1(w_{(x[1;1])} \cdot X_1 + w_{(x[2;1])} \cdot X_2 + w_{(x[3;1])} \cdot X_3 + w_{(x[4;1])} \cdot X_4) + \text{bias}$$



$$y_2 = f_2(w_{(x[1;2])} \cdot X_1 + w_{(x[2;2])} \cdot X_2 + w_{(x[3;2])} \cdot X_3 + w_{(x[4;2])} \cdot X_4) + \text{bias}$$



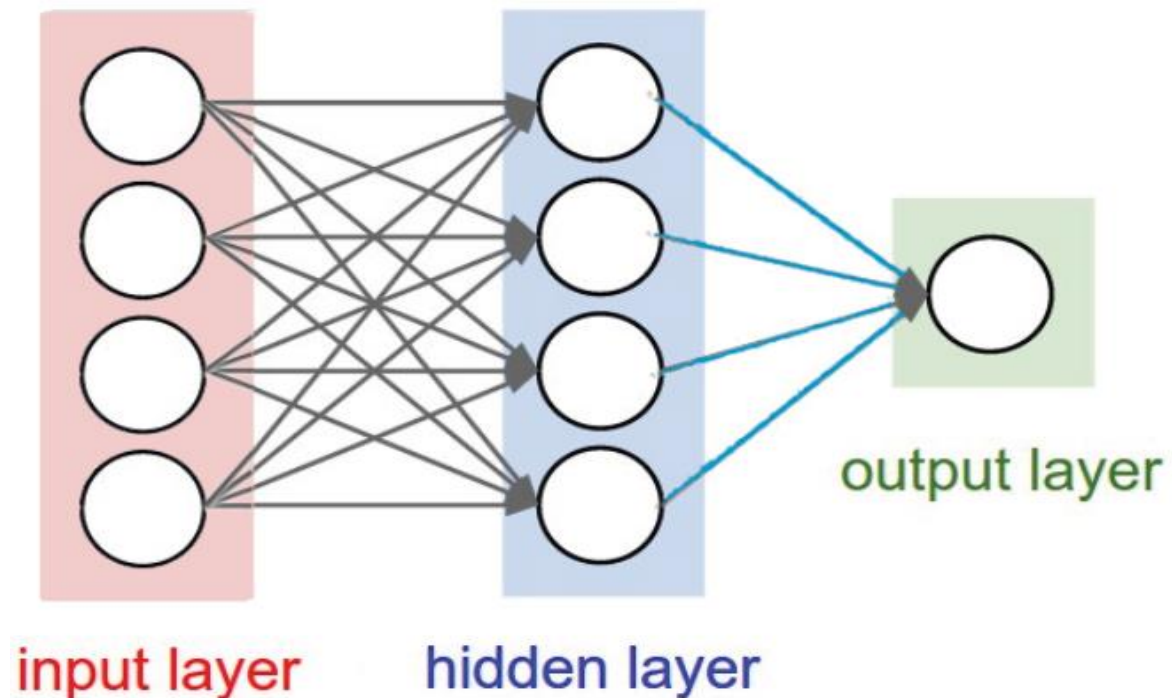
$$y_3 = f_3(w_{(x[1;3])} \cdot X_1 + w_{(x[2;3])} \cdot X_2 + w_{(x[3;3])} \cdot X_3 + w_{(x[4;3])} \cdot X_4) + \text{bias}$$



$$y_4 = f_4(w_{(x[1;4])} \cdot X_1 + w_{(x[2;4])} \cdot X_2 + w_{(x[3;4])} \cdot X_3 + w_{(x[4;4])} \cdot X_4) + \text{bias}$$



Sự lan truyền của tín hiệu đến lớp đầu ra:



$$y_5 = f_5(w_{(x[5;1])}.y_1 + w_{(x[5;2])}.y_2 + w_{(x[5;3])}.y_3 + w_{(x[5;4])}.y_4) + \text{bias}$$

Bước tiếp theo của giải thuật là so sánh giá trị đầu ra của mạng là  $y$  với giá trị  $z$  là kết quả của mẫu. Giảm tín hiệu lỗi bằng cách sử dụng hàm loss function.

$$\text{Error} = \frac{1}{2} (z - y)^2$$

Giai đoạn tiếp theo là điều chỉnh lặp đi lặp lại các trọng số để giảm thiểu lỗi của mạng. Ở đây, thuật toán sử dụng là “gradient descent” trên error function.

### 3.3.2. Lan truyền ngược:

#### 3.3.2.1 Cấu trúc của luật học lan truyền ngược.

Nền tảng của thuật toán cập nhật trọng số này là phương pháp hạ Gradient.

Với cặp mẫu đầu vào – đầu ra mong muốn  $(x(k), d(k))$

Thuật toán lan truyền ngược thực hiện 2 pha.

- Pha 1: Mẫu đầu vào  $x(k)$  được truyền tiến (từ lớp vào tới lớp ra), kết quả của luồng dữ liệu thẳng (forward) là đầu ra thực  $y(k)$ .



- Pha 2: Tín hiệu lỗi được tính trên cơ sở sai khác giữa  $d(k)$  và  $y(k)$  được lan truyền ngược (từ lớp ra quay trở lại các lớp trước đó) để hiệu chỉnh trọng số.

### 3.3.2.2 Luật học lan truyền ngược.

Giải thuật BP (Back-Propagation) bao gồm hai giai đoạn:

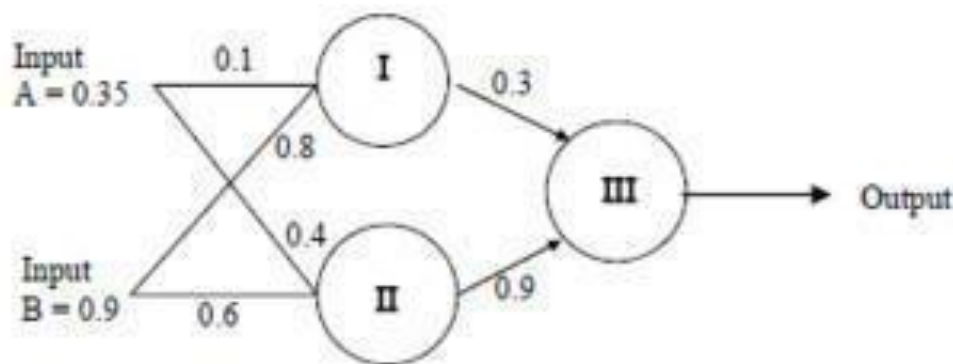
Giai đoạn lan truyền tiến tín hiệu (Signal Forward). Các tín hiệu đầu vào (vector các giá trị đầu vào) được lan truyền tiến từ tầng đầu vào đến tầng đầu ra (đi qua các tầng ẩn).

Giai đoạn lan truyền ngược lỗi (Error Backward).

#### Ví dụ điển hình:

Sau đây chúng ta sẽ xét một ví dụ đơn giản về giải thuật này:

Giả sử chúng ta có một mạng nơron như sau:



Với mạng nơron cho trước các trọng số như hình với hàm kích hoạt là hàm sigmoid (sigmoid như đã nói ở trên), với hệ số hiệu chỉnh bias bằng 1 và hệ số  $\eta = 1$ :

$$\text{Hàm sigmoid: } y = \frac{1}{1+e^{-t}}$$

$$\text{và đạo hàm của hàm sigmoid: } \frac{df(t)}{dt} = \frac{e^t}{(1+e^t)^2}$$

- Mẫu gồm 3 thành phần:

Đầu vào:  $A = 0.35$ ,  $B = 0.9$ .

Đầu ra mong muốn:  $z = 0.5$ .

- Trước hết chúng ta sẽ tính đầu ra cho các nơron:

Nơ - ron 1 : Ta có :  $t1 = (0.35 \times 0.1) + (0.9 \times 0.8) + 1 = 1.755$

Từ đó suy ra:  $y1 = \frac{1}{1+e^{-1.755}} = 0.855$ .

Nơ -ron 2 : ta có  $t2 = (0.9 \times 0.6) + (0.35 \times 0.4) + 1 = 1.68$

Từ đó suy ra :  $y2 = \frac{1}{1+e^{-1.68}} = 0.842$ .

Nơ-ron 3:  $t3 = (0,3 \times 0.855) + (0,9 \times 0.842) + 1 = 2.0143$

$Y3 = \frac{1}{1+e^{-2.0143}} = 0.8823$

- Bước tiếp theo là chúng ta sẽ tính tín hiệu lỗi

$$\delta_3 = z - y = 0.5 - 0.8823 = -0.3823.$$

- Lan truyền tín hiệu lỗi:

Nơ-ron 1:  $\delta_1 = -0.3823 \times 0.3 = -0.11469$

Nơ-ron 2:  $\delta_2 = -0.3823 \times 0.9 = -0.344$

- Bây giờ chúng ta sẽ dựa vào tín hiệu lỗi của mỗi nơron để thay đổi trọng số weights.

$$W'_{A1} = W_{A1} + \eta \delta_1 \frac{df(t)}{dt} x_1 = 0.1 + 1(-0.11469) \frac{e^{1.755}}{(1+e^{1.755})^2} \cdot 0.35 = 0.089.$$

$$W'_{B1} = W_{B1} + \eta \delta_1 \frac{df(t)}{dt} x_1 = 0.8 + 1(-0.11469) \frac{e^{1.755}}{(1+e^{1.755})^2} \cdot 0.35 = 0.789.$$

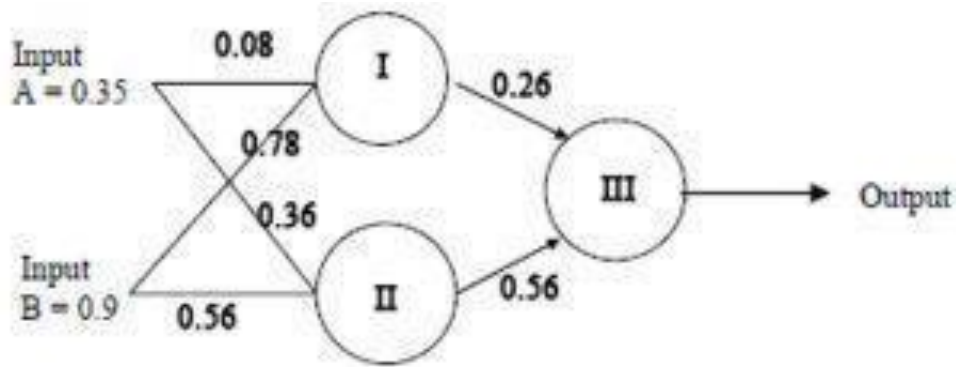
$$W'_{A2} = W_{A2} + \eta \delta_2 \frac{df(t)}{dt} x_2 = 0.4 + 1(-0.344) \frac{e^{1.68}}{(1+e^{1.68})^2} \cdot 0.9 = 0.3067.$$

$$W'_{B2} = W_{B2} + \eta \delta_2 \frac{df(t)}{dt} x_2 = 0.6 + 1(-0.344) \frac{e^{1.68}}{(1+e^{1.68})^2} \cdot 0.9 = 0.5067.$$

$$W'_{13} = W_{13} + \eta \delta_3 \frac{df(t)}{dt} y_1 = 0.3 + 1(-0.3823) \frac{e^{2.0143}}{(1+e^{2.0143})^2} \cdot 0.8525 = 0.258.$$

$$W'_{23} = W_{23} + \eta \delta_3 \frac{df(t)}{dt} y_2 = 0.9 + 1(-0.3823) \frac{e^{2.0143}}{(1+e^{2.0143})^2} \cdot 0.842 = 0.5589.$$

**Và đây là kết quả:**



### 3.3. Cấu trúc dữ liệu

Dữ liệu của bài toán được đưa vào từ file training.txt sau đó được chương trình xử lý và xuất các input và output ra file training-out.txt.

Chương trình chính sẽ đọc dữ liệu từ file training-out.txt để thực hiện bài toán.

### 3.4. Thuật toán

Trình bày các thuật toán và phân tích độ phức tạp của các thuật toán.

#### 3.4.1. Hàm sigmoid

```
float sigmoid(float x)
{
    return 1 / (1 + exp(-x));
}
```

#### 3.4.2. Đạo hàm của sigmoid

```
float sigmoid_derivative(float x)
{
    return sigmoid(x) * (1 - sigmoid(x));
}
```

### 3.4.3. Hàm tính đầu ra cho các Nơ-ron

```

for( j = 1 ; j <= NumHidden ; j++ ) { /* compute hidden unit activations */
    SumH[p][j] = WeightIH[0][j] ;
    for( i = 1 ; i <= NumInput ; i++ ) {
        SumH[p][j] += Input[p][i] * WeightIH[i][j] ;
    }

    Hidden[p][j] = sigmoid(SumH[p][j]);
}
for( k = 1 ; k <= NumOutput ; k++ ) { /* compute output unit activations and errors */
    SumO[p][k] = WeightHO[0][k] ;
    for( j = 1 ; j <= NumHidden ; j++ ) {
        SumO[p][k] += Hidden[p][j] * WeightHO[j][k] ;
    }
    Output[p][k] = sigmoid(SumO[p][k]);
}

```

### 3.4.4. Hàm tính lại trọng số weights

```

for( j = 1 ; j <= NumHidden ; j++ ) { /* update weights WeightIH */
    DeltaWeightIH[0][j] = eta * DeltaH[j] + alpha * DeltaWeightIH[0][j] ;
    WeightIH[0][j] += DeltaWeightIH[0][j] ;
    for( i = 1 ; i <= NumInput ; i++ ) {
        DeltaWeightIH[i][j] = eta * Input[p][i] * DeltaH[j] + alpha * DeltaWeightIH[i][j];
        WeightIH[i][j] += DeltaWeightIH[i][j] ;
    }
}
for( k = 1 ; k <= NumOutput ; k ++ ) { /* update weights WeightHO */
    DeltaWeightHO[0][k] = eta * DeltaO[k] + alpha * DeltaWeightHO[0][k] ;
    WeightHO[0][k] += DeltaWeightHO[0][k] ;
    for( j = 1 ; j <= NumHidden ; j++ ) {
        DeltaWeightHO[j][k] = eta * Hidden[p][j] * DeltaO[k] + alpha * DeltaWeightHO[j][k] ;
        WeightHO[j][k] += DeltaWeightHO[j][k] ;
    }
}

```

### 3.4.5. Hàm tính tín hiệu lỗi

```

Error += 0.5 * (Target[p][k] - Output[p][k]) * (Target[p][k] - Output[p][k]) ; /* SSE */
DeltaO[k] = (Target[p][k] - Output[p][k]) * Output[p][k] * (1.0 - Output[p][k]) ; /* Sigmoidal Outputs, SSE */

```

### 3.4.6. Hàm lan truyền tín hiệu lỗi

```

for( j = 1 ; j <= NumHidden ; j++ ) { /* 'back-propagate' errors to hidden layer */
    SumDOW[j] = 0.0 ;
    for( k = 1 ; k <= NumOutput ; k++ ) {
        SumDOW[j] += WeightHO[j][k] * DeltaO[k] ;
    }
    DeltaH[j] = SumDOW[j] * Hidden[p][j] * (1.0 - Hidden[p][j]) ;
}

```

## **4. CHƯƠNG TRÌNH VÀ KẾT QUẢ**

### **4.3. Tổ chức chương trình**

Chương trình được code dựa trên ngôn ngữ C++

Gồm 6 files training.cpp, proc.cpp, training.exe , proc.exe, training.txt và training-out.txt.

- + training.cpp và proc.cpp là 2 file code chính của chương trình có nhiệm vụ compile và build ra các file training.exe và proc.exe
- + training.exe sẽ xử lý và xuất các input và output ra file training-out.txt.
- + proc.exe sẽ đọc dữ liệu từ file training-out.txt để thực hiện bài toán
- + training.txt là file chứa doanh thu của nhà bán hàng
- + training-out.txt là file chứa các input, output cho chương trình

### **4.4. Cài đặt chương trình**

Chương trình có thể chạy trên DevC, Visual Studio Code,...

## 4.5. Kết quả thực hiện

### 4.5.1. Giao diện chính của chương trình

### 4.5.2. Các kết quả thực thi của chương trình

NETWORK DATA - EPOCH 20000

Pat	Input1	Input2	Input3	Input4	Target1	Output1
1	0.409257	0.282782	0.397066	0.109257	0.190729	0.166331
2	0.272061	0.382012	0.105115	0.183498	0.405115	0.397475
3	0.382012	0.105115	0.183498	0.405115	0.354217	0.348823
4	0.105115	0.183498	0.405115	0.354217	0.125840	0.121878
5	0.197116	0.435178	0.380503	0.135178	0.203022	0.244660
6	0.435178	0.380503	0.135178	0.203022	0.405646	0.399736
7	0.422050	0.149938	0.225189	0.449938	0.312268	0.304410
8	0.147463	0.221472	0.442510	0.307113	0.142510	0.173927
9	0.221472	0.442510	0.307113	0.142510	0.427732	0.410648
10	0.442510	0.307113	0.142510	0.427732	0.440424	0.434902
11	0.301855	0.140070	0.420409	0.432883	0.132883	0.125776
12	0.140070	0.420409	0.432883	0.132883	0.256061	0.232566
13	0.420409	0.432883	0.132883	0.256061	0.332742	0.348532
14	0.432883	0.132883	0.256061	0.332742	0.367873	0.384689
15	0.169645	0.326900	0.424794	0.469645	0.198124	0.178257
16	0.295976	0.384609	0.425217	0.179382	0.125217	0.144765
17	0.279111	0.308580	0.130177	0.090870	0.390870	0.394278
18	0.302707	0.127699	0.089141	0.383430	0.083430	0.094731
19	0.127699	0.089141	0.383430	0.083430	0.379917	0.375930
20	0.089141	0.383430	0.083430	0.379917	0.289582	0.290828

Goodbye!

Inputs 0.383430 0.083430 0.379917 0.289582= 0.164383

### 4.5.3. Nhận xét đánh giá kết quả

- **Ưu điểm:**
  - Tốc độ xử lý nhanh (tùy theo CPU của từng máy mà thời gian có thể dao động từ 0.05s -> 0.5s)
  - Mạng nơron nhân tạo có tính ứng dụng rất đa dạng, khá hiệu quả trong các bài toán dự báo.
- **Nhược điểm:**
  - Tốc độ vẫn chậm hơn so với mạng có 2 lớp ẩn.
  - Khó lựa chọn weights cho lần đầu một cách tương đối chính xác cho bài toán.
  - Khó có thể dự đoán trước được hiệu quả của mạng trong tương lai (khả năng tổng quát hoá).
  - Mức độ chính xác của mô hình phụ thuộc vào rất nhiều thông số, tuy nhiên lại chưa có một phương pháp nào để xác định được chính xác định tính

cũng như định lượng của các thông số. Ta phải thông qua phương pháp thực nghiệm để xác định giá trị thông số tối ưu.

## **5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **5.3. Kết luận**

Đề tài tập trung nghiên cứu kỹ thuật sử dụng mạng nơron và giải thuật di truyền trong khai phá dữ liệu để dự đoán doanh số bán hàng. Với ứng dụng dự đoán này là một bước tiến quan trọng để tận dụng thời cơ phát triển cũng như phòng tránh rủi ro trong tình hình kinh tế đầy biến động.

### **5.4. Hướng phát triển**

- Thu thập số liệu nhiều mẫu hơn để huấn luyện mạng nơron để cho ra một trọng số tối ưu hơn để dự đoán độ chính xác có thể tăng lên.
- Cải tiến giải thuật để nâng cao hiệu quả cho bài toán.

## TÀI LIỆU THAM KHẢO

[1]<https://towardsdatascience.com/simple-neural-network-implementation-in-c-663f51447547>

[2] <https://medium.com/analytics-vidhya/building-neural-network-framework-in-c-using-backpropagation-8ad589a0752d>

[3]<https://codereview.stackexchange.com/questions/191498/simple-neural-network-in-c>

[4]<https://courses.cs.washington.edu/courses/cse599/01wi/admin/Assignments/bpn.html>

[5]<https://www.codeproject.com/Articles/1237026/Simple-MLP-Backpropagation-Artificial-Neural-Netwo>

[6]<https://takinginitiative.wordpress.com/2008/04/23/basic-neural-network-tutorial-c-implementation-and-source-code/>

### Sách tham khảo:

Nguyễn Đình Thúc (2000), Mạng nơron nhân tạo, NXB Giáo dục

Dương Thị Huyền Thanh (2006), Kỹ thuật mạng nơron và giải thuật di truyền trong khai phá dữ liệu và thử nghiệm ứng dụng.

Nguyễn Văn Minh, Mạng Nơron và giải thuật Back Propagation.



## **PHỤ LỤC**

*Link code:* [https://github.com/PBL-Freshman/PBL/tree/master/PBLFinal\\_1](https://github.com/PBL-Freshman/PBL/tree/master/PBLFinal_1)