

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**DƯƠNG THỊ HIỀN THANH**

**KỸ THUẬT MẠNG NƠON VÀ GIẢI THUẬT  
DI TRUYỀN TRONG KHAI PHÁ DỮ LIỆU  
VÀ THỬ NGHIỆM ỨNG DỤNG**

**LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN**

**HÀ NỘI – 2008**

## MỤC LỤC

Mục lục.....	1
Danh mục các từ viết tắt.....	3
Danh mục các bảng.....	4
Danh mục các hình vẽ và đồ thị.....	5
Lời nói đầu.....	6
<b>Chương 1. KHAI PHÁ DỮ LIỆU VÀ PHÁT HIỆN TRI THỨC TRONG CSDL .....</b>	<b>8</b>
1.1. TỔNG QUAN VỀ KHAI PHÁ DỮ LIỆU VÀ PHÁT HIỆN TRI THỨC TRONG CSDL .....	8
1.1.1. Tại sao cần phát hiện tri thức? .....	8
1.1.2. Khai phá dữ liệu và phát hiện tri thức trong cơ sở dữ liệu .....	9
1.2. QUÁ TRÌNH PHÁT HIỆN TRI THỨC TRONG CƠ SỞ DỮ LIỆU.....	10
1.2.2. Thu thập và tiền xử lý dữ liệu .....	10
1.2.3. Khai phá dữ liệu.....	12
1.2.4. Minh họa và đánh giá.....	12
1.2.5. Đưa kết quả vào thực tế.....	13
1.3. CÁC KỸ THUẬT KHAI PHÁ DỮ LIỆU .....	13
1.3.1. Kiến trúc của hệ thống khai phá dữ liệu .....	13
1.3.3. Nhiệm vụ chính của khai phá dữ liệu.....	17
1.3.4. Một số phương pháp khai phá dữ liệu phổ biến .....	19
1.3.5. Những ưu thế và khó khăn thách thức trong nghiên cứu và ứng dụng kỹ thuật khai phá dữ liệu .....	24
❖ KẾT LUẬN CHƯƠNG 1 .....	27
<b>Chương 2. KỸ THUẬT KHAI PHÁ DỮ LIỆU SỬ DỤNG MẠNG NƠON VÀ GIẢI THUẬT DI TRUYỀN .....</b>	<b>21</b>
2.1. MẠNG NƠON TRONG KHAI PHÁ DỮ LIỆU .....	28
2.1.1. Khái niệm mạng nơon .....	28
2.1.2. Nơon sinh học và mạng nơon sinh học .....	29
2.1.3. Mô hình và quá trình xử lý trong nơon nhân tạo .....	30
2.1.4. Cấu trúc và phân loại mạng nơon .....	33
2.1.5. Học và lan truyền trong mạng.....	36
2.1.6. Đánh giá về mạng nơon .....	40

2.2. GIẢI THUẬT DI TRUYỀN TRONG KHAI PHÁ DỮ LIỆU .....	42
2.2.1. Cơ bản về giải thuật di truyền .....	42
2.2.2. Một số cách biểu diễn lời giải của giải thuật di truyền.....	45
2.2.3. Các toán tử di truyền .....	46
2.2.4. Cơ sở toán học của giải thuật di truyền.....	52
2.2.5. Những cải tiến của giải thuật di truyền .....	54
❖ KẾT LUẬN CHƯƠNG 2 .....	56
<b>Chương 3. TÍCH HỢP GIẢI THUẬT DI TRUYỀN VỚI GIẢI THUẬT HUẤN LUYỆN</b>	
<b>MẠNG NƠON TRUYỀN THẮNG NHIỀU LỚP .....</b>	<b>50</b>
3.1. ĐẶT VẤN ĐỀ .....	57
3.2. MẠNG NƠON TRUYỀN THẮNG NHIỀU LỚP VỚI GIẢI THUẬT LAN TRUYỀN	
NGƯỢC SAI SỐ VÀ MỘT SỐ CẢI TIẾN .....	57
3.2.1. Kiến trúc của mạng nơon truyền thẳng nhiều lớp.....	57
3.2.2. Cơ chế học của mạng nơon truyền thẳng nhiều lớp.....	59
3.2.3. Thuật toán lan truyền ngược sai số .....	60
3.2.2. Một số cải tiến của giải thuật BP .....	71
3.3. KẾT HỢP GIẢI THUẬT DI TRUYỀN VỚI GIẢI THUẬT BP .....	73
3.3.1. Giải thuật GA trong huấn luyện mạng nơon truyền thẳng nhiều lớp .....	73
3.3.2. Ghép nối với giải thuật lan truyền ngược sai số.....	75
❖ KẾT LUẬN CHƯƠNG 3 .....	76
<b>Chương 4. ỨNG DỤNG TRONG BÀI TOÁN DỰ BÁO DỮ LIỆU .....</b>	<b>71</b>
4.1. GIỚI THIỆU BÀI TOÁN.....	78
4.2. MÔ HÌNH HOÁ BÀI TOÁN, THIẾT KẾ DỮ LIỆU VÀ GIẢI THUẬT.....	80
4.2.1. Mô hình hoá bài toán .....	80
4.2.2. Thiết kế dữ liệu .....	81
4.2.3. Thiết kế giải thuật .....	82
4.3. CHƯƠNG TRÌNH DỰ BÁO DỮ LIỆU .....	93
❖ KẾT LUẬN CHƯƠNG 4 .....	98
Kết luận .....	99
Tài liệu tham khảo.....	100

**DANH MỤC CÁC TỪ VIẾT TẮT**

STT	TỪ VIẾT TẮT	NGHĨA TIẾNG VIỆT	TIẾNG ANH
1	ANN	Mạng nơon nhân tạo	Artificial Neural Network
2	BNN	Mạng nơon sinh học	Biological Neural Network
3	BP	Giải thuật lan truyền ngược của sai số	Back-Propagation of Error
4	CSDL	Cơ sở dữ liệu	Data Base
5	DM	Khai phá dữ liệu	Data Mining
6	GA	Giải thuật di truyền	Genetic Algorithm
7	KDD	Phát hiện tri thức trong CSDL	Knowledge Discover in Database

## DANH MỤC CÁC BẢNG

Bảng 1.1: Dữ liệu học trong ví dụ quyết định đi chơi tennis.....	20
Bảng 2.1: Ví dụ dùng phép tái tạo.....	48
Bảng 2.2: Quá trình tái tạo .....	51
Bảng 2.3: Quá trình lai ghép.....	51
Bảng 3.1: Các hàm kích hoạt.....	69
Bảng 4.1: Số liệu thử nghiệm của bài toán dự báo .....	79

## DANH MỤC CÁC HÌNH VẼ VÀ ĐỒ THỊ

Hình 1.1: Quá trình phát hiện tri thức trong CSDL .....	10
Hình 1.2: Kiến trúc của hệ thống khai phá dữ liệu .....	14
Hình 1.3: Quá trình khai phá dữ liệu.....	15
Hình 1.4: Kết quả của phân cụm.....	18
Hình 1.5: Cây quyết định đi chơi tennis.....	20
Hình 2.1: Cấu tạo của nơon.....	29
Hình 2.2: Thu nhận tín hiệu trong nơon.....	30
Hình 2.3: Mô hình của một nơon nhân tạo .....	31
Hình 2.4: Hàm Sigmoidal.....	33
Hình 2.5: Mạng nơon truyền thẳng nhiều lớp.....	35
Hình 2.6: Mạng hồi quy .....	35
Hình 2.7: Sơ đồ học tham số có giám sát .....	37
Hình 2.8: Sơ đồ học tăng cường .....	38
Hình 2.9: Sơ đồ học không giám sát .....	38
Hình 3.1: Mạng nơon truyền thẳng 2 lớp.....	58
Hình 3.2: Sơ đồ hiệu chỉnh các trọng số của giải thuật BP .....	59
Hình 3.3: Sơ đồ mã hoá các trọng số của mạng nơon.....	74
Hình 3.4: Sơ đồ của giải thuật lai .....	76
Hình 4.1: Sơ đồ khối giải thuật <i>Phân hệ 1</i> .....	84
Hình 4.2: Sơ đồ khối giải thuật <i>Phân hệ 1.1</i> .....	86
Hình 4.3: Sơ đồ khối giải thuật <i>Phân hệ 1.2</i> .....	89
Hình 4.4: Sơ đồ khối giải thuật <i>Phân hệ 2</i> .....	91
Hình 4.5: Màn hình chính của chương trình dự báo.....	93
Hình 4.6: Dữ liệu tệp huấn luyện .....	94
Hình 4.7: Màn hình nhập tham số cho mạng nơon.....	94
Hình 4.8: Màn hình nhập tham số cho giải thuật GA .....	95
Hình 4.9: Tìm kiếm bằng giải thuật GA.....	95
Hình 4.10: Huấn luyện bằng giải thuật BP.....	96
Hình 4.11: Màn hình dự báo .....	98

## LỜI NÓI ĐẦU

Trong những năm gần đây, vai trò của máy tính trong việc lưu trữ và xử lý thông tin ngày càng trở nên quan trọng. Bên cạnh đó, các thiết bị thu thập dữ liệu tự động cũng phát triển mạnh góp phần tạo ra những kho dữ liệu khổng lồ. Dữ liệu được thu thập và lưu trữ ngày càng nhiều nhưng người ra quyết định lại cần có những thông tin bổ ích, những “tri thức” rút ra từ những nguồn dữ liệu hơn là chính dữ liệu đó cho việc ra quyết định của mình.

Với những yêu cầu đó, các mô hình CSDL truyền thống và ngôn ngữ thao tác dữ liệu không còn thích hợp nữa. Để có được tri thức từ CSDL, người ta đã phát triển các lĩnh vực nghiên cứu về tổ chức các kho dữ liệu và kho thông tin, các hệ trợ giúp ra quyết định, các phương pháp khai phá dữ liệu và phát hiện tri thức trong CSDL. Trong số đó, khai phá dữ liệu và phát hiện tri thức đã trở thành một lĩnh vực nghiên cứu rất sôi động.

Luận văn tập trung nghiên cứu kỹ thuật sử dụng mạng nơon và giải thuật di truyền trong khai phá dữ liệu, đặc biệt là giải pháp tích hợp giải thuật di truyền với giải thuật huấn luyện mạng nơon. Trên cơ sở đó, luận văn xây dựng chương trình dự báo dữ liệu sử dụng mạng nơon truyền thẳng huấn luyện bằng giải thuật lai GA-BP.

Luận văn được trình bày gồm 4 chương với nội dung chính như sau :

Chương 1: Trình bày một cách tổng quan về khai phá dữ liệu và phát hiện tri thức trong CSDL. Trong đó đề cập đến các khái niệm, quá trình phát hiện tri thức, nhiệm vụ chính và các phương pháp khai phá dữ liệu cũng như những vấn đề thách thức trong nghiên cứu và áp dụng kỹ thuật khai phá dữ liệu vào thực tế.

Chương 2: Nghiên cứu kỹ thuật khai phá dữ liệu sử dụng mạng nơon và giải thuật di truyền, cụ thể là những vấn đề về lựa chọn cấu trúc mạng và các tham số, xây dựng giải thuật học và lan truyền trong mạng nơon, cũng như cách biểu diễn lời giải, các toán tử di truyền cơ bản và những cải tiến của giải thuật di truyền. Đồng thời, chương 2 cũng đưa ra những đánh giá về hiệu quả của kỹ thuật sử dụng mạng nơon và giải thuật di truyền trong khai phá dữ liệu, qua đó có thể định hướng cho việc lựa chọn phương pháp khai phá thích hợp cho các vấn đề thực tế.

Chương 3 : Giới thiệu kiến trúc mạng nơon truyền thẳng nhiều lớp, giải thuật BP, các vấn đề về sử dụng giải thuật BP và trình bày giải pháp tích hợp giải thuật GA với giải thuật BP trong huấn luyện mạng nơon truyền thẳng nhiều lớp.

Chương 4 : Giới thiệu bài toán ứng dụng dự báo lũ trên sông, từ đó mô hình hoá bài toán, thiết kế thuật toán, dữ liệu và cài đặt chương trình thử nghiệm với công cụ mạng nơon truyền thẳng huấn luyện bằng giải thuật lai GA-BP.



## **CHƯƠNG 1:**

# **KHAI PHÁ DỮ LIỆU VÀ PHÁT HIỆN TRI THỨC TRONG CSDL**

## **1.1. TỔNG QUAN VỀ KHAI PHÁ DỮ LIỆU VÀ PHÁT HIỆN TRI THỨC TRONG CƠ SỞ DỮ LIỆU**

### **1.1.1. Tại sao cần phát hiện tri thức?**

Hơn hai thập niên trở lại đây, lượng thông tin được lưu trữ trên các thiết bị điện tử không ngừng tăng lên. Việc tích lũy dữ liệu diễn ra với một tốc độ bùng nổ. Người ta ước đoán rằng lượng thông tin trên toàn cầu tăng gấp đôi sau khoảng hai năm và theo đó kích thước cơ sở dữ liệu (CSDL) cũng tăng lên một cách nhanh chóng, cả về số bản ghi của CSDL lẫn số trường, thuộc tính trong bản ghi.

Lượng dữ liệu khổng lồ này thực sự là nguồn tài nguyên rất giá trị vì thông tin chính là yếu tố then chốt trong mọi hoạt động. Tuy nhiên, dữ liệu sẽ không có đầy đủ ý nghĩa nếu không phát hiện ra những tri thức tiềm ẩn có giá trị trong đó. Những tri thức này thường rất nhỏ so với lượng dữ liệu, do đó phát hiện ra chúng là một vấn đề khá khó khăn.

Việc xây dựng các hệ thống có khả năng phát hiện được các mẫu tri thức có giá trị trong khối dữ liệu đồ sộ như vậy gọi là phát hiện tri thức trong cơ sở dữ liệu (Knowledge Discover in Database\_KDD). Các kỹ thuật xử lý cơ bản chính là kỹ thuật khai phá dữ liệu (Data Mining\_DM). Việc phân tích dữ liệu một cách tự động và mang tính dự báo của KDD có ưu thế hơn hẳn so với các phương pháp phân tích thông thường, dựa trên những sự kiện trong quá khứ của các hệ hỗ trợ ra quyết định truyền thống trước đây.

Với tất cả những ưu thế đó, KDD đã chứng tỏ được tính hữu dụng của nó trong môi trường đầy tính cạnh tranh ngày nay. KDD đã và đang trở thành một hướng nghiên cứu chính của lĩnh vực khoa học máy tính và công nghệ tri thức. Phạm vi ứng dụng của KDD ban đầu chỉ là trong lĩnh vực thương mại và tài chính.

Cho đến nay, KDD đã được ứng dụng rộng rãi trong các lĩnh vực khác như viễn thông, giáo dục, điều trị y học, ... Có thể nói, KDD là một sự cố gắng để giải quyết vấn đề nan giải của kỷ nguyên thông tin số: vấn đề tràn dữ liệu.

### 1.1.2. Khai phá dữ liệu và phát hiện tri thức trong cơ sở dữ liệu

Khái niệm “phát hiện tri thức trong cơ sở dữ liệu” được đưa ra lần đầu tiên vào năm 1989, trong đó nhấn mạnh rằng tri thức là sản phẩm cuối cùng của quá trình khai phá dữ liệu. Phát hiện tri thức trong cơ sở dữ liệu được định nghĩa như là quá trình chất lọc tri thức từ một lượng lớn dữ liệu. Nói cách khác, có thể quan niệm KDD là một ánh xạ dữ liệu từ mức thấp thành các dạng cô đọng hơn, tóm tắt và hữu ích hơn. Một ví dụ trực quan thường được dùng là việc khai thác vàng từ đá và cát, người khai thác muốn chất lọc vàng từ đá và cát trong điều kiện lượng đá và cát rất lớn.

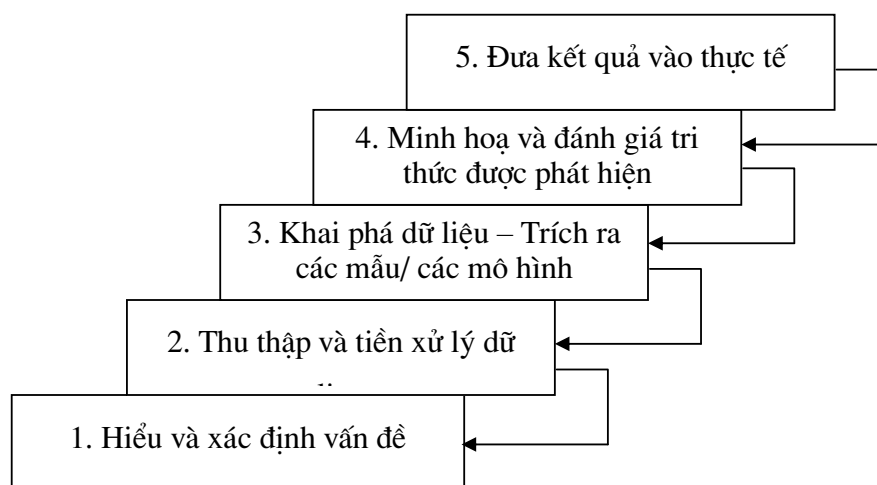
Thuật ngữ “data mining” ám chỉ việc tìm kiếm một tập hợp nhỏ tri thức, thông tin có giá trị từ một lượng lớn các dữ liệu thô [7]. Nó bao hàm một loạt các kỹ thuật nhằm phát hiện ra những thông tin có giá trị tiềm ẩn trong các CSDL lớn. Nhiều thuật ngữ hiện được dùng cũng có nghĩa tương tự với từ data mining như knowledge mining (khai phá tri thức), knowledge extraction (chất lọc tri thức), data/pattern analysis (Phân tích dữ liệu/mẫu), data archaeology (khảo cổ dữ liệu), data dredging (nạo vét dữ liệu).

Như vậy, nếu quan niệm tri thức là mối quan hệ giữa các phần tử dữ liệu thì phát hiện tri thức chỉ quá trình chiết suất tri thức từ cơ sở dữ liệu, trong đó trải qua nhiều giai đoạn khác nhau. Khai phá dữ liệu sử dụng các giải thuật đặc biệt để chiết xuất ra các mẫu, các mô hình từ dữ liệu và chỉ là một giai đoạn trong quá trình phát hiện tri thức trong CSDL.

Phát hiện tri thức trong CSDL và khai phá dữ liệu là một kỹ thuật mới xuất hiện và có tốc độ phát triển rất nhanh. Ngoài ra nó còn là một lĩnh vực đa ngành, liên quan đến nhiều lĩnh vực khác như: lý thuyết thuật toán, Data Warehouse, OLAP, tính toán song song, ... nhưng chủ yếu dựa trên nền tảng của xác suất thống kê, cơ sở dữ liệu và học máy.

## 1.2. QUÁ TRÌNH PHÁT HIỆN TRI THỨC TRONG CƠ SỞ DỮ LIỆU

Hình 1.1 mô tả 5 giai đoạn trong quá trình phát hiện tri thức từ cơ sở dữ liệu. Mặc dù có 5 giai đoạn, song phát hiện tri thức từ cơ sở dữ liệu là một quá trình tương tác và lặp đi lặp lại thành một chu trình liên tục theo kiểu xoáy tròn ốc, trong đó lần lặp sau hoàn chỉnh hơn lần lặp trước. Ngoài ra, giai đoạn sau lại dựa trên kết quả của giai đoạn trước theo kiểu thác nước [7, 4].



Hình 1.1: Quá trình phát hiện tri thức trong CSDL

Sau đây sẽ trình bày cụ thể hơn từng giai đoạn của quá trình này:

### 1.2.1. Xác định vấn đề

Quá trình này mang tính định tính với mục đích xác định được lĩnh vực yêu cầu phát hiện tri thức và xây dựng bài toán tổng thể. Trong thực tế, các cơ sở dữ liệu được chuyên môn hoá và phân chia theo các lĩnh vực khác nhau. Với mỗi tri thức phát hiện được, có thể có giá trị cho lĩnh vực này nhưng lại không mang lại nhiều ý nghĩa đối với một lĩnh vực khác. Vì vậy, việc xác định bài toán giúp định hướng cho giai đoạn thu thập và tiền xử lý dữ liệu.

### 1.2.2. Thu thập và tiền xử lý dữ liệu

Trong quá trình thu thập dữ liệu cho bài toán, các cơ sở dữ liệu thu được thường chứa rất nhiều thuộc tính nhưng lại không đầy đủ, không thuần nhất, có

nhiều lỗi và có các giá trị đặc biệt. Nguyên nhân có thể là do ý kiến phát biểu của các chuyên gia không thống nhất, do các sai số khi đo đạc dữ liệu,... Vì vậy, giai đoạn thu thập và tiền xử lý dữ liệu trở nên rất quan trọng trong quá trình phát hiện tri thức từ cơ sở dữ liệu. Giai đoạn này thường chiếm từ 70% đến 80% giá thành của toàn bộ bài toán.

Giai đoạn thu thập và tiền xử lý dữ liệu được chia thành các công đoạn như: lựa chọn dữ liệu, làm sạch dữ liệu, làm giàu dữ liệu, mã hoá dữ liệu. Các công đoạn được thực hiện theo trình tự nhằm đưa ra một cơ sở dữ liệu thích hợp cho các giai đoạn sau. Tuy nhiên, tùy từng dữ liệu cụ thể mà quá trình trên được điều chỉnh cho phù hợp

#### ***1.2.2.1. Chọn lọc dữ liệu***

Đây là bước chọn lọc các dữ liệu liên quan trong các nguồn dữ liệu khác nhau. Các thông tin được chọn ra là những thông tin có nhiều liên quan đến lĩnh vực cần phát hiện tri thức đã xác định trong giai đoạn xác định vấn đề.

#### ***1.2.2.2. Làm sạch dữ liệu***

Dữ liệu thực tế, đặc biệt là những dữ liệu được lấy từ nhiều nguồn khác nhau thường không đồng nhất. Do đó, cần có biện pháp xử lý để thống nhất các dữ liệu thu được phục vụ cho khai phá. Giai đoạn làm sạch dữ liệu thường bao gồm các phép xử lý như: điều hoà dữ liệu, xử lý các giá trị khuyết, xử lý nhiễu và các ngoại lệ,...

#### ***1.2.2.3. Làm giàu dữ liệu***

Việc thu thập dữ liệu đôi khi không đảm bảo tính đầy đủ của dữ liệu. Một số thông tin rất quan trọng có thể thiếu hoặc không đầy đủ. Việc làm giàu dữ liệu chính là tìm cách bổ sung các thông tin có ý nghĩa và quan trọng cho quá trình khai phá dữ liệu sau này. Quá trình làm giàu dữ liệu cũng bao gồm việc tích hợp và chuyển đổi dữ liệu. Các dữ liệu từ nhiều nguồn khác nhau được tích hợp thành một kho thống nhất. Các khuôn dạng khác nhau của dữ liệu cũng được quy đổi, tính toán lại để đưa về một kiểu thống nhất, tiện cho quá trình phân tích. Đôi khi, một số thuộc tính mới cũng có thể được xây dựng dựa trên các thuộc tính cũ.

#### **1.2.2.4. Mã hoá**

Đây là giai đoạn mã hoá các phương pháp dùng để chọn lọc, làm sạch, làm giàu dữ liệu thành các thủ tục, chương trình hay các tiện ích nhằm tự động hoá việc kết xuất, biến đổi và di chuyển dữ liệu. Các hệ thống con đó có thể được thực thi định kỳ để làm tươi dữ liệu phục vụ cho việc phân tích.

#### **1.2.3. Khai phá dữ liệu**

Giai đoạn khai phá dữ liệu được bắt đầu sau khi dữ liệu đã được thu thập và xử lý. Trong giai đoạn này, công việc chủ yếu là xác định được bài toán khai phá dữ liệu, tiến hành lựa chọn các phương pháp khai phá thích hợp với dữ liệu có được và tách ra các tri thức cần thiết.

Thông thường, các bài toán khai phá dữ liệu bao gồm: các bài toán mang tính chất mô tả, đưa ra những tính chất chung nhất của dữ liệu, các bài toán khai phá, dự báo, bao gồm cả việc thực hiện các suy diễn dựa trên dữ liệu hiện có. Tùy theo từng bài toán xác định được mà ta lựa chọn các phương pháp khai phá dữ liệu cho phù hợp.

#### **1.2.4. Minh hoạ và đánh giá**

Các tri thức phát hiện được từ cơ sở dữ liệu cần được tổng hợp và biểu diễn dưới dạng gần gũi với người sử dụng như đồ thị, cây, bảng biểu, hay các luật, các báo cáo,... phục vụ cho các mục đích hỗ trợ quyết định khác nhau.

Do nhiều phương pháp khai phá có thể được áp dụng nên các kết quả có thể có nhiều mức độ tốt xấu khác nhau và việc đánh giá các kết quả thu được là rất cần thiết. Thông thường, các kết quả sẽ được tổng hợp, so sánh bằng các biểu đồ và được kiểm nghiệm, tinh lọc. Để đánh giá tri thức, người ta thường dựa vào các tiêu chí nhất định như:

- Tri thức phải đủ độ đáng quan tâm: thể hiện ở tính hữu dụng (useful), tính mới lạ (novel) của tri thức và quá trình trích rút không tầm thường.
- Tri thức phải đủ độ tin cậy.

Đây là công việc của các nhà chuyên gia, các nhà phân tích và ra quyết định.

### 1.2.5. Đưa kết quả vào thực tế

Các kết quả của quá trình phát hiện tri thức có thể được đưa vào ứng dụng trong các lĩnh vực khác nhau. Do các kết quả có thể là các dự báo hoặc các mô tả nên có thể đưa vào các hệ thống hỗ trợ ra quyết định nhằm tự động hoá quá trình này.

Như vậy, quá trình phát hiện tri thức từ cơ sở dữ liệu thường được thực hiện theo năm bước nêu trên. Tuy nhiên, trong quá trình khai thác, có thể thực hiện những cải tiến, nâng cấp cho phù hợp với từng ứng dụng cụ thể. Trong số các bước, tiền xử lý dữ liệu và khai phá dữ liệu hai bước rất quan trọng, chiếm phần lớn công sức và giá thành của toàn bộ bài toán. Việc lựa chọn các phương pháp thực hiện cụ thể cho quá trình tiền xử lý và khai phá dữ liệu phụ thuộc rất nhiều vào đặc điểm dữ liệu và yêu cầu của bài toán. Sau đây, ta sẽ xem xét cụ thể hơn quá trình khai phá dữ liệu.

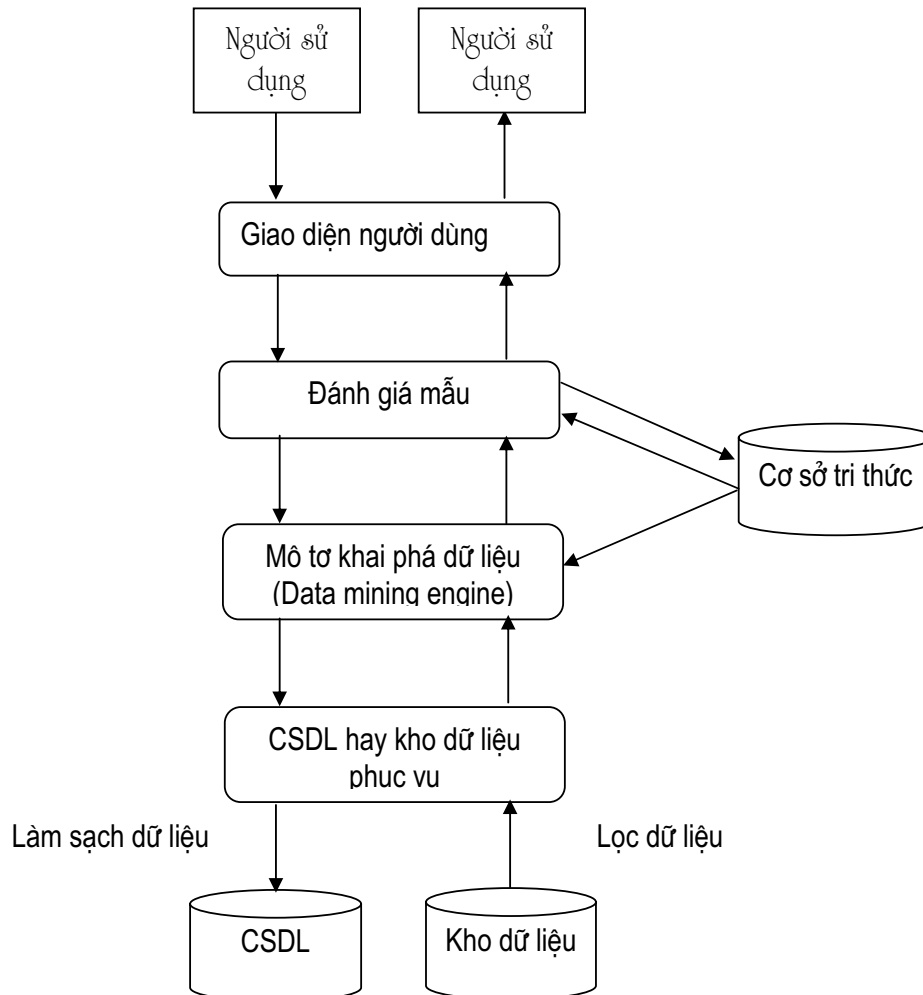
## 1.3. CÁC KỸ THUẬT KHAI PHÁ DỮ LIỆU

Ta đã biết, quá trình phát hiện tri thức, về nguyên lý, trải qua nhiều giai đoạn khác nhau mà khai phá dữ liệu chỉ là một giai đoạn trong quá trình đó. Tuy nhiên, đây lại là giai đoạn đóng vai trò chủ chốt và là giai đoạn chính tạo nên tính đa ngành của KDD.

### 1.3.1. Kiến trúc của hệ thống khai phá dữ liệu

Khai phá dữ liệu là một bước quan trọng trong quá trình phát hiện tri thức từ số lượng lớn dữ liệu đã lưu trữ trong các CSDL, kho dữ liệu hoặc các nơi lưu trữ khác. Bước này có thể tương tác lẫn nhau giữa người sử dụng hoặc cơ sở tri thức. Các mẫu đáng quan tâm được đưa đến cho người sử dụng hoặc lưu trữ như là tri thức mới trong cơ sở tri thức.

Kiến trúc của hệ thống khai phá dữ liệu có thể có các thành phần chính sau:



Hình 1.2: Kiến trúc của hệ thống khai phá dữ liệu

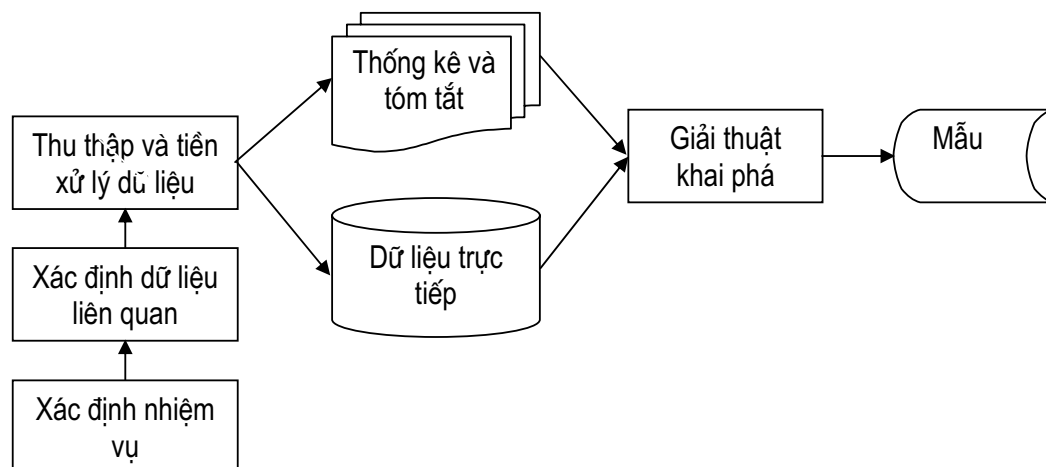
- CSDL, kho dữ liệu hay các kho lưu trữ khác: là một hoặc một tập các CSDL, kho dữ liệu, ... Các kỹ thuật làm sạch dữ liệu, tích hợp, lọc dữ liệu có thể thực hiện trên dữ liệu.
- CSDL hay kho dữ liệu phục vụ: là những dữ liệu có liên quan được lọc và làm sạch từ kho dữ liệu trên cơ sở yêu cầu khai phá dữ liệu của người dùng.
- Cơ sở tri thức: là lĩnh vực tri thức được sử dụng để hướng dẫn việc tìm hoặc đánh giá các mẫu kết quả tìm được.

- Mô tơ khai phá dữ liệu: bao gồm tập các modul chức năng để thực hiện các nhiệm vụ như mô tả đặc điểm, kết hợp, phân lớp, phân cụm dữ liệu, ...
- Modul đánh giá mẫu: thành phần này sử dụng các độ đo và tương tác với các modul khai phá dữ liệu để tập trung tìm các mẫu đáng quan tâm.
- Giao diện người dùng: cho phép người dùng tương tác với hệ thống trên cơ sở những truy vấn hay tác vụ, cung cấp các thông tin cho việc tìm kiếm.

### 1.3.2. Quá trình khai phá dữ liệu và giải thuật khai phá dữ liệu

#### 1.3.2.1. Quá trình khai phá dữ liệu

Các giải thuật khai phá dữ liệu thường được mô tả như những chương trình hoạt động trực tiếp trên tập dữ liệu. Quá trình khai phá dữ liệu được thể hiện bởi mô hình sau:



Hình 1.3: Quá trình khai phá dữ liệu

- Xác định nhiệm vụ: Xác định chính xác vấn đề cần được giải quyết
- Xác định dữ liệu liên quan: Trên cơ sở vấn đề cần được giải quyết, xác định các nguồn dữ liệu liên quan để có thể xây dựng giải pháp.
- Thu thập và tiền xử lý dữ liệu: Thu thập các dữ liệu có liên quan và xử lý chúng đưa về dạng sao cho giải thuật khai phá dữ liệu có thể hiểu được. ở đây có thể gặp một số vấn đề như: dữ liệu phải được sao ra nhiều bản (nếu được



chiết xuất vào các tệp), quản lý các tệp dữ liệu, phải lặp đi lặp lại nhiều lần toàn bộ quá trình (nếu mô hình dữ liệu thay đổi), ...

- Thống kê và tóm tắt dữ liệu, đồng thời kết hợp với các dữ liệu trực tiếp để làm đầu vào cho bước thực hiện giải thuật khai phá dữ liệu.
- Chọn thuật toán khai phá dữ liệu thích hợp và thực hiện việc khai phá dữ liệu để tìm được các mẫu có ý nghĩa. Với các nhiệm vụ khác nhau của khai phá dữ liệu, dạng của các mẫu chiết xuất được cũng khác nhau. Mẫu chiết xuất được có thể là một mô tả xu hướng, có thể là dưới dạng văn bản, một đồ thị mô tả các mối quan hệ trong mô hình,...

#### ***1.3.2.2. Các thành phần của giải thuật khai phá dữ liệu***

Giải thuật khai phá dữ liệu gồm ba thành phần chính:

- ***Biểu diễn mô hình:*** Mô hình được biểu diễn bằng một ngôn ngữ L để mô tả các mẫu có thể khai thác được. Nếu mô hình mô tả quá hạn chế thì sẽ không thể học được hoặc sẽ không có các mẫu tạo ra được một mô hình chính xác cho dữ liệu. Tuy nhiên, khả năng mô tả của mô hình càng lớn thì càng tăng mức độ nguy hiểm do bị học quá và làm giảm khả năng dự đoán của các dữ liệu chưa biết. Do đó, việc quan trọng là người phân tích dữ liệu và thiết kế giải thuật cần phải hiểu đầy đủ các giả thiết mô tả và cần phải diễn tả được các giả thiết mô tả nào được tạo ra từ luật nào.

- ***Đánh giá mô hình:*** Đánh giá xem một mẫu có đáp ứng được các tiêu chuẩn của quá trình phát hiện tri thức hay không. Việc đánh giá độ chính xác dự đoán được thực hiện dựa trên đánh giá chéo (cross validation). Đánh giá chất lượng liên quan đến độ chính xác dự đoán, độ mới, khả năng sử dụng, khả năng hiểu được của mô hình. Có thể sử dụng chuẩn thống kê và chuẩn logic để đánh giá mô hình.

- ***Phương pháp tìm kiếm:*** Phương pháp tìm kiếm gồm hai thành phần: tìm kiếm tham số và tìm kiếm mô hình.

- Trong tìm kiếm tham số, giải thuật cần tìm kiếm các tham số để tối ưu hoá các tiêu chuẩn đánh giá mô hình với các dữ liệu quan sát được và một miêu tả mô hình đã định trước.

- Tìm kiếm mô hình thực hiện giống như một vòng lặp qua phương pháp tìm kiếm tham số, miêu tả mô hình bị thay đổi tạo nên một họ các mô hình. Với mỗi một miêu tả mô hình, phương pháp tìm kiếm tham số được thực hiện để đánh giá chất lượng mô hình. Các phương pháp tìm kiếm mô hình thường sử dụng các phương pháp tìm kiếm heuristic vì kích thước của không gian tìm kiếm các mô hình thường ngăn cản các kỹ thuật tìm kiếm tổng thể.

### 1.3.3. Nhiệm vụ chính của khai phá dữ liệu

Đối với khai phá dữ liệu, có hai bài toán chính là:

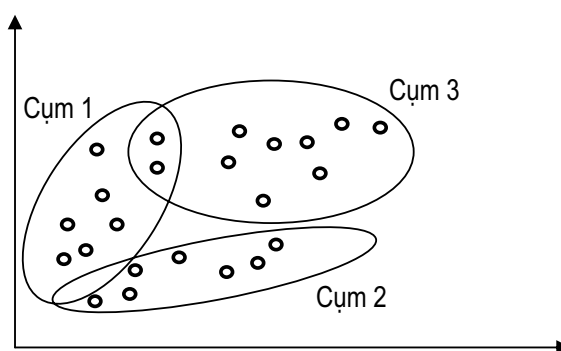
- Bài toán mô tả (description): Đưa ra mô hình biểu thị những tính chất chung nhất của dữ liệu mẫu.
- Bài toán khai phá dự báo (prediction): Suy diễn dựa trên dữ liệu mẫu hiện có để đưa ra một kết quả nào đó.

Như vậy, có thể coi mục đích chính của khai phá dữ liệu là mô tả và dự báo. Các mẫu được phát hiện nhằm vào hai mục đích này. Bài toán dự báo liên quan đến việc sử dụng các biến hoặc các trường trong CSDL để chiết xuất ra các mẫu, trên cơ sở đó dự đoán các giá trị chưa biết hoặc các giá trị tương lai của các biến đáng quan tâm. Bài toán mô tả tập trung vào việc tìm kiếm các mẫu mô tả dữ liệu có thể hiểu được cho các ứng dụng thực tế.

Để đạt được hai mục đích này, nhiệm vụ chính của khai phá dữ liệu bao gồm các vấn đề sau:

- Phân lớp (classification): Phân lớp tương ứng với việc xác lập một ánh xạ (hay phân loại) một tập dữ liệu vào một trong số các lớp đã xác định.
- Hồi quy (Regression): Hồi quy tương ứng với việc xác lập ánh xạ từ một tập dữ liệu vào một biến dự đoán có giá trị thực.
- Phân cụm (Clustering): Phân cụm nhằm ghép nhóm các đối tượng dữ liệu. Các đối tượng dữ liệu được coi là giống nhau, nếu chúng thuộc cùng một cụm và khác nhau nếu chúng thuộc các cụm khác nhau. Các cụm có thể tách rời nhau hoặc phân cấp hoặc gối lên nhau. Nghĩa là một đối tượng dữ liệu có thể vừa thuộc cụm này, vừa thuộc cụm kia. Quá trình nhóm các đối tượng thành các cụm được gọi là

phân cụm hay phân nhóm. Một ví dụ ứng dụng của khai phá dữ liệu có nhiệm vụ phân cụm là phát hiện tập những khách hàng có hành vi giống nhau trong cơ sở dữ liệu tiếp thị.



Hình 1.4: Kết quả của phân cụm

Hình 1.4 mô tả các mẫu của quá trình khai phá dữ liệu với nhiệm vụ phân cụm. Các mẫu là nhóm khách hàng được xếp vào ba nhóm gộp lên nhau. Những khách hàng ở cả hai cụm chứng tỏ khách hàng đó có thể thuộc hai trạng thái.

- Tóm tắt (summarization): liên quan đến các phương pháp tìm kiếm một mô tả tóm tắt cho một tập con dữ liệu.
- Mô hình hoá sự phụ thuộc (Dependency Modeling): Bao gồm việc tìm kiếm một mô hình mô tả sự phụ thuộc giữa các biến. Các mô hình phụ thuộc tồn tại dưới hai mức:
  - Mức cấu trúc, là mô hình xác định các biến nào là phụ thuộc cục bộ với nhau (thường ở dạng đồ hoạ).
  - Mức định lượng là mô hình xác định độ lớn của sự phụ thuộc theo một thước đo nào đó.
- Phát hiện thay đổi và sai lệch (Change and Deviation detection): Xác định những thay đổi đáng kể nhất trong dữ liệu từ các giá trị chuẩn đo được trước đó.

Rõ ràng, những nhiệm vụ khác nhau kể trên yêu cầu về số lượng và các dạng thông tin rất khác nhau. Do đó, tùy theo từng nhiệm vụ cụ thể, sẽ có những ảnh hưởng đến việc thiết kế và lựa chọn giải thuật khai phá dữ liệu.

### 1.3.4. Một số phương pháp khai phá dữ liệu phổ biến

#### 1.3.4.1. Phương pháp quy nạp

Có hai kỹ thuật chính để thực hiện là suy diễn và quy nạp.

- Suy diễn: nhằm rút ra thông tin là kết quả logic của các thông tin trong CSDL. Phương pháp suy diễn dựa trên những sự kiện chính xác để suy ra các tri thức mới từ các thông tin cũ. Mẫu chiết xuất theo kỹ thuật này thường là các luật suy diễn.

- Quy nạp: Phương pháp quy nạp suy ra thông tin được sinh ra từ cơ sở dữ liệu, có nghĩa là nó tự tìm kiếm, tạo mẫu và sinh ra tri thức chứ không phải bắt đầu với các tri thức đã biết trước. Các thông tin do phương pháp này mang lại là những thông tin hay tri thức cấp cao diễn tả về các đối tượng trong CSDL. Phương pháp này liên quan đến việc tìm kiếm các mẫu trong CSDL.

Phương pháp quy nạp thường được nói đến trong kỹ thuật cây quyết định và tạo luật.

#### 1.3.4.2. Cây quyết định và tạo luật

- Cây quyết định: là một dạng mô tả tri thức đơn giản nhằm phân các đối tượng dữ liệu thành một số lớp nhất định. Các nút của cây được gán nhãn là tên các thuộc tính, các cung được gán giá trị có thể của các thuộc tính, các lá miêu tả các lớp khác nhau. Các đối tượng được phân lớp theo các đường đi trên cây, qua các cung tương ứng với giá trị của thuộc tính của đối tượng tới lá.

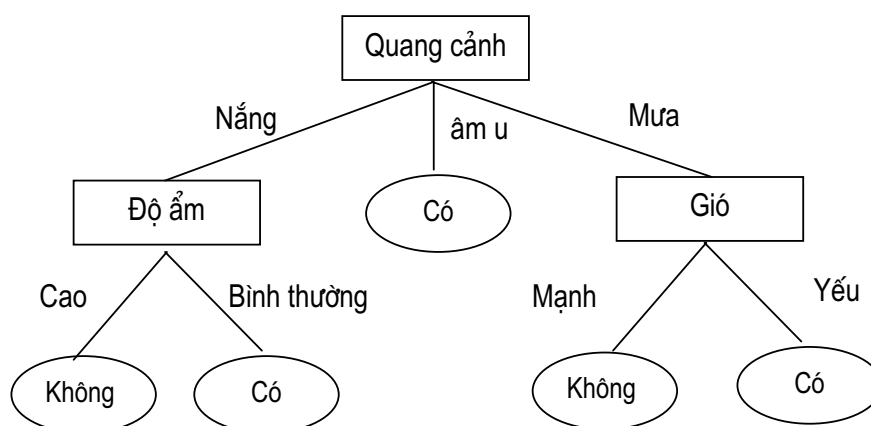
Ví dụ: Bảng dữ liệu học trong ví dụ quyết định đi chơi tennis:

Ngày	Quang cảnh	Nhiệt độ	Độ ẩm	Gió	Chơi tennis
D1	Nắng	Nóng	Cao	Yếu	Không
D2	Nắng	Nóng	Cao	Mạnh	Không
D3	âm u	Nóng	Cao	Yếu	Có
D4	Mưa	ấm áp	Cao	Yếu	Có
D5	Mưa	Lạnh	Bình thường	Yếu	Có

D6	Mưa	Lạnh	Bình thường	Mạnh	Không
D7	âm u	Lạnh	Bình thường	Mạnh	Có
D8	Nắng	ấm áp	Cao	Yếu	Không
D9	Nắng	Lạnh	Bình thường	Yếu	Có
D10	Mưa	ấm áp	Bình thường	Yếu	Có
D11	Nắng	ấm áp	Bình thường	Mạnh	Có
D12	âm u	ấm áp	Cao	Mạnh	Có
D13	âm u	Nóng	Bình thường	Yếu	Có
D14	Mưa	ấm áp	Cao	Mạnh	Không

Bảng 1.1: Dữ liệu học trong ví dụ quyết định đi chơi tennis

Từ bảng dữ liệu trên, người ta xây dựng được cây quyết định trợ giúp quyết định đi hay không đi chơi tennis như sau:



Hình 1.5: Cây quyết định đi chơi tennis

- **Tạo luật:** Các luật được tạo ra nhằm suy diễn một số mẫu dữ liệu có ý nghĩa về mặt thống kê. Các luật có dạng “Nếu P thì Q”, với P là mệnh đề đúng với một phần dữ liệu có trong CSDL, Q là mệnh đề dự đoán.

Cây quyết định và luật có ưu điểm là hình thức mô tả đơn giản, mô hình biểu diễn khá dễ hiểu đối với người sử dụng. Tuy nhiên, mô tả cây và luật chỉ có thể biểu diễn được một số chức năng, vì vậy chúng giới hạn về độ chính xác của mô hình.

#### 1.3.4.3. Phát hiện luật kết hợp

Phương pháp này nhằm phát hiện các luật kết hợp giữa các thành phần dữ liệu trong CSDL. Đầu ra của thuật toán khai phá dữ liệu là một tập luật kết mà mỗi luật có dạng:  $X \Rightarrow Y$  (nếu có X thì có Y). Kèm theo mỗi luật tìm được là các tham số độ hỗ trợ và độ tin cậy của luật. Độ hỗ trợ và độ tin cậy là hai độ đo chỉ sự đáng quan tâm, phản ánh sự hữu ích và sự chắc chắn của luật, chúng được tính theo công thức:

Độ hỗ trợ (Support) = Số bản ghi chứa X / Tổng số bản ghi.

Độ tin cậy (Confidence) = Số bản ghi chứa cả X và Y / Số bản ghi chứa X

**Ví dụ:** Phân tích CSDL bán hàng, người ta nhận được thông tin về những khách hàng mua máy tính đồng thời cũng có khuynh hướng mua phần mềm quản lý tài chính trong cùng một lần mua được mô tả trong luật kết hợp như sau:

*“Máy tính  $\Rightarrow$  Phần mềm quản lý”*

*[Độ hỗ trợ: 2%, độ tin cậy: 60%]*

Luật trên thể hiện có 2% trên tổng số các khách hàng đã mua máy tính, trong số những khách hàng mua máy tính, 60% cũng mua phần mềm quản lý.

Phát hiện các luật kết hợp là phải tìm tất cả các luật thỏa mãn ngưỡng độ tin cậy và độ hỗ trợ cho trước. Thuật toán tìm các luật kết hợp trước tiên phải đi tìm các tập mục thường xuyên, sau đó từ các tập mục thường xuyên tạo nên luật kết hợp.

#### 1.3.4.4. Phân nhóm và phân đoạn

Kỹ thuật phân nhóm và phân đoạn là những kỹ thuật phân chia dữ liệu sao cho mỗi phần hoặc mỗi nhóm sẽ giống nhau theo một tiêu chuẩn nào đó. Mối quan hệ thành viên của các nhóm có thể dựa trên mức độ giống nhau của các thành viên và từ đó xây dựng nên các luật ràng buộc giữa các thành viên trong nhóm. Một kỹ thuật phân nhóm khác là xây dựng các hàm đánh giá các thuộc tính của các thành phần như là hàm của các tham số của các thành phần. Phương pháp này được gọi là phương pháp phân hoạch tối ưu (optimal partitioning).

Mẫu đầu ra của quá trình khai phá dữ liệu dùng kỹ thuật này là các tập mẫu chứa các dữ liệu có chung những tính chất nào đó được phân tách từ CSDL. Khi các mẫu được thiết lập, chúng có thể được sử dụng để tái tạo các tập dữ liệu ở dạng dễ

hiều hơn, đồng thời cũng cung cấp các nhóm dữ liệu cho các hoạt động cũng như công việc phân tích. Đối với CSDL lớn, việc lấy ra các nhóm này là rất quan trọng.

#### ***1.3.4.5. Các phương pháp dựa trên mẫu***

Sử dụng các mẫu miêu tả từ CSDL để tạo nên một mô hình dự đoán các mẫu mới bằng cách rút ra các thuộc tính tương tự như các mẫu đã biết trong mô hình. Các kỹ thuật được sử dụng bao gồm phân lớp theo k láng giềng gần nhất (K\_nearest neighbour), các giải thuật hồi quy và các hệ thống suy diễn dựa trên tình huống (case based reasoning).

#### ***1.3.4.6. Mô hình phụ thuộc dựa trên đồ thị xác suất***

Các mô hình đồ thị xác định sự phụ thuộc xác suất giữa các sự kiện thông qua mối liên hệ trực tiếp theo các cung của đồ thị. Ở dạng đơn giản nhất, mô hình xác định những biến nào phụ thuộc nhau một cách trực tiếp. Mô hình phụ thuộc dựa trên đồ thị xác suất thường được sử dụng với các biến có giá trị rời rạc hoặc phân loại. Tuy nhiên, các mô hình này cũng được mở rộng cho một số trường hợp đặc biệt như mật độ Gaussian hoặc cho các biến có giá trị thực.

#### ***1.3.4.7. Mô hình học quan hệ***

Mẫu chiết suất được bằng các luật suy diễn và cây quyết định gắn chặt với mệnh đề logic, còn mô hình học quan hệ (còn gọi là lập trình logic quy nạp) sử dụng ngôn ngữ mẫu theo thứ tự logic trước (first – order logic) khá linh hoạt. Mô hình này có thể dễ dàng tìm ra công thức  $X=Y$ . Cho đến nay, hầu hết các nghiên cứu về các phương pháp đánh giá mô hình học quan hệ đều theo logic trong tự nhiên.

#### ***1.3.4.8. Khai phá dữ liệu văn bản (Text Mining)***

Khai phá dữ liệu văn bản phù hợp với việc tìm kiếm, phân tích và phân loại các dữ liệu văn bản không định dạng. Các lĩnh vực ứng dụng của khai phá dữ liệu văn bản như nghiên cứu thị trường, thu nhập, tình báo, .... Phương pháp này được sử dụng để phân tích câu trả lời cho các câu hỏi mở trong khảo sát thị trường, tìm kiếm các tài liệu phức tạp.

#### **1.3.4.9. Mạng nơon**

Mạng nơon là cách tiếp cận tính toán mới liên quan đến việc phát triển các cấu trúc toán học với khả năng học. Mạng nơon là kết quả của việc nghiên cứu mô hình học của hệ thần kinh con người. Mạng có thể đưa ra ý nghĩa từ các dữ liệu phức tạp hoặc không chính xác và có thể được sử dụng để chiết suất các mẫu và phát hiện ra các xu hướng phức tạp mà con người cũng như các kỹ thuật máy tính khác không thể phát hiện được.

Khi đề cập đến khai thác dữ liệu, người ta thường đề cập nhiều đến mạng nơon. Tuy mạng nơon có một số hạn chế gây khó khăn trong việc áp dụng và triển khai nhưng nó cũng có những ưu điểm đáng kể. Một trong số những ưu điểm đó là khả năng tạo ra các mô hình dự đoán có độ chính xác cao, có thể áp dụng được cho rất nhiều bài toán khác nhau đáp ứng được nhiệm vụ đặt ra của khai phá dữ liệu như phân lớp, phân nhóm, mô hình hoá, dự báo các sự kiện phụ thuộc vào thời gian, ....

#### **1.3.4.10. Giải thuật di truyền**

Giải thuật di truyền chính là sự mô phỏng lại quá trình tiến hoá di truyền trong tự nhiên. Một cách chính xác thì đó là giải thuật chỉ ra tập các cá thể được hình thành, ước lượng và biến đổi như thế nào. Cụ thể là các vấn đề như làm thế nào để lựa chọn các cá thể tái tạo và các cá thể nào sẽ bị loại bỏ, quá trình lai ghép và đột biến sẽ diễn ra như thế nào? Giải thuật cũng mô phỏng lại yếu tố gen trong nhiễm sắc thể sinh học trên máy tính để có thể giải quyết được các bài toán thực tế khác nhau.

Giải thuật di truyền là một giải thuật tối ưu hoá, được sử dụng rộng rãi trong việc tối ưu hoá các kỹ thuật khai phá dữ liệu trong đó có kỹ thuật mạng nơon. Sự liên hệ của giải thuật di truyền với các giải thuật khai phá là ở chỗ việc tối ưu hoá rất cần thiết cho quá trình khai phá dữ liệu, ví dụ như trong các kỹ thuật cây quyết định, tạo luật, ....

#### **❖ Vấn đề lựa chọn phương pháp:**

Qua phần trình bày trên, ta nhận thấy có rất nhiều phương pháp khai phá dữ liệu. Mỗi phương pháp có những đặc điểm riêng phù hợp với một lớp các bài toán,



với các dạng dữ liệu và miền dữ liệu nhất định. Hiện người ta vẫn chưa đưa ra được một tiêu chuẩn nào trong việc quyết định sử dụng phương pháp khai phá nào trong trường hợp nào thì hiệu quả.

Hầu hết các kỹ thuật khai phá dữ liệu đều còn mới mẻ với lĩnh vực kinh doanh. Hơn nữa, lại có rất nhiều kỹ thuật, mỗi kỹ thuật được sử dụng cho nhiều bài toán khác nhau. Vì vậy, trả lời cho câu hỏi “Dùng kỹ thuật nào?” là một vấn đề không đơn giản. Mỗi kỹ thuật đều có điểm mạnh và điểm yếu nhất định, nên vấn đề đối với người sử dụng là phải lựa chọn và áp dụng các kỹ thuật một cách thật đơn giản, dễ sử dụng để không cảm thấy những phức tạp vốn có của kỹ thuật đó.

### **1.3.5. Những ưu thế và khó khăn thách thức trong nghiên cứu và ứng dụng kỹ thuật khai phá dữ liệu**

#### ***1.3.5.1. Ưu thế của khai phá dữ liệu so với các phương pháp cơ bản***

Khai phá dữ liệu là lĩnh vực liên quan tới rất nhiều ngành học khác như: hệ CSDL, thống kê, hiển thị trực quan hoá,... Hơn nữa, tùy vào cách tiếp cận, khai phá dữ liệu còn có thể áp dụng một số kỹ thuật như mạng nơon, lý thuyết tập thô hoặc tập mờ, biểu diễn tri thức,... Tuy nhiên, khai phá dữ liệu có một số ưu điểm rõ rệt so với các phương pháp cơ bản khác, cụ thể như sau:

- So với phương pháp học máy, khai phá dữ liệu có lợi thế hơn ở chỗ nó có thể sử dụng các CSDL chứa nhiều, dữ liệu không đầy đủ hoặc biến đổi liên tục. Trong khi phương pháp học máy chủ yếu được áp dụng trong những CSDL đầy đủ, ít biến động và tập dữ liệu không quá lớn.
- Phương pháp hệ chuyên gia: phương pháp này khác với khai phá dữ liệu ở chỗ các ví dụ của chuyên gia thường ở mức chất lượng cao hơn nhiều so với dữ liệu trong CSDL và chúng chỉ bao hàm các trường hợp quan trọng. Hơn nữa, các chuyên gia sẽ xác nhận giá trị và tính hữu ích của các mẫu phát hiện được và như thế đòi hỏi phải có sự tham gia của con người trong việc phát hiện tri thức.
- Phương pháp thống kê là một trong những nền tảng lý thuyết của khai phá dữ liệu, nhưng khi so sánh chúng với nhau, có thể thấy phương pháp thống kê còn có một số điểm yếu mà khai phá dữ liệu đã khắc phục được:

- Các phương pháp thống kê chuẩn không phù hợp với các kiểu dữ liệu có cấu trúc trong rất nhiều các CSDL.
- Các phương pháp thống kê hoạt động hoàn toàn theo dữ liệu, nó không sử dụng tri thức sẵn có về lĩnh vực.
- Kết quả phân tích của thống kê có thể sẽ rất nhiều và khó có thể làm rõ được.
- Phương pháp thống kê cần có sự hướng dẫn của người dùng để xác định phân tích dữ liệu như thế nào và ở đâu.

#### ***1.3.5.2. Những vấn đề khó khăn thách thức***

Mặc dù khai phá dữ liệu là một kỹ thuật khai phá tri thức hiệu quả, nhưng cũng bộc lộ nhiều khó khăn. Những khó khăn đó chính là những thách thức lớn trong quá trình nghiên cứu và ứng dụng các kỹ thuật khai phá dữ liệu vào thực tế.

##### **➤ Các vấn đề về cơ sở dữ liệu:**

Đầu vào của hệ thống phát hiện tri thức chủ yếu là các dữ liệu thô trong CSDL. Những vấn đề phát sinh trong quá trình khai phá dữ liệu chính từ các nguyên nhân là dữ liệu trong thực tế thường động, không đầy đủ, lớn và bị nhiễu. Trong một số trường hợp, người ta không biết dữ liệu có chứa thông tin cần thiết cho việc khai thác hay không và làm thế nào để giải quyết sự dư thừa những thông tin không thích hợp.

- Vấn đề dữ liệu lớn: Các CSDL thông thường là rất lớn, với hàng trăm trường và bảng có hàng triệu bản ghi. Khi đó kích thước lưu trữ cũng rất lớn, hàng gigabytes thậm chí terabytes. Do đó, làm tăng không gian tìm kiếm, tăng quá trình suy diễn, đồng thời cũng làm tăng khả năng giải thuật khai phá dữ liệu tìm được các mẫu giả. Phương pháp khắc phục vấn đề này hiện nay là đưa ra một ngưỡng cho CSDL, lấy mẫu, các phương pháp xấp xỉ, xử lý song song, giảm kích thước tác động của bài toán và sử dụng các tri thức đã biết trước để xác định các biến không phù hợp.

- Vấn đề dữ liệu động: Hầu hết các CSDL có nội dung thay đổi liên tục theo thời gian và việc khai phá dữ liệu bị ảnh hưởng bởi thời điểm quan sát. Việc thay đổi dữ liệu nhanh chóng có thể làm cho các mẫu khai phá được trước đó mất giá trị. Hơn

nữa, các biến trong CSDL của ứng dụng có thể bị thay đổi, bị xóa hoặc tăng lên theo thời gian. Vấn đề này được giải quyết bằng giải pháp tăng trưởng để nâng cấp các mẫu và coi những thay đổi như là cơ hội để khai thác bằng cách sử dụng nó để tìm kiếm các mẫu bị thay đổi.

- Vấn đề các trường không phù hợp: Một đặc điểm quan trọng khác là tính không thích hợp của dữ liệu, nghĩa là dữ liệu trở thành không thích hợp với mục tiêu trọng tâm hiện tại của việc khai phá. Một khía cạnh khác đôi khi cũng liên quan đến độ phù hợp là tính ứng dụng của một thuộc tính đối với một tập con của CSDL.

- Vấn đề các trường hay các giá trị bị thiếu: Một quan sát không đầy đủ của CSDL có thể làm cho dữ liệu có giá trị bị xem như là có lỗi. Việc quan sát CSDL phải phát hiện được toàn bộ các thuộc tính có thể dùng để khai phá dữ liệu trong bài toán. Giả sử ta có các thuộc tính để phân biệt các tình huống đáng quan tâm, nếu chúng không thể hiện được điều đó thì có nghĩa là đã có lỗi trong dữ liệu. Đây cũng là vấn đề thường xảy ra trong CSDL kinh doanh, các thuộc tính quan trọng có thể bị thiếu dữ liệu, không sẵn sàng cho việc khai phá dữ liệu.

- Độ nhiễu và không chắc chắn: Độ nhiễu của dữ liệu (độ chính xác, dung sai, ...) cũng là một nhân tố ảnh hưởng đến quá trình khai phá dữ liệu.

- Mối quan hệ phức tạp giữa các trường: các thuộc tính hoặc các giá trị dữ liệu có cấu trúc phân cấp, các mối quan hệ giữa các thuộc tính để diễn tả tri thức về nội dung của CSDL dẫn tới các giải thuật phải có khả năng khai phá một cách hiệu quả các dữ liệu này.

➤ **Một số vấn đề khác:**

- Quá phù hợp: Khi một thuật toán tìm kiếm các tham số tốt nhất cho một mô hình nào đó sử dụng một tập dữ liệu hữu hạn, có thể xảy ra tình trạng “quá độ”, nghĩa là chỉ phù hợp với một tập dữ liệu mà không có khả năng đáp ứng với các dữ liệu lạ. Điều đó làm cho mô hình hoạt động rất kém với các dữ liệu thử. Có thể khắc phục bằng cách đánh giá chéo, thực hiện theo nguyên tắc nào đó hoặc sử dụng các biện pháp thống kê khác.

- Khả năng biểu đạt mẫu: trong rất nhiều ứng dụng, điều quan trọng là những mẫu khai thác được phải càng dễ hiểu đối với con người càng tốt. Vì vậy, các giải

pháp thường là diễn tả dưới dạng đồ hoạ, xây dựng cấu trúc luật với các đồ thị có hướng, biểu diễn bằng ngôn ngữ tự nhiên và các kỹ thuật khác nhằm biểu diễn tri thức và dữ liệu.

- Tương tác với người sử dụng và các tri thức sẵn có: rất nhiều công cụ và phương pháp khai phá dữ liệu không thực sự tương tác với người dùng và không dễ dàng kết hợp cùng với các tri thức đã biết trước đó. Việc sử dụng tri thức miền là rất quan trọng trong khai phá dữ liệu. Đã có nhiều biện pháp nhằm khắc phục vấn đề này như sử dụng CSDL suy diễn để phát hiện tri thức, sau đó sử dụng những tri thức phát hiện được để hướng dẫn cho việc tìm kiếm khai phá dữ liệu hoặc sử dụng sự phân bố xác suất dữ liệu trước đó như một dạng mã hoá dữ liệu có sẵn.

### ❖ Kết luận chương 1

Quá trình phát hiện tri thức trong CSDL là quá trình rút ra những tri thức có ích, tiềm tàng trong CSDL. Quá trình phát hiện tri thức, về nguyên lý, trải qua nhiều giai đoạn khác nhau trong đó, khai phá dữ liệu là giai đoạn quan trọng nhất, đóng vai trò chủ chốt và là giai đoạn chính tạo nên tính đa ngành của KDD. Nhiệm vụ của khai phá dữ liệu là khám phá các mẫu có ích từ nguồn dữ liệu, trong đó, dữ liệu có thể được lưu trữ trong các CSDL, kho dữ liệu. Chương này cũng trình bày các nhiệm vụ chính của khai phá dữ liệu, các phương pháp khai phá dữ liệu cũng như các vấn đề thách thức trong nghiên cứu và áp dụng kỹ thuật khai phá dữ liệu vào thực tế.

Trong các phương pháp khai phá dữ liệu đã giới thiệu, mạng nơon và giải thuật di truyền là các kỹ thuật khai phá đang được quan tâm nghiên cứu mạnh mẽ. Chương sau sẽ trình bày chi tiết hơn về kỹ thuật khai phá dữ liệu dùng mạng nơon và giải thuật di truyền.

## **CHƯƠNG 2:**

# **KỸ THUẬT KHAI PHÁ DỮ LIỆU SỬ DỤNG MẠNG NƠON VÀ GIẢI THUẬT DI TRUYỀN**

## **2.1. MẠNG NƠON TRONG KHAI PHÁ DỮ LIỆU**

Khi đề cập đến khai thác dữ liệu, người ta thường đề cập nhiều đến mạng nơon. Tuy mạng nơon có một số hạn chế gây khó khăn cho quá trình áp dụng và triển khai, nhưng nó cũng có những ưu điểm đáng kể. Một trong số các ưu điểm phải kể đến là mạng có khả năng tạo ra các mô hình dự đoán có độ chính xác cao, có thể áp dụng cho rất nhiều loại bài toán khác nhau, đáp ứng được các nhiệm vụ đặt ra của khai phá dữ liệu như phân lớp, phân nhóm, mô hình hoá, dự báo các sự kiện phụ thuộc thời gian,....

### **2.1.1. Khái niệm mạng nơon**

Mạng nơon nhân tạo (Artificial Neural Network - ANN) là hệ thống được xây dựng mô phỏng theo các chức năng của một mạng nơon sinh học nói chung, hay mạng nơon sinh học của con người nói riêng. Trong luận văn này, khi nói đến mạng nơon có nghĩa là mạng nơon nhân tạo, bởi vì trong thực tế, mạng nơon sinh học (Biological Neural Network - BNN) có cấu tạo phức tạp hơn nhiều so với mạng nơon nhân tạo mà ta đề cập đến. Thực chất, mạng nơon nhân tạo là các mô hình toán học mà con người xây dựng nên. Cho đến nay, chưa có một định nghĩa tổng quát nào về mạng nơon, song phần lớn những nhà nghiên cứu trong lĩnh vực này đều thống nhất với khái niệm:

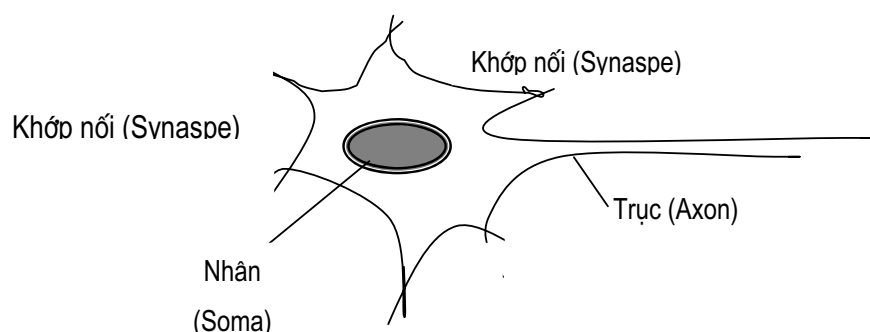
*Mạng nơon là một hệ thống gồm nhiều phần tử xử lý đơn giản gọi là các nơon được liên kết với nhau và cùng hoạt động song song. Tính năng hoạt động của mạng phụ thuộc vào cấu trúc mạng, trọng số liên kết giữa các nơon và quá trình xử*

lý bên trong các nơon. Ngoài chức năng xử lý, hệ thống còn có khả năng học số liệu và tổng quát hoá từ các số liệu đã học.

Chúng ta sẽ lần lượt phân tích mô hình nơon sinh học, sau đó là mô hình nơon nhân tạo để dễ dàng thấy được sự tương quan này, đồng thời hiểu rõ hơn về mạng nơon nhân tạo.

### 2.1.2. Nơon sinh học và mạng nơon sinh học

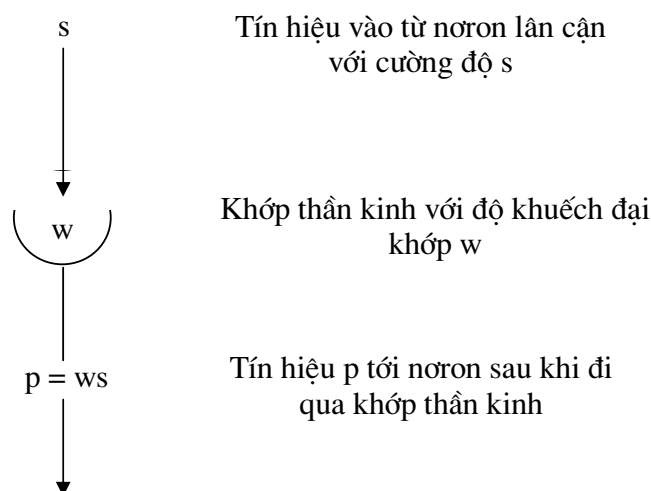
Hệ thần kinh con người có khoảng  $10^{10}$  tế bào thần kinh được gọi là các nơon, mỗi nơon có thể liên kết với  $10^4$  nơon khác thông qua các khớp nối [12].



Hình 2.1: Cấu tạo của nơon

Mỗi nơon gồm có ba phần: thân nơon 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 nơon khác, nhân nơon sẽ sinh ra tín hiệu ở đầu ra của nơon và truyền tới các nơon khác được nối với đầu ra qua trục.

Độ lớn của các tín hiệu vào có thể bị thay đổi khi được truyền qua các khớp thần kinh có trên các nhánh thần kinh vào. Tỷ lệ biến đổi tín hiệu ở khớp thần kinh được gọi là độ khuếch đại khớp và được gọi là các trọng số trong các nơon nhân tạo.



Hình 2.2: Thu nhận tín hiệu trong nơon

Theo các nghiên cứu về sinh học, chức năng của hệ thần kinh không phụ thuộc nhiều vào vai trò của từng nơon đơn lẻ mà phụ thuộc vào cách mà toàn bộ các nơon được nối với nhau, gọi là *mạng nơon sinh học* [12].

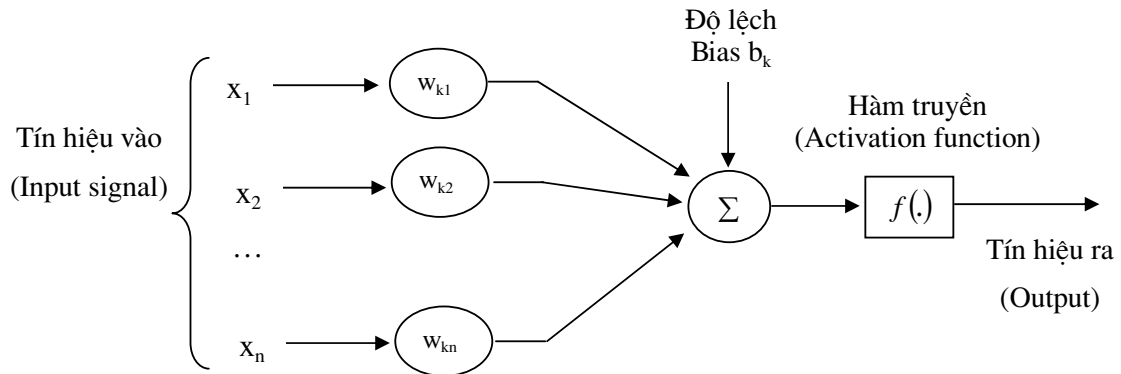
Tất cả các đặc điểm trên đều được vận dụng một cách triệt để trong việc xây dựng một mạng nhân tạo nhằm tạo ra một mạng nơon giống với mạng nơon sinh học nhất.

### 2.1.3. Mô hình và quá trình xử lý trong nơon nhân tạo

#### 2.1.3.1. Nơon nhân tạo

Giống như nơon sinh học, mỗi nơon nhân tạo được nối với các nơon khác và nhận tín hiệu từ chúng với các trọng số liên kết.

Một nơon nhân tạo phản ánh các tính chất cơ bản của nơon sinh học được mô phỏng trong hình 2.3.



Hình 2.3: Mô hình của một nơon nhân tạo

+ Đầu vào của nơon gồm  $n$  tín hiệu  $x = (x_1, x_2, \dots, x_n)$ , đầu ra là tín hiệu  $y = (y_1, y_2, \dots, y_m)$ .

+ Một tập các khớp nối và trọng số tương ứng  $w_{ki}$ , tín hiệu vào  $x_i$  của khớp nối thứ  $i$  của nơon  $k$  được nhân với trọng số  $w_{ki}$ .

+ Một bộ cộng  $\Sigma$  thực hiện trên các trọng số của các khớp nối thường được gọi là bộ kết hợp tuyến tính.

+ Một hàm chuẩn khống chế giá trị đầu ra của mạng nơon được gọi là hàm truyền hay hàm kích hoạt. Thông thường, tín hiệu đầu ra của một nơon trong khoảng  $[0, 1]$  hoặc  $[-1, 1]$ .

Trạng thái bên trong của nơon 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  $f$

Như vậy, mô hình toán học của nơon nhân tạo  $k$  tính toán tại thời điểm  $t$  như sau:

$$net(t) = \sum_{i=1}^n w_{ki} x_i(t) + b_k \quad y_k(t) = f\left(\sum_{i=1}^n w_{ki} x_i(t) + b_k\right)$$

Trong đó: là tín hiệu tổng hợp đầu vào,

$b_k$  là độ lệch bias.

Đầu ra thường được ký hiệu là  $out = y(t) = f(net)$

Tín hiệu vào được xử lý nhờ hàm kích hoạt (activation function) hay còn gọi là hàm truyền (transfer function) để tạo tín hiệu ra, tín hiệu ra sẽ được truyền đi nếu khác 0. Tóm lại, có thể xem nơon là một hàm phi tuyến nhiều đầu vào và một đầu ra.



### 2.1.3.2. Hàm truyền trong nơron

Cấu trúc của mạng nơron chủ yếu được đặc trưng bởi loại của các nơron và mối liên hệ xử lý thông tin giữa chúng. Về cấu trúc của nơron, chủ yếu người ta quan tâm tới cách tổng hợp các tín hiệu vào, ngưỡng tại mỗi nơron và các hàm truyền.

Hàm truyền xác định mức độ liên kết bên trong các nơron. Hàm truyền có nhiệm vụ tạo mức độ kích thích của nơron, từ đó sẽ làm hưng phấn hoặc ức chế các nơron khác trong mạng.

Trong lý thuyết mạng nơron, phép tổng hợp tín hiệu đầu vào của nơron  $i$  có  $m$  tín hiệu đầu vào  $x_j$  thường được ký hiệu:

$$net_i = \sum_{j=1}^m w_{ij} x_j; \quad w_{ij} = (w_{i1}, w_{i2}, \dots, w_{im})$$

Tín hiệu ra tại nơron  $i$  thường ký hiệu là  $out_i$  hoặc  $f_i$ , được tính theo công thức sau với  $f$  là hàm truyền:

$$out_i(t) = f(net_i(t))$$

Có nhiều hàm truyền khác nhau được sử dụng trong từng trường hợp cụ thể, các hàm truyền nói chung nên thỏa mãn các tính chất sau:

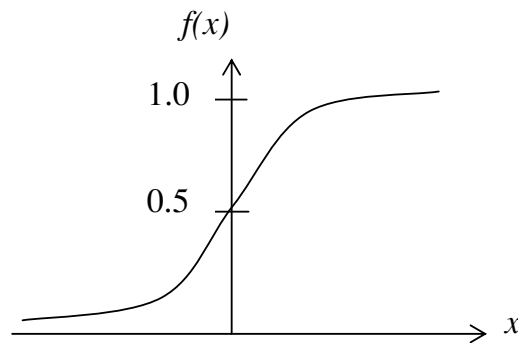
- ◆ Bị chặn:  $|f(x)| \leq M, \forall x$
- ◆ Đơn điệu tăng:  $f(x_1) > f(x_2), \forall x_1 > x_2$
- ◆ Khả vi liên tục:  $f(x)$  có đạo hàm  $f'(x)$  và  $f'(x)$  là hàm liên tục

Trong thực tế, khi xét các nơron, chúng chỉ có thể có hai trạng thái là bị kích hoạt hoặc không bị kích hoạt. Nghĩa là tín hiệu ra một của nơron cần phải đảm bảo sao cho có thể nhận biết được nơron đó có bị kích hoạt hay không. Vì lý do đó, hàm truyền phải thỏa mãn điều kiện tín hiệu ra cuối cùng của nơron phải liên tục và nằm trong một giới hạn xác định (có thể là giữa 0 và 1). Có một số dạng hàm truyền thường được sử dụng sau:

- Hàm ranh giới cứng (Hard – limiter):  $f(x) = \begin{cases} 1, & \text{if } (x \geq \theta) \\ 0, & \text{if } (x < \theta) \end{cases}$
- Hàm ranh giới cứng đối xứng:  $f(x) = \begin{cases} 1, & \text{if } (x \geq \theta) \\ -1, & \text{if } (x < \theta) \end{cases}$

➤ Hàm Gauss:  $f(x) = e^{-x^2}$

➤ Hàm Sigmoidal hay hàm logistic (còn gọi là hàm chữ S):  $f(x) = \frac{1}{1 + e^{-x}}$



Hình 2.4: Hàm Sigmoidal

Hàm Sigmoidal là hàm thường được sử dụng nhiều nhất trong các loại mạng nơron, bởi giá trị của hàm là liên tục trong khoảng (0,1). Tín hiệu ra của hàm có hai trạng thái ổn định và một vùng chuyển đổi. Nơron có hàm kích hoạt sigmoidal sẽ sinh giá trị thực bất kỳ giữa giá trị lớn nhất 1.0 và giá trị nhỏ nhất 0. Output dạng sigmoidal có giá trị  $> 0.8$  được coi như output kích hoạt. Nếu có giá trị  $< 0.2$  coi như giá trị không kích hoạt. Các giá trị output nằm trong khoảng 0.2 đến 0.8 là trong vùng chuyển đổi. Khi Net có giá trị âm lớn, hàm sẽ trả lại giá trị 0, khi Net có giá trị dương lớn, hàm sẽ trả lại giá trị 1, đó là các giá trị thường được dùng để biểu diễn các kết quả đúng, sai.

Hàm sigmoidal có thể dùng để phát hiện các đặc trưng của dữ liệu và dùng cho mục đích phân lớp dữ liệu.

#### 2.1.4. Cấu trúc và phân loại mạng nơron

Khi xét mạng nơron sinh học người ta nhận thấy: các tín hiệu do các nơron tạo ra rất giống nhau và hầu như không thể phân biệt được cho dù đó là nơron của loại sinh vật nào. Rõ ràng cường độ tín hiệu được tạo ra bởi các nơron có thể khác nhau phụ thuộc vào cường độ kích thích nhưng bề ngoài của các tín hiệu lại rất

giống nhau. Điều đó chứng tỏ rằng việc thực hiện chức năng của bộ não không phụ thuộc quá nhiều vào vai trò của một nơon đơn lẻ mà phụ thuộc vào toàn bộ hệ thống nơon. Nghĩa là phụ thuộc vào cách liên kết giữa các nơon, hay có thể nói việc thực hiện các chức năng phụ thuộc vào cấu trúc của mạng nơon.

Trong mô hình mạng nơon nhân tạo, các nơon được nối với nhau bởi các liên kết nơon, mỗi liên kết có một trọng số đặc trưng cho đặc tính kích hoạt hay ức chế giữa các nơon. Đồng thời, các nơon 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).

➤ Lớp vào: Các nút trong lớp vào gọi là các nút vào, chúng mã hoá mẫu được đưa vào mạng xử lý. Các nơon vào không xử lý thông tin, chỉ phân tán thông tin cho nút khác (trên biểu đồ chúng được vẽ khác các nút ẩn và các nút ra để phân biệt giữa các nút có xử lý và không xử lý thông tin)

➤ Lớp ẩn: Các nơon trong lớp ẩn gọi là các nút ẩn vì chúng không thể quan sát được trực tiếp. Chúng tạo thành các mô hình toán học phi tuyến cho mạng.

➤ Lớp ra: Các nơon trong lớp này gọi là các nút ra, chúng có nhiệm vụ đưa thông tin ra thích nghi mẫu mã người sử dụng cần.

Một mạng được gọi là kết nối đầy đủ nếu tất cả các nút của một lớp được nối với tất cả các nút của lớp kế liên nó. Có nhiều loại kết nối khác nhau:

- Kết nối liên lớp là kết nối giữa các nút trong các lớp khác nhau
- Kết nối trong lớp là kết nối giữa các nút trong cùng một lớp.
- Tự kết nối là kết nối từ một nút tới chính nó.
- Kết nối siêu lớp là kết nối giữa các lớp cách nhau (không kề nhau).

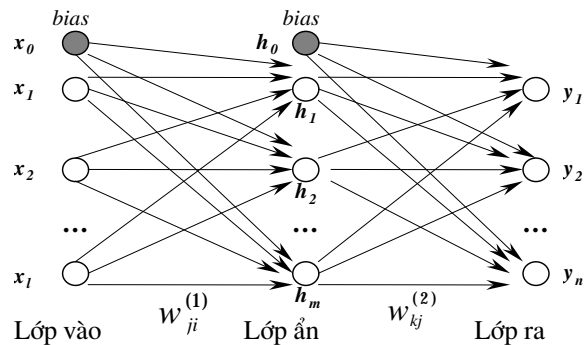
Một kết nối bậc cao là một kết nối với nhiều nút đầu vào. Số các nút đầu vào xác định bậc kết nối và bậc kết nối của mạng là bậc của kết nối bậc cao nhất.

#### **2.1.4.1. Phân loại mạng nơon**

Một cách hình thức, có thể biểu diễn mạng nơon như một đồ thị có hướng  $G = (N, A)$ . Trong đó tập đỉnh  $N$  biểu diễn các phần tử xử lý, tập các cung  $A$  biểu diễn liên kết giữa các phần tử xử lý, chiều của cung chỉ hướng của tín hiệu xử lý.

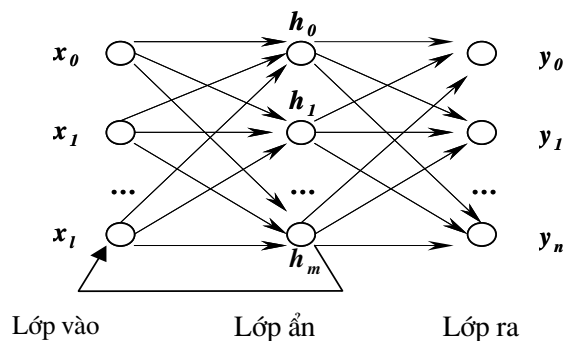
❖ Phân loại theo kiểu liên kết nơon:

➤ Mạng nơon truyền thẳng (feed – forward Neural Network): Trong mạng, các liên kết nơon chỉ đi theo một hướng từ lớp vào đến lớp ra, không tạo thành chu trình với các đỉnh là các nơon, các cung là các liên kết giữa chúng [10].



Hình 2.5: Mạng nơon truyền thẳng nhiều lớp (Feed-Forward Neural Network)

➤ Mạng hồi quy: cho phép các liên kết nơon tạo thành chu trình, có thông tin được xử lý theo hai chiều. Vì các thông tin ra của các nơon được truyền lại cho các nơon đã góp phần kích hoạt chúng nên mạng hồi quy còn có khả năng lưu giữ trạng thái trong của nó dưới dạng các ngưỡng kích hoạt ngoài các trọng số liên kết nơon [10].



Hình 2.6: Mạng hồi quy (Recurrent Neural Network)

➤ Mạng kết nối đối xứng và không đối xứng: Mạng kết nối đối xứng là mạng thoả mãn nếu 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 là bằng nhau:  $w_{ji} = w_{ij}$ . Mạng không thoả mãn điều kiện trên là kết nối không đối xứng.

❖ Phân loại theo số lớp:

Mạng chỉ gồm một lớp vào và một lớp ra gọi là mạng đơn lớp hay mạng một lớp. Mạng có từ một lớp ẩn trở lên được gọi là mạng đa lớp hay mạng nhiều lớp. Một mạng đa lớp được gọi là mạng  $n$  lớp với  $n$  là tổng số lớp ẩn và lớp ra.

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

### 2.1.5. Học và lan truyền trong mạng

#### 2.1.5.1. Học và tổng quát hoá

Mạng nơon thực hiện hai chức năng quan trọng là học và tổng quát hoá. Học là quá trình hiệu chỉnh các tham số và các trọng số liên kết trong mạng để tối thiểu hoá sai số với vectơ đầu vào cho trước. Quá trình học dừng khi mạng thoả mãn một tiêu chuẩn dừng nào đó, chẳng hạn 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.

Tổng quát hoá 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.

Bài toán học có thể được mô tả như sau: Cho tập mẫu  $(X_i, Y_i)$  với  $X_i$  và  $Y_i$  là hai véc tơ trong không gian một hoặc nhiều chiều, cần xác định bộ trọng số  $W_0$  trên không gian tham số để computer  $(X_i, W_0) = Y_i$ .

Quá trình học được thực hiện theo hai bước: Xác định hàm giá trị trên các tham số và tối thiểu hoá tham số trong không gian của các tham số.

Học chia thành hai loại: học tham số và học cấu trúc.

- **Học tham số:** Là quá trình xác định một tập hợp tham số  $W_0$  là các trọng số tốt nhất với một cấu trúc mạng cố định. Để làm được điều này cần xây dựng một hàm giá dựa trên tập dữ liệu  $T_{\text{train}}$  và tập trọng số  $W$ . Hàm giá có thể là một hàm khả vi bất kỳ có tính chất đạt đến cực tiểu khi các đầu ra  $O_i$  đúng bằng đầu ra lý tưởng  $Y_i$  của tập mẫu. Có thể xây dựng hàm giá dưới dạng  $L_n$  – norm như sau:

$$E = \frac{1}{p} \sum_i (y_i - O_i)^p \text{ với } 1 \leq p \leq \infty$$

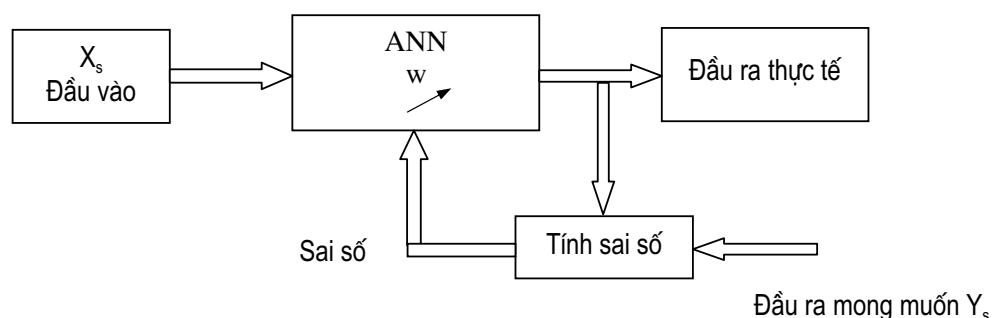
Với bộ tham số này, có thể áp dụng một giải thuật tìm kiếm nào đó trên không gian  $R^m$  của tập trọng số. Nếu thu được kết quả tốt với một cực tiểu toàn cục, ta sẽ có một bộ tham số tốt nhất cho mạng.

- **Học cấu trúc:** Với học tham số ta giả định rằng mạng có một cấu trúc cố định. Việc học cấu trúc của mạng truyền thẳng gắn với yêu cầu tìm ra số lớp của mạng  $L$  và số nơon trên mỗi lớp  $n_i$ . Tuy nhiên, với các mạng hồi quy còn phải xác định thêm các tham số ngưỡng  $\theta$  của các nơon trong mạng. Một cách tổng quát là phải xác định bộ tham số  $P = (L, n_1, \dots, n_l, \theta_1, \dots, \theta_k)$ .

Các kỹ thuật học của mạng Nơon chỉ ra cách chỉnh sửa các trọng số liên kết mạng khi một mẫu học được đưa vào mạng. Sau đây sẽ trình bày cụ thể về các kỹ thuật học [3]:

#### a. Học có giám sát

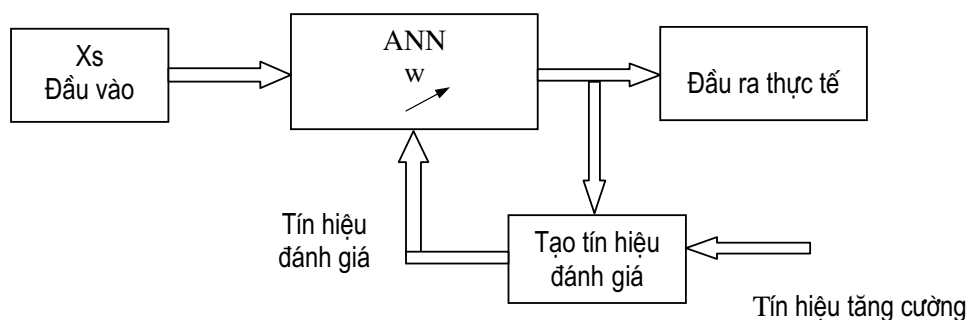
Với phương pháp học có giám sát hay học có thầy (supervised learning), mạng được cung cấp một tập mẫu học  $\{(X_s, Y_s)\}$  theo nghĩa  $X_s$  là các tín hiệu vào, thì kết quả ra đúng của hệ phải là  $Y_s$ . Ở mỗi lần học, véc tơ tín hiệu vào  $X_s$  được đưa vào mạng, sau đó so sánh sự sai khác giữa các kết quả ra đúng  $Y_s$  với kết quả tính toán qua mạng  $out_s$ . Sai số này sẽ được dùng để hiệu chỉnh lại các trọng số liên kết trong mạng. Quá trình cứ tiếp tục cho đến khi thỏa mãn một tiêu chuẩn nào đó. Có hai cách sử dụng tập mẫu học: hoặc dùng các mẫu lần lượt, hết mẫu này đến mẫu khác, hoặc sử dụng đồng thời tất cả các mẫu.



Hình 2.7: Sơ đồ học tham số có giám sát

### b. Học tăng cường

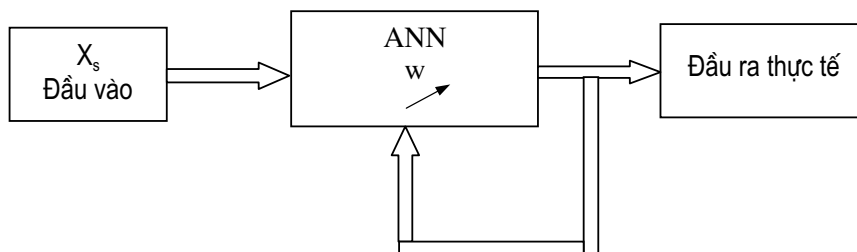
Ta thấy trong kỹ thuật học có giám sát, các vector đầu ra được biết một cách chính xác, nhưng trong một số trường hợp có ít thông tin, chẳng hạn chỉ có thể nói là mạng sinh Output quá lớn hoặc chỉ đúng khoảng 40%. Khi đó chỉ có một tín hiệu đánh giá là “True” hoặc “False” quay lại mạng, các thủ tục học đó gọi là thủ tục học tăng cường.



Hình 2.8: Sơ đồ học tăng cường

### c. Học không giám sát

Trong phương pháp học không giám sát (unsepervised learning), đầu ra mong muốn của mạng 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. Hệ thống học không giám sát phân chia các mẫu vào các nhóm hoặc các lớp quyết định bằng cách chọn các nơon “chiến thắng” và thay đổi các trọng số tương ứng của chúng. Thông thường, việc học không giám sát dùng nhiều tham số hơn kỹ thuật học có giám sát.



Hình 2.9: Sơ đồ học không giám sát

Như vậy, giải thuật học là giải thuật xuất phát từ một tập mẫu, qua quá trình huấn luyện để tìm ra bộ trọng số liên kết giữa các nơon, có thể mô tả tổng quát như sau:

**Đầu vào:** Một tập mẫu gồm  $n$  phần tử.

**Đầu ra:** Cấu trúc mạng và bộ trọng số các liên kết nơon

**Giải thuật:**

1. Khởi tạo trọng số của mạng, đặt  $i = 1$ ;
2. Đưa mẫu  $i$  vào lớp vào của mạng;
3. Sử dụng thuật toán lan truyền, nhận được giá trị các nút ra.

Nếu giá trị đầu ra của mạng đạt yêu cầu hoặc thỏa mãn tiêu chuẩn dừng thì kết thúc.

4. Sửa đổi trọng số bằng luật học của mạng;
  5. Nếu  $i = n$  thì đặt lại  $i = 1$ , nếu không thì tăng  $i$  lên 1:  $i = i + 1$
- Quay lại bước 2.

Có nhiều tiêu chuẩn dừng quá trình học, chẳng hạn:

- Chuẩn lỗi  $E$  nhỏ hơn một ngưỡng cho trước:  $E < \theta$ .
- Các trọng số của mạng không thay đổi nhiều sau khi hiệu chỉnh:

$$|w_{ij}^{new} - w_{ij}^{old}| < \theta.$$

- Việc lặp bị bão hoà, tức là số lần lặp vượt quá một ngưỡng  $N$  cho trước.

#### 2.1.5.2. Lan truyền trong mạng

Mạng nơon 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 của quá trình lan truyền.

Giải thuật lan truyền được mô tả như sau:

**Đầu vào:** Một tập tín hiệu vào

**Đầu ra:** Kết quả ra tương ứng với tập tín hiệu vào

**Giải thuật:**

1. Đưa tập tín hiệu vào vào lớp vào của mạng.
2. Tính mức tích cực của các nút trong mạng.



3. Với mạng truyền thẳng: Nếu mức tích cực của nút ra đã biết thì kết thúc.

Với mạng phản hồi: Nếu mức tích cực của nút ra bằng hoặc xấp xỉ bằng hằng số thì kết thúc.

Nếu không thì quay lại bước 2.

### **2.1.6. Đánh giá về mạng nơon**

Mạng nơon là một công cụ hữu hiệu trong các mô hình tính toán thông minh với một số đặc điểm chính sau:

- Cho phép xây dựng một mô hình tính toán có khả năng học dữ liệu cao: Chỉ cần đưa vào cho mạng một tập dữ liệu trong quá trình học là mạng có thể phát hiện những ràng buộc dữ liệu và áp dụng những ràng buộc này trong quá trình sử dụng mà không cần có thêm các tri thức về miền ứng dụng. Khả năng này cho phép xây dựng mô hình dữ liệu khá dễ dàng.

- Xử lý các quá trình phi tuyến: Mạng có khả năng xấp xỉ những ánh xạ phi tuyến tùy ý nên có thể giải được những bài toán phi tuyến phức tạp. Nó có thể thực hiện nhiều phép lọc nằm ngoài khả năng của những bộ lọc tuyến tính thông thường. Đặc trưng này rất quan trọng, ví dụ trong xấp xỉ mạng, miễn nhiễu (chấp nhận nhiễu) và có khả năng phân lớp.

- Khả năng của các quá trình xử song song và phân tán: Có thể đưa vào mạng một lượng lớn các nơon liên kết với nhau theo những lược đồ với các kiến trúc khác nhau. Mạng có cấu trúc song song lớn, có khả năng tăng tốc độ tính toán và hy vọng sẽ đáp ứng được yêu cầu của những hệ thống cần có độ chính xác cao hơn những hệ thống truyền thống.

- Mạng nơon có khả năng dung thứ lỗi cao: Cố gắng bắt chước khả năng dung thứ lỗi của não theo nghĩa hệ thống có thể tiếp tục làm việc và điều chỉnh khi nhận tín hiệu vào có một phần thông tin bị sai lệch hoặc bị thiếu.

- Khả năng thích nghi và tự tổ chức: về đặc trưng này, người ta đề cập tới khả năng xử lý thích nghi và điều chỉnh bền vững dựa vào các thuật toán thích nghi và các quy tắc tự tổ chức.

- Hơn nữa, mặc dù có rất nhiều kỹ thuật và giải thuật được sử dụng trong khai phá dữ liệu, một số kỹ thuật còn được kết hợp để sử dụng có hiệu quả, song mạng nơon vẫn có những ưu điểm đáng chú ý như:

- Tự động tìm kiếm tất cả các mối quan hệ có thể giữa các nhân tố chính.
- Mô hình hoá tự động các bài toán phức tạp mà không cần biết trước mức độ phức tạp.
- Có khả năng chiết xuất ra những thông tin nhanh hơn rất nhiều so với nhiều công cụ khác.

Với các đặc điểm trên ta thấy: Mạng nơon cho phép dễ dàng xây dựng các mô hình thích nghi mà trong đó sự thay đổi liên tục về quy luật dữ liệu có thể dễ dàng được cập nhật trong quá trình học lại của mạng. Tuy nhiên, mạng nơon không phải một công cụ vạn năng, nó có một số nhược điểm:

- Mạng chỉ có thể làm việc với những dữ liệu số.
- Để mạng đạt hiệu quả cần có một bộ dữ liệu mẫu đủ lớn cho quá trình học.
- Mạng chỉ có tính chất nội suy. Khả năng ngoại suy rất kém.
- Mạng không đưa ra được cơ chế giải thích.
- Đôi khi mạng chưa đảm bảo độ hội tụ cần thiết cho quá trình sử dụng.

Như vậy, một mạng nơon nhân tạo khi đem vào sử dụng trước tiên phải cho mạng học các mẫu học. Bộ trọng số ban đầu của mạng thường được khởi tạo ngẫu nhiên. Quá trình học sẽ dần dần thay đổi bộ trọng số này để cực tiểu hoá sai số. Tuy nhiên, với bộ trọng số khởi tạo ngẫu nhiên, mạng thường bị rơi vào các giá trị cực tiểu địa phương và quá trình hiệu chỉnh trọng số này thường không mang lại kết quả mong muốn, tức là không làm giảm đáng kể sai số hoặc thậm chí có lúc còn làm tăng sai số. Một phương pháp tránh được trường hợp cực trị địa phương là kết hợp giải thuật di truyền với mạng nơon. Giải thuật di truyền sẽ tìm kiếm một cách toàn cục các bộ trọng số tốt nhất cho mạng nơon và cho kết quả là vùng lân cận với điểm cực trị toàn cục. Sau đó, một vài bộ trọng số tốt nhất sẽ được dùng làm các giá trị trọng số khởi tạo cho mạng nơon và kết quả sẽ là cực trị toàn cục.

## 2.2. GIẢI THUẬT DI TRUYỀN TRONG KHAI PHÁ DỮ LIỆU

Giải thuật di truyền (Genetic Algorithm - GA) là một phương pháp tìm kiếm cực trị tổng thể, kỹ thuật tối ưu tổng thể có tầm quan trọng rất lớn đối với nhiều vấn đề khác nhau trong khoa học và kỹ thuật. Trong khai phá dữ liệu, giải thuật di truyền thường được sử dụng trên nền của các kỹ thuật khác như mạng nơon hay phân lớp theo k láng giềng gần nhất. Mặc dù vậy, giải thuật di truyền là một kỹ thuật rất cần thiết vì hầu hết các kỹ thuật khai phá dữ liệu tóm lại đều là vấn đề tối ưu hoá. Đối với mạng nơon, đó là vấn đề tìm kiếm các trọng số cho một cấu trúc mạng tối ưu. Đối với k láng giềng gần nhất, đó là vấn đề tìm các trọng số quan trọng tối ưu để áp dụng cho mỗi yếu tố dự đoán. Đối với cây quyết định, đó là bài toán tìm kiếm các yếu tố dự đoán tốt nhất và các giá trị để phân tách trong việc tối ưu hoá cây. Giải thuật di truyền được đánh giá bằng hàm thích nghi để xác định các mô hình dự đoán tối ưu cho việc khai phá dữ liệu.

### 2.2.1. Cơ bản về giải thuật di truyền

Ý tưởng của giải thuật di truyền là mô phỏng theo cơ chế của quá trình chọn lọc và di truyền trong tự nhiên. Từ tập các lời giải có thể ban đầu, thông qua nhiều bước tiến hoá để hình thành các tập mới với những lời giải tốt hơn, cuối cùng sẽ tìm được lời giải gần tối ưu nhất [1, 6].

GA sử dụng các thuật ngữ lấy từ di truyền học:

- Một tập hợp các lời giải được gọi là một Lớp hay Quần thể (population).
- Mỗi lời giải được biểu diễn bởi một Nhiễm sắc thể hay Cá thể (chromosome).
- Nhiễm sắc thể được tạo thành từ các gen

Một quá trình tiến hoá được thực hiện trên một quần thể tương đương với sự tìm kiếm trên không gian các lời giải có thể của bài toán. Quá trình tìm kiếm này luôn đòi hỏi sự cân bằng giữa hai mục tiêu: *Khai thác lời giải tốt nhất và xem xét toàn bộ không gian tìm kiếm.*

GA thực hiện tìm kiếm theo nhiều hướng bằng cách duy trì tập hợp các lời giải có thể và khuyến khích sự hình thành và trao đổi thông tin giữa các hướng.

Tập lời giải phải trải qua nhiều bước tiến hoá, tại mỗi thế hệ, một tập mới các cá thể được tạo ra có chứa các phần của những cá thể thích nghi nhất trong thế hệ cũ. Đồng thời giải thuật di truyền khai thác một cách có hiệu quả thông tin trước đó để suy xét trên điểm tìm kiếm mới với mong muốn có được sự cải thiện qua từng thế hệ. Như vậy, các đặc trưng được đánh giá tốt sẽ có cơ hội phát triển và các tính chất tồi (không thích nghi với môi trường) sẽ có xu hướng biến mất.

Giải thuật di truyền tổng quát được mô tả như sau:

*PROCEDURE GeneticAlgorithm;*

*BEGIN*

*T:=0;*

*Khởi tạo lớp P(t);*

*Đánh giá lớp P(t);*

*While not (Điều\_kiện\_kết\_thúc) do*

*Begin*

*t:=t+1;*

*Chọn lọc P(t) từ P(t-1);*

*Kết hợp các cá thể của P(t);*

*Đánh giá lớp P(t);*

*end;*

*END;*

Trong đó:

- Tập hợp các lời giải ban đầu được khởi tạo ngẫu nhiên.
- Trong vòng lặp thứ t, GA xác định tập các nhiễm sắc thể  $P(t)=\{x_1^t, x_2^t, \dots, x_n^t\}$  bằng cách chọn lựa các nhiễm sắc thể thích nghi hơn từ  $P(t-1)$ . Mỗi nhiễm sắc thể  $x_i^t$  được đánh giá để xác định độ thích nghi của nó và một số thành viên của  $P(t)$  lại được tái sản xuất nhờ các toán tử *Lai ghép* và *Đột biến*.

Khi áp dụng GA để quyết một bài toán cụ thể, phải làm rõ các vấn đề sau:

1. Chọn cách biểu diễn di truyền nào đối với những lời giải có thể của bài toán?
2. Tạo tập lời giải ban đầu như thế nào?

3. Xác định hàm đánh giá để đánh giá mức độ thích nghi của các cá thể.
4. Xác định các toán tử di truyền để sản sinh con cháu.
5. Xác định giá trị các tham số mà GA sử dụng như kích thước tập lời giải, xác suất áp dụng các toán tử di truyền,...

Như vậy GA là một giải thuật lặp nhằm giải quyết các bài toán tìm kiếm, nó khác với các thủ tục tối ưu thông thường ở những điểm cơ bản sau:

- Giải thuật di truyền làm việc với bộ mã của tập thông số chứ không làm việc trực tiếp với giá trị của các thông số.
- Giải thuật di truyền tìm kiếm song song trên một quần thể chứ không tìm kiếm từ một điểm, mặt khác nhờ áp dụng các toán tử di truyền, nó sẽ trao đổi thông tin giữa các điểm, như vậy sẽ giảm bớt khả năng kết thúc tại một cực tiểu cục bộ mà không tìm thấy cực tiểu toàn cục.
- Giải thuật di truyền chỉ sử dụng thông tin của hàm mục tiêu để đánh giá quá trình tìm kiếm chứ không đòi hỏi các thông tin bổ trợ khác.
- Các luật chuyển đổi của giải thuật di truyền mang tính xác suất chứ không mang tính tiền định.

Các thông số của bài toán được mã hoá thành các chuỗi. Cách đơn giản nhất là chúng ta dùng chuỗi bit để mã hoá các thông số. Mỗi thông số được mã hoá bằng một chuỗi bit có độ dài nào đó, sau đó nối chúng lại với nhau, ta sẽ có một chuỗi mã hoá cho tập các thông số. Để tính toán giá trị hàm mục tiêu tương ứng với mỗi chuỗi thông số, ta phải giải mã bộ thông số này theo một quy tắc nào đó. Giải thuật di truyền tìm kiếm song song trên một tập các chuỗi, do đó giảm thiểu được khả năng bỏ qua các cực trị toàn cục và dừng lại ở cực trị địa phương. Điều này giải thích vì sao giải thuật di truyền mang tính toàn cục.

Hiện nay giải thuật di truyền được áp dụng ngày càng nhiều trong kinh doanh, khoa học và kỹ thuật vì tính chất không quá phức tạp mà hiệu quả của nó. Hơn nữa, giải thuật di truyền không đòi hỏi khắt khe đối với không gian tìm kiếm như giả định về sự liên tục, sự có đạo hàm,... Bằng lý thuyết và thực nghiệm, giải thuật di truyền đã được chứng minh là giải thuật tìm kiếm toàn cục mạnh trong các không gian lời giải phức tạp.

### 2.2.2. Một số cách biểu diễn lời giải của giải thuật di truyền

Biểu diễn lời giải là vấn đề đầu tiên được quan tâm tới khi bắt tay vào giải quyết một bài toán bằng GA. Việc lựa chọn cách biểu diễn lời giải như thế nào phụ thuộc vào từng lớp bài toán thậm chí vào từng bài toán cụ thể.

GA kinh điển dùng chuỗi nhị phân có chiều dài xác định để biểu diễn lời giải. Tuy nhiên, thực tế cho thấy cách biểu diễn này khó áp dụng trực tiếp cho các bài toán tối ưu cỡ lớn có nhiều ràng buộc. Vì lý do đó, GA cải tiến hay còn gọi là *Chương trình tiến hoá* đã tìm kiếm các cách biểu diễn thích nghi và tự nhiên hơn với các bài toán thực tế như: Biểu diễn theo trật tự, biểu diễn theo giá trị thực, biểu diễn bằng các cấu trúc cây, ma trận, ... Phần này sẽ trình bày tổng quan về các cách biểu diễn đó.

#### 2.2.2.1. Biểu diễn nhị phân (Binary encoding)

Trong biểu diễn nhị phân, mỗi nhiễm sắc thể là một chuỗi các bit 0 hoặc 1.

Chẳng hạn:

NST A: 101100101100101011100101

NST B: 111111100000110000011111

**Ví dụ:** Bài toán “Xếp ba lô” được phát biểu: “Cho một tập các đồ vật, mỗi đồ vật có giá trị và kích thước xác định, cho biết sức chứa của ba lô. Hãy chọn cách xếp các đồ vật vào ba lô sao cho tổng giá trị của các đồ vật là cao nhất”.

Biểu diễn mỗi lời giải của bài toán trên bằng một chuỗi nhị phân, ở đó mỗi bit 0 hoặc 1 ứng với một đồ vật không được chọn hoặc được chọn.

Với cách biểu diễn đó, bài toán được phát biểu lại như sau: “ Cho một tập các khối lượng  $W[i]$ , tập các giá trị  $P[i]$  và sức chứa  $C$ . Tìm một vectơ nhị phân  $x = \langle x_1, x_2, \dots, x_n \rangle$  thoả mãn:

$$\sum_{i=1}^n x[i] \cdot W[i] \leq C$$

với  $P(x) = \sum_{i=1}^n x[i] \cdot P[i]$  là cực đại.

#### 2.2.2.2. Biểu diễn hoán vị (Permutation encoding)

Sử dụng trong bài toán mà thứ tự các thành phần của lời giải quyết định mức độ phù hợp của lời giải, điển hình như bài toán “ Người du lịch”.

Với cách biểu diễn thứ tự, cách sắp xếp của các gen khác nhau cho ta các nhiễm sắc thể khác nhau, mỗi nhiễm sắc thể là một chuỗi các số nguyên diễn tả quan hệ tiếp nối. Lời giải được biểu diễn bằng một vectơ số nguyên  $v=(i_1, i_2, \dots, i_n)$  với  $v$  là một hoán vị của tập thứ tự.

Ví dụ: NST A: ( 1 5 3 2 6 4 7 9 8 )

NST B: ( 8 5 6 7 2 3 1 4 9 )

#### 2.2.2.3. Biểu diễn giá trị (Value encoding)

Thường dùng trong các bài toán mà cách biểu diễn chuỗi nhị phân là khó thực hiện như miền xác định của các thành phần lời giải khá lớn với độ chính xác yêu cầu cao, miền xác định không rõ ràng, hay các bài toán mà việc biểu diễn nhị phân là “ không tự nhiên”.

Trong biểu diễn giá trị, mỗi cá thể là một chuỗi các giá trị liên quan đến bài toán, các giá trị có thể là số thực, số nguyên, ký tự hay các đối tượng phức tạp khác.

Ví dụ: NST A: (0.1229 2.9234 3.0012, 0.3567, 4.3828)

NST B (AJUHNEOLDOGSGLLIKUFSEJHJH)

#### 2.2.2.4. Biểu diễn dạng cây (Tree encoding)

Cách biểu diễn lời giải dùng cấu trúc cây được dùng chủ yếu trong các chương trình tiến hoá, trong biểu diễn biểu thức, hay lập các chương trình di truyền học. Với cách biểu diễn này, mỗi cá thể là một cây các đối tượng.

### 2.2.3. Các toán tử di truyền

Các cá thể trong giải thuật di truyền là các chuỗi bit được tạo bởi việc cắt dán các chuỗi bit con. Mỗi chuỗi bit đại diện cho một tập thông số trong không gian tìm kiếm, nên được coi là lời giải tiềm năng của bài toán tối ưu. Từ mỗi chuỗi bit ta giải mã để tính lại tập thống số, sau đó tính được giá trị hàm mục tiêu. Từ đó, giá trị hàm mục tiêu được biến đổi thành giá trị độ phù hợp của từng chuỗi.

Quần thể chuỗi ban đầu được khởi tạo ngẫu nhiên, sau đó tiến hoá từ thế hệ này sang thế hệ khác bằng các toán tử di truyền (tổng số chuỗi trong mỗi quần thể là không thay đổi). Có ba toán tử di truyền đơn giản là:

- Tái tạo
- Lai ghép
- Đột biến

#### **1.2.3.1. Đánh giá độ thích nghi của cá thể và phép tái tạo**

Mỗi bài toán trong thực tế có các điều kiện ràng buộc khác nhau đối với lời giải. Quá trình tìm kiếm lời giải chính là quá trình tiến hoá mà ở mỗi bước, cần phải lựa chọn các cá thể thích nghi hơn để tái sản xuất ở thế hệ sau bằng phép tái tạo.

Để đánh giá các lời giải, người ta xây dựng hàm thích nghi Fitness(). Tái tạo là quá trình sao chép các chuỗi (các cá thể) từ thế hệ trước sang thế hệ sau theo giá trị hàm thích nghi (còn gọi là hàm mục tiêu hay hàm sức khoẻ).

Coi giá trị của hàm là số đo độ phù hợp, giải thuật di truyền sử dụng giá trị hàm thích nghi để quyết định số con của một chuỗi: Những chuỗi với giá trị hàm thích nghi lớn sẽ có xác suất lớn trong việc đóng góp một hay nhiều con cháu trong thế hệ tiếp theo.

Toán tử này mô phỏng theo học thuyết sinh tồn của Darwin, chỉ có các cá thể khoẻ mới có cơ hội sống sót và đóng góp con cháu vào các thế hệ sau.

Hàm thích nghi được xây dựng như sau:

Xét lớp lời giải P có n cá thể, với mỗi cá thể  $h_i$  thuộc P, tính độ thích nghi Fitness( $h_i$ ).

Xác suất chọn cá thể  $h_i$  để tái sản xuất được xác định bởi công thức:

$$\text{Pr}(h_i) = \frac{\text{Fitness}(h_i)}{\sum_{j=1}^n \text{Fitness}(h_j)}$$

Tại mỗi bước tiến hoá, các cá thể được chọn tái tạo là các cá thể có xác suất Pr() cao, điều này cho phép tạo ra các thế hệ sau có độ thích nghi tốt hơn thế hệ trước.

Fitness() còn được dùng để xác định điểm dừng của quá trình tìm kiếm lời giải khi đã đạt được độ thích nghi chấp nhận được.



Có nhiều cách để chọn lựa cá thể khỏe, tuy nhiên cần phải thận trọng trong thuật toán chọn lựa sao cho đảm bảo các chuỗi khỏe nhất có đóng góp nhiều con trong quần thể, còn các chuỗi yếu vẫn có khả năng đóng góp vào quần thể theo một xác suất nào đó. Điều này làm hạn chế khả năng các cá thể siêu khỏe sẽ nhanh chóng chiếm toàn bộ quần thể và thuật toán sẽ dừng rất nhanh vì toàn bộ quần thể chỉ gồm một vài nhóm các chuỗi giống nhau. Vì trong trường hợp đó, kết quả tìm được có nhiều khả năng chỉ là giá trị cực trị địa phương.

Một trong các cách đơn giản và hiệu quả để thực hiện toán tử tái tạo là sử dụng vòng tròn Rulet. Trong vòng tròn Rulet, mỗi cá thể sẽ chiếm một vùng có diện tích tỷ lệ với độ thích nghi của chúng. Diện tích của cả vòng tròn tương ứng với 100% tổng mức thích nghi của toàn quần thể. Việc thực hiện lựa chọn chuỗi con trong tái tạo được thực hiện như sau:

- Đánh số các cá thể trong quần thể, tính tổng độ thích nghi sumfitness của toàn quần thể đồng thời ứng với mỗi cá thể, tính một tổng chạy bằng tổng độ thích nghi của cá thể đó và các cá thể đứng trước đó.

- Sinh một số ngẫu nhiên  $n$  trong khoảng từ 0 đến tổng mức thích nghi sumfitness.

Cá thể đầu tiên trong quần thể có tổng chạy lớn hơn hoặc bằng  $n$  sẽ được chọn.

Ví dụ:

STT	Chuỗi	Độ thích nghi	Tỷ lệ %	Tổng chạy
1	1011001	86	24.5	86
2	0100001	58	16.6	144
3	1101001	176	50.3	320
4	1001010	30	8.6	350
	<b>Tổng</b>	<b>350</b>	<b>100</b>	

*Bảng 2.1: Ví dụ dùng phép tái tạo*

Sinh số ngẫu nhiên  $n = 175$  thì chuỗi thứ 3 trong bảng 2.1 là chuỗi được chọn.

Khi đã chọn được cá thể cho tái tạo, chuỗi đó sẽ được sao chép vào quần thể mới. Cách này cho phép các cá thể có độ thích nghi lớn có nhiều cơ hội được đóng góp con cháu vào các thế hệ tiếp theo. Tuy nhiên mỗi thế hệ tiến hoá còn phải có thêm các toán tử lai ghép và đột biến nữa thì mới thực sự hoàn thành.

#### 1.2.3.2. Lai ghép (Crossover)

Các cá thể trong quần thể sau khi đã tái tạo được chọn lai ghép với nhau. Toán tử lai ghép được coi là toán tử di truyền quan trọng nhất, nó kết hợp các đặc trưng của các cá thể bố mẹ để tạo ra hai cá thể con bằng cách trao đổi các đoạn gen tương ứng trên hai cá thể cha mẹ.

Phép lai ghép chọn ngẫu nhiên hai chuỗi bất kì trong quần thể sau khi đã thực hiện tái tạo, đồng thời sinh một số ngẫu nhiên, nếu nhỏ hơn xác suất lai ghép  $p_c$  thì thực hiện lai ghép, ngược lại thì chỉ thực hiện sao chép đơn giản hai chuỗi vào quần thể mới. Phép lai ghép hai chuỗi thực hiện trao đổi hai đoạn mã cho nhau, rồi đưa hai chuỗi kết quả vào một quần thể mới. Chú ý rằng lực lượng của quần thể là không thay đổi, do đó ở mỗi thế hệ tiến hoá, chỉ tiến hành lai ghép cho tới khi nào quần thể mới có đủ số chuỗi thì dừng. Vị trí trao đổi khi lai ghép được chọn ngẫu nhiên trong khoảng  $[1, L-1]$ , với  $L$  là độ dài chuỗi.

**Ví dụ:** Giả sử chúng ta có hai chuỗi bố mẹ là:

$$A_1 = 0 \ 1 \ 1 \ 0 \ 1$$

$$A_2 = 1 \ 1 \ 0 \ 1 \ 0$$

Với vị trí lai ghép là 3 thì hai chuỗi con sinh ra sẽ là:

$$A'_1 = 1 \ 1 \ 0 \ 0 \ 1$$

$$A'_2 = 0 \ 1 \ 1 \ 1 \ 0$$

#### 1.2.3.3. Đột biến (Mutation)

Tái tạo và lai ghép chỉ tạo ra các chuỗi mới chứ không đem lại cho quần thể một thông tin mới. Phép đột biến ngăn ngừa khả năng GA chỉ tìm kiếm trên một vùng cục bộ và kết quả chỉ là cực trị địa phương.

Toán tử đột biến sẽ thay đổi ngẫu nhiên một bit thông tin của một chuỗi với xác suất đột biến  $p_m$ . Xác suất đột biến thể hiện mức độ thường xuyên được thực

hiện của toán tử đột biến, Tuy nhiên, xác suất đột biến phải đủ nhỏ vì thực tế toán tử đột biến là toán tử tìm kiếm ngẫu nhiên.

Với phương pháp mã hoá chuỗi bit, một bit thông tin A nếu bị đột biến được biến đổi bằng công thức đơn giản:  $A = 1 - A$

Ba toán tử tái tạo, lai ghép và đột biến được tiến hành lặp đi lặp lại cho đến khi các chuỗi con chiếm toàn bộ quần thể mới. Quần thể mới sẽ bao gồm các cá thể của ba loại: Lai ghép nhưng không đột biến, bị đột biến sau khi lai ghép và không lai ghép cũng không đột biến mà chỉ đơn thuần là sao chép lại.

Như vậy, trong một giải thuật di truyền đơn giản, chúng ta cần xác định các thông số sau:

- Số cá thể trong quần thể n
- Xác suất lai ghép  $p_c$
- Xác suất đột biến  $p_m$
- Độ gối của các quần thể G

Ba thông số đầu rất dễ hiểu và đã được nhắc tới trong các phần trên. Còn độ gối G được tác giả De Jong đưa vào năm 1975, ý nghĩa của nó là cho phép quần thể mới chứa một phần của quần thể cũ: Với  $G = 1$ , tất cả các cá thể mới đều được sinh ra bởi các toán tử của giải thuật di truyền, với  $0 < G < 1$ , sẽ có  $G \cdot n$  cá thể được đưa trực tiếp từ quần thể cũ sang quần thể mới.

Sau đây, ta sẽ xét một ví dụ đơn giản để thấy được sự hoạt động của giải thuật di truyền cũng như tác dụng của chúng:

Giả sử cần tìm giá trị cực đại của hàm số  $f(x) = x^2$ , với x nằm trong khoảng [0,31].

Ta mã hoá biến x thành chuỗi có độ dài 5 bit. Như vậy, số 7 sẽ được mã thành chuỗi '00111'. Hàm thích nghi của các chuỗi được tính bằng  $f(x)$ . Với các giá trị thông số:  $n = 4$ ,  $p_m = 0.01$ ;  $G = 1$  và quần thể ban đầu được khởi tạo một cách ngẫu nhiên, bảng 2.2 và bảng 2.3 mô tả các quá trình tái tạo, lai ghép và đột biến trong một thế hệ.

STT	Các cá thể ban đầu	x	Độ thích nghi $f = x^2$	Tỷ lệ thích nghi $f_i / \sum f$	Số con
-----	--------------------	---	----------------------------	------------------------------------	--------

1	01001	9	81	0.08	1
2	11000	24	576	0.55	2
3	00100	4	16	0.02	0
4	10011	19	361	0.35	1
Tổng thích nghi			1043		
Giá trị thích nghi trung bình: 259					

Bảng 2.2: Quá trình tái tạo

STT	Quần thể tạm thời	Cá thể ghép đôi	Vị trí lai ghép	Quần thể mới	x	Độ thích nghi
1	01001	2	4	01000	8	64
2	11000	1	4	11001	25	625
3	11000	4	2	11011	27	729
4	10011	2	2	10000	16	256
Tổng thích nghi						1674
Giá trị thích nghi trung bình						419

Bảng 2.3: Quá trình lai ghép

Quan sát các giá trị trong các bảng 2.2 và 2.3, ta thấy chuỗi 1 và 4 đóng góp một bản copy vào quần thể tạm thời, chuỗi số 2 đóng góp 2 bản copy, chuỗi số 3 có độ thích nghi quá nhỏ so với các chuỗi còn lại do đó không đóng góp con nào vào quần thể tạm thời. Trong ví dụ này, không có bit nào bị đột biến. Giá trị thích nghi trung bình của toàn quần thể đã tăng lên từ 259 thành 419, điều này chứng tỏ các cá thể trong quần thể đã tốt lên sau một thế hệ tiến hoá.

## 2.2.4. Cơ sở toán học của giải thuật di truyền

### 1.2.4.1. Một số khái niệm

• **Giản đồ:** Là mẫu mô tả một tập các chuỗi giống nhau tại một số điểm nào đó.

Ví dụ về các giản đồ: 101\*\*\*\* 1, \*\*1\*0\*\*1, 10110\*\*\*\*....

Trong các giản đồ đó, các vị trí chứa các số 0, 1 là các vị trí cố định trong chuỗi, các dấu \* đại diện cho một kí tự thay đổi, có thể là 0 hoặc 1. Tập kí tự của giản đồ là  $\{0,1,*\}$ .

- **Bậc của giản đồ H,** kí hiệu là  $o(H)$ , là số các vị trí cố định có trong giản đồ.
- **Độ dài của giản đồ,** kí hiệu là  $\delta(H)$ , là khoảng cách giữa vị trí cố định đầu tiên và vị trí cố định cuối cùng trong giản đồ.

### 1.2.4.2. Các định lý giản đồ

Số chuỗi của mỗi giản đồ thường bị ảnh hưởng ít nhiều trong các toán tử tái tạo, lai ghép và đột biến, tùy thuộc vào giá trị thích nghi trung bình của các cá thể trong giản đồ.

Đối với toán tử tái tạo, gọi  $m = m(H,t)$  là tần số chuỗi của giản đồ H tại thời điểm t. Gọi  $f_i$  là giá trị thích nghi của chuỗi  $A_i$  ta sẽ có  $p_i = f_i / \sum f_i$  là xác suất của chuỗi  $A_i$  được chọn. Số mẫu của giản đồ H trong quần thể mới sẽ là:

$$m(H, t+1) = \frac{m(H,t) * n * f(H)}{\sum f_i} \quad (2.1)$$

Trong đó, n là số cá thể trong quần thể,  $f(H)$  là giá trị thích nghi trung bình của các cá thể thuộc giản đồ H.

Thay giá trị thích nghi trung bình của toàn quần thể  $f_{th} = \frac{\sum f_i}{n}$  (2.2) vào phương trình trên ta có:

$$m(H, t+1) = \frac{m(H,t) * f(H)}{f_{th}} \quad (2.3)$$

Như vậy, các giản đồ có giá trị thích nghi trung bình lớn hơn giá trị thích nghi trung bình của toàn quần thể sẽ có số cá thể tăng trong các thế hệ tiếp theo và ngược lại.

Đối với toán tử lai ghép, các giản đồ có độ dài càng lớn thì càng dễ bị phá vỡ. Giả sử các chuỗi có độ dài 1 thì xác suất giản đồ bị phá huỷ sẽ là:

$$p_d = \frac{\delta(H)}{1-1} \quad (2.4)$$

Điều này rất dễ hiểu bởi vì tất cả các vị trí lai ghép nằm giữa các điểm cố định đều làm cho giản đồ bị phá vỡ. Vậy xác suất tồn tại của một giản đồ sẽ là:

$$p_\delta = 1 - \frac{\delta(H)}{1-1} \quad (2.5)$$

Nếu tính cả xác suất tạp lai  $P_c$  ta có xác suất tồn tại của một giản đồ qua toán tử lai ghép là:

$$p_\delta \geq 1 - \frac{p_\delta * \delta(H)}{1-1} \quad (2.6)$$

Đối với toán tử đột biến, xác suất tồn tại của một bit trong chuỗi là  $1-p_m$ , với  $p_m$  là xác suất đột biến của một bit. Để cho giản đồ tồn tại, tất cả các bit cố định của giản đồ đều phải tồn tại, do đó xác suất tồn tại của một giản đồ  $H$  qua toán tử đột biến là:

$$(1-p_m)^{m(H)} \approx 1 - o(H) * p_m \text{ vì } p_m \text{ nhỏ} \quad (2.7)$$

Vậy số chuỗi của một giản đồ sau cả 3 toán tử được đánh giá bằng công thức:

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{f_{th}} \left[ 1 - p_c \frac{\delta(H)}{1-1} - o(H) * p_m \right] \quad (2.8)$$

Kết quả ở công thức (2.8) cho thấy, các giản đồ bậc thấp, có độ dài ngắn và có giá trị thích nghi trung bình lớn hơn giá trị thích nghi trung bình của toàn quần thể sẽ có số chuỗi tăng trong các thế hệ tiếp theo.

## 2.2.5. Những cải tiến của giải thuật di truyền

Dựa trên những toán tử di truyền đơn giản, các sơ đồ lựa chọn, các toán tử cao cấp đã được đưa vào nhằm cải tiến hoạt động của giải thuật di truyền trong những bài toán phức tạp.

### 1.2.5.1. Các toán tử cao cấp

Toán tử cao cấp được chia thành hai loại: Toán tử vi mô và toán tử vĩ mô. Toán tử vi mô chỉ tác động đến từng gen của các cá thể, còn các toán tử vĩ mô thì tác động đến toàn quần thể các cá thể.

#### ➤ Toán tử vi mô:

- Lai ghép nhiều điểm:

Trong toán tử lai ghép nhiều điểm, số vị trí lai ghép được chọn  $N_c$  là lớn hơn một. Hai chuỗi lai ghép với nhau không chỉ trao đổi một đoạn mã cho nhau mà chúng trao đổi  $N_c$  đoạn mã cho nhau.

Toán tử lai ghép nhiều điểm có nhiều lợi ích như chúng có thể giải quyết được nhiều bài toán mà toán tử lai ghép một điểm không giải quyết được. Tuy nhiên, phải thận trọng khi sử dụng toán tử lai ghép nhiều điểm, vì toán tử này dễ phá vỡ các giản đồ có ích, đồng thời khi  $N_c$  quá lớn, toán tử này sẽ trở thành phép tìm kiếm ngẫu nhiên.

- Toán tử sắp xếp lại:

Trong thực tế, giá trị mức thích nghi của một chuỗi không những phụ thuộc vào giá trị của một gen mà còn phụ thuộc vào vị trí của gen đó trong chuỗi, hoặc tổ hợp của gen đó với một số gen khác. Do đó, toán tử sắp xếp lại sẽ tìm kiếm và sắp xếp lại các gen trong chuỗi, hoặc tổ hợp của gen đó với một số gen khác. Do đó, toán tử sắp xếp lại sẽ tìm kiếm và sắp xếp lại các gen trong chuỗi. Một toán tử sắp xếp lại thường được sử dụng trong giải thuật di truyền là toán tử đảo ngược. Dưới tác động của toán tử này, hai điểm được chọn dọc theo chiều dài của chuỗi, rồi cắt chuỗi tại hai điểm đó. Tiếp theo, hai chuỗi gen con ở hai đầu sẽ được đổi chỗ cho nhau.

Ví dụ: Có một chuỗi có độ dài 8 như sau:

$$A = 1\ 0\ |0100|01$$

Chuỗi trên được cắt tại vị trí số 2 và vị trí số 6, sau đó tráo đổi hai chuỗi ở hai đầu cho nhau, chúng ta có chuỗi kết quả:

$$A' = 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0$$

#### ➤ **Toán tử vĩ mô:**

Toán tử vĩ mô là toán tử hoạt động ở mức quần thể, ý tưởng để thực hiện toán tử vĩ mô là sử dụng hàm chia sẻ. Ý tưởng này nhằm mô phỏng quá trình hình thành của các loài khác nhau trong tự nhiên, hạn chế sự phát triển không kiểm soát được của một số nhóm cá thể trong quần thể. Thực hiện hàm chia sẻ nhằm làm cho các cá thể trong quần thể san sẻ mức thích nghi cho nhau, giảm sự sai khác đáng kể giữa chúng.

#### **1.2.5.2. Các sơ đồ lựa chọn**

Trong giải thuật di truyền đơn giản, sơ đồ lựa chọn sử dụng là vòng tròn Rulet. Sơ đồ này có thể bỏ qua các cá thể khỏe nhất với một xác suất nào đó. Để khắc phục, chúng ta có một số sơ đồ lựa chọn mới như sau [5]:

- Ưu tiên cá thể tốt: Chiến lược của sơ đồ này sẽ sao chép các cá thể tốt nhất cho thế hệ tiếp theo. Nó sẽ làm tăng tốc độ của quá trình hội tụ, tuy nhiên điểm hội tụ có thể là các cực trị địa phương. Thực nghiệm đã cho thấy chiến lược này có cải tiến sự hoạt động của giải thuật di truyền.

- Lấy mẫu một cách tiên định: Trong chiến lược này, mỗi chuỗi có số con sao chép được xác định trước và bằng  $\left\lfloor n * \frac{f_i}{\sum f_j} \right\rfloor$ , trong đó  $f_i$  là mức thích nghi của cá thể thứ  $i$ . Dễ dàng nhận thấy nếu thực hiện như trên thì quần thể mới sẽ không có đủ số chuỗi như quần thể cũ, ta sẽ lấp đầy chỗ trống ấy bằng các cá thể tốt nhất của quần thể cũ theo một phương pháp nào đó.

- Lấy mẫu xác suất phân dư và thay thế: Phương pháp này giống như sự kết hợp giữa phương pháp lấy mẫu tiên định và sử dụng vòng tròn Rulet. Phần trống của



quần thể sau khi lựa chọn tiền định các chuỗi sẽ được lấp đầy bằng cách sử dụng vòng tròn Rulet, với các trọng số là phần thập phân của  $n * \frac{f_i}{\sum f_j}$ .

- Lấy mẫu xác suất phần dư không thay thế: Sơ đồ này cũng xuất phát từ lấy mẫu tiền định nhưng phần trống trong quần thể sẽ được lấp đầy bằng các cá thể với xác suất của mỗi cá thể bằng phần thập phân của  $n * \frac{f_i}{\sum f_j}$ .

## ❖ Kết luận chương 2

Như vậy, mặc dù có rất nhiều kỹ thuật và giải thuật được dùng trong khai phá dữ liệu, một số kỹ thuật còn được kết hợp với nhau để sử dụng có hiệu quả hơn, song mạng nơon và giải thuật di truyền vẫn là những kỹ thuật đáng chú ý. Chương 2 cũng đã trình bày một cách khá chi tiết về mạng nơon và giải thuật di truyền. Cụ thể, là những vấn đề về lựa chọn cấu trúc mạng và các tham số, xây dựng giải thuật học và lan truyền trong mạng nơon, cũng như cách biểu diễn lời giải, các toán tử di truyền cơ bản và những cải tiến của giải thuật di truyền.

### **CHƯƠNG 3:**

## **TÍCH HỢP GIẢI THUẬT DI TRUYỀN VỚI GIẢI THUẬT HUẤN LUYỆN MẠNG NƠON TRUYỀN THĂNG NHIỀU LỚP**

### **3.1. ĐẶT VẤN ĐỀ**

Từ những nghiên cứu trình bày ở chương 2, ta nhận thấy mạng nơon có khả năng chiết xuất thông tin từ những dữ liệu không chắc chắn hay những dữ liệu phức tạp nhằm phát hiện ra những xu hướng không quan sát được bằng một số các kỹ thuật khác. Tuy nhiên, với bộ trọng số khởi tạo ngẫu nhiên, mạng thường bị rơi vào các giá trị cực tiểu địa phương. Trong khi đó, giải thuật di truyền có thể tìm ra vùng quanh tâm chứa cực trị toàn cục, song lại không đảm bảo cho sự hội tụ. Hơn nữa, do hàm sức khỏe không cần phải tính đạo hàm nên có thể sử dụng các hàm ngưỡng logic thay cho việc phải dùng hàm sigmoid và luyện đến bão hòa. Chính vì thế, giải pháp tích hợp giải thuật di truyền với mạng nơon là một lẽ rất tự nhiên. Có hai khả năng lai ghép giữa hai giải thuật này:

- 1) Xuất phát bằng giải thuật GA, kết quả thu được từ tìm kiếm bằng giải thuật GA là điểm xuất phát của giải thuật huấn luyện mạng nơon.
- 2) Giải thuật huấn luyện mạng nơon có thể được thực hiện như một toán tử của giải thuật GA.

Luận văn này trình bày giải thuật lai theo hướng thứ nhất. Cụ thể là kết hợp GA với giải thuật huấn luyện mạng nơon truyền thẳng nhiều lớp.

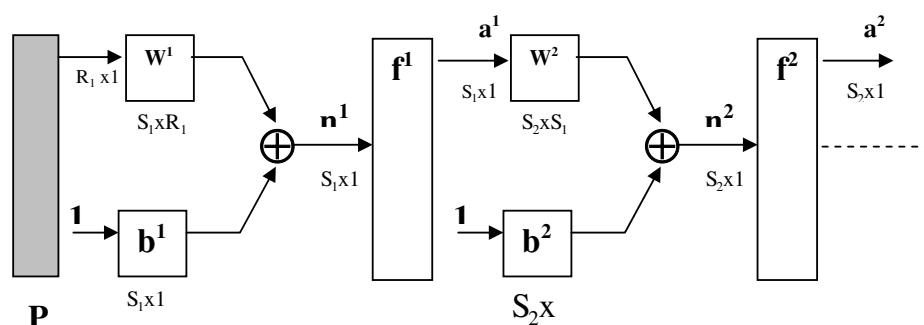
### **3.2. MẠNG NƠON TRUYỀN THĂNG NHIỀU LỚP VỚI GIẢI THUẬT LAN TRUYỀN NGƯỢC SAI SỐ VÀ MỘT SỐ CẢI TIẾN**

#### **3.2.1. Kiến trúc của mạng nơon truyền thẳng nhiều lớp**

Một mạng nơon truyền thẳng nhiều lớp (Multi-layer Feed Forward) gồm một lớp vào, một lớp ra và một hoặc nhiều các lớp ẩn. Các nơon đầu vào thực chất

không phải các nơ ron theo đúng nghĩa, bởi lẽ chúng không thực hiện bất kỳ một tính toán nào trên dữ liệu vào. Các nơ ron ở lớp ẩn và lớp ra mới thực sự thực hiện các tính toán. Cụm từ “truyền thẳng” có nghĩa là tất cả các nơ ron chỉ có thể được kết nối với nhau theo một hướng tới một hay nhiều các nơ ron khác trong lớp kế tiếp (loại trừ các nơ ron ở lớp ra).

Mỗi liên kết gắn với một trọng số, trọng số này được thêm vào trong quá trình tín hiệu đi qua liên kết đó. Các trọng số có thể dương (kích thích) hay âm (kiềm chế). Mỗi nơ ron tính toán mức kích hoạt của chúng bằng cách cộng tổng các đầu vào và đưa ra hàm chuyển (hàm kích hoạt). Một khi đầu ra của tất cả các nơ ron trong một lớp mạng cụ thể đã thực hiện tính toán thì lớp kế tiếp có thể bắt đầu thực hiện tính toán của mình bởi vì đầu ra của lớp hiện tại là đầu vào của lớp kế tiếp. Khi tất cả các nơ ron đã thực hiện tính toán thì các nơ ron đầu ra thể hiện kết quả của chúng. Dưới đây là hình vẽ minh họa mạng nơ ron truyền thẳng hai lớp.



Hình 3.1: Mạng nơ ron truyền thẳng 2 lớp

Trong đó: **P**: Vector đầu vào (vector cột)

**W<sup>1</sup>**: Ma trận trọng số của các nơ ron lớp thứ 1 có kích thước  $S_1 \times R_1$

**W<sup>2</sup>**: Ma trận trọng số của các nơ ron lớp thứ 2 có kích thước  $S_2 \times R_2$

**b<sup>1</sup>, b<sup>2</sup>**: Vector độ lệch (bias) của lớp thứ 1 và 2 (kích thước  $S_1 \times 1$  và  $S_2 \times 1$ )

**n<sup>1</sup>, n<sup>2</sup>**: Vector vào của lớp thứ 1 và thứ 2 (kích thước  $S_1 \times 1$  và  $S_2 \times 1$ )

**f<sup>1</sup>, f<sup>2</sup>**: Hàm chuyển (hàm kích hoạt) của lớp thứ 1 và 2

**a<sup>1</sup>, a<sup>2</sup>**: Đầu ra của lớp thứ 1 và 2 (kích thước  $S_1 \times 1$  và  $S_2 \times 1$ )

**⊕**: Hàm tổng thông thường (Sum)

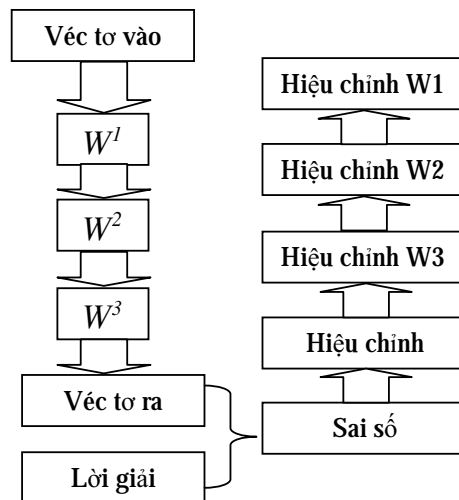
Hình 3.1 là một ví dụ về mạng hai lớp. Số nơ ron ở lớp thứ nhất và lớp thứ hai là  $S_1$  và  $S_2$  tương ứng với ma trận trọng số là  $W^1$  và  $W^2$ . Véc tơ đầu vào ở lớp thứ hai chính là véc tơ đầu ra của lớp thứ nhất, công thức tính toán cho đầu ra của lớp thứ hai là:  $a^2 = f^2(W^2(f^1(W^1.P + b^1)) + b^2)$  (3.1)

### 3.2.2. Cơ chế học của mạng nơ ron truyền thẳng nhiều lớp

Mạng nơ ron truyền thẳng nhiều lớp thường được huấn luyện bằng giải thuật lan truyền ngược của sai số (BP), giải thuật này được sử dụng thường xuyên và thông dụng tới mức nhiều tác giả đã đồng khái niệm mạng nơ ron với mạng nơ ron nhiều lớp lan truyền ngược của sai số.

Giải thuật BP là giải thuật học có giám sát, do đó nó cần một tập mẫu gồm các cặp véc tơ  $(X_i, Y_i)$ , với  $X_i$  là véc tơ vào,  $Y_i$  là véc tơ ra mong muốn. Đối với một cặp véc tơ vào và véc tơ ra mong muốn, giải thuật BP thực hiện hai giai đoạn theo dòng chảy số liệu:

- Tín hiệu vào  $X_i$  được lan truyền qua mạng từ lớp vào đến lớp ra. Kết quả của việc lan truyền là sản sinh véc tơ tín hiệu ra  $Out_i^{last}$
- Tín hiệu sai số là kết quả của việc so sánh giữa véc tơ ra mong muốn và véc tơ tín hiệu ra. Sai số được lan truyền ngược từ lớp ra tới các lớp phía trước để hiệu chỉnh các trọng số.



Hình 3.2: Sơ đồ hiệu chỉnh các trọng số của giải thuật BP

Đối với mỗi cặp tín hiệu vào ra này, hàm giá được xây dựng như sau:

$$E(w) = \frac{1}{2} \sum_{k=1}^n (y_{ik} - Out_{ik}^{last})^2 \quad (3.2)$$

Trong đó  $n$  là số nơ ron trên lớp ra;  $y_{ik}$  là thành phần thứ  $k$  của véc tơ ra mong muốn  $y_i$ ,  $out_{ik}$  là thành phần thứ  $k$  của véc tơ ra  $out_i$  do lan truyền véc tơ vào  $X_i$ .

Việc học của giải thuật thực chất là việc tìm kiếm một tập trọng số  $W$  trong không gian  $R^M$  ( $M$  là số trọng số của mạng) để lần lượt tối thiểu hoá hàm giá nêu trên. Giá trị hàm sai số  $E_i$  đối với một mẫu được tính toán dựa trên giá trị các trọng số hiện tại. Các giá trị trọng số này sau đó được hiệu chỉnh và trở thành các giá trị trọng số hiện tại để tính giá trị hàm sai số tiếp theo  $E_{i+1}$ . Dễ nhận thấy, cách làm này có khả năng tạo ra sự dao động trong quá trình hiệu chỉnh các trọng số. Kết quả hiệu chỉnh hiện tại có thể làm hỏng kết quả hiệu chỉnh ở các lần trước đó.

### 3.2.3. Thuật toán lan truyền ngược sai số

Về cơ bản, giải thuật BP là dạng tổng quát của thuật toán bình phương lỗi nhỏ nhất (Least Means Square), viết tắt là LMS. Thuật toán LMS thuộc dạng thuật toán xấp xỉ để tìm các điểm mà tại đó, hiệu năng của mạng là tối ưu. Chỉ số tối ưu (performance index) thường được xác định bởi một hàm số của ma trận trọng số và các đầu vào nào đó trong quá trình tìm hiểu bài toán đặt ra.

#### a. Mô tả giải thuật

Giải thuật áp dụng cho dạng tổng quát của mạng nơ ron truyền thẳng nhiều lớp. Khi đó, đầu ra của một lớp trở thành đầu vào của lớp kế tiếp. Phương trình thể hiện hoạt động này như sau:

$$a^{m+1} = f^{m+1}(W^{m+1}a^m + b^{m+1}) \text{ với } m = 0, 1, \dots, M-1$$

trong đó  $M$  là số lớp trong mạng. Các nơ ron trong lớp thứ nhất nhận các tín hiệu từ bên ngoài:  $a^0 = p$ , chính là điểm bắt đầu của phương trình trên. Đầu ra của lớp cuối cùng được xem là đầu ra của mạng:  $a = a^M$ .

*Chỉ số hiệu năng (performance index)*

Tương tự thuật toán LMS, giải thuật BP sử dụng chỉ số hiệu năng là trung bình bình phương lỗi của đầu ra so với giá trị đích. Đầu vào của thuật toán chính là tập các cặp mô tả hoạt động đúng của mạng (các mẫu dùng để huấn luyện mạng):

$$\{(p_1, t_1), (p_2, t_2), \dots, (p_Q, t_Q)\},$$

trong đó  $p_i$  là một đầu vào và  $t_i$  là đầu ra mong muốn tương ứng, với  $i = 1..Q$ . Mỗi đầu vào đưa vào mạng, đầu ra của mạng đối với nó được đem so sánh với đầu ra mong muốn. Thuật toán sẽ điều chỉnh các tham số của mạng để tối thiểu hóa trung bình bình phương lỗi:

$$F(x) = e[e^2] = e[(t - a)^2],$$

trong đó  $x$  là biến được tạo thành bởi các trọng số và độ lệch,  $e$  là ký hiệu kỳ vọng toán học. Nếu như mạng có nhiều đầu ra, phương trình trên có thể được viết lại dưới dạng ma trận:

$$F(x) = e[e^T e] = e[(t - a)^T (t - a)].$$

Tương tự như thuật toán LMS, xấp xỉ của trung bình bình phương lỗi như sau:

$$\hat{F}(\mathbf{x}) = (\mathbf{t}(k) - \mathbf{a}(k))^T (\mathbf{t}(k) - \mathbf{a}(k)) = \mathbf{e}^T(k) \mathbf{e}(k),$$

trong đó kỳ vọng toán học của bình phương lỗi được thay bởi bình phương lỗi tại bước  $k$ .

Thuật toán giảm theo hướng cho trung bình bình phương lỗi xấp xỉ là:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha \frac{\partial \hat{F}}{\partial w_{i,j}^m}, \quad (*)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial \hat{F}}{\partial b_i^m}, \quad (**)$$

trong đó  $\alpha$  là hệ số học.

Như vậy, mọi chuyện đến đây đều giống như thuật toán trung bình bình phương tối thiểu, tiếp theo sẽ đi vào phần khó nhất của thuật toán: tính các đạo hàm từng phần.

*Luật xích (Chain Rule):*

Đối với các mạng nơ ron truyền thẳng nhiều lớp, lỗi không phải là một hàm của chỉ các trọng số trong các lớp ẩn, do vậy việc tính các đạo hàm từng phần này là không đơn giản. Chính vì lý do đó mà phải sử dụng luật xích để tính. Luật này được mô tả như sau: giả sử có một hàm  $f$  là một hàm của biến  $n$ , muốn tính đạo hàm của  $f$  có liên quan đến một biến  $w$  khác. Luật xích này như sau:

$$\frac{df(n(w))}{dw} = \frac{df(n)}{dn} \mathbf{x} \frac{dn(w)}{dw}$$

Phương pháp này được dùng để tính các đạo hàm trong (\*) và (\*\*) ở phần trước

$$\begin{aligned} \frac{\partial \hat{F}}{\partial w_{i,j}^m} &= \frac{\partial \hat{F}}{\partial n_i^m} x \frac{\partial n_i^m}{\partial w_{i,j}^m}, \\ \frac{\partial \hat{F}}{\partial b_i^m} &= \frac{\partial \hat{F}}{\partial n_i^m} x \frac{\partial n_i^m}{\partial b_i^m}, \end{aligned}$$

trong đó hạng thức thứ 2 của các phương trình trên có thể dễ dàng tính toán bởi vì đầu vào của mạng tới lớp  $m$  là một hàm của trọng số và độ lệch:

$$n_i^m = \sum_{j=1}^{S^{m-1}} w_{i,j}^m a_j^{m-1} + b_i^m.$$

trong đó  $S^{m-1}$  là số đầu ra của lớp  $(m-1)$ . Do vậy:

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1}, \frac{\partial n_i^m}{\partial b_i^m} = 1.$$

Ký hiệu

$$s_i^m = \frac{\partial \hat{F}}{\partial n_i^m}$$

được gọi là *độ nhạy cảm* của  $\hat{F}$  đối với các thay đổi của phần tử thứ  $i$  của đầu vào của mạng tại lớp thứ  $m$ . Khi đó:

$$\frac{\partial \hat{F}}{w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} x \frac{\partial n_i^m}{\partial w_{i,j}^m} = s_i^m a_j^{m-1}$$

$$\frac{\partial \hat{F}}{b_i^m} = \frac{\partial \hat{F}}{\partial n_i^m} x \frac{\partial n_i^m}{\partial b_i^m} = s_i^m.$$

Bây giờ, thuật toán giảm nhanh nhất xấp xỉ được phát biểu như sau:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1},$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m$$

ở dạng ma trận:

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T,$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m$$

trong đó:

$$\mathbf{s}^m = \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial \hat{F}}{\partial n_1^m} \\ \frac{\partial \hat{F}}{\partial n_2^m} \\ \vdots \\ \frac{\partial \hat{F}}{\partial n_{s^m}^m} \end{bmatrix}$$

*Lan truyền ngược độ nhạy cảm*

Vấn đề là tính nốt ma trận độ nhạy cảm  $\mathbf{s}^m$ . Để thực hiện điều này cần sử dụng một áp dụng khác của luật xích. Quá trình này cho khái niệm về sự “lan truyền ngược” bởi vì nó mô tả mối quan hệ hồi quy trong đó độ nhạy cảm  $\mathbf{s}^m$  được tính qua độ nhạy cảm  $\mathbf{s}^{m+1}$  của lớp  $m+1$ .

Để dẫn đến quan hệ đó, ma trận *Jacobian* được sử dụng như sau:



$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \frac{\partial n_1^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_1^{m+1}}{\partial n_{s^m}^m} \\ \frac{\partial n_2^{m+1}}{\partial n_1^m} & \frac{\partial n_2^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_2^{m+1}}{\partial n_{s^m}^m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_1^m} & \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_{s^m}^m} \end{bmatrix}$$

Xét phần tử  $(i, j)$  của ma trận trên:

$$\begin{aligned} \frac{\partial n_i^{m+1}}{\partial n_j^m} &= \frac{\partial \left( \sum_{l=1}^{s^m} w_{i,l}^{m+1} a_l^m + b_i^{m+1} \right)}{\partial n_j^m} = w_{i,j}^{m+1} \frac{\partial a_i^m}{\partial n_j^m} \\ &= w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} = w_{i,j}^{m+1} \dot{f}^m(n_j^m) \end{aligned}$$

trong đó:

$$\dot{f}^m(n_j^m) = \frac{\partial f^m(n_j^m)}{\partial n_j^m}.$$

Như vậy, ma trận *Jacobian* có thể viết lại như sau:

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} = \mathbf{W}^{m+1} \dot{\mathbf{F}}^m(\mathbf{n}^m),$$

trong đó:

$$\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \dots & 0 \\ 0 & \dot{f}^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \dot{f}^m(n_{s^m}^m) \end{bmatrix}$$

Bây giờ viết lại quan hệ hồi quy cho độ nhạy cảm dưới dạng ma trận:

$$\begin{aligned} \mathbf{s}^m &= \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \left( \frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \right)^T \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} = \dot{\mathbf{F}}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} \\ &= \dot{\mathbf{F}}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}. \end{aligned}$$

Đến đây ta có thể thấy độ nhạy cảm được lan truyền ngược qua mạng từ lớp cuối cùng trở về lớp đầu tiên:

$$s^M \rightarrow s^{M-1} \rightarrow \dots (s^1).$$

Cần nhấn mạnh rằng ở đây thuật toán lan truyền ngược lỗi sử dụng cùng một kỹ thuật giảm theo hướng như thuật toán LMS. Sự phức tạp duy nhất ở chỗ để tính gradient cần phải lan truyền ngược độ nhạy cảm từ các lớp sau về các lớp trước như đã nêu trên.

Bây giờ cần phải biết điểm bắt đầu lan truyền ngược, xét độ nhạy cảm  $s^M$  tại lớp cuối cùng:

$$s_i^M = \frac{\partial \hat{F}}{\partial n_i^M} = \frac{\partial ((t-a)^T (t-a))}{\partial n_i^M} = \frac{\partial \sum_{l=1}^{s^M} (t_l - a_l)^2}{\partial n_i^M} = -2(t_i - a_i) \frac{\partial a_i}{\partial n_i^M}$$

Bởi vì:

$$\frac{\partial a_i}{\partial n_i^M} = \frac{\partial a_i^M}{\partial n_i^M} = \frac{\partial f^M(n_i^M)}{\partial n_i^M} = \dot{f}^M(n_i^M)$$

nên ta có thể viết:

$$s_i^M = -2(t_i - a_i) \dot{f}^M(n_i^M)$$

ở dạng ma trận sẽ là:

$$\mathbf{s}^M = -2 \dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}).$$

Tóm lại, thuật toán lan truyền ngược có thể phát biểu như sau:

**Bước 1:** Lan truyền xuôi đầu vào qua mạng:

$$\begin{aligned} a^0 &= p \\ a^{m+1} &= f^{m+1} (W^{m+1} a^m + b^{m+1}) \text{ với } m = 0, 1, \dots, M-1. \\ a &= a^M \end{aligned}$$

**Bước 2:** Lan truyền độ nhạy cảm (lỗi) ngược lại qua mạng:

$$\begin{aligned} s^M &= -2 \dot{F}^M (n^M) (t - a) \\ s^m &= \dot{F}^m (n^m) (W^{m+1})^T s^{m+1}. \quad \forall i \ m = M-1, \dots, 2, 1. \end{aligned}$$

**Bước 3:** Cuối cùng, các trọng số và độ lệch được cập nhật bởi công thức sau:

$$\begin{aligned} W^m(k+1) &= W^m(k) - \alpha s^m (a^{m-1})^T \\ b^m(k+1) &= b^m(k) - \alpha s^m \end{aligned}$$

## b. Sử dụng giải thuật

Trên đây là giải thuật BP dạng tổng quát. Phần tiếp theo sẽ trình bày các vấn đề về khía cạnh ứng dụng của giải thuật BP trong việc huấn luyện mạng nơ ron truyền thẳng nhiều lớp, như chọn lựa cấu trúc mạng, các hàm kích hoạt, các hàm giá, khởi động các trọng số ban đầu, tập mẫu học, sự hội tụ...

### *Chọn lựa cấu trúc mạng*

Vấn đề đầu tiên cần quan tâm là lựa chọn cấu trúc mạng. Như đã trình bày ở phần 3.1.1, mạng nơ ron truyền thẳng nhiều lớp luôn có một lớp vào và một lớp ra, số lớp ẩn có thể từ 0 đến vài lớp. Đối với một bài toán cụ thể, số nơ ron trên lớp vào cố định bằng số biến của véc tơ vào, số nơ ron trên lớp ra cố định bằng số biến của véc tơ đích. Vì vậy, vấn đề thiết kế cấu trúc mạng là vấn đề chọn *số lớp ẩn* và *số nơ ron trên mỗi lớp ẩn*

#### *Số lớp ẩn*

Bằng thực nghiệm đã chứng minh rằng đối với phần lớn các bài toán trong thực tế, chỉ cần sử dụng một lớp ẩn cho mạng [12]. Bởi vì:

1) Phần lớn các thuật toán huấn luyện cho các mạng nơon truyền thẳng nhiều lớp dựa trên phương pháp giảm gradient. Các lớp thêm vào sẽ thêm việc lan truyền các lỗi làm cho véc tơ gradient rất không ổn định, mà sự thành công của bất kỳ một thuật toán tối ưu theo gradient phụ thuộc vào độ không thay đổi của hướng khi mà các tham số thay đổi.

2) Số các cực trị địa phương tăng lên rất lớn khi có nhiều lớp ẩn. Phần lớn các thuật toán tối ưu dựa trên gradient chỉ có thể tìm ra các cực trị địa phương, do vậy chúng không thể tìm ra cực trị toàn cục. Mặc dù thuật toán luyện mạng có thể tìm ra cực trị toàn cục, nhưng xác suất bị tắc trong một cực trị địa phương là khá cao, và khi đó, ta phải bắt đầu luyện mạng lại.

3) Có thể đối với một bài toán cụ thể, sử dụng nhiều hơn một lớp ẩn với chỉ một vài nơ ron thì tốt hơn là sử dụng ít lớp ẩn với số nơ ron là lớn, đặc biệt đối với các mạng cần phải học các hàm không liên tục. Về tổng thể, người ta khuyên rằng nên xem xét khả năng sử dụng mạng có một lớp ẩn đầu tiên trong khi thiết kế các mạng truyền thẳng trong thực tế. Nếu dùng một lớp ẩn với một số lượng lớn các nơ ron mà không có hiệu quả thì nên sử dụng thêm một lớp ẩn với một số ít các nơ ron.

Tóm lại, mạng nơ ron truyền thẳng thường sử dụng một lớp ẩn để giải quyết các bài toán trong thực tế. Vấn đề thiết cấu trúc mạng quy về việc tìm ra *số nơ ron trong lớp ẩn*.

#### *Số nơ ron trong lớp ẩn*

Mạng nơ ron truyền thẳng có khả năng tổng quát hóa từ những dữ liệu mà nó đã học, nói cách khác, mạng nơ ron truyền thẳng có khả năng dự báo tốt. Khả năng dự báo phụ thuộc nhiều vào số nơ ron trong lớp ẩn, hay phụ thuộc nhiều vào số lượng các trọng số của mạng.

Một mạng có số lượng các trọng số lớn với một số lượng nhỏ các mẫu học có thể học rất tốt, song việc dự báo có thể không tốt, hiện tượng này thường gọi là *học quá* (overfitting). Nói cách khác, khi số lượng trọng số lớn hơn số mẫu học, kết quả luyện tham số không phản ánh đúng hoạt động của mạng. Để tránh trường hợp *học*

quá, số trọng số trong các mạng nơ ron phải nhỏ hơn hoặc tương đương với số mẫu có trong tập mẫu [12].

Nếu số lượng các trọng số quá nhỏ, mạng nơ ron có thể không nhận dạng được đầy đủ các tín hiệu trong một tập dữ liệu phức tạp, còn gọi là hiện tượng thiếu ăn khớp (*underfitting*)

Số lượng tốt nhất của các nơ ron ẩn phụ thuộc vào rất nhiều yếu tố, đó là số đầu vào, số đầu ra của mạng, số các mẫu khác nhau trong tập mẫu, độ nhiễu của dữ liệu đích, độ phức tạp của hàm lỗi, kiến trúc mạng, và thuật toán huấn luyện mạng...

Có rất nhiều “luật” để lựa chọn số nơ ron trong các lớp ẩn [6, 12]:

- $m \in [l, n]$  - nằm giữa khoảng kích thước lớp vào, lớp ra
- $m = \frac{2(l+n)}{3}$  - 2/3 tổng kích thước lớp vào và lớp ra
- $m < 2l$  - nhỏ hơn 2 lần kích thước lớp vào
- $m = \sqrt{l \cdot n}$  - căn bậc hai của tích kích thước lớp vào, lớp ra.

Các luật này chỉ mang tính lý thuyết mà không phản ánh được thực tế bởi vì, chúng chỉ xem xét đến nhân tố kích thước đầu vào, đầu ra mà bỏ qua các nhân tố quan trọng khác như số lượng mẫu đưa vào huấn luyện, độ nhiễu ở các đầu ra, độ phức tạp của hàm lỗi ...

### ***Các hàm kích hoạt***

Các hàm kích hoạt của lớp ẩn dùng trong mạng nơ ron truyền thẳng nhiều lớp thông dụng nhất là hàm sigmoid . Ngoài ra, một số hàm kích hoạt ở bảng dưới đây có tính chất tương tự như hàm sigmoid có thể được sử dụng:

$f(x)$	$f'(x)$	$f'(f(x))$
$\frac{1}{1+e^{-x}}$	$\frac{-e^{-x}}{(1+e^{-x})^2}$	$f(x)(1-f(x))$
$\frac{1-e^{-x}}{1+e^{-x}}$	$\frac{2e^{-x}}{(1+e^{-x})^2}$	$1 - (f(x))^2$
$\frac{x}{1+x}$	$\frac{1}{(1+x)^2}$	$(1 - f(x))^2$
$1+e^{-x}$	$e^{-x}$	$1 - f(x)$

Bảng 3.1. Các hàm kích hoạt

Đối với lớp ra, các hàm kích hoạt phải được chọn sao cho phù hợp với sự phân phối của các giá trị đích mong muốn. Ta thấy rằng, đối với các giá trị đích trong khoảng  $[0,1]$ , hàm sigmoid là có ích, đối với các giá trị đích mong muốn liên tục trong khoảng đó thì hàm này vẫn có nghĩa. Nhưng nếu giá trị đích không biết trước khoảng xác định thì hàm hay được sử dụng nhất là hàm tuyến tính (xem phần 1.2 chương 1). Nếu giá trị đích mong muốn là dương nhưng không biết cận trên thì một hàm kích hoạt dạng mũ có thể được sử dụng.

### Các hàm giá

Hàm giá bình phương được định nghĩa theo phương trình 3.1 không phải là sự lựa chọn duy nhất có thể. Số hạng bình phương sai số  $(y_{jk} - Out_{jk}^{last})^2$  có thể được thay thế bằng bất cứ hàm có khả năng đạo hàm nào khác  $F(y_{ik}, out_{jk})$ , với điều kiện là các hàm này sẽ đạt cực tiểu khi hai đối số  $y_{ik}$  và  $out_{jk}$  bằng nhau. Dựa vào dạng của hàm giá mới có thể thu nhận được các phương trình khác nhau cho việc hiệu chỉnh các trọng số. Một điều dễ nhận thấy là chỉ có phương trình hiệu chỉnh trọng số ở lớp ra là thay đổi theo các dạng hàm giá khác nhau, trong khi đó các phương trình khác của giải thuật BP vẫn giữ nguyên.

Các hàm giá thường được sử dụng là những hàm dựa trên chuẩn  $L_p$  ( $1 \leq p \leq \infty$ ) bởi sự lợi thế của dạng phương trình toán học đơn giản. Các hàm như vậy có dạng là:

$$E = \frac{1}{p} \sum_{k=1}^n |y_{jk} - Out_{jk}^{last}|^p \quad \text{với} \quad 1 \leq p \leq \infty \quad (3.3)$$

Trường hợp hàm giá định nghĩa theo phương trình (3.1) là trường hợp riêng với  $p = 2$  của phương trình (3.2).

#### **Khởi động các trọng số ban đầu**

Giá trị các trọng số khởi tạo ban đầu của mạng khi áp dụng giải thuật BP ảnh hưởng rất mạnh tới lời giải cuối cùng. Việc khởi tạo tất cả các trọng số bằng nhau sẽ làm cho việc học của mạng trở nên không tốt. Các trọng số khởi tạo ban đầu cùng không được quá lớn vì điều này làm cho hàm sigmoid hoặc các hàm dạng này sẽ bị bão hòa (hoặc bằng 0, hoặc bằng 1) ngay từ lúc bắt đầu, làm cho hệ thống bị tắc tại một cực trị địa phương gần điểm xuất phát. Do vậy, các trọng số ban đầu thường được khởi tạo bằng những số ngẫu nhiên nhỏ. Giá trị khởi động ban đầu của các trọng số trên lớp thứ nhất thường được chọn ngẫu nhiên trong khoảng  $[-1/n, 1/n]$ , trong đó  $n$  là tổng trọng số trên lớp thứ nhất [12].

#### **Tập mẫu học**

Tập mẫu học phải đủ và đúng. Tuy nhiên, không có một thủ tục hay một nguyên tắc chọn tập mẫu học phù hợp cho mọi trường hợp. Một nguyên tắc theo kinh nghiệm là tập mẫu học phải được chọn sao cho chúng bao phủ toàn bộ không gian của tín hiệu vào. Trong quá trình học, các cặp véc tơ vào - lời giải nên được chọn ngẫu nhiên từ tập học. Các tập mẫu học sử dụng để huấn luyện mạng phải được tỷ lệ hóa đúng đắn, chi tiết về việc tỷ lệ hóa sẽ được đề cập trong chương 4, phân tổ chức số liệu.

#### **Sự hội tụ**

Trong thuật toán LMS, điểm cực trị toàn cục là luôn tồn tại bởi lẽ hàm trung bình bình phương lỗi của LMS là một hàm bậc hai, hơn nữa, do là hàm bậc hai nên đạo hàm bậc hai của hàm lỗi sẽ là hằng số do vậy mà độ cong của hàm theo một

hướng cho trước là không thay đổi. Trong khi đó, giải thuật BP áp dụng cho các mạng nhiều lớp sử dụng các hàm chuyển phi tuyến sẽ có nhiều điểm cực trị địa phương và độ cong của hàm lỗi có thể không cố định theo một hướng cho trước, việc thực hiện thuật toán có thể đưa mạng hội tụ về điểm cực trị địa phương, làm cho mạng không trả lại kết quả chính xác.

Một nhân tố khác ảnh hưởng tới hiệu lực và độ hội tụ của giải thuật là hệ số học  $\alpha$ . Không có một giá trị duy nhất đủ tốt phù hợp với tất cả các bài toán khác nhau, hệ số học này thường được chọn bằng thực nghiệm cho mỗi bài toán cụ thể bằng phương pháp thử sai. Giá trị  $\alpha$  lớn làm tăng tốc độ hội tụ nhưng không có lợi vì thủ tục học sẽ kết thúc rất nhanh tại một cực trị địa phương gần nhất. Nếu giá trị của  $\alpha$  quá nhỏ, tốc độ hội tụ của giải thuật lại trở nên chậm. Do đó, cần chọn một giá trị thỏa hiệp giữa tốc độ học và việc ngăn chặn hội tụ về các cực trị đại phương. Các giá trị  $\alpha$  nằm trong khoảng  $10^{-3}$  và  $10$  đã được sử dụng thành công cho rất nhiều bài toán cụ thể. Nói chung, giá trị hằng số học nằm trong khoảng  $[0.3, 0.6]$  được chứng minh bằng thực nghiệm là khá tốt cho việc chọn hệ số học ban đầu. Vấn đề nảy sinh là giá trị tốt của hệ số học tại thời điểm bắt đầu có thể là không tốt cho giai đoạn sau của quá trình học. Do đó, một phương pháp cải tiến là sử dụng hệ số học biến đổi sẽ được trình bày ở phần tiếp theo.

### 3.2.2. Một số cải tiến của giải thuật BP

Bản chất của giải thuật BP là giải thuật tìm kiếm sử dụng kỹ thuật tìm kiếm *ngược hướng gradient*. Mặc dù dễ thực thi nhưng giải thuật này bộc lộ một số nhược điểm như sau:

- 1) Tại các vùng hoặc một số hướng mà bề mặt sai số bằng phẳng, các giá trị gradient nhỏ dẫn đến tốc độ hội tụ của giải thuật chậm [12].
- 2) Bề mặt sai số trong đa số các bài toán thường không “lồi” mà có nhiều vùng “lồi” khác nhau, nó phụ thuộc vào quá trình khởi tạo các trọng số ban đầu của mạng, điều đó dẫn đến giải thuật có thể bị tắc tại các cực trị địa phương (tắc tại các vùng “lồi”).



3) Các hàm kích hoạt của nơ ron và hàm giá tính tín hiệu sai số phải khả vi [22]. Điều này là điểm bất lợi trong các ứng dụng sử dụng các hàm ngưỡng làm hàm kích hoạt do tính không khả vi của chúng.

4) Hiệu năng tìm kiếm của giải thuật phụ thuộc vào các tham số luyện như số nơ ron trên lớp ẩn (tham số cấu trúc), giá trị các trọng số khởi tạo ban đầu, hằng số học  $\alpha$ ... Việc xác định các giá trị của chúng để đưa tới tình thế tiến thoái lưỡng lan giữa tốc độ hội tụ và sự dao động trong quá trình tìm kiếm.

Có rất nhiều các nghiên cứu đã đề xuất các cải tiến nhằm khắc phục các nhược điểm trên như sử dụng tham số bước đà, sử dụng hệ số học biến đổi, sử dụng gradient kết hợp, sử dụng thuật toán giả luyện kim, sử dụng giải thuật di truyền, ...

Luận văn nghiên cứu giải pháp tích hợp giải thuật GA với giải thuật BP như một giải thuật lai sử dụng để huấn luyện mạng nơ ron. Do sử dụng giải pháp này mà giải thuật BP sử dụng trong giải thuật lai chỉ sử dụng phương pháp cải tiến hệ số học biến đổi.

### ***Phương pháp sử dụng hệ số học biến đổi***

Trong thực tế, các hàm hiệu năng có dạng biểu diễn hình học là không đồng đều, có lúc có dạng phẳng (hàm không thay đổi giá trị hoặc thay đổi rất ít) hoặc có dạng phễu (giá trị của hàm thay đổi rất nhanh khi thay đổi tham số đầu vào). Nếu ta chỉ sử dụng hệ số học cố định thì có thể sẽ tốn thời gian tại các vùng phẳng. Vì vậy, ý tưởng của giải thuật BP sử dụng hệ số học biến đổi là khi gặp vùng phẳng thì tăng hệ số học lên và ngược lại khi gặp vùng dạng phễu thì giảm hệ số học đi.

Người ta đã đưa ra rất nhiều phương pháp để thực hiện giải pháp trên, ở đây chỉ xin nêu ra một cách biến đổi hệ số học dựa trên hiệu năng của mạng [8]:

*Bước 1:* Nếu bình phương lỗi trên toàn bộ tập huấn luyện tăng một số phần trăm cho trước  $\xi$  (thông thường là từ 1% cho đến 5%) sau một lần cập nhật trọng số thì bỏ qua việc cập nhật này, hệ số học được nhân với một số hạng  $\rho$  nào đó (với  $0 < \rho < 1$ ).

*Bước 2:* Nếu bình phương lỗi giảm sau một lần cập nhật trọng số, thì cập nhật đó là chấp nhận được và hệ số học được nhân với một số hạng nào đó  $> 1$ .

*Bước 3:* Nếu bình phương lỗi tăng một lượng  $< \xi$  thì cập nhật trọng số là chấp nhận được nhưng hệ số học không thay đổi.

### 3.3. KẾT HỢP GIẢI THUẬT DI TRUYỀN VỚI GIẢI THUẬT BP

#### 3.3.1. Giải thuật di truyền trong việc huấn luyện mạng nơon truyền thẳng nhiều lớp

Để có thể sử dụng giải thuật GA trong thủ tục huấn luyện mạng (hiệu chỉnh các trọng số của mạng), cần phải giải quyết các vấn đề về xây dựng hàm giá, mã hoá và giải mã các trọng số, khởi động quần thể đầu tiên, xác định các tham số của giải thuật GA.

##### *Xây dựng hàm giá*

Giải thuật BP xây dựng một hàm giá duy nhất (hàm giá 3.2). Đối với giải thuật GA, với một tập học cho trước  $(X_s, y_s)$  cần lan truyền lần lượt các véc tơ vào và tích lũy các sai số lại thành một sai số tổng thể cho tập học đó:

$$E(w) = \frac{1}{2} \sum_{s=1}^p \sum_{i=1}^n (y_{si} - Out_{si}^{last})^2 \quad (3.4)$$

Trong đó:  $y_{si}$  là thành phần thứ  $i$  của đầu ra mong muốn

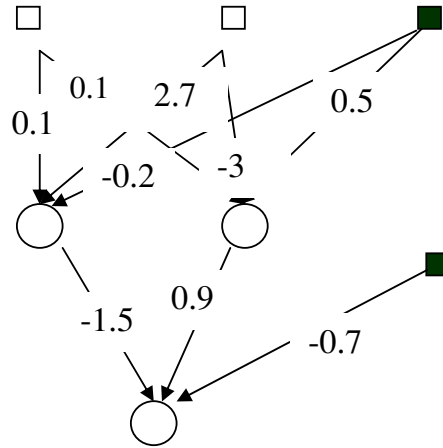
$out_{si}^{last}$  là thành phần thứ  $i$  của đầu ra thực tế

Giải thuật GA sẽ tìm kiếm tập trọng số  $W$  trong không gian  $R^M$  để tối thiểu hàm giá 3.4.

##### *Mã hóa và giải mã các trọng số*

Như đã đề cập ở chương 2, các toán tử của giải thuật GA hoạt động ở mức cuối đối với các bit. Do đó, một tập các trọng số của một cấu trúc mạng phải được mã hóa thành một chuỗi.

Đối với một cấu trúc mạng cho trước, các trọng số được sắp xếp thành một danh sách. Hình 3.3 là một ví dụ về việc sắp xếp như vậy.



Hình 3.3: Sơ đồ mã hóa các trọng số của mạng nơ ron

(0.1, 2.7, -0.2, 0.1, -3, 0.5, -1.5, 0.9, -0.7)

Giả sử có một mạng nơ ron truyền thẳng với  $L$  lớp,  $m$  và  $n$  lần lượt là số nơ ron trên lớp vào và lớp ra. Trọng số  $w_{ji}^l$  thứ  $i$  của nơ ron thứ  $j$  trên lớp  $l$  sẽ chiếm vị trí thứ  $k$  trong danh sách theo công thức sau:

$$k = \left[ \sum_{s=1}^{l-1} N(s)(N(s-1)+1) \right] + (j-1)[N(l-1)+1] + i \quad (3.5)$$

Danh sách này sau đó được mã hóa tiếp thành một chuỗi (một cá thể). Do đó một chuỗi mô tả một tập các trọng số. Chú ý rằng, tất cả các chuỗi trong quần thể đều mô tả các tập trọng số của cùng một cấu trúc mạng.

Danh sách được mã hóa thành chuỗi nhị phân như sau: mỗi trọng số được mã hóa thành một chuỗi con có độ dài 20 bit với giá trị nằm trong khoảng  $[-10, 10]$  để tránh trường hợp các giá trị hoạt hóa của các nơ ron bị bão hòa [3]. Các chuỗi con 20 bit của các trọng số được nối với nhau tạo thành một chuỗi dài. Chuỗi dài này đại diện cho một tập trọng số của một cấu trúc mạng.

Việc giải mã các trọng số từ một chuỗi dài là việc cắt chuỗi dài thành các chuỗi con 20 bit. Giả sử giá trị của chuỗi nhị phân con được cắt ra là  $x$ , lúc này giá trị của trọng số tương ứng với chuỗi con đó là  $20 \cdot x / (2^{20} - 1) - 10$ .

Để đánh giá sức khỏe của một chuỗi trong quần thể, giải mã chuỗi thành tập trọng số của mạng nơ ron có cấu trúc đã định trước. Sau đó, lan truyền toàn bộ các

mẫu có trong tập huấn luyện, tính giá trị của hàm giá theo công thức 3.3. Nói cách khác, mạng nơ ron đóng vai trò như một hàm định giá trong giải thuật GA.

### ***Khởi động quần thể đầu tiên***

Các trọng số của các cá thể trong quần thể ban đầu được chọn ngẫu nhiên trong khoảng  $[-10, 10]$ . Các trọng số này được khởi động ngẫu nhiên với xác suất cho bởi phân bố  $e^{-x}$ . Xác suất này được rút ra từ các quan sát trong thực nghiệm là lời giải tối ưu có phần lớn các trọng số với giá trị tuyệt đối nhỏ và đồng thời chúng cũng có số ít các trọng số có giá trị tuyệt đối lớn. Do đó, việc khởi động các trọng số sử dụng phân bố xác suất này cho phép giải thuật GA thăm dò các khoảng chứa tất cả các lời giải có thể, đồng thời nó hướng giải thuật vào những vùng hay chứa lời giải nhất.

### ***Các tham số của giải thuật di truyền***

Các tham số của giải thuật GA như xác suất tạp lai, xác suất đột biến, số cá thể trong quần thể và số thế hệ được chọn theo phương pháp thử và sai. Các giá trị  $p_{\text{cross}}=0.7$ ,  $p_{\text{mutation}}=0.001$  được coi là các giá trị xuất phát khá tốt. Nếu giá trị  $p_{\text{mutation}}$  lớn, giải thuật GA trở thành giải thuật tìm kiếm ngẫu nhiên. Số cá thể trong quần thể thường được chọn cỡ trung bình từ 100 đến 200 cá thể cho một quần thể. Số thế hệ cần thiết theo thực nghiệm cỡ 5 đến 10 lần tổng số trọng số có trong mạng.

### **3.3.2. Ghép nối với giải thuật lan truyền ngược sai số**

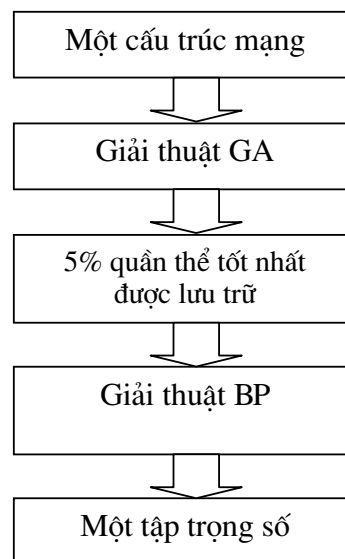
Giải thuật GA có thể tìm ra vùng chứa cực trị toàn cục, song nó không có khả năng leo lên đỉnh của cực trị đó. Nói cách khác, giải thuật GA không bảo đảm sự hội tụ. Trái lại, giải thuật BP đảm bảo cho sự hội tụ nhưng không có khả năng tìm kiếm cực trị toàn cục. Do đó, việc kết hợp hai giải thuật trên trong một giải thuật lai là lẽ tự nhiên.

Đối với một cấu trúc mạng cho trước, xuất phát bằng giải thuật GA với việc khởi động một quần thể ban đầu gồm  $N$  chuỗi nhị phân ( $N$  cá thể). Mỗi chuỗi là một bản mã nhị phân của một tập trọng số của một cấu trúc mạng đã cho. Giải thuật GA tiến hành tiến hóa quần thể ban đầu bằng cách sử dụng các toán tử chọn lọc, lai ghép và đột biến. Sau  $G$  thế hệ, 5% cá thể tốt nhất trong  $G$  được lưu trữ lại. Như vậy, đầu vào của giải thuật GA là một cấu trúc mạng và tập huấn luyện, đầu ra là  $0.05*N$  tập

trọng số. Các tập trọng số này lần lượt được giải thuật BP luyện đến bão hòa. Kết quả của quá trình luyện bằng giải thuật BP là  $0.05 \cdot N$  tập trọng số mới. Tập trọng số cho kết quả tốt nhất (giá trị hàm giá nhỏ nhất) trong  $0.05 \cdot N$  tập trọng số này được giữ lại là kết quả của giải thuật lai GA - BP. Giải thuật BP trong giải thuật lai này là giải thuật BP đã trình bày ở phần 3.1 với:

- Các giá trị trọng số ban đầu không cần phải khởi động mà tiếp nhận tập trọng số là kết quả từ giải thuật GA như tập trọng số ban đầu.
- Giải thuật BP sử dụng hằng số học biến đổi để đảm bảo giá trị của các hàm giá 3.3 luôn giảm. Nói một cách khác, làm tăng tốc độ hội tụ của giải thuật.

Hình 3.4 là sơ đồ khối tổng thể của giải thuật lai GA - BP. Giải thuật lai này được dùng trong thủ tục huấn luyện mạng nơ ron truyền thẳng nhiều lớp.



Hình 3.4: Sơ đồ của giải thuật lai

### ❖ Kết luận chương 3

Chương 3 mô tả giải thuật BP và các vấn đề khi sử dụng giải thuật BP trong huấn luyện mạng nơ ron truyền thẳng nhiều lớp như lựa chọn cấu trúc mạng, hàm kích hoạt, xây dựng hàm giá, khởi tạo tập trọng số ban đầu, ... Chương 3 cũng trình bày giải pháp tích hợp giải thuật GA và BP thành một giải thuật lai để học tham số cho mạng nơ ron.

Cùng với thực tế mạng nơon được ứng dụng rộng rãi trong lĩnh vực dự báo dữ liệu, đặc biệt là các bài toán dự báo tiêu thụ năng lượng, dự báo kinh tế, dự báo các hiện tượng tự nhiên.... Chương 4 của luận văn sẽ thực hiện cài đặt thử nghiệm chương trình dự báo lũ trên sông Trà Khúc sử dụng mạng nơon truyền thẳng huấn luyện bằng giải thuật lai GA – BP.

## **CHƯƠNG 4:**

# **ỨNG DỤNG TRONG BÀI TOÁN DỰ BÁO DỮ LIỆU**

### **4.1. GIỚI THIỆU BÀI TOÁN**

Dự báo đỉnh lũ trên sông là một trong những bài toán quan trọng trong lĩnh vực dự báo thủy văn, nó có ý nghĩa to lớn trong đời sống xã hội vì nó giúp con người dự báo được các trận lũ lớn trước một thời gian dài, tránh được thiệt hại về người và vật chất do chúng gây ra.

Dòng chảy sông suối được hình thành dưới ảnh hưởng của nhiều nhân tố. Song trong số đó nổi lên hai nhân tố quan trọng là lượng mưa và lượng trữ nước trên lưu vực sông. Mưa là nhân tố quyết định độ lớn của đỉnh lũ, tuy nhiên, cùng một lượng mưa trên cùng một lưu vực, vẫn có thể sinh ra các đỉnh lũ khác nhau. Ví dụ, trên sông Hồng lượng mưa sinh ra trận lũ lớn nhất năm 1969 và 1996 tương ứng là 250 và 300 mm, lớn hơn lượng mưa gây trận lũ tháng 8/1971 là 218 mm, song do lượng trữ nước tại thời điểm trước lũ năm 1971 lớn hơn đã làm cho đỉnh lũ tháng 8/1971 lớn hơn nhiều so với hai trận lũ kia. Như vậy, lượng trữ nước trước lũ, hay gọi là chân lũ, có thể xem là nhân tố quan trọng thứ hai, quyết định độ lớn của đỉnh lũ. Ngoài ra còn có các yếu tố khác tác động đến lũ lụt như điều kiện thời tiết... chúng chỉ là các nhân tố gián tiếp.

Sông Trà Khúc bắt nguồn từ vùng rừng núi Giá Vực, phía tây nam tỉnh Quảng Ngãi, ở vào khoảng 14°34'30''B và 108°25'20''Đ. Độ cao nguồn sông khoảng 900 m, chiều dài sông 135 km, chiều dài lưu vực 123 km, diện tích lưu vực 3240 km<sup>2</sup>, độ dốc lưu vực 18,5%, chiều rộng lưu vực 26,3 km. Có hai dạng lũ trên sông, lũ đơn và lũ kép.

Luận văn xây dựng chương trình dự báo dữ liệu sử dụng mạng nơ ron truyền thẳng huấn luyện bằng giải thuật lai GA - BP được thử nghiệm với bài toán dự báo đỉnh lũ sông Trà Khúc trạm Sơn Giang.

Số liệu huấn luyện mạng và kiểm tra khả năng dự báo của mạng được lấy từ Trung tâm Thông tin tư liệu - Tổng cục Khí tượng Thủy văn, là số liệu đo được tại trạm Sơn Giang từ năm 2001 đến nay và được lưu trữ dưới dạng sau:

Năm	Thời gian		Lượng mưa trung bình	Mức nước lũ trung bình	
	Bắt đầu	Kết thúc		Chân lũ	Đỉnh lũ
2001	1h/6/10	13h/6/10	191.5	2831	3352
	1h/7/10	13h/7/10	184.5	3088	3594
	19h/9/10	13h/10/10	118.5	3041	3414
2002	7h/11/10	13h/11/10	74.5	3185	3340
	1h/9/11	19h/10/11	289	3025	3717
	7h/22/10	7h/23/10	199	2931	3449
	1h/12/9	13h/12/9	67	2820	3084
	7h/2/11	1h/3/11	298	3077	4020
	19h/17/10	7h/18/10	82	2955	3203
	1h/25/10	13h/25/10	121.5	3143	3578
	9h/28/10	19h/28/10	62	3159	3382
	11h/29/10	11h/29/10	84.5	3312	3548
	7h/16/11	19h/16/11	173.5	3112	3643
	1h/19/11	7h/19/11	95.5	3362	3585
	21h/19/11	7h/20/11	121	3433	3615
	7h/30/11	19h/30/11	150.5	3097	3572
	21h/30/11	3h/1/12	60	3519	3710
	7h/19/12	3h/20/12	165.5	3004	3451
	.....				

*Bảng 4.1: Số liệu thử nghiệm của bài toán dự báo*

Trong đó:

- Năm: là năm lấy mẫu số liệu, không tham gia vào dữ liệu dự báo
- Thời gian: là khoảng thời gian đo số liệu, không tham gia vào số liệu dự báo



- Lượng mưa trung bình: là lượng mưa trung bình đo được trong khoảng thời gian trên tính bằng mm, là một đầu vào của dữ liệu dự báo
- Mức nước chân lũ: là giá trị mức nước chân lũ tính bằng cm, là đầu vào thứ hai của dữ liệu dự báo
- Mức nước đỉnh lũ: là giá trị mức nước đỉnh lũ tính bằng cm, là giá trị dự báo.

## 4.2. MÔ HÌNH HOÁ BÀI TOÁN, THIẾT KẾ DỮ LIỆU VÀ GIẢI THUẬT

### 4.2.1. Mô hình hoá bài toán

#### *Tiền xử lý:*

Với dữ liệu đã cho, có thể thiết lập mô hình gồm có ba hiệu ứng sau:

- Lượng mưa trung bình: nhận giá trị thực của nó.
- Mức nước chân lũ: nhận giá trị thực của nó.
- Mức nước đỉnh lũ: nhận giá trị thực của nó.

Tất cả các dữ liệu đưa vào mạng sẽ được chuẩn hóa về khoảng (0,1) theo công thức:  $SV = OV \cdot (0.9 - 0.1) / (MAX - MIN)$  (4.1)

trong đó:

OV: Giá trị trước khi biến đổi

SV: Giá trị sau khi biến đổi (giá trị đưa vào mạng)

MAX, MIN: Giá trị lớn nhất và nhỏ nhất của tập giá trị

0.9, 0.1: Giá trị lớn nhất và nhỏ nhất của hàm sigmoid

#### *Mô hình dự báo:*

Ta dùng các ký hiệu sau:

X: Lượng mưa trung bình

$H_c$ : Mức nước chân lũ

$H_d$ : Mức nước đỉnh lũ

Như vậy, mô hình dự báo mức nước đỉnh lũ theo mức nước chân lũ và lượng mưa trung bình được biểu diễn bằng hàm số:  $H_d = f(H_c, X)$  (4.2)

**Lựa chọn kiến trúc mạng:**

Mạng bao gồm một lớp ra và một lớp ẩn. Đầu vào của mạng là lượng mưa trung bình  $X$  và mực nước chân lũ  $H_c$ , đầu ra của mạng là giá trị dự báo mực nước đỉnh lũ  $H_d$ .

Mạng sẽ yêu cầu số nơ ron trong lớp ẩn vừa đủ để học được các đặc trưng tổng quát về mối quan hệ giữa đầu vào và đầu ra. Mục tiêu là sử dụng số nơ ron trong lớp ẩn càng ít càng tốt. Số nơ ron trong lớp ẩn được xác định bằng cách huấn luyện với một số tập kiểm tra.

Hàm kích hoạt của các nơ ron trong lớp ẩn là hàm sigmoid. Hàm kích hoạt của các nơ ron ở lớp ra chọn là hàm đồng nhất.

**4.2.2. Thiết kế dữ liệu*****Giải thuật di truyền***

Các toán tử của giải thuật GA hoạt động ở mức chuỗi nên cấu trúc dữ liệu cơ bản là quần thể các chuỗi. Một trong những cấu trúc dữ liệu sử dụng là bảng hai chiều với mỗi hàng là một cá thể và số cột là độ dài của mỗi cá thể. Do độ dài của mỗi cá thể và số cá thể thường xuyên biến động nên bảng hai chiều được cấp phát động. Hai quần thể cũ và mới được định nghĩa là hai con trỏ chỉ đến hai bảng hai chiều có kích thước động Oldpop( ) và NewPop( ).

Đồng thời với quần thể các cá thể là hai véc tơ được cấp phát động của các số thực nhằm ghi nhận giá trị của hàm mục tiêu tương ứng với các cá thể và giá trị sức khỏe tương ứng : Objective( ) và Fitness( ).

Các biến Popsizes ghi số cá thể trong quần thể, Pcross ghi xác suất tạp lai, Pmutation ghi xác suất đột biến, Gen ghi số thế hệ cần tiến hóa và độ dài chuỗi là Lchrom.

***Mạng nơ ron***

Mạng nơ ron truyền thẳng được cài đặt là một lớp có tên gọi là Network, các tham số của mạng là các biến thành viên; NumInputs, NumOutputs, NumNeurals tương ứng là số đầu vào, số đầu ra, số nơ ron trên lớp ẩn, Inputs( ) và Expected\_Outputs( ) là hai véc tơ chứa đầu vào và đầu ra mong muốn của mạng, Layers( ) là véc tơ có kiểu phần tử thuộc lớp Layer chứa các lớp mạng. Lớp Layer có

biến thành viên là hai véc tơ Inputs( ), Outputs( ) và Output\_Errors( ) chứa đầu vào, đầu ra và sai số đầu ra của lớp.

Để tích hợp giải thuật GA với giải thuật BP cần sử dụng một bảng hai chiều cấp phát động GA\_Weights( ) để lưu trữ các trọng số của mạng tại mỗi thế hệ tiến hóa, số cột là số trọng số của mạng, số hàng là số cá thể trong quần thể. Mỗi hàng của bảng tương ứng với một bộ trọng số của mạng, việc đưa vào mạng bộ trọng số này được thực hiện nhờ thủ tục GA\_loadWeight( ) của lớp network.

Ngoài ra còn sử dụng bảng hai chiều cấp phát động BP\_weights( ) để lưu trữ  $0.05 \cdot N$  bộ trọng số kết quả của giải thuật GA sau Gen thế hệ tiến hóa làm đầu vào cho giải thuật BP.

#### **Số liệu mẫu và tổ chức số liệu:**

Số liệu thực nghiệm được tổ chức trong một tệp số liệu. Các cặp véc tơ tín hiệu vào và tín hiệu ra được viết trên một dòng. Do hàm biến đổi dùng trong mạng là hàm sigmoid nên các số liệu này sẽ được chương trình tự động tỷ lệ hóa tuyến tính trong khoảng [0.1, 0.9] theo công thức (4.1). Tập dữ liệu sau khi đã được tỷ lệ hóa như trên được lưu trữ trong hai véc tơ cấp phát động Inputs( ) và Expected\_outputs( ).

#### **4.2.3. Thiết kế giải thuật**

Sơ đồ chính của chương trình như sau:

- Vào:
  - Tên file chứa số liệu mẫu.
  - Cấu trúc mạng nơ ron (m, n, a).
  - Số thế hệ cần tiến hóa Gen.
- Ra:
  - Tập trọng số ứng với cấu trúc mạng trên.
  - Sai số của mạng trên.
- Giải thuật:
  - Tiền xử lý số liệu bằng việc tỷ lệ hóa tập huấn luyện (**Thủ tục 1**).
  - Học tham số bằng giải thuật di truyền (**Phân hệ 1**)

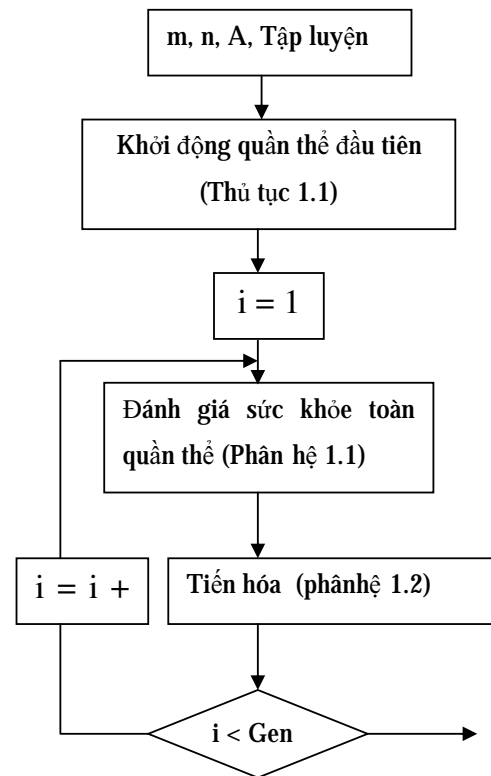
- Lưu trữ 5% cá thể tốt nhất từ quần thể cuối cùng.
- Học tham số bằng giải thuật BP với hằng số học thích nghi cho từng cá thể trong 5% cá thể từ giải thuật di truyền chuyển sang (**phân hệ 2**)
- Tập trọng số của cá thể tốt nhất sau giai đoạn học bằng giải thuật BP được giữ lại như kết quả của chương trình.

### **Thủ tục 1**

- *Chức năng:*
  - Tỷ lệ hóa tuyến tính tập huấn luyện vào khoảng [0.1, 0.9]
- *Vào:*
  - Tập mẫu huấn luyện
- *Ra:*
  - Giá trị hai tập con đã được tỷ lệ hóa Xtrain( ), Ytrain( ) với Xtrain( ) là véc tơ đầu vào và Ytrain( ) là véc tơ đầu ra mong muốn .
  - Số lượng mẫu có trong tập P.
- *Giải thuật:*
  - Xác định số lượng mẫu có trong tập P.
  - Xác định số biến của tín hiệu vào m và tín hiệu ra n.
  - Lặp i = 1 đến P
    - Lặp j = 1 đến m + n
      - +  $Scale[j] = (0.9-0.1) / (\max[j] - \min[j])$ .
      - +  $Xtrain[i,j] = (\text{input}[i,j] - \min[j]) * Scale[j] + 0.1$ .
      - +  $Ytrain[i,j] = (\text{Target}[i,j] - \min[j]) * Scale[j] + 0.1$

**Phân hệ 1**

- *Chức năng:*
  - Sử dụng giải thuật di truyền để huấn luyện (học tham số) mạng nơ ron truyền thẳng nhiều lớp.
- *Vào:*
  - Cấu trúc mạng m, n, a.
  - Tập mẫu luyện.
- *Ra:*
  - Quần thể các cá thể của thế hệ cuối cùng, mỗi cá thể là một bộ trọng số của mạng.
- *Giải thuật:*
  - Khởi động quần thể đầu tiên (**Thủ tục 1.1**)
  - Lặp i = 1 đến Gen
    - + Đánh giá sức khỏe toàn quần thể (**Phân hệ 1.1**)
    - + Tiến hóa từ thế hệ cũ sang thế hệ mới (**phân hệ 1.2**)

Hình 4.1: Sơ đồ khối giải thuật **Phân hệ 1****Thủ tục 1.1**

- *Chức năng:*
  - Sản sinh một bảng OldPop với Popsiz dòng là Popsiz chuỗi nhị phân, mỗi chuỗi là bảng mã của một tập các trọng số của mạng.
  - Các trọng số được khởi tạo ngẫu nhiên trong khoảng  $[-10, 10]$  tuân theo xác suất  $e^{-|x|}$ .
- *Vào:*
  - Cấu trúc mạng m, n, a.

- Số lượng chuỗi nhị phân Popsiz.
- Ra:
  - Quần thể gồm Popsiz chuỗi nhị phân được lưu trữ trong bảng OldPop.
- Giải thuật:
  - Tính tổng số trọng số M trong mạng, số trọng số trong mạng bằng:  
$$M = (m + n) * a + n + a$$
  - Lặp i = 1 đến Popsiz
    - Lặp j = 1 đến M
      - + Sinh số ngẫu nhiên  $p_0$  trong khoảng  $[0,1]$ .
      - + Tính giá trị  $x = \ln(1 - p_0)$
      - + Sinh số ngẫu nhiên  $p_1$
      - + Nếu  $p_1 < 0.5$  thì  $x = -x$
      - + Mã hóa giá trị x thành chuỗi nhị phân con 20 bit trong khoảng  $[-10,10]$ .
      - + Nối kết M chuỗi nhị phân con thành một chuỗi lớn, chính là chuỗi cá thể.

**Phân hệ 1.1**

- Chức năng:

- Đối với mỗi chuỗi cá thể trong quần thể OldPop giải mã thành tập trọng số, sau đó lan truyền toàn bộ tập luyện qua mạng, tích lũy sai số theo hàm giá 3.3 ở chương 3.

- Chuyển đổi giá trị hàm giá thành giá trị sức khỏe.

- Vào:

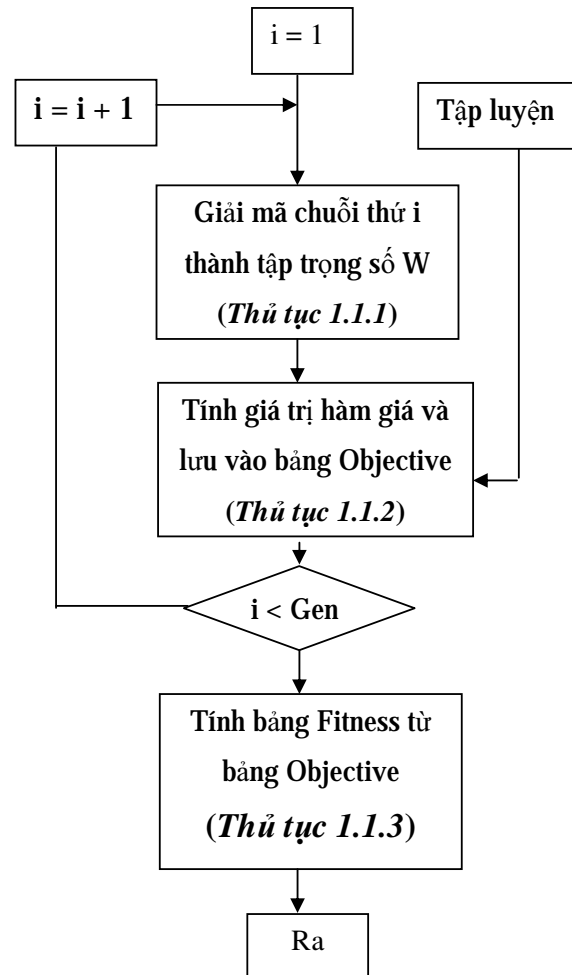
- Quần thể OldPop.
- Tập luyện.

- Ra:

- Giá trị sức khỏe toàn quần thể được chứa trong bảng Fitness( ).

- Giải thuật

- Lặp  $i = 1$  đến PopSize
  - + Giải mã chuỗi thứ  $i$  trong quần thể oldPop thành tập trọng số  $W$  (**Thủ tục 1.1.1**).
  - + Tính giá trị hàm giá cho mạng nơ ron có tập trọng số vừa được giải mã và lưu giá trị đó vào bảng objective( ) (**Thủ tục 1.1.2**).



Hình 4.2: Sơ đồ khối giải thuật **Phân hệ 1.1**

- Tính bảng sức khỏe Fitness( ) từ bảng giá trị hàm giá objective( )  
(**Thủ tục 1.1.3**).

**Thủ tục 1.1.1**

- Chức năng:
  - Giải mã chuỗi nhị phân thành bảng tuyến tính các trọng số W
- Vào:
  - Chuỗi nhị phân độ dài Lchrom
  - Tổng số trọng số M
- Ra:
  - Bảng W( ) của các trọng số (số thực).
- Giải thuật:
  - Lặp i =1 đến M
    - + Cắt liên tiếp một chuỗi con độ dài 20 bit từ chuỗi cá thể.
    - + Tính giá trị x của chuỗi nhị phân (x là số nguyên dài)
    - + Giá trị  $W(i) = (20.x / (2^{20} - 1)) - 10$ .

**Thủ tục 1.1.2**

- Chức năng:
  - Tính sai số cho một cấu trúc mạng m, n, a và bộ trọng số W với một tập luyện cho trước.
- Vào:
  - Cấu trúc mạng m, n, a và bộ trọng số.
  - Tập số liệu huấn luyện gồm P mẫu (hai véc tơ vào và ra X, y).
- Ra:
  - Sai số e sinh ra sau khi lan truyền toàn bộ các mẫu qua mạng.
- Giải thuật
  - Gán  $e = 0$
  - Lặp i = 1 đến P
    - + Gán các tín hiệu ra của các bias = 1.
    - + Gán tín hiệu ra ở lớp vào  $out^0$  bằng tín hiệu vào X.



+ Lặp đối với mọi nơ ron thứ  $j$  ở trên lớp ẩn và lớp ra

Tính tổng tín hiệu vào theo công thức  $Net_j^l = \sum_{i=1}^m w_{ji}^l \cdot x_i^l$

Tính tín hiệu ra  $Out_j^l = \frac{1}{1 + \exp(-Net_j^l)}$

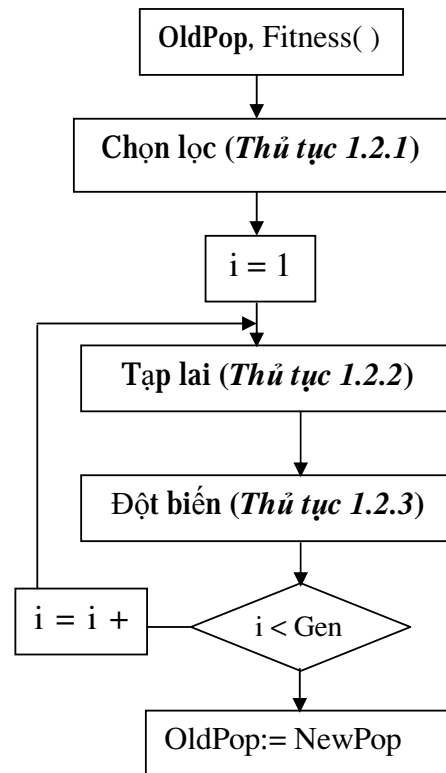
+ Tích lũy sai số vào  $e$ :  $E = E + \frac{1}{2} \sum_{j=1}^n (y_j^i - Out_j^{last})^2$

### **Thủ tục 1.1.3**

- Chức năng:
  - Tính bảng giá trị sức khỏe Fitness( ) của quần thể oldPop từ bảng giá trị hàm giá objective( ).
- Vào:
  - Bảng giá trị hàm giá objective( ).
  - Số cá thể trong quần thể PopSize.
- Ra:
  - Bảng giá trị hàm sức khỏe Fitness( )
- Giải thuật:
  - Tính giá trị Max của bảng giá trị hàm giá objective( ).
  - Lặp  $j = 1$  đến Popsize:  $Fitness[i] = Max - objective(i)$
  - Tính giá trị Max, giá trị trung bình ave của bảng Fitness.
  - Nếu  $Max > 2*ave$  thì  $a = ave / (Max - ave)$ ,  $b = (Max - 2*ave)*a$   
Không thì  $a = 1$ ,  $b = 0$ .
  - Lặp  $j = 1$  đến PopSize  $Fitness[j] = Fitness[j]*a + b$ .

**Phân hệ 1.2**

- Chức năng:
  - Sản sinh quần thể mới NewPop từ quần thể cũ OldPop
  - Thế quần thể cũ bằng quần thể mới.
- Vào:
  - Quần thể cũ OldPop.
  - Bảng giá trị sức khỏe của quần thể cũ.
- Ra:
  - Quần thể OldPop đã được thế bởi thế hệ mới.



Hình 4.3: Sơ đồ giải thuật Phân hệ 1.2

- Giải thuật:
  - Toán tử chọn lọc (**Thủ tục 1.2.1**)
  - Lặp  $i = 1$  đến khi  $i$  lớn hơn hoặc bằng PopSize, bước nhảy 2
    - + Toán tử tạo lại (**Thủ tục 1.2.2**)
    - + Toán tử đột biến (**Thủ tục 1.2.3**)
  - Thế quần thể cũ OldPop bằng quần thể mới NewPop.

**Thủ tục 1.2.1**

- Chức năng:
  - Chọn lọc quần thể bố mẹ từ quần thể con, mỗi cá thể được chọn với xác suất tỷ lệ với sức khỏe của cá thể đó.
- Vào:
  - Quần thể cũ OldPop và bảng giá trị sức khỏe của từng cá thể trong quần thể.

- Ra:
  - Quần thể mới NewPop các cá thể bố mẹ được chọn
- Giải thuật:
  - Tính tổng sức khỏe toàn quần thể Sumfitness.
  - Lặp  $i = 1$  đến khi  $i$  lớn hơn hoặc bằng PopSize
    - + Sinh một số ngẫu nhiên  $p_0$ .
    - + Tính giá trị  $S_u = p_0 * \text{Sumfitness}$ .
    - + Chỉ số  $j$  để tổng chạy sức khỏe của cá thể lớn hơn  $S_u$  là chỉ số của cá thể được chọn.
    - + Đưa cá thể được chọn vào quần thể mới NewPop.

#### **Thủ tục 1.2.2**

- Chức năng:
  - TẠP lai hai chuỗi bố mẹ để tạo thành hai con mới
- Vào:
  - Chỉ số của hai chuỗi bố mẹ trong quần thể cũ
  - Xác suất TẠP lai Pcross.
- Ra:
  - Hai chuỗi con mới.
- Giải thuật
  - Sinh một số ngẫu nhiên  $p_0$
  - Nếu  $p_0 < \text{Pcross}$  thì
    - + Sinh một số ngẫu nhiên mới  $p_1$
    - + Tính vị trí TẠP lai  $l = p_1 * (\text{Lchrom} - 1)$
  - Không thì Vị trí TẠP lai là Lchrom.
  - Sao chép gen từ 1 đến  $l$  của bố mẹ 1 sang con 1 và bố mẹ 2 sang con 2
  - Sao chép gen từ  $l+1$  đến Lchrom của bố mẹ 1 sang con 2 và từ bố mẹ 2 sang con 1.

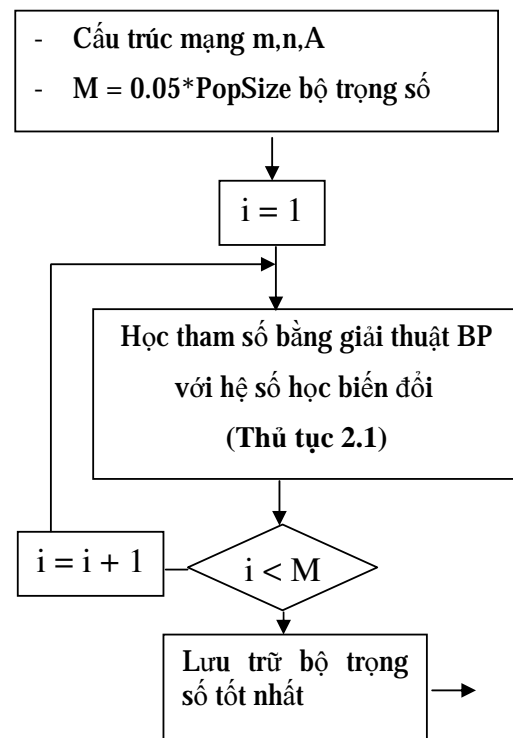
**Thủ tục 1.2.3**

- Chức năng:
  - Làm đột biến gen của hai chuỗi con mới được sinh ra.
- Vào:
  - Hai chuỗi con sinh ra sau tập lai.
  - Xác suất đột biến Pmutation.
- Ra:
  - Hai chuỗi con sau đột biến.
- Giải thuật:
  - Duyệt từ gen của hai chuỗi con mới được sinh ra sau tập lai.
  - Sinh số ngẫu nhiên  $p_0$ .
  - Nếu  $p_0 < Pmutation$  thì Gen đó được biến đổi từ 0 sang 1 hoặc ngược lại.
  - Không thì Gen đó được giữ nguyên.

**Phân hệ 2**

- Chức năng:
  - Luyện tham số bằng giải thuật BP với hệ số học biến đổi đối với bộ trọng số chuyển từ kết quả luyện của giải thuật GA chuyển sang.
  - Lưu trữ bộ trọng số tốt nhất.
- Vào:  $0.05 * PopSize$  bộ trọng số cùng một cấu trúc mạng  $m, n, a$ .
- Ra: Một bộ trọng số  $W$ .
- Giải thuật:
 

Lặp  $i = 1$  đến  $0.05 * PopSize$

Hình 4.4: Sơ đồ khối giải thuật **phân hệ 2**

- + Học tham số với giải thuật BP với hệ số học biến đổi (***Thủ tục 2.1***).
- + Lưu trữ bộ trọng số cho giá trị sai số tích lũy  $e$  là nhỏ nhất.

**Thủ tục 2.1**

- Chức năng: Học tham số bằng giải thuật BP với hệ số học biến đổi
- Vào: Cấu trúc mạng  $m, n, a, W$  và tập mẫu luyện, số bước thực hiện biến đổi *Step*, hệ số học  $\alpha$ , bước tăng giảm của hệ số học  $a$  và sai số tối thiểu làm tiêu chuẩn dừng  $\varepsilon$ .
- Ra: Bộ trọng số  $W$  sau khi học.
- Giải thuật:

Lập các bước sau đây cho đến khi sai số MSE nhỏ hơn tiêu chuẩn dừng  $\varepsilon$ .

- Khởi tạo tổng sai số trên tập huấn luyện  $e = 0$ , bước thực hiện biến đổi  $k = 0$
- Lập  $i = 1$  đến số mẫu có trong tập luyện
  - + Gán tín hiệu ra ở lớp vào  $out^0 = X_i$
  - + Lập đối với các nơ ron thứ  $j$  ở trên lớp ẩn ( $l = 1$ ) và lớp ra ( $l = 2$ )

$$\text{Tín tổng tín hiệu vào theo công thức } Net_j^l = \sum_{i=1}^m w_{ji}^l \cdot x_i^l$$

$$\text{Tín giá trị tín hiệu ra } Out_j^l = \frac{1}{1 + \exp(-Net_j^l)}$$

$$+ \text{ Tính sai số ở lớp ra } \varepsilon^{\text{last}} = \sum_{j=1}^n (y_j - Out_j^{\text{last}})^2$$

+ Bắt đầu từ lớp ra ( $l = 2$ ) cho tới lớp ẩn ( $l = 1$ ) tính:

$$\text{Hệ số hiệu chỉnh } \delta_j^l$$

$$\text{Lượng hiệu chỉnh } \Delta w_{ji}^l = \eta \cdot \delta_j^l \cdot Out_i^{l-1}$$

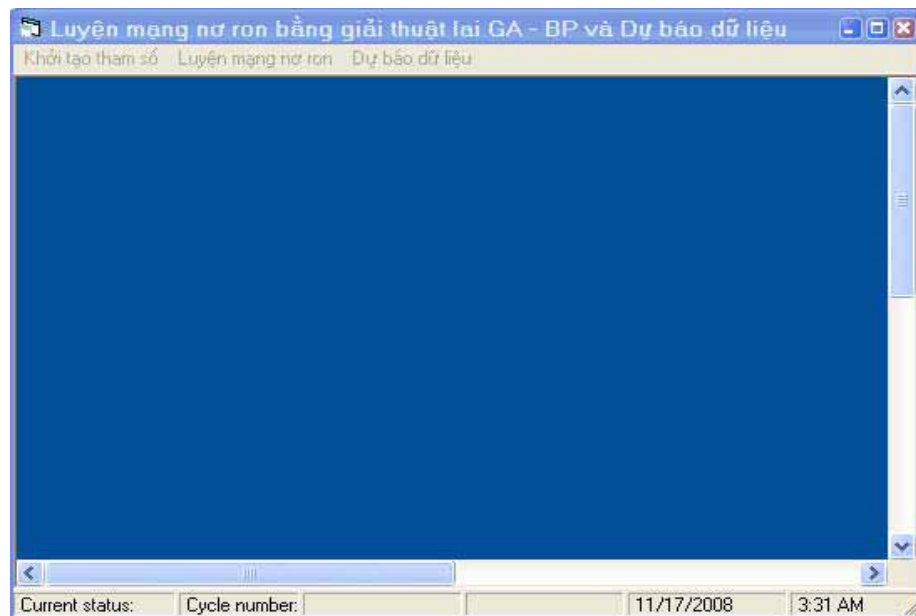
$$\text{Hiệu chỉnh các trọng số } w_{ji}^l = w_{ji}^l + \Delta w_{ji}^l$$

- Tính giá trị hàm giá  $e$  theo ***Thủ tục 1.1.2***
- Thực hiện quá trình biến đổi hệ số học:

- + Nếu  $\Delta e < 0$ , kiểm tra nếu  $k < \text{Step}$  thì  $k = k + 1$ , không thì gán  $k = 0$  và  $\alpha = \alpha + a$
- + Nếu  $\Delta e \geq 0$  thì  $\alpha = \alpha * (1 - a)$  và gán  $k = 0$ .

#### 4.3. CHƯƠNG TRÌNH DỰ BÁO DỮ LIỆU

Màn hình chính của chương trình như sau :



Hình 4.5. Màn hình chính của chương trình dự báo

Chương trình xây dựng gồm các mục thực đơn : **Khởi tạo tham số**, **Luyện mạng nơ ron**, **Dự báo dữ liệu**. Sau đây là mô tả chi tiết các chức năng của chương trình:

- **Mở tệp huấn luyện**

Tệp dữ liệu huấn luyện là tệp có cấu trúc được lưu trữ trong một tệp TXT, chứa 43 mẫu số liệu từ năm 2001 đến năm 2005 về mực nước đỉnh lũ, mực nước chân lũ và lượng mưa trung bình đo được tại trạm Sơn Giang. Số liệu đưa vào mạng được mã hóa trong đoạn  $[0.1, 0.9]$  theo nguyên tắc nêu phần 4.2.1.

- Các trường dữ liệu được phân cách nhau bằng dấu “;”
- Trường dữ liệu dự báo là trường cuối cùng, là đầu ra của mạng.

Ví dụ : tệp dữ liệu sau khi được mã hóa như sau :



Hình 4.6: Dữ liệu tệp huấn luyện

- **Màn hình nhập các tham số cấu trúc mạng**

Cho phép người sử dụng nhập các tham số đầu vào cho mạng nơ-ron. Số lớp mạng ngầm định là 2, số đầu vào là 2 và số đầu ra là 1 lấy theo tệp huấn luyện.



Hình 4.7: Màn hình nhập tham số cho mạng nơ-ron

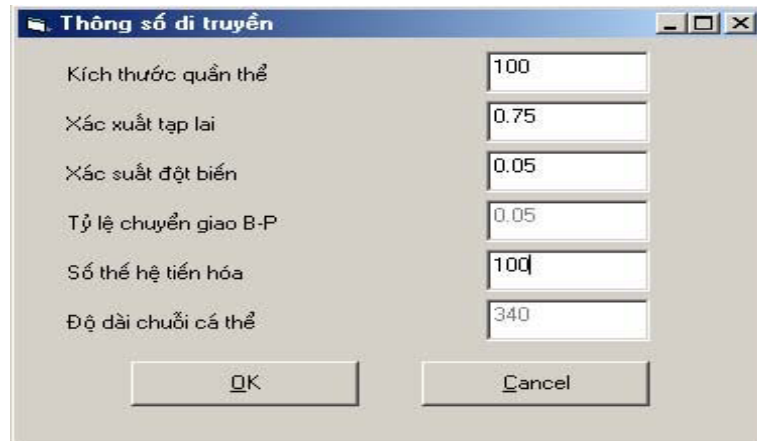
Với bài toán này, số nơ-ron trên lớp ẩn chọn là 4, giá trị các tham số khác ngầm định trên màn hình nhập được coi là các giá trị khởi đầu khá tốt. Sau khi nhập xong, nhấn OK để gán giá trị các tham số cho mạng nơ-ron.

- **Màn hình nhập các tham số của giải thuật di truyền**

Cho phép người sử dụng nhập các tham số của giải thuật di truyền như kích thước quần thể, xác suất tạp lai, xác suất đột biến, số thế hệ tiến hóa... Các giá trị ngầm định ở màn hình dưới được xem là các giá trị xuất phát khá tốt tìm được theo phương pháp thử và sai, kích thước quần thể chọn là 100, số thế hệ tiến hóa là 100.

Tỷ lệ chuyển giao số cá thể sang luyện tiếp bằng giải thuật BP ngầm định là 0.05. Số trọng số của mạng tương ứng với bài toán thử nghiệm khi chọn 4 nơ-ron

trong lớp ẩn là  $4*2 + 4 + 4*1 + 1 = 17$  trọng số, do vậy độ dài của chuỗi cá thể là  $17*20 = 340$ .

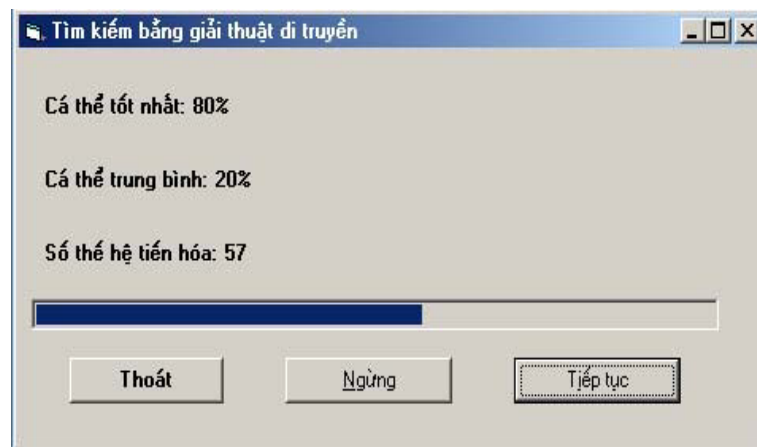


Hình 4.8: Màn hình nhập tham số cho giải thuật GA

Bước tiếp theo là thực thi giải thuật lai GA - BP

- **Tìm kiếm bằng giải thuật di truyền**

Màn hình tìm kiếm các cá thể tốt bằng giải thuật di truyền có dạng sau



Hình 4.9: Tìm kiếm bằng giải thuật GA

Tại mỗi thế hệ tiến hóa, màn hình thông báo số cá thể tốt có sức khỏe lớn hơn sức khỏe trung bình toàn quần thể và số cá thể trung bình có sức khỏe nhỏ hơn sức khỏe trung bình. Nhận thấy rằng ở giai đoạn cuối của số thế hệ tiến hóa, số cá thể tốt chiếm đa số, giá trị sức khỏe của chúng gần với giá trị sức khỏe trung bình.

Sau 100 thế hệ tiến hóa, 5 cá thể có sức khỏe tốt nhất trong số 100 cá thể ở quần thể cuối cùng được lưu trữ lại làm đầu vào cho giải thuật BP.

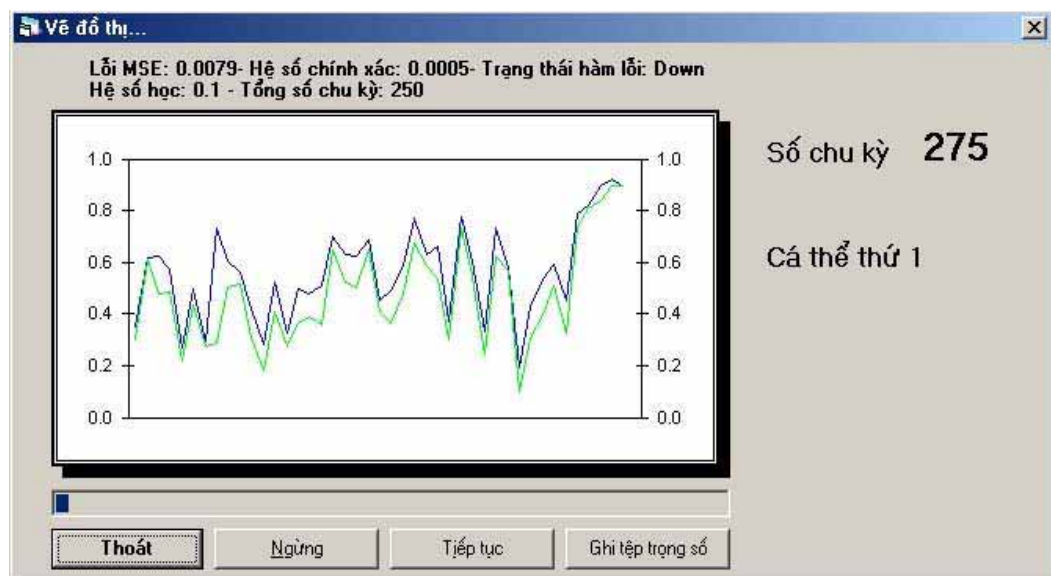


- **Huấn luyện bằng giải thuật BP**

5 cá thể lần lượt được giải thuật BP sử dụng hàng số học biến đổi luyện đến bão hòa với các tham số ban đầu đã được khởi tạo.

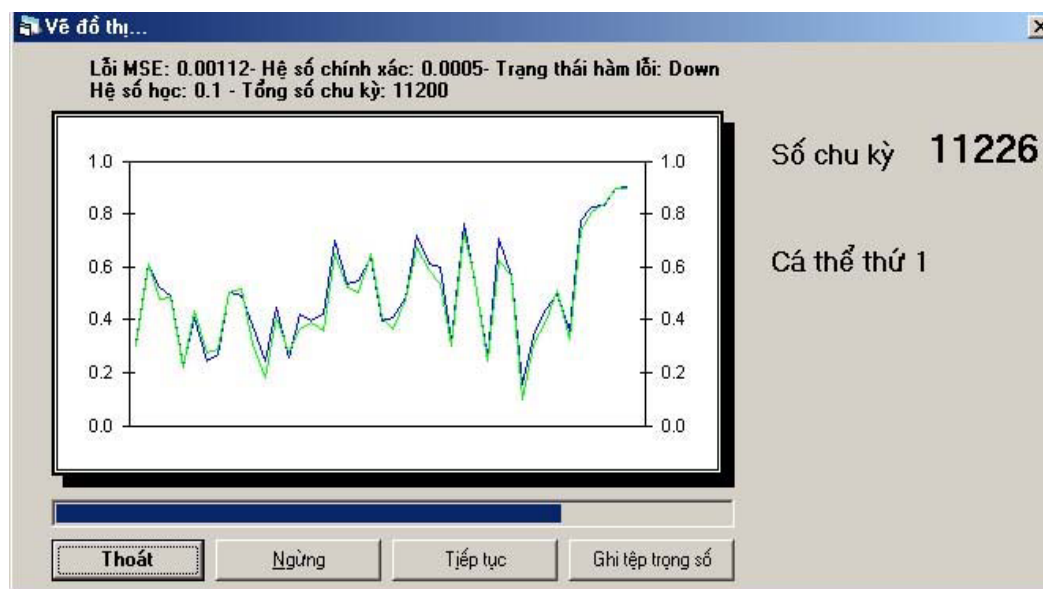
Các đồ thị dưới đây mô tả một chu kỳ luyện đối với một cá thể.

Trên đồ thị, đường màu xanh nhạt là các đầu ra mong muốn đối với tập dữ liệu, đường màu xanh đậm là trả lời của mạng đối với dữ liệu đầu vào. Đối với mỗi cá thể, tại điểm xuất phát luyện bằng giải thuật BP, hai đường này đã khá gần nhau, do vậy giải thuật di truyền tìm kiếm các cá thể đã khá gần lời giải.



Hình 4.10.a: Huấn luyện bằng giải thuật BP

Tập dữ liệu huấn luyện đồng thời cũng dùng làm tập kiểm tra để kiểm tra khả năng tổng quát hóa của mạng. Việc kiểm tra này được thực hiện với việc cập nhật đồ thị đều đặn sau 50 chu kỳ huấn luyện. Sau một số lớn chu kỳ huấn luyện, khả năng tổng quát hóa của mạng đã khá tốt so với ban đầu. Trên hình vẽ, hai đường gần như trùng nhau. Đồng thời, lỗi MSE tiếp tục giảm cho đến khi nhỏ hơn hệ số chính xác, tập trọng số được ghi lại và thuật toán lại tiếp tục với cá thể tiếp theo

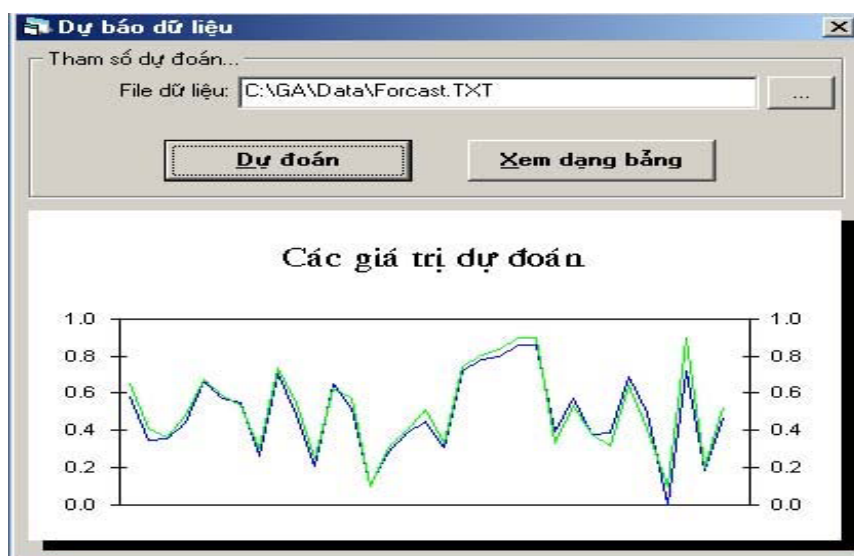


Hình 4.10.b:Huấn luyện bằng giải thuật BP

Kết thúc chu kỳ huấn luyện 5 cá thể, cá thể có tập trọng số tốt nhất (có sai số MSE nhỏ nhất) được chọn làm kết quả của giải thuật. Tập trọng số này được ghi lại dưới dạng một tệp TXT.

- **Dự báo dữ liệu**

Mạng sau khi được huấn luyện sử dụng để dự báo dữ liệu. Tệp dữ liệu dự báo là tệp TXT chứa số liệu về mối quan hệ giữa mực nước đỉnh lũ với mực nước chân lũ và lượng mưa đo được tại trạm Sơn Giang. Tệp này có cấu trúc và được tỷ lệ hóa giống như tệp huấn luyện Màn hình dự báo như sau:



Hình 4.11: Màn hình dự báo

Trên màn hình, đường biểu diễn đầu ra mong muốn và trả lời của mạng sát nhau, chứng tỏ khả năng tổng quát hóa của mạng sau khi được học là khá tốt.

#### ❖ Kết luận chương 4

Chương 4 giới thiệu bài toán dự báo lũ trên sông Trà Khúc và thực hiện các bước xây dựng chương trình dự báo dựa trên cơ sở giải thuật lai GA-BP đã trình bày trong chương 3. Kết quả của chương trình đã cho thấy, sau khi được huấn luyện bằng giải thuật lai GA-BP, mạng cho kết quả dự báo khá tốt.

## KẾT LUẬN

Luận văn tập trung nghiên cứu kỹ thuật sử dụng mạng nơon và giải thuật di truyền trong khai phá dữ liệu. Kết hợp tính chất tìm kiếm toàn cục của giải thuật GA với tính hội tụ của giải thuật BP, luận văn nghiên cứu giải pháp xây dựng giải thuật lai GA-BP trong huấn luyện mạng nơon truyền thẳng nhiều lớp và áp dụng thử nghiệm mô hình đó cho bài toán dự báo trong lĩnh vực khí tượng thủy văn.

Một số kết quả đạt được của luận văn:

- Tổng kết những vấn đề nghiên cứu về khai phá dữ liệu và phát hiện tri thức trong CSDL.
- Tìm hiểu về kỹ thuật sử dụng mạng nơon, giải thuật di truyền trong khai phá dữ liệu và các vấn đề liên quan. Nghiên cứu giải pháp tích hợp giải thuật GA và giải thuật BP thành một giải thuật lai dùng để huấn luyện mạng nơon truyền thẳng nhiều lớp.
- Áp dụng những vấn đề đã nghiên cứu vào xây dựng mô hình và cài đặt mạng nơon dự báo cho bài toán dự báo lũ trên sông.

***Một số hướng phát triển:***

- Tích hợp giải thuật GA và PB trong việc học cấu trúc của mạng nơon nhằm tìm ra số nơon trong lớp ẩn tốt nhất cho một bài toán.
- Cải tiến các toán tử của giải thuật GA để nâng cao hiệu quả tìm kiếm các cá thể tốt nhất.

## TÀI LIỆU THAM KHẢO

### *Tài liệu tiếng Việt*

- [1]. Nguyễn Đình Thúc (2001), *Lập trình tiến hóa*, Nhà xuất bản giáo dục.

### *Tài liệu tiếng Anh*

- [2]. Back T. and Schwefel H.-P. (1993), “An overview of evolutionary algorithms for parameter optimization”, *evolutionary Computation*, vol. 1, no. 1, pp. 1-23.
- [3]. Bose N. and Liang P. (1996), *Neural Network Fundamentals with Graphs, algorithms, and applications*, McGraw-Hill.
- [4]. Fayyad, Gregory Piatetsky, Shapiro, Padhraic Smith, (1996), *From Data mining to Knowledge Discovery: An overview*.
- [5]. Gero J. S., Kazakov V. a., and Schinier T., (1997), “Genetic engineering and design problems”, *In Evolutionary Algorithms in Engineering Applications*, pages 47-68. Springer-Verlag.
- [6]. Goldberg D. E., (1989), *Genetic algorithm in search, optimization and machine learning*, Addison-Wesley, Reading, Massachusets.
- [7]. Ho Tu Bao, *Introduction to Knowledge Discovery and Data Mining*, Institute of Information Technology, <http://www.ebook.edu.vn/?page=1.39&view=1694>.
- [8]. Lawrence S., C. L. Giles, a. C. Tsoj, “What size Neural Network Gives optimal Generalization? Convergence Properties of Backpropagation”, *Technical Report*, Institute for Advanced Computer Studies - University of Maryland College Park, June 1996.
- [9]. Oh S. H., Lee yj., a modified error function to improve the error Back-Propagation algorithm for Multi-layer perceptrons, *eTRi Journal Vol 17, No 1*, april 1995.
- [10]. Patterson D. (1996), *Artificial Neural Networks, Theory and Application*, Prentice Hall.

- [11]. Randall S. Sexton and Naheel A. Sikander, “Data Mining using a Genetic algorithm trained Neural network”, Computer introduction system, Southwest Missouri State University, USA.
- [12]. **Schalkoff R. (1997), *Artificial neural networks*, McGraw-Hill.**
- [13]. Udoseiffert, Michaelis B., On the gradient desert in back-propagation and its substitution by a genetic algorithm, *Proceedings of the IASTED international Conference Applied Informatics 14-17/02/2000*, Innsbruck, Austria.