



TRƯỜNG CAO ĐẲNG CNTT HỮU NGHỊ VIỆT - HÀN

KHOA KHOA HỌC MÁY TÍNH

-----***-----

TRÍ TUỆ NHÂN TẠO

(Artificial Intelligence - AI)

Nguyễn Thanh Cẩm



Mục đích của môn học



Contents



- 1 **Tổng quan về khoa học trí tuệ nhân tạo**
- 2 **Các phương pháp giải quyết vấn đề cơ bản**
- 3 **Tri thức và các phương pháp biểu diễn tri thức**
- 4 **Máy học**
- 5 **Mạng Nơron**

Mạng Nơron

- 5.1 Tổng quan về mạng Nơron nhân tạo.....●
- 5.2 Các thành phần cơ bản của mạng Nơron nhân tạo.....●
- 5.3 Mạng truyền thẳng và thuật toán lan truyền ngược.....●
- 5.4 Một số ứng dụng của mạng Nơron.....●

5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO



5.1.1

Nguồn gốc sinh học

5.1.2


Nơon nhân tạo

5.1.3

Lịch sử phát triển

5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO

5.1.1. Nguồn gốc sinh học

- 
- ◆ Bộ não người chứa khoảng 10^{11} các phần tử liên kết chặt chẽ với nhau gọi là các Nơon.
 - ◆ Người làm tin học, Nơon được cấu tạo bởi các thành phần:
 - Tế bào hình cây (dendrite)
 - Tế bào thân (cell body)
 - Và sợi trục thần kinh (axon).


5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO

5.1.1. Nguồn gốc sinh học

- ◆ Cấu trúc của mạng nơon luôn luôn phát triển và thay đổi.
- ◆ Các thay đổi có khuynh hướng chủ yếu là việc làm tăng hay giảm độ mạnh của các mối liên kết thông qua các khớp thần kinh.

5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO

5.1.1. Nguồn gốc sinh học

- 
- ◆ Tính toán của não khác với tính toán theo thuật toán:
 - Quá trình tính toán được tiến hành song song và phân tán trên nhiều nơon gần như đồng thời.
 - Tính toán thực chất là quá trình học, chứ không phải theo sơ đồ định sẵn từ trước.

5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO



5.1.1

Nguồn gốc sinh học

5.1.2


Nơon nhân tạo

5.1.3

Lịch sử phát triển

5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO

5.1.2. Nơon nhân tạo

- 
- ◆ bắt chước não, các nhà khoa học đã có mô hình tính toán: các mạng nơon nhân tạo (Artificial Neural Networks – ANN).
 - ◆ Mạng nơon nhân tạo không phức tạp của bộ não.
 - **Cấu trúc khối là các thiết bị tính toán đơn giản được liên kết chặt chẽ với nhau.**
 - **Các liên kết giữa các nơon quyết định chức năng của mạng.**


5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO

5.1.2. Nơon nhân tạo

- ◆ Mạng nơon sinh học hoạt động rất chậm so với các linh kiện điện tử (10^{-3} giây so với 10^{-9} giây),
- ◆ Bộ não có khả năng thực hiện nhiều công việc nhanh hơn nhiều so với các máy tính thông thường.
- ◆ Do cấu trúc song song của mạng nơon sinh học: toàn bộ các nơon hoạt động đồng thời tại một thời điểm.
- ◆ hiện nay, các mạng nơon chủ yếu được thực nghiệm trên các máy tính số, nhưng cấu trúc song song của chúng phù hợp nhất là thực nghiệm chúng trên các vi mạch tích hợp lớn (VLSI: Very Large Scale Integrated-circuit), các thiết bị quang và các bộ xử lý song song.

5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO

5.1.2. Nơon nhân tạo

- 
- ◆ Mạng nơon có các đặc trưng phân biệt sau:
 - Tập các đơn vị xử lý (các nơon nhân tạo);
 - Trạng thái kích hoạt hay là đầu ra của đơn vị xử lý;
 - Liên kết giữa các đơn vị. Xét tổng quát, mỗi liên kết được định nghĩa bởi một trọng số w_{jk} cho ta biết hiệu ứng mà tín hiệu của đơn vị j có trên đơn vị k ;
 - Một luật lan truyền quyết định cách tính tín hiệu ra của từng đơn vị từ đầu vào của nó;
 - Một hàm kích hoạt, hay hàm chuyển (activation function, transfer function), xác định mức độ kích hoạt khác dựa trên mức độ kích hoạt hiện tại;
 - Một đơn vị điều chỉnh (độ lệch) (bias, offset) của mỗi đơn vị;
 - Phương pháp thu thập thông tin (luật học - learning rule);
 - Môi trường hệ thống có thể hoạt động.

5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO



5.1.1

Nguồn gốc sinh học

5.1.2


Nơon nhân tạo

5.1.3

Lịch sử phát triển

5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO

5.1.3. Lịch sử phát triển

- 
- ◆ Các mốc đáng chú ý trong lịch sử phát triển của mạng nơon.
 - Cuối TK 19, đầu TK 20, sự phát triển chủ yếu chỉ là những ngành Vật lý học, Tâm lý học và Thần kinh học, bởi các nhà khoa học như Hermann von Hemholtz, Ernst Mach, Ivan Pavlov.
 - Các công trình nghiên cứu của họ chủ yếu đi sâu vào các lý thuyết tổng quát về HỌC (Learning), NHÌN (vision) và LẬP LUẬN (conditioning),... và không hề đưa ra những mô hình toán học cụ thể mô tả hoạt động của các nơon.

5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO

5.1.3. Lịch sử phát triển

- vào 1940s với công trình của Warren McCulloch và Walter Pitts. Họ chỉ ra rằng về nguyên tắc, mạng của các nơon nhân tạo có thể tính toán bất kỳ một hàm số học hay logic nào
- Donald Hebb, đã phát biểu rằng việc thuyết lập luận cổ điển (classical conditioning) (như Pavlov đưa ra) là hiện thực bởi do các thuộc tính của từng nơon riêng biệt. Ông cũng nêu ra một phương pháp học của các nơon nhân tạo.

5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO

5.1.3. Lịch sử phát triển

- Ứng dụng thực nghiệm đầu tiên của các nơon nhân tạo vào cuối những năm 50 cùng với phát minh của mạng nhận thức (perceptron network) và luật học tương ứng bởi Frank Rosenblatt.
- Bernard Widrow và Ted Hoff đã đưa ra một thuật toán học mới và sử dụng nó để huấn luyện cho các mạng nơon tuyến tính thích nghi, mạng có cấu trúc và chức năng tương tự như mạng của Rosenblatt. Luật học Widrow-Hoff vẫn còn được sử dụng cho đến nay.

5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO

5.1.3. Lịch sử phát triển

- Tuy nhiên cả Rosenblatt và Widrow-Hoff đều cùng vấp phải một vấn đề do Marvin Minsky và Seymour Papert phát hiện ra, đó là các mạng nhận thức chỉ có khả năng giải quyết các bài toán khả phân tuyến tính. Họ cố gắng cải tiến luật học và mạng để có thể vượt qua được hạn chế này nhưng họ đã không thành công trong việc cải tiến luật học để có thể huấn luyện được các mạng có cấu trúc phức tạp hơn.
- Do những kết quả của Minsky-Papert nên việc nghiên cứu về mạng nơon gần như bị đình lại trong suốt một thập kỷ do nguyên nhân là không có được các máy tính đủ mạnh để có thể thực nghiệm.

5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO

5.1.3. Lịch sử phát triển

- Năm 1972, Teuvo Kohonen và James Anderson độc lập nhau phát triển một loại mạng mới có thể hoạt động như một bộ nhớ.
- Stephen Grossberg cũng rất tích cực trong việc khảo sát các mạng tự tổ chức (Self organizing networks).

5.1 TỔNG QUAN VỀ MẠNG NƠN NHÂN TẠO

5.1.3. Lịch sử phát triển

- Vào những năm 80, việc nghiên cứu mạng nơon phát triển rất mạnh mẽ cùng với sự ra đời của PC. Có hai khái niệm mới liên quan đến sự hồi sinh này, đó là:
 - 1. Việc sử dụng các phương pháp thống kê để giải thích hoạt động của một lớp các mạng hồi quy (recurrent networks) có thể được dùng như bộ nhớ liên hợp (associative memory) trong công trình của nhà vật lý học John Hopfield.
 - 2. Sự ra đời của thuật toán lan truyền ngược (back-propagation) để luyện các mạng nhiều lớp được một vài nhà nghiên cứu độc lập tìm ra như: David Rumelhart, James McClelland,.... Đó cũng là câu trả lời cho Minsky-Papert.

5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo

A decorative graphic on the left side of the slide showing a globe with a glowing effect, surrounded by a yellow and white light aura.

5.2.1 Đơn vị xử lý

5.2.2 Hàm kết hợp

5.2.3 Hàm kích hoạt (hàm chuyển)

5.2.4 Các hình trạng của mạng

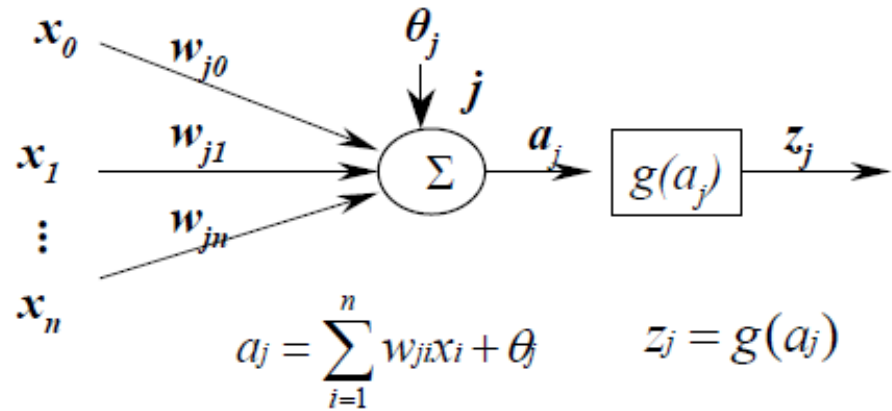
5.2.5 Mạng học (huấn luyện mạng)

5.2.6 Hàm mục tiêu

5.2 Các thành phần cơ bản của mạng nơon nhân tạo

5.2.1. Đơn vị xử lý

◆ Đơn vị xử lý (Processing unit)



x_i : các đầu vào

w_{ji} : các trọng số tương ứng với các đầu vào

θ_j : độ lệch (bias)


a_j : đầu vào mạng (net-input)

z_j : đầu ra của nơon

$g(x)$: hàm chuyển (hàm kích hoạt).

5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo

5.2.1. Đơn vị xử lý

- 
- ◆ Trong một mạng nơ-ron có ba kiểu đơn vị:
 - Các đơn vị đầu vào (Input units), nhận tín hiệu từ bên ngoài;
 - Các đơn vị đầu ra (Output units), gửi dữ liệu ra bên ngoài;
 - Các đơn vị ẩn (Hidden units), tín hiệu vào (input) và ra (output) của nó nằm trong mạng.
 - ◆ Mỗi đơn vị j có thể có một hoặc nhiều đầu vào: $x_0, x_1, x_2, \dots, x_n$, nhưng chỉ có một đầu ra z_j . Một đầu vào tới một đơn vị có thể là dữ liệu từ bên ngoài mạng, hoặc đầu ra của một đơn vị khác, hoặc là đầu ra của chính nó.

5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo

A decorative graphic on the left side of the slide showing a globe with a glowing effect, surrounded by a yellow and white light aura.

5.2.1 Đơn vị xử lý

5.2.2 Hàm kết hợp

5.2.3 Hàm kích hoạt (hàm chuyển)

5.2.4 Các hình trạng của mạng

5.2.5 Mạng học (huấn luyện mạng)

5.2.6 Hàm mục tiêu

5.2 Các thành phần cơ bản của mạng nơon nhân tạo

5.2.2. Hàm kết hợp

- ◆ Mỗi một đơn vị trong một mạng kết hợp các giá trị đưa vào nó thông qua các liên kết với các đơn vị khác, sinh ra một giá trị gọi là net input. Hàm thực hiện nhiệm vụ này gọi là hàm kết hợp (combination function), được định nghĩa bởi một luật lan truyền cụ thể. Trong phần lớn các mạng nơon, chúng ta giả sử rằng mỗi một đơn vị cung cấp một bộ cộng như là đầu vào cho đơn vị mà nó có liên kết. Tổng đầu vào đơn vị j đơn giản chỉ là tổng trọng số của các đầu ra riêng lẻ từ các đơn vị kết nối cộng thêm ngưỡng hay độ lệch (bias) θ_j :

$$a_j = \sum_{i=1}^n w_{ji}x_i + \theta_j$$

5.2.2. Hàm kết hợp

- ◆ Trường hợp $w_{ji} > 0$, nơron được coi là đang ở trong trạng thái kích thích. Tương tự, nếu như $w_{ji} < 0$, nơron ở trạng thái kiềm chế. Chúng ta gọi các đơn vị với luật lan truyền như trên là các sigma units.
- ◆ Trong một vài trường hợp người ta cũng có thể sử dụng các luật lan truyền phức tạp hơn. Một trong số đó là luật sigma-pi, có dạng như sau:
- ◆ Rất nhiều hàm kết hợp sử dụng một "độ ngưỡng" để tính net input tới đơn vị. Đối với một đơn vị đầu ra tuyến tính, thông thường, θ_j được chọn là hằng số và trong bài toán xấp xỉ đa thức $\theta_j = 1$

$$a_j = \sum_{i=1}^n w_{ji} \prod_{k=1}^m x_{ik} + \theta_j$$

5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo



5.2.1 Đơn vị xử lý

5.2.2 Hàm kết hợp

5.2.3 Hàm kích hoạt (hàm chuyển)

5.2.4 Các hình trạng của mạng

5.2.5 Mạng học (huấn luyện mạng)

5.2.6 Hàm mục tiêu

5.2.2. Hàm kích hoạt (hàm chuyển)

- ◆ Phần lớn các đơn vị trong mạng nơon chuyển net input bằng cách sử dụng một hàm vô hướng (scalar-to-scalar function) gọi là hàm kích hoạt, kết quả của hàm này là một giá trị gọi là mức độ kích hoạt của đơn vị (unit's activation). Loại trừ khả năng đơn vị đó thuộc lớp ra, giá trị kích hoạt được đưa vào một hay nhiều đơn vị khác. Các hàm kích hoạt thường bị ép vào một khoảng giá trị xác định, do đó thường được gọi là hàm bẹp (squashing). Các hàm kích hoạt này thường được sử dụng là:

- ◆ 1) Hàm đồng nhất (Linear function, Identity function)

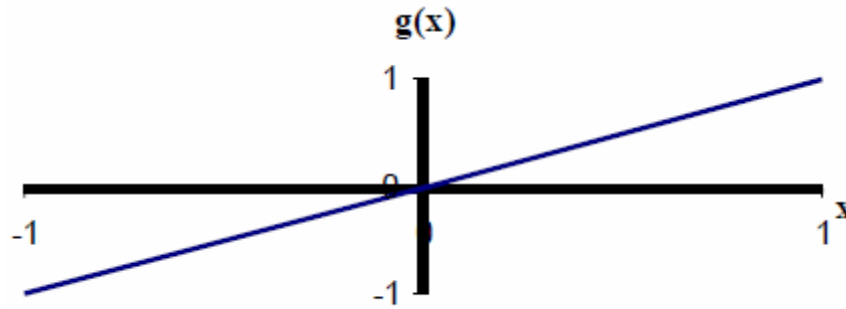
- ◆ $g(x) = x$

- ◆ Nếu coi các đầu vào là một đơn vị thì chúng sẽ sử dụng hàm này. Đôi khi một hằng số được nhân với

5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo

5.2.2. Hàm kích hoạt (hàm chuyển)

◆ Hình 5.2 Hàm đồng nhất (Identity function)



5.2 Các thành phần cơ bản của mạng nơon nhân tạo

5.2.2. Hàm kích hoạt (hàm chuyển)

- ◆ 2) Hàm bước nhị phân (Binary step function, Hard limit function)
- ◆ Hàm này cũng được biết đến với tên "Hàm ngưỡng" (Threshold function hay Heaviside function). Đầu ra của hàm này được giới hạn vào một trong hai giá trị.
- ◆ Dạng hàm này được $g(x) = \begin{cases} 1, & \text{nếu } (x \geq \theta) \\ 0, & \text{nếu } (x < \theta) \end{cases}$ các mạng chỉ có một lớp. Trong hình vẽ sau, θ được chọn bằng 1.

5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo

5.2.2. Hàm kích hoạt (hàm chuyển)

◆ Hàm bước nhị phân (Binary step function)

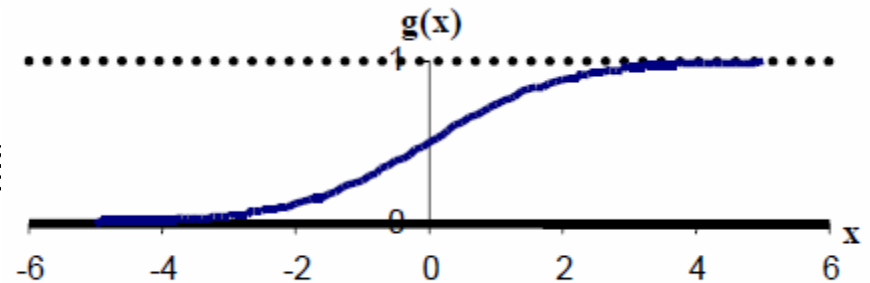


5.2.2. Hàm kích hoạt (hàm chuyển)

◆ 3) Hàm sigmoid (Sigmoid function (logsig))

$$g(x) = \frac{1}{1 + e^{-x}}$$

Hàm này đặc biệt thuận lợi khi sử dụng cho các mạng được huấn luyện (trained) bởi thuật toán Lan truyền ngược (back-propagation), vì nó dễ lấy đạo hàm, do đó có thể giảm đáng kể tính toán trong quá trình huấn luyện. Hàm này được ứng dụng cho các chương trình ứng dụng mà các đầu ra mong muốn rơi vào khoảng $[0, 1]$.



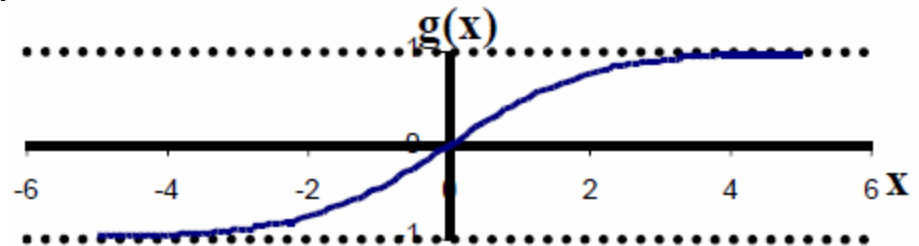
Hàm Sigmoid

5.2.2. Hàm kích hoạt (hàm chuyển)

- ◆ 4) Hàm sigmoid lưỡng cực (Bipolar sigmoid function (tansig))

$$g(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

Hàm này có các thuộc tính tương tự hàm sigmoid. Nó làm việc tốt đối với các ứng dụng có đầu ra yêu cầu trong khoảng $[-1; 1]$



Hình 5.5 Hàm sigmoid lưỡng cực

5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo

5.2.2. Hàm kích hoạt (hàm chuyển)

- ◆ Các hàm chuyển của các đơn vị ẩn (hidden units) là cần thiết để biểu diễn sự phi tuyến vào trong mạng. Lý do là hợp thành của các hàm đồng nhất là một hàm đồng nhất. Mặc dù vậy nhưng nó mang tính chất phi tuyến (nghĩa là, khả năng biểu diễn các hàm phi tuyến) làm cho các mạng nhiều tầng có khả năng rất tốt trong biểu diễn các ánh xạ phi tuyến. Tuy nhiên, đối với luật học lan truyền ngược, hàm phải khả vi (differentiable) và sẽ có ích nếu như hàm được gán trong một khoảng nào đó. Do vậy, hàm sigmoid là lựa chọn thông dụng nhất.
- ◆ Đối với các đơn vị đầu ra (output units), các hàm chuyển cần được chọn sao cho phù hợp với sự phân phối của các giá trị đích mong muốn. Chúng ta đã thấy rằng đối với các giá trị ra trong khoảng $[0, 1]$, hàm sigmoid là có ích; đối với các giá trị đích mong muốn là liên tục trong khoảng đó thì hàm này cũng vẫn có ích, nó có thể cho ta các giá trị ra hay giá trị đích được căn trong một khoảng của hàm kích hoạt đầu ra. Nhưng nếu các giá trị đích không được biết trước khoảng xác định thì hàm hay được sử dụng

5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo



5.2.1 Đơn vị xử lý

5.2.2 Hàm kết hợp

5.2.3 Hàm kích hoạt (hàm chuyển)

5.2.4 Các hình trạng của mạng

5.2.5 Mạng học (huấn luyện mạng)

5.2.6 Hàm mục tiêu

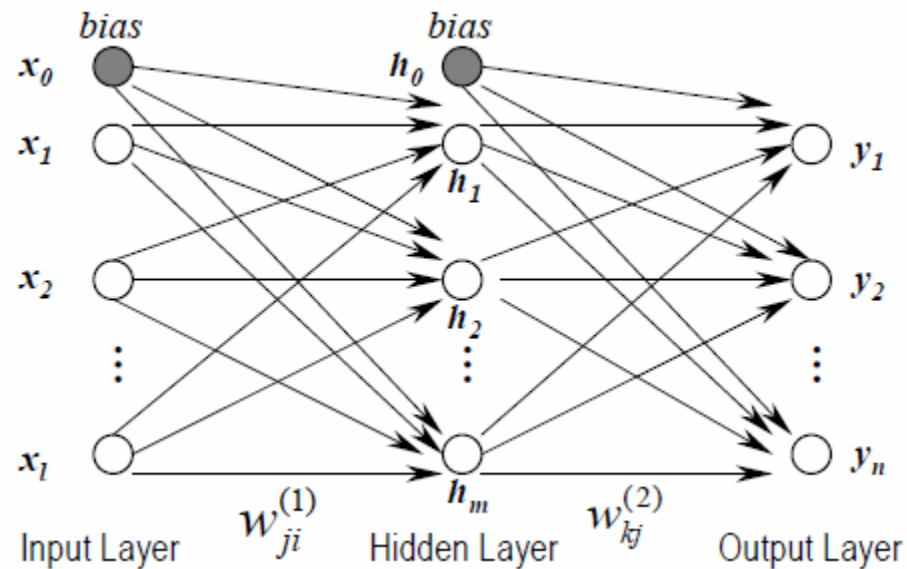
5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo

5.2.4. Các hình trạng của mạng

- ◆ Hình trạng của mạng được định nghĩa bởi: số lớp (layers), số đơn vị trên mỗi lớp, và sự liên kết giữa các lớp như thế nào. Các mạng về tổng thể được chia thành hai loại dựa trên cách thức liên kết các đơn vị:
- ◆ Mạng truyền thẳng (Feed-forward neural network): Dòng dữ liệu từ đơn vị đầu vào đến đơn vị đầu ra chỉ được truyền thẳng. Việc xử lý dữ liệu có thể mở rộng ra nhiều lớp, nhưng không có các liên kết phản hồi. Nghĩa là, các liên kết mở rộng từ các đơn vị đầu ra tới các đơn vị đầu vào trong cùng một lớp hay các lớp trước đó là không cho phép.

5.2.4. Các hình trạng của mạng

◆ Hình 5.6 Mạng nơ-ron truyền thẳng nhiều lớp



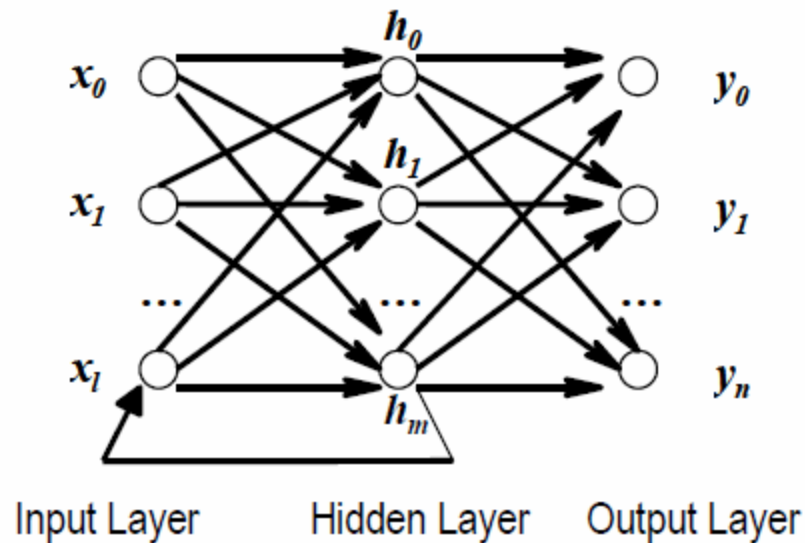
5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo

5.2.4. Các hình trạng của mạng

- ◆ Mạng hồi quy (Recurrent neural network): Có chứa các liên kết ngược. Khác với mạng truyền thẳng, các thuộc tính động của mạng mới quan trọng. Trong một số trường hợp, các giá trị kích hoạt của các đơn vị trải qua quá trình nới lỏng (tăng giảm số đơn vị và thay đổi các liên kết) cho đến khi mạng đạt đến một trạng thái ổn định và các giá trị kích hoạt không thay đổi nữa. Trong các ứng dụng khác mà cách thực hiện động tạo thành đầu ra của mạng thì những sự thay đổi các giá trị kích hoạt là đáng quan tâm.

5.2.4. Các hình trạng của mạng

◆ Hình 5.7 Mạng nơ-ron hồi quy



5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo



5.2.1 Đơn vị xử lý

5.2.2 Hàm kết hợp

5.2.3 Hàm kích hoạt (hàm chuyển)

5.2.4 Các hình trạng của mạng

5.2.5 Mạng học (huấn luyện mạng)

5.2.6 Hàm mục tiêu

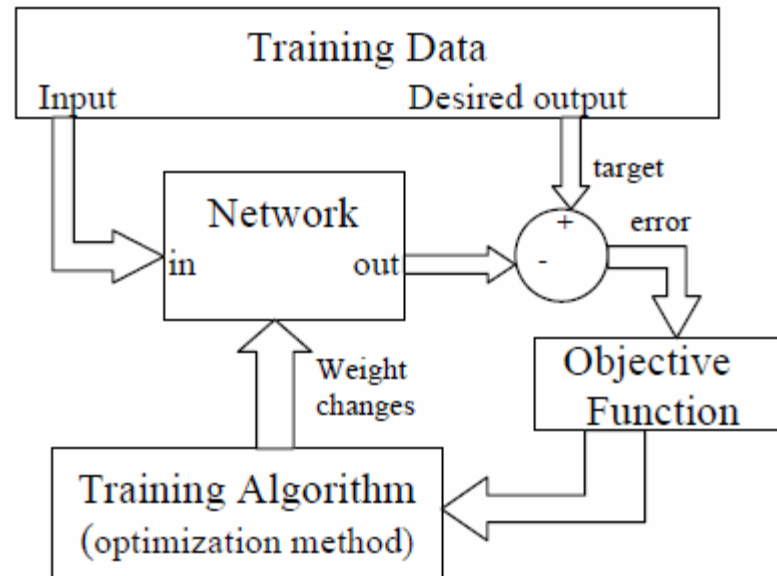
5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo

5.2.5. Mạng học (huấn luyện mạng)

- ◆ Chức năng của một mạng nơ-ron được quyết định bởi các nhân tố như: hình trạng mạng (số lớp, số đơn vị trên mỗi tầng, và cách mà các lớp được liên kết với nhau) và các trọng số của các liên kết bên trong mạng. Hình trạng của mạng thường là cố định, và các trọng số được quyết định bởi một thuật toán huấn luyện (training algorithm). Tiến trình điều chỉnh các trọng số để mạng “nhận biết” được quan hệ giữa đầu vào và đích mong muốn được gọi là học (learning) hay huấn luyện (training). Rất nhiều thuật toán học đã được phát minh để tìm ra tập trọng số tối ưu làm giải pháp cho các bài toán. Các thuật toán đó có thể chia làm hai nhóm chính: Học có thầy (Supervised learning) và Học không có thầy (Unsupervised Learning).
- ◆ Học có thầy: Mạng được huấn luyện bằng cách cung cấp cho nó các cặp mẫu đầu vào và các đầu ra mong muốn (target values). Các cặp được cung cấp bởi "thầy giáo", hay bởi hệ thống trên đó mạng hoạt động. Sự khác biệt giữa các đầu ra thực tế so với các đầu ra mong muốn được thuật toán sử dụng để thích ứng các

5.2.5. Mạng học (huấn luyện mạng)

◆ Hình 5.8 Mô hình Học có thầy



5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo

5.2.5. Mạng học (huấn luyện mạng)

- **Học không có thầy:** Với cách học không có thầy, không có phản hồi từ môi trường để chỉ ra rằng đầu ra của mạng là đúng. Mạng sẽ phải khám phá các đặc trưng, các điều chỉnh, các mối tương quan, hay các lớp trong dữ liệu vào một cách tự động. Trong thực tế, đối với phần lớn các biến thể của học không có thầy, các đích trùng với đầu vào. Nói một cách khác, học không có thầy luôn thực hiện một công việc tương tự như một mạng tự liên hợp, cô đọng thông tin từ dữ liệu vào.

5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo



5.2.1 Đơn vị xử lý

5.2.2 Hàm kết hợp

5.2.3 Hàm kích hoạt (hàm chuyển)

5.2.4 Các hình trạng của mạng

5.2.5 Mạng học (huấn luyện mạng)

5.2.6 Hàm mục tiêu

5.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo

5.2.6. Hàm mục tiêu

- ◆ Để huấn luyện một mạng và xét xem nó thực hiện tốt đến đâu, ta cần xây dựng một hàm mục tiêu (hay hàm giá) để cung cấp cách thức đánh giá khả năng hệ thống một cách không nhập nhằng. Việc chọn hàm mục tiêu là rất quan trọng bởi vì hàm này thể hiện các mục tiêu thiết kế và quyết định thuật toán huấn luyện nào có thể được áp dụng. Để phát triển một hàm mục tiêu đo được chính xác cái chúng ta muốn không phải là việc dễ dàng. Một vài hàm cơ bản được sử dụng rất rộng rãi. Một trong số chúng là hàm tổng bình phương lỗi (sum of squares error function),

5.2 Các thành phần cơ bản của mạng nơon nhân tạo

5.2.6. Hàm mục tiêu

- ◆ trong đó:
- ◆ p : số thứ tự mẫu trong tập huấn luyện
- ◆ i : số thứ tự của đơn vị đầu ra
- ◆ t_{pi} và y_{pi} : tương ứng là đầu ra mong muốn và đầu ra thực tế của mạng cho đơn vị đầu ra thứ i trên mẫu thứ p .
- ◆ Trong các ứng dụng thực tế, nếu cần thiết có thể làm phức tạp hàm số với một vài yếu tố khác để có thể kiểm soát được sự phức tạp của mô hình.

$$E = \frac{1}{NP} \sum_{p=1}^P \sum_{i=1}^N (t_{pi} - y_{pi})^2$$

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

A vertical graphic on the left side of the slide showing a globe with a blue and white color scheme, surrounded by a yellow and white glow.

5.3.1

Mạng truyền thẳng

5.3.2

Vấn đề thiết kế cấu trúc mạng

5.3.3

Thuật toán lan truyền ngược (Back-Propagation)

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

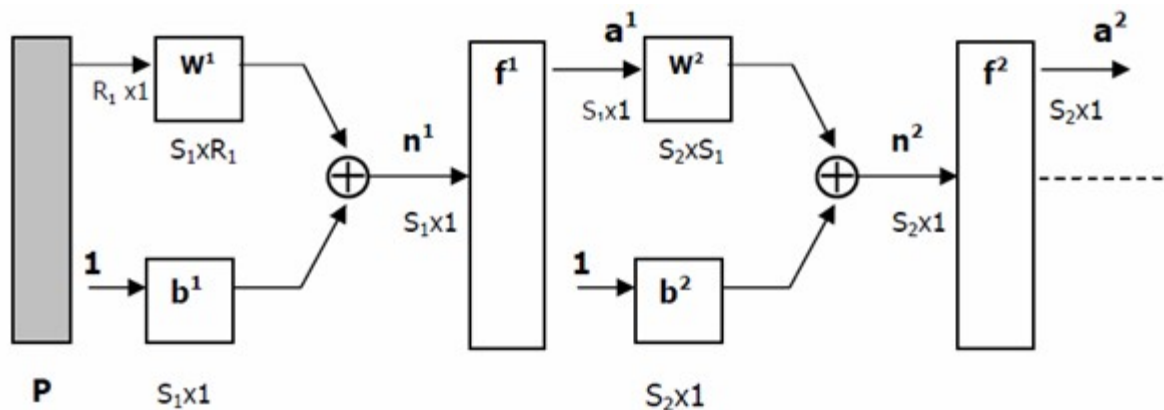
5.3.1. Mạng truyền thẳng

- ◆ Một mạng truyền thẳng nhiều lớp bao gồm một lớp vào, một lớp ra và một hoặc nhiều lớp ẩn. Các nơon đầu vào thực chất không phải các nơon theo đúng nghĩa, bởi lẽ chúng không thực hiện bất kỳ một tính toán nào trên dữ liệu vào, đơn giản nó chỉ tiếp nhận các dữ liệu vào và chuyển cho các lớp kế tiếp. Các nơon ở lớp ẩn và lớp ra mới thực sự thực hiện các tính toán, kết quả được định dạng bởi hàm đầu ra (hàm chuyển). Cụm từ “truyền thẳng” (feed forward) (không phải là trái nghĩa của lan truyền ngược) liên quan đến một thực tế là tất cả các nơon chỉ có thể được kết nối với nhau theo một hướng: tới một hay nhiều các nơon khác trong lớp kế tiếp (loại trừ các nơon ở lớp ra).

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.1. Mạng truyền thẳng

- ◆ Hình sau ở dạng tóm tắt biểu diễn mạng nơon một cách cô đọng và tránh gây ra sự hiểu nhầm.
- ◆ **Hình 5.9 Mạng nơon truyền thẳng nhiều lớp**



5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.1. Mạng truyền thẳng

- ◆ trong đó:
- ◆ P : Vector đầu vào (vector cột)
- ◆ W_i : Ma trận trọng số của các nơon lớp thứ i .
- ◆ ($S_i R_i$: S hàng (nơon) - R cột (số đầu vào))
- ◆ b_i : Vector độ lệch (bias) của lớp thứ i ($S_i \times 1$: cho S nơon)
- ◆ n_i : net input ($S_i \times 1$)
- ◆ f_i : Hàm chuyển (hàm kích hoạt)
- ◆ a_i : net output ($S_i \times 1$)
- ◆ \oplus : Hàm tổng thông thường.

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.1. Mạng truyền thẳng

- ◆ 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, thể hiện trạng thái kích thích, hay âm, thể hiện trạng thái kiềm chế. Mỗi nơon tính toán mức kích hoạt của chúng bằng cách cộng tổng các đầu vào và đưa ra hàm chuyển. Một khi đầu ra của tất cả các nơon trong một lớp mạng cụ thể đã thực hiện xong tính toán thì lớp kế tiếp có thể bắt đầu thực hiện tính toán của mình bởi vì đầu ra của lớp hiện tại tạo ra đầu vào của lớp kế tiếp. Khi tất cả các nơon đã thực hiện tính toán thì kết quả được trả lại bởi các nơon đầu ra. Tuy nhiên, có thể là chưa đúng yêu cầu, khi đó một thuật toán huấn luyện cần được áp dụng để điều chỉnh các tham số của mạng.

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.1. Mạng truyền thẳng

- ◆ Trong Hình 5.9, số nơon ở lớp thứ nhất, và lớp thứ hai tương ứng là S_1 và S_2 . Ma trận trọng số đối với các lớp tương ứng là W_1 và W_2 . Có thể thấy sự liên kết giữa các lớp mạng thể hiện trong Hình 5.9: ở lớp thứ 2, vector đầu vào chính là net output của lớp thứ nhất.
- ◆ Tương tự như vậy, nếu thêm vào các lớp khác nữa vào trong cấu trúc này thì lớp mạng cuối cùng thường là lớp cho ra kết quả của toàn bộ mạng, lớp đó gọi là lớp ra (OUTPUT LAYER).

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.1. Mạng truyền thẳng

- ◆ Mạng có nhiều lớp có khả năng tốt hơn là các mạng chỉ có một lớp, chẳng hạn như mạng hai lớp với lớp thứ nhất sử dụng hàm sigmoid và lớp thứ hai dùng hàm đồng nhất có thể áp dụng để xấp xỉ các hàm toán học khá tốt, trong khi các mạng chỉ có một lớp thì không có khả năng này.
- ◆ Xét trường hợp mạng có hai lớp như Hình 5.9, công thức tính toán cho đầu ra như sau:
- ◆ $a_2 = f_2(W_2(f_1(W_1P + b_1))) + b_2$
- ◆ trong đó, ý nghĩa của các ký hiệu như đã nêu trong Hình 5.9.

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

A vertical graphic on the left side of the slide showing a globe with a blue and white color scheme, surrounded by a yellow and white glow.

5.3.1

Mạng truyền thẳng

5.3.2

Vấn đề thiết kế cấu trúc mạng

5.3.3

Thuật toán lan truyền ngược (Back-Propagation)

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.2. Vấn đề thiết kế cấu trúc mạng

- ◆ Mặc dù, về mặt lý thuyết, có tồn tại một mạng có thể mô phỏng một bài toán với độ chính xác bất kỳ. Tuy nhiên, để có thể tìm ra mạng này không phải là điều đơn giản. Để định nghĩa chính xác một kiến trúc mạng như: cần sử dụng bao nhiêu lớp ẩn, mỗi lớp ẩn cần có bao nhiêu đơn vị xử lý cho một bài toán cụ thể là một công việc hết sức khó khăn. Dưới đây trình bày một số vấn đề cần quan tâm khi ta thiết kế một mạng.

- **Số lớp ẩn**

- ◆ Vì các mạng có hai lớp ẩn có thể thể hiện các hàm với dáng điệu bất kỳ, nên, về lý thuyết, không có lý do nào sử dụng các mạng có nhiều hơn hai lớp ẩn. Người ta đã xác định rằng đối với phần lớn các bài toán cụ thể, chỉ cần sử dụng một lớp ẩn cho mạng là đủ. Các bài toán sử dụng hai lớp ẩn hiếm khi xảy ra trong thực tế. Thậm chí đối với các bài toán cần sử dụng nhiều hơn một lớp ẩn thì trong phần lớn các trường hợp trong thực tế, sử dụng chỉ một lớp ẩn cho ta hiệu năng tốt hơn là sử dụng nhiều hơn một lớp.

Việc huấn luyện mạng thường rất chậm khi mà số lớp ẩn sử dụng

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.2. Vấn đề thiết kế cấu trúc mạng

- ◆ 1) Phần lớn các thuật toán luyện mạng cho các mạng nơon truyền thẳng đều dựa trên phương pháp gradient. Các lớp thêm vào sẽ thêm việc phải lan truyền các lỗi làm cho vector gradient rất không ổn định. 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 có thể không tìm ra cực trị toàn cục. Mặc dù thuật toán luyện mạng có thể tìm ra cực trị toàn cục, nhưng xác suất khá cao là chúng ta sẽ bị tắc trong một cực trị địa phương sau rất nhiều thời gian lặp và khi đó, ta phải bắt đầu lại.

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.2. Vấn đề thiết kế cấu trúc mạng


- ◆ 3) Dĩ nhiên, có thể đối với một bài toán cụ thể, sử dụng nhiều hơn một lớp ẩn với chỉ một vài đơn vị thì tốt hơn là sử dụng ít lớp ẩn với số đơn vị là lớn, đặc biệt đối với các mạng cần phải học các hàm không liên tục. Về tổng thể, người ta cho rằng việc đầu tiên là nên xem xét khả năng sử dụng mạng chỉ có một lớp ẩn. Nếu dùng một lớp ẩn với một số lượng lớn các đơn vị mà không có hiệu quả thì nên sử dụng thêm một lớp ẩn nữa với một số ít các đơn vị.

■ Số đơn vị trong lớp ẩn

- ◆ Một vấn đề quan trọng trong việc thiết kế một mạng là cần có bao nhiêu đơn vị trong mỗi lớp. Sử dụng quá ít đơn vị có thể dẫn đến việc không thể nhận dạng được các tín hiệu đầy đủ trong một tập dữ liệu phức tạp, hay thiếu ăn khớp (underfitting). Sử dụng quá nhiều đơn vị sẽ tăng thời gian luyện mạng, có lẽ là quá nhiều để luyện khi mà không thể luyện mạng trong một khoảng thời gian hợp lý. Số lượng lớn các đơn vị có thể dẫn đến tình trạng thừa ăn khớp (overfitting), trong trường hợp này mạng có quá nhiều thông

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.2. Vấn đề thiết kế cấu trúc mạng

- 
- ◆ Số lượng tốt nhất của các đơn vị ẩn phụ thuộc vào rất nhiều yếu tố - số đầu vào, đầu ra của mạng, số trường hợp trong tập mẫu, độ nhiễu của dữ liệu đích, độ phức tạp của hàm lỗi, kiến trúc mạng và thuật toán luyện mạng. Có rất nhiều “luật” để lựa chọn số đơn vị trong các lớp, chẳng hạn:
 - ◆ • $m \in [l, n]$ - nằm giữa khoảng kích thước lớp vào, lớp ra
 - ◆ • tổng kích thước lớp vào và lớp ra
 - ◆ • $m < 2l$ - nhỏ hơn hai lần kích thước lớp vào
 - ◆ • - căn bậc hai của tích kích thước lớp vào và lớp ra.

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.2. Vấn đề thiết kế cấu trúc mạng

- ◆ Các luật này chỉ có thể được coi như là các lựa chọn thô khi chọn lựa kích thước của các lớp. Chúng không phản ánh được thực tế, bởi lẽ chúng chỉ xem xét đến nhân tố kích thước đầu vào, đầu ra mà bỏ qua các nhân tố quan trọng khác như: số trường hợp đưa vào huấn luyện, độ nhiễu ở các đầu ra mong muốn, độ phức tạp của hàm lỗi, kiến trúc của mạng (truyền thẳng hay hồi quy), và thuật toán học.
- ◆ Trong phần lớn các trường hợp, không có một cách để có thể dễ dàng xác định được số tối ưu các đơn vị trong lớp ẩn mà không phải luyện mạng sử dụng số các đơn vị trong lớp ẩn khác nhau và dự báo lỗi tổng quát hóa của từng lựa chọn. Cách tốt nhất là sử dụng phương pháp thử-sai (trial-and-error). Trong thực tế, có thể sử dụng phương pháp Lựa chọn tiến (forward selection) hay Lựa chọn lùi (backward selection) để xác định số đơn vị trong lớp ẩn.

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.2. Vấn đề thiết kế cấu trúc mạng

- ◆ Lựa chọn tiến bắt đầu với việc chọn một luật hợp lý cho việc đánh giá hiệu năng của mạng. Sau đó, ta chọn một số nhỏ các đơn vị ẩn, luyện và thử mạng; ghi lại hiệu năng của mạng. Sau đó, tăng một chút số đơn vị ẩn; luyện và thử lại cho đến khi lỗi là chấp nhận được, hoặc không có tiến triển đáng kể so với trước.
- ◆ Lựa chọn lùi, ngược với lựa chọn tiến, bắt đầu với một số lớn các đơn vị trong lớp ẩn, sau đó giảm dần đi. Quá trình này rất tốn thời gian nhưng sẽ giúp ta tìm được số lượng đơn vị phù hợp cho lớp ẩn.

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

A vertical graphic on the left side of the slide showing a globe with a blue and white color scheme, surrounded by a yellow and white glow.

5.3.1

Mạng truyền thẳng

5.3.2

Vấn đề thiết kế cấu trúc mạng

5.3.3

Thuật toán lan truyền ngược (Back-Propagation)


5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.3. Thuật toán lan truyền ngược (Back-Propagation)

- ◆ Cần có một sự phân biệt giữa kiến trúc của một mạng và thuật toán học của nó,
- ◆ về cơ bản có hai dạng thuật toán để luyện mạng:
 - **học có thầy**
 - **và học không có thầy.**
- ◆ Các mạng nơ-ron truyền thẳng nhiều lớp được luyện bằng phương pháp học có thầy. mạng sẽ học thông qua những sai sót của nó.

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.3. Thuật toán lan truyền ngược (Back-Propagation)

- 
- ◆ Về cơ bản, thuật toán lan truyền ngược là dạng tổng quát của thuật toán trung bình bình phương tối thiểu (Least Means Square-LMS). Thuật toán này 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 mà trong quá trình tìm hiểu bài toán đặt ra.
 - **Mô tả thuật toán**
 - ◆ Ta sẽ sử dụng dạng tổng quát của mạng nơron truyền thẳng nhiều lớp như trong hình vẽ 5.9 của phần trước. Khi đó, đầu ra của một lớp trở thành đầu vào của lớp kế tiếp. Phương trình thể hiện hoạt động này như sau:

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.3. Thuật toán lan truyền ngược (Back-Propagation)

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

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

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.3. Thuật toán lan truyền ngược (Back-Propagation)

■ Chỉ số hiệu năng (performance index)

- ◆ Cũng tương tự như thuật toán LMS, thuật toán lan truyền ngược 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:
 - ◆ $\{(p_1, t_1), (p_2, t_2), \dots, (p_Q, t_Q)\}$,
 - ◆ 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.

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.3. Thuật toán lan truyền ngược (Back-Propagation)

- ◆ 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, ta có thể viết lại phương trình trên ở 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:
- ◆ ký hiệu \hat{x} là giá trị xấp xỉ của $F(x)$ thì:
- ◆ $\hat{x} = (t(k) - a(k))^T (t(k) - a(k)) = e^T(k)e(k)$,
- ◆ 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:

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.3. Thuật toán lan truyền ngược (Back-Propagation)

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

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

- ◆ trong đó α là hệ số học.
- ◆ Như vậy, mọi chuyện đến đây đều giống như thuật toán trung bình bình phương tối thiểu. Tiếp theo chúng ta sẽ đi vào phần khó nhất của thuật toán: tính các đạo hàm từng phần.

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.3. Thuật toán lan truyền ngược (Back-Propagation)

▪ 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 ta phải sử dụng luật xích để tính. Luật này được mô tả như sau: giả sử ta có một hàm f là một hàm của biến n , ta muốn tính đạo hàm của f có liên quan đến một biến w khác. Luật xích này như sau:

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

Ta sẽ dùng phương pháp này để tính các đạo hàm trong (+) và (++) ở phần trước.

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.3. Thuật toán lan truyền ngược (Back-Propagation)

$$\frac{\partial \hat{F}}{w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \cdot \frac{\partial n_i^m}{\partial w_{i,j}^m},$$

$$\frac{\partial \hat{F}}{b_i^m} = \frac{\partial \hat{F}}{\partial n_i^m} \cdot \frac{\partial n_i^m}{\partial b_i^m},$$

- ◆ trong đó hạng thức thứ hai 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 ta có:

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.3. Thuật toán lan truyền ngược (Back-Propagation)

- ◆ là độ nhạy cảm của đối với các thay đổi của phần tử thứ i của đầu vào của mạng tại lớp thứ m . Khi đó ta có:

Ký hiệu

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

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

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

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

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.3. Thuật toán lan truyền ngược (Back-Propagation)

- ◆ Bây giờ, ta có thể phát biểu thuật toán giảm theo hướng (gradient descent) 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:

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

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.3. Thuật toán lan truyền ngược (Back-Propagation)

▪ Lan truyền ngược độ nhạy cảm

- ◆ Bây giờ ta cần tính nốt ma trận độ nhạy cảm sm. Để thực hiện điều này cần sử dụng một áp dụng khác của luật xích. Quá trình này cho ta khái niệm về sự “lan truyền ngược” bởi vì nó mô tả mối quan hệ hồi quy trong đó độ nhạy cảm sm được tính qua độ nhạy cảm sm+1 của lớp m + 1.
- ◆ Để dẫn đến quan hệ đó, ta sử dụng ma trận Jacobi sau:

5.3 Mạng truyền thẳng và thuật toán lan truyền ngược

5.3.3. Thuật toán lan truyền ngược (Back-Propagation)

- ◆ 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, ta có thể viết lại phương trình trên ở 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:
- ◆ ký hiệu \hat{x} là giá trị xấp xỉ của $F(x)$ thì:
- ◆ $\hat{x} = (t(k) - a(k))^T (t(k) - a(k)) = e^T(k)e(k)$,
- ◆ 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:

5.4 Một số ứng dụng của mạng nơron

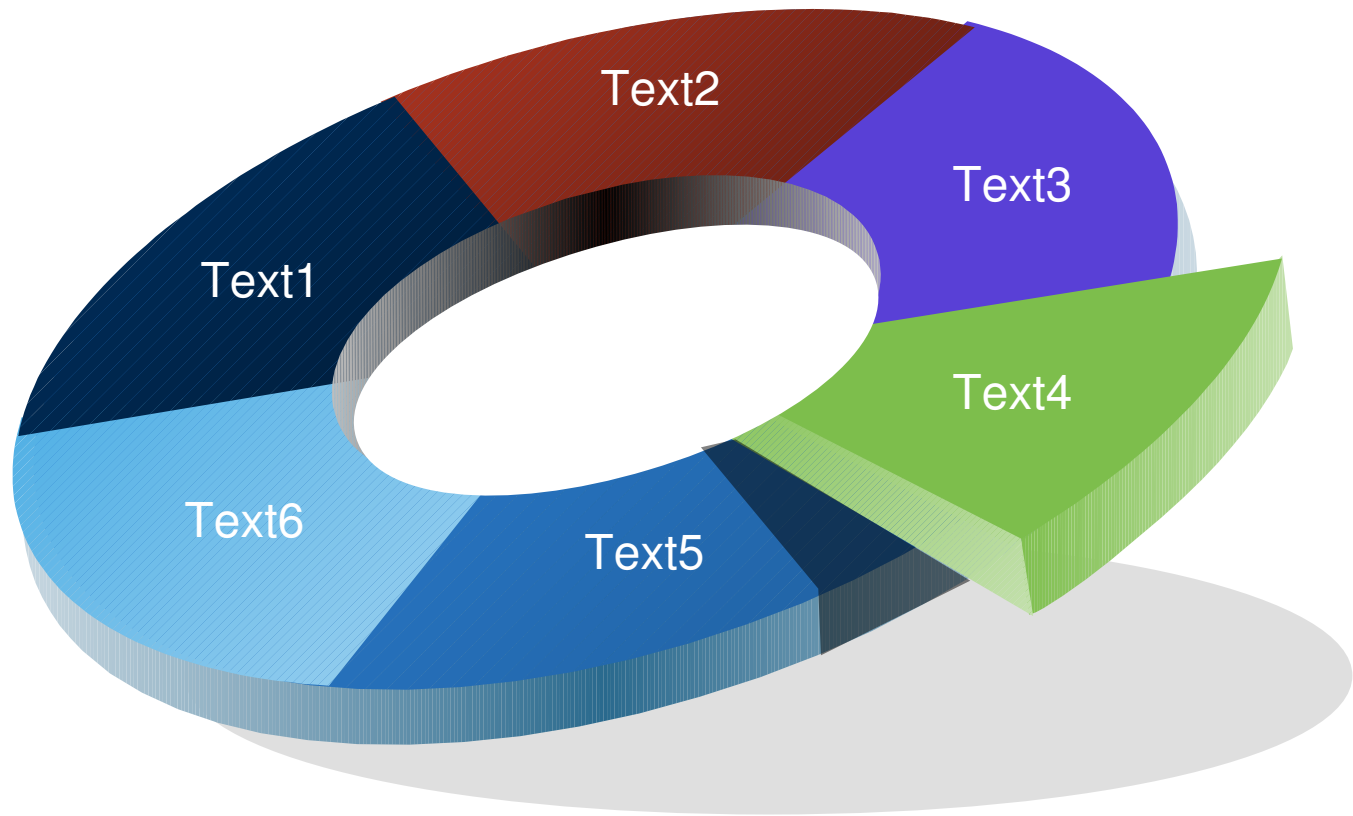


- ◆ How do I incorporate my logo to a slide that will apply to all the other slides?
 - On the [View] menu, point to [Master], and then click [Slide Master] or [Notes Master]. Change images to the one you like, then it will apply to all the other slides.

[Image information in product]

- Image :CD_ digital-globe (ImageState)
- Note to customers : This image has been licensed to be used within this PowerPoint template only.
You may not extract the image for any other use.

3-D Pie Chart



Progress Diagram



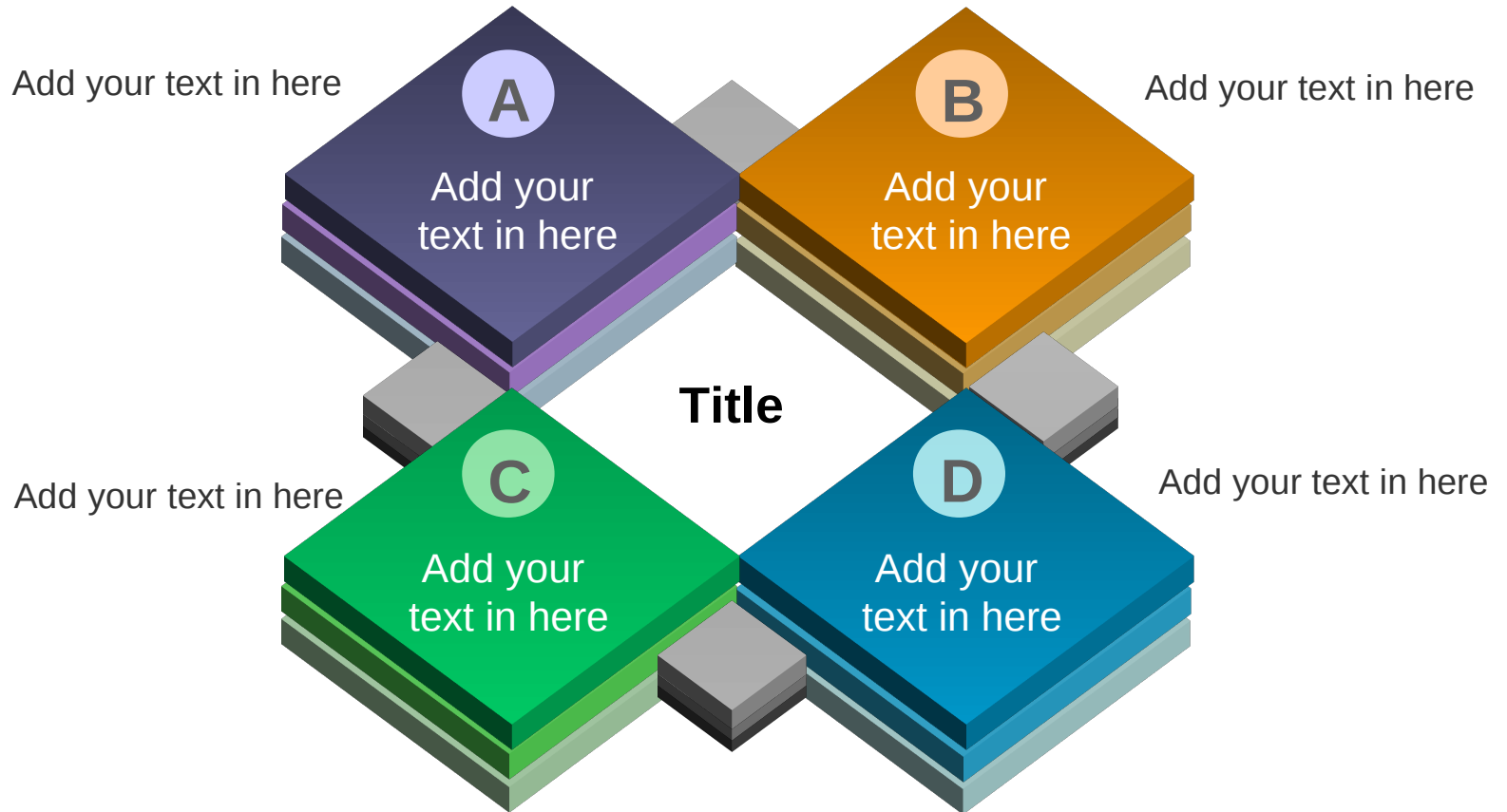
Phase 1

Phase 2

Phase 3



Diagram





TRƯỜNG CAO ĐẲNG CNTT HỮU NGHỊ VIỆT - HÀN

KHOA KHOA HỌC MÁY TÍNH

-----***-----

Thank You !

www.themegallery.com