Chương 8: Máy học

Nội dụng

- Các khái niệm về máy học
- Các kỹ thuật học của máy
  - Cây quyết định
  - Mạng neuron
  - Giải thuật di truyền (Genetic)

Học Máy (Machine Learning)

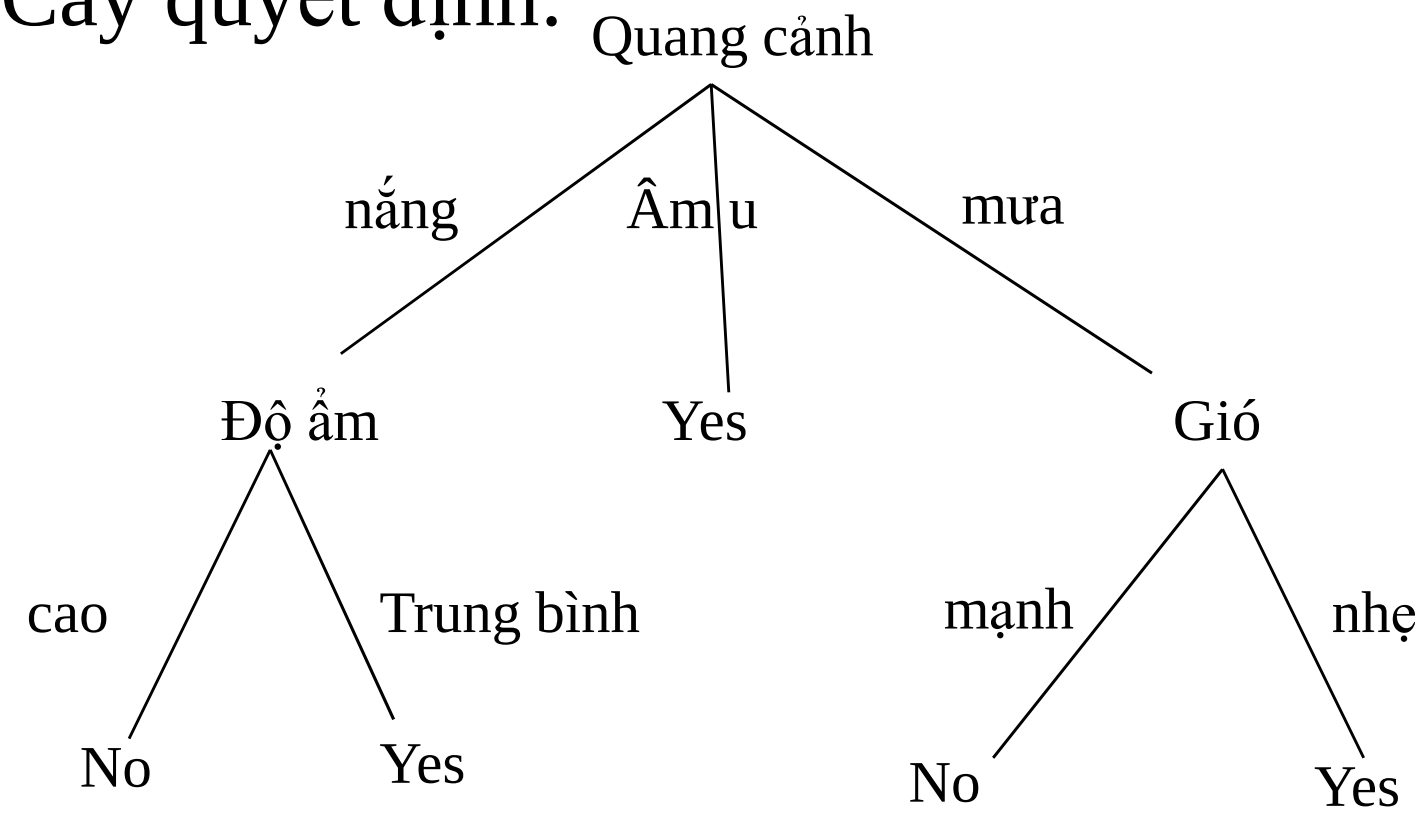
- Học (learning) là bất cứ sự thay đổi nào trong một hệ thống cho phép nó tiến hành tốt hơn trong lần thứ hai khi lặp lại cùng một nhiệm vụ hoặc với nhiệm vụ khác từ cùng một quần thể đó. (Herbert Simon)
- Học liên quan đến vấn đề khái quát hóa từ kinh nghiệm (dữ liệu rèn luyện) => bài toán *quy nạp* (induction)
- Vì dữ liệu rèn luyện thường hạn chế, nên thường khái quát hóa theo một số khía cạnh nào đó (heuristic) => tính *thiên lệch quy nạp* (inductive bias)
- Có ba tiếp cận học:
  - Các phương pháp học dựa trên ký hiệu (symbol-based): ID3
  - Tiếp cận kết nối: Các mạng neuron sinh học
  - Tiếp cận di truyền hay tiến hóa: giải thuật genetic

Cây quyết định (ID3)

- Là một giải thuật học đơn giản nhưng thành công
- Cây quyết định (QĐ) là một cách biểu diễn cho phép chúng ta xác định phân loại của một đối tượng bằng cách kiểm tra giá trị của một số thuộc tính.
- Giải thuật có:
  - **Đầu vào:** Một đối tượng hay một tập hợp các thuộc tính mô tả một tình huống
  - **Đầu ra:** thường là quyết định yes/no, hoặc các phân loại.
- Trong cây quyết định:
  - Mỗi nút trong biểu diễn một sự kiểm tra trên một thuộc tính nào đó, mỗi giá trị có thể của nó tương đương với một nhánh của cây
  - Các nút lá thể hiện sự phân loại.
- Kích cỡ của cây QĐ tùy thuộc vào thứ tự của các kiểm tra trên các thuộc tính.

Ví dụ Cây QĐ: Chơi Tennis

- Mục đích: học để xem có chơi Tennis không?
- Cây quyết định:



Quy nạp cây QĐ từ các ví dụ

- Ví dụ (hay dữ liệu rèn luyện cho hệ thống) gồm:

  Giá trị của các thuộc tính + Phân loại của ví dụ

Ngày	Quang cảnh	Nhiệt độ	Độ ẩm	Gió	Chơi Tennis
D1	Nắng	Nóng	Cao	nhẹ	Không
D2	Nắng	Nóng	Cao	Mạnh	Không
D3	Âm u	Nóng	Cao	Nhẹ	Có
D4	Mưa	ấm áp	Cao	nhẹ	Có
D5	Mưa	Mát	TB	nhẹ	Có
D6	Mưa	Mát	TB	Mạnh	Không
D7	Âm u	Mát	TB	Mạnh	Có
D8	Nắng	ấm áp	Cao	nhẹ	Không
D9	Nắng	Mát	TB	nhẹ	Có
D10	Mưa	ấm áp	TB	nhẹ	Có
D11	Nắng	ấm áp	TB	Mạnh	Có
D12	Âm u	ấm áp	Cao	Mạnh	Có
D13	Âm u	Nóng	TB	nhẹ	Có
D14	Mưa	ấm áp	Cao	Mạnh	không

Làm sao để học được cây QĐ

- Tiếp cận đơn giản
  - Học một cây mà có một lá cho mỗi ví dụ.
  - Học thuộc lòng một cách hoàn toàn các ví dụ.
  - Có thể sẽ không thực hiện tốt trong các trường hợp khác.
- Tiếp cận tốt hơn:
  - Học một cây nhỏ nhưng chính xác phù hợp với các ví dụ
  - Occam's razor – cái đơn giản thường là cái tốt nhất!

    Giả thuyết có khả năng nhất là giả thuyết đơn giản nhất thống nhất với tất cả các quan sát.

Xây dựng cây QĐ: Trên - xuống

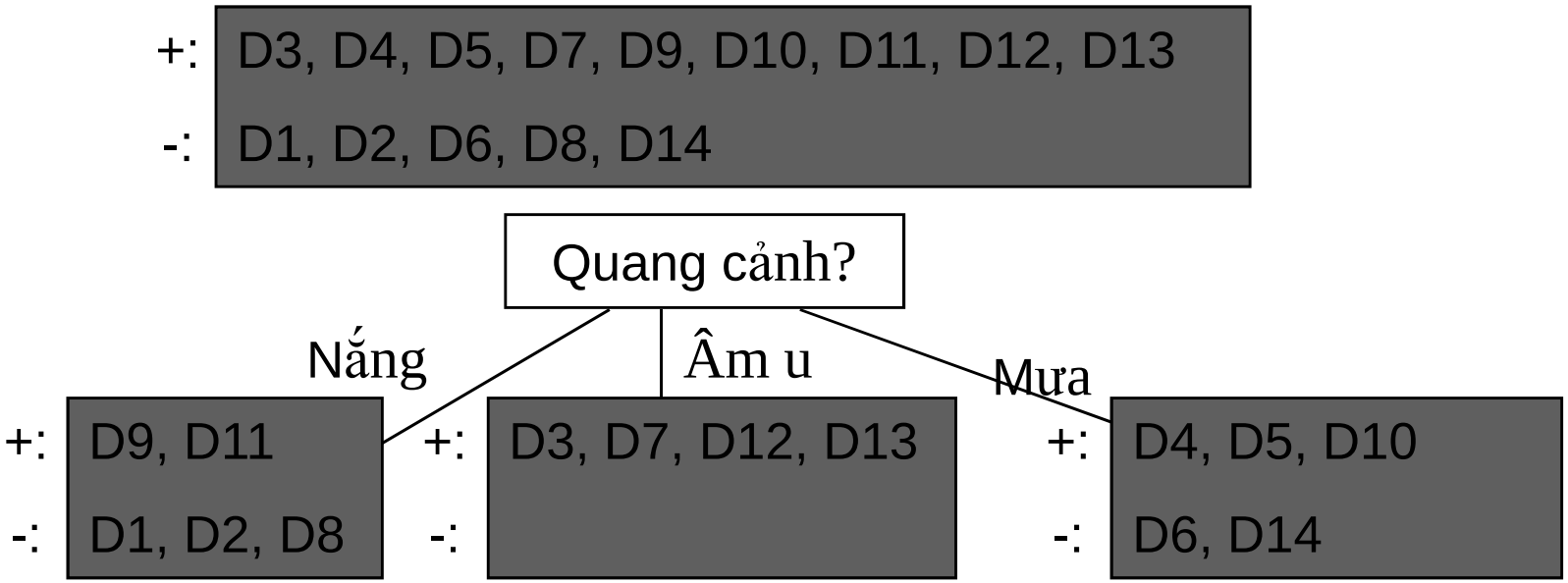
Vòng lặp chính:

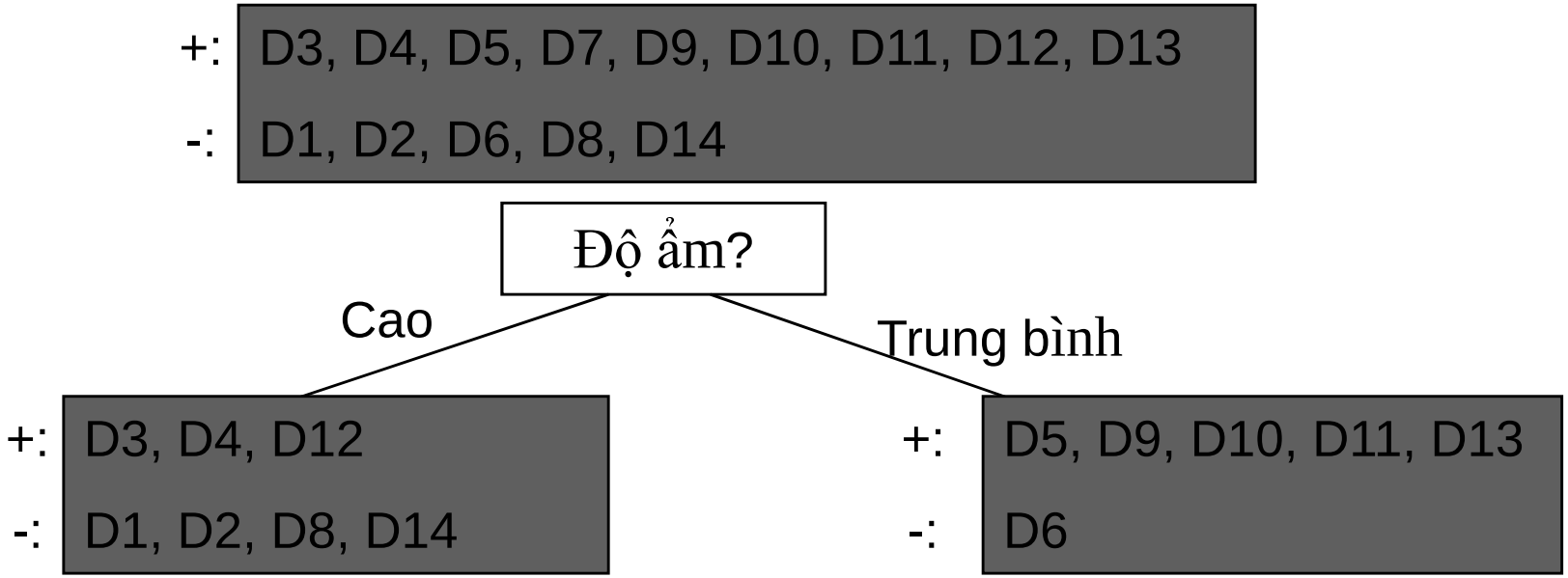
1. A <- thuộc tính quyết định tốt nhất cho nút kế
2. Gán A là thuộc tính quyết định cho nút
3. Với mỗi giá trị của A, tạo một nút con mới cho nút
4. Sắp xếp các ví dụ vào các nút lá
5. If các ví dụ đã được phân loại đúng, dừng ctr; Else lặp lại trên mỗi nút lá mới

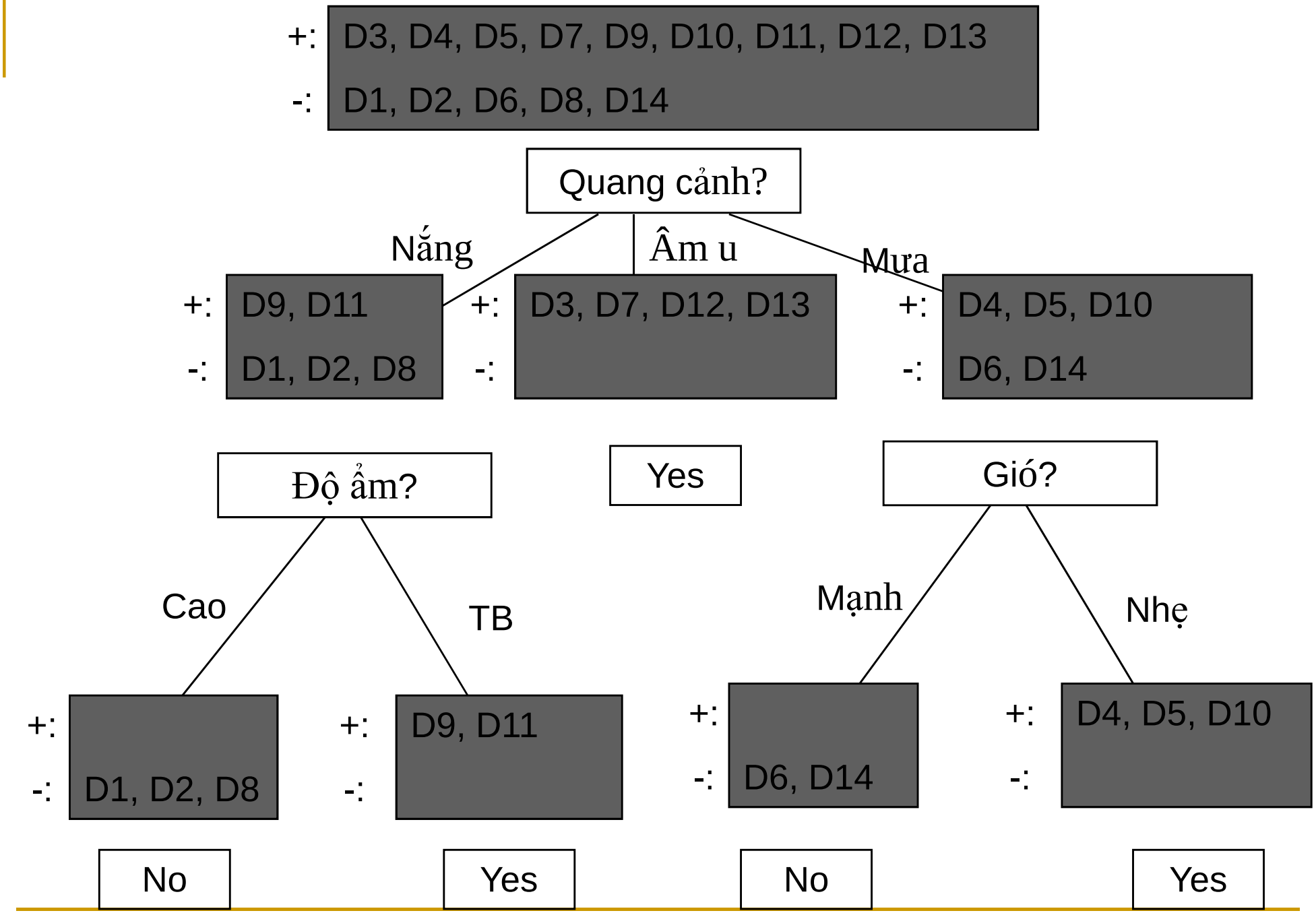
Để phân loại một trường hợp, có khi cây QĐ không cần sử dụng tất cả các thuộc tính đã cho, mặc dù nó vẫn phân loại đúng tất cả các ví dụ.

Các khả năng có thể của nút con

- Các ví dụ có cả âm và dương:
  - Tách một lần nữa
- Tất cả các ví dụ còn lại đều âm hoặc đều dương
  - trả về cây quyết định
- Không còn ví dụ nào
  - trả về mặc nhiên
- Không còn thuộc tính nào (nhiễu)
  - Quyết định dựa trên một luật nào đó (luật đa số)


+: D3, D4, D5, D7, D9, D10, D11, D12, D13
-: D1, D2, D6, D8, D14
Quang cảnh?
Nắng
Âm u
Mưa
+: D9, D11
-: D1, D2, D8
+: D3, D7, D12, D13
-:
+: D4, D5, D10
-: D6, D14


+: D3, D4, D5, D7, D9, D10, D11, D12, D13
-: D1, D2, D6, D8, D14
Độ ẩm?
Cao
Trung bình
+: D3, D4, D12
-: D1, D2, D8, D14
+: D5, D9, D10, D11, D13
-: D6


+: D3, D4, D5, D7, D9, D10, D11, D12, D13
-: D1, D2, D6, D8, D14
Quang cảnh?
Nắng
Âm u
Mưa
+: D9, D11
-: D1, D2, D8
+: D3, D7, D12, D13
-:
+: D4, D5, D10
-: D6, D14
Độ ẩm?
Yes
Gió?
Cao
TB
Mạnh
Nhẹ
+:
-: D1, D2, D8
+: D9, D11
-:
+:
-: D6, D14
+: D4, D5, D10
-:
No
Yes
No
Yes

ID3 xây dựng cây QĐ theo giải thuật sau:

```
function induce_tree (example_set, Properties)

begin
if all entries in example_set are in the same class
   then return a leaf node labeled with that class
   else if Properties is empty
      then return leaf node labeled with disjunction of all classes in example_set
      else begin
         select a property, P, and make it the root of the current tree;
         delete P from Properties;
            for each value, V, of P,
               begin
                  create a branch of the tree labeled with V;
                  let partition_V be elements of example_set with values V for property P;
                  call induce_tree(partition_V, Properties), attach result to branch V
               end
      end
end
```

Đánh giá hiệu suất

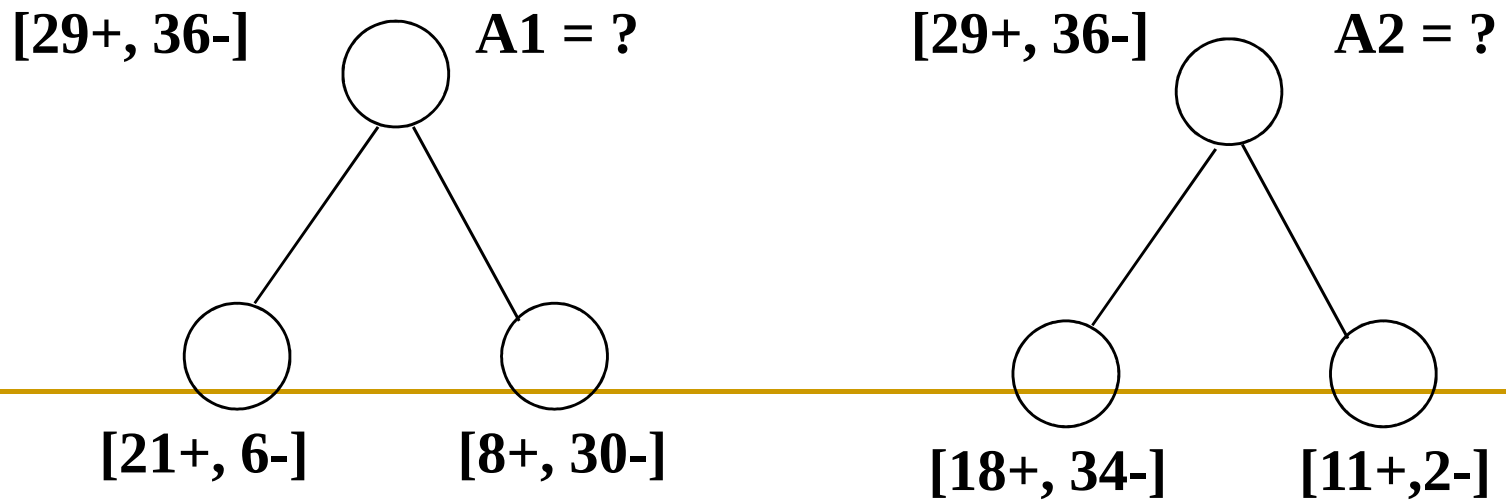
- Chúng ta muốn có một cây QĐ có thể phân loại đúng một ví dụ mà nó chưa từng thấy qua.
- Việc học sử dụng một "tập rèn luyện" (traning set), và
- Việc đánh giá hiệu suất sử dụng một "tập kiểm tra" (test set):

1. Thu thập một tập hợp lớn các ví dụ
2. Chia thành tập rèn luyện và tập kiểm tra
3. Sử dụng giải thuật và tập rèn luyện để xây dựng giả thuyết h (cây QĐ)
4. Đo phần trăm tập kiểm tra được phân loại đúng bởi h
5. Lặp lại bước 1 đến 4 cho các kích cỡ tập kiểm tra khác nhau được chọn một cách nhẫu nhiên.

Sử dụng lý thuyết thông tin

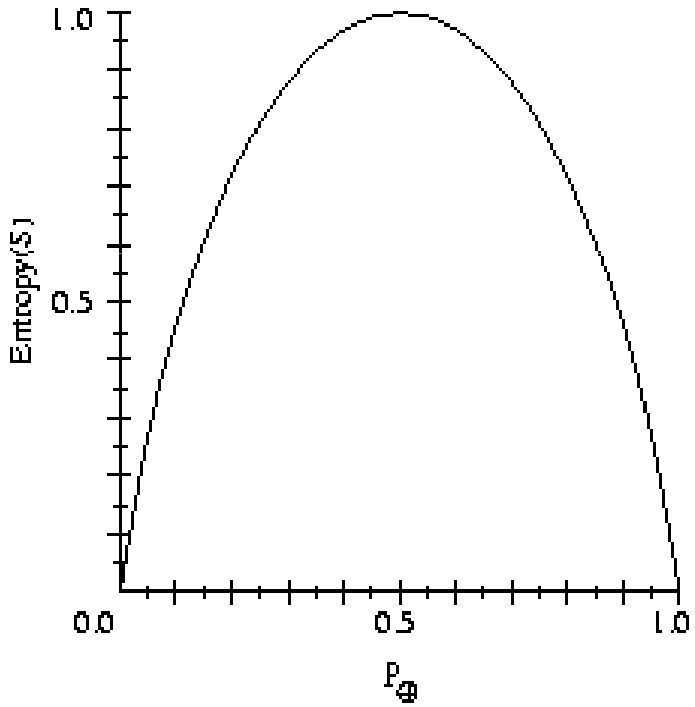
- Chúng ta muốn chọn các thuộc tính có thể giảm thiểu chiều sâu của cây QĐ.
- Thuộc tính tốt nhất: chia các ví dụ vào các tập hợp chứa toàn ví dụ âm hoặc ví dụ dương.
- Chúng ta cần một phép đo để xác định thuộc tính nào cho khả năng chia tốt hơn.

Thuộc tính nào tốt hơn?



Entropy

- Entropy(S) = số lượng mong đợi các bit cần thiết để mã hóa một lớp (+ hay –) của một thành viên rút ra một cách ngẫu nhiên từ S (trong trường hợp tối ưu, mã có độ dài ngắn nhất).
- Theo lý thuyết thông tin: mã có độ dài tối ưu là mã gán $-\log_2 p$ bits cho thông điệp có xác suất là p.



- S là một tập rèn luyện
- $p_\oplus$ là phần các ví dụ dương trong tập S
- $p_\ominus$ là phần các ví dụ âm trong tập S
- Entropy đo độ pha trộn của tập S:

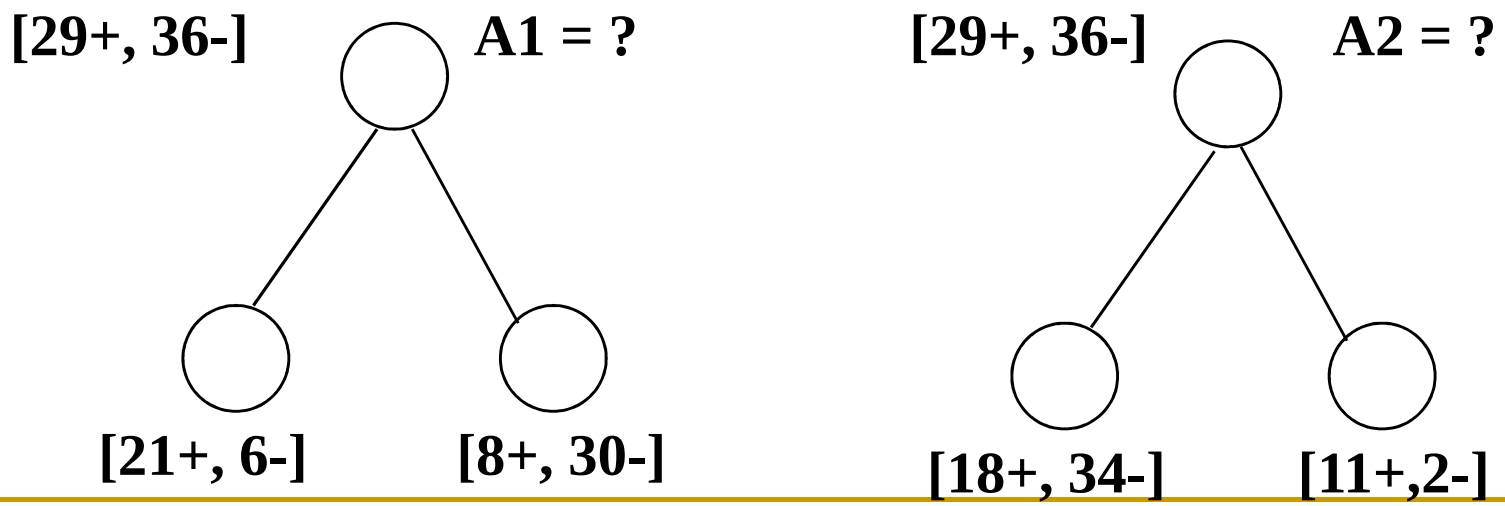
$$Entropy(S) = -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$

$$Entropy(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

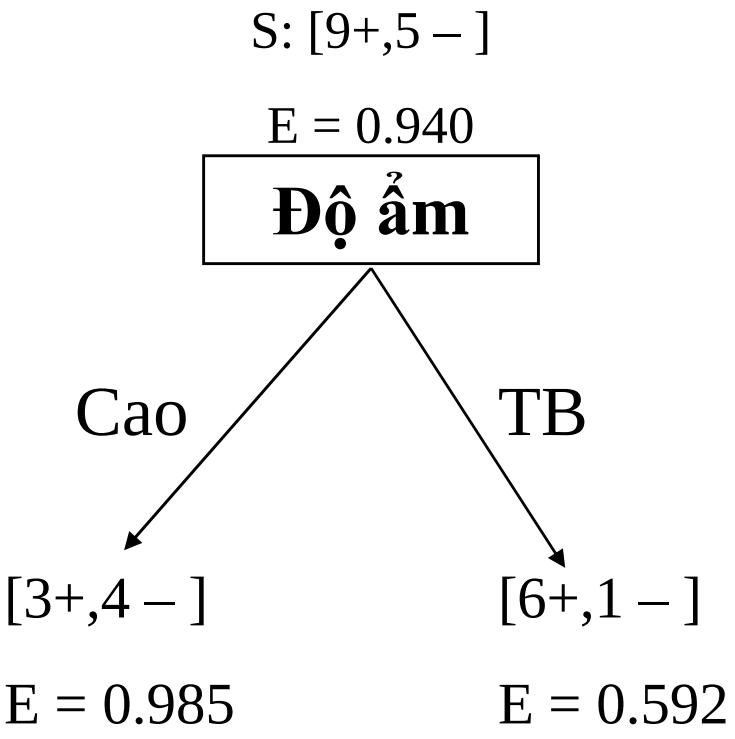
Lượng thông tin thu được
Information Gain

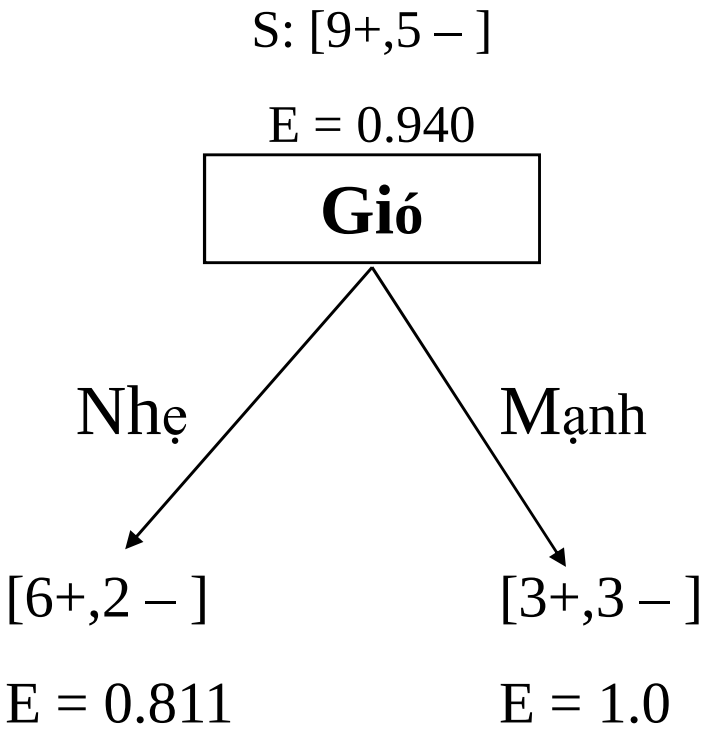
- Gain(S, A) = Lượng giảm entropy mong đợi qua việc chia các ví dụ theo thuộc tính A

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$



Chọn thuộc tính kế tiếp

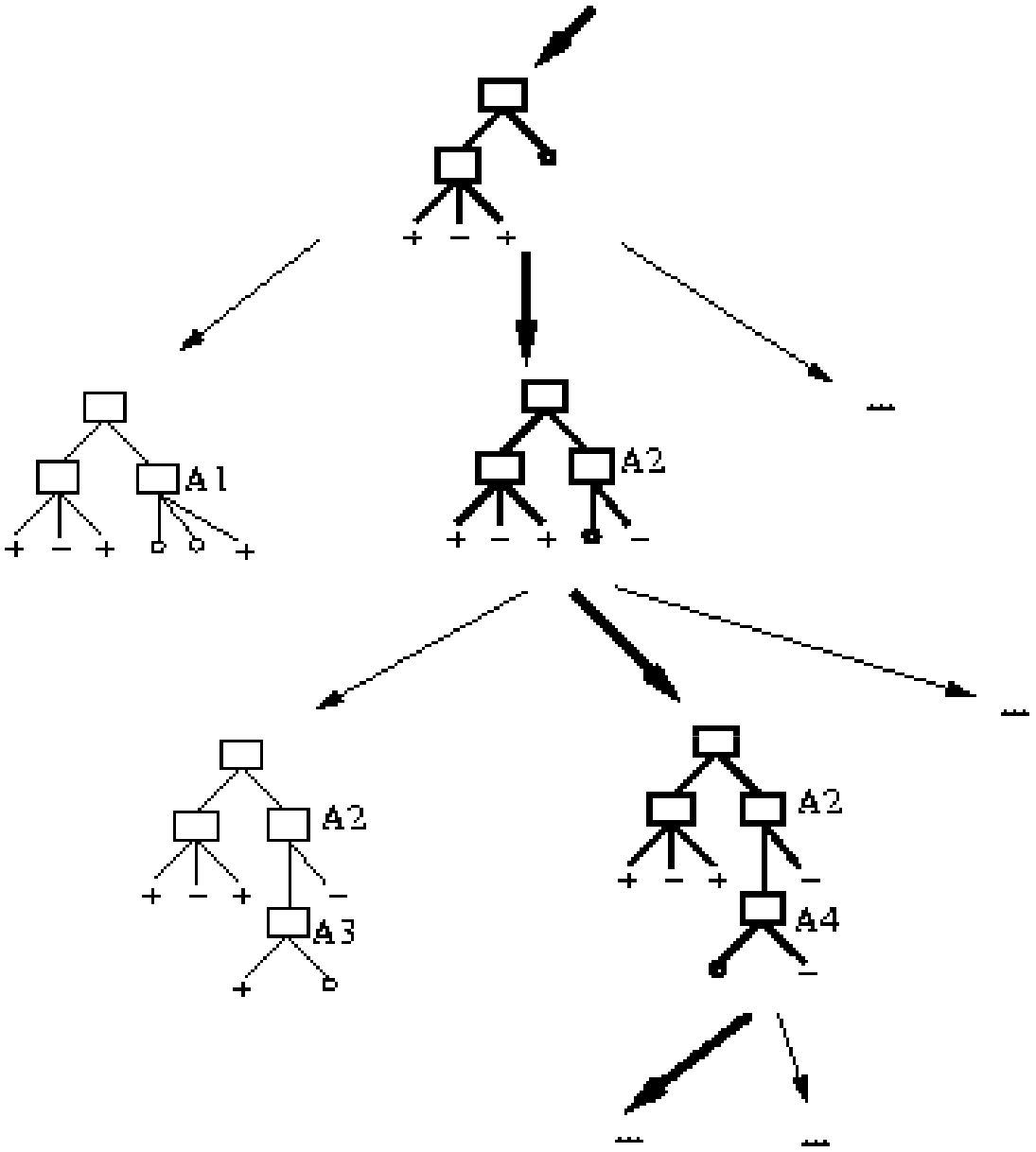




Gain(S, Độ ẩm)
= .940 – (7/14).985 – (7/14).592
= .151

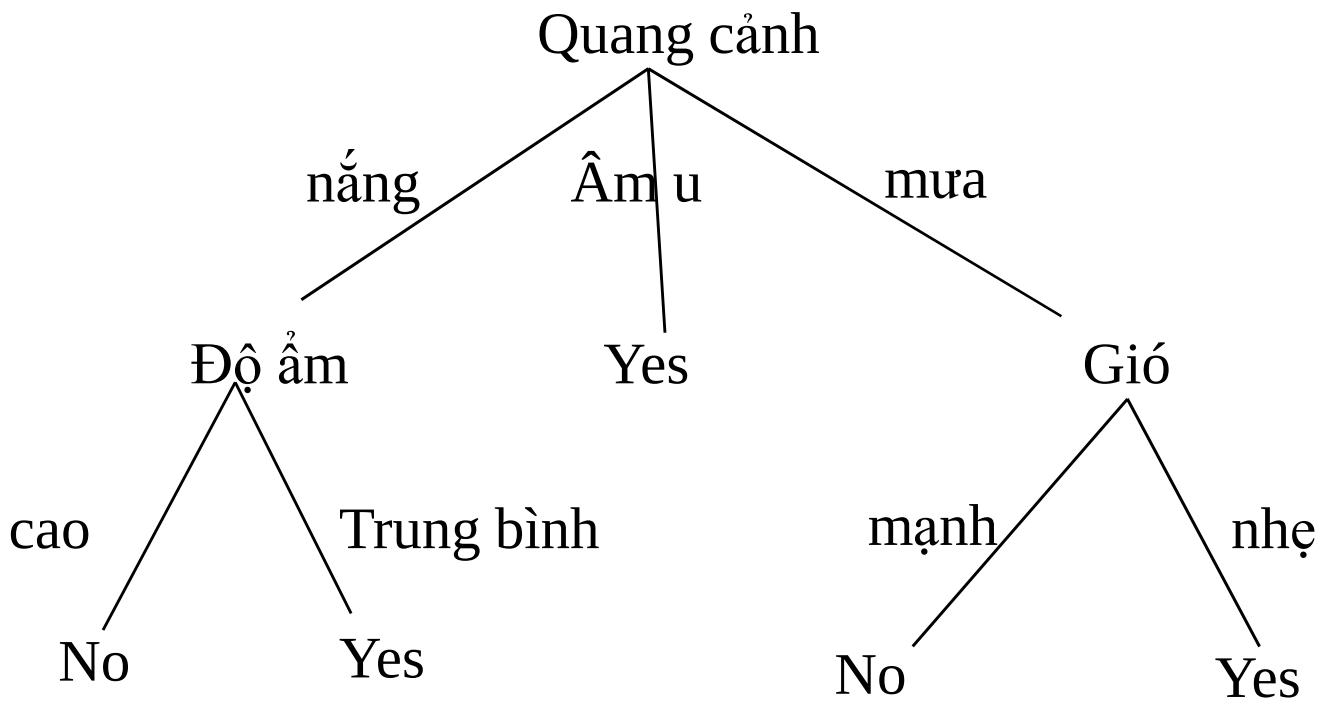
Gain(S, Gió)
= .940 – (8/14).811 – (6/14)1.0
= .048

Tìm kiếm KG giả thuyết trong ID3 (1)



- KG giả thuyết đầy đủ =>giả thuyết chắc chắn thuộc KG này
- Đầu ra là một giả thuyết (cây QĐ) =>Cây nào? Không thể chọn cây với 20 câu hỏi
- Không quay lui => cực tiểu địa phương
- Lựa chọn tìm kiếm dựa trên thống kê => chịu được dữ liệu nhiễu
- Thiên lệch quy nạp: thích cây ngắn hơn.

Chuyển cây về thành các luật



If (Quang-cảnh =nắng) $\wedge$ (Độ ẩm = Cao) Then Chơi-Tennis = No

If (Quang-cảnh =nắng) $\wedge$ (Độ ẩm = TB) Then Chơi-Tennis = Yes

If (Quang-cảnh =Âm u) Then Chơi-Tennis = Yes

…

Khi nào nên sử dụng cây QĐ

- Các ví dụ được mô tả bằng các cặp "thuộc tính – giá trị", vd: Gió - mạnh, Gió - nhẹ
- Kết quả phân loại là các giá trị rời rạc, vd: Yes, No
- Dữ liệu rèn luyện có thể chứa lỗi (bị nhiễu)
- Dữ liệu rèn luyện có thể thiếu giá trị thuộc tính

Ví dụ:

- Phân loại bệnh nhân theo các bệnh của họ
- Phân loại hỏng hóc thiết bị theo nguyên nhân
- Phân loại người vay tiền theo khả năng chi trả

Ví dụ: ước lượng độ an toàn của một tài khoản tín dụng

Table 13.1: Data from credit history of loan applications.

NO.	RISK	CREDIT HISTORY	DEBT	COLLATERAL	INCOME
1.	high	bad	high	none	$0 to $15k
2.	high	unknown	high	none	$15 to $35k
3.	moderate	unknown	low	none	$15 to $35k
4.	high	unknown	low	none	$0 to $15k
5.	low	unknown	low	none	over $35k
6.	low	unknown	low	adequate	over $35k
7.	high	bad	low	none	$0 to $15k
8.	moderate	bad	low	adequate	over $35k
9.	low	good	low	none	over $35k
10.	low	good	high	adequate	over $35k
11.	high	good	high	none	$0 to $15k
12.	moderate	good	high	none	$15 to $35k
13.	low	good	high	none	over $35k
14.	high	bad	high	none	$15 to $35k

Figure 13.13: Một cây QĐ cho bài toán đánh giá độ an toàn của tín dụng.

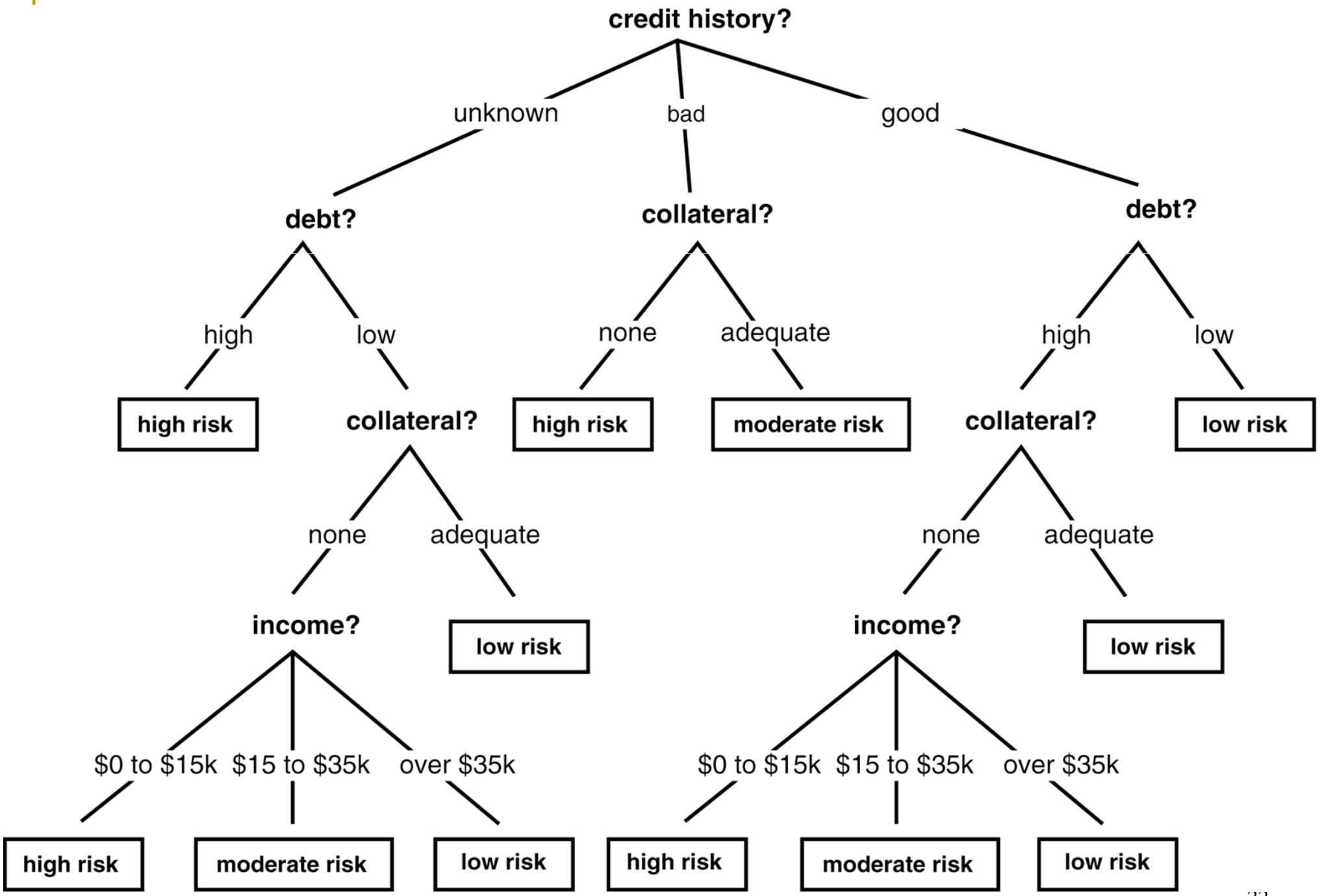


Figure 13.14: Một cây QĐ đơn giản hơn.

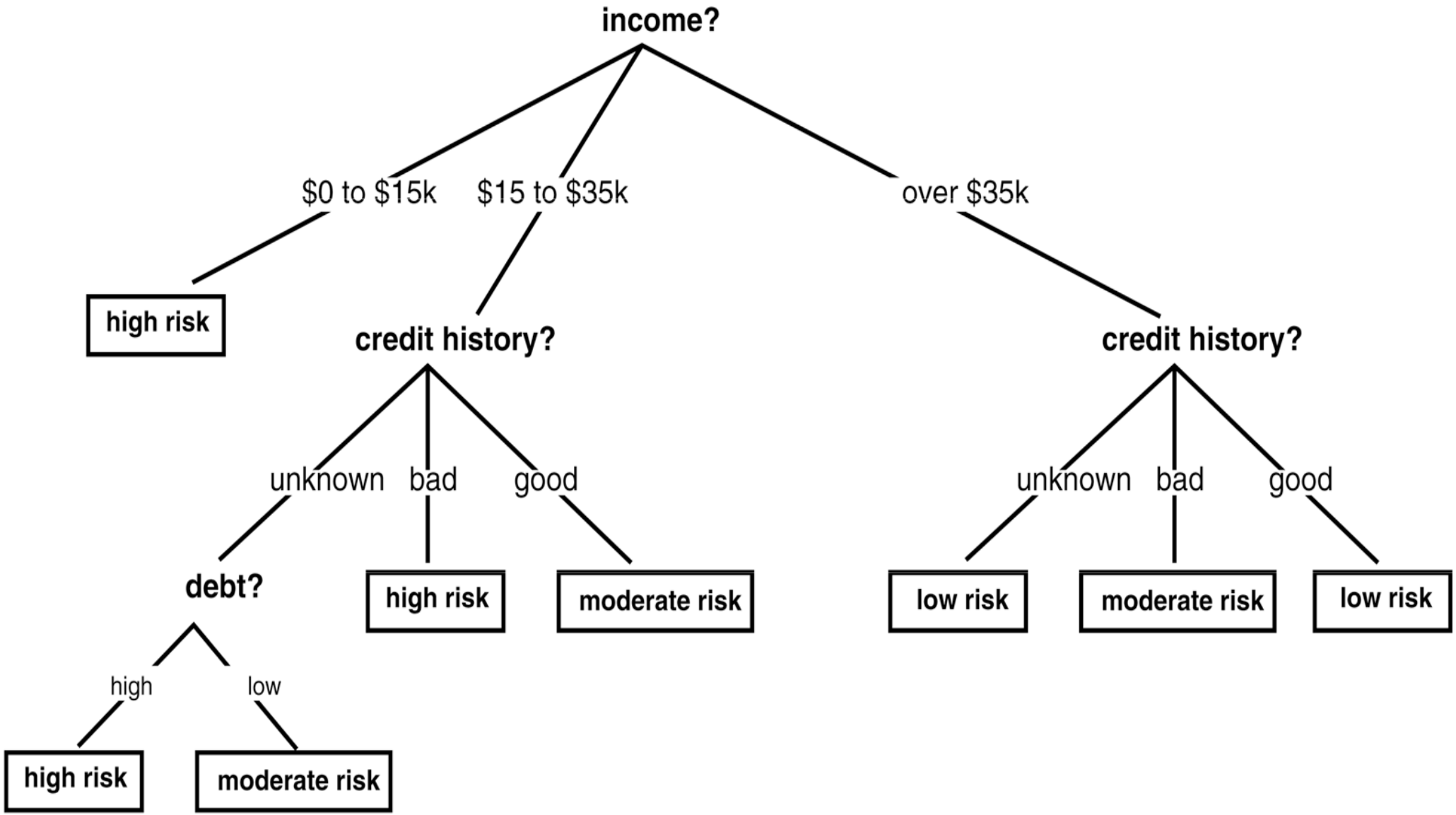


Figure 13.15: Một cây QĐ đang xây dựng.

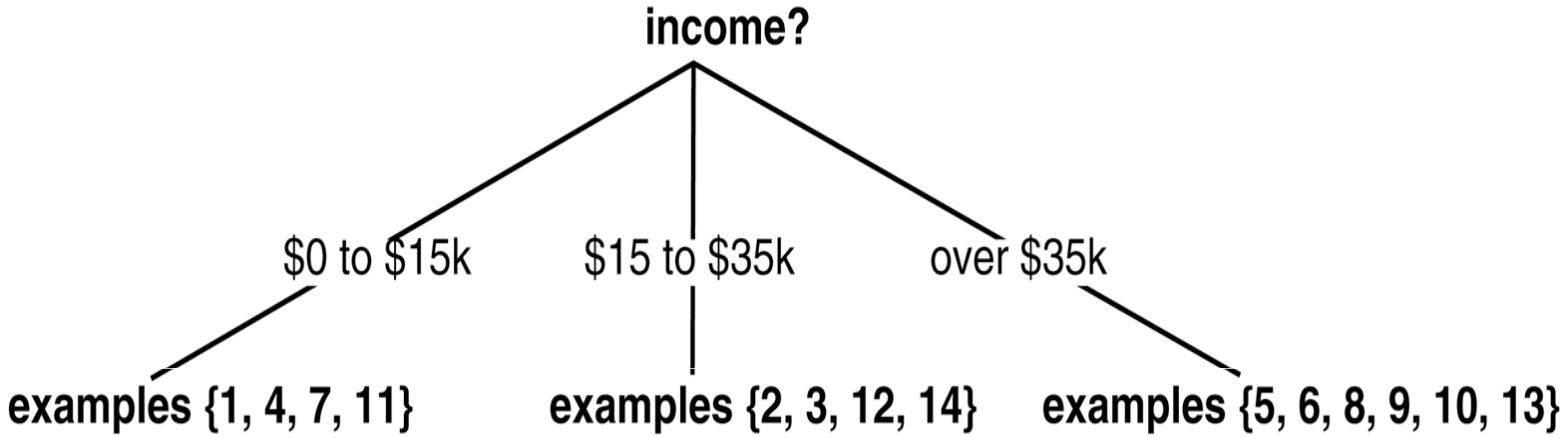
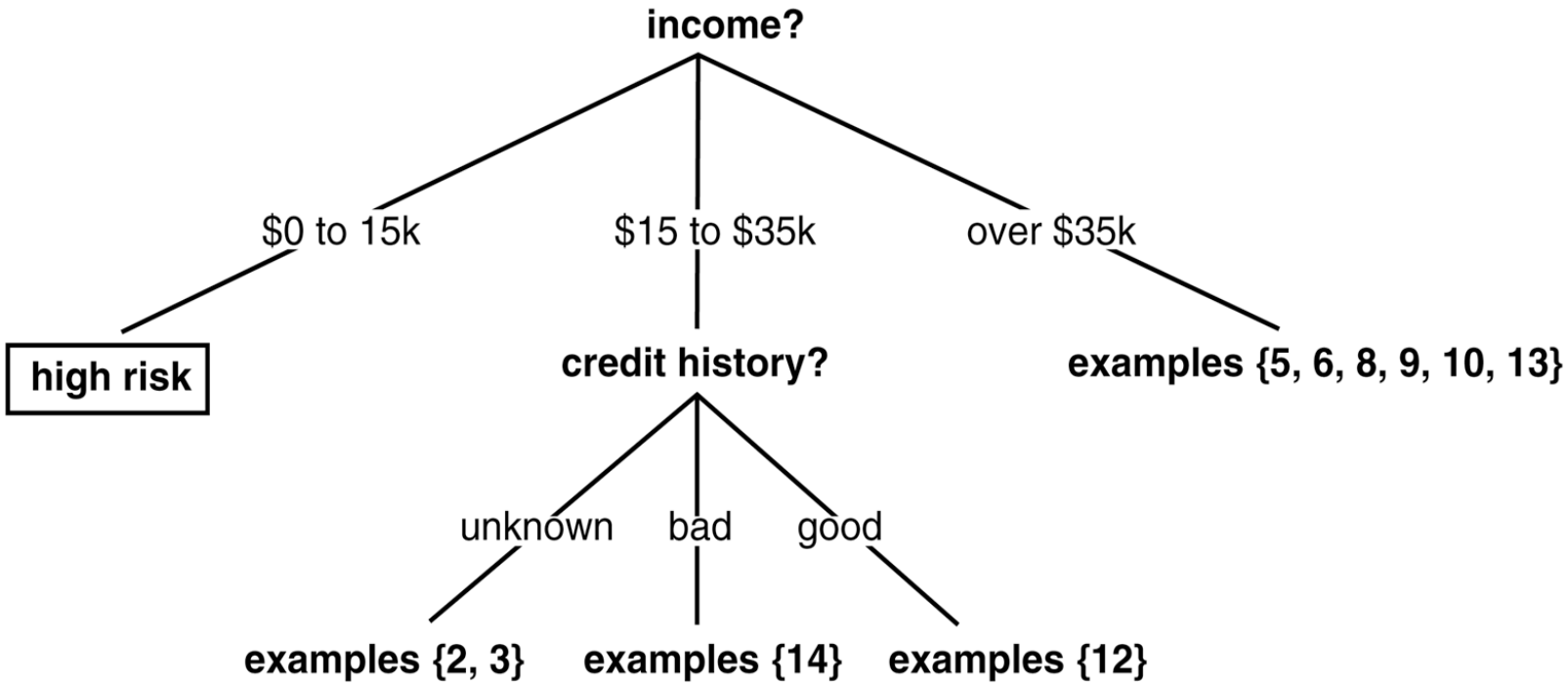


Figure 13.16: Một cây QĐ khác đang xây dựng.

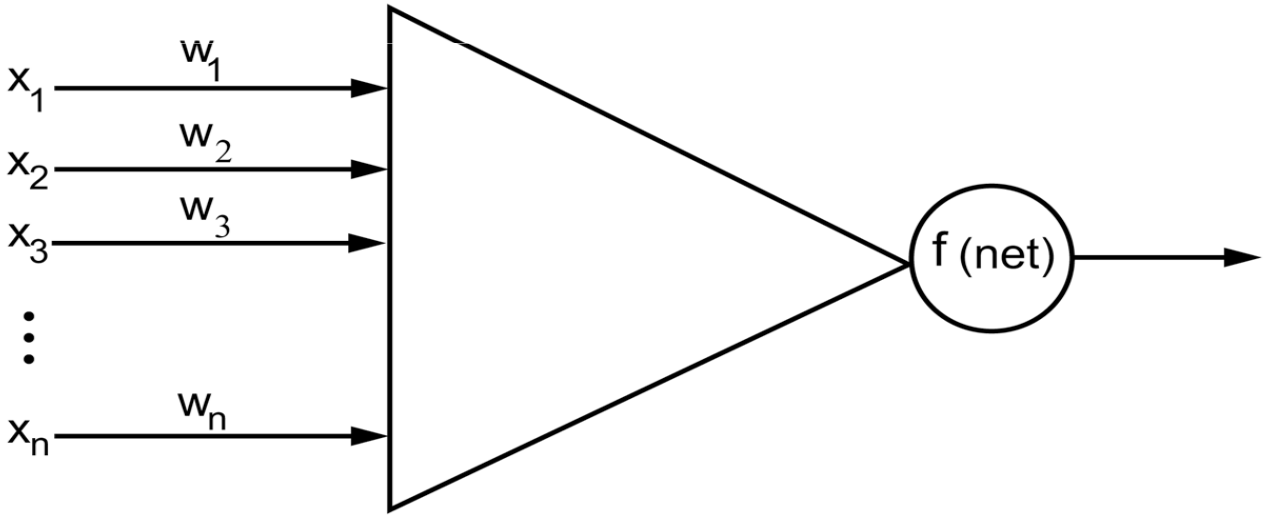


Neural Networks

- Ngược lại với các mô hình dựa trên ký hiệu: Không chú trọng việc sử dụng các ký hiệu một cách tường minh để giải quyết vấn đề.
- Ý tưởng dựa trên các hệ não: Xem trí tuệ là sự phát sinh từ các hệ thống gồm những thành phần đơn giản (neuron), tương tác với nhau thông qua một quá trình học hoặc thích nghi mà ở đó các kết nối giữa các thành phần được điều chỉnh.
- Gặt hái rất nhiều thành công trong những năm gần đây.
- Từ đồng nghĩa:
  - Tính toán neural (neural computing)
  - Các mạng neural (neural networks)
  - Các hệ kết nối (connectionist system)
  - Các hệ xử lý phân tán song song (parallel distributed processing)

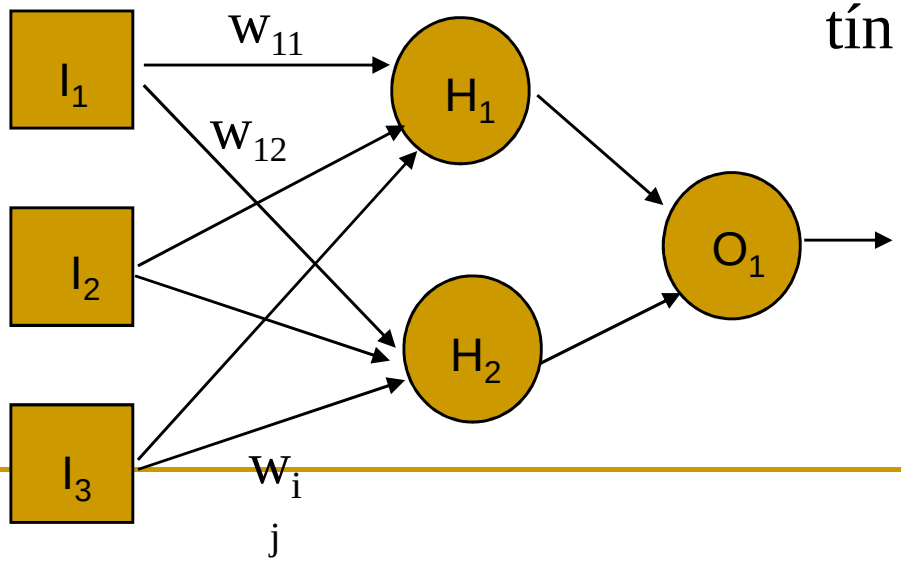
Neuron nhân tạo

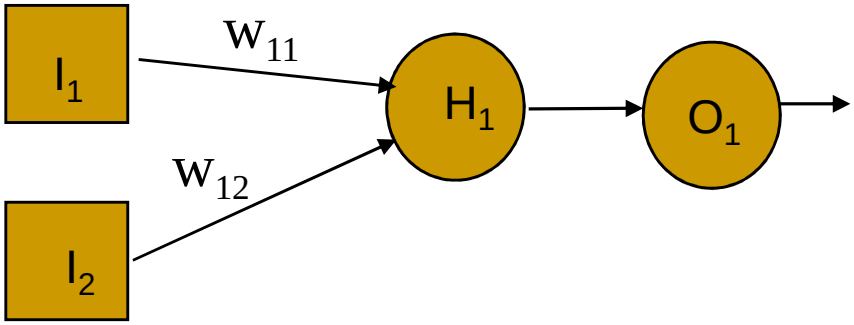
- Thành phần cơ bản của mạng neuron là một neuron nhân tạo.
- Các thành phần của một neuron nhân tạo:
  - Các tín hiệu vào x_i {0,1} {1,-1} real
  - Các trọng số w_i real
  - Một mức kích hoạt $\sum_i w_i x_i$
  - Một hàm ngưỡng f : $\sum_i w_i x_i \rightarrow$ tín hiệu ra



Neural Networks

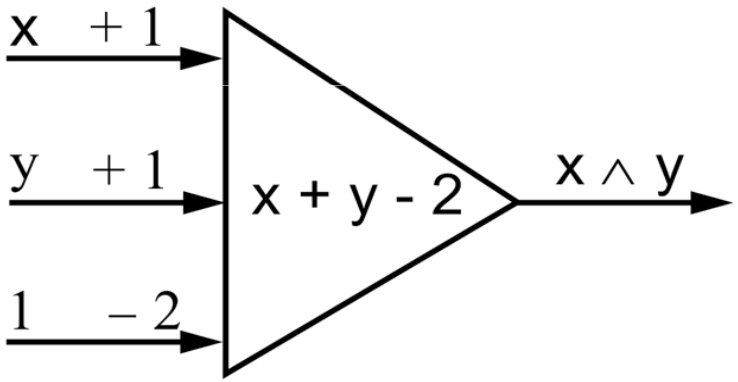
- Các thuộc tính tổng quát của một mạng là:
  - Hình thái mạng: mẫu kết nối giữa (các tầng của) các neuron.
  - Giải thuật học: cách điều chỉnh các trọng số trong quá trình xử lý tập dữ liệu rèn luyện
  - Cơ chế mã hóa: sự thông dịch của các tín hiệu vào và tín hiệu ra

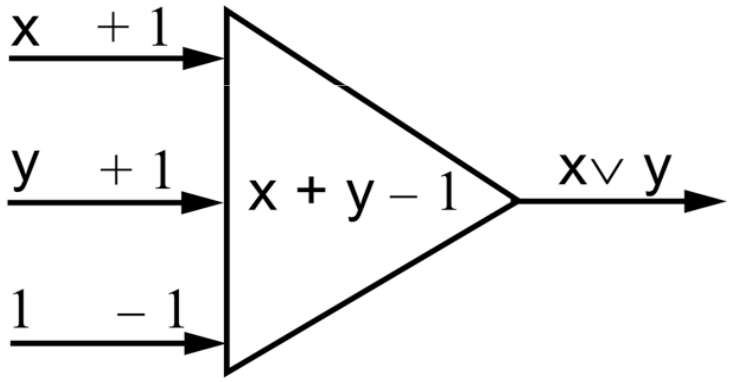




Ví dụ: Neuron McCulloch-Pitts

x	y	x + y – 2	Output
1	1	0	1
1	0	–1	–1
0	1	–1	–1
0	0	–2	–1

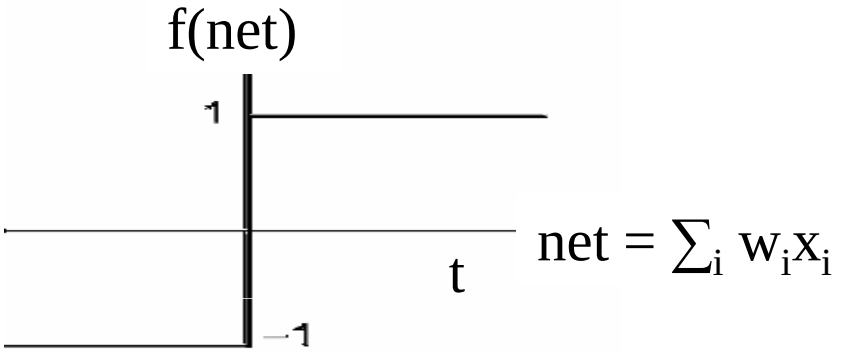




Các neurron dùng để tính các hàm logic and và or

Học Perceptron

- Mạng neuron đơn tầng
- Các giá trị vào 1 hoặc -1
- Các trọng số kiểu thực
- Mức kích hoạt $\sum_i w_i x_i$
- Hàm ngưỡng giới hạn cứng f :



$$1 \text{ if } \sum_i w_i x_i >= t$$
$$-1 \text{ if } \sum_i w_i x_i < t$$

- Điều chỉnh trọng số: $\Delta w_i = c(d - f(\sum_i w_i x_i))\, x_i$

c: hằng số chỉ tốc độ học

d: đầu ra mong muốn

Nếu kết quả thực và kết quả mong muốn giống nhau, không làm gì

Nếu kết quả thực là -1 và kết quả mong muốn là 1, tăng trọng số của đường thứ i lên $2cx_i$

Nếu kết quả thực là 1 và kết quả mong muốn là -1, giảm trọng số của đường thứ i xuống $2cx_i$

Phân loại của các hệ thống Học

- Học có sự hướng dẫn (Supervised learning)
  - Cho hệ thống một tập các ví dụ và một câu trả lời cho mỗi ví dụ.
  - Rèn luyện hệ thống cho đến khi nó có thể đưa ra câu trả lời đúng cho các ví dụ này.
- Học không có sự hướng dẫn (Unsupervised learning)
  - Cho hệ thống một tập hợp các ví dụ và cho nó tự khám phá các mẫu thích hợp trong các ví dụ.

Mạng neuron sử dụng một hình thức học có sự hướng dẫn

Sử dụng perceptron trong bài toán phân loại

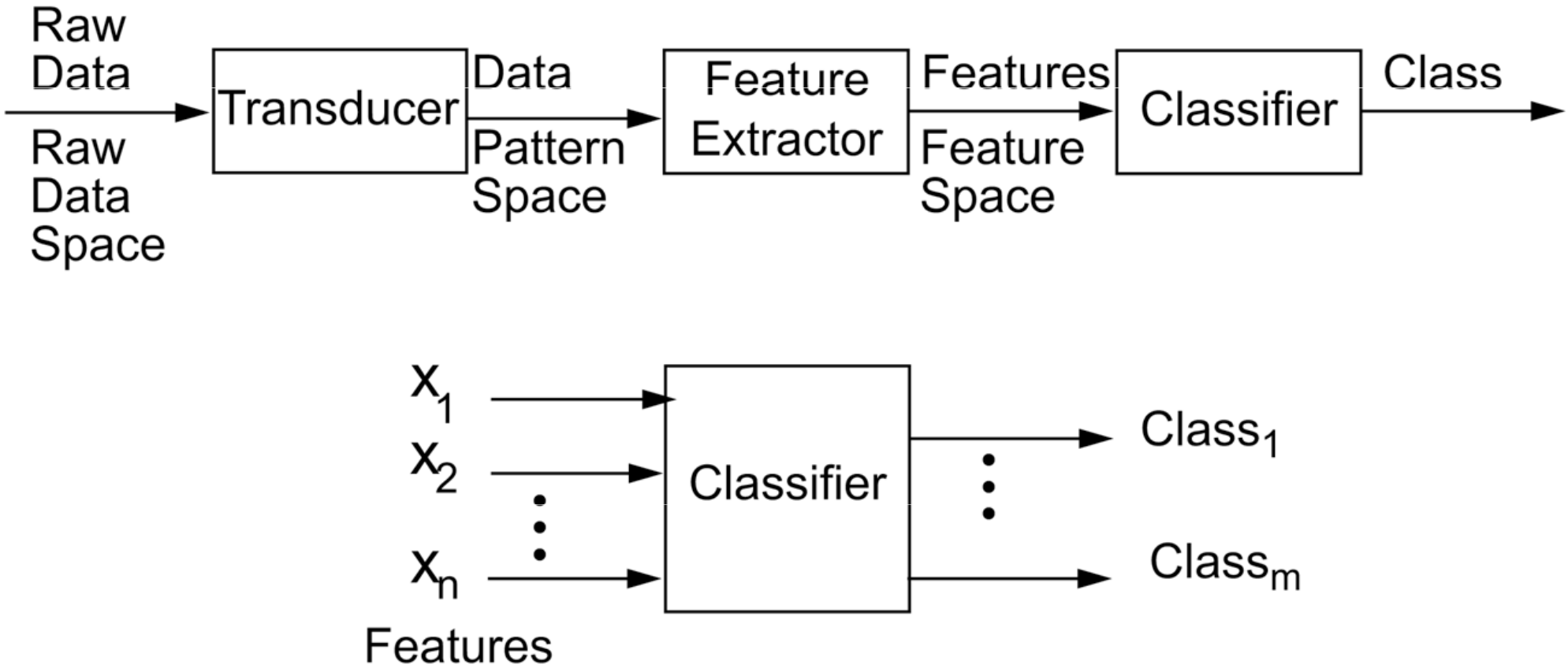
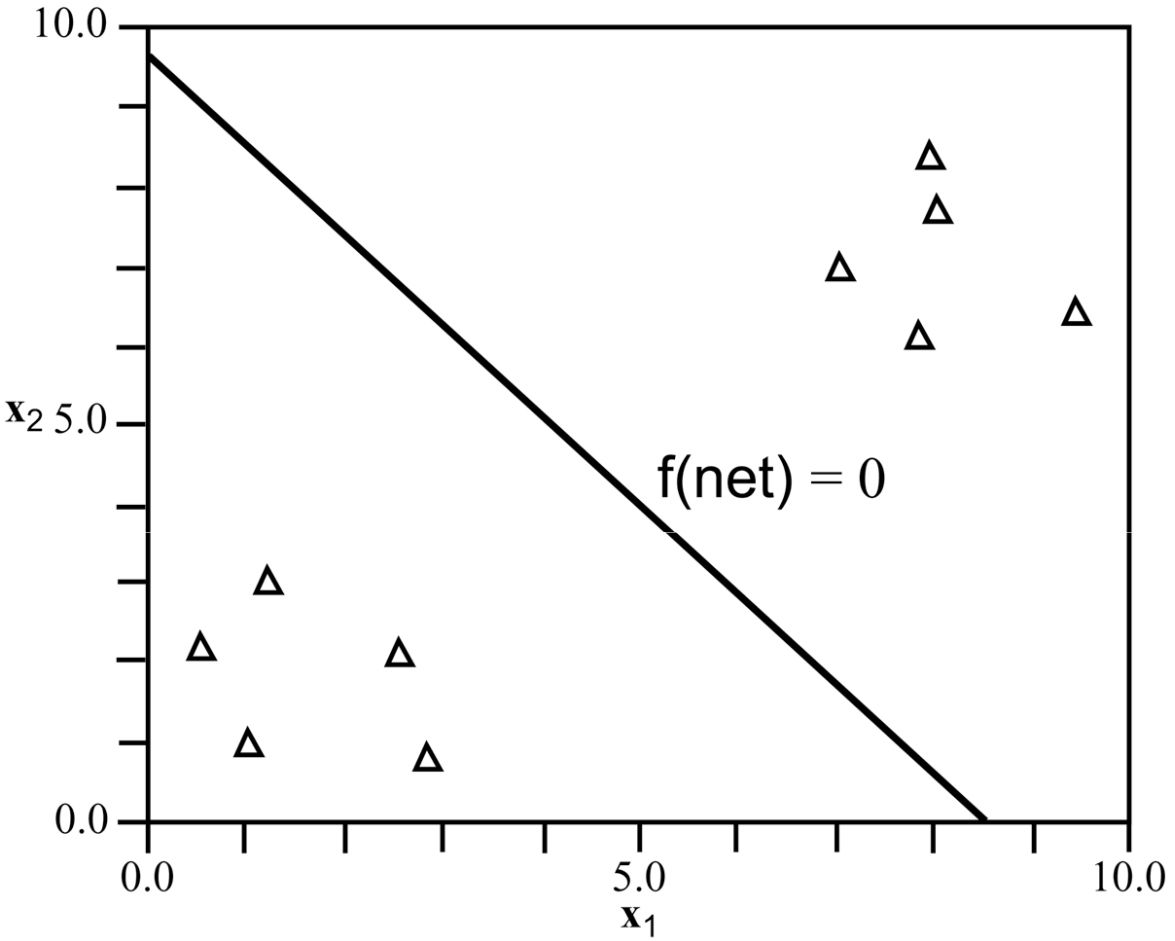


Fig 14-4: Một hệ thống phân loại đầy đủ

Ví dụ Perceptron

- Cho trước: một tập các dữ liệu vào
- Yêu cầu: rèn luyện perceptron sao cho nó phân loại các đầu vào một cách đúng đắn

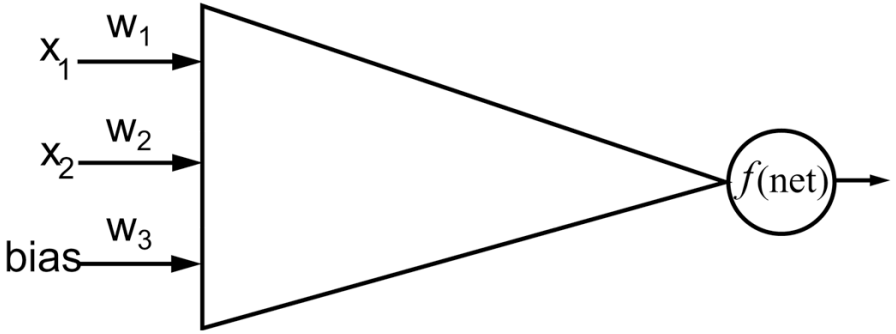
x_1	x_2	Output
1.0	1.0	1
9.4	6.4	−1
2.5	2.1	1
8.0	7.7	−1
0.5	2.2	1
7.9	8.4	−1
7.0	7.0	−1
2.8	0.8	1
1.2	3.0	1
7.8	6.1	−1



Ví dụ Perceptron: giải pháp

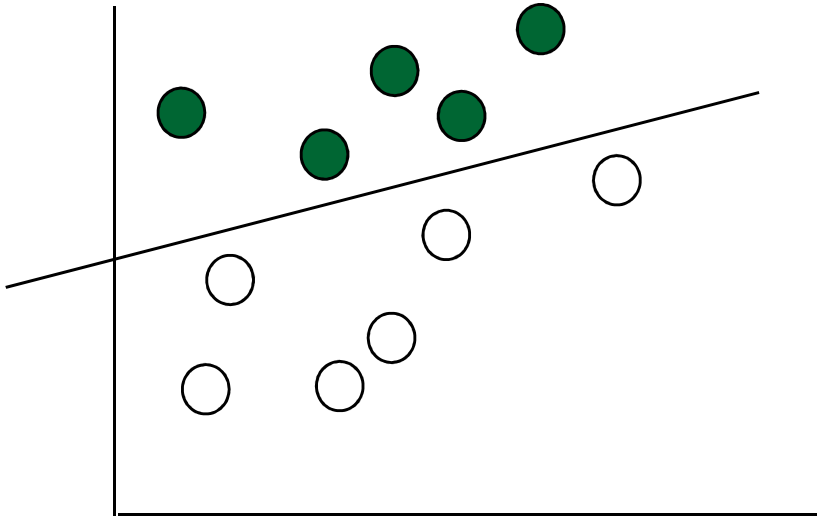
- 2 tín hiệu vào x_1 x_2
- Một tín hiệu vào thứ ba được sử dụng như một thiên vị và có giá trị cố định bằng 1, cho phép dịch chuyển đường phân cách
- Mức kích hoạt: $w_1x_1 + w_2x_2 + w_3$
- Hàm ngưỡng: hàm dấu, >0 = +1, <0 = -1

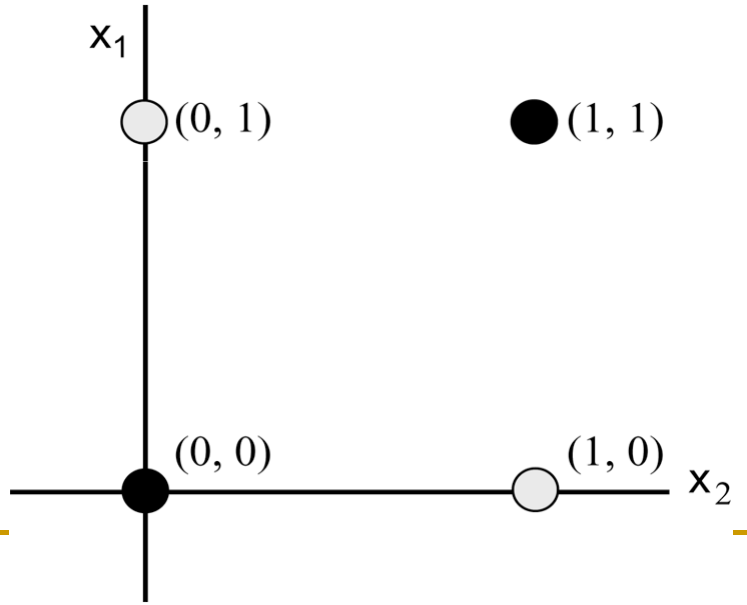
  đây là ngưỡng giới hạn cứng tuyến tính hai cực
- Các trọng số: được khởi tạo ngẫu nhiên, cập nhật 10 lần, với tốc độ học là 0.2
- Kết quả: **$-1.3x_1 + -1.1x_2 + 10.9 = 0$**



Tính tách rời tuyến tính (linearly seperatable)

- Trong một không gian n chiều, một sự phân loại mang tính tuyến tính nếu các lớp của nó có thể được tách rời bởi một mặt n-1 chiều.
- Perceptron không thể giải quyết các bài toán phân loại không tách rời tuyến tính.
  - Ví dụ: bài toán X-OR





Luật Delta

Tổng quát hóa perceptron bằng cách:

1. Thay thế hàm ngưỡng giới hạn cứng bằng các hàm kích hoạt khác có khả năng lấy vi phân

   Ví dụ: một hàm kích hoạt sigmoidal

   $f(net) = 1/(1 + e^{-\lambda * net})$ với $net = \sum_i w_i x_i$

   $f'(net) = f(net) * (1- f(net))$

2. Sử dụng luật delta để điều chỉnh trọng số trên đầu vào thứ k của nút thứ i

   $\Delta w = c(d_i - O_i) f'(net_i) x_k$

   $= c(d_i - O_i) O_i (1 - O_i) x_k$

f’: đạo hàm bậc nhất

c: tốc độ học

d_i: đầu ra mong muốn

O_i: đầu ra thật sự

Lan truyền ngược (backpropagation)

- Tại các nút của các mạng đa tầng, lỗi mà một nút phải chịu trách nhiệm cũng phải được chia phần cho các nút ở tầng ẩn và các trọng số phải được điều chỉnh một cách phù hợp.
- *Giải thuật lan truyền ngược* bắt đầu tại tầng ra và truyền các lỗi ngược về xuyên qua các tầng ẩn.

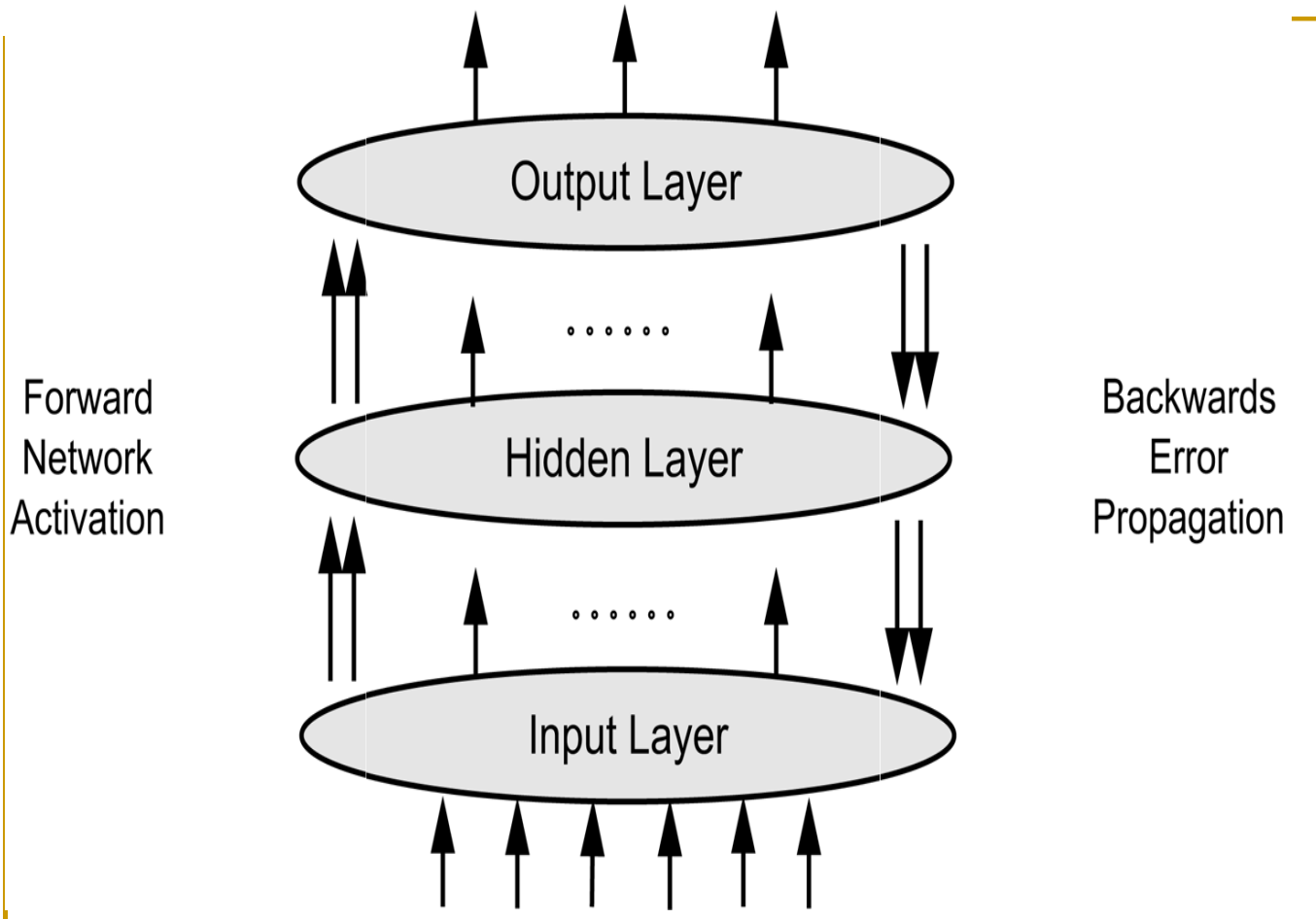
Luật delta tổng quát để điều chỉnh trọng số của đầu vào thứ k của nút thứ i:

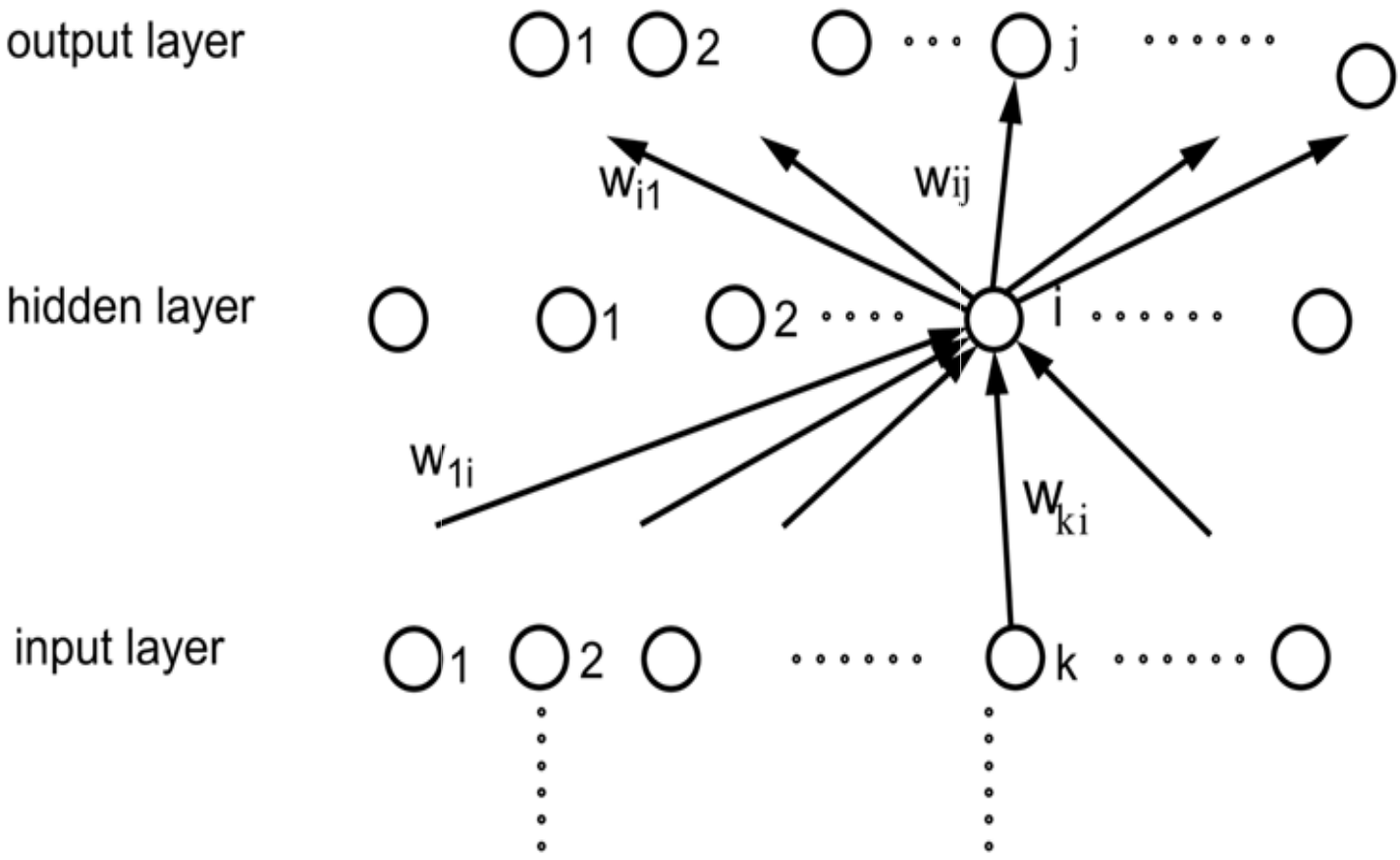
$\Delta w_k = c(d_i - O_i)\, O_i\, (1 - O_i)\, x_k$ cho nút ở tầng ra

$\Delta w_k = c \sum_j (delta_j\, w_{ij})\, O_i\, (1 - O_i)\, x_k$ cho nút ở tầng ẩn

với $delta_j = (d_j - O_j)\, O_j\, (1 - O_j)$

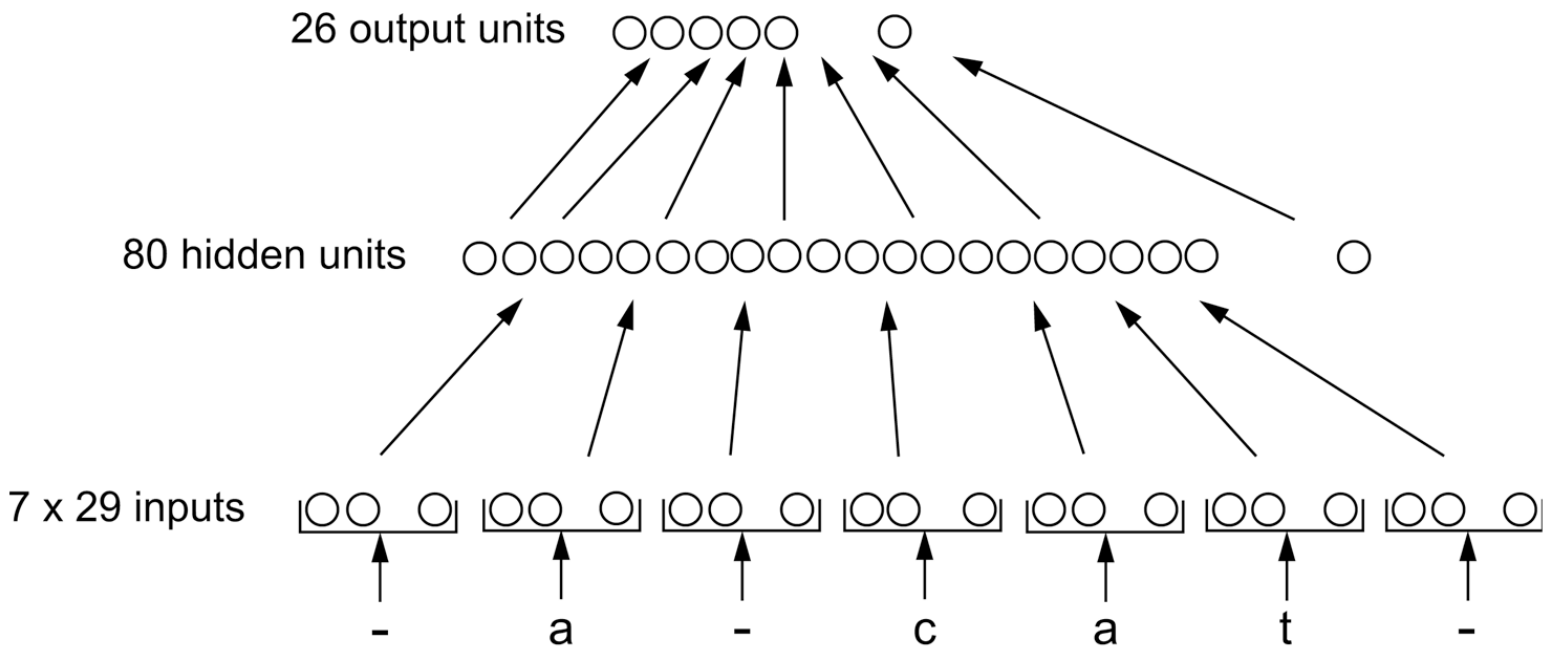
j chạy trên các nút của tầng kế tiếp mà tại đó nút i truyền các đầu ra của nó.


Output Layer
Hidden Layer
Input Layer
Forward
Network
Activation
Backwards
Error
Propagation


output layer
hidden layer
input layer
1
2
j
i
k
w_{i1}
w_{ij}
w_{1i}
w_{ki}

Ví dụ mạng Neuron: NETtalk

- Vấn đề: phát âm văn bản tiếng Anh đúng
- Đầu vào: một chuỗi
- Đầu ra: âm vị và trọng âm kèm theo cho mỗi ký tự
- Giải pháp:

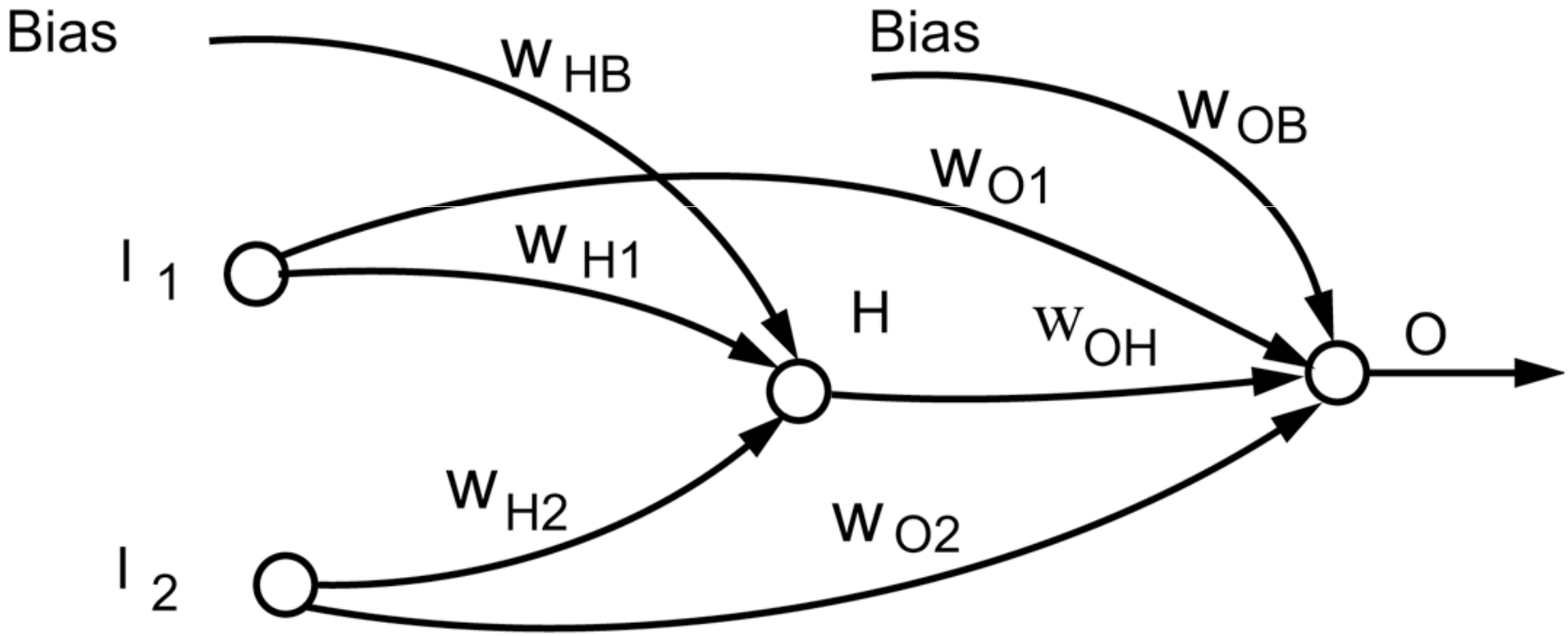


- Kết quả thụ

đúng 60% sau khi rèn luyện với 500 ví dụ (100 lượt)

càng nhiều ví dụ rèn luyện => kết quả càng tốt

Figure 10.12: A backpropagation net to solve the exclusive-or problem. The W_{ij} are the weights and H is the hidden node.



Sử dụng 4 mẫu ví dụ để luyện tập:

(0,0) -> 0; (1,0) ->1; (0,1) -> 1; (1,1) ->0

Sau 1400 lượt: $W_{H1} = -7.0$ $W_{HB} = 2.6$ $W_{O1} = -5.0$

$W_{H2} = -7.0$ $W_{OB} = 7.0$ $W_{O2} = -4.0$

$W_{HO} = -11.0$

Các vấn đề liên quan khi sử dụng Neural Networks

- Các mạng đa tầng là đầy đủ về mặt tính toán, tuy nhiên:
  - Làm sao để chọn số nút ẩn và số tầng ẩn
  - Khi nào sử dụng các nút thiên lệch
  - Cách chọn một tập rèn luyện
  - Điều chỉnh các trọng số hay tốc độ học nên n.t.n?
  - …

Giải thuật Genetic

- Nắm bắt ý tưởng từ thuyết tiến hóa
- Học được xem như là sự cạnh tranh giữa các quần thể các giải pháp khả dĩ đang tiến hóa của bài toán
- Thành phần:
  - Quần thể các giải pháp khả dĩ
  - Hàm đánh giá
  - Các phép toán tạo con mới:
    - giao nhau (crossover)
    - Đột biến (mutation)
- Giải thuật:
  - Điều kiện kết thúc: #vònglặp, Trung bình 'độ tốt' của quần thể