

Chương 6. Học máy

Lê Thanh Hương
Bộ môn HTTT, Khoa CNTT
Đại học Bách khoa Hà Nội

1

6.1. Học

*“Học đề cập đến các **thay đổi** của hệ thống theo hướng **thích nghi**: chúng cho phép hệ thống thực hiện các công việc trong cùng một môi trường hiệu quả hơn từ lần thực hiện thứ 2”*

2

Các phương pháp học

- **Học có giám sát**: biết trước câu trả lời đúng
- **Học không giám sát**: không biết trước câu trả lời đúng
- **Học tăng cường**: đôi khi có thưởng/phạt cho các hành động

3

Những gì cần học?

- Mẹo trong tìm kiếm
- Hàm đánh giá trò chơi
- Tri thức khai báo (các mệnh đề logic)
- Các bộ phân loại
 - Cấu trúc phân loại
 - Ngữ pháp

4

Học có giám sát: qui nạp

- Trường hợp tổng quát:
 - Cho tập các cặp $(x, f(x))$, tìm hàm f .
- Phân loại:
 - Cho tập các cặp (x, y) với y là 1 nhãn, tìm hàm cho phép gán x với giá trị đúng của nó.
- Phân loại đơn giản:
 - Cho tập các cặp (x, y) với x là 1 đối tượng và $y = +$ nếu x thuộc đúng lớp và $-$ nếu ngược lại. Tìm hàm cho phép gán nhãn chính xác.

5

Coi học như việc tìm kiếm

- Đoán hàm phù hợp với các đầu vào = xác định 1 giả thiết.
- Không gian giả thiết = tập tất cả các giả thiết có thể.
- Học là việc tìm kiếm 1 giả thiết phù hợp trong không gian giả thiết

6

Các phương pháp phân loại

- Học qui nạp
- Láng giềng gần
- Xác suất
- Cây quyết định
- Mạng nơron
- Giải thuật di truyền
- ...

7

6.2. Học cây quyết định

Bài toán: quyết định có đợi 1 bàn ở quán ăn không, dựa trên các thông tin sau:

1. **Lựa chọn khác:** có quán ăn nào khác gần đó không?
2. **Quán rượu:** có khu vực phục vụ đồ uống gần đó không?
3. **Fri/Sat:** hôm nay là thứ sáu hay thứ bảy?
4. **Đói:** chúng ta đã đói chưa?
5. **Khách hàng:** số khách trong quán (không có, vài người, đầy)
6. **Giá cả:** khoảng giá (\$,\$\$,\$\$\$)
7. **Mưa:** ngoài trời có mưa không?
8. **Đặt chỗ:** chúng ta đã đặt trước chưa?
9. **Loại:** loại quán ăn (Pháp, Ý, Thái, quán ăn nhanh)
10. **Thời gian đợi:** 0-10, 10-30, 30-60, >60

8

Phép biểu diễn dựa trên thuộc tính

- Các mẫu được miêu tả dưới dạng các giá trị thuộc tính (logic, rời rạc, liên tục)
- Ví dụ, tình huống khi đợi 1 bàn ăn

Example	Attributes										Target
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	
X ₁	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
X ₂	T	F	F	T	Full	\$	F	F	Thai	30-60	F
X ₃	F	T	F	F	Some	\$	F	F	Burger	0-10	T
X ₄	T	F	T	T	Full	\$	F	F	Thai	10-30	T
X ₅	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X ₆	F	T	F	T	Some	\$\$	T	T	Italian	0-10	T
X ₇	F	T	F	F	None	\$	T	F	Burger	0-10	F
X ₈	F	F	F	T	Some	\$\$	T	T	Thai	0-10	T
X ₉	F	T	T	F	Full	\$	T	F	Burger	>60	F
X ₁₀	T	T	T	T	Full	\$\$\$	F	T	Italian	10-30	F
X ₁₁	F	F	F	F	None	\$	F	F	Thai	0-10	F
X ₁₂	T	T	T	T	Full	\$	F	F	Burger	30-60	T

- Các loại (lớp) của mẫu là khẳng định (T) hoặc phủ định (F)

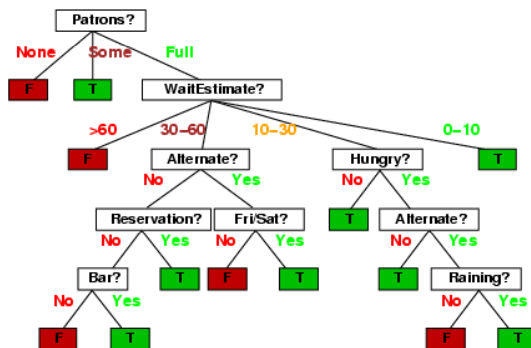
Attributes										Target
Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Wait
T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
T	F	F	T	Full	\$	F	F	Thai	30-60	F
F	T	F	F	Some	\$	F	F	Burger	0-10	T
T	F	T	T	Full	\$	F	F	Thai	10-30	T
T	F	T	F	Full	\$\$\$	F	T	French	>60	F
F	T	F	T	Some	\$\$	T	T	Italian	0-10	T
F	T	F	F	None	\$	T	F	Burger	0-10	F
F	F	F	T	Some	\$\$	T	T	Thai	0-10	T
F	T	T	F	Full	\$	T	F	Burger	>60	F
T	T	T	T	Full	\$\$\$	F	T	Italian	10-30	F
F	F	F	F	None	\$	F	F	Thai	0-10	F
T	T	T	T	Full	\$	F	F	Burger	30-60	T

Patrons, WaitEstimates, Alternative, Hungry, Rain

10

Cây quyết định

... là cách biểu diễn các giả thiết.



11

Không gian giả thiết

Khi có n thuộc tính Boolean, số lượng các cây quyết định là?

= số các hàm Boolean

= số các giá trị khác nhau trong bảng ví dụ mẫu với 2^n hàng

= 2^{2^n}

Ví dụ, với 6 thuộc tính Boolean, có

18,446,744,073,709,551,616 cây

12

Thuật toán ID3

Mục đích: tìm cây thoả mãn tập mẫu

Ý tưởng: (lập) chọn thuộc tính quan trọng nhất làm gốc của cây/cây con

ID3(Examples, Target_attribute, Attributes)

/* Examples: các mẫu luyện

Target_attribute: thuộc tính cần đoán giá trị

Attributes: các thuộc tính có thể được kiểm tra qua phép học cây quyết định. */

- Tạo 1 nút gốc Root cho cây
- If \forall Examples +, trả về cây chỉ có 1 nút Root, với nhãn +
- If \forall Examples -, trả về cây chỉ có 1 nút Root, với nhãn -
- If Attributes rỗng, trả về cây chỉ có 1 nút Root, với nhãn = giá trị thường xuất hiện nhất của Target_attribute trong Examples

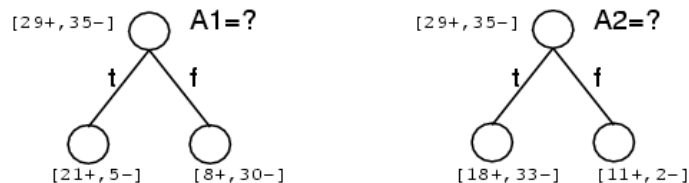
13

Thuật toán ID3

- Otherwise Begin:
 - $A \leftarrow$ thuộc tính trong Attributes cho phép phân loại tốt nhất Examples
 - Thuộc tính quyết định của nút gốc $\leftarrow A$
 - Với các giá trị v_i có thể có của A,
 - Thêm 1 nhánh mới dưới gốc, ứng với phép kiểm tra $A = v_i$
 - Đặt $Examples_{v_i} =$ tập con của Examples với giá trị thuộc tính $A = v_i$
 - If $Examples_{v_i}$ rỗng
 - Then, dưới nhánh mới này, thêm 1 lá với nhãn = giá trị thường xuất hiện nhất của Target_attribute trong Examples
 - Else, dưới nhánh mới này thêm cây con ID3($Examples_{v_i}$, Target_attribute, Attributes - {A}))
- End
- Return Root

14

Thuộc tính nào tốt nhất?

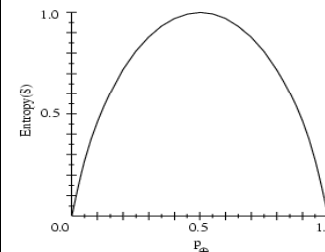


Sử dụng lượng thông tin đạt được Information Gain

\Rightarrow xác định thông qua độ đo Entropy

15

Entropy của một tập mẫu



- S là một tập mẫu của tập luyện
- p_+ là tỷ lệ các mẫu dương trong S
- p_- là tỷ lệ các mẫu âm trong S

• Entropy đo độ nhiễu của S = số các bit cần thiết để mã hoá lớp + hoặc - của các thành viên ngẫu nhiên của S

• $Entropy(S) = -p_+ \cdot \log_2 p_+ - p_- \cdot \log_2 p_-$

16

Entropy

Entropy $H(X)$ của biến ngẫu nhiên X :

$$H(X) = - \sum_{i=1}^n P(X = i) \log_2 P(X = i)$$

Ví dụ, với S gồm 9 mẫu dương và 5 mẫu âm, kí hiệu $S([9+,5-])$.

Entropy($[9+,5-]$)

$$= - (9/14) \log_2(9/14) - (5/14) \log_2(5/14)$$

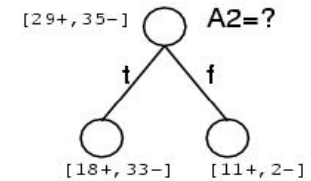
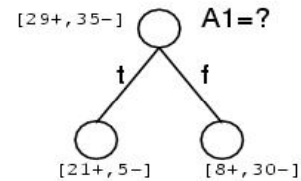
$$= 0.940$$

17

Information Gain

Gain(S, A) = độ giảm entropy do việc phân loại trong A

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



18

Ví dụ: tập luyện

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$S = [9+, 5-]$

Humidity
= $\{High, Normal\}$:

$S_{high} = [3+, 4-]$:

$S_{normal} = [6+, 1-]$

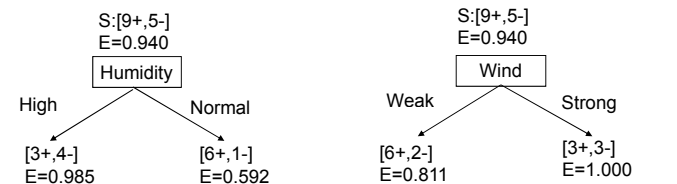
Wind = $\{Weak, Strong\}$:

$S_{weak} = [6+, 2-]$:

$S_{strong} = [3+, 3-]$

19

Thuộc tính nào phân loại tốt nhất?



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

$$= Entropy(S) - (8/14) Entropy(S_{Weak}) - (6/14) Entropy(S_{Strong})$$

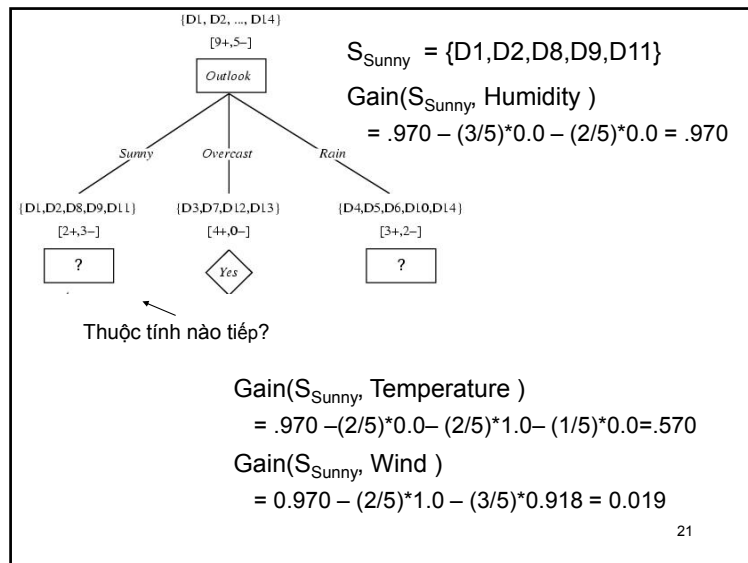
$$= 0.940 - (8/14) * 0.811 - (6/14) * 1.00 = 0.048$$

$$Gain(S, Humidity) = 0.940 - (7/14) * 0.985 - (7/14) * 0.592 = 0.151$$

$$Gain(S, Outlook) = 0.246; Gain(S, Humidity) = 0.151$$

$$Gain(S, Wind) = 0.048; Gain(S, Temperature) = 0.029$$

20



Cây quyết định sử dụng khi nào?

Các bài toán với các đặc tính sau thích hợp với học cây quyết định:

- Các mẫu mô tả được bởi các cặp thuộc tính-giá trị
- Hàm đích có giá trị rời rạc
- Cần có các giả thiết rời rạc
- Các dữ liệu luyện có thể có nhiễu
- Dữ liệu luyện có thể thiếu giá trị thuộc tính

Ví dụ:

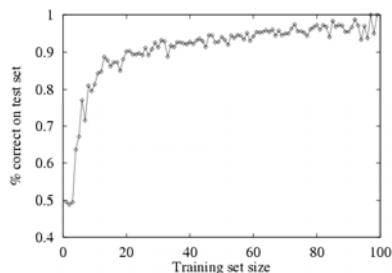
- Chẩn đoán y tế
- Phân tích các nguy cơ về tín dụng
- Mô hình hoá việc lập lịch

22

Đo độ chính xác

- Làm sao để biết $h \approx f$?
 - Sử dụng lý thuyết tính toán
1. Thử giả thiết h trên 1 tập các ví dụ mới (tập thử) (sử dụng cùng 1 mức độ phân bố các mẫu như tập luyện)

Learning curve = % chính xác trên tập thử, sử dụng hàm xây dựng trên tập luyện



23

6.3. Học dựa trên mẫu

Ý tưởng: lưu tất cả các mẫu luyện $\langle x_i, f(x_i) \rangle$

Láng giềng gần nhất:

- Cho mẫu hỏi x_q , trước tiên định vị mẫu luyện gần nhất x_n , sau đó đánh giá $\hat{f}(x_q) \leftarrow f(x_n)$

K láng giềng gần nhất:

- Cho x_q , quyết định dựa trên k láng giềng gần nhất (nếu hàm đích có giá trị rời rạc)
- Lấy trung bình giá trị f của k láng giềng gần nhất (nếu là giá trị thực)

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^k f(x_i)}{k}$$

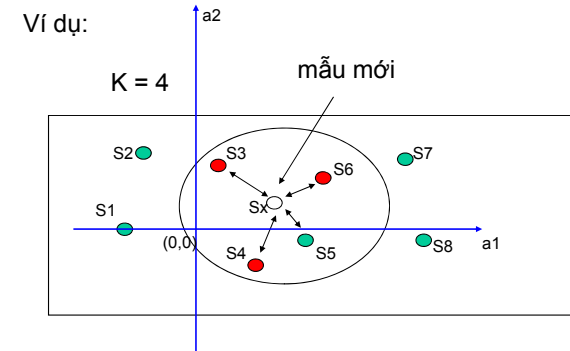
24

Phương pháp láng giềng gần

- Tổ chức dữ liệu dưới dạng bảng.
- Xây dựng ma trận cho phép tính khoảng cách giữa các cặp đối tượng.
- Khi có 1 đối tượng mới chưa có kết luận, lấy kết luận của láng giềng gần nhất gán cho nó.

25

kNN



26

	a1	a2	T
S1	-2	0	B
S2	-1.5	2	B
S3	0.4	1.8	R
S4	1.5	-1	R
S5	3	-0.2	B
S6	3.2	1.5	R
S7	4.5	2	B
S8	4.7	-0.2	B
Sx	2	0.5	X

27

kNN

- Để định nghĩa độ tương tự giữa 2 TH, ta dùng ma trận.
- Giả sử các mẫu là các điểm trong không gian n chiều R^n và dùng khoảng cách Euclidean
- Cho X_i và X_j là 2 ví dụ. Khoảng cách của chúng là

$$d^2(X_i, X_j) = \sum_k [x_{ik} - x_{jk}]^2$$

trong đó x_{ik} là giá trị của thuộc tính k trên ví dụ X_i .

28

Thuật toán kNN cho các giá trị rời rạc

Thuật toán (tham số k)

1. Với mỗi mẫu luyện (X, f(X)), bổ sung mẫu vào tập luyện
2. Khi có mẫu mới X_q, gán lớp:
f(X_q) = lớp của đa số các thành viên trong k láng giềng gần nhất của X_q

$$\hat{f}(X_q) = \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^k \delta(v, f(X_i))$$

với $\delta(a,b) = 1$ nếu $a = b$ và 0 nếu ngược lại

29

kNN cho các hàm giá trị thực

Thuật toán (tham số k)

1. Với mỗi mẫu luyện (X, f(X)), bổ sung mẫu vào tập luyện
2. Khi có mẫu mới X_q, gán lớp:
f(X_q) = giá trị trung bình của k láng giềng gần nhất của X_q

$$\hat{f}(X_q) = \sum \frac{f(X_i)}{k}$$

30

Đánh trọng số khoảng cách kNN

Có thể muốn các láng giềng gần nhất có trọng số cao

$$\hat{f}(X_q) = \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^k \omega_i \delta(v, f(X_i)) \quad \hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^k \omega_i f(x_i)}{\sum_{i=1}^k \omega_i}$$

trong đó

$$\omega_i = \frac{1}{d(x_q, x_i)^2}$$

và $d(x_q, x_i)$ là khoảng cách giữa x_q và x_i

Chú ý: từ đây có thể thấy lý do dùng tất cả các mẫu luyện thay vì chỉ có k

→ phương pháp Shepard

31

Khi nào nên dùng láng giềng gần

- Các mẫu tương ứng với các điểm trong \mathbb{R}^n
- Mỗi mẫu có dưới 20 thuộc tính
- Nhiều mẫu luyện

Ưu điểm:

- Luyện rất nhanh
- Học các hàm đích phức tạp
- Không mất thông tin

Nhược điểm:

- Chậm khi truy vấn
- Dễ bị ảnh hưởng bởi các thuộc tính không liên quan

32

Ảnh hưởng của số chiều

Giả thiết các mẫu được mô tả bởi 20 thuộc tính, nhưng chỉ có 2 thuộc tính liên quan đến hàm đích

Ảnh hưởng của số chiều: phương pháp kNN thường bị mất phương hướng khi X nhiều chiều

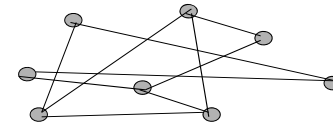
Một số giải pháp:

- Giảm chiều thứ j bởi trọng số z_j , trong đó z_1, \dots, z_n được chọn để tối thiểu hoá lỗi dự tính
- Sử dụng cross-validation để tự động chọn các trọng số z_1, \dots, z_n

33

6.4. Mạng nơ-ron nhân tạo

nghiên cứu và mô phỏng các tiến trình xử lý song song và phân tán khổng lồ diễn ra trong bộ não con người



Các vấn đề:

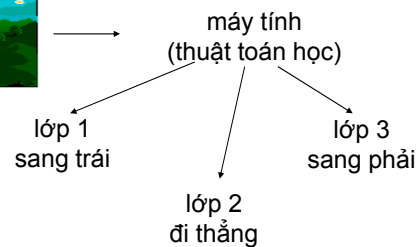
- Tốc độ bộ não nhận dạng hình ảnh
- Rất nhiều nơ-ron trong một bộ não
- Tốc độ một nơ-ron truyền dữ liệu

34

Ví dụ

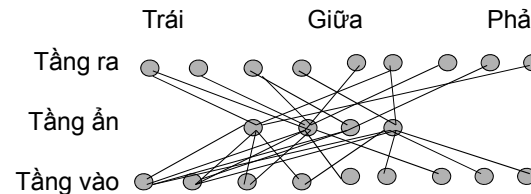
Lái xe

Luyện bộ phận điều khiển xe lái xe chính xác trên nhiều địa hình khác nhau



35

Biểu diễn mạng nơ-ron



36

Định nghĩa

- Là một hệ thống gồm rất nhiều phần tử xử lý đơn giản hoạt động song song.
- Tính năng: phụ thuộc vào
 - cấu trúc hệ thống
 - mức độ liên kết giữa các phần tử
 - quá trình xử lý bên trong các phần tử
- Có thể học từ số liệu và tổng quát hoá từ các số liệu đó.

37

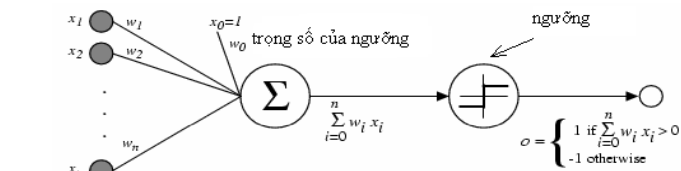
Khi nào sử dụng mạng nơron?

Mạng nơron thích hợp với những bài toán có đặc điểm sau:

- Các mẫu luyện được thể hiện bởi nhiều cặp giá trị-thuộc tính (ví dụ, điểm ảnh)
- Các mẫu luyện có thể có lỗi
- Chấp nhận thời gian huấn luyện dài
- Cần đánh giá nhanh hàm mục tiêu được học
- Không cần hiểu giả thiết cuối cùng vì NN được coi là hộp đen

38

Perceptron



$$o(x_1, \dots, x_n) = \begin{cases} 1 & \text{nếu } w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n > 0 \\ -1 & \text{trong trường hợp ngược lại} \end{cases}$$

hay:

$$o(\vec{x}) = \text{sgn}(\vec{w} \cdot \vec{x})$$

trong đó

$$\text{sgn}(y) = \begin{cases} 1 & \text{nếu } y > 0 \\ -1 & \text{nếu ngược lại} \end{cases}$$

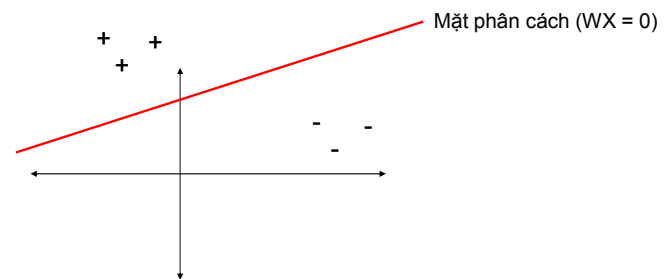
Nhiệm vụ học: tìm các giá trị của w

Không gian giả thiết: không gian các vector trọng số

39

Khả năng của Perceptron

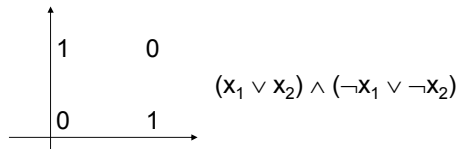
Mỗi perceptron tạo ra 1 mặt phân cách siêu phẳng trên không gian đầu vào n chiều



40

Khả năng của Perceptron

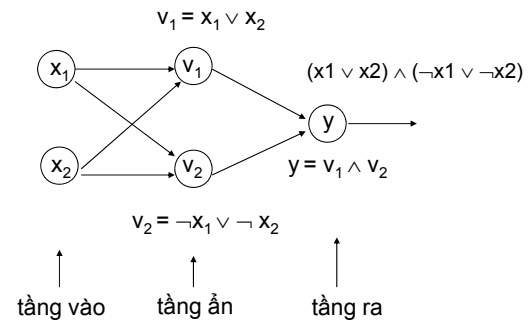
- có thể học các hàm \neg , \wedge , \vee , NAND, NOR
- không biểu diễn được các hàm không phân tách được bằng đường tuyến tính, vd XOR



- Mọi hàm logic đều có thể biểu diễn bằng 1 mạng perceptron có ít nhất 2 tầng

41

Mạng nơon biểu diễn hàm XOR



42

Học các trọng số mạng

- Luật perceptron:
dùng khi tập luyện
 - phân tách được bằng 1 đường tuyến tính
 - đủ nhỏ
- Luật delta:
dùng khi tập luyện không phân thể tách tuyến tính

43

Luật huấn luyện Perceptron

- Khởi tạo một vector có các trọng số ngẫu nhiên
- Lặp lại hàm perceptron cho mỗi mẫu luyện đến khi hàm perceptron phân loại đúng tất cả các mẫu luyện:
 - Các trọng số được sửa đổi tại mỗi bước dựa vào luật huấn luyện perceptron:

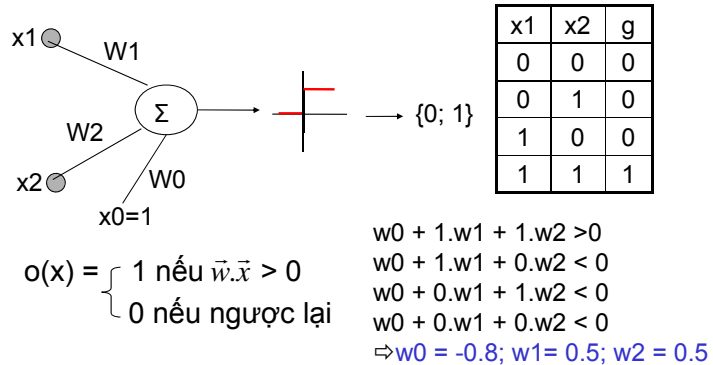
$$w_i \longleftarrow w_i + \Delta w_i$$
 trong đó

$$\Delta w_i \longleftarrow \Delta w_i + \eta(t - o)x_i$$
 với
 - $t = c(\vec{x})$ là hàm đích
 - o là đầu ra perceptron
 - η = tốc độ học, là hằng số nhỏ

44

Ví dụ...

Biểu diễn $g(x_1, x_2) = \text{AND}(x_1, x_2)$



45

Ví dụ...

$$\vec{w} \cdot \vec{x} = \sum_{i=0}^2 w_i * x_i$$

$$w_i \leftarrow w_i + \Delta w_i \quad \Delta w_i \leftarrow \Delta w_i + \eta(t - o)x_i$$

Khởi tạo các giá trị đầu: $\Delta w_0 = 0, \Delta w_1 = 0, \Delta w_2 = 0$

$w_0 = -1.5, w_1 = -0.5, w_2 = 0.5, \eta = 0.1$

$$\Delta w_0 = \Delta w_0 + \eta * (t - o) * x_0 = 0 + 0.1 * (0 - 0) * 1 = 0$$

$$\Sigma = x_0.w_0 + x_1.w_1 + x_2.w_2 = 1.w_0 + 0.w_1 + 0.w_2 = w_0 = -1.5$$

$$w_0 = w_0 + \Delta w_0 = -1.5 + 0 = -1.5, w_1 = -0.5, w_2 = 0.5$$

x0	x1	x2	t	Σ	o	Δw0	Δw1	Δw2
1	0	0	0	-1.5	0	0	0	0
1	0	1	0	-1	0	0	0	0
1	1	0	0	-2	0	0	0	0
1	1	1	1	-1.5	0	0.1	0.1	0.1

$$\Delta w_0 = \Delta w_0 + \eta * (t - o) * x_0 = 0 + 0.1 * (1 - 0) * 1 = 0.1$$

$$w_0 = w_0 + \Delta w_0 = -1.5 + 0.1 = -1.4, w_1 = -0.4, w_2 = 0.6$$

46

Ví dụ...

$$\vec{w} \cdot \vec{x} = \sum_{i=0}^2 w_i * x_i$$

$$w_i \leftarrow w_i + \Delta w_i \quad \Delta w_i \leftarrow \Delta w_i + \eta(t - o)x_i$$

$\Delta w_0 = 0.1, \Delta w_1 = 0.1, \Delta w_2 = 0.1$

$w_0 = -1.4, w_1 = -0.4, w_2 = 0.6, \eta = 0.1$

$$\Sigma = x_0.w_0 + x_1.w_1 + x_2.w_2 = 1.w_0 + 0.w_1 + 0.w_2 = w_0 = -1.4$$

$$\Delta w_0 = \Delta w_0 + \eta * (t - o) * x_0 = 0.1 + 0.1 * (0 - 0) * 1 = 0.1$$

$$w_0 = w_0 + \Delta w_0 = -1.4 + 0.1 = -1.3, w_1 = -0.3, w_2 = 0.7$$

x0	x1	x2	t	Σ	o	Δw0	Δw1	Δw2
1	0	0	0	-1.4	0	0.1	0.1	0.1
1	0	1	0	-0.6	0	0.1	0.1	0.1
1	1	0	0					
1	1	1	1					

47

Ứng dụng của mạng nơ-ron - Nhận dạng mặt



48

Ứng dụng của mạng nơron - Nhận dạng mặt

- Có nhiều hàm đích có thể học trong việc nhận dạng ảnh:
 - xác định người
 - hướng quay (trái, phải, thẳng, ...)
 - giới tính
 - có đeo kính hay không

49

Ứng dụng của mạng nơron - Nhận dạng mặt

- Nhiệm vụ học: phân loại các hình ảnh camera về mặt người với nhiều góc độ khác nhau
- CSDL hình ảnh
- Các hình ảnh với 624 grayscale: 20 người, mỗi người khoảng 32 ảnh
- Nhiều cách biểu cảm (vui, buồn, giận, bình thường)
- Các hướng khác nhau (trái, phải, thẳng, hướng lên)
- Độ phân giải 120x128
- Học hướng quay của mặt người:
 - không cần các lựa chọn tối ưu, phương pháp này cho kết quả tốt
 - sau khi luyện trên 260 hình ảnh, việc phân loại đạt độ chính xác trên tập thử là 90%

50

Các lựa chọn

1. Mã hoá đầu vào: hình ảnh hay các đặc tính
2. Mã hoá đầu ra: số lượng đầu ra, các hàm đích cho đầu ra
3. Cấu trúc mạng: số lượng nút mạng và liên kết giữa chúng
4. Các tham số thuật toán học
 - Tốc độ học
 - giá trị momentum

51

Mã hoá đầu vào

- Thiết kế các lựa chọn
- Tiền xử lý hình ảnh để rút ra các hướng, các vùng có mật độ giống nhau, hoặc các đặc tính hình ảnh cục bộ khác
- Khó khăn: số cạnh có thể thay đổi, trong khi NN có số lượng cố định các đầu vào
- Các hình đã mã hoá là 1 tập cố định các giá trị mật độ 30x32 điểm ảnh (tóm tắt về độ phân giải của ảnh ban đầu), từ 0 đến 255



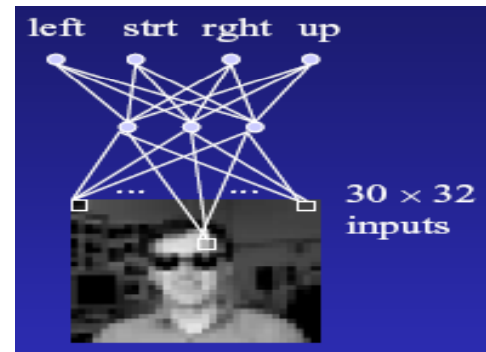
52

Mã hoá đầu ra

- Mỗi đầu ra: 4 giá trị xác định hướng mà người nhìn (trái, phải, thẳng, hướng lên)
- Mỗi đơn vị: phân loại sử dụng 1 đầu ra, gán 0.2, 0.4, 0.6 và 0.8 cho 4 giá trị
- Chọn 1 trong n đầu ra mã hoá:
 - cung cấp nhiều mức độ tự do để biểu diễn hàm đích (n lần số trọng số ở tầng ra)
 - độ khác nhau giữa giá trị cao nhất và nhì dùng để đo độ tin cậy

53

Cấu trúc mạng



mạng 960 x 3 x 4

54