CHƯƠNG 7: Kỹ thuật nhận dạng

- Giới thiệu
- Cây quyết định
- Rừng ngẫu nhiên
- Boosting
- Phương pháp máy vector hỗ trợ
- Kỹ thuật học sâu

CHƯƠNG 7: Kỹ thuật nhận dạng

- Kết hợp các phương pháp trí tuệ nhân tạo với xử lý đặc trưng ảnh vào việc phát hiện, nhận dạng và định danh đối tượng.
- Các kỹ thuật học máy liên quan:
  - Cây quyết định
  - Mạng nơron
  - SVM
  - Boosting
  - Rừng ngẫu nhiên

7.1. Giới thiệu

- Nhận dạng đối tượng nhằm mục đích phân loại các mẫu dựa trên kiến thức có trước về đối tượng hoặc dựa vào thông tin thống kê, đo lường được trích rút ra từ các mẫu trong tập dữ liệu huấn luyện
- Lĩnh vực nhận dạng đối tượng liên quan đến các phương pháp, kỹ thuật của xử lý ảnh và học máy

7.1. Giới thiệu

- Các mẫu dữ liệu thường được số hóa và biểu diễn thành vector đặc trưng trong một không gian đa chiều tương ứng, được gọi là không gian đặc trưng
- Nhờ vào vector đặc trưng này để có thể phân tích, đánh giá nhằm xây dựng được mô hình đối tượng phục vụ phân loại, nhận dạng chúng

7.1. Giới thiệu

- Trong nhận dạng, các phương pháp học máy có giám sát và không giám sát đều được nghiên cứu và ứng dụng
- Các kỹ thuật học máy có giám sát thường được sử dụng như cây quyết định, mạng neural, SVM, Boosting, rừng ngẫu nhiên (random forest).

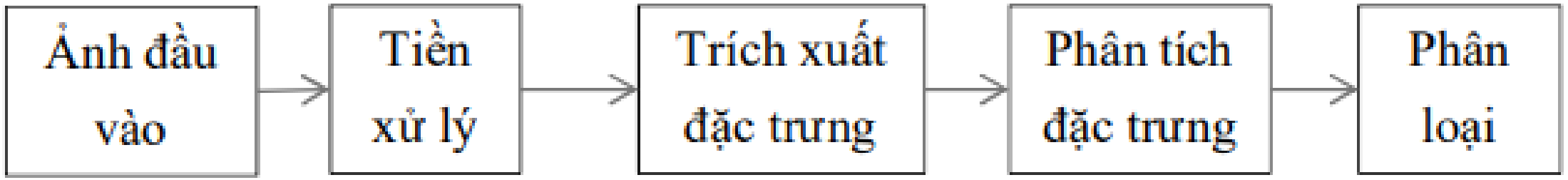
7.1. Giới thiệu

- Nhận dạng theo học máy có giám sát thì việc phân loại thường dựa vào tập dữ liệu mẫu đã được gán nhãn theo các lớp bởi các ‘chuyên gia’ để phân tích và xây dựng mô hình nhận dạng
- Tập dữ liệu mẫu để học được gọi là tập dữ liệu huấn luyện và quá trình phân tích, xây dựng mô hình đối tượng được gọi là quá trình huấn luyện máy nhận dạng hay huấn luyện mô hình

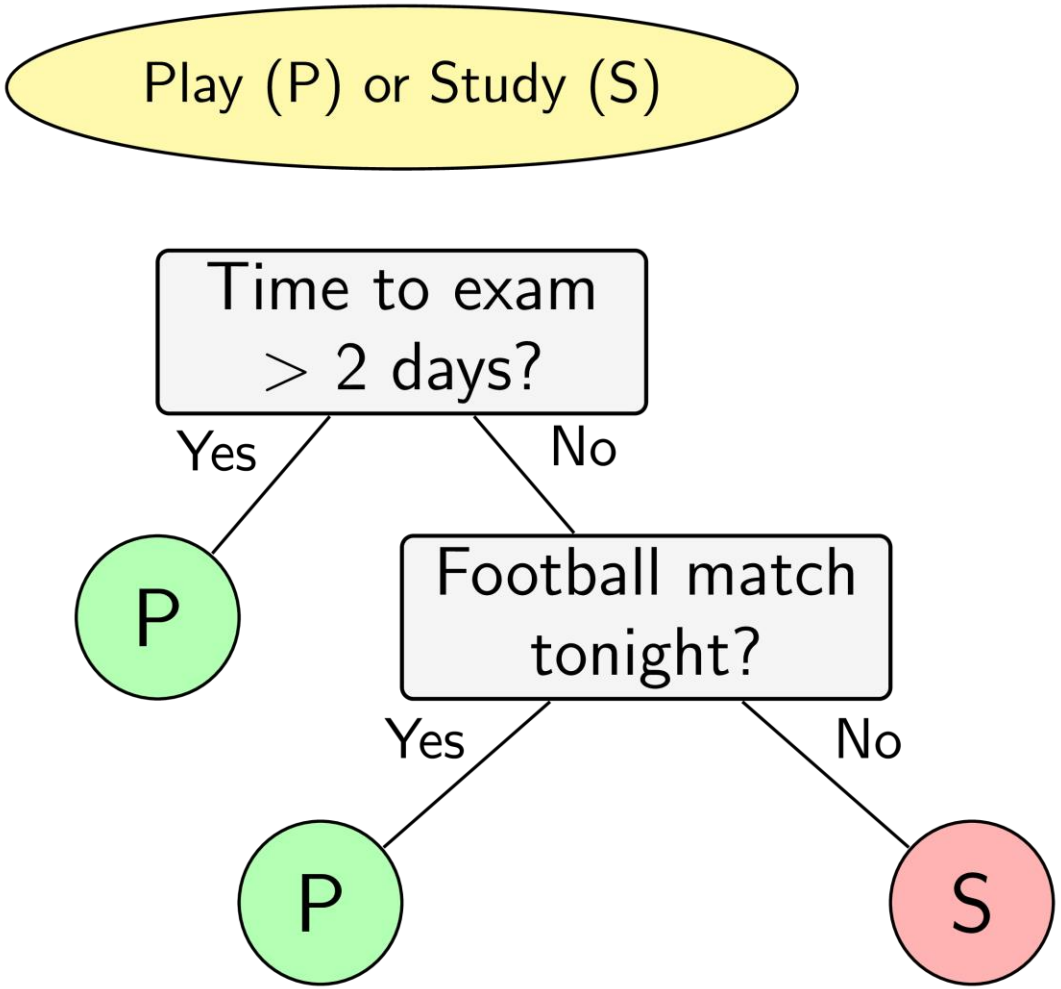
7.1. Giới thiệu

- Phương pháp học không có giám sát, tập dữ liệu phân loại không cần được gán nhãn trước mà bản thân thuật toán phải tự phân loại, xác định lớp của đối tượng dựa vào phân tích, thống kê từ các đặc trưng của tập mẫu dữ liệu đưa vào.

7.1. Giới thiệu



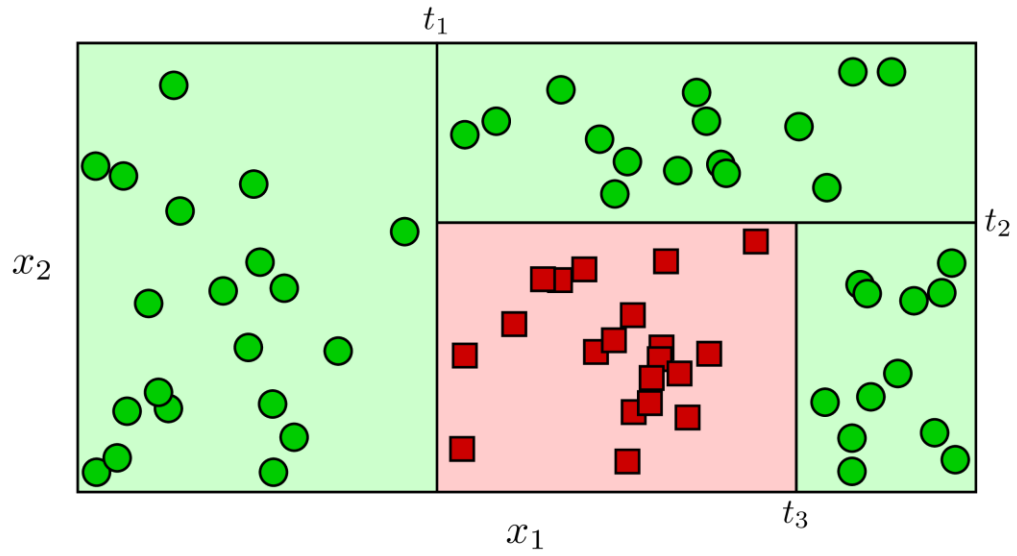
7.2. Cây quyết định (decision tree)



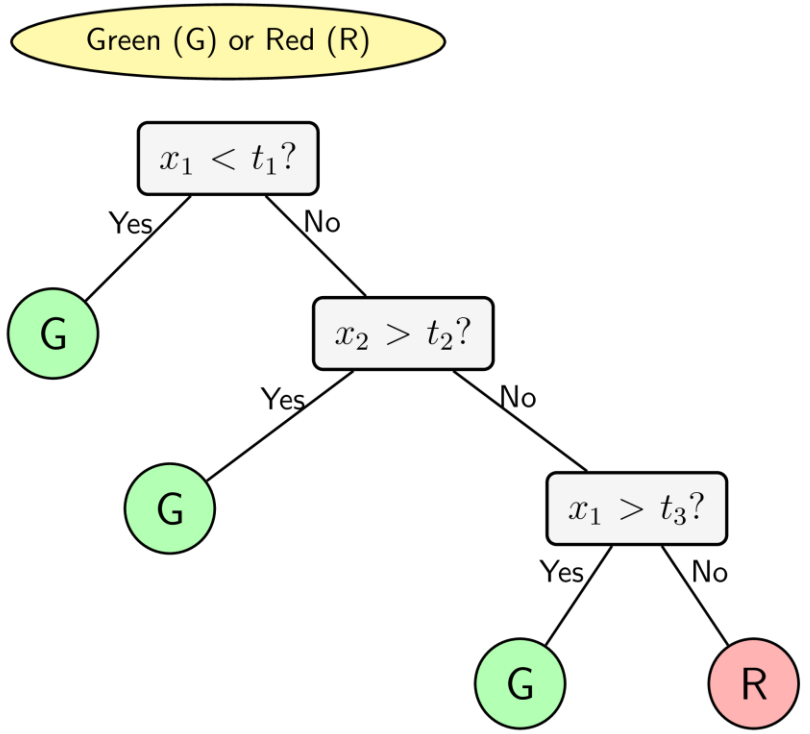
7.2. Cây quyết định (decision tree)

- Việc quan sát, suy nghĩ và ra các quyết định của con người thường được bắt đầu từ các câu hỏi. Machine learning cũng có một mô hình ra quyết định dựa trên các câu hỏi. Mô hình này có tên là *cây quyết định* (*decision tree*).

7.2. Cây quyết định (decision tree)



(a)



(b)

7.2. Cây quyết định (decision tree)

- Cây quyết định là một kiểu mô hình dự báo, được xây dựng trên cơ sở cấu trúc cây, dùng để phân lớp các mẫu dữ liệu dựa vào chuỗi các luật suy diễn.
- Các nút lá đại diện cho các quyết định phân loại, các nhánh đại diện cho luật kết hợp các thuộc tính để dẫn tới phân loại nào đó.
- Một cây quyết định có thể được huấn luyện bằng cách chia tập dữ liệu huấn luyện thành các tập con để kiểm tra theo từng giá trị thuộc tính đơn hoặc một nhóm các thuộc tính.

7.2. Cây quyết định (decision tree)

- Việc phân loại có thể được mô tả như là các kết hợp phân loại đơn giản bằng cách sử dụng kỹ thuật suy diễn toán học
- Quá trình huấn luyện mô hình phân loại là quá trình xây dựng cây quyết định. Hiện nay có nhiều phương pháp xây dựng cây quyết định cho bài toán phân loại đối tượng như ID3, C4.5,...

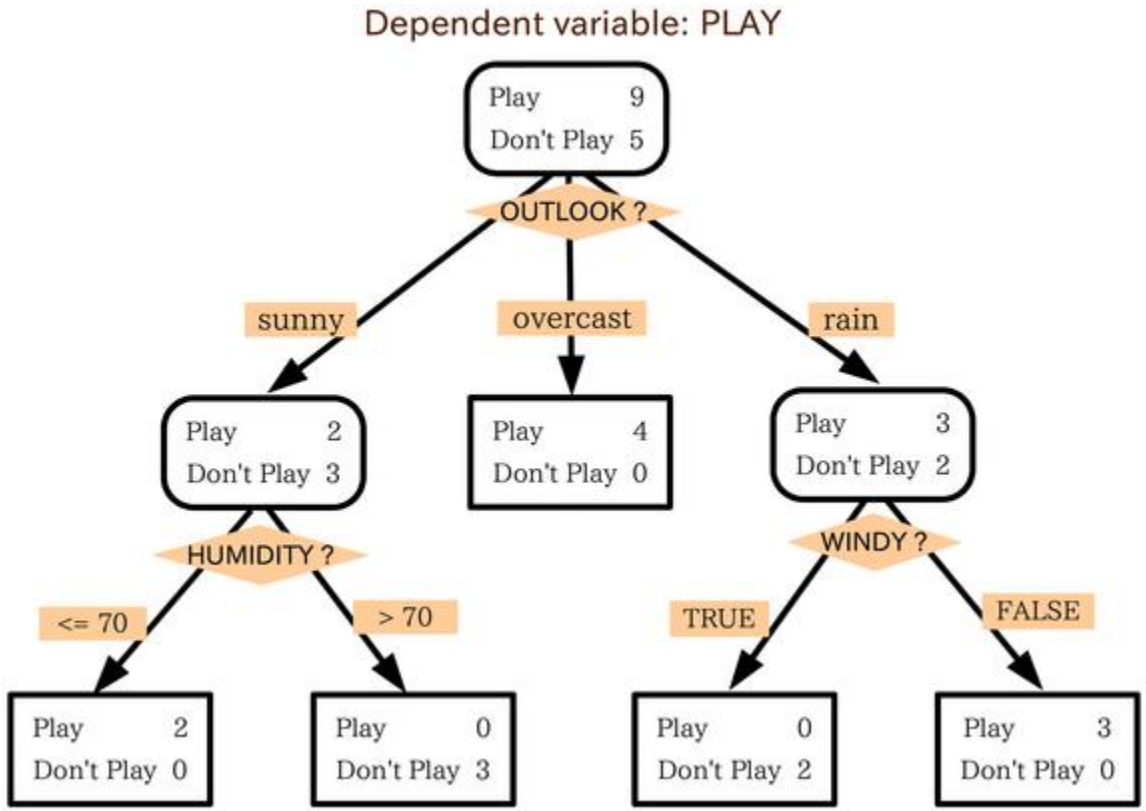
7.2. Cây quyết định (decision tree)

- David là quản lý của một câu lạc bộ đánh golf nổi tiếng. Anh ta đang có rắc rối chuyện các thành viên đến hay không đến. Có ngày ai cũng muốn chơi golf nhưng số nhân viên câu lạc bộ lại không đủ phục vụ. Có hôm, không hiểu vì lý do gì mà chẳng ai đến chơi, và câu lạc bộ lại thừa nhân viên.
- Mục tiêu của David là tối ưu hóa số nhân viên phục vụ mỗi ngày bằng cách dựa theo thông tin dự báo thời tiết để đoán xem khi nào người ta sẽ đến chơi golf. Để thực hiện điều đó, anh cần hiểu được tại sao khách hàng quyết định chơi và tìm hiểu xem có cách giải thích nào cho việc đó hay không.
- Vậy là trong hai tuần, anh ta thu thập thông tin về:
- Trời (*outlook*) (nắng (*sunny*),
- Và tất nhiên là số người đến chơi golf vào hôm đó. David thu được một bộ dữ liệu gồm 14 dòng và 5 cột.

7.2. Cây quyết định (decision tree)

Play golf dataset

Independent variables				Dep. var
OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
overcast	83	78	FALSE	Play
rain	70	96	FALSE	Play
rain	68	80	FALSE	Play
rain	65	70	TRUE	Don't Play
overcast	64	65	TRUE	Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
rain	75	80	FALSE	Play
sunny	75	70	TRUE	Play
overcast	72	90	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play



7.2. Cây quyết định (decision tree)

- Decision tree là một mô hình supervised learning, có thể được áp dụng vào cả hai bài toán classification và regression. Việc xây dựng một decision tree trên dữ liệu huấn luyện cho trước là việc đi xác định các *câu hỏi* và *thứ tự của chúng*. Một điểm đáng lưu ý của decision tree là nó có thể làm việc với các đặc trưng (trong các tài liệu về decision tree, các đặc trưng thường được gọi là *thuộc tính – attribute*) dạng *categorical*, thường là rời rạc và không có thứ tự. Ví dụ, *mưa, nắng* hay *xanh*, *đỏ*, v.v. Decision tree cũng làm việc với dữ liệu có vector đặc trưng bao gồm cả thuộc tính dạng categorical và liên tục (*numeric*). Một điểm đáng lưu ý nữa là decision tree ít yêu cầu việc chuẩn hoá dữ liệu.

7.2. Cây quyết định (decision tree)

Thuật toán ID3 (Iterative Dichotomiser 3) xây dựng cây quyết định

- Bước1: Tínhgiá trị entropy lần lượt cho tất cả thuộc tính của tập dữ liệu S.
- Bước2: Lựa chọn thuộc tính A "tốt nhất"(entropy nhỏ nhất hoặc độ lợi thông tin lớn nhất) để tạo một nút của cây quyết định.
- Bước3: Sử dụng thuộc tính A để phân chia tập S thành các tập con Su, mỗi tập con được đưa về mỗi nhánh nhất định tùy theo giá trị cụ thể.
- Bước4:Lặp lại quá trình này cho các cây con tương ứng với các tập Su cho mỗi cây con(Su đóng vai trò như tập S của lần lặp kế tiếp).

7.2. Cây quyết định (decision tree)

Quá trình xây dựng cây con được kết thúc nếu thỏa mãn một trong các điều kiện sau:

- Tất cả phần tử của tập con hiện tại Su đều thuộc về cùng một lớp ci cụ thể. Khi đó nút đang xét ở nhánh hiện tại được thiết lập thành nút lá và trả về giá trị lớp ci là lớp của tập con này. Nghĩa là khi một mẫu dữ liệu được phân về nhánh này thì mẫu dữ liệu sẽ được xác định thuộc về lớp ci.
- Không còn thuộc tính nào nữa để lựa chọn (đã chọn hết ở các lần lặp trước đó), nhưng vẫn còn các mẫu vẫn chưa xác định thuộc về lớp nào cả thì nút này sẽ được thiết lập thành nút lá và trả về giá trị là lớp có nhiều mẫu nhất trong tập con Su hiện tại (thực hiện theo nguyên tắc bỏ phiếu, lớp nào nhiều nhất sẽ được sử dụng làm giá trị trả về của nút lá).
- Không còn mẫu dữ liệu nào nữa trong tập con Su, thì tạo ra nút lá mới và giá trị trả về là lớp phổ biến nhất trong tập con tại nút cha của nó.

7.2. Cây quyết định (decision tree)

- Cây quyết định được xây dựng với mỗi nút không phải là nút lá đại diện cho một thuộc tính (nhóm thuộc tính) được chọn trên tập dữ liệu đã được chia, nút lá đại diện cho nhãn là giá trị của lớp được trả về của cây tại nhánh đó.
- Độ đo entropy H là phép đo mức độ không chắc chắn trong tập dữ liệu S được tính theo công thức sau:

$$H(S) = \sum_{c \in C} -p(c) \log_2 p(c)$$

- C là tập nhãn lớp của S, p(c) là tỷ lệ số phần tử thuộc lớp c đối với tổng số phần tử có trong tập S.

7.2. Cây quyết định (decision tree)

- Độ lợi thông tin IG cũng được sử dụng trong đánh giá độ chắc chắn thông tin, là độ đo sự khác biệt của entropy từ đầu cho đến khi S được phân tách dựa vào thuộc tính A. Nói cách khác mức độ không chắc chắn trong S được giảm xuống sau phân tách S dựa vào thuộc tính A, được tính theo công thức:

$$IG(A,S) = H(S) - \sum_{t \in T} p(t)H(t)$$

Với $H(t)$ là giá trị entropy của tập con t, T là tập con được tạo ra từ việc phân tách tập S dựa trên thuộc tính A sao cho $S = \bigcup_{t \in T} t$.

$P(t)$ là tỷ lệ giữa số phần tử trong t đối với số phần tử trong tập S.

7.2. Cây quyết định (decision tree)

- **Thuật toán C4.5 xây dựng cây quyết định**

Một số phát triển của C4.5 so với thuật toán cơ sở ID3 như sau:

- Giải quyết cả hai loại thuộc tính liên tục và rời rạc: Để giải quyết thuộc tính liên tục, C4.5 tạo ra một ngưỡng sau đó phân tách danh sách theo tập có giá trị thuộc tính trênngưỡng và tập có giá trị nhỏ hơn hoặc bằng ngưỡng.

- Giải quyết vấn đề dữ liệu huấn luyện có giá trị thuộc tính thiếu: C4.5 cho phép giá trịthuộc tính được đánh dấu"?"cho trường hợp thuộc tính thiếu. Giá trị thuộc tính thiếu sẽ không được sử dụng trong việc tính giá trị entropy và độ lợi thông tin.

Thuật toán C4.5 xây dựng cây quyết định

- Giải quyết các thuộc tính với chi phí (cost) khác nhau.
- Giải quyết vấn đề tỉa cành của cây sau khi tạo: C4.5 quay lại toàn bộ cây sau khi được tạo ra và cố gắng loại bỏ các cành không hữu ích thay thể chúng bằng các nút lá.

Thuật toán C4.5 xây dựng cây quyết định

Có một số trường hợp đặc biệt như sau:

- Tất cả các mẫu trong danh sách đều thuộc về cùng một lớp, khi đó đơn giản chỉ việc tạo ra nút lá cho cây quyết định và gán nhãn giá trị trả về là lớp của tập mẫu dữ liệu này.
- +Không có thuộc tính nào để cung cấp thông tin nữa, khi đó C4.5 tạo ra một nút quyết định cao hơn được gọi là nút lá và gán nhãn giá trị trả về là giá trị của lớp chiếm phần đông trong tập dữ liệu hiện tại.
- +Trường hợp lớp không nhìn thấy trước, C4.5 tạo ra một nút quyết định trên cây sử dụng giá trị kỳ vọng làm giá trị trả về của nút.

7.3. Rừng ngẫu nhiên (Random Forest)

- Rừng ngẫu nhiên (randomforest-RF) được phát triển chủ yếu dựa vào tập các cây quyết định (decisiontree) được xây dựng bằng cách lựa chọn ngẫu nhiên các tập con thuộc tính từ tập thuộc tính ban đầu. Việc lựa chọn tập con thuộc tính ngẫu nhiên không nhất thiết phải tách rời nhau. Như vậy việc lựa chọn thuộc tính và xây dựng mỗi cây được thực hiện theo thuật toán có tính chất ngẫu nhiên.

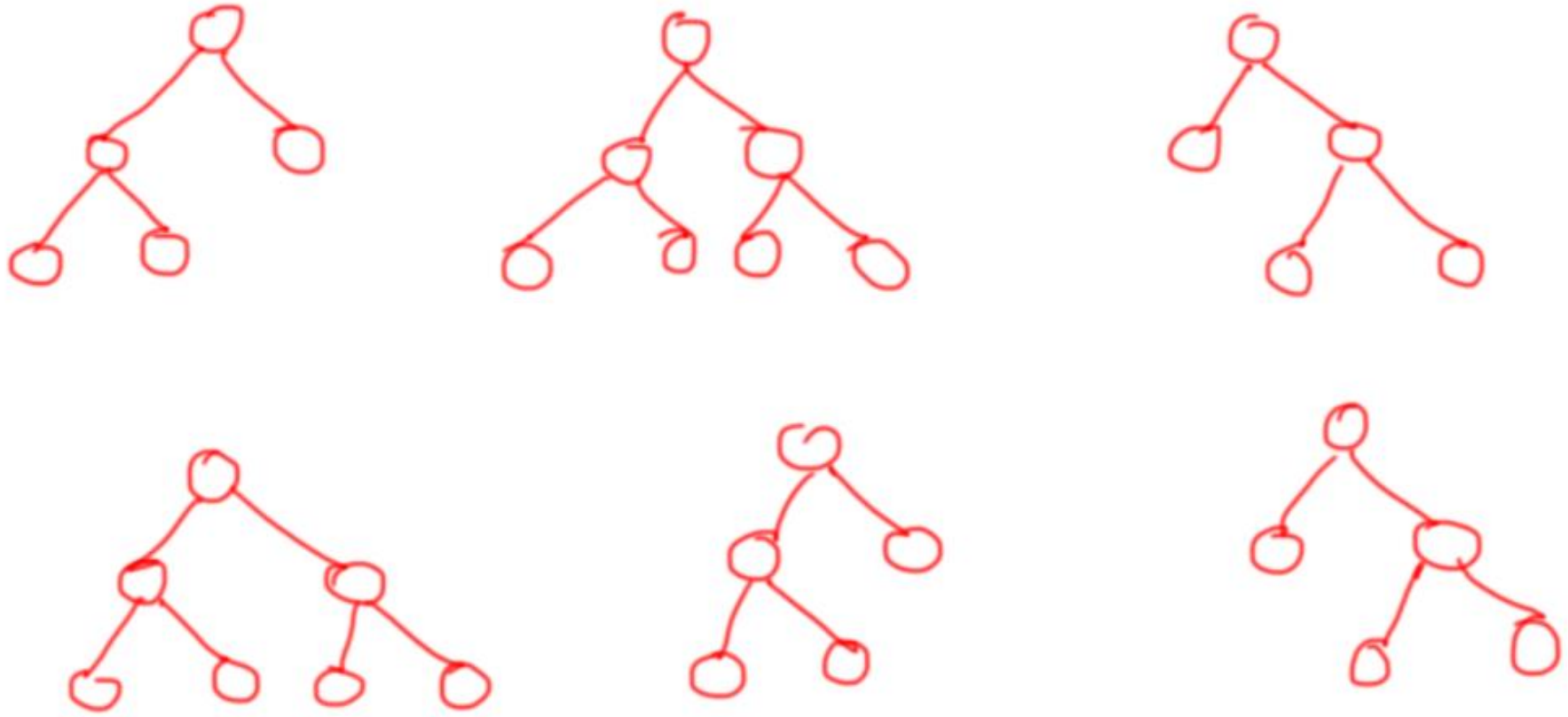
7.3. Rừng ngẫu nhiên (Random Forest)

- Rừng ngẫu nhiên cũng là một dạng thuật toán học có giám sát.
- Một số điểm mạnh của thuật toán RF là có thể sử dụng cho cả bài toán phân loại và bài toán hồi quy, có thể xử lý trong trường hợp dữ liệu thiếu giá trị, khi rừng có nhiều cây hơn có thể giúp giải quyết bài toán overfitting với dữ liệu.
- Kỹ thuật rừng ngẫu nhiên được sử dụng nhiều trong lĩnh vực thị giác máy tính, phân loại đối tượng.

7.3. Rừng ngẫu nhiên (Random Forest)

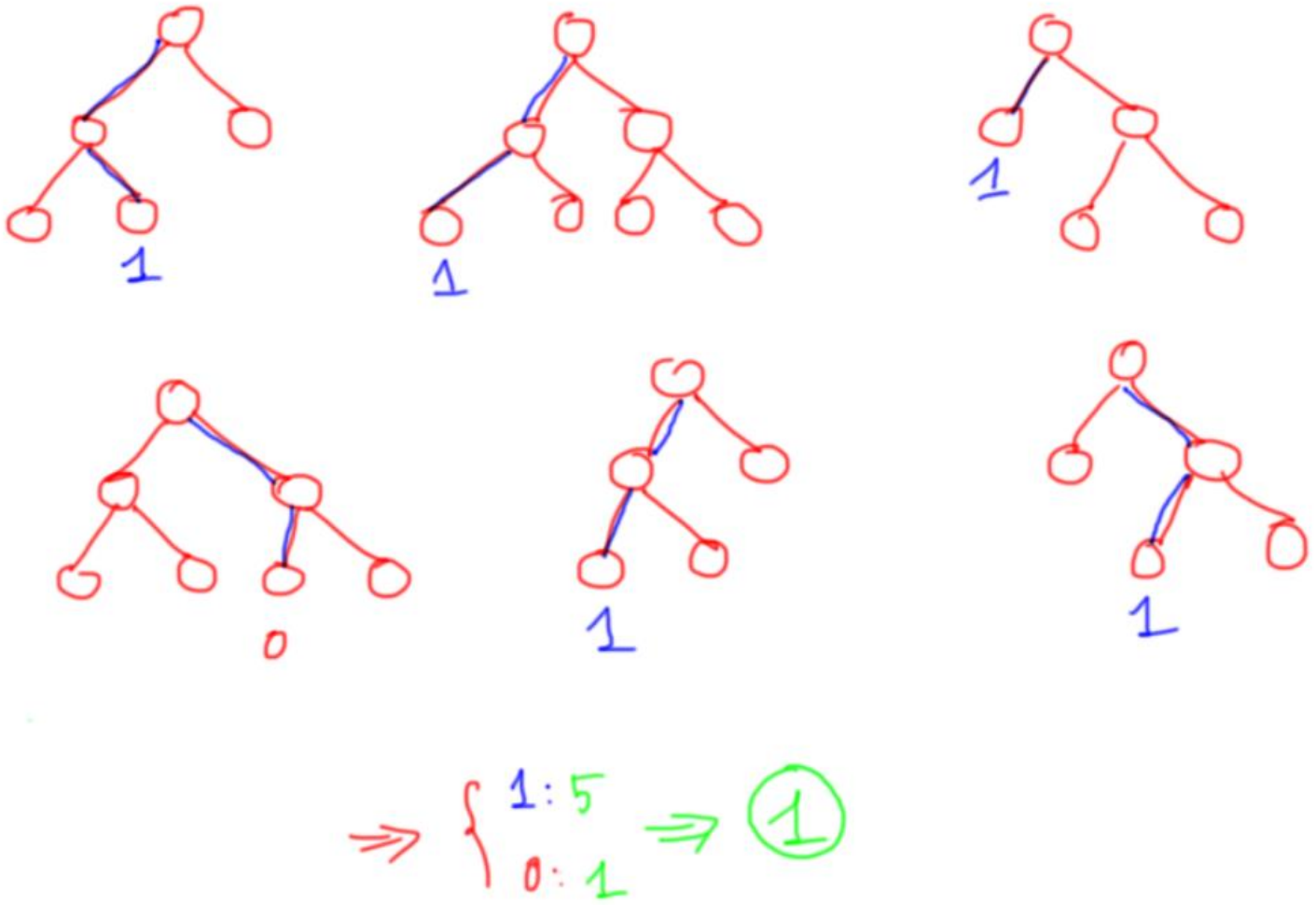
- Đầu vào là bộ dữ liệu huấn luyện (X,Y) với X là tập mẫu dữ liệu với m đặc trưng (thuộc tính) và Y là tập nhãn tương ứng;
- Bước1: Chọn ngẫu nhiên k thuộc tính từ tập m thuộc tính với k<<m; Bước2: Từ tập k thuộc tính, xác định thuộc tính d có khả năng phân loại “tốt nhất” và tạo ra một nút phân loại;
- Bước3: Chia tập dữ liệu trên tập thuộc tính k theo nút d vừa tìm được;
- Bước4: Lặp lại bước 1 -3 cho đến khi đã xây dựng xong cây với tập thuộc tính k;
- Bước5: Lặp lại bước1-4 để xây dựng tập các cây của rừng ngẫu nhiên.

7.3. Rừng ngẫu nhiên (Random Forest)



7.3. Rừng ngẫu nhiên (Random Forest)- dừng CNTT

Thuật toán Random Forest có 6 cây quyết định, 5 cây dự đoán 1 và 1 cây dự đoán 0, do đó sẽ vote là cho ra dự đoán cuối cùng là 1.



7.4. Kỹ thuật Boosting

Một dạng thuật toán học máy quần thể bằng cách xây dựng nhiều bộ phân loại cùng lúc và sau đó kết hợp chúng lại theo trọng số xác định của từng phân loại thành phần.

• Mỗi bộ phân loại thành phần được gọi là phân loại yếu, các bộ phân loại yếu được hợp lại với nhau tạo thành một phân loại mạnh.

7.4. Kỹ thuật Boosting

- AdaBoost là bộ phân loại mạnh phi tuyến hoạt động trên nguyên tắc kết hợp các bộ phân loại yếu theo trọng số để tạo ra một bộ phân loại mạnh hơn theo kiểu thích ứng (với mẫu dữ liệu).

- AdaBoost sử dụng các trọng số để đánh dấu các mẫu khó phân loại. Càng vào những mức sâu của phân loại yếu, bộ phân loại càng tập trung vào những mẫu khó phân loại, trong khi những mẫu phân loại dễ sẽ có giá trị ảnh hưởng nhỏ hơn.

7.5. Phương pháp máy véc tơ hỗ trợ

7.5.1 Giới thiệu

• Phương pháp máy vector hỗ trợ (Support vector machine-SVM): phương pháp học có giám sát.

• Một mô hình SVM là một cách biểu diễn các vector hỗ trợ phân loại trong không gian nhiều chiều và lựa chọn siêu phẳng (hyperplane) phân loại giữa hai lớp sao cho cực đại khoảng cách từ các mẫu dữ liệu huấn luyện (các điểm trong không gian n chiều) tới mặt phẳng phân loại.

7.5.1. Giới thiệu

- Phương pháp phân lớp được sử dụng rộng rãi
- SVM thực hiện hiệu quả trên tập dữ liệu lớn và xử lý hiệu quả trên không gian có số chiều lớn, đặc biệt áp dụng cho các bài toán phân loại dữ liệu hình ảnh, văn bản, và tiếng nói,…
- Thư viện LibSVM là một trong những công cụ được sử dụng khá phổ biến hiện nay.

7.5.2. Phân loại tuyến tính

- Tập dữ liệu huấn luyện S gồm n mẫu $(x_1, y_1),\ldots,(x_n, y_n)$
  - x_i là mẫu dữ liệu được biểu diễn dưới dạng các điểm trong không gian p chiều
  - $y_i \in \{-1\ 1\}$ là nhãn tương ứng để chỉ lớp của mẫu dữ liệu đó
- Mục tiêu: tìm một siêu phẳng có khả năng phân tách tập mẫu thành 2 tập theo nhãn của nó với lề đạt khoảng cách cực đại
  - Lề là khoảng cách từ siêu phẳng đến các điểm (trong p chiều) gần nhất

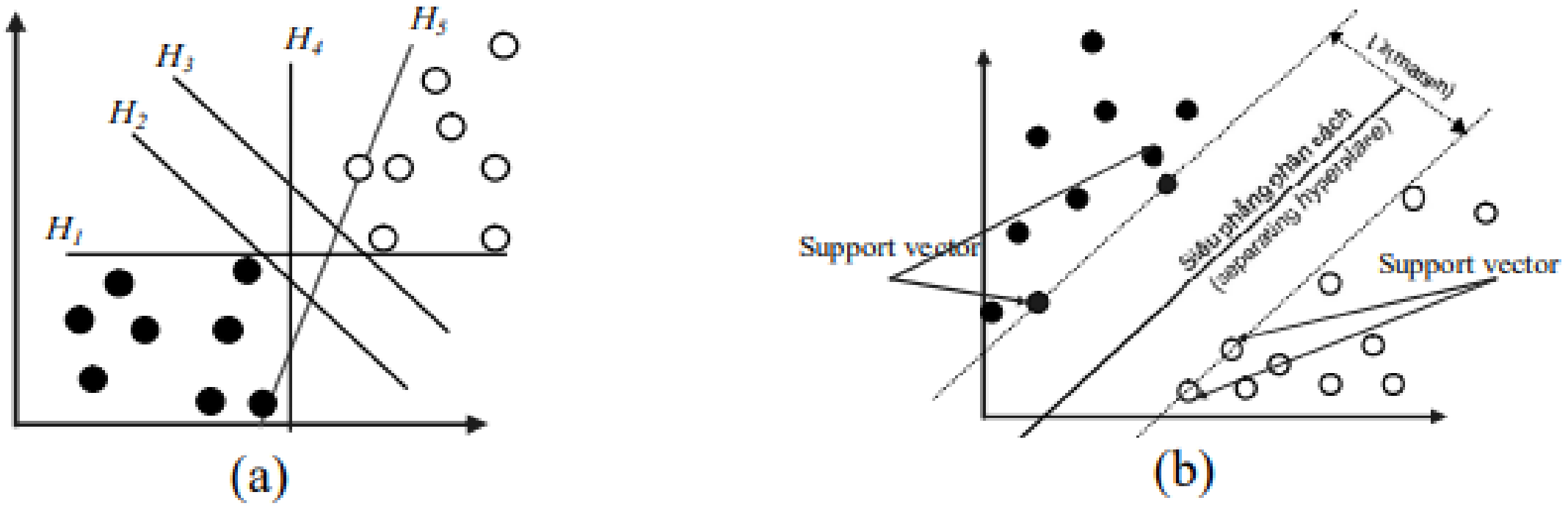
7.5.2. Phân loại tuyến tính

- Siêu phẳng trong không gian đa chiều có thể được viết dưới dạng
- $w \bullet x - b = 0$
- w là vector pháp tuyến của siêu phẳng
- b/|| w || được xác định là khoảng cách từ siêu phẳng đến gốc tọa độ theo vector pháp tuyến w.

7.5.2. Phân loại tuyến tính

- Lề cực đại (maximal margin) là khoảng cách lớn nhất từ siêu phẳng đến các điểm gần nhất, nghĩa là khoảng cách xa nhất từ siêu phẳng đến đường thẳng song song với nó đi qua điểm gần nhất, mà không chứa điểm dữ liệu nào bên trong nó
- Các vector hỗ trợ (support vectors) là các điểm dữ liệu gần nhất nằm trên lề cực đại

7.5.2. Phân loại tuyến tính



Hình 7.2. Siêu phẳng phân tách mẫu dữ liệu thành 2 lớp: a) các siêu phẳng ứng cử viên; b) lề phân loại và các vector hỗ trợ tìm được

7.5.2. Phân loại tuyến tính

$$w.x_i + b \geq +1 \quad khi\ y_i = +1$$
$$w.x_i + b \leq -1 \quad khi\ y_i = -1$$

Kết hợp hai công thức lại ta có :

$$y_i(w.x_i + b) \geq 1\ với\ i = 1,..,n$$

7.5.2. Phân loại tuyến tính

- Đối với vector hỗ trợ x_i là các điểm nằm trên đường biên giới hạn của siêu phẳng H_+:
  - $w.x_i+b= +1$
  - khoảng cách đến gốc tọa độ là $|1-b|/||w||$
- Các điểm nằm trên đường biên giới hạn của siêu phẳng H_-:
  - $w.x_i+b=-1$
  - khoảng cách đến gốc tọa độ là $|-1-b/||w||$

7.5.2. Phân loại tuyến tính

- Bài toán tối ưu tương đối khó giải vì hàm mục tiêu phụ thuộc vào ||w||, là một hàm có khai căn
- Có thể thay ||w|| bằng hàm mục tiêu ½ $||w||^2$ mà không làm thay đổi lời giải đối với bài toán
- Cực tiểu hóa mục tiêu ½ $||w||^2$ theo w và b với điều kiện $y_i(w.x_i+b) \geq 1$

7.5.2. Phân loại tuyến tính

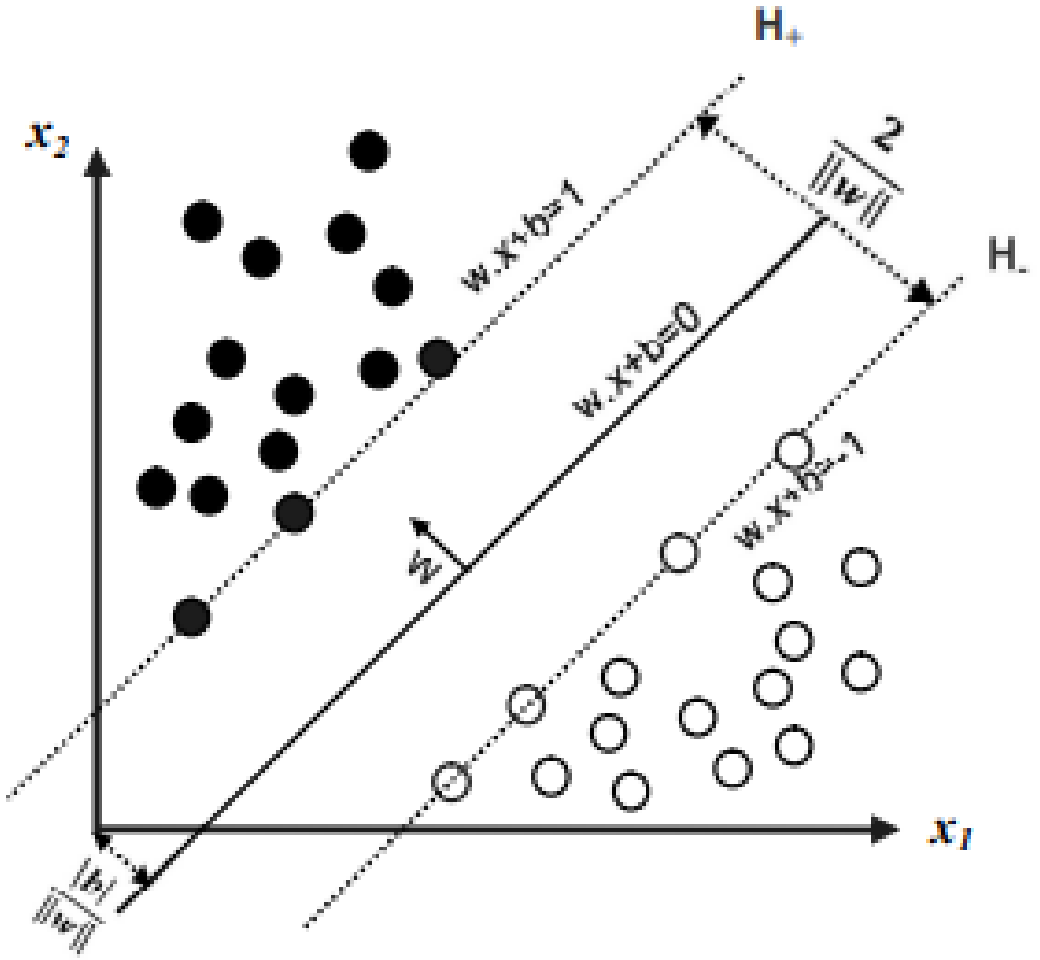
- Bằng cách thêm các nhân tử Lagrange α, bài toán trở thành dạng sau:
- $\min_{w,b} \max_{\alpha \geq 0} \left\{ \frac{1}{2} \|w\|^2 - \sum_{i=1}^{n} \alpha_i [y_i (w . x_i - b) - 1] \right\}$

7.5.2. Phân loại tuyến tính

- Sau khi huấn luyện, các mẫu có $\alpha_i > 0$ được gọi là vector hỗ trợ và nằm trên một trong hai siêu phẳng H_+ hoặc H_-. Khi đã xác định được tập vector hỗ trợ phân loại nghĩa là đã thu được mô hình máy phân loại SVM, với mỗi mẫu thử (đánh giá, testing) có thể đơn giản được phân loại theo siêu phẳng H_+ và H_- tạo ra bằng cách dùng hàm dấu theo công thức sau:

- Sign(w.x+b)

7.5.2. Phân loại tuyến tính



7.5.2. Phân loại tuyến tính

- Theo điều kiện Karush-Kuhn-tucker (KKT):
- $w = \sum_{i=1}^{n} \alpha_i \, y_i x_i$
- Điều kiện KKT được thể hiện như sau:
- $\alpha_i = 0 \leftrightarrow y_i(w.x_i+b) \geq 1$
- $\alpha_i = C \leftrightarrow y_i(w.x_i+b) \leq 1$
- $0 \leq \alpha_i \leq C \leftrightarrow y_i(w.x_i+b) = 1$

7.5.3. Phân loại tuyến tính lề mềm

- Phương pháp lề mềm sử dụng thêm các biến bổ sung ξ_i (xem như giá trị bù cho phần tử sai) nhằm đo độ sai lệch của mẫu bị phân loại sai x_i qua đó giá trị này có tác dụng kéo phần tử x_i về đúng phía nhãn.
- $y_i(w.x_i-b)\geq 1-\xi_i$ với i=1,…,n
- $\min_{w,\xi,b}\left\{\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i\right\}$ $với\ y_i(w.x_i-b)\geq 1-\xi_i,\ \ \xi_i\geq 0$
- $\min_{w,\xi,b}\max_{\alpha,\beta}\left\{\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[y_i(w.x_i-b)-1+\xi_i] - \sum_{i=1}^{n}\beta_i\xi_i\right\}$

7.5.4. Hàm nhân

- Hàm nhân (hay còn gọi là hàm lõi, Kernel function) ký hiệu K là một hàm số trên không gian X, với mọi cặp vector u, v $\in$ X, ta có:
- K(u, v) =$\Phi(u)\Phi(v)$
- Với Φ là ánh xạ từ không gian mẫu X tới một không gian thuộc tính F được xác định Φ: $R^d \rightarrow F$
- $f(x) = \sum_{e_i \neq 0} y_i \alpha_i K(x_i, x) + b$

7.5.4. Hàm nhân

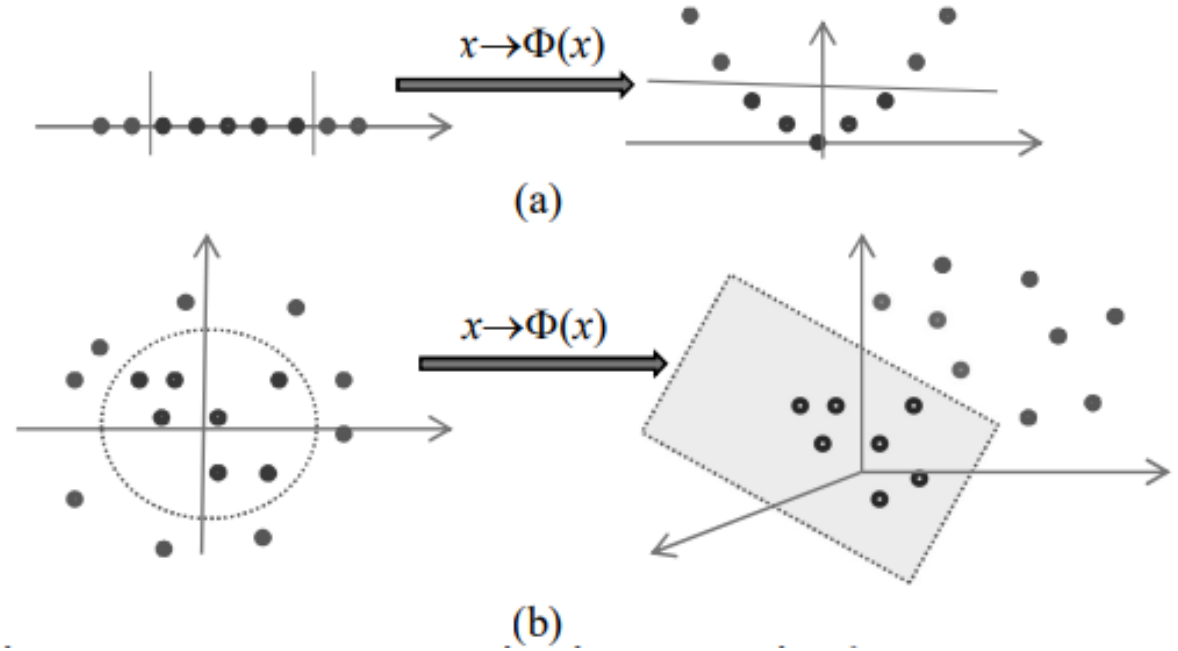
- Hàm nhân đa thức trên không gian R^d có dạng:
- $K(u, v) = (u.v)^p$
- Hàm nhân RBF: Hàm cơ sở bán kính là một hàm giá trị thực mà giá trị của nó chỉ phụ thuộc vào khoảng cách từ gốc tọa độ, sao cho thỏa mãn điều kiện $\Phi(x) = \Phi(\|x\|)$.
- Có một số dạng hàm RBF như hàm mũ $K(u, v) = e^{-\|u-v\|^2/2\sigma^2}$ với σ thường được gọi là độ rộng của hàm hay giá trị độ lệch chuẩn

7.5.5. Tuyến tính hóa phân loại phi tuyến

- Thực hiện ánh xạ vector đặc trưng phi tuyến vào không gian nhiều chiều hơn để có thể tuyến tính hóa rồi dùng phân loại tuyến tính trong không gian mới này
- $\Phi: R^d \rightarrow F$
- Đối với một vector đặc trưng ban đầu $x \in R^d$ và vector đặc trưng đã biến đổi được xác định theo hàm biến đổi $\Phi(x)$ với nhãn của nó vẫn giữ nguyên
- Mẫu huấn luyện ban đầu được biến đổi thành $(\Phi(x1),y1),\ldots,(\Phi(x_n),y_n)$

7.5.5. Tuyến tính hóa phân loại phi tuyến

- Phân loại phi tuyến thực hiện bằng cách biến đổi qua một không gian khác để có thể tuyến tính hóa phân loại



7.6. Mạng neural nhân tạo (ANN- artificial neural network)

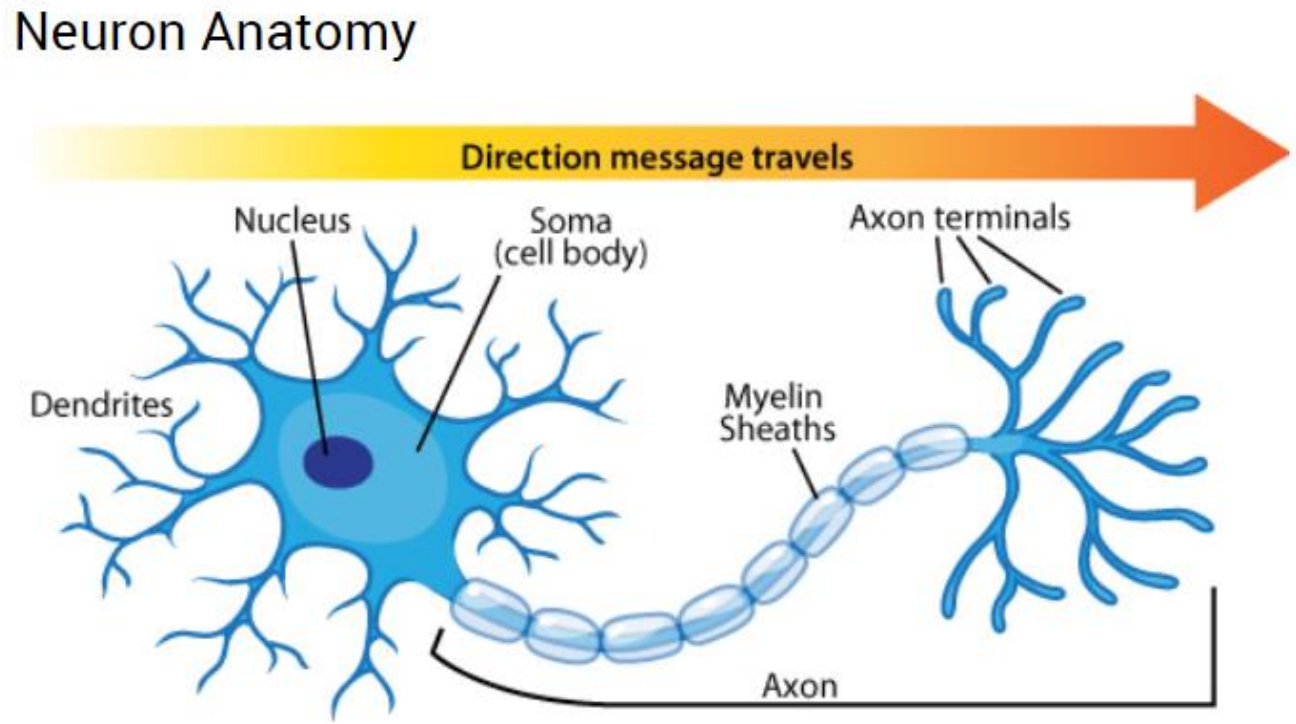
Mạng nơ ron là một hệ thống tính toán lấy cảm hứng từ sự hoạt động của các nơ ron trong hệ thần kinh của con người.

Hoạt động của các nơ ron

- Nơ-ron là đơn vị cơ bản cấu tạo hệ thống thần kinh và là thành phần quan trọng nhất của não.
- Đầu người gồm khoảng 10 triệu nơ-ron và mỗi nơ-ron lại liên kết với tầm 10.000 nơ ron khác.

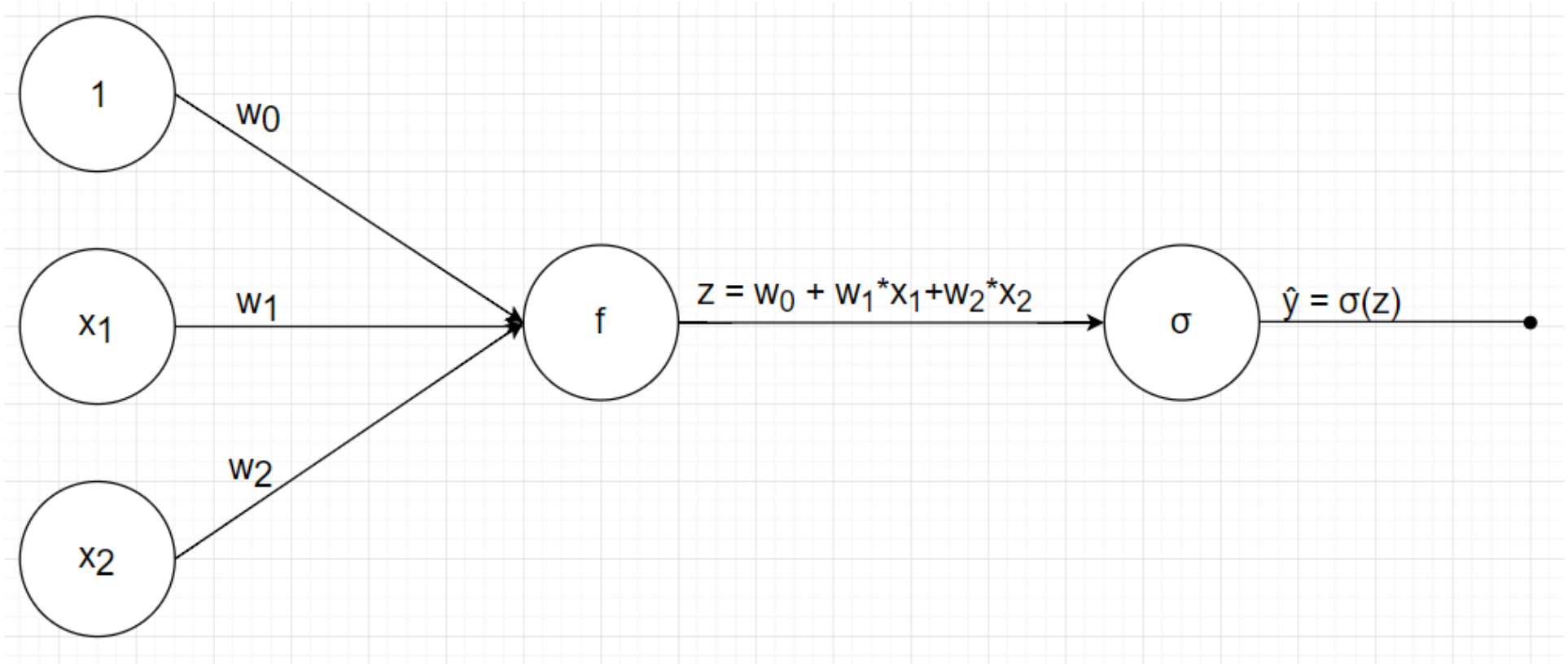
7.6. Mạng neural nhân tạo

- Mỗi nơ-ron có phần thân (soma) chứa nhân, các tín hiệu đầu vào qua sợi nhánh (dendrites) và các tín hiệu đầu ra qua sợi trục (axon) kết nối với các nơ-ron khác.



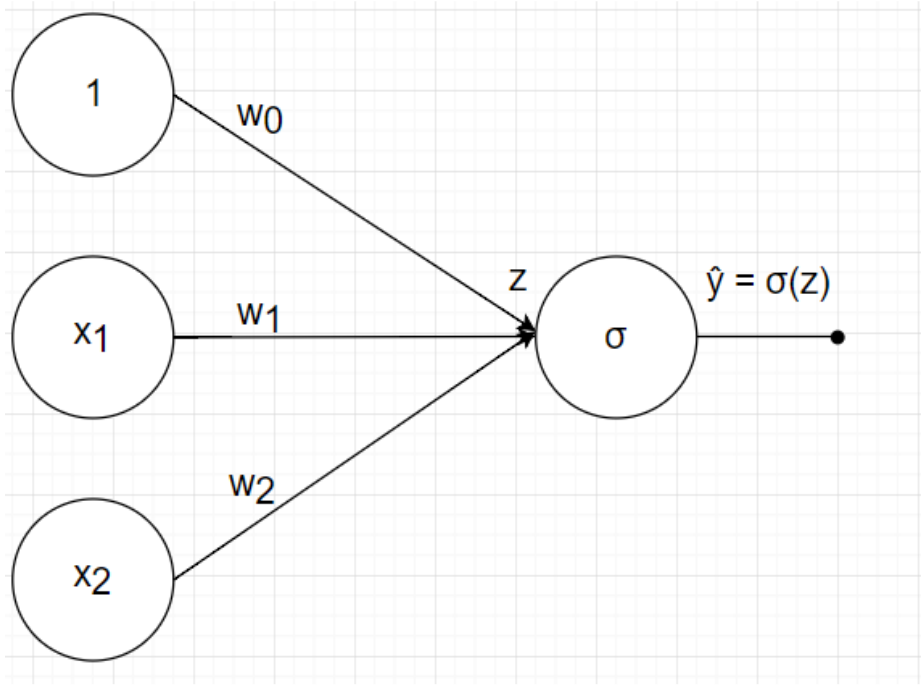
7.6. Mạng neural nhân tạo

Logistic regression là mô hình neural network đơn giản nhất chỉ với input layer và output layer



7.6. Mạng neural nhân tạo

- Để biểu diễn gọn lại ta sẽ gộp hai bước trên thành một trên biểu đồ sau:



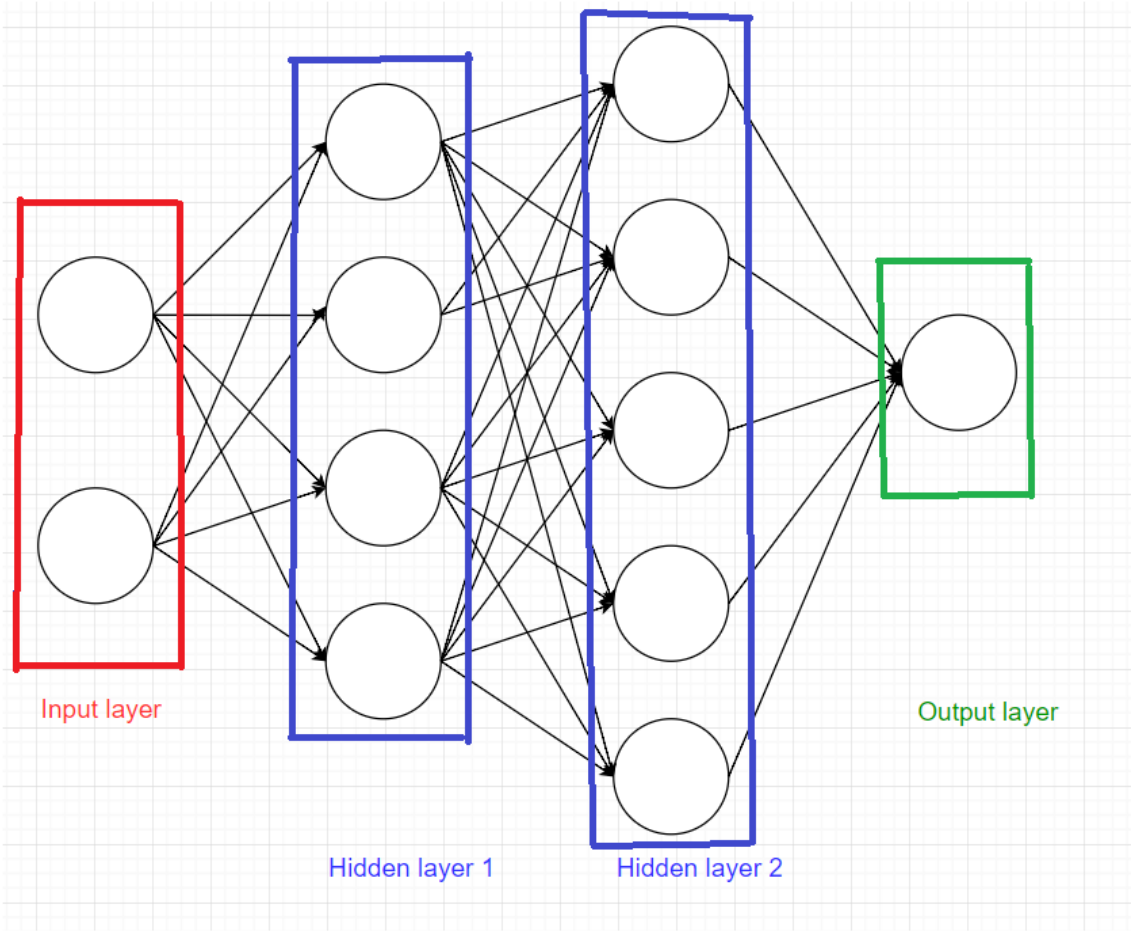
Hệ số w_0 được gọi là bias

7.6. Mạng neural nhân tạo

- Layer đầu tiên là input layer
- các layer ở giữa được gọi là hidden layer
- Layer cuối cùng gọi là output layer
- Các hình tròn được gọi là node
- Mỗi mô hình luôn có 1 input layer, 1 output layer, có thể có hoặc không có các hidden layer.
- Tổng số layer trong mô hình được quy ước là số layer-1(không tính input layer)
-

7.6. Mạng neural nhân tạo

- 1 input layer
- 2 hidden layer
- 1 output layer
- Số lượng layer của mô hình là 3 layer

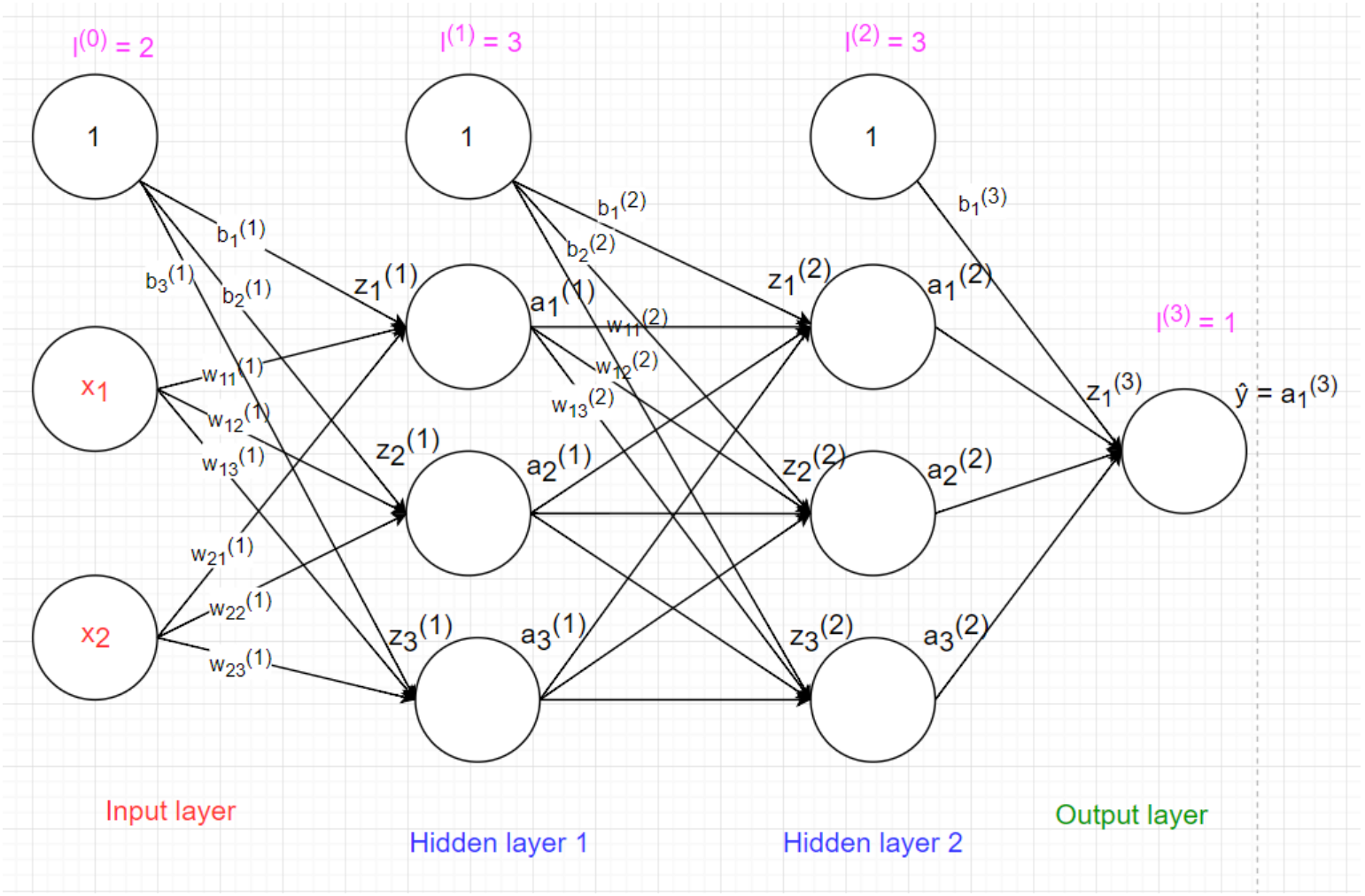


7.6. Mạng neural nhân tạo

Mỗi node trong hidden layer và output layer:

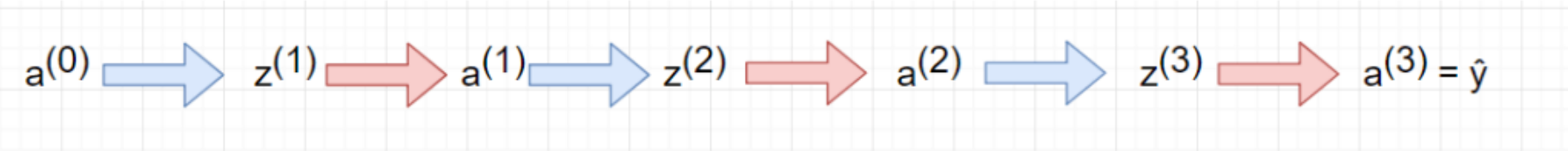
- Liên kết với tất cả các node ở layer trước đó với các hệ số w riêng
- Mỗi nút có 1 hệ số bias b riêng
- Diễn ra 2 bước: tính tổng linear và áp dụng activation function.

7.6. Mạng neural nhân tạo

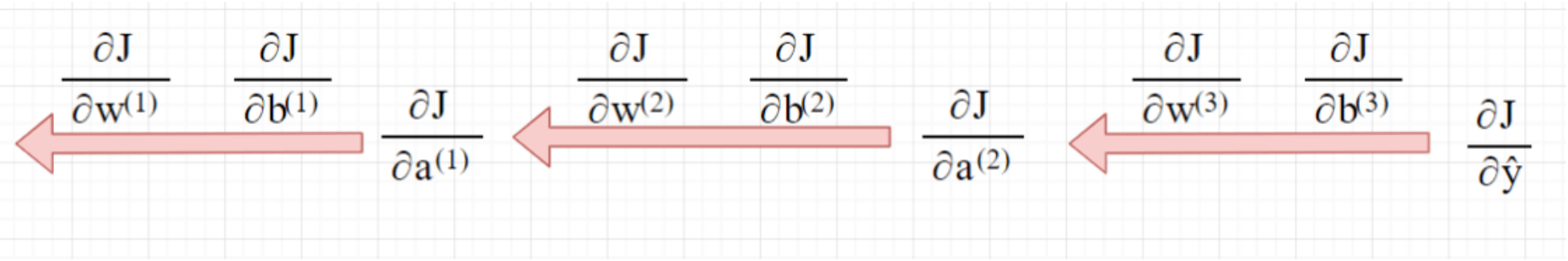


7.6. Mạng neural nhân tạo

- Quá trình feedforward



- Quá trình backpropagation



7.6. Mạng neural nhân tạo

- Một trong những lợi thế của NN là khả năng được sử dụng như một xấp xỉ hàm tùy ý học từ các mẫu dữ liệu quan sát được.
- Một số yêu cầu chính trong xây dựng mô hình mạng neural là vấn đề chọn mô hình phù hợp bài toán ứng dụng phụ thuộc vào cách mô tả mẫu dữ liệu. Nghĩa là một mô hình quá phức tạp sẽ dẫn đến khó khăn trong quá trình huấn luyện mô hình, trong khi đó những mô hình đơn giản lại không giải quyết được bài toán phức tạp. Vấn đề lựa chọn thuật toán học cũng có ý nghĩa hết sức quan trọng.

7.6. Mạng neural nhân tạo

- Thuật toán học chủ yếu liên quan đến việc xây dựng bộ tham số (hyperparameter) và làm sao ước lượng các tham số tối ưu và hội tụ nhanh nhất có thể để phân lớp tốt nhất
- Trong thực tế, lựa chọn và điều chỉnh một thuật toán để huấn luyện trên dữ liệu quan sát yêu cầu một số lượng lớn đáng kể trong thực nghiệm