

DEEP LEARNING

Perceptron đa tầng (MLP)

Phạm Nguyên Khang
pnkhang@cit.ctu.edu.vn

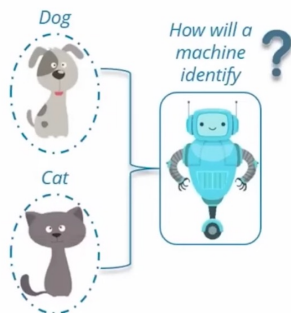
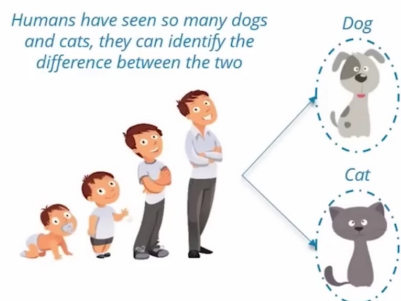
CAN THO, 22/12/2022

Nội dung

- Học sâu là gì?
- Học sâu hoạt động như thế nào?
- Perceptron đơn tầng
- Hạn chế của Perceptron đơn tầng
- Perceptron đa tầng (MLP)

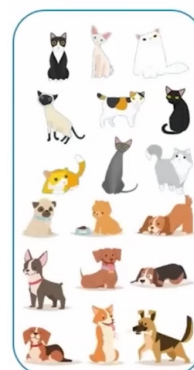
1

Học sâu là gì?

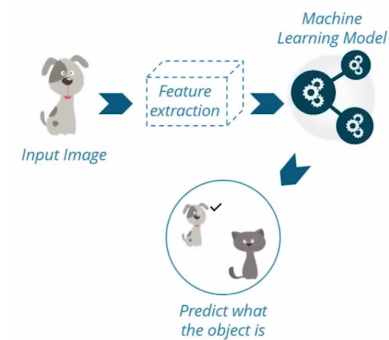
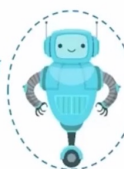


2

Học sâu là gì?



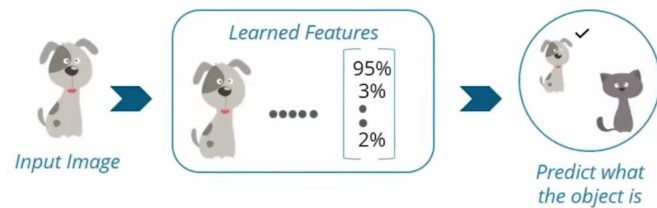
Training



3

Học sâu là gì?

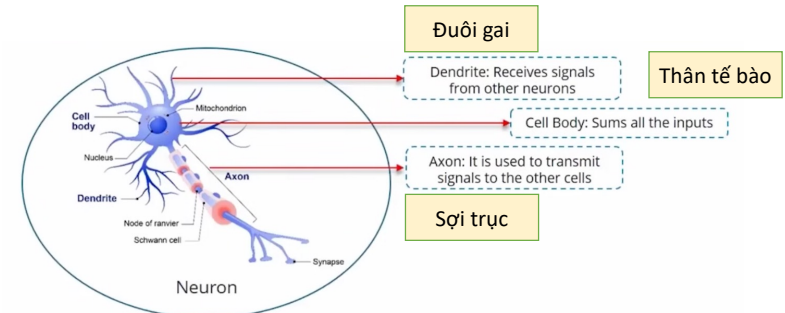
- Học sâu bỏ qua bước trích đặc trưng thủ công (feature extraction), ảnh đầu vào được đưa trực tiếp vào thuật toán/mô hình học sâu, để dự báo đối tượng là gì.



4

Học sâu hoạt động như thế nào?

- Học sâu là một dạng của học máy (machine learning), sử dụng một mô hình tính toán dựa trên bộ não của con người

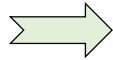


5

Tại sao ta cần nơ rơn nhân tạo?

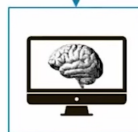
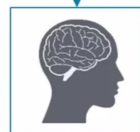
Giả sử ta có tập dữ liệu về các loài hoa, bao gồm:

- Sepal length
- Sepal width
- Petal length
- Petal width



Ta muốn phân loại các loài hoa dựa trên tập dữ liệu này.

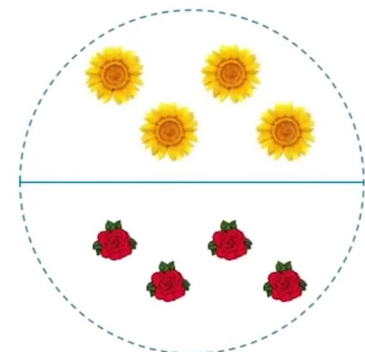
Ta có thể làm bằng 1 trong 2 cách:



6

Tại sao ta cần nơ rơn nhân tạo?

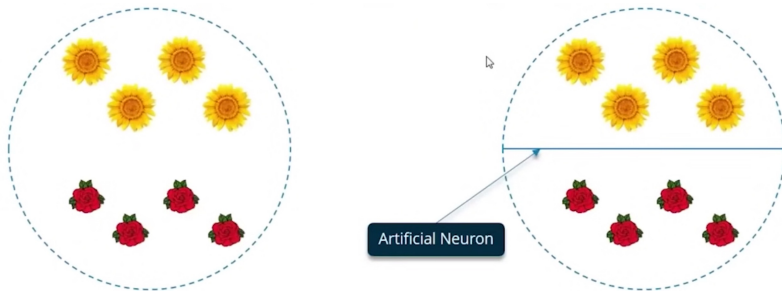
- Phân loại hoa bằng AI



7

Tại sao ta cần nơ ron nhân tạo?

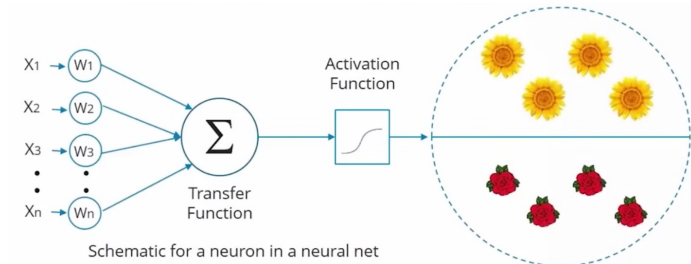
- Hệ thống AI phân loại 2 loài hoa
 - Nhờ vào 1 nơ ron nhân tạo, ta có thể phân loại 2 loài hoa



8

Perceptron

- Perceptron là một loại nơ ron nhân tạo có mô hình tính toán gồm:
 - Hàm truyền (transfer function), thường là tổng
 - Hàm kích hoạt (activation function): ngưỡng, sigmoid

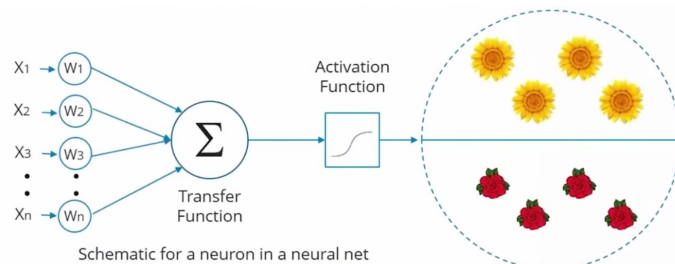


9

Perceptron

- x_j : đầu vào
- w_j : các trọng số
- f : hàm kích hoạt

$$\hat{y} = f\left(\sum_{j=1}^n x_j w_j\right)$$

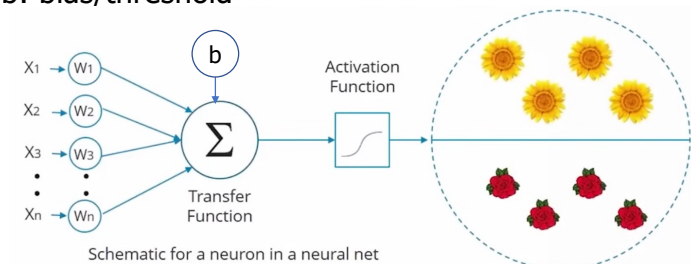


10

Perceptron

- x_j : đầu vào
- w_j : các trọng số
- f : hàm kích hoạt
- b : bias/threshold

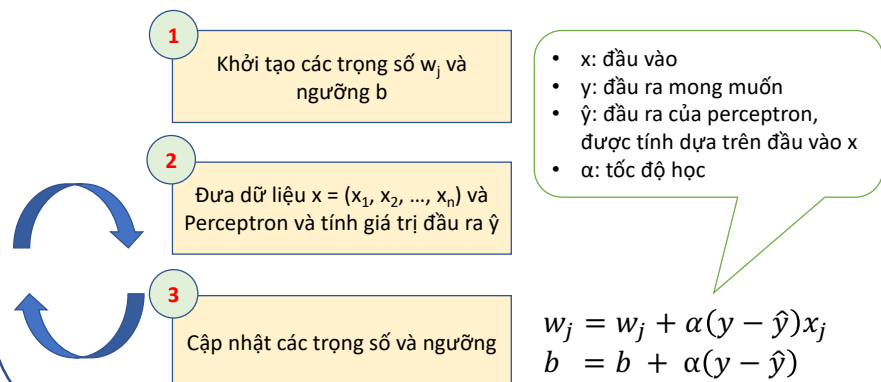
$$\hat{y} = f\left(\sum_{j=1}^n x_j w_j + b\right)$$



11

Perceptron

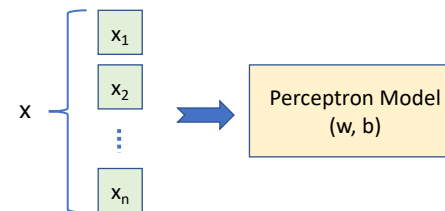
Thuật toán huấn luyện Perceptron



12

Perceptron

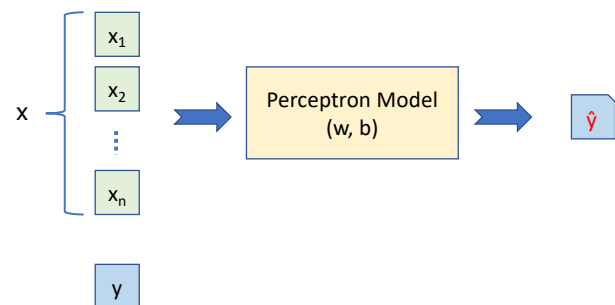
Tổng quát hoá



13

Perceptron

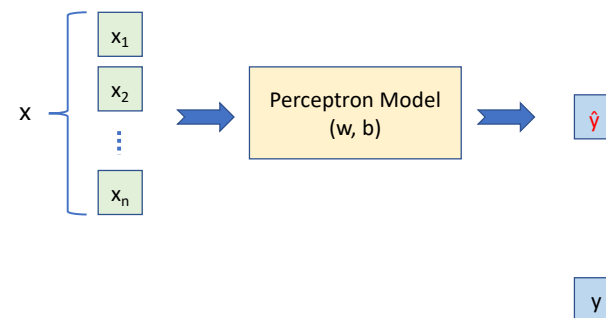
Tổng quát hoá



14

Perceptron

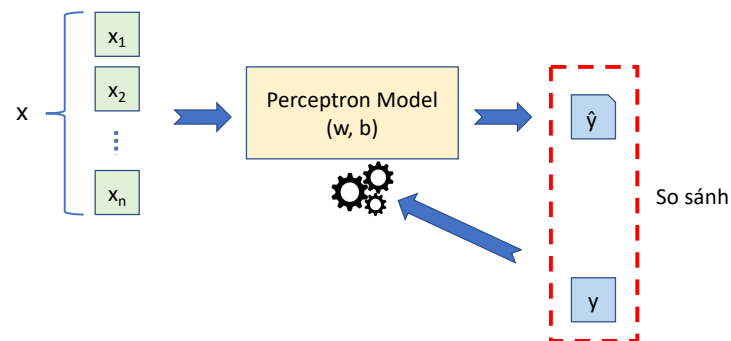
Tổng quát hoá



15

Perceptron

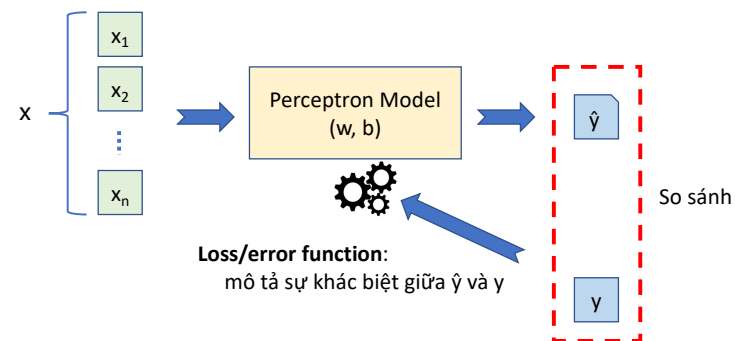
- Tổng quát hoá



16

Perceptron

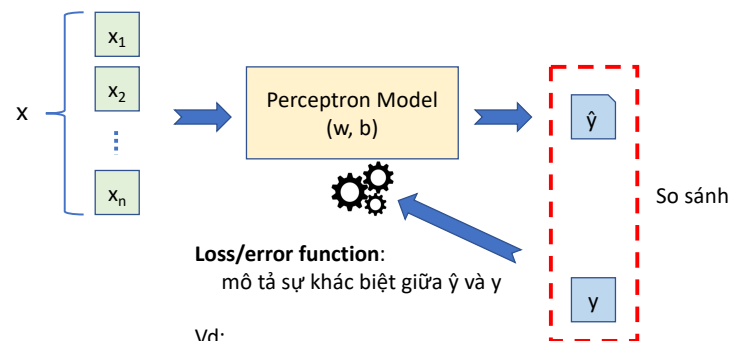
- Tổng quát hoá



17

Perceptron

- Tổng quát hoá



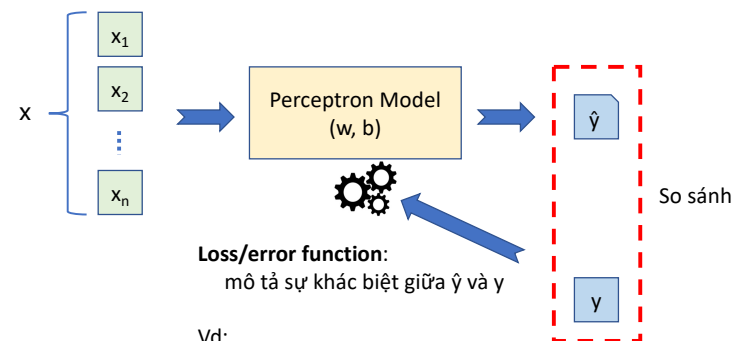
Vd:

$$L = \frac{1}{2}(y - \hat{y})^2$$

18

Perceptron

- Tổng quát hoá



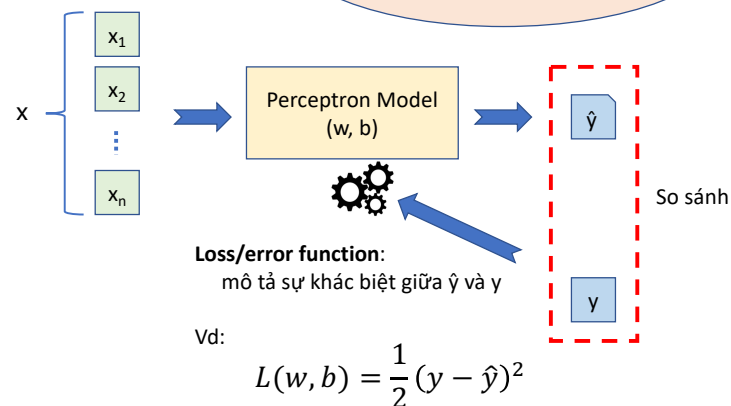
Vd:

$$L(w, b) = \frac{1}{2}(y - \hat{y})^2$$

19

Perceptron

- Tổng quát hoá

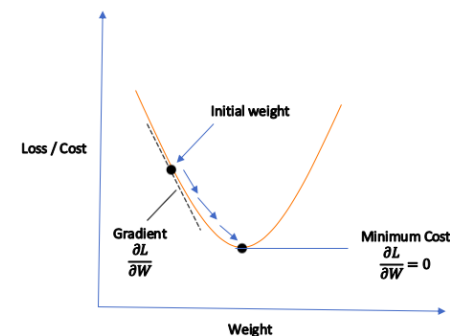


20

Perceptron

- Tìm w, b sao cho L nhỏ nhất
 - Thuật toán giảm gradient

$$L(w, b) = \frac{1}{2} (y - \hat{y})^2$$



21

Perceptron

- Tìm w, b sao cho L nhỏ nhất

- Thuật toán giảm gradient

- Khởi tạo ngẫu nhiên w, b

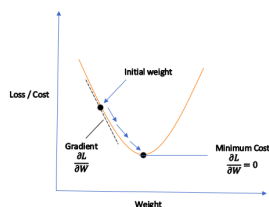
- Lặp

- Cập nhật w, b theo hướng ngược hướng của gradient L

$$w_j = w_j - \alpha \frac{\partial L}{\partial w_j}$$

$$b = b - \alpha \frac{\partial L}{\partial b}$$

$$L(w, b) = \frac{1}{2} (y - \hat{y})^2$$



22

Perceptron

- Tính đạo hàm riêng

$$L(w, b) = \frac{1}{2} (y - \hat{y})^2$$

$$\frac{\partial L}{\partial w_j} = ?$$

$$\hat{y} = f\left(\sum_{j=1}^n x_j w_j + b\right)$$

$$\frac{\partial L}{\partial b} = ?$$

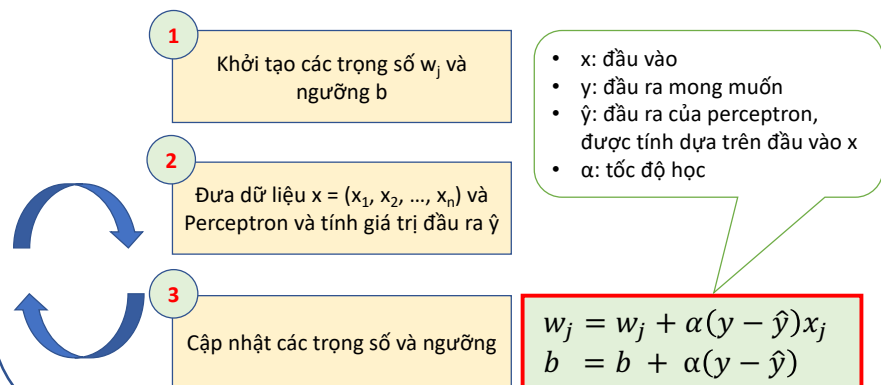
$$f(u) = u$$

$$x_j, y: \text{hằng số}$$

23

Perceptron

- Thuật toán huấn luyện Perceptron (xem lại)



24

Perceptron

- Quy trình
 - Xây dựng mô hình, xác định các tham số
 - Định nghĩa hàm loss/error
 - Tìm tham số sao cho hàm loss nhỏ nhất

25

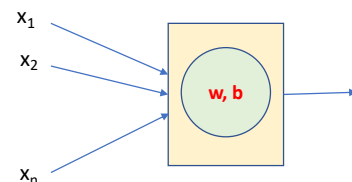
Perceptron

- Quy trình
 - Xây dựng mô hình, xác định các tham số
 - Định nghĩa hàm loss/error
 - Tìm tham số sao cho hàm loss nhỏ nhất
- Cài đặt bằng Keras
 - Tạo mô hình chỉ gồm 1 tầng, 1 nơ ron
 - Liên kết với mô hình với hàm loss và phương pháp tối ưu
 - Thực hiện tối ưu hàm loss

26

Perceptron

- Cài đặt Perceptron bằng Keras
 - Tạo mô hình chỉ gồm 1 tầng, 1 nơ ron
 - Liên kết với mô hình với hàm loss, phương pháp tối ưu và tiêu chí đánh giá
 - Thực hiện tối ưu hàm loss



27

Perceptron

```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

# 1. create model
model = Sequential()
model.add(Dense(1, use_bias=True, input_shape=(n,),
               activation=None))

# 2. compile the model
model.compile(optimizer='adam', loss='mean_squared_error',
              metrics=['accuracy'])

# 3. fit the model
model.fit(X_train, y_train, epochs=150,
          batch_size=32, verbose=0)
```

28

Perceptron

```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

# 1. create model
model = Sequential()
model.add(Dense(1, use_bias=True, input_shape=(n,),
               activation='sigmoid'))

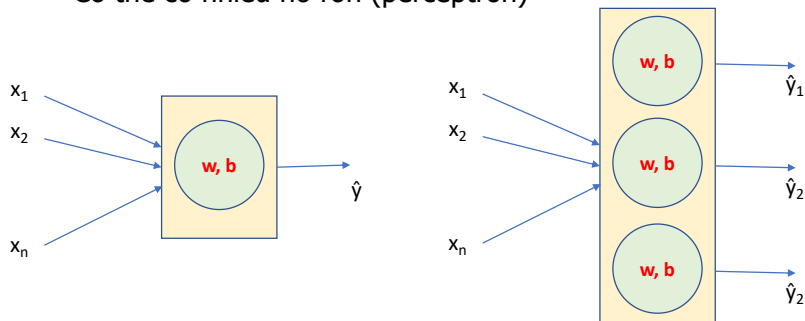
# 2. compile the model
model.compile(optimizer='adam', loss='binary_crossentropy',
              metrics=['accuracy'])

# 3. fit the model
model.fit(X_train, y_train, epochs=150,
          batch_size=32, verbose=0)
```

29

Perceptron đơn tầng (SLP)

- Mạng nơ ron chỉ gồm 1 tầng
 - Có thể có nhiều nơ ron (perceptron)



Phân lớp nhị phân (2 lớp)

Phân lớp đa lớp (> 2 lớp)

30

Thực hành

- Phân loại hoa iris (2 lớp)
 - Download tập dữ liệu iris (<https://archive.ics.uci.edu/ml/machine-learning-databases/iris/>)
 - Giữ lại 100 dòng đầu tiên => lưu vào biến Data
 - Biến đổi nhãn (cột cuối)
 - Iris-setosa => 0
 - Iris-versicolor => 1
 - Xáo trộn dữ liệu Data, tách ra thành train (80%) và test (20%)

31

Thực hành

- Xây dựng mô hình
 - 1 tầng (Dense) gồm 1 nơ ron duy nhất (2 lớp)
 - input_shape: (4,)
 - use_bias: true (mặc định)
 - Activation: sigmoid
- Biên dịch mô hình
 - Optimizer: 'adam'
 - Loss: 'binary_crossentropy'
 - Metrics: ['accuracy']

32

Thực hành

- Xây dựng mô hình
- Biên dịch mô hình
- Huấn luyện mô hình

```
model.fit(train[:, 0:4], train[:, -1],
          epochs=150, batch_size=32, verbose=0)
```
- Đánh giá mô hình

```
score = model.evaluate(test[:, 0:4],
                       test[:, -1], verbose=0)
print("Test loss:", score[0])
print("Test accuracy:", score[1])
```

33

Thực hành

- Phân loại hoa iris (3 lớp)
 - Download tập dữ liệu iris (<https://archive.ics.uci.edu/ml/machine-learning-databases/iris/>)
 - Xử lý toàn bộ tập dữ liệu
 - Biến đổi nhãn
 - Iris-setosa => 0
 - Iris-versicolor => 1
 - Iris-virginica => 2

34

Thực hành

- Xây dựng mô hình
 - 1 tầng (Dense) gồm 3 nơ ron (3 lớp)
 - input_shape: (4,)
 - use_bias: true (mặc định)
 - Activation: softmax
- Biên dịch mô hình
 - Optimizer: 'adam'
 - Loss: 'sparse_categorical_crossentropy'
 - Metrics: ['accuracy']

35

Tuần tới

- Nhược điểm của Perceptron
- Mạng perceptron đa tầng (MLP)
- Cài đặt mạng perceptron đa tầng bằng Keras
- Behind the scene of Keras
 - Tensorflow
 - Lan truyền ngược (Backpropagation)
 - Tính đạo hàm tự động (Gradients and automatic differentiation)

36

THANK YOU



37