



TRƯỜNG ĐẠI HỌC VINH

Mạng nơ ron truyền thẳng MLP

Phan Anh Phong, PhD.
Vinh University

1

Nội dung



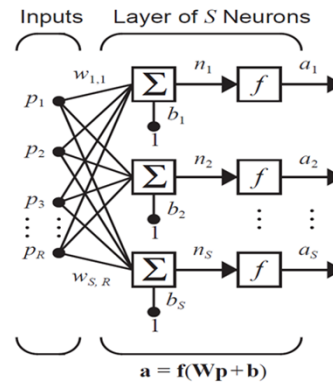
- Giới thiệu mạng nơ ron truyền thẳng
- Mạng MPL
 - Kiến trúc
 - Xử lý thông tin (Quá trình tính toán)
- Triển khai mạng MLP với sklearn và keras
- Ứng dụng của mạng MLP
- Bài tập
- Thảo luận

2

Nhắc lại mạng nơ ron 1 lớp



- Tín hiệu vào (đầu vào)?
- Đầu ra ra?
- Bao nhiêu nơ ron?
- Tham số W_{sR} là gì?
- Tham số b_s là gì?
- Hàm Σ là gì?
- Hàm f là gì?

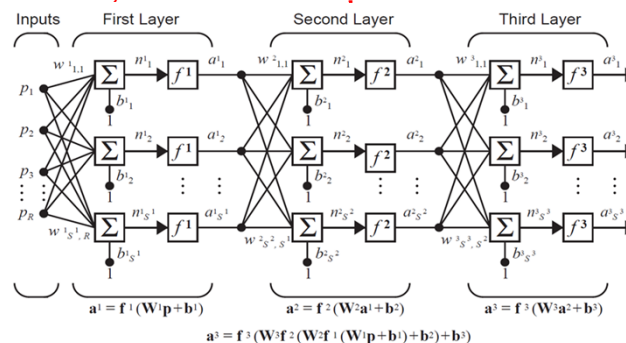


3

Mạng nơ ron nhiều lớp



- Đầu vào mạng? Đầu vào mỗi lớp
- Đầu ra mạng? Đầu ra mỗi lớp
- Bao nhiêu nơ ron?
- Hàm TỔNG, Hàm kích hoạt

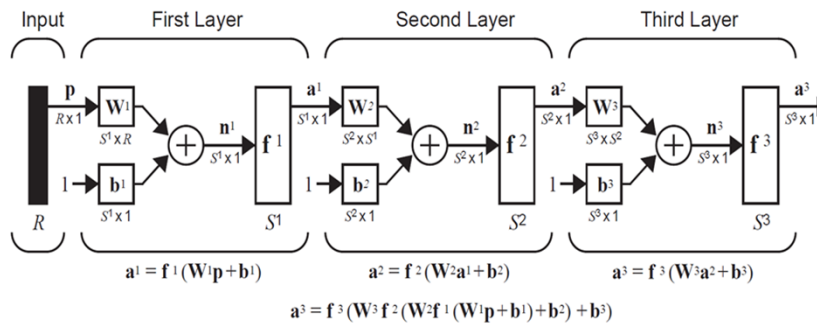


4

Mạng nơ ron nhiều lớp



- Dạng rút gọn (dạng ma trận)



- Mạng nhiều lớp, mỗi lớp có hàm kích hoạt tuyến tính thì vẫn chỉ có thể biểu diễn các hàm tuyến tính
- Để biểu diễn các hàm phi tuyến thì các hàm kích hoạt phải là hàm phi tuyến

5

Mạng nơ ron nhiều lớp



- Các hàm kích hoạt thông dụng

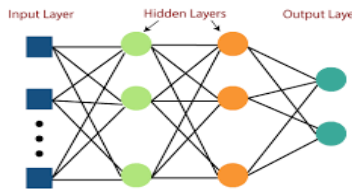
Hàm kích hoạt	Vào/Ra
Step (Hard Limit/Heaviside)	$a = f(n) = \begin{cases} 1, & \text{nếu } n \geq 0 \\ 0, & \text{nếu } n < 0 \end{cases}$
Linear	$a = f(n) = n$
Exponential Linear Unit	$a = f(n) = \begin{cases} n & \text{nếu } n \geq 0 \\ \alpha(e^\alpha - 1) & \text{nếu } n < 0 \end{cases}$
Scaled Exponential Linear Unit	$a = f(n) = \alpha \begin{cases} n & \text{nếu } n \geq 0 \\ \alpha(e^\alpha - 1) & \text{nếu } n < 0 \end{cases}$
Softplus	$a = f(n) = \log_n(1 + e^n)$
Rectified Linear Unit (ReLU)	$a = f(n) = \max(0, n)$
Tanh	$a = f(n) = \frac{2}{1 + e^{-2n}} - 1$
Sigmoid	$a = f(n) = \frac{1}{1 + e^{-n}}$

6

Giới thiệu mạng MLP



- MLP (Multi Layer Perceptron) là mạng nơ-ron nhiều lớp, cơ bản nhất trong Deep Learning.
- Kiến trúc: Mạng MLP gồm 3 lớp chính:
 - Input Layer: Nhận dữ liệu đầu vào.
 - Hidden Layer(s): Xử lý thông tin (có thể nhiều lớp).
 - Output Layer: Trả kết quả.
- Kết nối đầy đủ (Fully Connected) giữa các lớp



7

Ví dụ mạng nơ ron nhiều lớp



- Cho mạng MLP

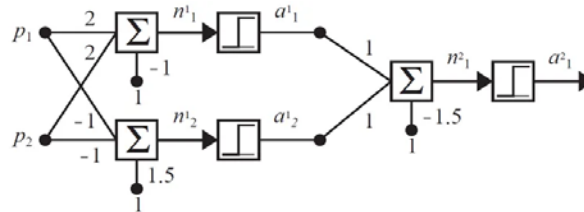
- Lớp ẩn: $\underline{W^1} = \begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix}$, $\underline{b^1} = \begin{bmatrix} -1 \\ 1.5 \end{bmatrix}$ và $\underline{f^1(n)} = \begin{cases} 1 & \text{nếu } n \geq 0, \\ -1 & \text{nếu } n < 0 \end{cases}$
- Lớp ra: $\underline{W^2} = \begin{bmatrix} 1 & 1 \end{bmatrix}$, $\underline{b^2} = \begin{bmatrix} -1.5 \end{bmatrix}$ và $\underline{f^2(n)} = \begin{cases} 1 & \text{nếu } n \geq 0, \\ 0 & \text{nếu } n < 0 \end{cases}$

8

Ví dụ mạng nơ ron nhiều lớp



- Cho mạng MLP



- Lớp ẩn: $W^1 = \begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix}$, $b^1 = \begin{bmatrix} -1 \\ 1.5 \end{bmatrix}$ và $f^1(n) = \begin{cases} 1 & \text{nếu } n \geq 0 \\ -1 & \text{nếu } n < 0 \end{cases}$
- Lớp ra: $W^2 = [1 \ 1]$, $b^2 = [-1.5]$ và $f^2(n) = \begin{cases} 1 & \text{nếu } n \geq 0 \\ 0 & \text{nếu } n < 0 \end{cases}$

9

Ví dụ mạng nơ ron nhiều lớp



- Lớp ẩn: $W^1 = \begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix}$, $b^1 = \begin{bmatrix} -1 \\ 1.5 \end{bmatrix}$ và $f^1(n) = \begin{cases} 1 & \text{nếu } n \geq 0 \\ -1 & \text{nếu } n < 0 \end{cases}$
- Lớp ra: $W^2 = [1 \ 1]$, $b^2 = [-1.5]$ và $f^2(n) = \begin{cases} 1 & \text{nếu } n \geq 0 \\ 0 & \text{nếu } n < 0 \end{cases}$
- Tính đầu ra lớp ẩn với đầu vào $dv1 = [0, 1]^T$
 - $z1 = [n11, n12]^T$
 - $z^1 = W^1 \cdot x + b^1 = \begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \\ 1.5 \end{bmatrix} = \begin{bmatrix} 2 \cdot 1 + (-1) \\ -1 \cdot 1 + 1.5 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$
 - Áp dụng hàm kích hoạt $f^1(n)$: $a^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

10

Ví dụ mạng nơ ron nhiều lớp



- Lớp ẩn: $W^1 = \begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix}$, $b^1 = \begin{bmatrix} -1 \\ 1.5 \end{bmatrix}$ và $f^1(n) = \begin{cases} 1 & \text{nếu } n \geq 0, \\ -1 & \text{nếu } n < 0 \end{cases}$
- Lớp ra: $W^2 = [1 \ 1]$, $b^2 = [-1.5]$ và $f^2(n) = \begin{cases} 1 & \text{nếu } n \geq 0, \\ 0 & \text{nếu } n < 0 \end{cases}$
- Tính đầu ra của lớp ra
 - $a1 = [a11, a12]^T = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
 - $z^2 = W^2 \cdot x + b^2 = [1 \ 1] \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} + [-1.5] = [1.1 + 1.1 + (-1.5)] = 0.5$
 - Áp dụng hàm kích hoạt $f^2(n)$ ta có $a2 = 1$ (vì $n = 0.5$)

11

Ví dụ mạng nơ ron nhiều lớp



- Tính toán tương tự đầu ra của mạng
 - Đầu vào $dv2 = [1, 1]^T$ thì đầu ra là: $a2 = 0$
 - $dv3 = [1, 0]^T$ đầu ra là $a2 = 1$

12

Ví dụ mạng nơ ron nhiều lớp



```
import numpy as np
# Đầu vào
x = np.array([[0], [1]]) # vector cột (2x1)

# Trọng số và bias tầng ẩn
W1 = np.array([[2, 2],
               [-1, -1]])
b1 = np.array([[-1], [1.5]])

# Hàm kích hoạt tầng ẩn: f(n) = 1 nếu n
# >= 0, -1 nếu n < 0
def activation_hidden(n):
    return np.where(n >= 0, 1, -1)

# Trọng số và bias tầng ra
W2 = np.array([[1, 1]])
b2 = np.array([[-1.5]])

# Hàm kích hoạt tầng ra: f(n) = 1 nếu n
# >= 0, 0 nếu n < 0
def activation_output(n):
    return np.where(n >= 0, 1, 0)

# Lan truyền tiến qua tầng ẩn
z1 = np.dot(W1, x) + b1
a1 = activation_hidden(z1)

# Lan truyền tiến qua tầng ra
z2 = np.dot(W2, a1) + b2
a2 = activation_output(z2)

# Kết quả
print("Đầu vào:", x.ravel())
print("Tầng ẩn a1:", a1.ravel())
print("Tầng ra a2 (kết quả):", a2[0][0])
```

13

Bài tập về mạng MLP



- Ví dụ đơn giản về mạng MLP cho bài toán phân lớp nhị phân:
 - Đầu vào: 3 đặc trưng
 - 1 tầng ẩn: 3 nơ-ron, hàm kích hoạt Relu
 - 1 tầng đầu ra: 1 nơ-ron, kích hoạt sigmoid (vì bài toán phân loại nhị phân)
 - Fully Connected

Input (3 nodes)



Dense(hidden): 3 neurons, ReLU



Dense(output): 1 neuron, Sigmoid

14

Bài tập về mạng MLP



- Ví dụ đơn giản về mạng MLP cho bài toán phân lớp nhị phân:
 - Đầu vào: 3 đặc trưng
 - 1 tầng ẩn: 3 nơ-ron, kích hoạt relu
 - 1 tầng đầu ra: 1 nơ-ron, kích hoạt sigmoid (vì bài toán phân loại nhị phân)
 - Đầu vào có mấy giá trị?
 - Trọng số và bias như thế nào? Kích thước ?

15

Bài tập về mạng MLP



- Đầu vào: 3 đặc trưng
- 1 tầng ẩn: 3 nơ-ron, kích hoạt relu
- 1 tầng đầu ra: 1 nơ-ron, kích hoạt sigmoid
- Đầu vào $x = [1.0, 0.5, -1.5]$
- Trọng số và bias tầng ẩn:
 - $W1 = \begin{bmatrix} 0.2 & -0.3 & 0.5 \\ 0.7 & -0.1 & -0.2 \\ -0.6 & 0.9 & 0.3 \end{bmatrix}$
 - $b1 = [0.1, 0.2, 0.0]$
- Trọng số và bias tầng ra:
 - $W2 = [0.4, -0.5, 0.6]$
 - $b2 = 0.3$

W1 khác
W2?

16

Code MLP với Numpy



```
import numpy as np

# 1. Dữ liệu đầu vào (1 mẫu với 3 đặc trưng)
x = np.array([1.0, 0.5, -1.5])

# 2. Trọng số và bias tầng ẩn (3 neuron ẩn)
W1 = np.array([
    [0.2, -0.3, 0.5],
    [0.7, -0.1, -0.2],
    [-0.6, 0.9, 0.3]])
b1 = np.array([0.1, 0.2, 0.0])

# 3. Trọng số và bias tầng đầu ra (1 neuron)
W2 = np.array([0.4, -0.5, 0.6])
b2 = 0.3

# 4. Hàm kích hoạt
def relu(z):
    return np.maximum(0, z)
def sigmoid(z):
    return 1 / (1 + np.exp(-z))

# 5. Lan truyền tiến
z1 = np.dot(W1, x) + b1
a1 = relu(z1)

z2 = np.dot(W2, a1) + b2
a2 = sigmoid(z2)

# 6. In kết quả
print("Đầu vào tầng ẩn (z1):", z1)
print("Đầu ra tầng ẩn (a1 - ReLU):", a1)
print("Đầu vào tầng ra (z2):", z2)
print("Đầu ra tầng ra (a2 - Sigmoid):", a2)
```

17

Code MLP với Keras



```
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.initializers import Constant
import tensorflow as tf

# 1. Dữ liệu đầu vào
x_input = np.array([[1.0, 0.5, -1.5]])

# 2. Trọng số và bias tầng ẩn
W1 = np.array([
    [0.2, 0.7, -0.6], # weights for input 1 to 3 hidden
    [-0.3, -0.1, 0.9], # input 2
    [0.5, -0.2, 0.3] # input 3
]) # shape (3, 3)
b1 = np.array([0.1, 0.2, 0.0])

# 3. Trọng số và bias tầng đầu ra
W2 = np.array([0.4, -0.5, 0.6]) # shape (3, 1)
b2 = np.array([0.3])

# 4. Xây dựng mô hình với Keras
model = Sequential([
    Dense(3, activation='relu',
        input_shape=(3,),
        kernel_initializer=Constant(W1),
        bias_initializer=Constant(b1)),
    Dense(1, activation='sigmoid',
        kernel_initializer=Constant(W2),
        bias_initializer=Constant(b2))
])

# 5. Dự đoán đầu ra
output = model.predict(x_input)
print("Đầu ra của mạng MLP:",
      output[0][0])
```

18

Ứng dụng của mạng nơ ron MLP



- Nhận dạng hình ảnh và phân loại: nhận dạng chữ viết tay (MNIST), phân loại đối tượng trong ảnh, và các tác vụ thị giác máy tính cơ bản.
- Xử lý ngôn ngữ tự nhiên: Phân tích cảm xúc trong văn bản, phân loại văn bản.
- Dự đoán chuỗi thời gian: Dự báo giá cổ phiếu, dự báo thời tiết, dự đoán nhu cầu điện năng
- Phân tích tài chính: Đánh giá rủi ro tín dụng, phát hiện gian lận, dự đoán xu hướng thị trường.

19

Ứng dụng của mạng nơ ron MLP



- Y tế và chẩn đoán: Hỗ trợ chẩn đoán bệnh dựa trên triệu chứng, phân tích hình ảnh y tế, dự đoán kết quả điều trị.
- Hệ thống gợi ý: Đề xuất sản phẩm, âm nhạc hoặc phim ảnh dựa trên sở thích người dùng.
- Điều khiển tự động: Trong robot học và hệ thống điều khiển, MLP giúp dự đoán và điều chỉnh các tham số điều khiển.
- Phân tích dữ liệu khoa học
- ...

20

Ứng dụng phân loại Apple vs Orange

