

# HỌC SÂU

PGS. TS. Nguyễn Hoài Nam

Trưởng NCM, Trưởng Lab (MASC)  
Khoa Tự động hóa - Trường Điện Điện tử  
Đại học Bách Khoa Hà Nội

*Email: [nam.nguyenhoai@hust.edu.vn](mailto:nam.nguyenhoai@hust.edu.vn)*

*Website: <https://sites.google.com/view/masc-lab>*

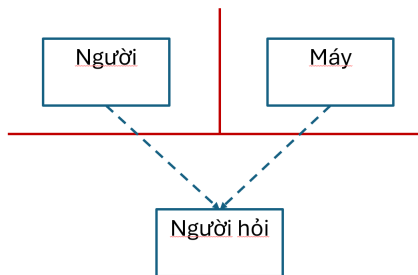
Ngày 24 tháng 9 năm 2025

# Tài liệu tham khảo

1. Ian Goodfellow, Yoshua Bengio, Aaron C. Courville, Deep learning, The MIT Press, 2016.
2. Chris Bishop, Pattern Recognition and Machine Learning, Springer, 2006
3. Martin T. Hagan and et al., Neural Network Design, second ed.
4. Nguyễn Hoài Nam (chủ biên), Nguyễn Thu Hà, "Giáo trình Điều khiển mờ và mạng nơ-ron. Tập I: Cơ sở hệ mờ và mạng nơ-ron", Nhà xuất bản Bách Khoa Hà Nội, 2021.
5. Mark Hudson Beale, Martin T. Hagan, Howard B. Demuth, Deep Learning Toolbox User's Guide.

# 1.1 Trí tuệ nhân tạo và học sâu

- (Artificial Intelligence, 1956 Dartmouth Conference, John McCarthy, Alan Turing).
- Deep Blue (siêu máy tính IBM) đánh bại nhà vô địch cờ thế giới Garry Kasparov 1997.
- Watson (IBM) thắng Brad Rutter và Ken Jennings trong trò chơi truyền hình Jeopardy năm 2011 (1 triệu đô la).



Hình 1.1: Bài thử Turing.

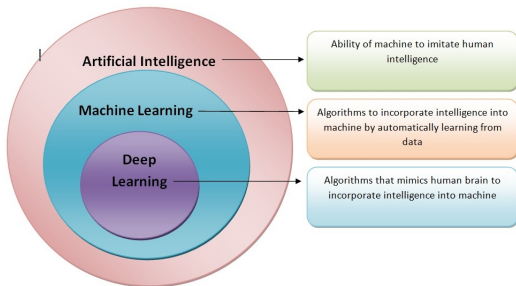
# 1.1 Trí tuệ nhân tạo và học sâu<sup>1</sup>

- Trí tuệ (Intelligence): Khả năng học và thực hiện các kỹ thuật phù hợp để giải quyết các vấn đề và đạt được các mục tiêu, phù hợp với môi trường trong một thế giới thay đổi bất định.
- Trí tuệ nhân tạo (AI, John McCarthy, 1955): Khoa học và kỹ thuật tạo ra các máy thông minh; Khả năng của máy bắt chước sự thông minh của con người.
- Học máy (Machine learning): Một lĩnh vực của AI, nghiên cứu làm thế nào để các chương trình máy tính cải thiện được sự nhận thức, kiến thức, suy nghĩ hoặc các tác động dựa trên kinh nghiệm hoặc dữ liệu; các thuật toán kết hợp sự thông minh vào máy bằng cách tự học từ dữ liệu.
- Học sâu (Deep learning): Sử dụng mạng nơ-ron nhiều lớp lớn giống như những nơ-ron được tổ chức trong bộ não người. Là hướng tiếp cận ML thành công nhất, sử dụng được cho các loại ML, với tính khái quát tốt hơn từ dữ liệu nhỏ và tăng quy mô tốt hơn với dữ liệu lớn.

---

<sup>1</sup>Christopher Manning, Đại học Stanford, 9/2020.

# 1.1 Trí tuệ nhân tạo và học sâu



Hình 1.2: AI, ML và DL<sup>2</sup>.

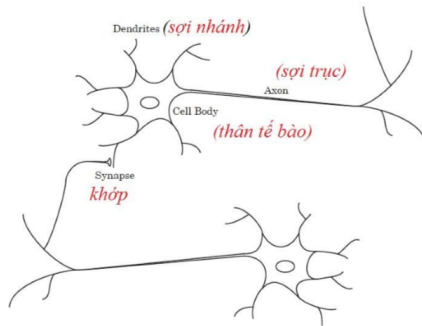
<sup>2</sup><https://www.analyticsvidhya.com/blog/2021/06/machine-learning-vs-artificial-intelligence-vs-deep-learning/>

## 1.1 Trí tuệ nhân tạo và học sâu

Học máy ML	Học sâu DL
Cây quyết định (decision trees), SVM (Support Vector Machine) Rừng cây ngẫu nhiên (Random Forest).	Mạng nơ-ron nhiều lớp (CNN, LSTM)
Tự động học từ dữ liệu	Tự động học và trích xuất đặc trưng
Nhiệm vụ phân loại, xấp xỉ	Nhận dạng ảnh, xử lý ngôn ngữ tự nhiên
Thời gian huấn luyện ngắn	Thời gian huấn luyện lớn
Gradient, Newton, LM, GA ...	Gradient, Adam, RMS
	Học chuyển đổi (transfer learning)

Bảng 1.1: ML và DL.

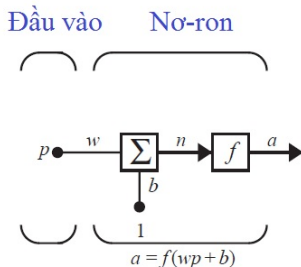
## 1.2 Mô hình nơ-ron và mạng nơ-ron nhân tạo



Hình 1.3: Nơ-ron sinh học

- $10^{11} - 10^{14}$  tế bào thần kinh, mỗi nơ-ron liên kết  $10^4$  tế bào thần kinh khác

## 1.2 Mô hình nơ-ron và mạng nơ-ron nhân tạo

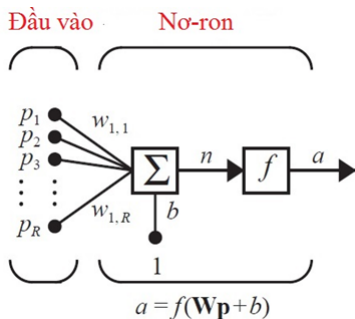


Hình 1.4: Một nơ-ron nhân tạo.

- $w$  trọng số (tham số),  $b$  ngưỡng (độ lệch)  $p$  đầu vào,  $n$  đầu vào tổng,  $f$  hàm truyền,  $a$  đầu ra của nơ-ron



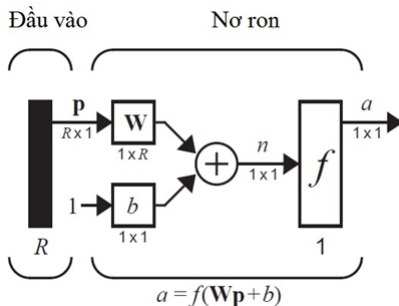
## 1.2 Mô hình nơ-ron và mạng nơ-ron nhân tạo



Hình 1.5: Một nơ-ron nhân tạo có nhiều đầu vào.

$$n = \sum_{i=1}^R w_{1,i} p_i + b, \quad a = f(n) \quad (1.1)$$

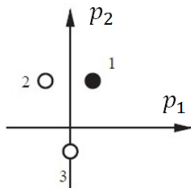
## 1.2 Mô hình nơ-ron và mạng nơ-ron nhân tạo



Hình 1.6: Sơ đồ rút gọn một nơ-ron

$$\begin{aligned}\mathbf{p} &= [p_1 \quad p_2 \quad \dots p_i \dots p_R]^T \\ \mathbf{W} &= [w_{1,1} \quad w_{1,2} \quad \dots w_{1,i} \dots w_{1,R}] \\ n &= \mathbf{W}\mathbf{p} + b, \quad a = f(n)\end{aligned}\tag{1.2}$$

## 1.2 Mô hình nơ-ron và mạng nơ-ron nhân tạo



Hình 1.7: Phân loại véc tơ đầu vào.

- Ví dụ: Phân loại véc tơ đầu vào thành hai loại.
- Luật học perceptron

$$\begin{aligned}W_{new} &= W_{old} + e\mathbf{p}^T \\ b_{new} &= b_{old} + e\end{aligned}\tag{1.3}$$

$e = t - a$  sai lệch giữa đầu ra mẫu  $t$  và đầu ra của nơ-ron  $a$ .

# Một số vấn đề về quá trình học của mạng nơ-ron

- Tập mẫu  $\{u_i, y_i\}$  (training 75%, test 15%, validation 15%)
- Hàm mục tiêu (Loss, cost function)

$$F(\underline{x}) = \frac{1}{Q} \sum_{i=1}^Q e_i^2, \quad e_i = y_i - a_i(\underline{x}, u_i), \quad \text{MSE} \quad (1.4)$$

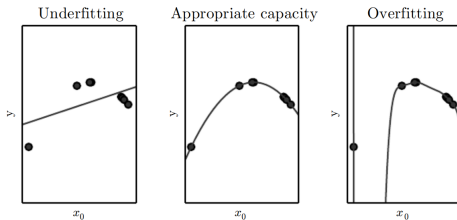
$a_i$  đầu ra của mô hình,  $\underline{x}$  véc tơ tham số của mô hình.

- Thuật toán học (training/learning method, learning rule): Gradient, Newton, GA ...

$$\underline{x}_{k+1} = \underline{x}_k - \alpha \frac{\partial F}{\partial \underline{x}} \Big|_{\underline{x}=\underline{x}_k}, \quad \alpha > 0, k = 0, 1, 2, \dots \quad (1.5)$$

# Một số vấn đề về quá trình học của mạng nơ-ron

- Quá khớp và chưa khớp (Overfitting/Underfitting)
- Gradient bằng không (Gradient vanishing): Do mạng nhiều lớp, gradient của các lớp đầu vào rất nhỏ hoặc nơ-ron bị bão hòa.



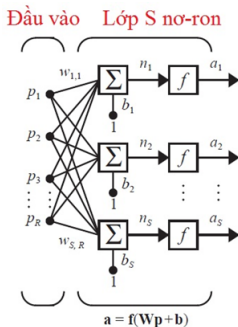
Hình 1.8: Ví dụ về hiện tượng quá khớp và chưa khớp<sup>3</sup>

<sup>3</sup>I. Goodfellow, Y. Bengio A. Courville, Deep Learning.

# Chương 2 Mạng nơ-ron đa tầng MLP

- 2.1 Khái niệm về mạng nơ-ron nhân tạo.
- 2.2 Các thành phần của mạng nơ-ron nhân tạo.
- 2.3 Thuật toán học của mạng MLP.
- 2.4 Ứng dụng mạng MLP trong nhận dạng ảnh.

## 2.1 Khái niệm về mạng nơ-ron nhân tạo.



Hình 2.1: Mạng nơ-ron một lớp

$S$ : số nơ-ron của lớp,  $R$ : số đầu vào của lớp.

$w_{i,j}$ : trọng số kết nối với đầu vào thứ  $j$  của nơ-ron  $i$ .

$b_i$ : bias của nơ-ron  $i$ .

## 2.1 Khái niệm về mạng nơ-ron nhân tạo.

Ma trận trọng số và véc tơ ngưỡng:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R-1} & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R-1} & w_{2,R} \\ \vdots & \vdots & & \vdots & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R-1} & w_{S,R} \end{bmatrix}_{S \times R} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{S-1} \\ b_S \end{bmatrix}_{S \times 1} \quad (2.1)$$

$$\mathbf{p} = [p_1 \quad p_2 \quad \dots \quad p_R]^T \quad \text{véc tơ đầu vào}$$

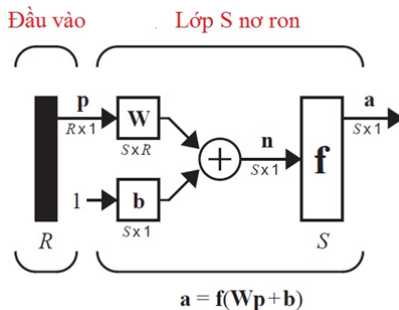
$$\mathbf{n} = \mathbf{W}\mathbf{p} + \mathbf{b} \quad \text{véc tơ đầu vào tổng} \quad (2.2)$$

$$\mathbf{a} = \mathbf{f}(\mathbf{n}) \quad \text{đầu ra của lớp}$$



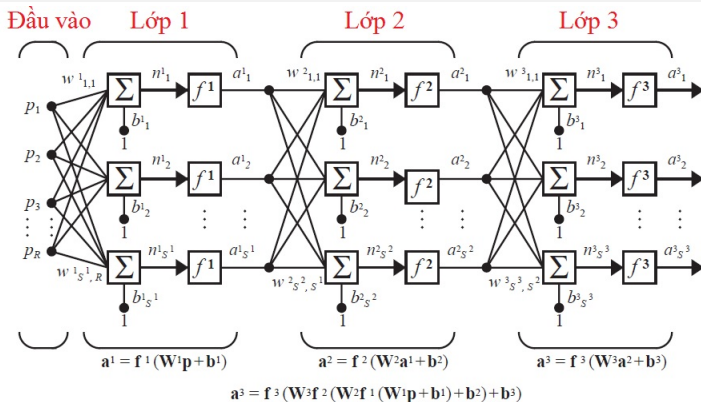
## 2.1 Khái niệm về mạng nơ-ron nhân tạo.

Từ hình 2.1, ta có sơ đồ khối rút gọn (tương đương):



Hình 2.2: Sơ đồ khối mạng một lớp.

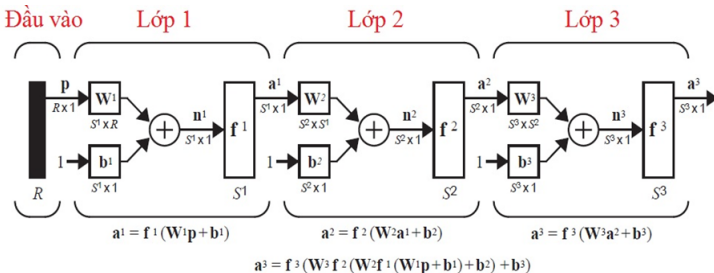
## 2.1 Khái niệm về mạng nơ-ron nhân tạo.



Hình 2.3: Sơ đồ mạng nhiều lớp.

$w^m_{i,j}$ : trọng số liên kết giữa đầu vào thứ  $j$  với nơ-ron  $i$  của lớp  $m$ ,  $m = 1 \div 3$ .  
 $b^m_i$ : bias nơ-ron  $i$  của lớp  $m$ ;  $S^m$ : số nơ-ron của lớp  $m$ .

## 2.1 Khái niệm về mạng nơ-ron nhân tạo.



Hình 2.4: Sơ đồ khối mạng nhiều lớp.

$\mathbf{p}$ : véc tơ đầu vào hay lớp đầu vào

Lớp 1, 2: lớp ẩn.

Lớp 3: lớp đầu ra của mạng.

## 2.1 Khái niệm về mạng nơ-ron nhân tạo.

Vai trò của lớp ẩn:

- Học các dạng mẫu phức tạp trong dữ liệu.
- Chuyển dữ liệu đầu vào thành dạng mới để học hơn cho mạng nơ-ron.

## 2.2. Các thành phần của mạng nơ-ron nhân tạo.

2.2.1 Đầu vào và chuẩn hóa đầu vào.

2.2.2 Mối liên hệ giữa đầu ra và đầu vào của lớp kế tiếp

2.2.3 Hàm kích hoạt

2.2.4 Đầu ra mạng nơ-ron cho ứng dụng xấp xỉ và phân loại

## 2.2. Các thành phần của mạng nơ-ron nhân tạo.

### 2.2.1 Đầu vào và chuẩn hóa đầu vào.

- a) Đầu vào của mạng  $\mathbf{p}$ .
- b) Đầu vào của lớp  $m + 1$  trong mạng:  $\mathbf{a}^m$ ,  $m = 1 \div M - 1$ ,  $M$  là số lớp
- c) Chuẩn hóa đầu vào:

$$y_i = \frac{u_i - u_i^{\min}}{u_i^{\max} - u_i^{\min}} \quad (2.3)$$

$u_i$ : tín hiệu cần chuẩn hóa  $u_i^{\min} \leq u_i \leq u_i^{\max}$ .

$y_i$ : tín hiệu sau khi chuẩn hóa  $0 \leq y_i \leq 1$ .

## 2.2. Các thành phần của mạng nơ-ron nhân tạo.

2.2.2 Mỗi liên hệ giữa đầu ra và đầu vào của lớp kế tiếp.

$$\begin{aligned}\mathbf{n}^i &= \mathbf{W}^i \mathbf{a}^{i-1} + \mathbf{b}^i \quad i = 2 \div M - 1 \\ \mathbf{a}^i &= \mathbf{f}^i(\mathbf{n}^i)\end{aligned}\tag{2.4}$$

## 2.2. Các thành phần của mạng nơ-ron nhân tạo.

### 2.2.3 Hàm kích hoạt:

a) Hàm hardlim

$$a = \text{hardlim}(n) = \begin{cases} 1, & \text{nếu } n \geq 0 \\ 0, & \text{nếu } n < 0 \end{cases} \quad (2.5)$$

b) Hàm logsig

$$a = \text{logsig}(n) = \frac{1}{1 + e^{-n}} \quad (2.6)$$

c) Hàm tansig

$$a = \text{tansig}(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (2.7)$$

d) Các hàm khác: purelin, satlins, poslin ...



## 2.2. Các thành phần của mạng nơ-ron nhân tạo.

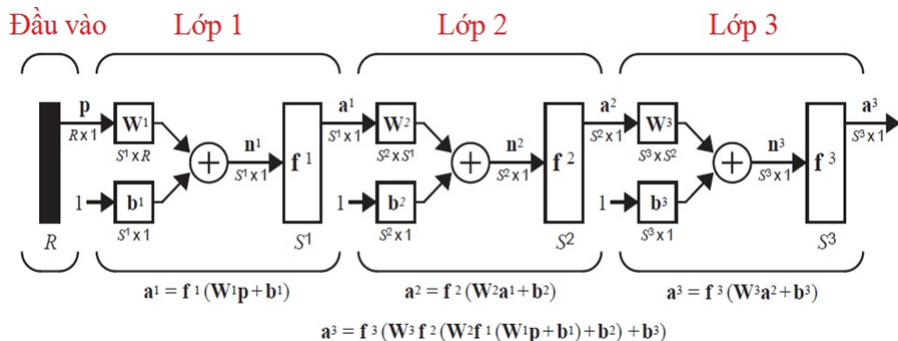
2.2.4. Đầu ra mạng nơ-ron cho ứng dụng xấp xỉ và phân loại. Hàm truyền của lớp đầu ra cho các bài toán:

- a) Xấp xỉ: logsig, tansig, purelin.
- b) Phân loại: hardlim, hardlims, softmax

## 2.3. Thuật toán học của mạng MLP.

- Chu trình thuận
- Thuật toán lan truyền ngược.

## 2.3.1. Chu trình thuận



Hình 2.5: Sơ đồ khối mạng nhiều lớp.

## 2.3.1. Chu trình thuận

- Mạng có  $M$  lớp.
- Đầu vào  $\mathbf{p}_{R \times 1}$ .
- Lớp  $i$  có  $S^i$  nơ-ron.

$$\begin{aligned}\mathbf{n}^i &= \mathbf{W}^i \mathbf{a}^{i-1} + \mathbf{b}^i \quad i = 1 \div M, \mathbf{a}^0 = \mathbf{p} \\ \mathbf{a}^i &= \mathbf{f}^i(\mathbf{n}^i)\end{aligned}\tag{2.8}$$

## 2.3.2. Thuật toán lan truyền ngược.

- Tập mẫu:  $\Omega = \{\mathbf{p}_i; \mathbf{t}_i\}, i = 1 \div Q$ ,  $\mathbf{p}_i$  - đầu vào mẫu và  $\mathbf{t}_i$  - đầu ra mẫu (đích).
- Hàm mục tiêu:

$$J = \frac{1}{Q} \sum_{q=1}^Q J_q \rightarrow \min, \quad (2.9)$$

trong đó  $J_q = \mathbf{e}_q^T \mathbf{e}_q$ ,  $\mathbf{e}_q = \mathbf{t}_q - \mathbf{a}_q^M$ ,  $\mathbf{a}_q^M$  - đầu ra tương ứng với đầu vào  $\mathbf{p}_q$ .

## 2.3.2. Thuật toán lan truyền ngược.

- Phương pháp gradient

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha \frac{\partial J_q}{\partial w_{i,j}^m}, \quad (2.10)$$

$w_{i,j}^m$ : trọng số của nơ ron thứ  $i$  thuộc lớp  $m$ , liên kết với đầu ra của nơ-ron thứ  $j$  thuộc lớp  $m-1$ ;  $k$ : thể hiện giá trị hiện tại;  $k+1$ : thể hiện giá trị mới,  $\alpha > 0$ : tốc độ học.

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial J_q}{\partial b_i^m}, \quad (2.11)$$

$b_i^m$ : ngưỡng của nơ-ron thứ  $i$  thuộc lớp  $m$ .

## 2.3.2. Thuật toán lan truyền ngược.

$$\frac{\partial J_q}{\partial w_{i,j}^m} = \frac{\partial J_q}{\partial n_i^m} \frac{\partial n_i^m}{\partial w_{i,j}^m} \quad (2.12)$$

$$n_i^m = b_i^m + \sum_{v=1}^{S^{m-1}} w_{i,v}^m a_v^{m-1}, \quad (2.13)$$

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1} \quad (2.14)$$

$$\frac{\partial n_i^m}{\partial b_i^m} = 1. \quad (2.15)$$

## 2.3.2. Thuật toán lan truyền ngược.

Độ nhạy của nơ-ron thứ  $i$  thuộc lớp  $m$ :

$$s_i^m = \frac{\partial J_q}{\partial n_i^m} \quad (2.16)$$

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1} \quad (2.17)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m \quad (2.18)$$

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T \quad (2.19)$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m, \quad (2.20)$$

$$\mathbf{s}^m = \frac{\partial J_q}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial J_q}{\partial n_1^m} & \frac{\partial J_q}{\partial n_2^m} & \cdots & \frac{\partial J_q}{\partial n_{S^m}^m} \end{bmatrix}^T, \quad (2.21)$$



## 2.3.2. Thuật toán lan truyền ngược.

$$\begin{aligned}\frac{\partial n_j^{m+1}}{\partial n_j^m} &= \frac{\partial \left( \sum_{l=1}^{s^m} w_{i,l}^{m+1} a_l^m + b_i^{m+1} \right)}{\partial n_j^m} \\ &= w_{i,j}^{m+1} \frac{\partial a_j^m}{\partial n_j^m} \\ &= w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} \\ &= w_{i,j}^{m+1} \dot{f}^m(n_j^m).\end{aligned}$$

## 2.3.2. Thuật toán lan truyền ngược.

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \frac{\partial n_1^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_1^{m+1}}{\partial n_{S^m}^m} \\ \frac{\partial n_2^{m+1}}{\partial n_1^m} & \frac{\partial n_2^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_2^{m+1}}{\partial n_{S^m}^m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_1^m} & \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_{S^m}^m} \end{bmatrix} = \mathbf{W}^{m+1} \dot{\mathbf{F}}^m(\mathbf{n}^m) \quad (2.22)$$

$$\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \cdots & 0 \\ 0 & \dot{f}^m(n_2^m) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \dot{f}^m(n_{S^m}^m) \end{bmatrix}. \quad (2.23)$$

## 2.3.2. Thuật toán lan truyền ngược.

$$\begin{aligned}\mathbf{s}^m &= \frac{\partial J_q}{\partial \mathbf{n}^m} \\ &= \left( \frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \right)^T \frac{\partial J_q}{\partial \mathbf{n}^{m+1}} \\ &= \dot{\mathbf{F}}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \frac{\partial J_q}{\partial \mathbf{n}^{m+1}} \\ &= \dot{\mathbf{F}}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}.\end{aligned}$$

Từ công thức này ta có thể tính được độ nhạy của các lớp lần lượt từ lớp đầu ra ngược trở về lớp đầu vào như sau:

$$\mathbf{s}^M \rightarrow \mathbf{s}^{M-1} \rightarrow \dots \rightarrow \mathbf{s}^2 \rightarrow \mathbf{s}^1. \quad (2.24)$$

## 2.3.2. Thuật toán lan truyền ngược.

$$\begin{aligned} s_i^M &= \frac{\partial J_q}{\partial n_i^M} \\ &= \frac{\partial (\mathbf{t}_q - \mathbf{a}_q)^T (\mathbf{t}_q - \mathbf{a}_q)}{\partial n_i^M} \\ &= \frac{\partial \sum_{j=1}^{S^M} (t_j - a_j)^2}{\partial n_i^M} \\ &= -2(t_i - a_i) \frac{\partial a_i}{\partial n_i^M}. \end{aligned} \tag{2.25}$$

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t}_q - \mathbf{a}_q). \tag{2.26}$$

## 2.3.2. Thuật toán lan truyền ngược.

- Quá trình khởi tạo mạng: Chọn giá trị ban đầu cho các tham số.
- Quá trình lan truyền thuận

$$\mathbf{a}^0 = \mathbf{p}, \quad (2.27)$$

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1} \mathbf{a}^m + \mathbf{b}^{m+1}), \forall m = 0, 1, \dots, M-1. \quad (2.28)$$

$$\mathbf{a} = \mathbf{a}^M. \quad (2.29)$$

- Quá trình lan truyền ngược độ nhạy

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}), \quad (2.30)$$

$$\mathbf{s}^m = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}, \forall m = M-1, \dots, 2, 1. \quad (2.31)$$

- Quá trình cập nhật các tham số của mạng

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T \quad (2.32)$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m \quad (2.33)$$

## Ví dụ

Xấp xỉ hàm số sau

$$y = 1 + \cos\left(\frac{\pi}{4}x\right)$$

với  $-2 \leq x \leq 2$ .

Chọn mạng có cấu trúc gồm 2 lớp, lớp 1 có hàm truyền là *logsig* và lớp 2 có hàm truyền *purelin*.

Đầu vào mẫu thứ  $i$ :

$$x_i = -2 + 4i/(Q - 1), \text{ với } i = 0, 1, 2, \dots, Q - 1. \quad (2.34)$$

Đầu ra mẫu tương ứng với  $x_i$  là:

$$y_i = 1 + \cos\left(\frac{\pi}{4}x_i\right). \quad (2.35)$$

## Ví dụ

Mẫu thử nhất là:  $x_1 = -2, y_1 = 1$ .

- Khởi tạo mạng:  $\mathbf{w}^1 = [1 \quad 1]^T$ ,  $\mathbf{b}^1 = [2 \quad 2]^T$ ,  $\mathbf{w}^2 = [1 \quad 1]$ ,  $b^2 = -1$  và tốc độ học là  $\alpha = 0.1$ .
- Quá trình lan truyền thuận:  $a^0 = -2$ ,

$$\begin{aligned}\mathbf{a}^1 &= \mathbf{f}^1(\mathbf{w}^1 a^0 + \mathbf{b}^1) \\ &= \text{logsig}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} (-2) + \begin{bmatrix} 2 \\ 2 \end{bmatrix}\right) \\ &= \text{logsig}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 0,5 \\ 0,5 \end{bmatrix}\end{aligned}\tag{2.36}$$

$$\begin{aligned}a^2 &= f^2(\mathbf{w}^2 \mathbf{a}^1 + b^2) \\ &= \text{purelin}\left([1 \quad 1] \begin{bmatrix} 0,5 \\ 0,5 \end{bmatrix} - 1\right) \\ &= 0\end{aligned}\tag{2.37}$$

## Ví dụ

- Quá trình lan truyền ngược:

$$a = \text{purelin}(n) \rightarrow \frac{\partial a}{\partial n} = 1, \quad a = \text{logsig}(n) \rightarrow \frac{\partial a}{\partial n} = a(1 - a).$$

Sai lệch là:  $e = y_1 - a^2 = 1$ .

Độ nhảy của lớp 2:

$$\begin{aligned} s^2 &= -2\dot{f}^2(n^2)(t - a^2) \\ &= -2 \end{aligned} \tag{2.38}$$

Độ nhảy của lớp 1:

$$\begin{aligned} \mathbf{s}^1 &= \dot{\mathbf{F}}^2(\mathbf{n}^2)(\mathbf{w}^2)^T \mathbf{s}^2 \\ &= \begin{bmatrix} 0,25 & 0 \\ 0 & 0,25 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} (-2) = \begin{bmatrix} -0,5 \\ -0,5 \end{bmatrix} \end{aligned} \tag{2.39}$$



## Ví dụ

- Quá trình cập nhật các tham số của mạng.

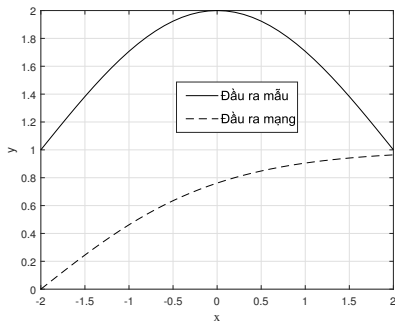
$$\begin{aligned}\mathbf{w}^1(1) &= \mathbf{w}^1(0) - \alpha \mathbf{s}^1 a^0 \\ &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 0.1 \begin{bmatrix} -0,5 \\ -0,5 \end{bmatrix} (-2) = \begin{bmatrix} 0,9 \\ 0,9 \end{bmatrix}\end{aligned}\quad (2.40)$$

$$\begin{aligned}\mathbf{b}^1(1) &= \mathbf{b}^1(0) - \alpha \mathbf{s}^1 \\ &= \begin{bmatrix} 2 \\ 2 \end{bmatrix} - 0.1 \begin{bmatrix} -0,5 \\ -0,5 \end{bmatrix} = \begin{bmatrix} 2,05 \\ 2,05 \end{bmatrix}\end{aligned}\quad (2.41)$$

$$\begin{aligned}\mathbf{w}^2(1) &= \mathbf{w}^2(0) - \alpha \mathbf{s}^2 (\mathbf{a}^1)^T \\ &= \begin{bmatrix} 1 & 1 \end{bmatrix} - 0.1(-2) \begin{bmatrix} 0,5 & 0,5 \end{bmatrix} = \begin{bmatrix} 1,1 & 1,1 \end{bmatrix}\end{aligned}\quad (2.42)$$

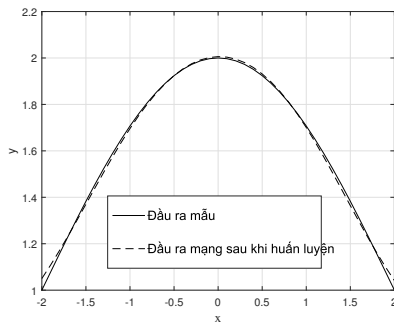
$$\begin{aligned}b^2(1) &= b^2(0) - \alpha s^2 \\ &= -1 - 0.1(-2) \\ &= -0,8\end{aligned}\quad (2.43)$$

# Ví dụ



Hình 2.6: Đầu ra mạng với bộ tham số ban đầu.

# Ví dụ



Hình 2.7: Đầu ra mạng sau khi huấn luyện

## 2.4 Ứng dụng mạng MLP trong nhận dạng ảnh

- 2.4.1. Bài toán nhận dạng ảnh chữ viết tay.
- 2.4.2. Đặc điểm và hạn chế của MLP trong xử lý ảnh.
- 2.4.3. Hạn chế của phương pháp trích chọn đặc trưng bằng tay.

## 2.4.1 Bài toán nhận dạng ảnh chữ viết tay

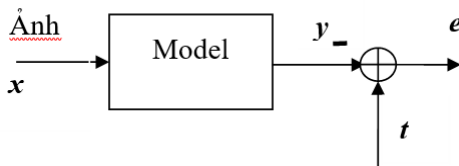


Hình 2.8: Hình ảnh một số chữ viết tay.

## 2.4.1 Bài toán nhận dạng ảnh chữ viết tay

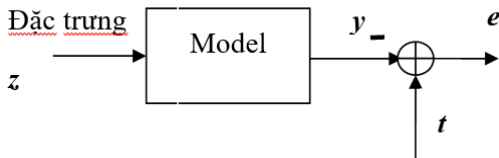
- 26 chữ (a, b, c ...) và 10 số viết tay
- Quét ảnh và tách ký tự
- Loại bỏ nhiễu, chỉnh hướng, kích thước ...
- Chuẩn hóa kích thước ảnh ( $32 \times 32$ )
- Trích xuất đặc trưng (đường biên)
- Huấn luyện mạng 2 lớp

## 2.4.1 Bài toán nhận dạng ảnh chữ viết tay



Hình 2.9: Nhận dạng chữ viết tay.

## 2.4.1 Bài toán nhận dạng ảnh chữ viết tay

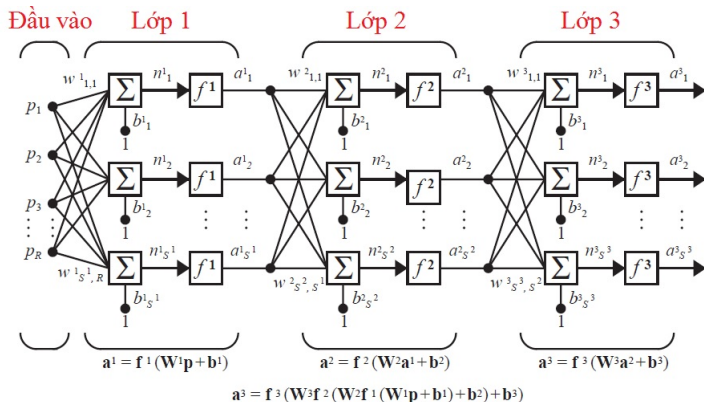


Hình 2.10: Nhận dạng chữ viết tay.



## 2.4.2. Đặc điểm và hạn chế của MLP trong xử lý ảnh.

- Số trọng số ở lớp đầu vào lớn, khối lượng tính toán nhiều.
- Độ chính xác thấp, số lớp ít.



Hình 2.11: Mạng nhiều lớp với kích thước đầu vào  $R$  lớn ( $28 \times 28$ ).

## 2.4.3. Hạn chế của phương pháp trích chọn đặc trưng bằng tay.

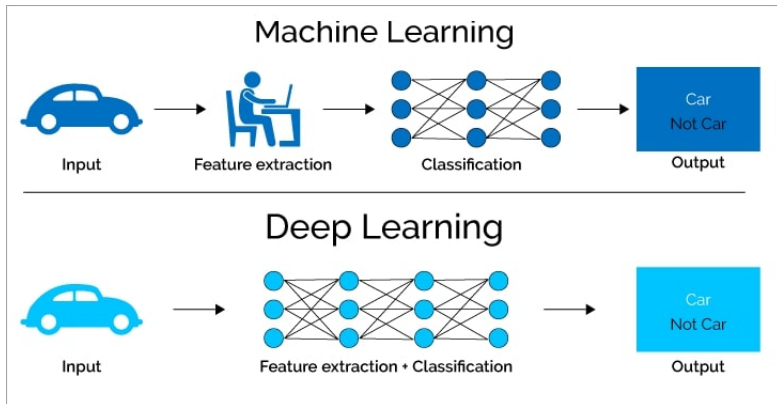
- Trích chọn đặc trưng (Feature Extraction)<sup>4</sup>
  - HOG (Histogram of oriented gradients)<sup>5</sup>
  - SURF (Speeded-up robust features)
  - LBP (Local binary pattern features)
- Dựa vào kinh nghiệm, do người thiết kế (manual).
- Tiềm xử lý dữ liệu.
- Số lượng bộ trích chọn đặc trưng hạn chế.

---

<sup>4</sup>Image Processing Toolbox, Matlab

<sup>5</sup>Dalal, N. and B. Triggs. "Histograms of Oriented Gradients for Human Detection", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.

## 2.4.3. Hạn chế của phương pháp trích chọn đặc trưng bằng tay.



Hình 2.12: Sự khác biệt giữa DL và ML

# Mạng tích chập (Convolutional Neural Networks)

- Trích xuất đặc trưng tự động
- Số tham số ít hơn.
- Các nơ-ron của lớp tích chập có chung bộ trọng số.
- Mỗi nơ-ron của lớp tích chập chỉ liên kết với một số đầu vào thay vì tất cả.