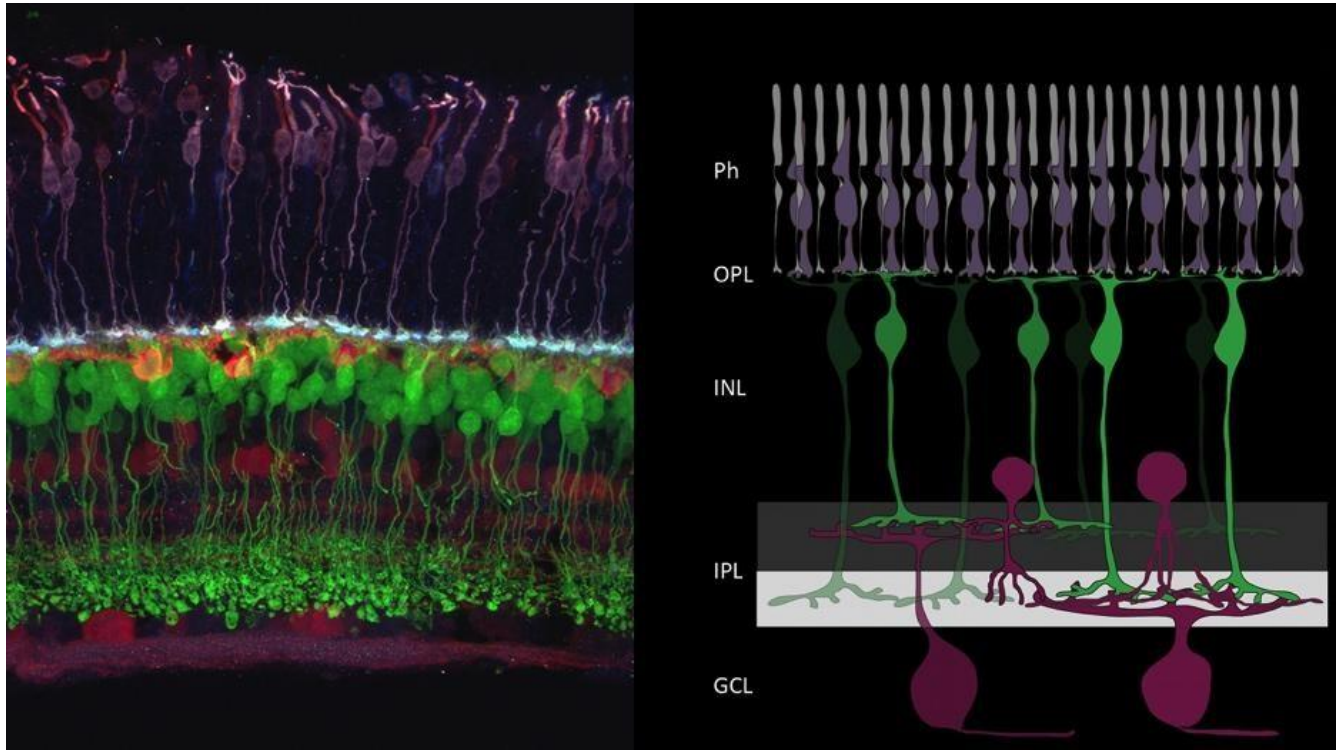
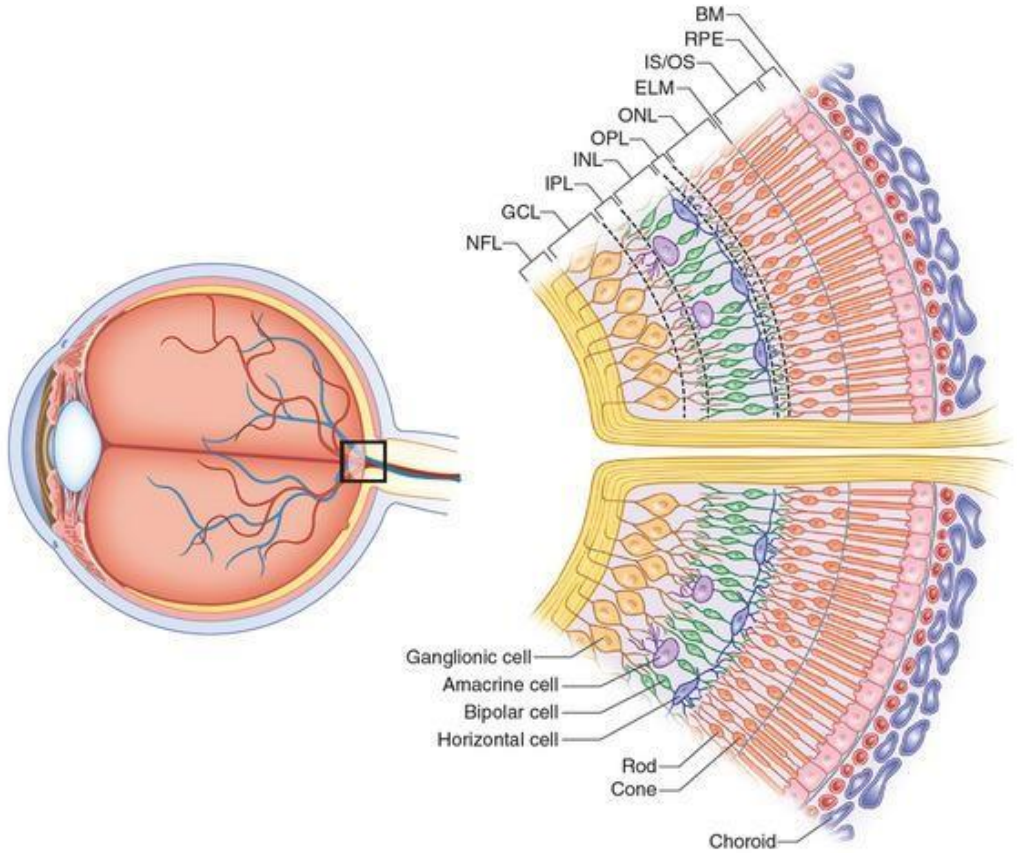
Mạng Nơ-ron nhân tạo
Artificial Neural Networks

Nguyễn Thanh Tùng, **Trần Thị Ngân**
Khoa Công nghệ thông tin – Đại học Thủy lợi
tungnt@tlu.edu.vn, ***ngantt@tlu.edu.vn***

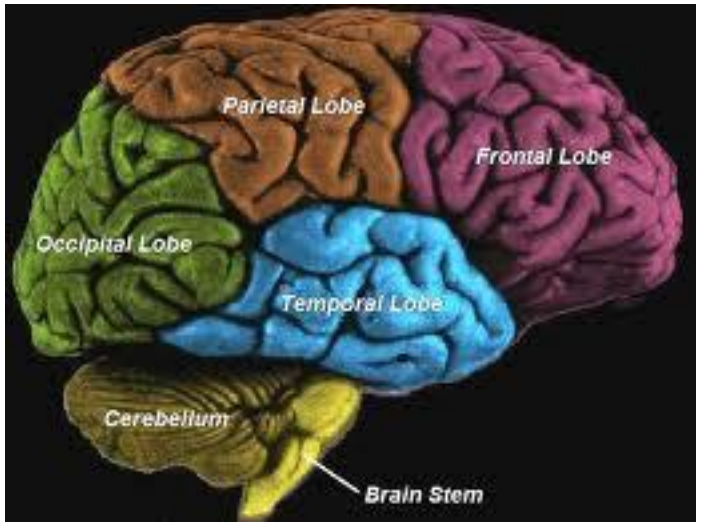
Mạng nơ-ron nhân tạo

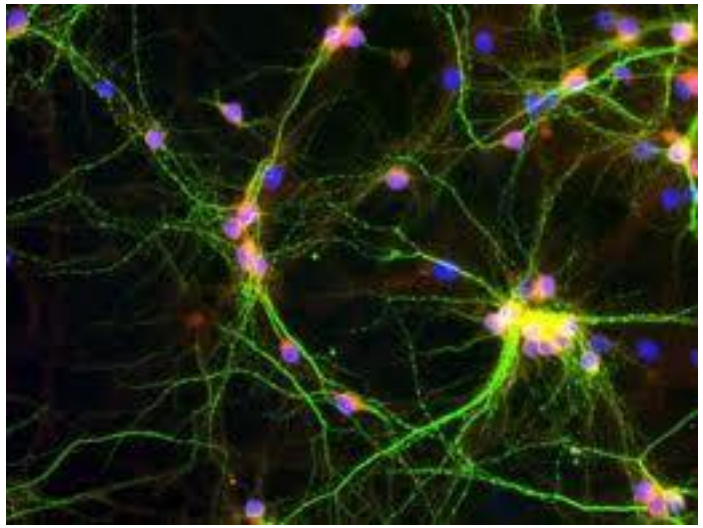


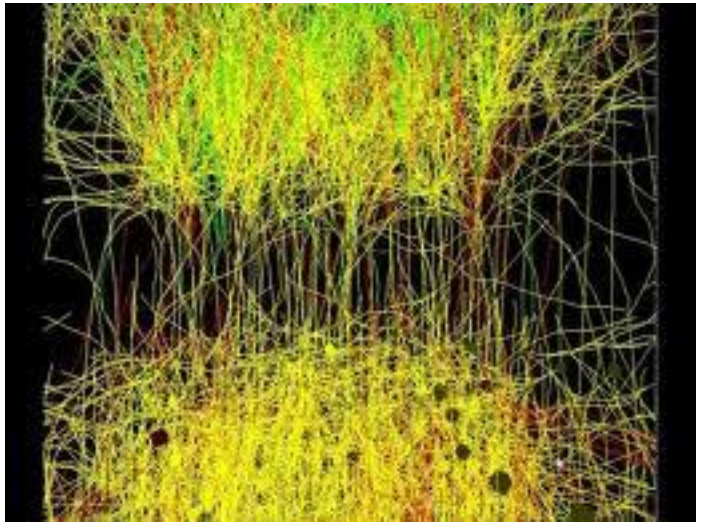


Bleckert A, Schwartz GW, Turner MH, Rieke F, Wong RO. Visual space is represented by nonmatching topographies of distinct mouse retinal ganglion cell types. Curr Biol. 2014 Feb 3;24(3):310-5.

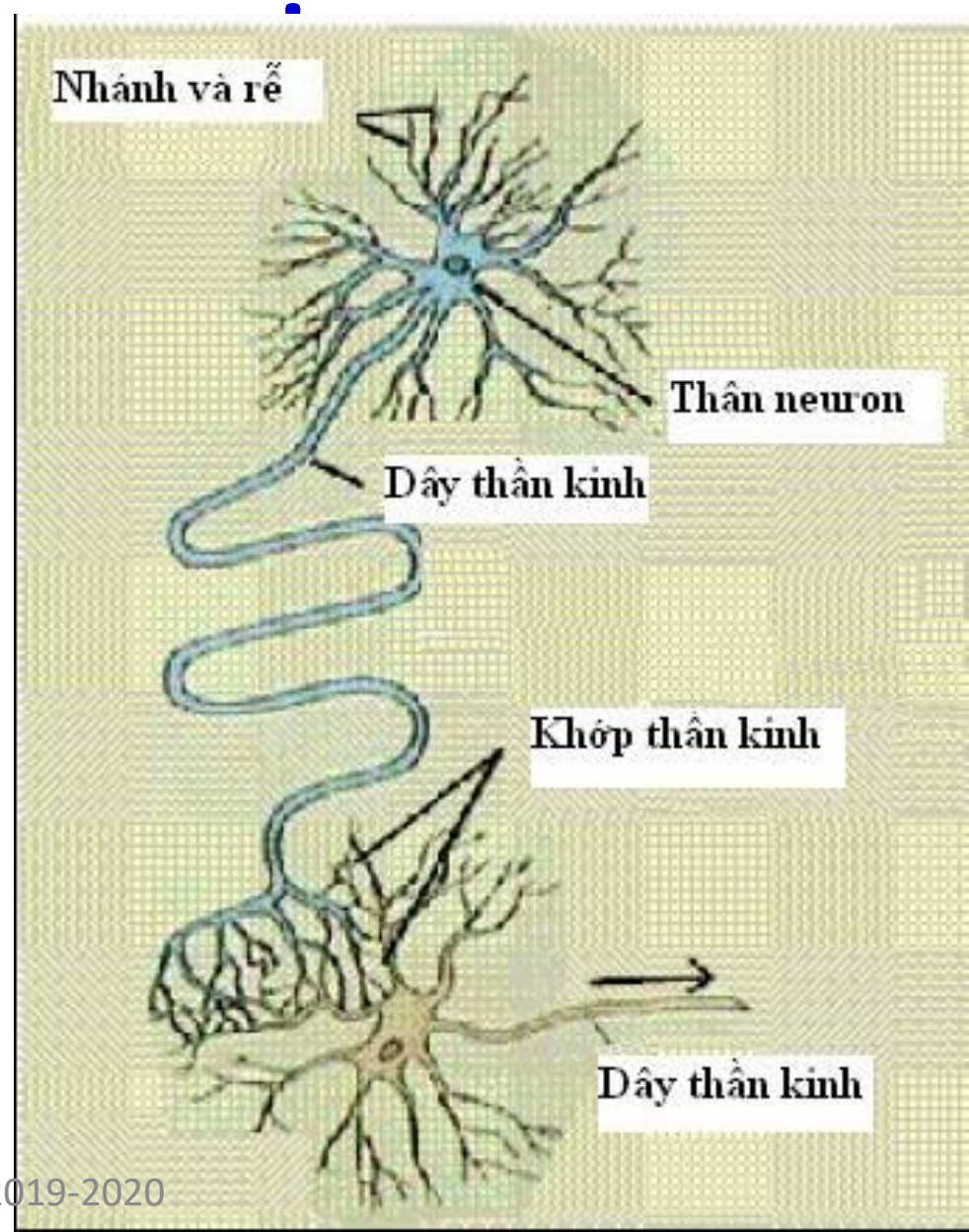
Mạng nơ-ron sinh học



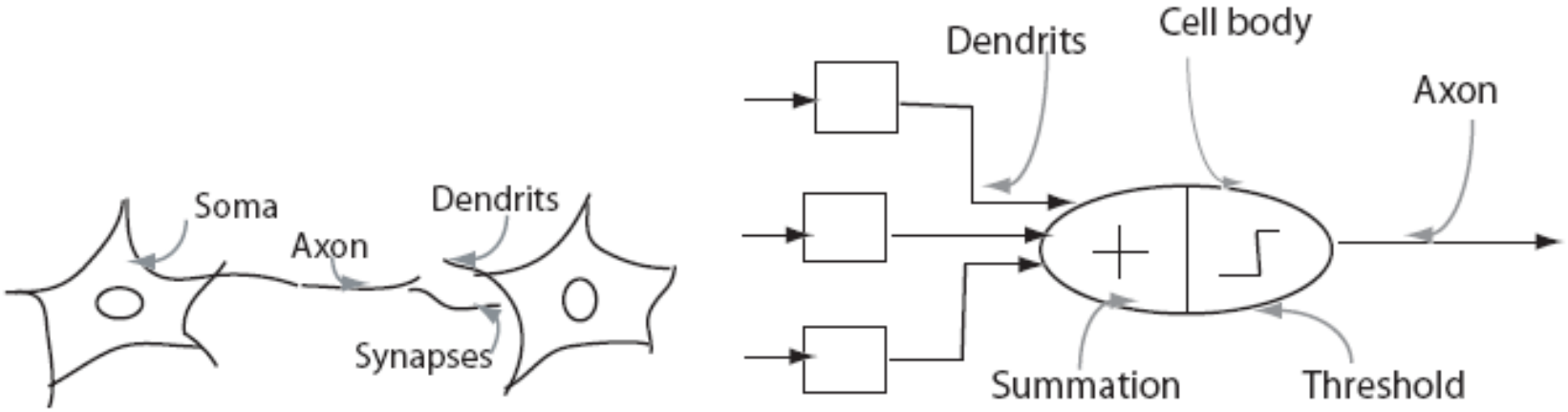




- ~ 100 tỷ nơ-ron
- Mỗi nơ-ron có hàng ngàn kết nối:
  - Thu nhận tín hiệu
  - Lan truyền thông tin



Mô hình mạng nơ-ron nhân tạo

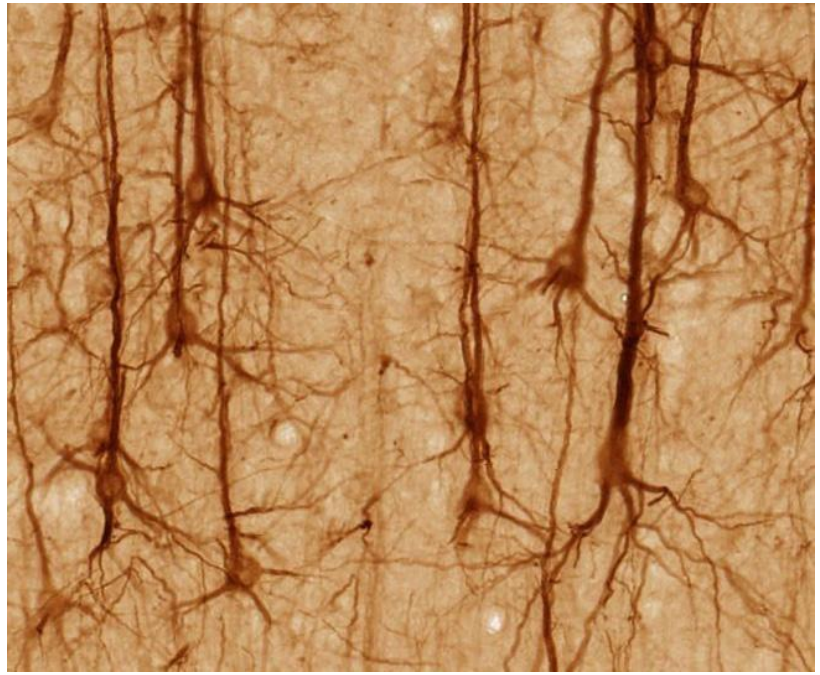


Biological	vs.	Artificial
Soma		Node
Dendrits		Inputs
Axon		Output
Synapse		Weight

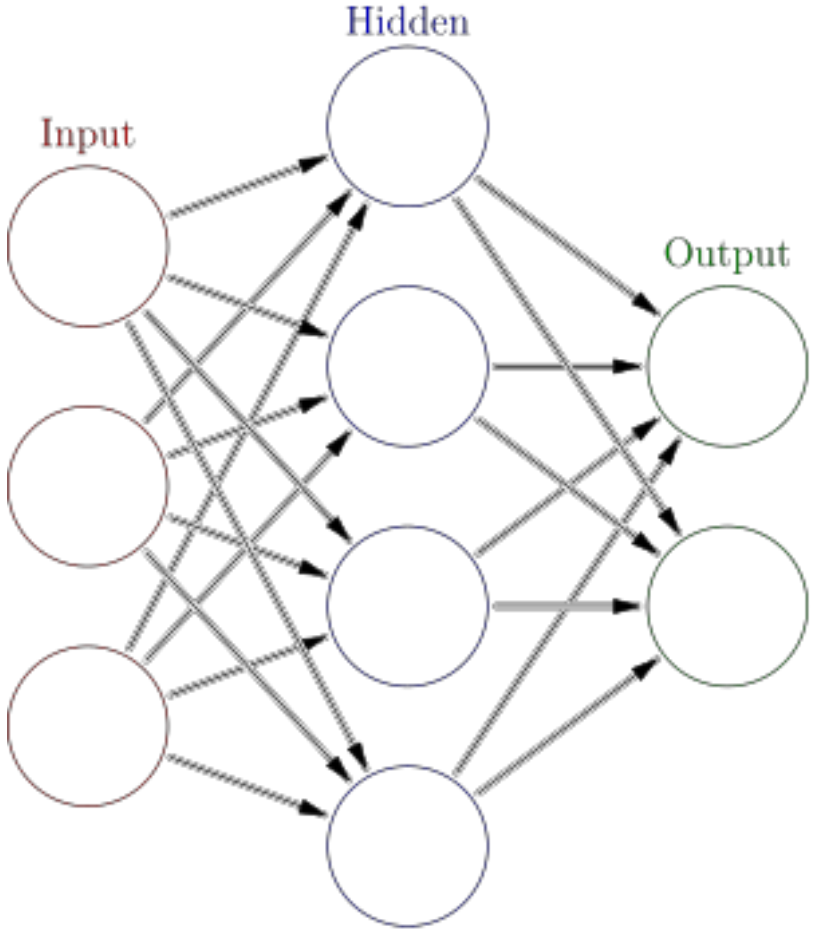
Biological vs. Artificial

Mạng nơ-ron nhân tạo

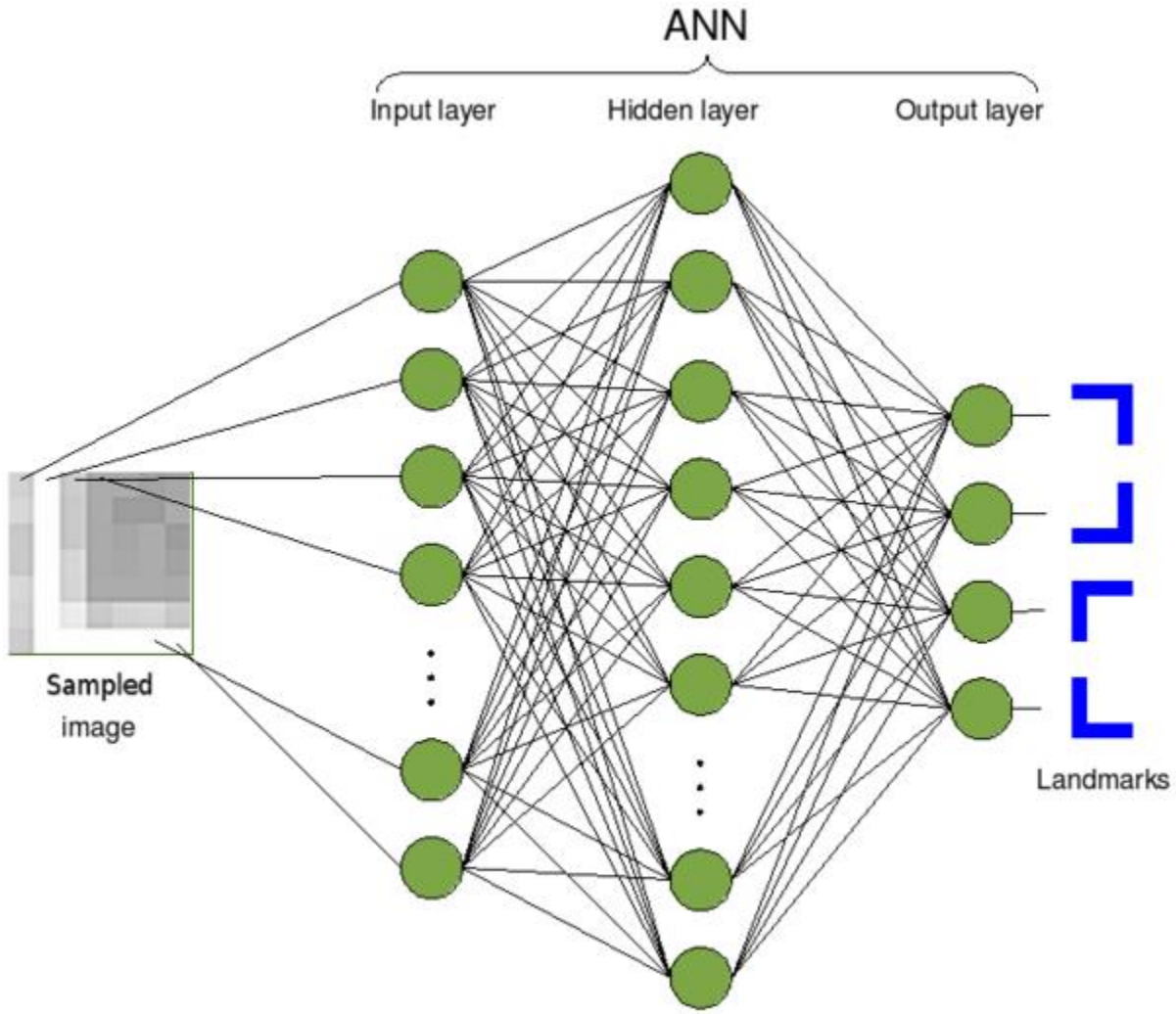
Mạng Nơ-ron



Mô hình mạng Nơ-ron nhân tạo

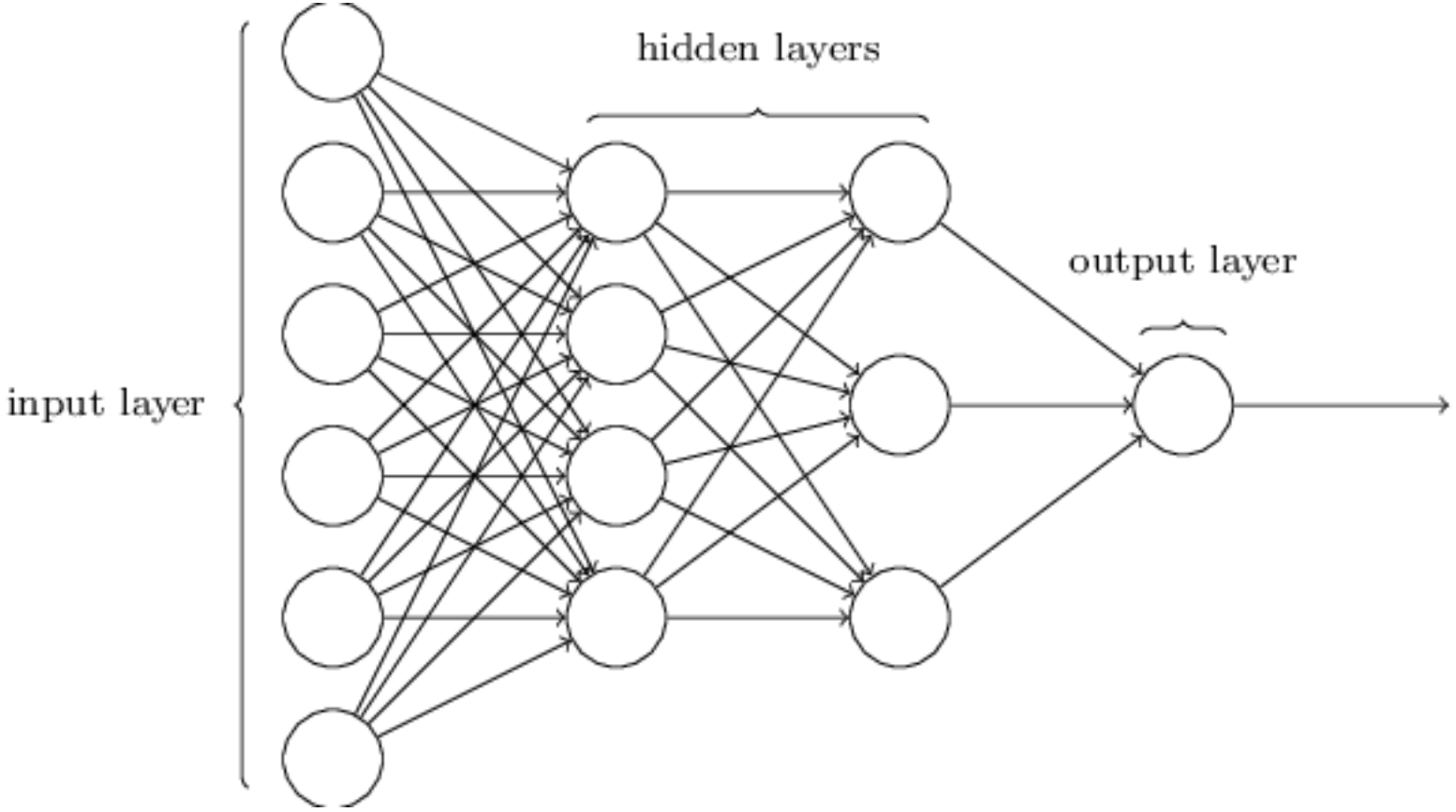


Mạng nơ-ron nhân tạo



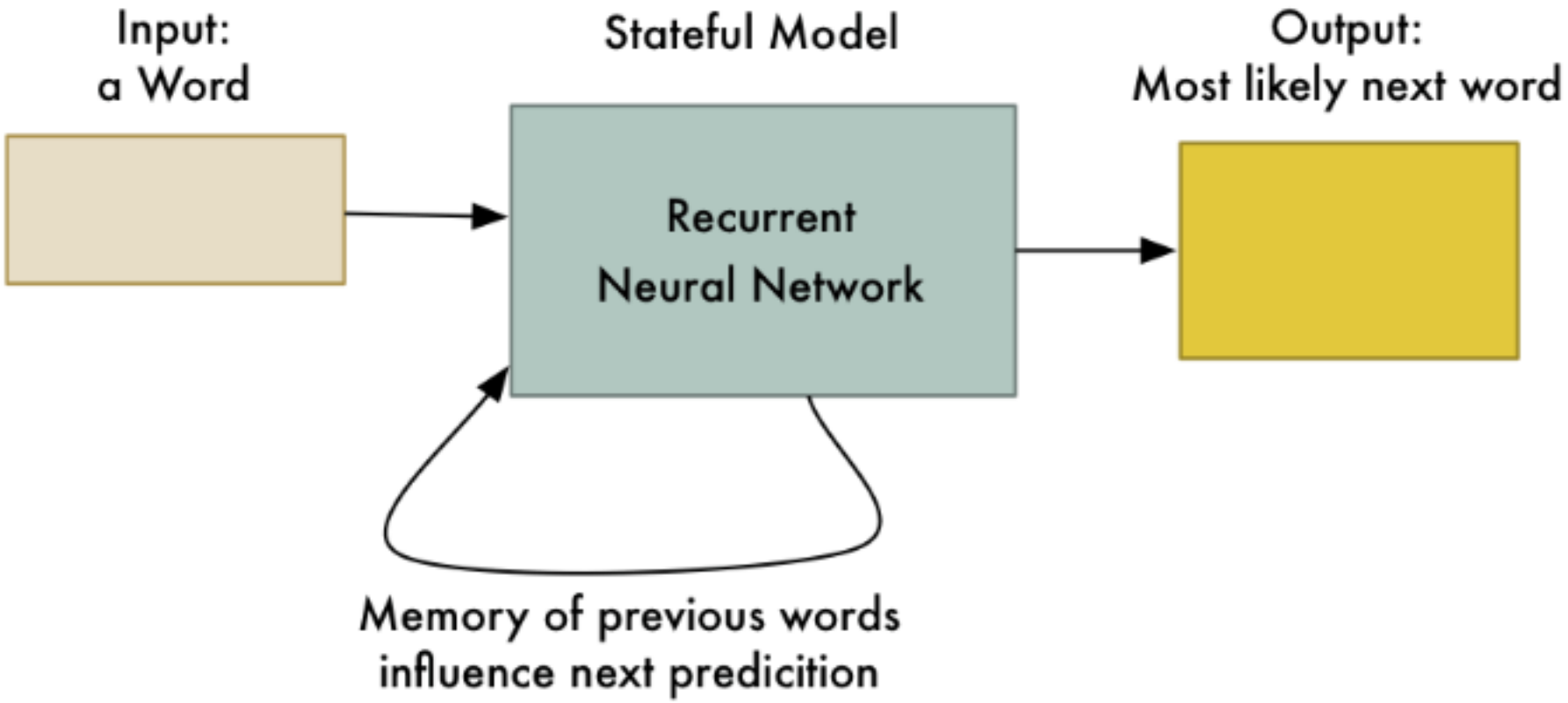
Các loại mạng nơ-ron nhân tạo

- **Feedforward Neural Network**



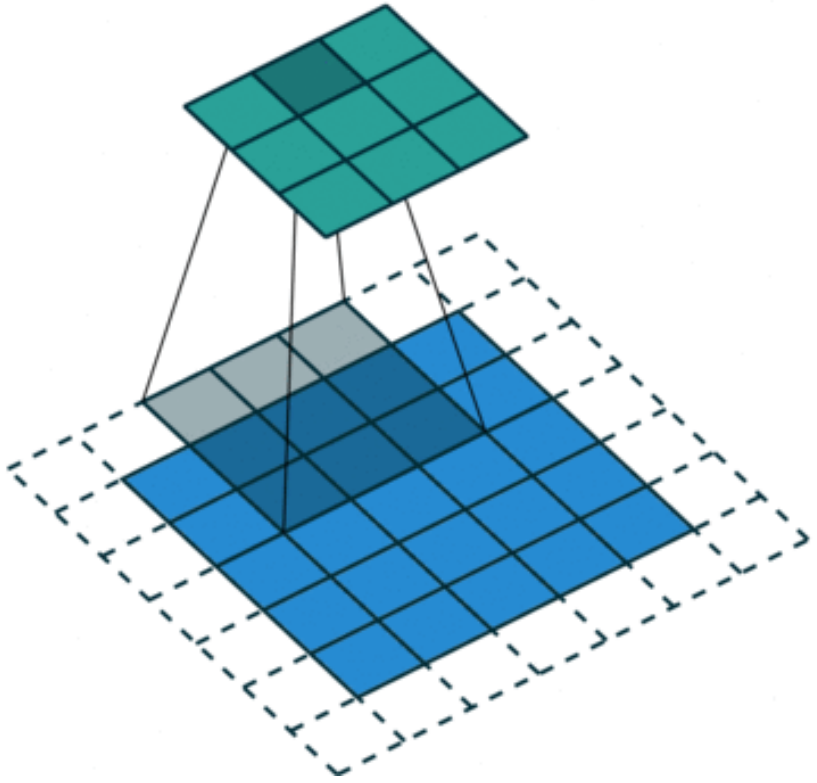
Các loại mạng nơ-ron nhân tạo

- **Recurrent Neural Network(RNN)**



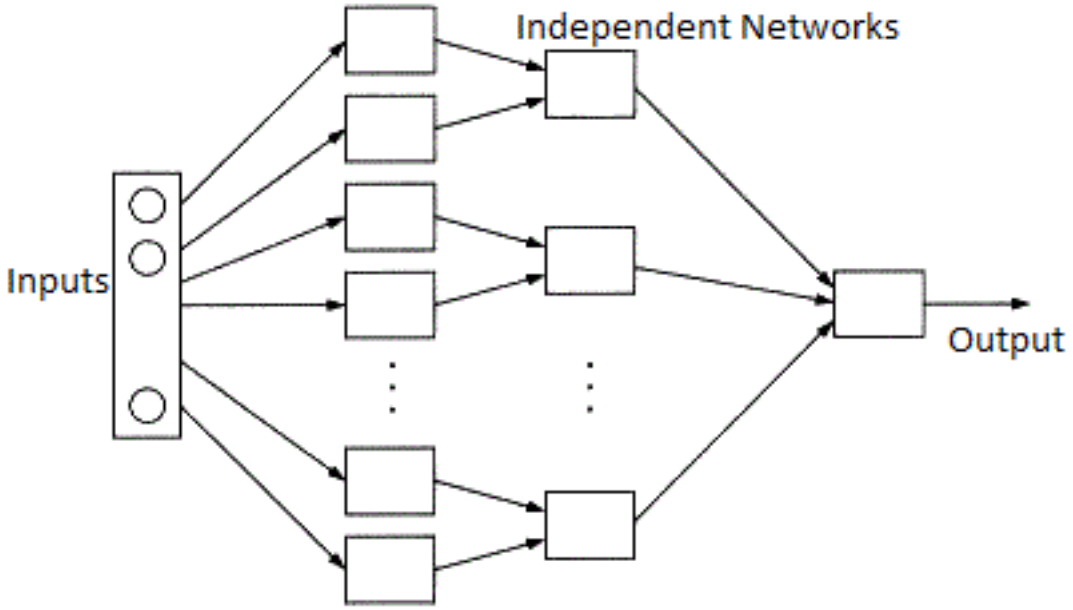
Các loại mạng nơ-ron nhân tạo

- **Convolutional Neural Network**

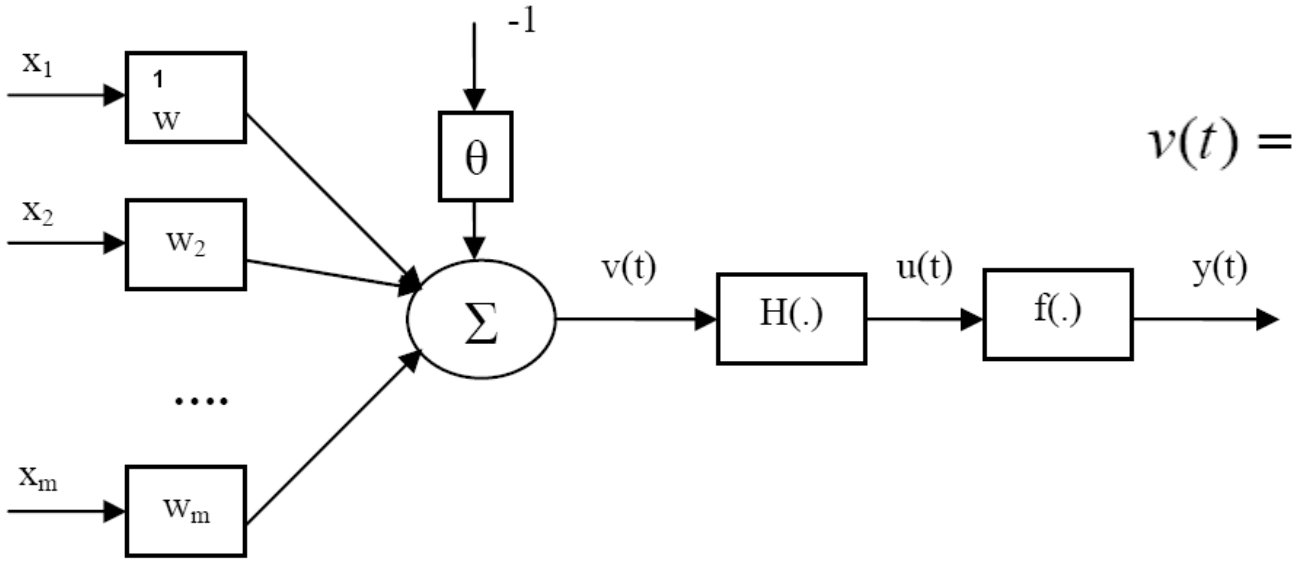


Các loại mạng nơ-ron nhân tạo

- **Modular Neural Network**



Cấu trúc nơ-ron nhân tạo



$$v(t) = \sum_{k=1}^{m} w_k x_k(t) - \theta$$

Trong đó:

v(t): Tổng tất cả các đầu vào mô tả toàn bộ thế năng tác động ở thân nơ-ron.

$X_k(t)$: Các biến đầu vào (các đặc trưng), k=1..M.

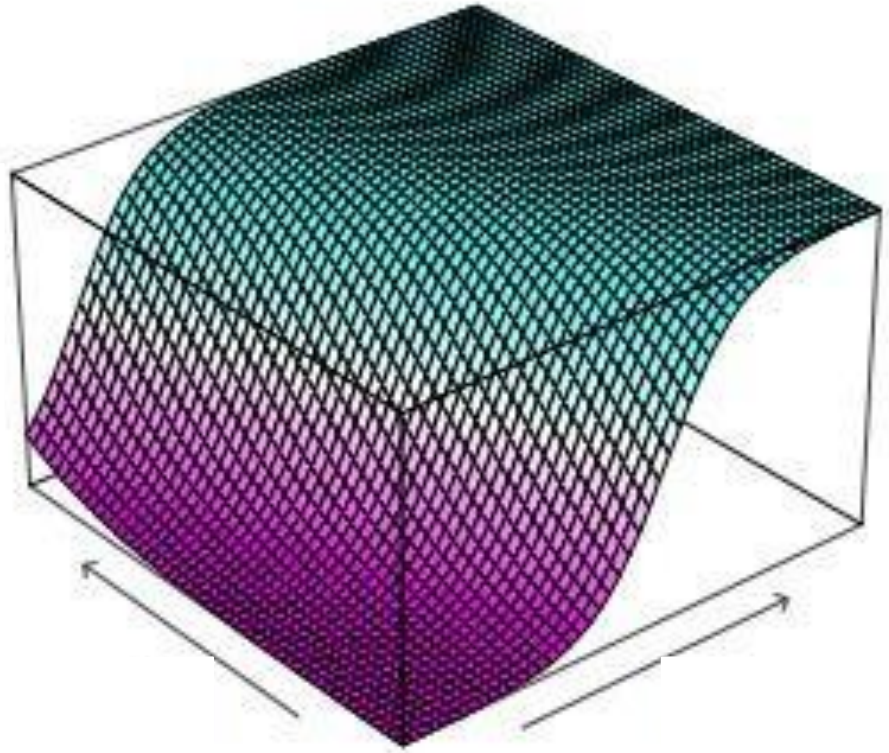
w_k: Trọng số liên kết ngoài giữa các đầu vào k với nơ-ron hiện tại.

H(.): Hàm kích hoạt.

Y(t): Tín hiệu đầu ra nơ-ron.

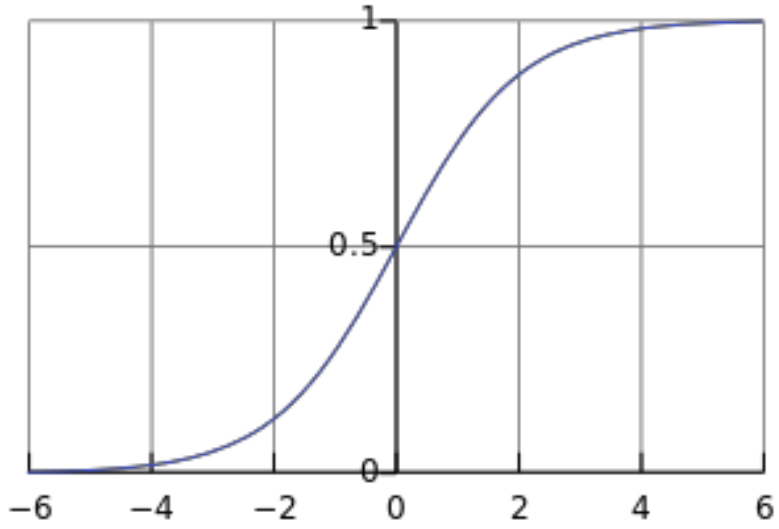
θ: Ngưỡng (là hằng số), xác định ngưỡng kích hoạt.

Hàm Ridge

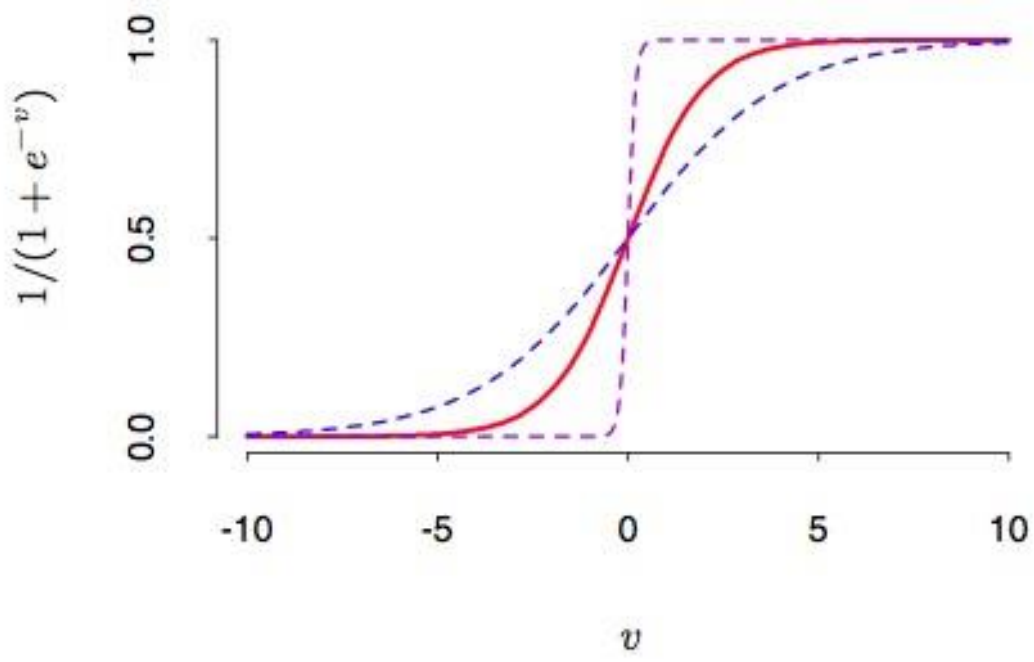


Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

Hàm logistic

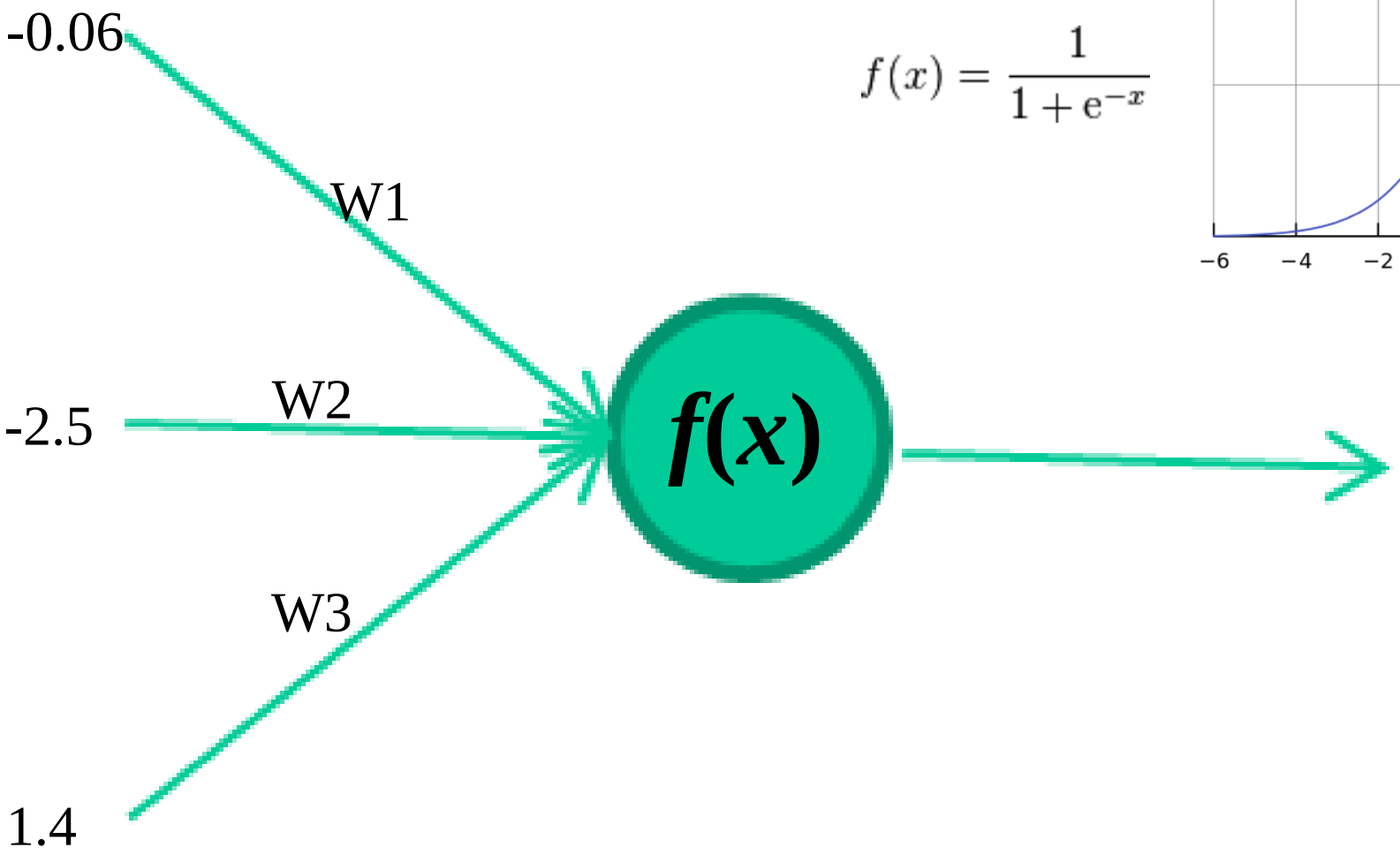


Hàm kích hoạt Sigmoidal

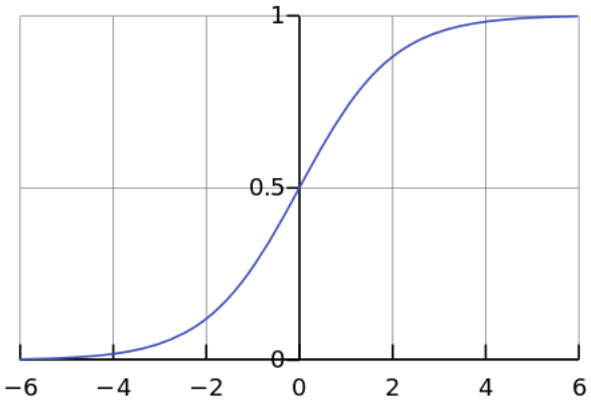


Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

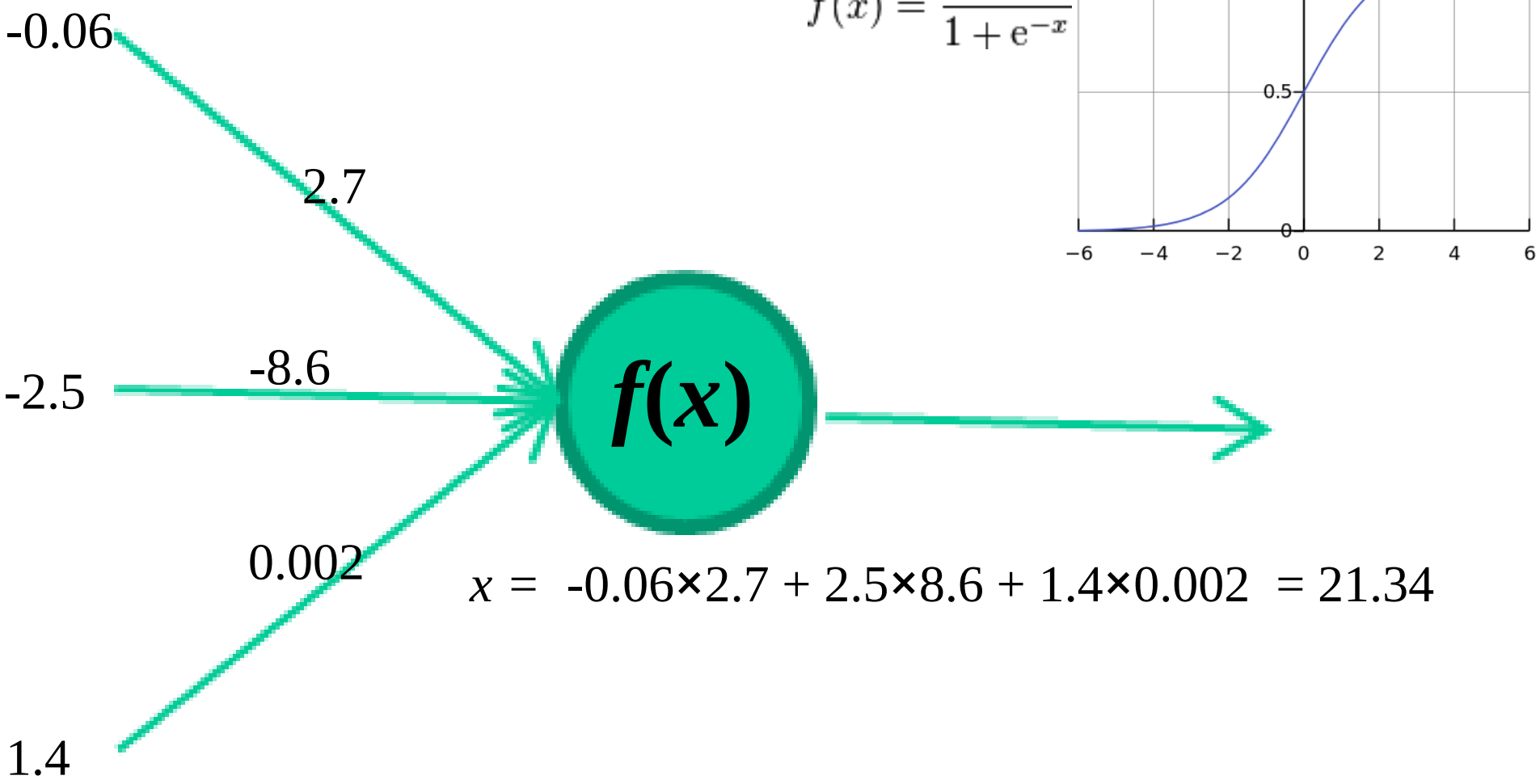
FIGURE 11.3. *Plot of the sigmoid function* $\sigma(v) = 1/(1+\exp(-v))$ *(red curve), commonly used in the hidden layer of a neural network. Included are* $\sigma(sv)$ *for* $s = \frac{1}{2}$ *(blue curve) and* $s = 10$ *(purple curve). The scale parameter* s *controls the activation rate, and we can see that large* s *amounts to a hard activation at* $v = 0$*. Note that* $\sigma(s(v - v_0))$ *shifts the activation threshold from* 0 *to* v_0*.*



$$f(x) = \frac{1}{1 + e^{-x}}$$



David Corne, Heriot-Watt University

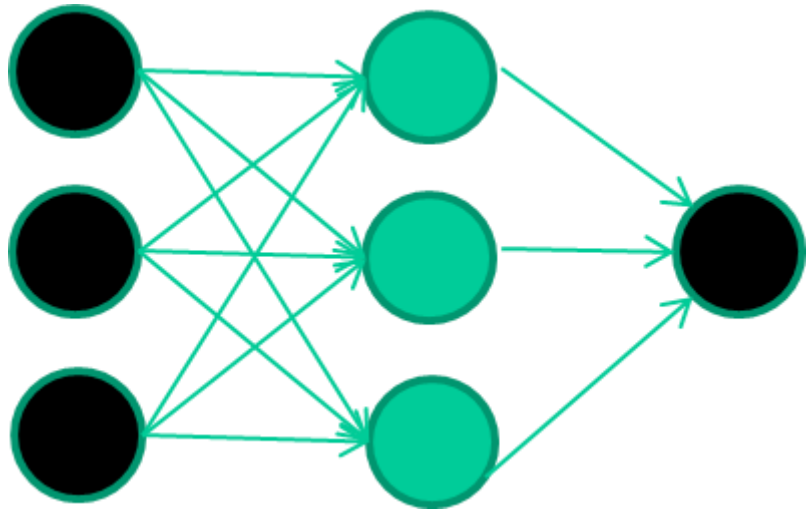


David Corne, Heriot-Watt University

Dữ liệu

Các trường			***Lớp***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

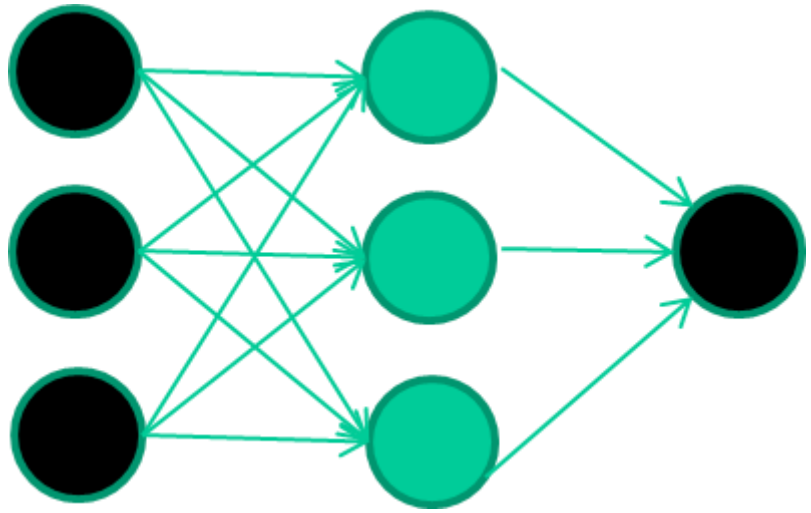
v.v…



Huấn luyện mạng Nơ-ron

Các trường			***Lớp***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

v.v…

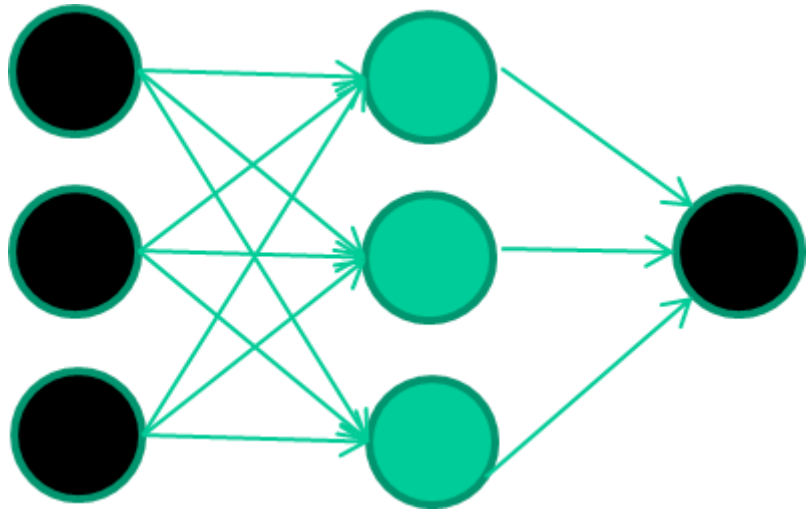


Khởi tạo các trọng số ngẫu nhiên

Dữ liệu huấn luyện

Các trường			***Lớp***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

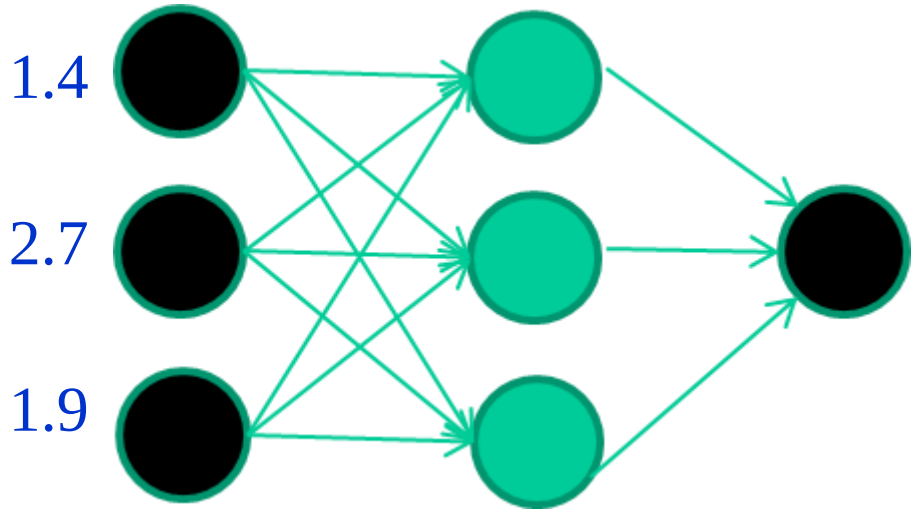
v.v…



Huấn luyện mẫu

Dữ liệu huấn luyện

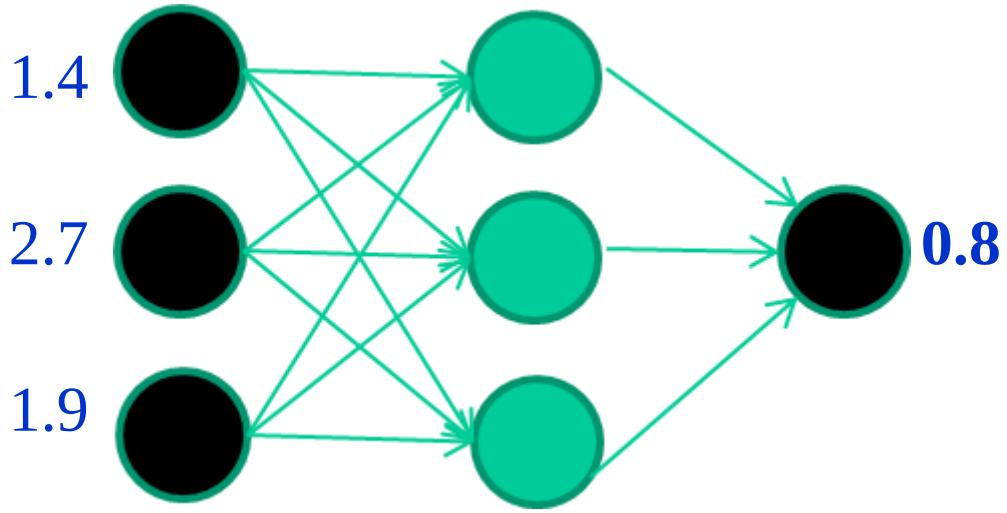
Các trường			***Lớp***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
v.v…			



cung cấp đầu ra

Dữ liệu huấn luyện

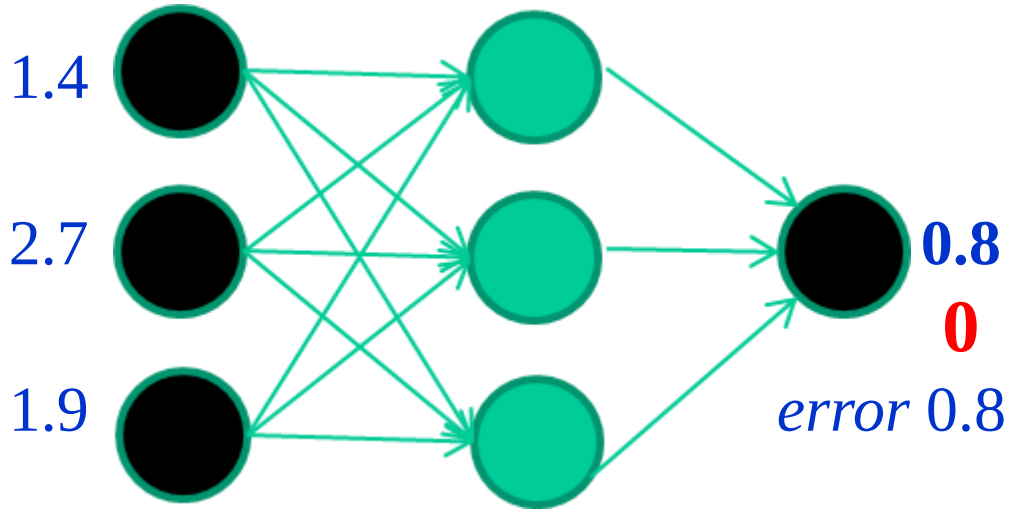
Các trường			***Lớp***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
v.v…			



So sánh giá trị đầu ra

Dữ liệu huấn luyện

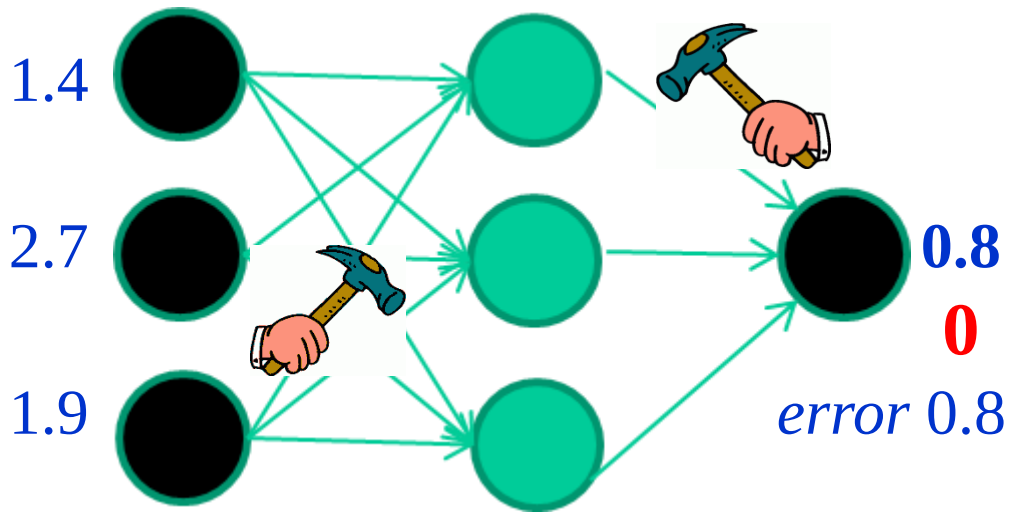
Các trường			***Lớp***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
v.v..			



Điều chỉnh các trọng số dựa vào đầu ra

Dữ liệu huấn luyện

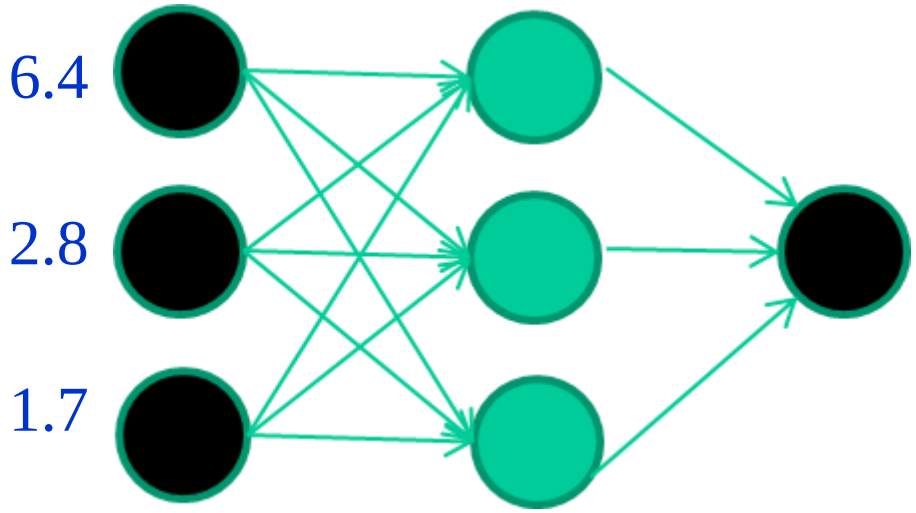
Các trường			***Lớp***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
v.v…			



Huấn luyện mẫu

Dữ liệu huấn luyện

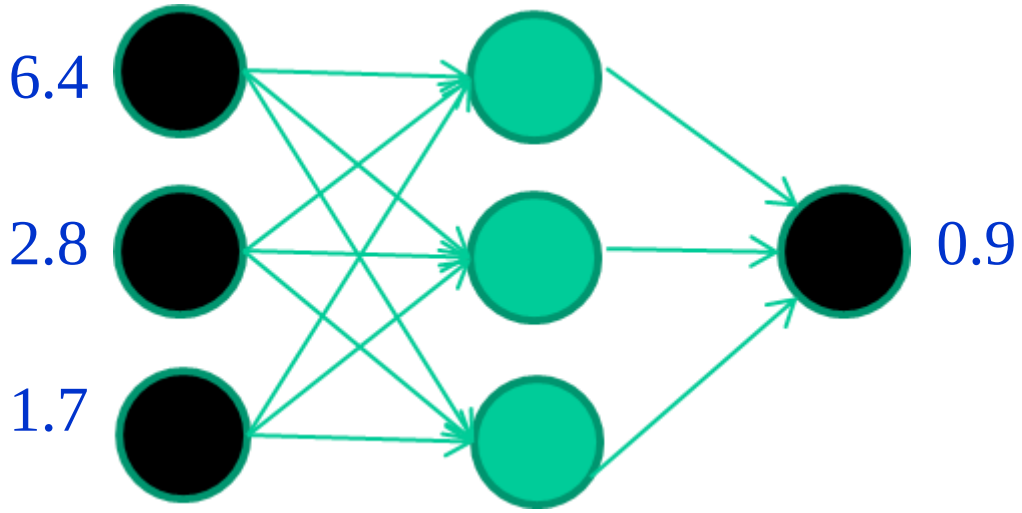
Các trường			***Lớp***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
v.v…			



cung cấp đầu ra

Dữ liệu huấn luyện

Các trường			***Lớp***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
v.v…			

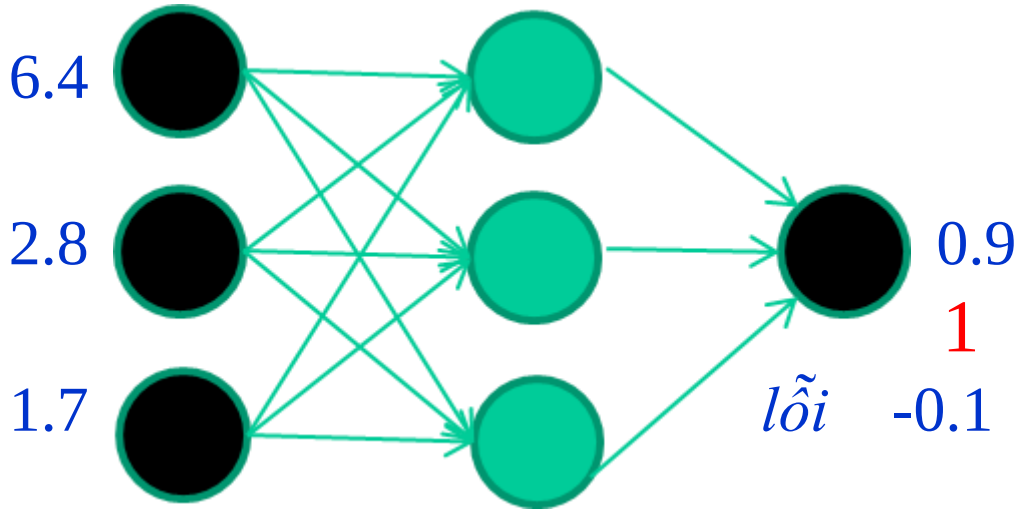


So sánh giá trị đầu ra

Dữ liệu huấn luyện

Các trường			***Lớp***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

v.v…

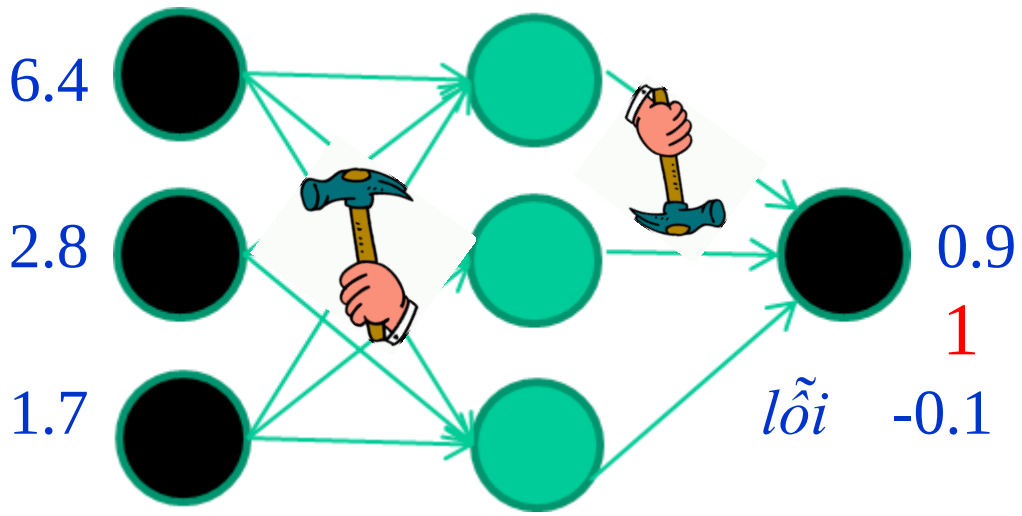


Điều chỉnh các trọng số dựa vào đầu ra

Dữ liệu huấn luyện

Các trường			***Lớp***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

v.v…

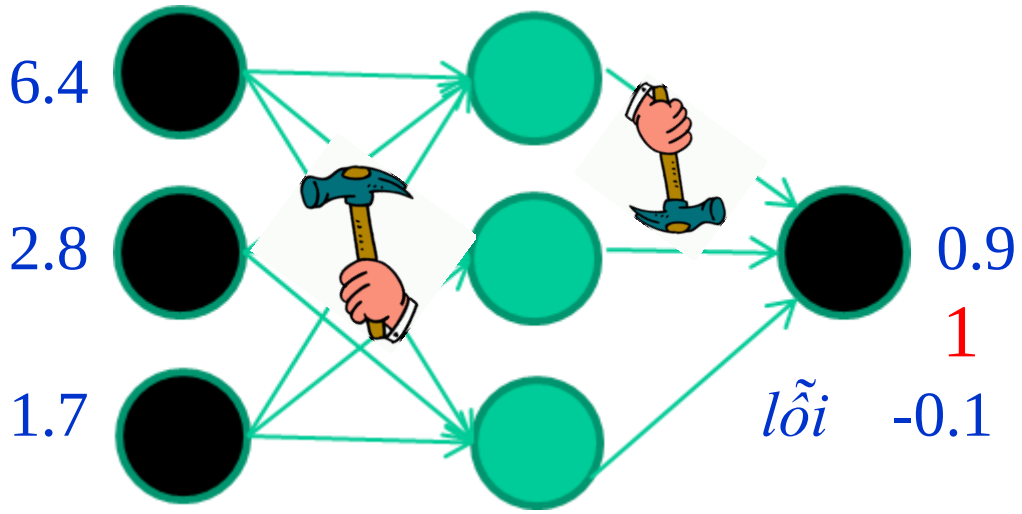


tiếp tục ....

Dữ liệu huấn luyện

Các trường			***Lớp***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

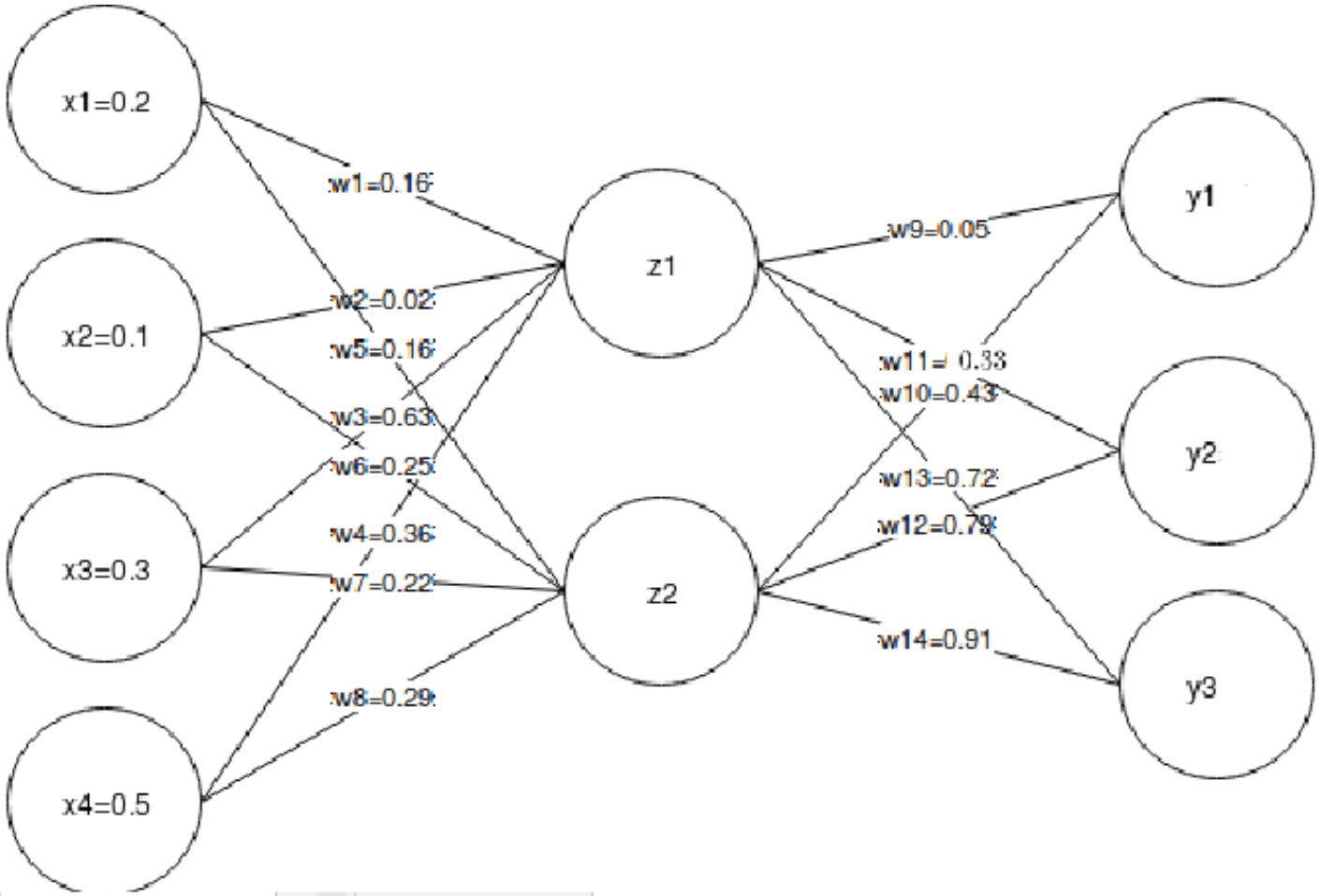
v.v…



Lặp lại hàng ngàn, hang triệu lần – mỗi lần sẽ lấy tập mẫu ngẫu nhiên, và tạo các điều chỉnh về trọng số

Các giải thuật điều chỉnh trọng số được thiết kế để tạo ra các thay đổi mà chúng sẽ giúp giảm lỗi của mô hình

Ví dụ: Tìm đầu ra Y = (y_1, y_2, y_3)



$$W_1 = \begin{bmatrix} 0.16 & 0.16 \\ 0.02 & 0.25 \\ 0.63 & 0.22 \\ 0.36 & 0.29 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 0.05 & 0.33 & 0.72 \\ 0.43 & 0.79 & 0.91 \end{bmatrix}$$

Ví dụ - Sử dụng mạng truyền thẳng

$$f(x) = \frac{1}{(1 + e^{-x})}$$

$$Z = f(X W_1)$$

Ưu điểm của mạng nơ ron nhân tạo

- ANN có tính linh hoạt và thích ứng cao.
- Được sử dụng trong các hệ thống nhận dạng chuỗi và mẫu, xử lý dữ liệu, robot, mô hình hóa, v.v.
- ANN có được kiến thức từ môi trường xung quanh bằng cách thích ứng với các thông số bên trong và bên ngoài và họ giải quyết các vấn đề phức tạp khó quản lý.
- Nó khái quát hóa kiến thức để tạo ra phản ứng đầy đủ cho các tình huống chưa biết.
- Tính linh hoạt - Mạng lưới thần kinh nhân tạ

- o có tính linh hoạt và có khả năng học hỏi, khái quát hóa và thích nghi với các tình huống dựa trên những phát hiện của nó.

Ưu điểm của mạng nơ ron nhân tạo

- Phi tuyến tính - Chức năng này cho phép mạng thu nhận kiến thức một cách hiệu quả bằng cách học. Đây là một lợi thế khác biệt so với mạng tuyến tính truyền thống không đầy đủ khi mô hình hóa dữ liệu phi tuyến tính.
- Mạng nơ-ron nhân tạo có khả năng chịu lỗi lớn hơn mạng truyền thống. Không mất dữ liệu được lưu trữ, mạng có thể tạo lại lỗi trong bất kỳ thành phần nào của nó.
- Một mạng nơ-ron nhân tạo dựa trên học thích ứng

Ví dụ:
Gọi thư viện và khởi tạo các dãy số

```
library("neuralnet")
traininginput =  as.data.frame(runif(50, min=0, max=100))
trainingoutput = sqrt(traininginput)
```

Ví dụ:
Tạo train data

```
Trainingdata=data.frame(traininginput,trainingoutput)
colnames(trainingdata) =c("Input","Output")
```

Ví dụ:
Xây dựng mô hình

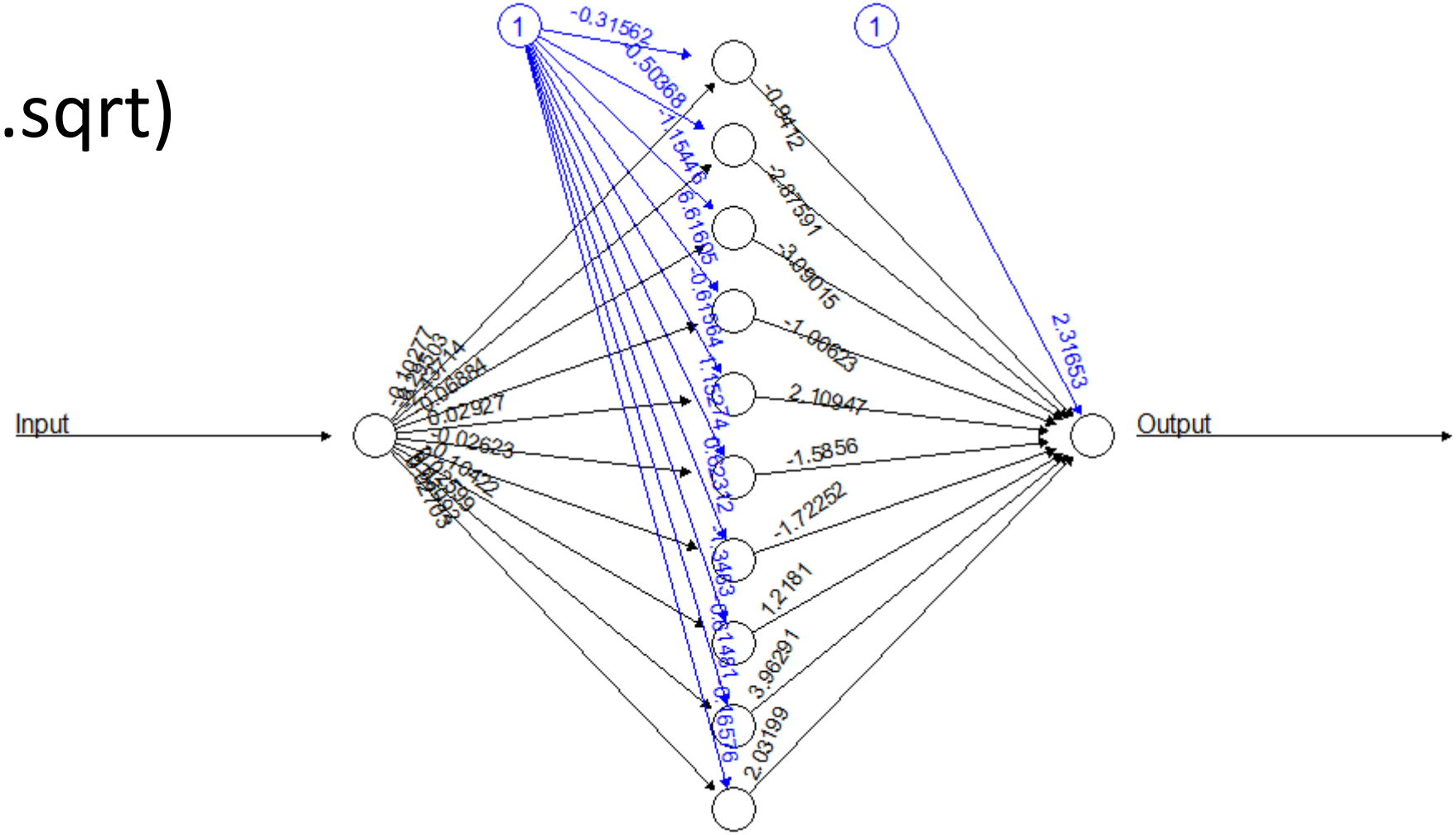
net.sqrt = neuralnet(Output~Input,trainingdata,
hidden=10, threshold=0.01)
print(net.sqrt)

Ví dụ: Các thông tin về mạng nơ ron

```
> names(net.sqrt)
 [1] "call"                "response"                "covariate"
 [4] "model.list"          "err.fct"                 "act.fct"
 [7] "linear.output"       "data"                    "net.result"
[10] "weights"             "startweights"            "generalized.weights"
[13] "result.matrix"
```

Ví dụ: Biểu diễn mô hình

plot(net.sqrt)



Ví dụ:

Kiểm thử mô hình trên test data

```
testdata = as.data.frame((1:10)^2)
net.results = compute(net.sqrt, testdata)
#Xem các thuộc tính của kết quả
ls(net.results)
```

```
> ls(net.results)
[1] "net.result" "neurons"
```

Ví dụ:
Kiểm thử mô hình trên test data

print(net.results$net.result)

```
> print(net.results$net.result)
             [,1]
 [1,] 1.060946
 [2,] 1.992386
 [3,] 3.003752
 [4,] 3.998260
 [5,] 5.000871
 [6,] 6.000122
 [7,] 6.998849
 [8,] 7.999310
 [9,] 9.002404
[10,] 9.989319
```

Ví dụ:
Mô tả lại kết quả rút gọn

```
cleanoutput <- cbind(testdata,sqrt(testdata),
                     as.data.frame(net.results$net.result))
colnames(cleanoutput) <- c("Input","Expected
Output","Neural Net Output")
print(cleanoutput)
```

Ví dụ:
Mô tả lại kết quả rút gọn

```
> print(cleanoutput)
   Input Expected Output Neural Net Output
1      1               1          1.060946
2      4               2          1.992386
3      9               3          3.003752
4     16               4          3.998260
5     25               5          5.000871
6     36               6          6.000122
7     49               7          6.998849
8     64               8          7.999310
9     81               9          9.002404
10   100              10          9.989319
```

Ví dụ: Ảnh hưởng của threshold

Threshold	Error	Steps
0.01	0.000122	2927
0.1	0.001332	1594
1	0.059381	152

CÂU HỎI