

What is Deep Learning?

Deep Learning là một trường con của Machine Learning. Có liên quan với các thuật toán lấy cảm hứng từ cấu trúc và chức năng của bộ não được gọi là Artificial Neural Networks (mạng nơron nhân tạo).

1. “Deep Learning is Large Neural Networks”

Large Neural Networks được xây dựng và được ‘đào tạo’ với lượng dữ liệu lớn, dữ liệu đào tạo càng nhiều thì hiệu suất càng tăng.

2. “Deep Learning is Hierarchical Feature Learning”

Deep Learning có khả năng thực hiện tính năng trích xuất tự động từ dữ liệu thô (dữ liệu ban đầu), hay còn gọi là feature learning (tính năng học tập).

Thuật toán Deep Learning khai thác các phần (thành phần, cấu trúc,...) của dữ liệu đầu vào để khám phá các đại diện nổi bật hơn, và khai thác nhiều mức độ khác nhau.

“Hệ thống phân cấp các khái niệm cho phép máy tính học các khái niệm phức tạp bằng cách xây dựng chúng ra khỏi các khái niệm đơn giản hơn. ... Vì lý do này, chúng tôi gọi phương pháp này là AI deep learning.”

Nghĩa là các khái niệm học được sẽ phân thành nhiều cấp độ khác nhau, khái niệm cấp cao sẽ được định nghĩa bởi các khái niệm cấp thấp hơn.

3. “Deep Learning as Scalable Learning Across Domains “ (Deep Learning : học tập có thể mở rộng trên toàn miền)

Deep Learning có thể ứng dụng và phát triển mạnh trên nhiều lĩnh vực là vì dữ liệu đa dạng như hình ảnh (dữ liệu pixel), văn bản ,âm thanh,....

4. Các kỹ thuật phổ biến

- Multilayer Perceptron Networks.
- Convolutional Neural Networks (kiến trúc mạng nhận dạng đối tượng trong dữ liệu hình ảnh).
- Long Short-Term Memory (LSTM), một loại của Recurrent Neural Networks (học dữ liệu trình tự).

Link : <https://machinelearningmastery.com/what-is-deep-learning/>

Neural Network?

Neural network - mô hình hóa cách thức mà bộ não thực hiện một nhiệm vụ hoặc chức năng cụ thể;

Neural networks sử dụng một kết nối lớn các tế bào tính toán đơn giản được gọi là “neurons” hoặc “processing units” (các đơn vị xử lý).

Neural network là một bộ xử lý phân tán song song được tạo thành từ các đơn vị xử lý đơn giản, tự động lưu trữ kiến thức, kinh nghiệm được học và luôn sẵn sàng để sử dụng. Nó giống như bộ não với hai khía cạnh:

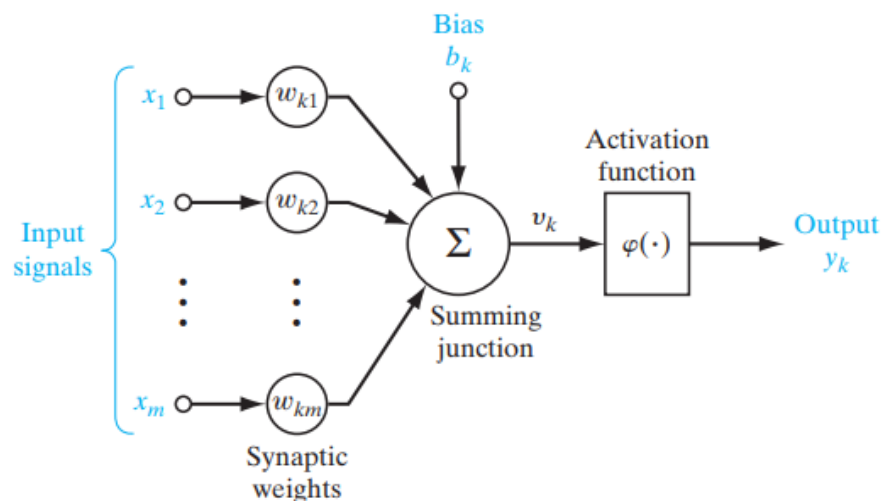
1. Kiến thức được tích lũy lại bởi network từ môi trường của nó thông qua một quá trình học tập.???
2. Mối liên kết giữa các interneuron, còn gọi là synaptic weights, được dùng để chứa những thông tin nhận được

*Interneuron – nơron trung gian

*Synaptic weights – cường độ hoặc biên độ của một kết nối giữa hai nút (neuron)

A neuron?

Neuron là một đơn vị xử lý thông tin cơ bản cho hoạt động của neural network



Hình 1: Mô hình phi tuyến của một neuron, có nhãn k

Neural model có ba yếu tố cơ bản:

1. Một tập hợp các **synapse** hoặc **connecting links** (các liên kết kết nối), mỗi liên kết được đặc trưng bởi **weight** (trọng lượng) hay **strength** (sức mạnh) của riêng nó.

Tín hiệu x_j là đầu vào của **synapse j** được kết nối với **nơron k** được nhân với **synaptic weight w_{kj}** . Không giống như weight của một synapse trong não, synaptic weight của một tế bào thần kinh nhân tạo có thể có giá trị âm hoặc dương.

2. **Adder** - bộ cộng để tổng hợp các tín hiệu đầu vào, được cân bằng bởi các synaptic strength tương ứng của nơon; Các hoạt động được mô tả ở đây tạo thành **linear combiner** (bộ kết hợp tuyến tính).
3. **Activation function** (chức năng kích hoạt) để hạn chế biên độ của đầu ra của tế bào thần kinh. Activation function cũng được gọi là **squashing function**, trong đó nó squashes (giới hạn) phạm vi biên độ cho phép của tín hiệu đầu ra tới một số giá trị hữu hạn.

- Phương trình viết lại từ mô hình

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (1)$$

và

$$y_k = \varphi(u_k + b_k) \quad (2)$$

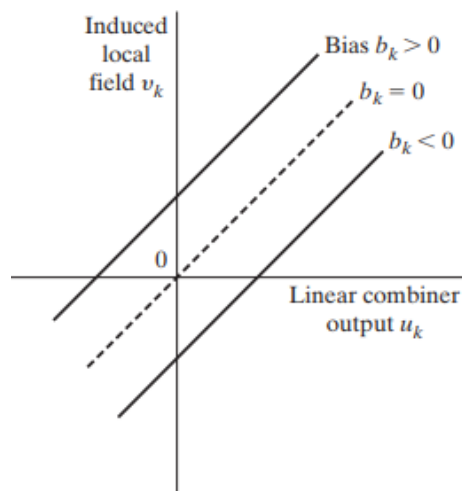
Với

- x_1, x_2, \dots, x_m : input
- $w_{k1}, w_{k2}, \dots, w_{km}$: synaptic weight tương ứng của neuron k
- u_k : **linear combiner output** ??? (không hiển thị trong mô hình)
- b_k : bias - tăng hoặc giảm đầu vào của activation function, tùy thuộc vào việc nó là dương hay âm, tương ứng. ???
- $\varphi(.)$: activation function
- y_k : output

Việc sử dụng bias (b_k) có ảnh hưởng của việc áp dụng **affine transformation** đến đầu ra u_k

$$v_k = u_k + b_k \quad (3)$$

Tùy thuộc vào bias b_k là dương hay âm, mối quan hệ giữa **induced local field** (trường cục bộ cảm ứng), hay **activation potential** (tiềm năng kích hoạt) - v_k của nơon k và **linear combiner output** u_k được thay đổi theo cách minh họa trong hình 2:



Hình 2: Affine transformation do sự có mặt của bias b_k , $v_k = b_k$ khi $u_k = 0$

Do bias b_k là một tham số bên ngoài của neuron k (được thể hiện trong phương trình (2)), phương trình (1) và (3) có thể kết hợp thành :

$$v_k = \sum_{j=1}^m w_{kj} x_j \quad (4)$$

và

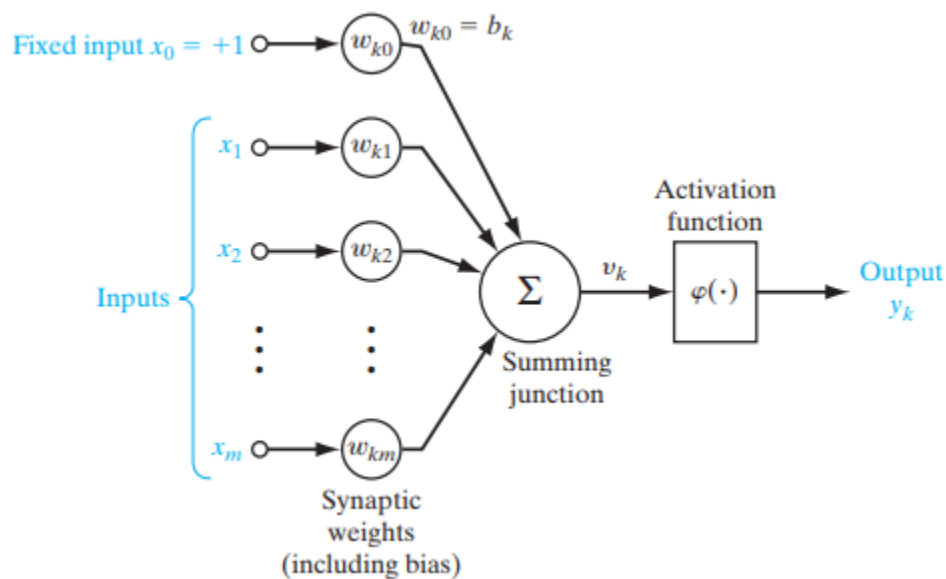
$$y_k = \varphi(v_k) \quad (5)$$

Trong phương trình (4) ta đã có thêm một synapse mới với

- Input: $x_0 = +1$ (6)

- Weight: $w_{k0} = b_k$ (7)

Neural model cải tiến như hình (3)



Hình 3: Một mô hình phi tuyến của một neuron khác, với w_{k0} thể hiện cho b_k

* weight - trọng số - vd hệ số .-> thể hiện mức độ quan trọng

* Synapse - khớp thần kinh

* Squashing function???? : trong neural network, chúng có thể được sử dụng tại các nút trong một hidden layer (lớp ẩn) để nén đầu vào.

**Chỉ số đầu tiên trong w_{kj} đề cập đến nơ-ron được đề cập và chỉ số thứ hai đề cập đến đầu vào của khớp thần kinh mà trọng lượng đề cập đến.

Activation Function? 10 func

Activation function, ký hiệu $\varphi(v)$: xác định đầu ra của một nơron về mặt **induced local field** v . Dưới đây là 2 loại cơ bản.

1. Threshold Function (Chức năng Ngưỡng):

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \quad (8)$$

Tương ứng, đầu ra của nơron k sử dụng một Threshold Function như vậy:

$$y_k = \begin{cases} 1 & \text{if } v_k \geq 0 \\ 0 & \text{if } v_k < 0 \end{cases} \quad (9)$$

Với

$$v_k = \sum_{j=1}^m w_{kj} x_j + b_k \quad (10)$$

2. Sigmoid Function:

Sigmoid Function có đồ thị là hình chữ "S", là hình thức phổ biến nhất của Activation function được sử dụng trong việc xây dựng neural network.

Sigmoid Function là một hàm tăng cường nghiêm ngặt thể hiện một sự cân bằng duyên dáng giữa hành vi tuyến tính và phi tuyến. ?????

Sigmoid function là một logistic function.

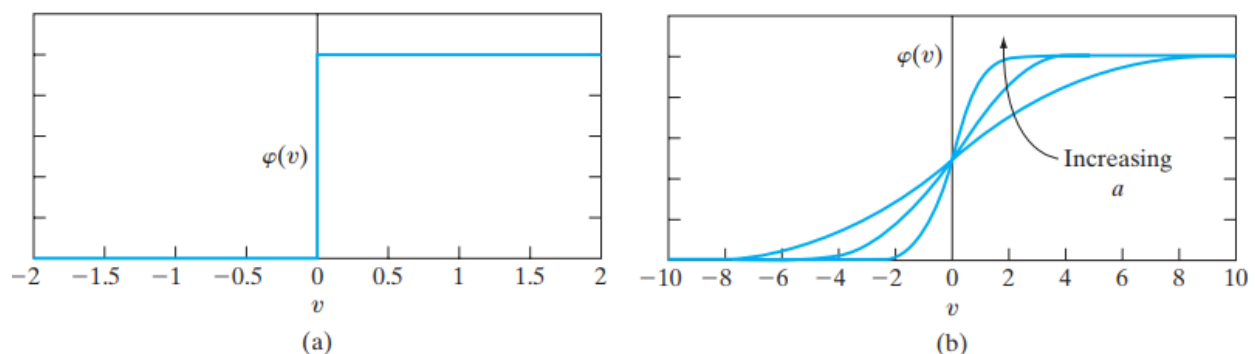
$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (11)$$

Với

a : slope parameter của Sigmoid function : độ dốc

Khi thay đổi tham số a , ta có được hàm sigmoid của các sườn dốc khác nhau.

Trong giới hạn (?), khi tham số dốc (a) tiến đến vô cùng, Sigmoid Function trở thành một Threshold Function. Trong khi Threshold Function giả định giá trị 0 hoặc 1, hàm sigmoid giả định một phạm vi giá trị liên tục từ 0 đến 1.



Hình 4: a) Threshold Function

b) Sigmoid function khi thay đổi độ dốc - a .

Các hàm kích hoạt được xác định trong các phương trình(8) và (11) dao động từ 0 đến +1 .

Trong trường hợp mong muốn có phạm vi activation function từ -1 đến 1.Khi đó Threshold Function được định nghĩa lại:

$$\varphi(v) = \begin{cases} 1 & \text{if } v > 0 \\ 0 & \text{if } v = 0 \\ -1 & \text{if } v < 0 \end{cases} \quad (12)$$

Đây là **Signum function**.

Đối với dạng tương ứng của Sigmoid function, chúng ta có **Hyperbolic tangent function**:

$$\varphi(v) = \tanh(v) \quad (13)$$

Network architecture?

1. Single-Layer Feedforward Networks

Trong một mạng nơron phân lớp, các nơron được tổ chức dưới dạng các lớp.

Ở dạng đơn giản nhất của một mạng phân lớp, chúng ta có **input layer** của các nút nguồn truyền trực tiếp đến output layer của các nơron (các nút tính toán), nhưng không ngược lại. Nói cách khác, nó thuộc loại mạng feedforward.

- 2. Multilayer Feedforward Networks
- 3. Recurrent Networks

Nguồn : Simon Haykin (2009), *Neural Networks and Learning Machines Third Edition*, Pearson,? (nhiều tên thành phố quá nên em không xác định dc).