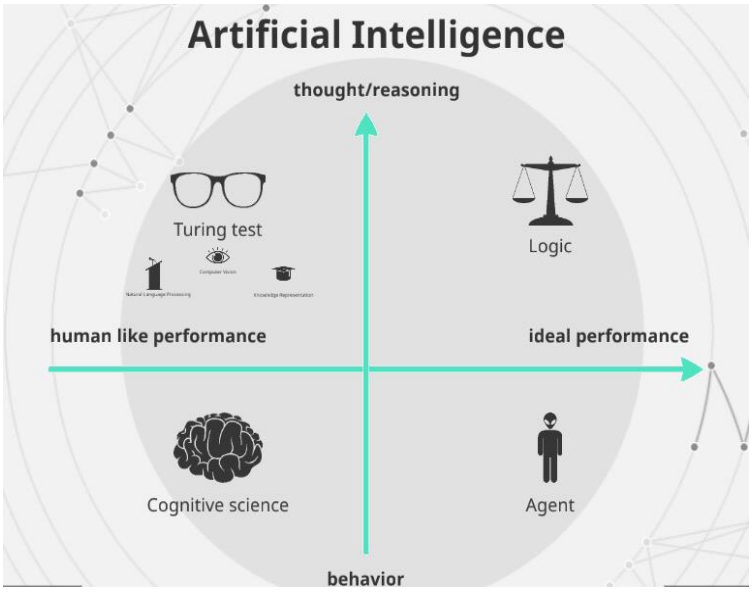
DEEP LEARNING REPORT

1. Khái niệm Deep Learning



Hình 1. Tổng quan AI[1]

AI (Artificial Intelligence) là một ngành thuộc lĩnh vực khoa học máy tính. AI được nghiên cứu với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người. Trong lịch sử phát triển của AI, các nhà nghiên cứu chia thành 4 hướng tiếp cận chính:

- Hành động như người (acting humanly)
- Suy nghĩ như người (thinking humanly)
- Suy nghĩ hợp lý (thinking rationally)
- Hành động hợp lý (acting rationally)

Với hướng tiếp cận "hành động như người", các nhà khoa học đã đạt một số thành tựu như:

[1] https://ongxuanhong.wordpress.com/2017/09/04/ai-machine-learning-deep-learning-phan-biet-nhu-the-nao-cho-dung/

- **Natural language processing**: máy có khả năng đọc hiểu và giao tiếp bằng ngôn ngữ tự nhiên với người.
- **Knowledge representation**: máy có khả năng lưu trữ tri thức thông qua thị giác, thính giác hay văn bản.
- **Automated reasoning**: máy có khả năng sử dụng tri thức đã lưu trữ để trả lời câu hỏi hay đưa ra kết luận hữu ích.
- **Machine learning**: máy có khả năng thích nghi với các điều kiện môi trường xung quanh để rút trích ra các nguyên lý từ tri thức thu nhận được phục vụ cho việc ra quyết định.
- **Computer vision**: máy có khả năng quan sát và xác định được các đối tượng xung quanh.
- **Robotics**: máy có khả năng tương tác với đối tượng và di chuyển trong môi trường xung quanh.

Trong đó, **Machine learning** là một trong những lĩnh vực nhận được sự quan tâm nhiều nhất hiện nay. Để đạt được mục tiêu "hành động như người", các nhà khoa học đã nghiên cứu ra nhiều giải thuật và các hướng giải quyết khác nhau:

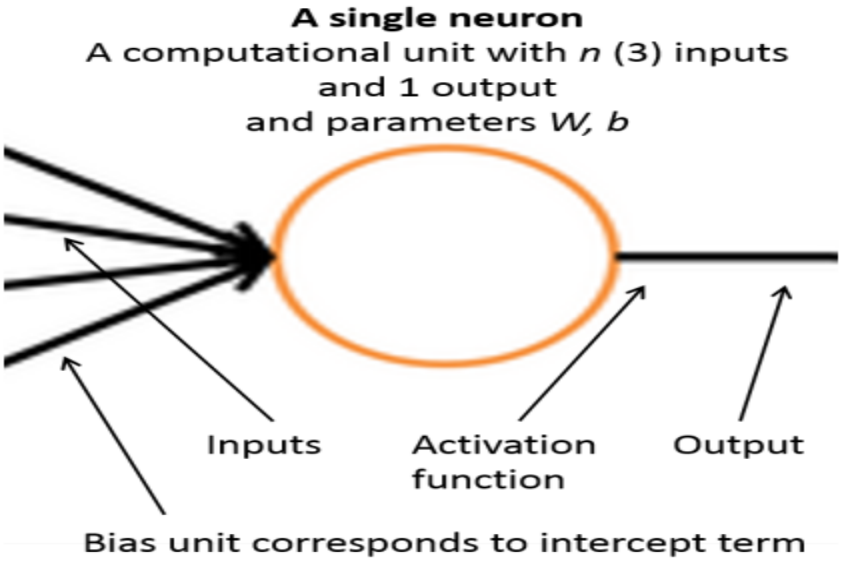
- **Supervised-learning**: decision tree, k-NN, naive bayes, SVM, neural network, deep learning,…
- **Unsupervised-learning**: k-means, hierachical clustering,…
- **Reinforcement learning**: passive/acive/generalization,…

Trong đó, **Deep Learning (DL)** là một phương pháp nằm trong hướng giải quyết học có giám sát của Machine Learning (ML). Thật ra, Deep Learning (DL) đã ra đời từ rất lâu nhưng sau một thời gian vắng bóng, Deep Learning đang hồi sinh mạnh mẽ trong những năm gần đây. Trong báo cáo này sẽ trình bày deep learning là gì, xây dựng deep learning như thế nào và tại sao Deep Learning lại hồi sinh mạnh mẽ như vậy.

1.1. Mạng neural

Deep Learning được giới thiệu từ rất lâu và dựa trên cơ sở của neural network. Chúng ta hãy tìm hiểu một số kiến thức cơ bản về neural network trước khi phân tích kỹ về Deep Learning.

Năm 1943, McCulloch & Pitts lần đầu tiên giới thiệu mô hình toán học của neural network. Sau đó, có rất nhiều công trình nghiên cứu về các bài toán dự đoán áp dụng neural network được công bố. Các công trình này chủ yếu tập trung vào xây dựng hệ thống dự báo cho một lĩnh vực cụ thể như y khoa, tài chính,… Một vài trong số đó tiến hành so sánh các phương pháp dự đoán và đưa ra các nhận xét cho từng phương pháp này.



Hình 2. Một neural đơn giản [4]

Mạng neural gồm rất nhiều neural và được phân thành từng lớp. Một neural đơn giản gồm inputs, hàm kích hoạt (activation) và output. Inputs là một vector n chiều. Output được tính bởi hàm:

$$a = f(w^T x + b)$$

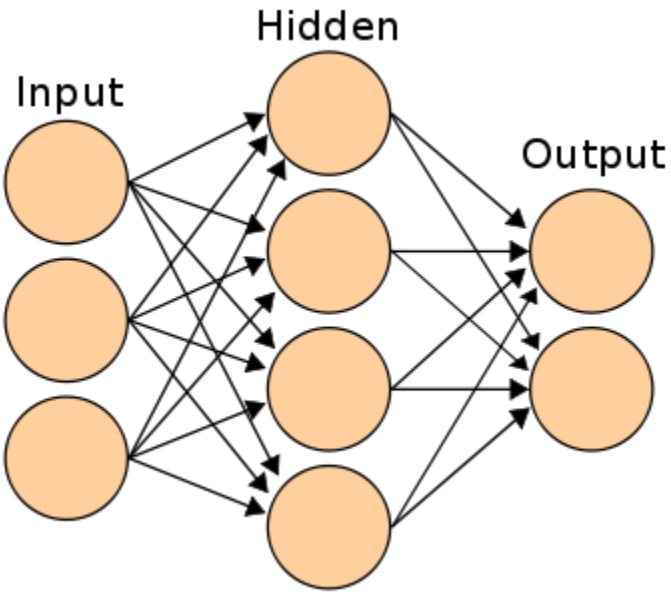
với f(.) được gọi là hàm kích hoạt. Hàm này còn được gọi là hàm phi tuyến tính và thường sử dụng là hàm *sigmoid*[2]:

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

hoặc là một hàm *hypepol*[3]:

$$f(x) = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

Hàm kích hoạt *sigmoid* được chuẩn hóa với dãy số thực từ khoảng [0; 1]. Output của mạng neural là một sự chuyển đổi của đầu vào với các tương tác khác nhau của các đầu vào ban đầu. Một neural network sắp xếp các neural đơn lẻ theo chiều ngang (bên cạnh nhau) và theo chiều dọc (trên đầu) và sau đó là một lớp đầu ra cuối cùng.



Hình 3. Mô hình neural netwok

Mỗi hàm kích hoạt của mỗi neural a_i sẽ được tính toán với các tham số W_i của nó, thêm vào đó là bias b_i của nó: $a_i = f(W_i x + b_i)$, trong đó mỗi vector tham đố là $W_i \in R^n$ (R^n là tập n số thực). Nhiệm vụ của neural network là tìm W và b sao cho độ lỗi huấn luyện là nhỏ nhất (khi huấn luyện giá trị output của mô hình và giá trị thực – groundtruth sẽ khác,

[2] https://en.wikipedia.org/wiki/Sigmoid_function
[3] https://vi.wikipedia.org/wiki/H%C3%A0m_hypebolic

thay đổi các tham số để sự khác biệt của giá trị output của mô hình và giá trị thực là nhỏ nhất).

1.2. Deep Learning[4]

Deep Learning là một lĩnh vực nghiên cứu của Machine Learning. Deep Learning là một cách học với nhiều cấp độ được biểu diễn và rút trích để giúp nhận biết được dữ liệu như hình ảnh, âm thanh và văn bản. Deep learning là một neural network "***rộng***" và "***sâu***". Vậy như thế nào là "***rộng***" và "***sâu***". Với "***rộng***" là số input đầu vào lớn, thường là một ma trận với số chiều lớn, chủ yếu là số mẫu lớn, mỗi mẫu là một hàng của ma trận. Với "***sâu***" là có nhiều lớp neural ẩn (hơn hoặc bằng 2). Hay nói cách khác, Deep Learning là một neural network (1) bao gồm nhiều lớp hoặc giai đoạn của xử lý thông tin (2) phương pháp học có giám sát hoặc học không giám sát với số đặc trưng rất lớn. Deep learning là sự giao thoa giữa nhiều lĩnh vực nghiên cứu như mạng neural, AI, mô hình đồ họa, tối ưu, phát hiện mẫu và xử lý tín hiệu.

Thật ra, Deep Learning ra đời từ rất lâu nhưng có một thời gian dài không được các nhà khoa học nghiên cứu phát triển. Chỉ vài năm trở lại đây, Deep Learning mới thực sự hồi sinh mạnh mẽ. Ba lý do quan trọng để lý giải sự hồi sinh của Deep Learning là (1) sự nâng cao khả năng xử lý của GPU (2) sự tăng lên nhanh chóng của dữ liệu có giá trị (3) sự phát triển của các nghiên cứu về kỹ thuật xử lý trong máy học. Nhờ những sự phát triển vượt bậc này mà kích hoạt các phương pháp Deep Learning để khám phá các vấn đề phức tạp, các hàm không tuyến tính, để học phân tán và phân cấp đặc trưng, rất hiệu quả trong phân lớp có dán nhãn hoặc không dán nhãn.

Nếu máy học có thể học các đặc trưng một cách tự động, toàn bộ quá trình học có thể được tự động một cách dễ dàng hơn thì nhiều vấn đề phức tạp có thể được giải quyết. Deep learning cung cấp một cách học đặc trưng tự động. Mô hình Deep learning như CNN huấn luyện học hình ảnh như bộ não con người. Hinton and Salakhutdinov (2006) giới thiệu một phương pháp mới để tiền huấn luyện Deep Learning. Ý tưởng sử dụng Restricted

[4] https://machinelearningmastery.com/what-is-deep-learning/

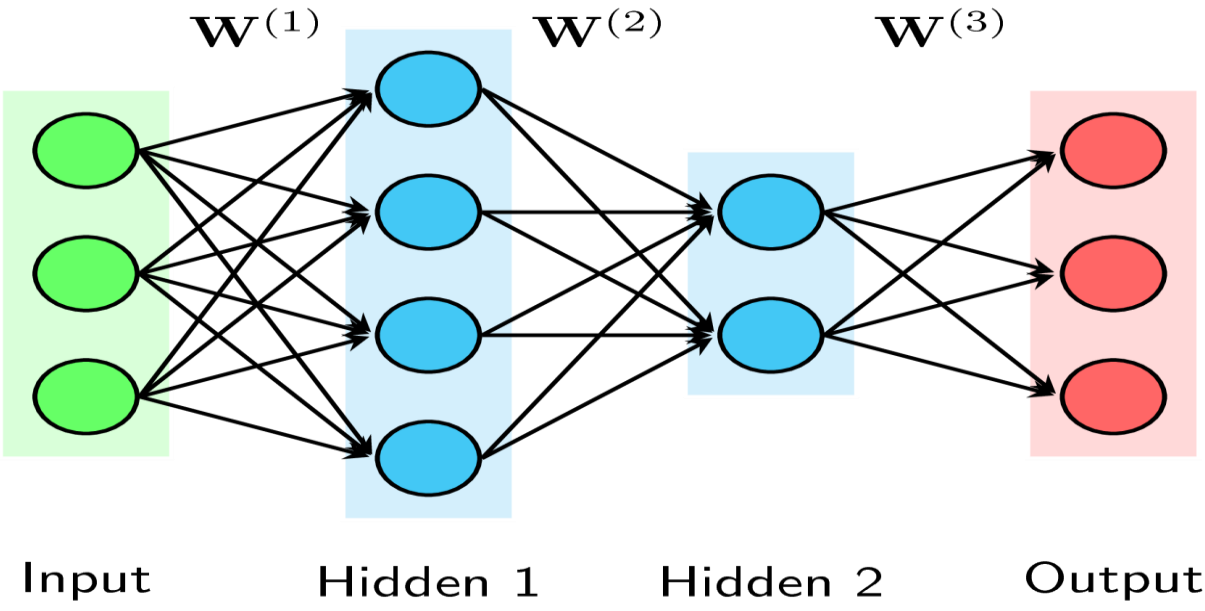
Boltzmann Machines để khởi tạo trọng số của một lớp tại một thời điểm tính toán. Nằm 2010, Erhan et al. giới thiệu thuật toán tham lam khởi tạo trọng số của mạng neural đầy đủ để tối ưu cục bộ tốt hơn (better local optima). Trước đó, Vincent et al. (2008) trình bày cải tiến Deep Learning với autoencoders. Có thể thấy, Deep Learning đang là một chủ đề nghiên cứu rất hot hiện nay với nhiều cải tiến mới.

Phần kế tiếp, chúng ta sẽ tìm hiểu các kỹ thuật xây dựng Deep Learning.

2. Quy trình xây dựng Deep Learning

2.1. Các khái niệm cơ bản

Layers: Như đã biết, neural network có một lớp input, một lớp output. Ngoài ra, neural network còn có một hoặc nhiều lớp hidden layers ở giữa. Các hidden layers theo thứ tự từ input layer đến output layer được đánh số thứ thự là hidden layer 1, hidden layer 2, … Hình dưới đây là một ví dụ với 2 hidden layers.



Hình 4. Neural network với 2 hidden layers[6]

Unit: Một node hình tròn trong một layer được gọi là một unit. Unit ở các input layer, hidden layers và output layer được lần lượt gọi là input unit, hidden unit, và output unit. Đầu vào của các hidden layer được ký hiệu bởi z, đầu ra của mỗi unit thường được ký

hiệu là a (thể hiện *activation*, tức giá trị của mỗi unit sau khi ta áp dụng activation function lên z). Đầu ra của unit thứ i trong layer thứ l được ký hiệu là $a_i^{(l)}$.

Weight và **Bias**: Weight thể hiện mỗi liên kết giữa unit này với unit kia trong neural network. Đầu vào của mỗi unit là một hàm số với hệ số là Weight W và số tự do là Bias b. Khi huấn luyện neural network thì chúng ta đi tìm tập hợp các weights và biases sao cho được hàm error có giá trị nhỏ nhất (tối ưu).

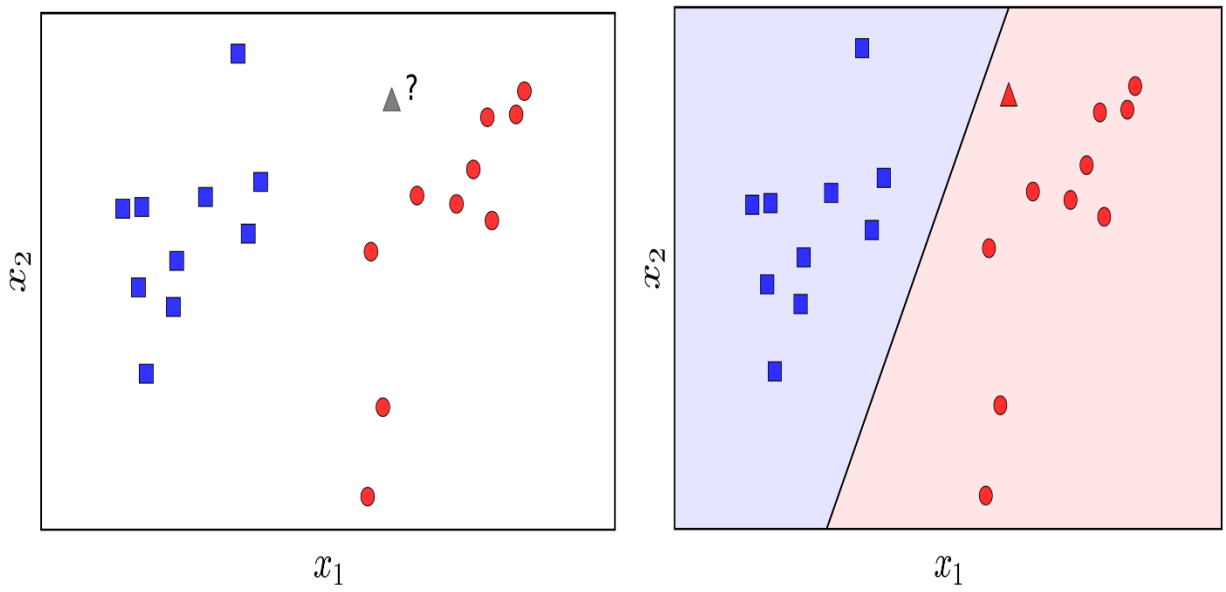
2.2. Các kỹ thuật

Sau khi đã phần nào hiểu được khái niệm cũng như ý tưởng của Deep Learning, chúng ta sẽ đi tìm hiểu cặn kẽ các kỹ thuật xây dựng Deep Learning nói riêng và neural network nói chung. Các kỹ thuật này thường nặng về toán học.

2.2.1. Multi-layer perceptron

Single Perceptron:

Perceptron là một thuật toán phân lớp nhị phân được sử dụng phổ biến trong Machine Learning. Perceptron là nền tảng quan trọng để xây dựng neural network và deep learning. Perceptron được giới thiệu vào năm 1957 bởi Frank Rosenblatt. Để hiểu rõ cơ chế hoạt động của Perceptron, chúng ta tìm lời giải cho bài toán phân lớp đơn giản dưới đây:



Hình 5. Bài toán phân lớp đối tượng Δ đen vào lớp nào

Bài toán trên được giải quyết bằng cách tìm một phương trình đường thẳng phân chia 2 lớp xanh và đỏ với nhau. Sau đó, quan sát xem đối tượng Δ đen thuộc bên phần nào của đường thẳng thì nó thuộc lớp đó. Phương trình đường thẳng có dạng:

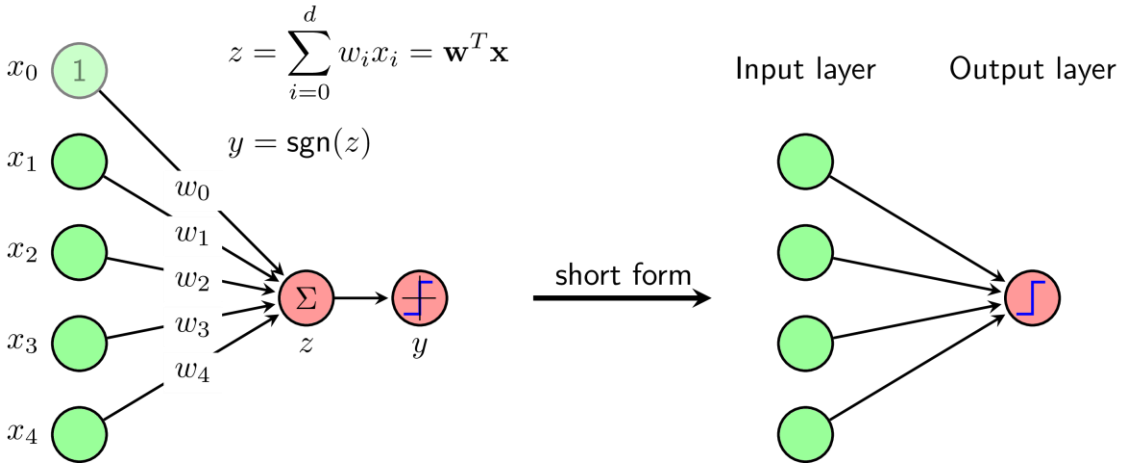
$$f_{\mathbf{w}}(\mathbf{x}) = w_1 x_1 + \cdots + w_d x_d + w_0 = \mathbf{w}^T \bar{\mathbf{x}} = 0$$

Nếu tồn tại một phương trình đường thẳng như vậy, người ta gọi 2 lớp được phân chia là *linearly separable.*

Thuật toán Perceptron được mô tả như sau:

- Bước 1. Chọn ngẫu nhiên một vector hệ số w với các phần tử gần 0.
- Bước 2. Duyệt ngẫu nhiên qua từng điểm dữ liệu x_i:
  - Nếu x_i được phân lớp đúng, chúng ta không cần làm gì.
  - Nếu x_i bị phân lớp sai, cập nhật w theo công thức: $w=w+y_i x_i$
- Bước 3. Kiểm tra xem có bao nhiêu điểm bị phân lớp sai. Nếu không còn điểm nào, dừng thuật toán. Nếu còn, quay lại bước 2.

<u>Biểu diễn Perceptron với neural network đơn giản:</u>

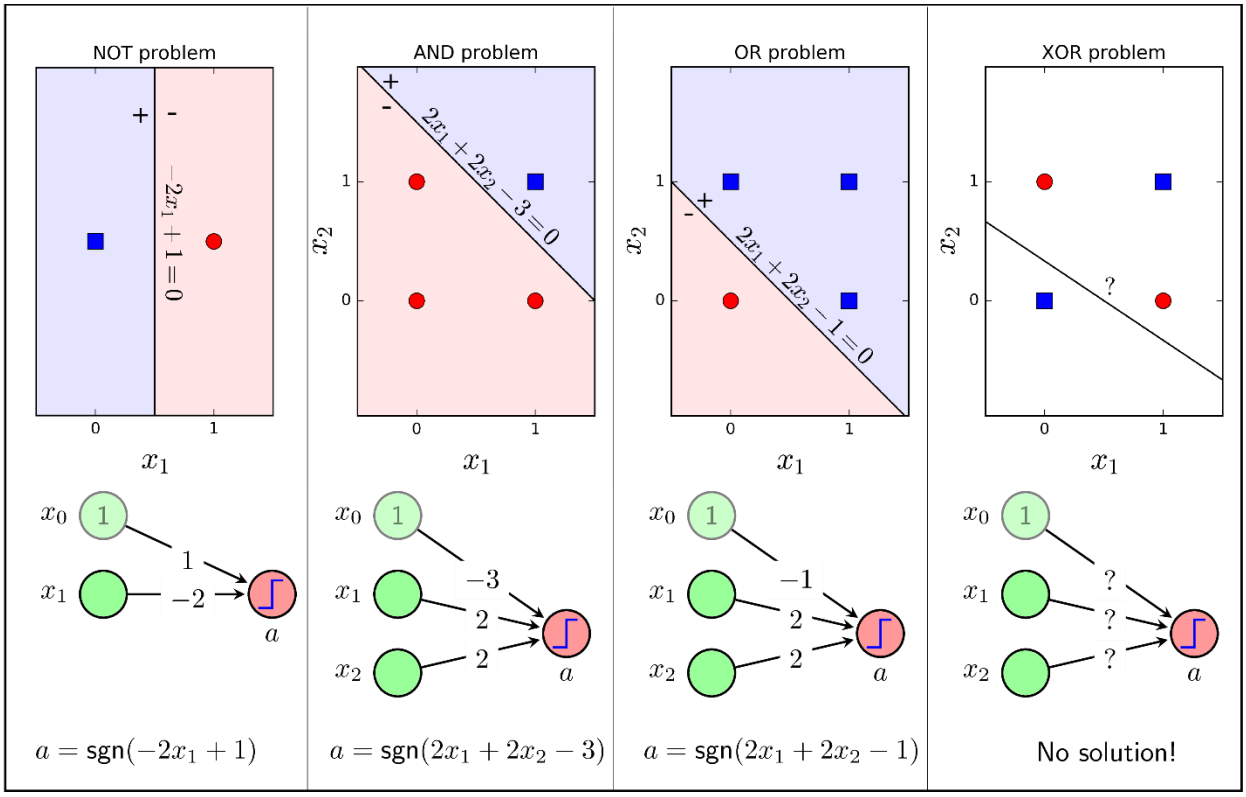


Hình 6. Biểu diễn Perceptron với neural network [6]

Đầu vào của neural network là tập hợp x được minh họa bằng các node màu xanh lục với node x_0 luôn luôn bằng 1. Tập hợp các node màu xanh lục được gọi là *Input layer*. Trong ví dụ này, số chiều của dữ liệu d=4. Số node trong input layer luôn luôn là d+1 với một node là 1 được thêm vào. Node x_0=1 này đôi khi được ẩn đi. Các trọng số (*weights*) $w_0,w_1,\ldots,w_d$ được gán vào các mũi tên đi tới node z = $\sum_{i=0}^{d} w_i x_i = \mathbf{w}^T\mathbf{x}$

Multi-layer Perceptron:

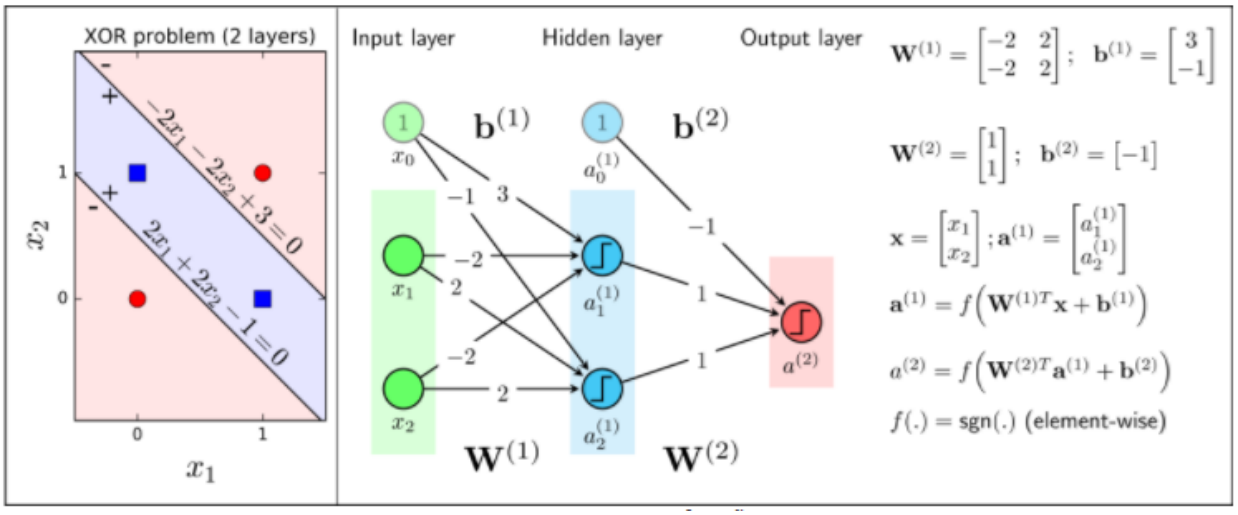
Trong mô hình biểu diễn Single Perceptron, chúng ta thấy neural network chỉ có lớp inputs, output mà không hề có lớp hidden nào. Đó là do bài toán đơn giản, không cần phải qua nhiều lần xử lý để tìm ra output một cách chính xác nhất. Chúng ta hãy xem một ví dụ về biểu diễn các hàm logic thông qua neural network:



Hình 7. Neural network biểu diễn NOT, AND, OR, XOR [6]

Chúng ta thấy các hàm NOT, AND, OR dễ dàng được biểu diễn với neural network như là một Single Perceptron. Tuy nhiên, đối với hàm XOR thì chúng ta không cách nào

biểu diễn được. Chúng ta phải biểu diễn hàm XOR với neural network có một hay nhiều lớp ẩn như sau:



Hình 8. Biểu diễn XOR với neural network [6]

Đối với các bài toán phức tạp, việc tính toán, tìm ra một mô hình cho lượng dữ liệu lớn thì khó tìm được lời giải bằng một phương trình duy nhất. Do đó, chúng ta phải có nhiều giai đoạn tính toán (nhiều lớp ẩn) để tìm ra lời giải cho mô hình bằng cách cập nhật weights và bias ở mỗi giai đoạn tính toán.

2.2.2. Gradient Descent

Multi-layer perceptron là hệ thống cốt lõi của Deep Learning (xử lý qua nhiều giai đoạn, khối lượng dữ liệu lớn). Qua mô hình này, chúng ta phải tìm ra một phương trình (tuyến tính hoặc không tuyến tính) để tìm ra giá trị output chính xác khi biết giá trị input. Trong nhiều trường hợp, một phương trình như thế là không tồn tại. Người ta chuyển sang tìm một phương trình đưa ra các giá trị output xấp xỉ với giá trị output thật – groundtruth. Vậy vấn đề đặt ra ở đây là làm sao tối ưu sự sai lệch kết quả đó, tức là làm cho độ lỗi của mô hình là nhỏ nhất hoặc chấp nhận được. Một trong những kỹ thuật để giải quyết vấn đề trên là Gradient Descent.

Gradient Descent thực hiện với ý tưởng khá đơn giản. Mô hình sự sai lệch kết quả output của mô hình và groundtruth bằng một hàm số f(x). Sau đó, tìm điểm cực tiểu cho

hàm số này. Mô hình neural network trong Deep Learning sẽ liên tục cập nhật để tìm một giá trị x trong hàm số f(x) sao cho f(x) nhỏ nhất: $x_{t+1} = x_t - \eta f'(x_t)$

với η gọi là hệ số học. Nếu hệ số học càng lớn thì hàm số sẽ tiến nhanh đến điểm cực tiểu nhưng sẽ khó hội tụ (không thể đạt đến gần x mà là điểm cực tiểu). Nếu hệ số học nhỏ thì thuật toán sẽ chạy rất lâu (lý giải một phần vì sao Deep Learning đòi hỏi quá trình huấn luyện rất lâu).

Tùy vào từng bài toán, chúng ta sẽ áp dụng các kỹ thuật Gradient Descent cho hàm một biến hay nhiều biến và một số trường hợp chúng ta cần tối ưu phương pháp tìm cực trị bằng cách phương pháp như Momentum, Nesterov accelerated gradient,…

2.2.3. Back-propagation

Back-propagation (lan truyền ngược) là thuật toán chủ đạo trong Deep Learning. Khi áp dụng back-propagation để huấn luyện mô hình, nó gắn liền với một kỹ thuật Gradient Descent. Ý tưởng cơ bản của back-propagation như sau:

- Bước 1. Xây dựng mô hình neural network (inputs, hidden, output).
- Bước 2. Từ các mẫu trong tập training, tính toán giá trị output.
- Bước 3. So sánh và tính độ lỗi kết quả tính toán của mô hình và giá trị thực.
- Bước 4. Sử dụng kỹ thuật Gradient Descent để cập nhật lại các trọng số của mô hình.
- Bước 5. Lặp lại bước Bước 2 đến Bước 4. Nếu độ lỗi ở mức chấp nhận được thì dừng thuật toán.

2.2. Các kiến trúc Deep Learning

2.2.1. **RNN** (Recurrent neural network)

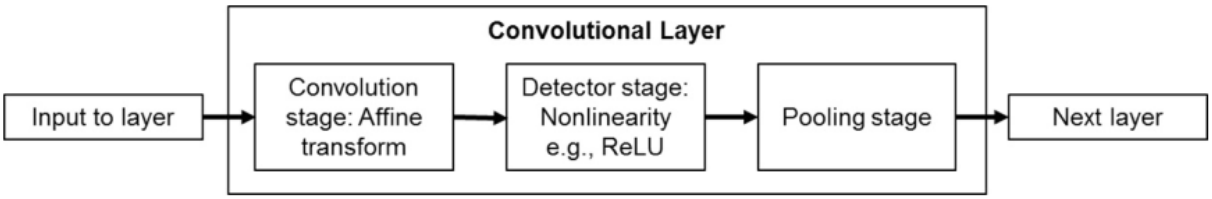
RNN (Recurrent neural network) là một neural network được tạo ra để xử lý dữ liệu tuần tự (sequential data) với giá trị $x^{(1)}, \ldots, x^{(\tau)}$. RNN lần đầu được giới thiệu bởi Rumelhart et al. vào năm 1986.

Đặc điểm của RNN:

- RNN xử lý độ dài input và output một cách linh động.
- RNN sử dụng bộ nhớ để lưu trữ input tùy ý.
- RNN sử dụng thông tin chuỗi thời gian.
- RNN được tạo ra để chủ yếu xử lý văn bản, giọng nói.

2.2.2. CNN (Convolutional neural network)

CNN (Convolutional neural network) là một nhóm nhỏ trong thuật toán Deep Learning, tập trung vào thị giác máy tính và xử lý ảnh. Mô hình CNN được Fukushima giới thiệu lần đầu vào năm 1980. Sau đó được Yann LeCun cải tiến sử dụng kỹ thuật Stochastic Gradient Descent cho nhận dạng chữ viết tay. Hai CNN nổi tiếng là AlexNet và GoogLeNet được Krizhevsky et. al và Szegedy et. al tạo ra. Chúng đều là những CNN với số hidden layers lớn. CNN sử dụng toán tử chập (convolution) giữa các ma trận để huấn luyện trong các ma trận biểu diễn hình ảnh.



Hình 9. Mô hình CNN

Đặc điểm của CNN:

- CNN có input đầu và cố định và tạo ra output cũng cố định.
- CNN là biến thể của mô hình Multi-layer perceptron được thiết kế để hạn chế quá trình tiền xử lý input.
- CNN chủ yếu sử dụng cho xử lý hình ảnh và videos.

3. Ứng dụng

Deep Learning có rất nhiều ứng dụng như nhận dạng giọng nói, xử lý ngôn ngữ tự nhiên, xử ly ảnh, truy tìm thông tin.

Nhận dạng giọng nói: Một trong những ứng dụng của Deep Learning trong lĩnh vực này là trợ lý ảo kích hoạt bằng giọng nói. Siri của Apple nổi tiếng về ứng dụng trợ lý kích hoạt bằng giọng nói (được giới thiệu vào tháng 10/2011). Google Now, trợ lý giọng nói cho Android, được tung ra thị trường sau Siri chưa đầy một năm. Sản phẩm trợ lý giọng nói mới nhất là Microsoft Cortana được giới thiệu vào tháng 4/2014 trên Windows Phone 8.1.

Xử lý ngôn ngữ tự nhiên: Ứng dụng tiêu biểu là những con chatbot. Những con chatbot tự động thu thập những dữ liệu thông qua trò chuyện với mọi người rồi từ từ học hỏi và đưa ra những câu trả lời "như người". Một số chatbot nổi tiếng là Bob trên Skype, Jessie trên facebook, Alexa trên Amazon,…

Xử lý ảnh: Các ứng dụng giúp phân biệt các đối tượng trong một hình ảnh, theo vết nhân vật trong một video sẽ là tương lai của Khoa học máy tính.

Truy tìm thông tin: Ứng dụng nổi tiếng là các hệ thống khuyến nghị của Netflix, Amazon, Google, Facebook, và Twitter. Việc truy cập vào dữ liệu cho phép các doanh nghiệp này có thể triển khai những hệ thống khuyến nghị nhằm cung cấp nhiều giá trị hơn cho cả người dùng và chính họ.

4. Công cụ phát triển (search web)

4.1. Tensorflow

Là một thư viện phần mềm mã nguồn mở dành cho máy học trong nhiều loại hình tác vụ nhận thức và hiểu ngôn ngữ. Với TensorFlow, người dùng không cần phải có kiến thức chuyên sâu về các mô hình toán học và các thuật toán tối ưu để cài đặt được Deep Neural Networks. Người dùng chỉ cần download một vài đoạn code mẫu, đọc qua một vài tutorials online để có thể cài đặt hoàn tất mà không hề tốn quá nhiều thời gian.

4.2. Torch và OverFeat

Torch được viết bằng ngôn ngữ Lua, được sử dụng tại NYU, Facebook AI lab và Google DeepMind. Công cụ này cung cấp một môi trường lập trình tương tự như MATLAB chuyên dùng cho các thuật toán machine learning. Tại sao họ lại sử dụng

Lua/LuaJIT để phát triển thay vì Python? Trong bài báo về Torch7, họ cho rằng "Lua dễ dàng tích hợp với C. Do đó, chỉ trong vài giờ, bất kì thư viện C hay C++ nào cũng đều trở thành thư viện Lua". Với Lua được viết thuần theo ANSI C, mã nguồn có thể dễ dàng được biên dịch cho bất kì đối tượng nào.

Riêng OverFeat là một feature extractor được huấn luyện trên tập dữ liệu ImageNet với Torch7 và dễ dàng để bắt đầu sử dụng.

4.3. Caffe

Caffe được phát triển bởi Berkeley Vision và Learning Center, được viết bởi Yangqing Jia và được dẫn dắt bởi Evan Shelhamer. Đây là một bản cài đặt khá dễ hiểu giúp cài đặt nhanh ConvNets bằng C++. Đánh giá hiệu suất ở trang chủ cho thấy, Caffe có thể xử lý hơn 60 triệu bức ảnh mỗi ngày chỉ với một card đồ họa NVIDIA K40 với AlexNet. Ta có thể sử dụng nó như một toolkit cho tác vụ phân lớp ảnh. Tuy nhiên, toolkit này khó áp dụng cho các bài toán bên xử lý văn bản và tiếng nói.

TÀI LIỆU THAM KHẢO

[1] Ian Goodfellow, Yoshua Bengio, Aaron Courville. *Deep Learning*. The MIT Press, 2016.

[2] LISA lab. *Deep Learning Tutorial*. University of Montreal, 2015.

[3] Niklas Barkmeyer. *Learning to recognize new objects using deep learning and contextual information*. Technische Universit¨at M¨unchen Institute for Cognitive Systems (ICS), 2016.

[4] Richard Socher. *Recursive Deep Learning For Natural Language Processing And Computer Vision*. Stanford University, 2014.

[5] Li Deng, Dong Yu. *Deep Learning: Methods and Applications*. Now Publishers Inc, 2013.

[6] https://machinelearningcoban.com (Truy cập 11/11/2017).