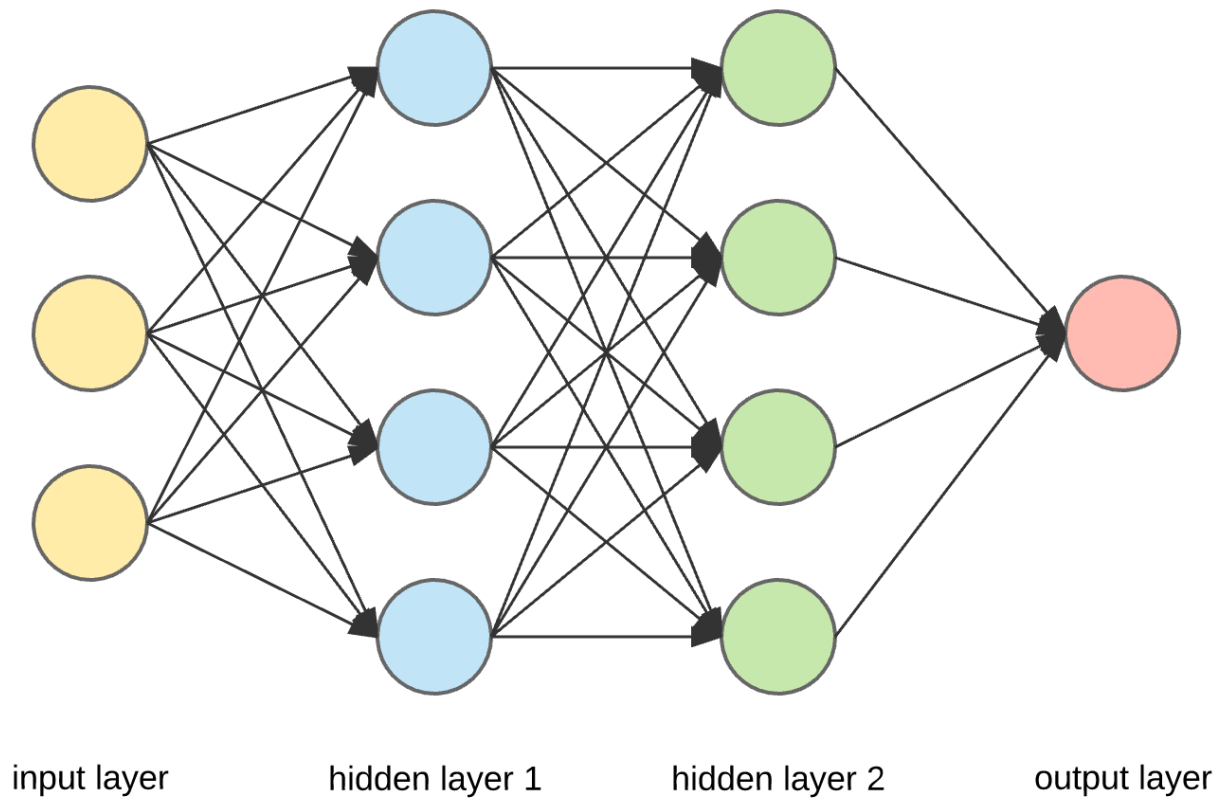


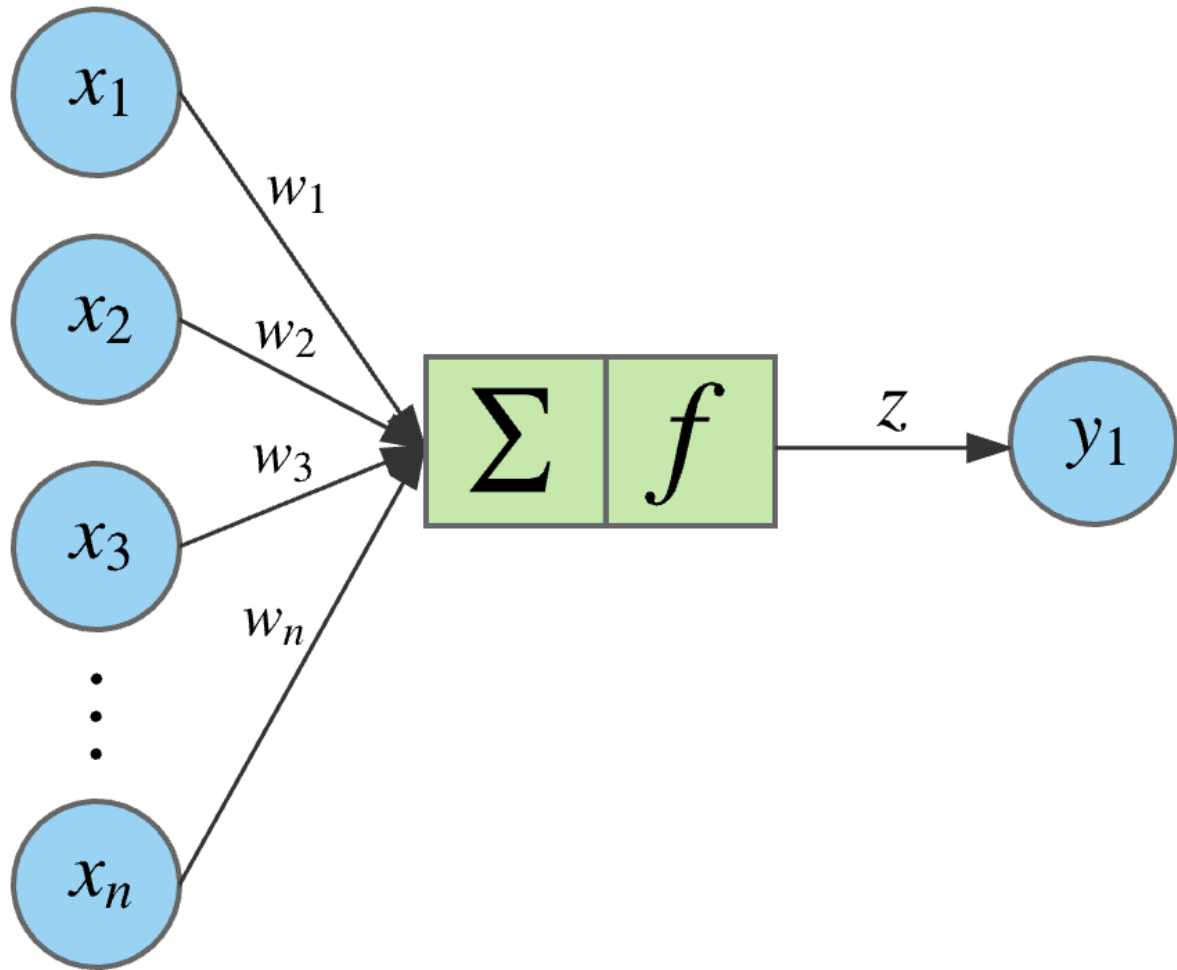
ANN (Artificial Neural Network)

1. Introduction

- ANN là “multi-layer fully-connected neural nets” như hình dưới
- Nó bao gồm:
 - + Một input layer.
 - + Nhiều hidden layer.
 - + Một output layer.
- Mỗi node trong một layer được kết nối với từng node một ở một layer khác. Ta làm mạng “sâu hơn” nhờ vào việc tăng số lượng hidden layer.



- Chi tiết hơn, ta có hình sau:



- Ta lấy tổng các tích của trọng số ứng với các input của nó và đưa nó qua một hàm kích hoạt phi tuyến tính.

=> Sau khi đưa qua hàm kích hoạt phi tuyến tính rồi, thì đó sẽ là output của Hidden Layer (ở giữa hình), và trở thành input của node khác ở layer tiếp theo.

(*) LUỒNG HOẠT ĐỘNG CHẠY TỪ TRÁI SANG PHẢI.

=> Huấn luyện “Deep Neural Network” này tức cho mạng học các trọng số ở tất cả các cạnh.

- Ta có phương trình sau:

$$z = f(x \cdot w) = f \left(\sum_{i=1}^n x_i w_i \right)$$

$$x \in d_{1 \times n}, w \in d_{n \times 1}, z \in d_{1 \times 1}$$

=> Phương trình trên tức ta đưa tổng các tích giữa trọng số và đầu vào tương ứng với nó qua một hàm phi tuyến tính. (n: số lượng input/ số lượng đặc trưng của một điểm dữ liệu)

- Tổng quát hơn:

$$z = f(b + x \cdot w) = f \left(b + \sum_{i=1}^n x_i w_i \right)$$

$$x \in d_{1 \times n}, w \in d_{n \times 1}, b \in d_{1 \times 1}, z \in d_{1 \times 1}$$

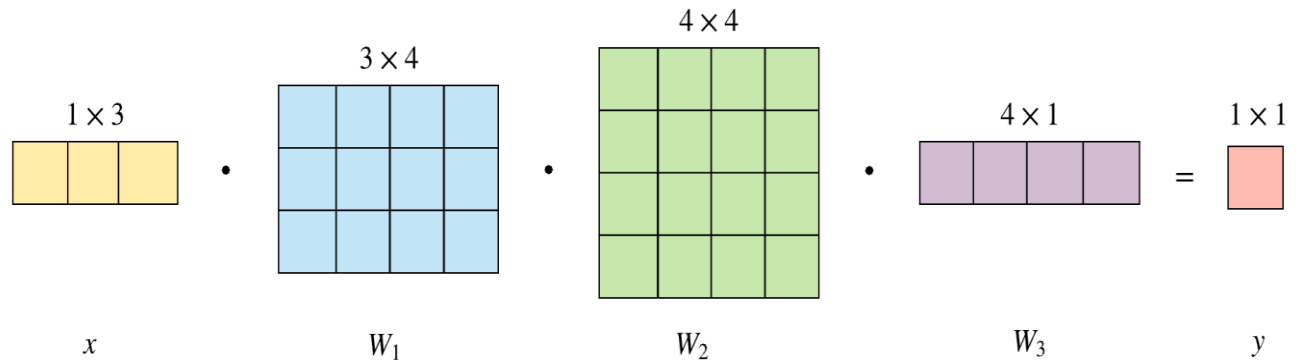
2. Intuition

- Sử dụng hình ANN đầu tiên

+ Ta có dữ liệu đầu vào có 3 chiều, tương đương với một vector trong không gian 3 chiều. Sau đó, ta cho nó qua 2 Hidden Layer với mỗi Hidden Layer có 4 Nodes. Và output cuối cùng là một vector 1 chiều hoặc 1 số vô hướng.

- Nếu ta hình dung đây là một chuỗi các biến đổi vector, đầu tiên ta sẽ map dữ liệu đầu vào 3 chiều sang một không gian vector 4 chiều, sau đó ta lại biểu diễn phép biến đổi khác tới một không gian 4 chiều mới.

=> Đây đơn giản chỉ là CHUỖI PHÉP NHÂN MA TRẬN.



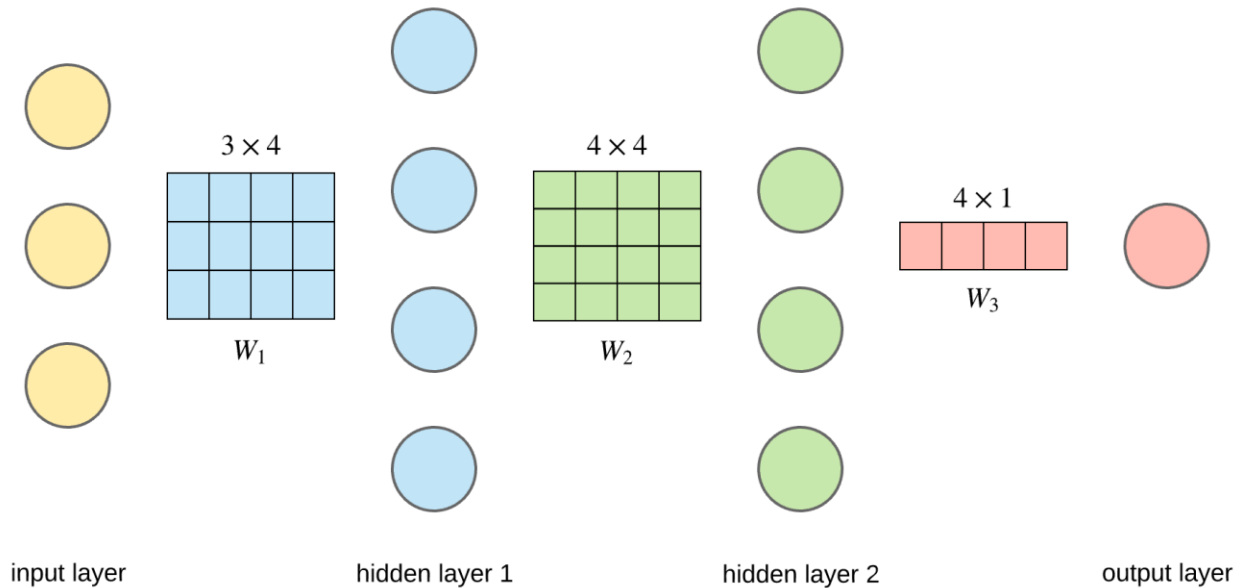
- Ở hình trên, ta thấy:

- + Vector đầu vào x có 1 hàng và 3 cột. (1)
- + Để biến đổi nó sang một không gian 4 chiều, ta cần nhân nó với một ma trận 3×4 . (2)
- + Tiếp theo, để chuyển sang một không gian 4 chiều khác, ta nhân đầu ra của (2) với một ma trận 4×4 (3).
- + Và cuối cùng để giảm kết quả cuối cùng xuống thành 1 chiều, ta cần nhân đầu ra của (3) với một ma trận 4×1 .

(*) LƯU Ý:

- Ta thấy kết nối giữa một layer có 3 nodes và một layer có 4 nodes là một phép nhân ma trận sử dụng ma trận 3×4 . Ma trận này đại diện cho các trọng số định nghĩa ANN – Để dự đoán đầu ra bằng cách sử dụng ANN và đầu vào đã cho, ta chỉ cần biết các trọng số này (WEIGHTS) và hàm kích hoạt (bao gồm cả BIAS).

- Ta có hình sau:



=> Ta đã bỏ qua hàm kích hoạt ở hình trên. Trong thực tế, sau mỗi phép nhân ma trận, ta áp dụng hàm kích hoạt vào mỗi phần tử của ma trận kết quả:

$$a_1 = f(x \cdot W_1)$$

$$a_2 = f(a_1 \cdot W_2)$$

$$y = f(a_2 \cdot W_3)$$



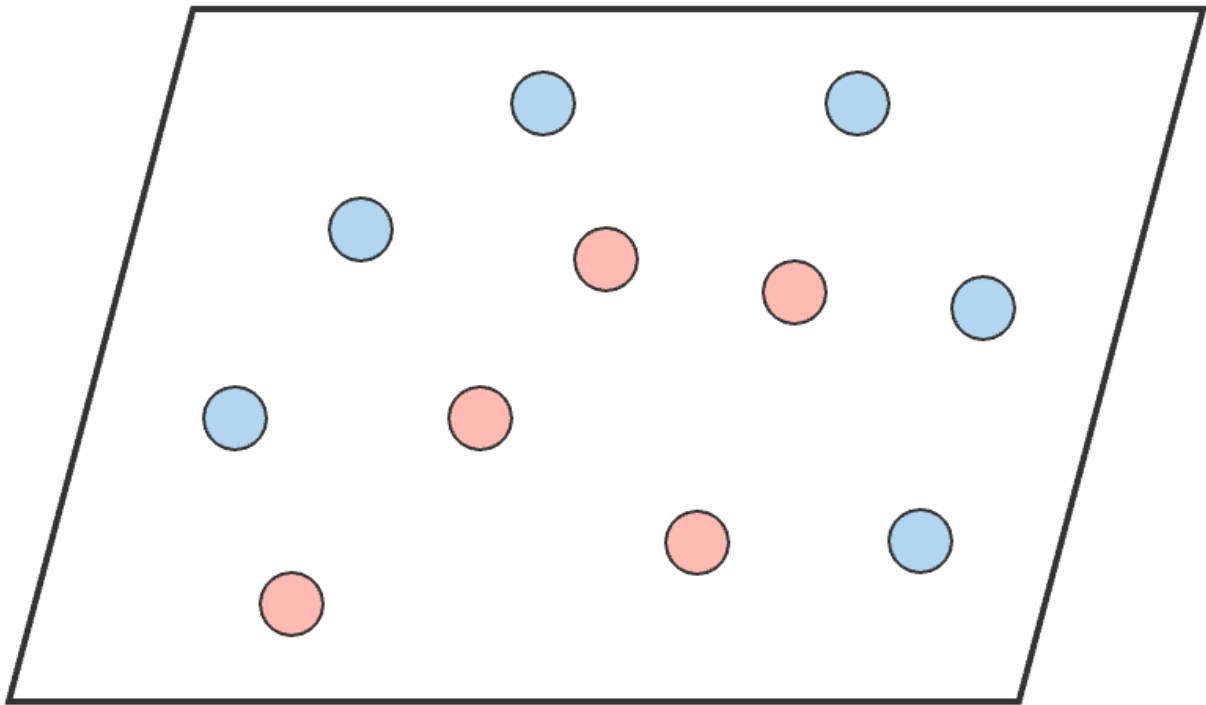
$$y = f(f(f(x \cdot W_1) \cdot W_2) \cdot W_3)$$

3. Reasoning

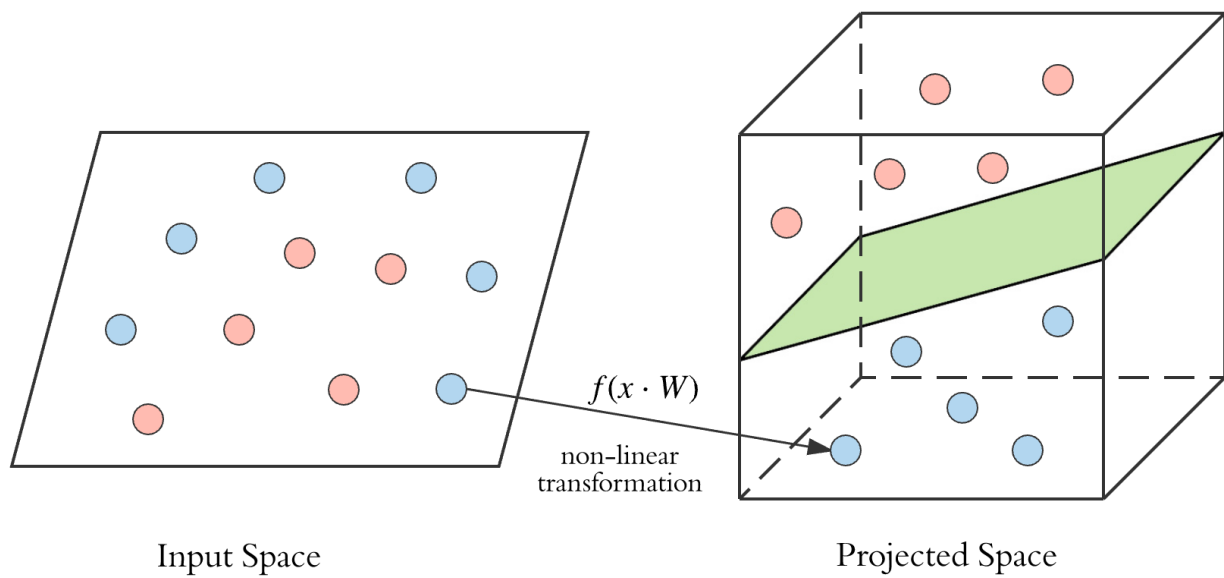
- Một layer của ANN chỉ biểu diễn các phép biến đổi phi tuyến tính (đưa input từ không gian gốc sang không gian khác).

- Lấy ví dụ về phân lớp, ta muốn phân lớp dữ liệu bằng việc vẽ một đường biên (Decision Boundary). Dữ liệu đầu vào (chưa qua xử lý) thì không thể được phân chia => Bằng việc biểu diễn các phép biến đổi phi tuyến tính ở mỗi layer, ta sẽ có thể chuyển input sang chiều không gian mới và vẽ đường biên phức tạp để phân lớp.

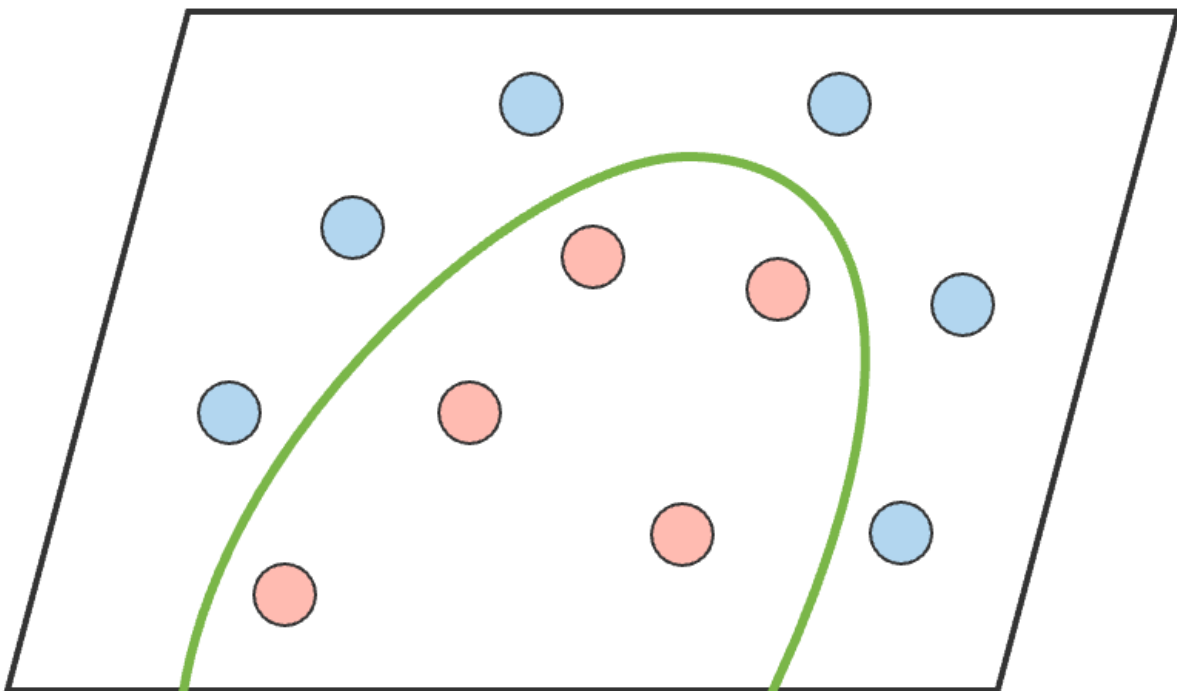
- VD: Cung cấp dữ liệu đầu vào mà không thể phân chia tuyến tính như hình



- Để giải quyết việc phân lớp ở hình trên, ta biểu diễn phép biến đổi phi tuyến tính và các điểm dữ liệu trở thành có thể phân chia tuyến tính.



- Điều này tương đương với việc vẽ ra một đường biên phức tạp trên không gian gốc (không gian ban đầu) của dữ liệu đầu vào như hình dưới:



(*) Vậy, lợi ích chính của việc có một mô hình sâu hơn (A Deeper Model) là để có thể thực hiện nhiều phép biến đổi phi tuyến tính trên dữ liệu đầu vào và vẽ nhiều đường biên phức tạp (Complex Decision Boundary).

TÀI LIỆU THAM KHẢO

<https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6>