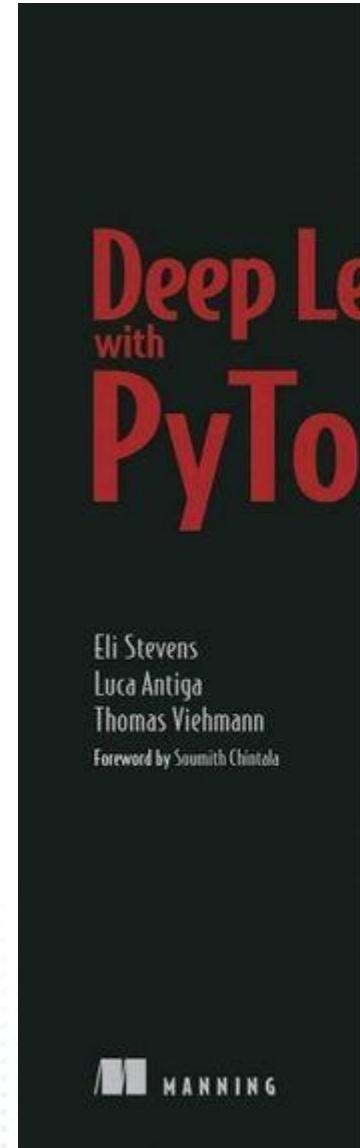


# AI 스터디

조상구  
2025.12.27



# *Using a neural network to fit the data*

---

## ***This chapter covers***

- Nonlinear activation functions as the key difference compared with linear models
- Working with PyTorch's `nn` module
- Solving a linear-fit problem with a neural network

## Our mental model of the learning process

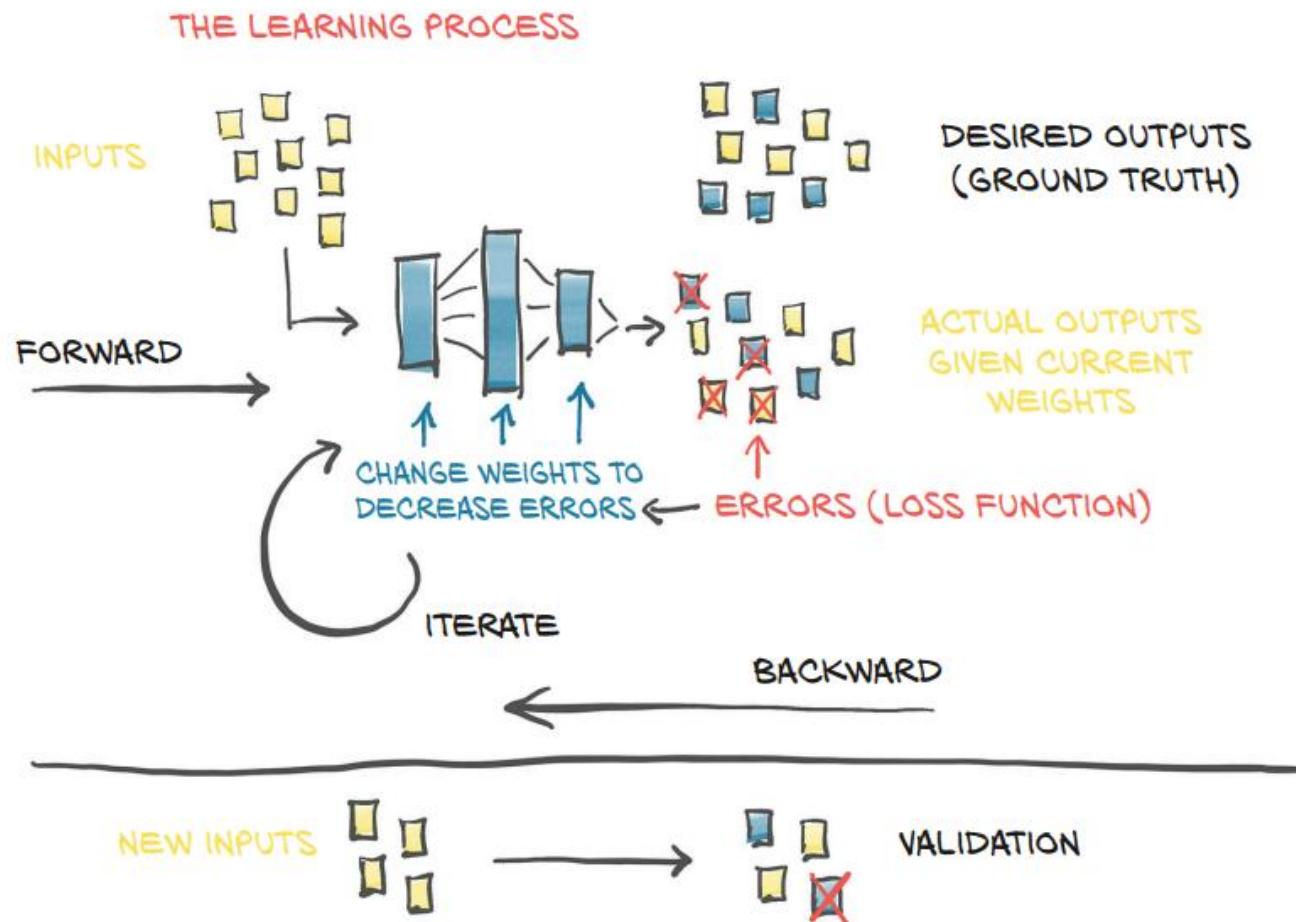
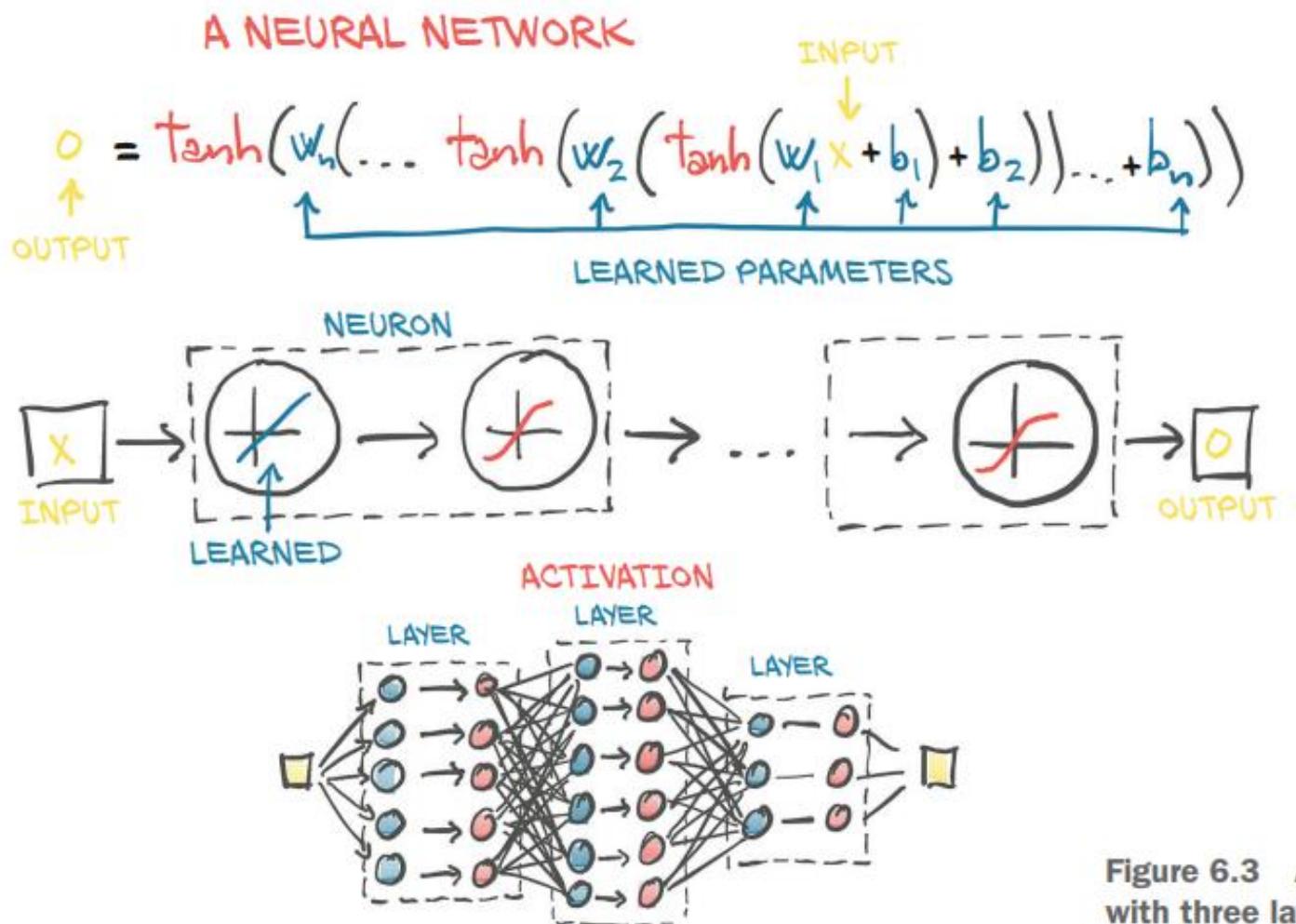


Figure 6.1 Our mental model of the learning process, as implemented in chapter 5

# A neural network with three layers



**Figure 6.3** A neural network with three layers

# 1. 변수 정의

## 1. 변수 정의 (Definition of Variables)

각 층의 가중치 행렬과 편향 벡터의 크기를 정의한다.

모든 층의 노드 수는 2개이므로, 모든 가중치 행렬은  $2 \times 2$  크기를 가진다.

### 입력 벡터

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

### 가중치 행렬 ( $W$ )

각 행은 해당 층 뉴런의 가중치를 의미한다.

$$W_l = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

### 편향 벡터 ( $b$ )

각 뉴런에 더해지는 상수항이다.

$$b_l = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

## 2. 층별 선형대수 계산식 (Layer-by-Layer Computation)

활성화 함수로는  $\tanh(\cdot)$ 를 사용한다.

### ◆ Layer 1 (첫 번째 은닉층)

입력 벡터  $x$ 가 첫 번째 가중치 행렬  $W_0$ 와 곱해진 후 편향이 더해진다.

$$z_1 = W_0x + b_0 = \begin{bmatrix} w_{0,11} & w_{0,12} \\ w_{0,21} & w_{0,22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{0,1} \\ b_{0,2} \end{bmatrix}$$

활성화 함수 적용:

$$x_1 = \tanh(z_1)$$

## 2. 층별 선형대수 계산식 (Layer-by-Layer Computation)

- ◆ Layer 2 (두 번째 은닉층)

이전 층의 출력  $x_1$ 이 입력으로 사용된다.

$$z_2 = W_1x_1 + b_1 = \begin{bmatrix} w_{1,11} & w_{1,12} \\ w_{1,21} & w_{1,22} \end{bmatrix} \begin{bmatrix} x_{1,1} \\ x_{1,2} \end{bmatrix} + \begin{bmatrix} b_{1,1} \\ b_{1,2} \end{bmatrix}$$

활성화 함수 적용:

$$x_2 = \tanh(z_2)$$

- ◆ Layer 3 (출력층)

최종 출력을 계산한다.

$$z_3 = W_2x_2 + b_2 = \begin{bmatrix} w_{2,11} & w_{2,12} \\ w_{2,21} & w_{2,22} \end{bmatrix} \begin{bmatrix} x_{2,1} \\ x_{2,2} \end{bmatrix} + \begin{bmatrix} b_{2,1} \\ b_{2,2} \end{bmatrix}$$

출력층 활성화 함수:

$$y = \tanh(z_3)$$

