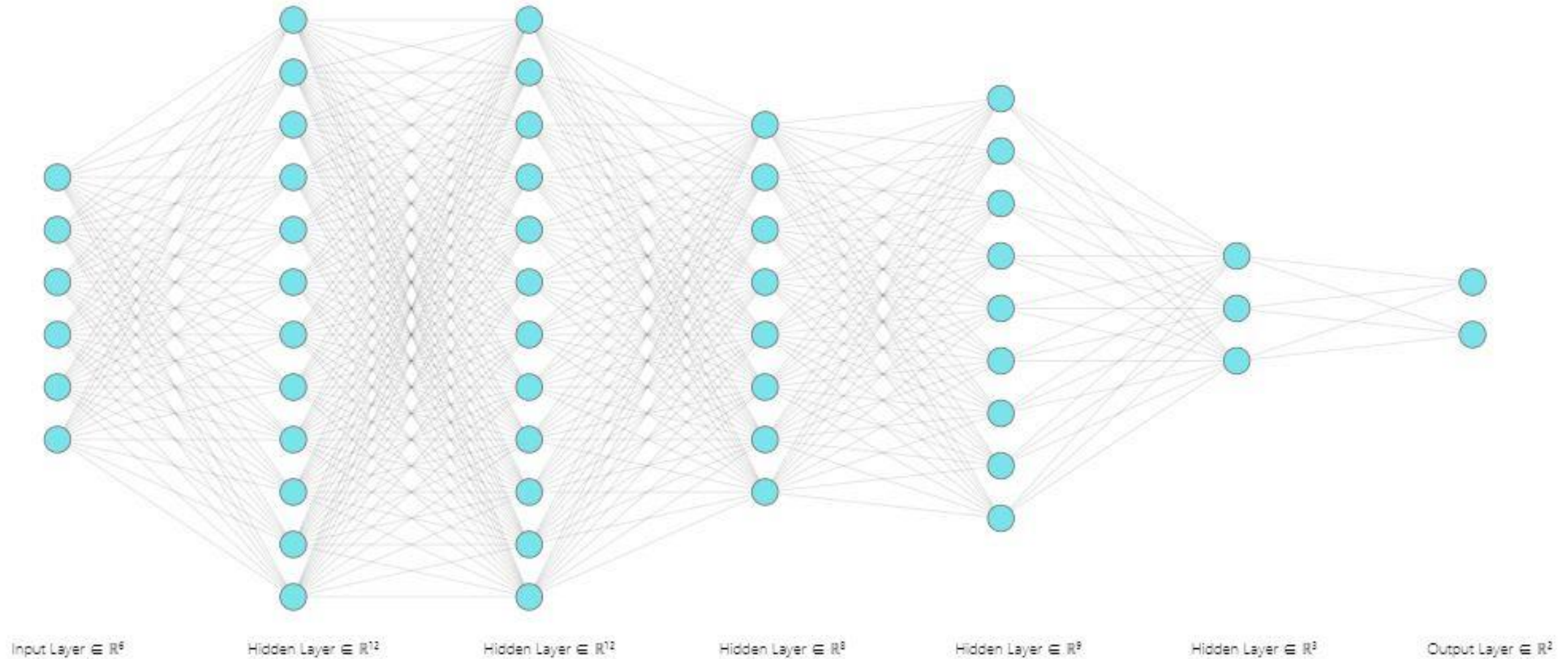


# 딥러닝 강의 01

## 기본 레이어

# Deep Neural Network

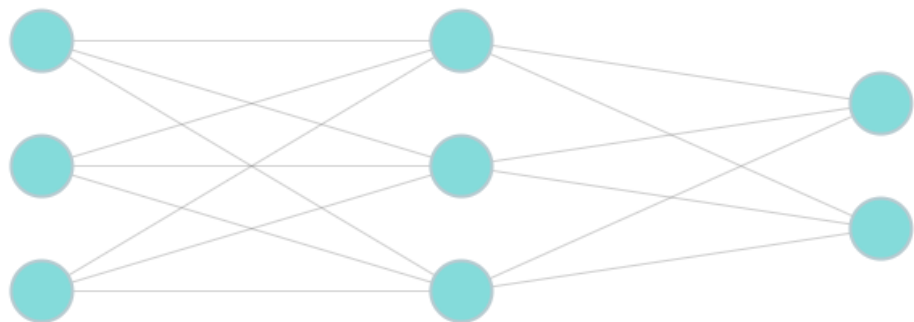


# 읽을거리(케라스 창시자에게 배우는 딥러닝 중)

- 딥러닝은 머신 러닝의 특정한 한 분야로서 연속된 층(layer)에서 점진적으로 의미 있는 표현을 배우는 데 강점이 있으며, 데이터로부터 표현을 학습하는 새로운 방식입니다. 딥러닝의 딥(deep)이라는 단어는 연속된 층으로 학습을 한다는 개념을 뜻하며 층 기반 표현학습 (layered representations learning) 또는 계층적 표현 학습 (hierarchical representation learning) 이라고도 얘기할 수 있습니다. 모델을 만드는데 얼마나 많은 층을 사용했는지가 그 모델의 깊이가 되며, 최근의 딥러닝 모델은 표현 학습을 위해 수십개, 수백개의 연속된 층을 가지고 있습니다.
- 딥러닝은 머신러닝의 오래된 하위 분야이지만 2010년 초가 되어서야 유명해졌으며, 사람에게 자연스럽고 직관적으로 보이지만 기계로는 오랫동안 해결하기 어려웠던 시각과 청각 같은 지각 문제에서 괄목할 만한 성과를 냈습니다. (이미지 분류, 음성 인식, 필기 인식, 기계 번역, 자율 주행 능력, 사람을 능가하는 바둑 실력, 질문에 대답하는 능력 등)
- 딥러닝이 최근 몇년간 놀라운 성과를 이끌어 냈지만 아직 일반 지능에 관한 이야기는 멀었으며 딥러닝의 단점을 보완하지 못한다면 오랫동안 정체될 것입니다. 과거 1960년에도 심볼릭 AI를 이용해 일반 지능을 구현할 수 있을 거라 믿었지만 이는 아직도 해결하기 어려운 일이 되었습니다. 우리는 현재 1960년대에 첫번째 겨울과 1990년의 두번째 겨울 이후 세번째 AI의 부흥기에 있다고 할 수 있습니다.

# 네트워크 시각화

데이터만 시각화

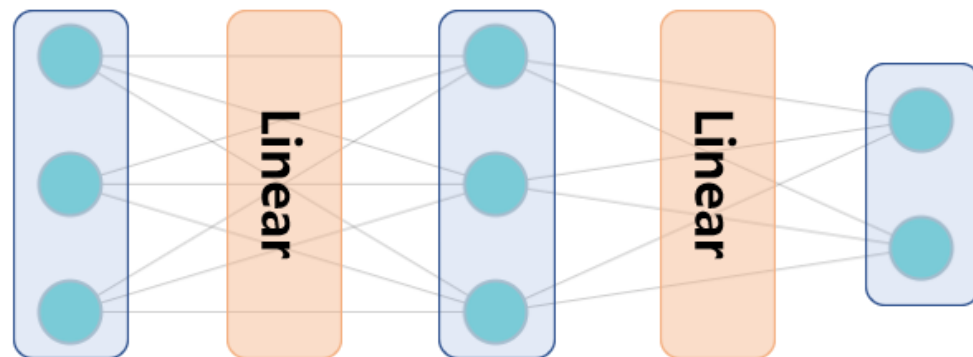


Input Layer  $\in \mathbb{R}^3$

Hidden Layer  $\in \mathbb{R}^3$

Output Layer  $\in \mathbb{R}^2$

데이터와 레이어 시각화

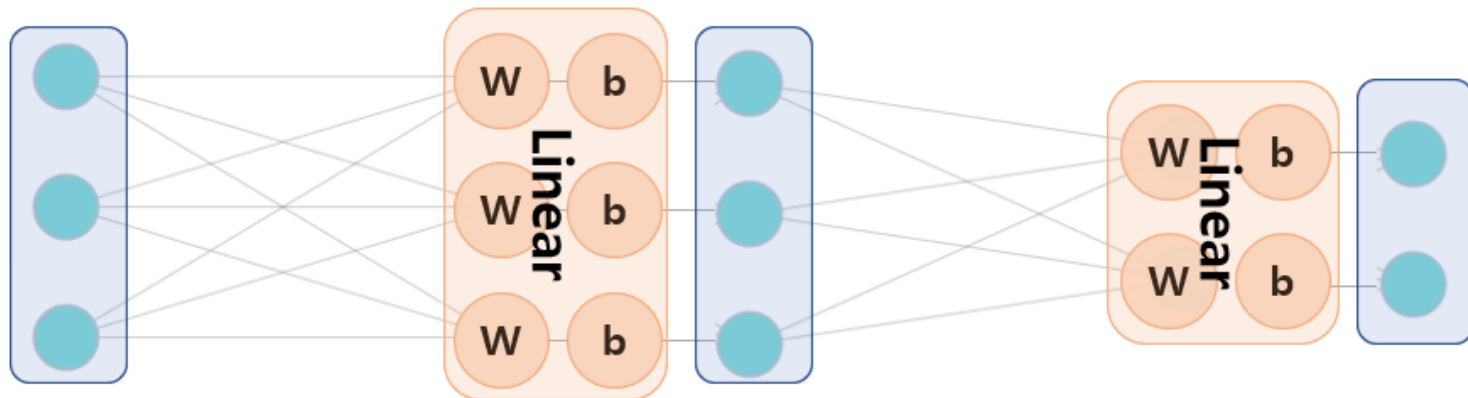


Input Layer  $\in \mathbb{R}^3$

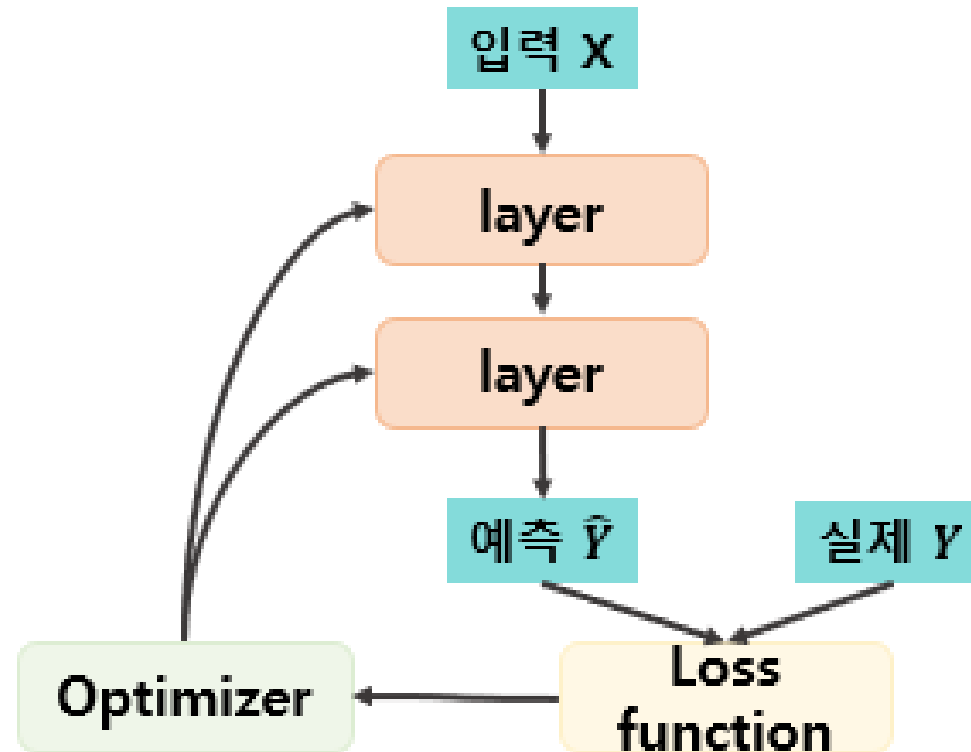
Hidden Layer  $\in \mathbb{R}^3$

Output Layer  $\in \mathbb{R}^2$

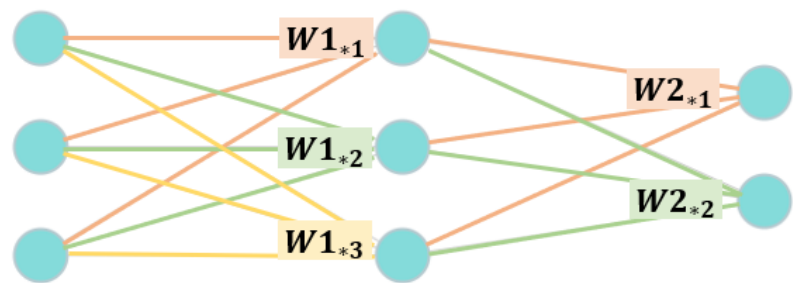
데이터와 자세한 레이어 연산 시각화



# 딥러닝 알고리즘의 전체 흐름



# 선형함수 (Linear Function)



Input Layer  $\in \mathbb{R}^3$

Hidden Layer  $\in \mathbb{R}^3$

Output Layer  $\in \mathbb{R}^2$

$$X$$

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

$$W1$$

$$\begin{bmatrix} 1 & 1 & -1 \\ 1 & 0.5 & -1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$B1$$

$$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

$$W2$$

$$\begin{bmatrix} 1 & 0.5 \\ -1 & 0.5 \\ 0 & 1 \end{bmatrix}$$

$$B2$$

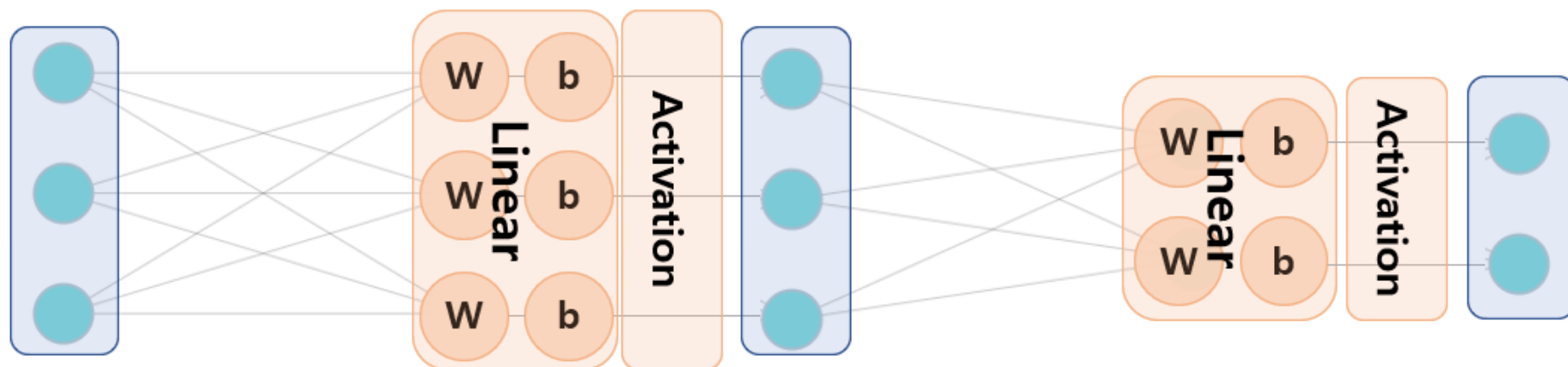
$$\begin{bmatrix} 1 & -0.5 \end{bmatrix}$$

$$Y$$

$$\begin{bmatrix} y_1 & y_2 \end{bmatrix}$$

순전파( forward-propagation, feed-forward) 계산

# 활성화 함수 (Activation Function)



**활성화 함수(Activation function)**이란 입력된 데이터의 가중 합을 출력 신호로 변환하는 함수이다. 인공 신경망에서 이전 레이어에 대한 가중 합의 크기에 따라 활성 여부가 결정된다. 신경망의 목적에 따라, 혹은 레이어의 역할에 따라 선택적으로 적용한다.

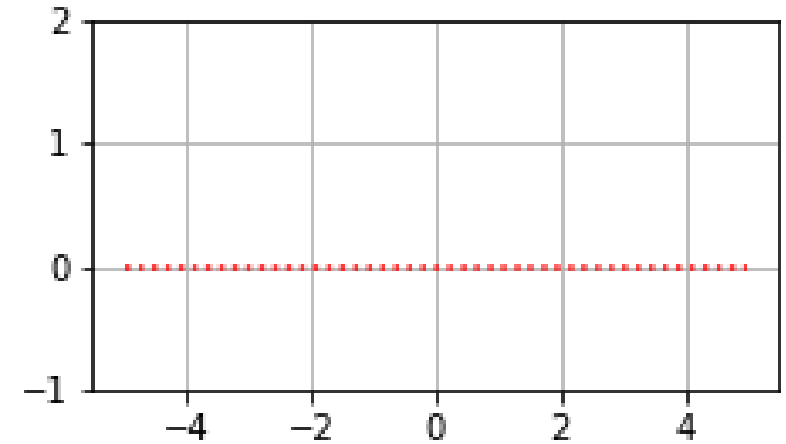
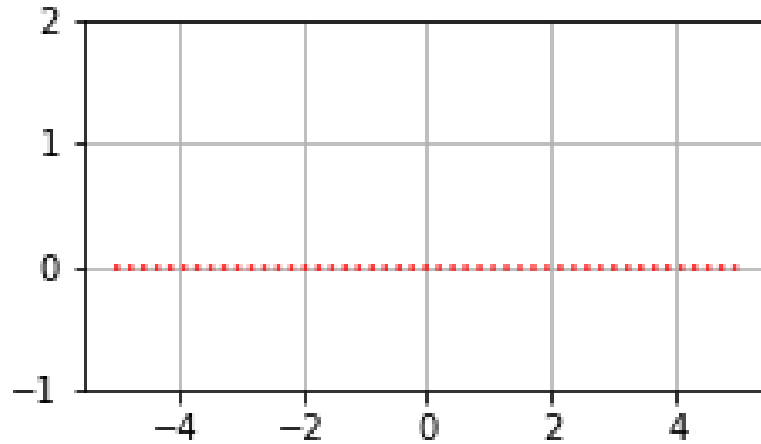
- **활성화 함수를 사용하는 이유**

선형함수인  $h(x)=cx$ 를 활성화함수로 사용한 3층 네트워크를 떠올려 보세요. 이를 식으로 나타내면  $y(x) = h(h(h(x)))$ 가 됩니다. 이는 실은  $y(x) = ax$ 와 똑같은 식입니다.  $a = c^3$ 이라고만 하면 끝이죠. 즉, 은닉층이 없는 네트워크로 표현할 수 있습니다. 뉴럴네트워크에서 층을 쌓는 혜택을 얻고 싶다면 활성화함수로는 반드시 비선형 함수를 사용해야 합니다. [밑바닥부터 시작하는 딥러닝]

# 활성화 함수의 종류 - Sigmoid

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

미분 계산

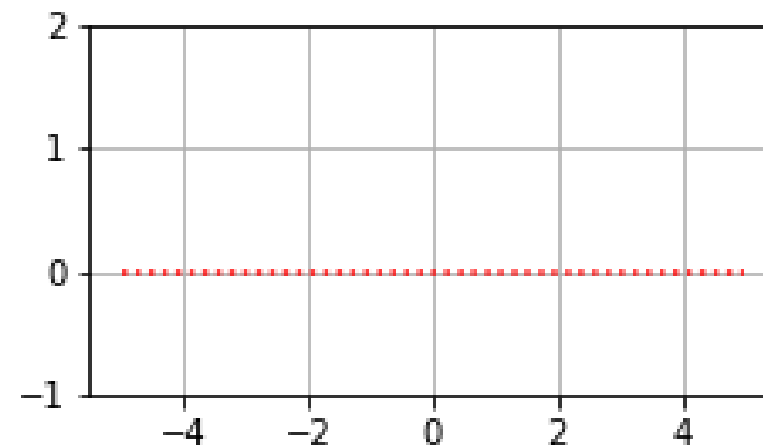
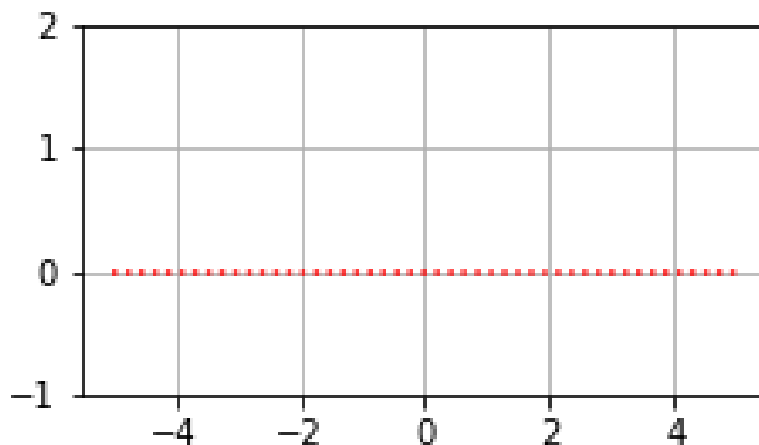




# 활성화 함수의 종류 - Softmax

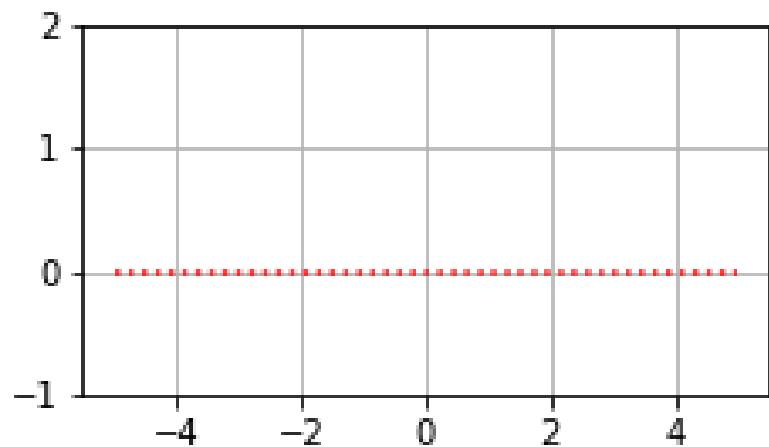
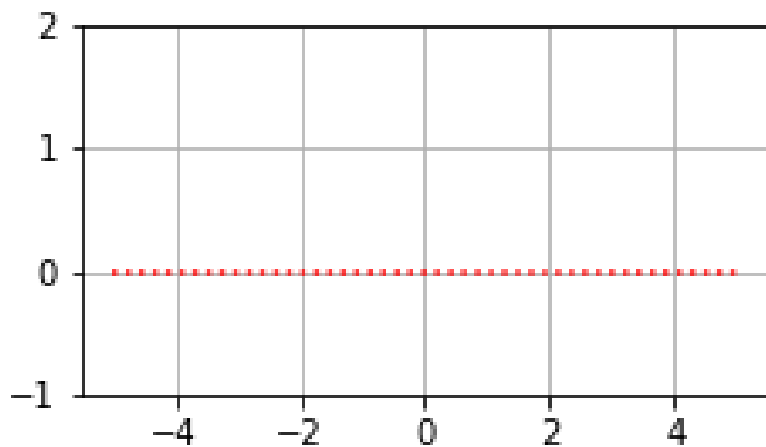
$$\text{softmax}(x) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

미분 계산



# 활성화 함수의 종류 - Tanh

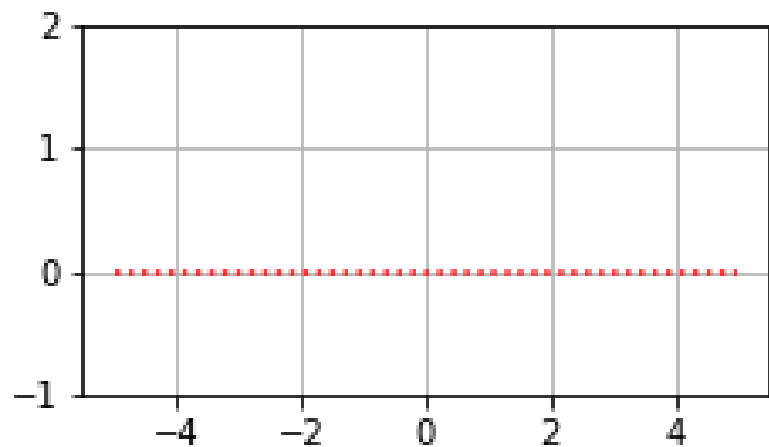
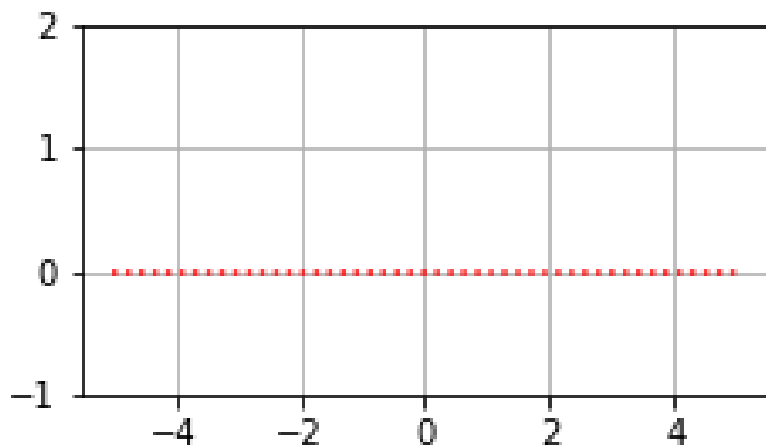
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



미분 계산

# 활성화 함수의 종류 - ReLU

$$\text{ReLU}(x) = \max(0, x)$$

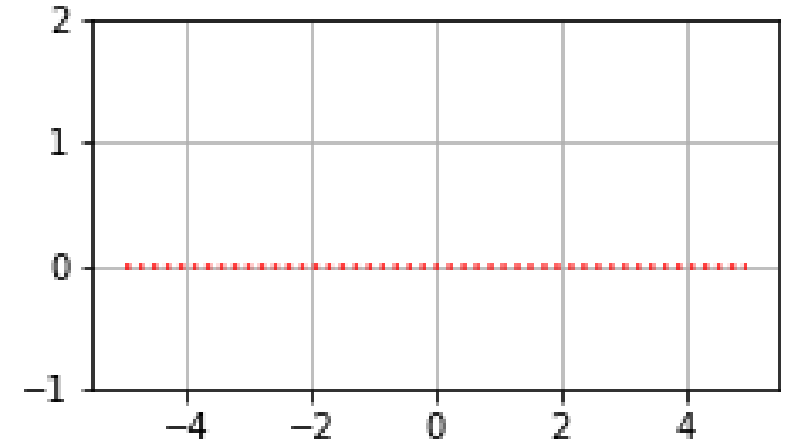
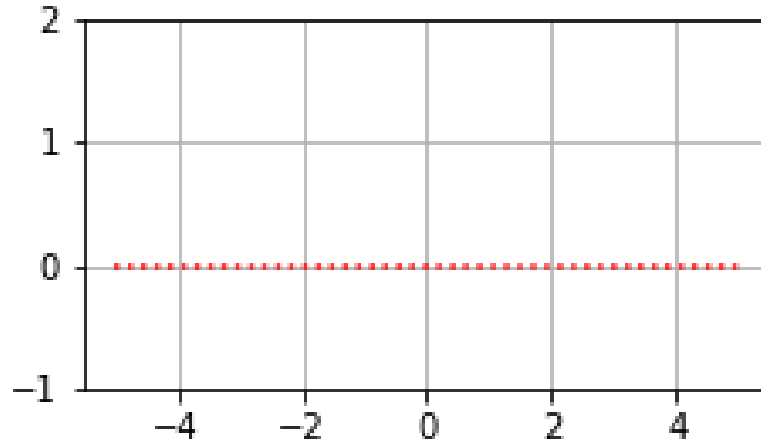


미분 계산

# 활성화 함수의 종류 - LeakyReLU & PReLU

$$\text{LeakyReLU}(x) = \max(0.01x, x)$$

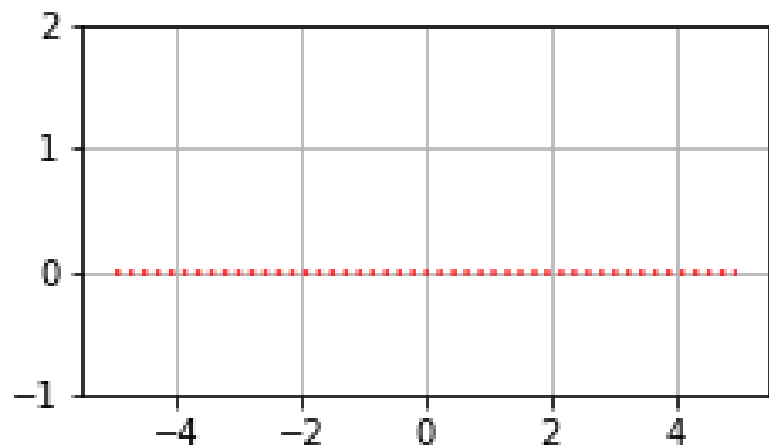
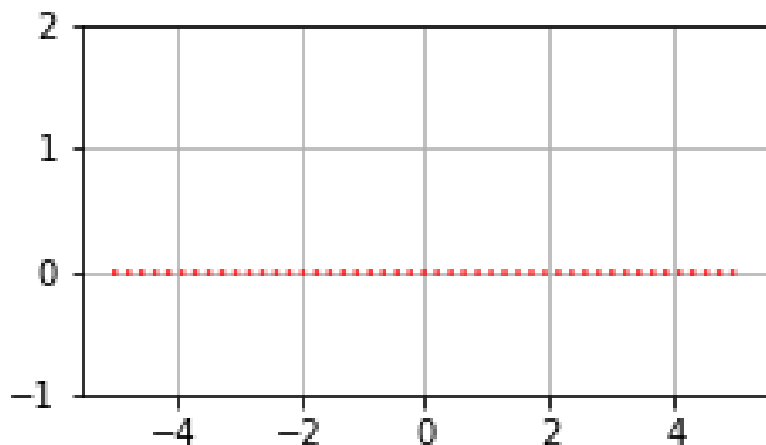
$$\text{PReLU}(x; \alpha) = \max(\alpha x, x)$$



미분 계산

# 활성화 함수의 종류 - ELU & SELU

$$ELU(x) = \begin{cases} x & \text{if } x \geq 0 \\ (e^x - 1) & \text{if } x < 0 \end{cases}$$



$$SELU(x; \alpha) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha(e^x - 1) & \text{if } x < 0 \end{cases}$$

미분 계산