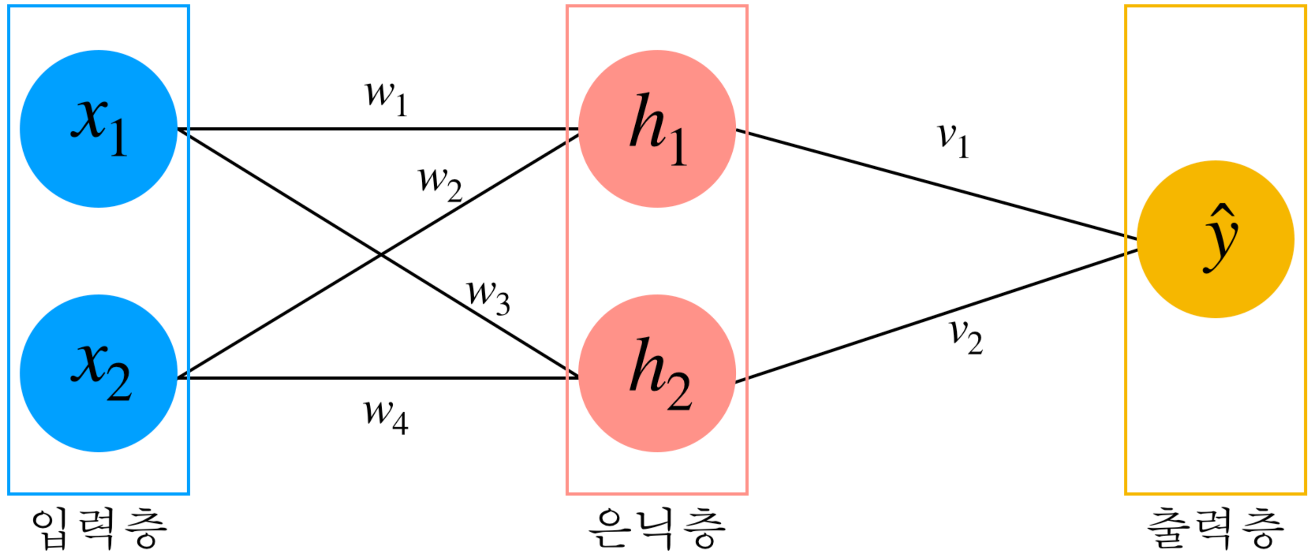
딥러닝 올인원

활성화 함수
4강

딥러닝호형

활성화 함수(Activation Function)

인공 신경망의 일차 결합 형태

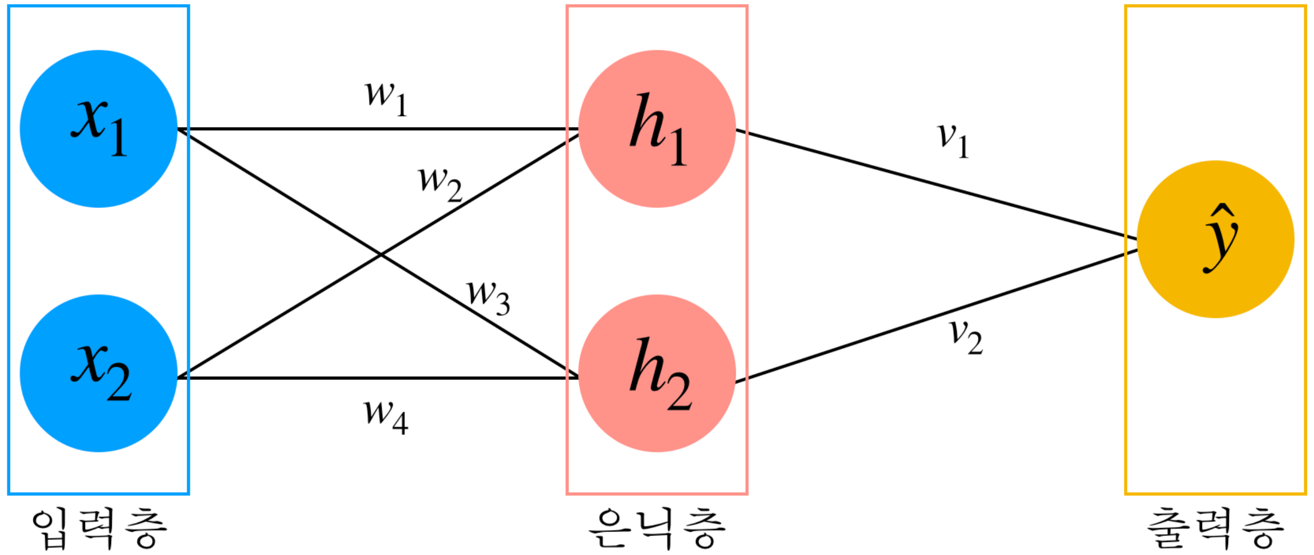


$$h_1 = w_1x_1 + w_2x_2 + b_1,\ h_2 = w_3x_1 + w_4x_2 + b_2$$

$$\hat{y} = v_1h_1 + v_2h_2 + b_3$$

활성화 함수(Activation Function)

인공 신경망의 일차 결합 형태



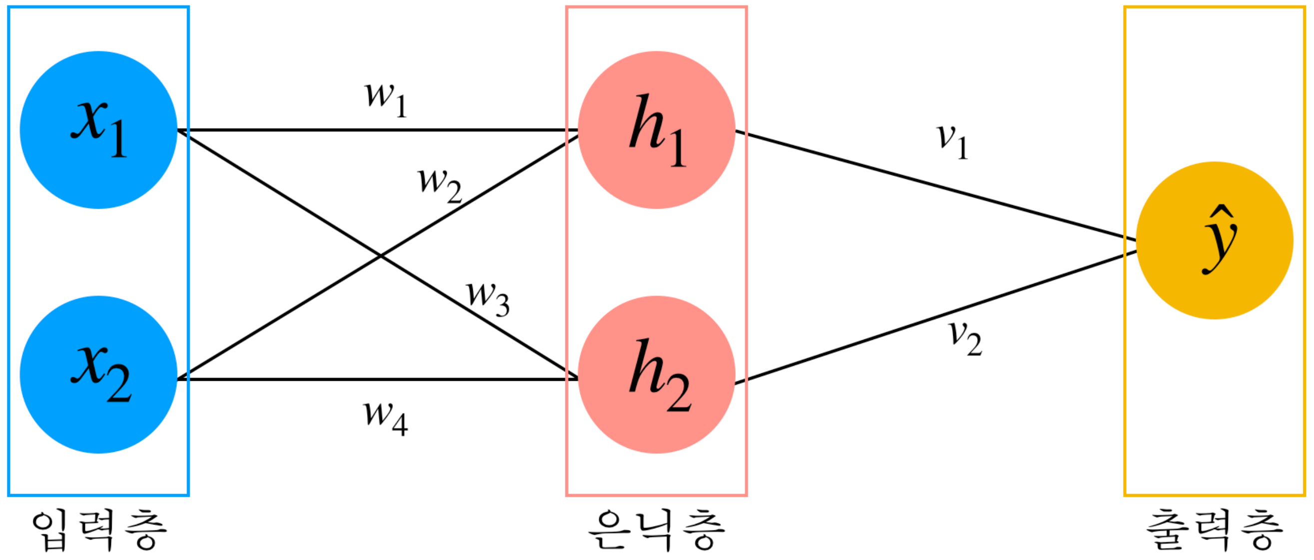
$$h_1 = w_1x_1 + w_2x_2 + b_1,\ h_2 = w_3x_1 + w_4x_2 + b_2$$

$$\hat{y} = v_1h_1 + v_2h_2 + b_3$$

$$\hat{y} = v_1h_1 + v_2h_2 + c = v_1(w_1x_1 + w_2x_2 + b_1) + v_2(w_3x_1 + w_4x_2 + b_2) + b_3$$

활성화 함수(Activation Function)

인공 신경망의 비선형 연산



$$\mathbf{a}(x) = \frac{1}{1+e^{-x}}$$

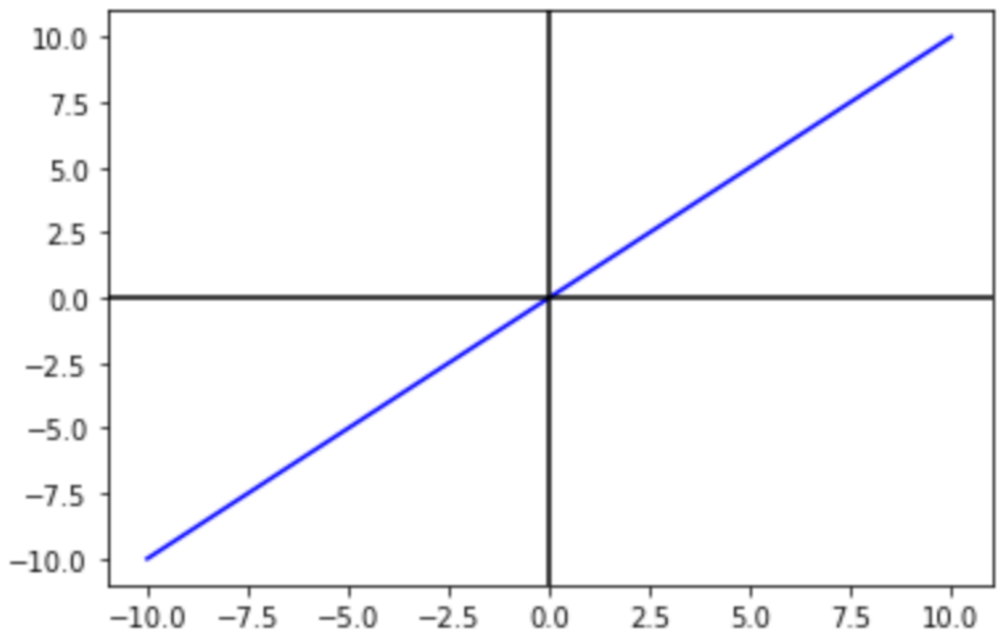
$$h_1 = w_1x_1 + w_2x_2 + b_1 \xrightarrow[\text{활성화 함수}]{} h_1 = \mathbf{a}(w_1x_1 + w_2x_2 + b_1) = \frac{1}{1+e^{-(w_1x_1+w_2x_2+b_1)}}$$

활성화 함수(Activation Function)

알아두기 2.2.1 — **선형(linear)함수.** 그대로 값을 보내주는 항등함수

$$\mathbf{a}(x) = x$$

선형함수는 $\mathbf{a}(x) = ax + b$ 형태인데 $\mathbf{a}(x) = x$인 경우는 이전 노드에서 계산되어 넘어 온 값을 그대로 받는다는 의미다.

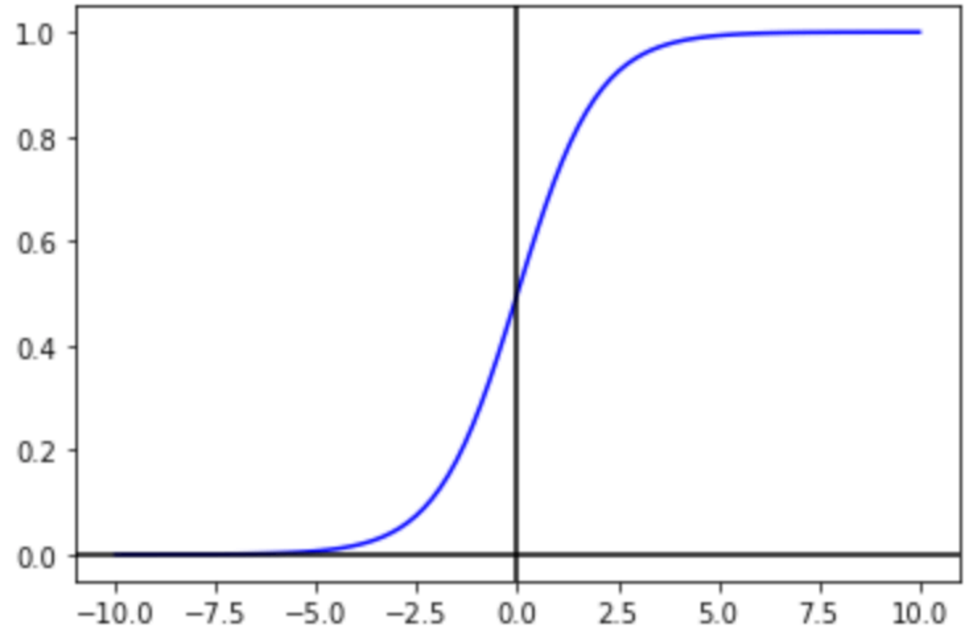


활성화 함수(Activation Function)

알아두기 2.2.2 — 시그모이드(**sigmoid**) 함수. 희대의 발견, 다양한 분야에서 쓰이는 함수

$$\sigma(x) = \frac{1}{1+e^{-x}} \qquad \text{(e는 자연상수)}$$

시그모이드는 모든 입력값에 대해 0과 1사이로 변환하는 역할을 하며 일반적으로 σ(sigma)로 표기한다. 또한 0.5를 기준으로 0.5이하면 0, 초과면 1로 변환하여 두 가지 클래스를 분류하는 이진분류(binary classification) 문제에도 활용할 수 있다.

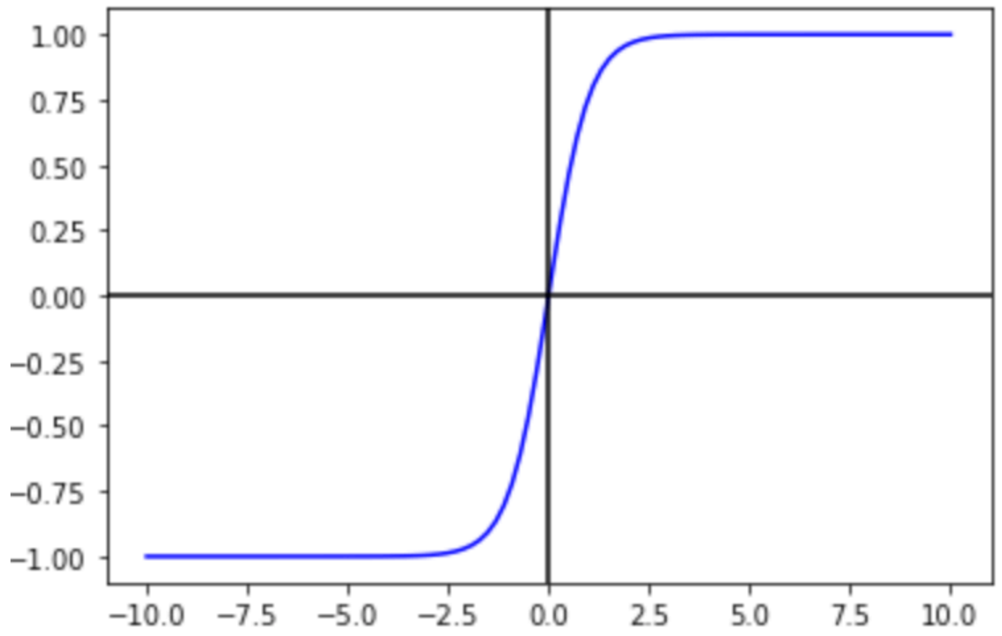


활성화 함수(Activation Function)

알아두기 2.2.3 — **tanh**함수. 시그모이드 함수와 비슷하다.

$$\mathbf{a}(x) = tanh(x)$$

시그모이드 함수와 형태는 유사하지만 -1과 1사이의 값을 취할 수 있어 0과 음수값을 갖을 수 있다. 또한 0 부근에서 $\sigma(x)$보다 더 가파른 기울기를 갖는다는 것이 차이점이다.

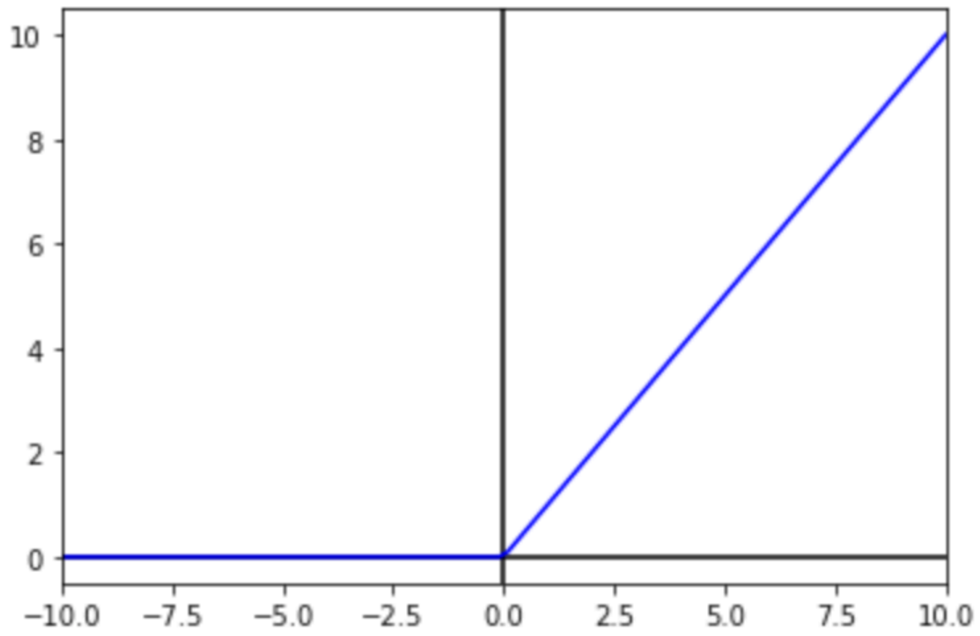


활성화 함수(Activation Function)

알아두기 2.2.4 — **ReLU(Rectified Linear Unit)** 함수. 가장 많이 쓰이는 활성화 함수

$$ReLU(x) = max(0,x)$$

두 개의 직선을 이어 만든 것으로 비선형 함수지만 선형과 매우 유사한 성질을 가지고 있다. 따라서 계산이 쉽고 미분도 쉽게 풀 수 있다. 또한 가장 중요한 점은 모델 최적화의 유용한 선형적 성질들을 보존하고 있어서 최적화를 쉽게 할 수 있게 한다. 따라서 가장 널리 사용되며 성능이 매우 좋은 것으로 알려져 있다.

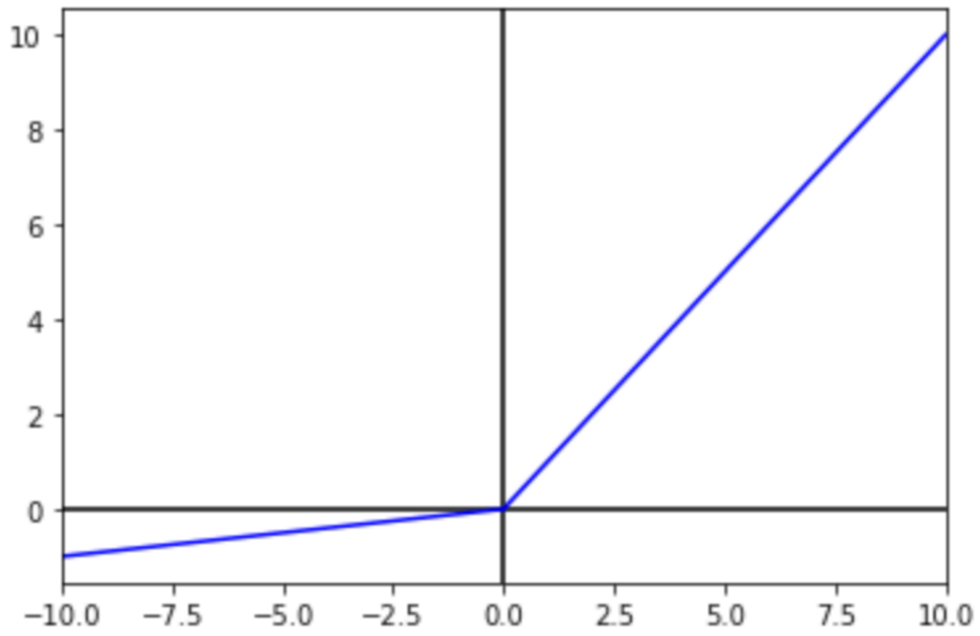


활성화 함수(Activation Function)

알아두기 2.2.5 — **Leaky ReLU 함수.** 마이너스 값도 취할 수 있는 ReLU

$$\mathbf{a}(x) = \begin{cases} x & (x > 0) \\ \alpha x & (x \leq 0) \end{cases}, \ \alpha\text{는 상수}$$

ReLU의 변형된 식이다.

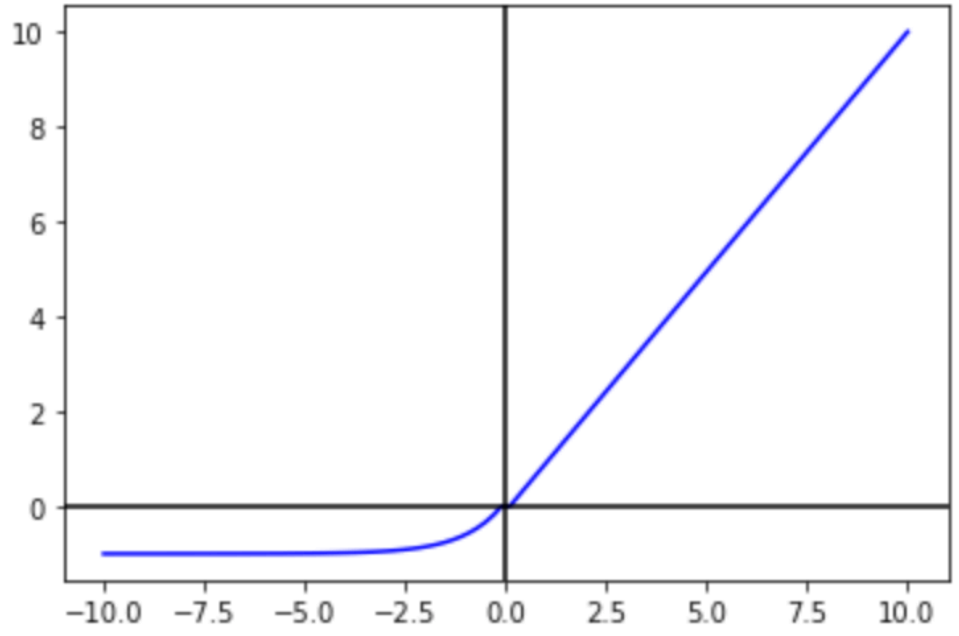


활성화 함수(Activation Function)

알아두기 **2.2.6 — ELU(Exponential Linear Unit)** 함수. Leaky ReLU보다 부드러운 함수

$$\mathbf{a}(x) = \begin{cases} x & (x > 0) \\ \alpha(e^x - 1) & (x \le 0) \end{cases}, \ \alpha\text{는 상수}$$

지수함수을 이용하여 $x \le 0$ 부분을 부드러운(smooth) 곡선으로 만든 함수다.



활성화 함수(Activation Function)

알아두기 2.2.7 — 소프트맥스 함수(**softmax function**). 분류 문제에 최적화 된 함수

벡터 $\mathbf{x} = (x_1, x_2, \ldots, x_n)$에 대하여

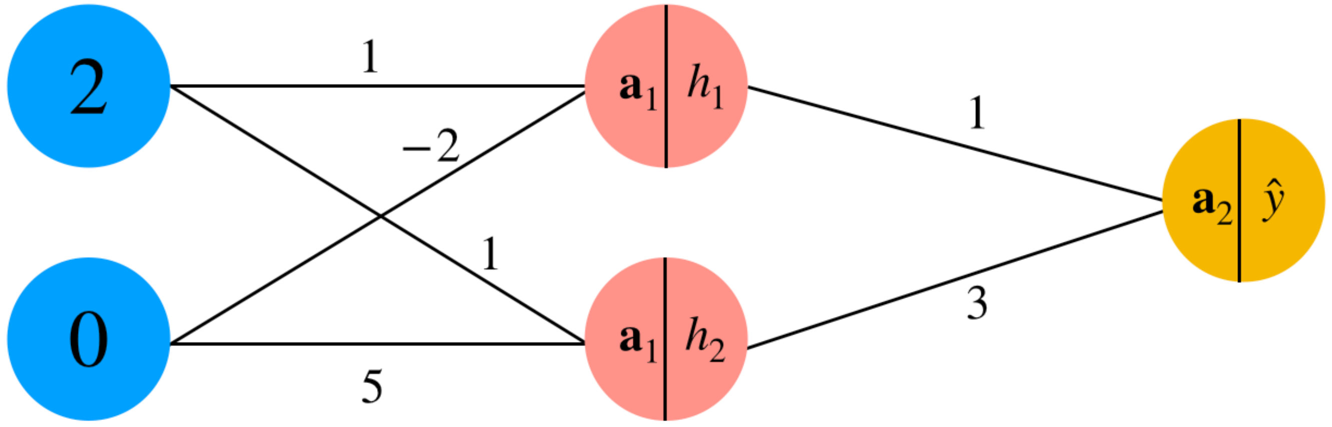
$$softmax(\mathbf{x}) = \frac{e^{\mathbf{x}}}{\sum_{k=1}^{n} e^{x_k}} \quad \text{(e는 자연상수)}$$

$e^{\mathbf{x}}$의 성분을 각각 $\sum_{k=1}^{n} e^{x_k}$으로 나눠 소프트맥스의 모든 성분의 합은 항상 1이다. 따라서 각 성분을 확률처럼 사용할 수 있는 중요한 특징을 가지고 있다.

예를 들어 $\mathbf{x} = (0, 1, 2)$이면 $softmax(\mathbf{x}) = \frac{1}{1+e+e^2}(1, e, e^2) = (0.09, 0.245, 0.665)$

- Swish, SELU, maxout, step function 등 다양한 활성화 함수 존재

활성화 함수(Activation Function)



(편향 $\mathbf{b} = \begin{pmatrix} -3 \\ 0 \end{pmatrix}$, $\mathbf{c} = 0$ 라고 하자.)

$$\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = ReLU\left(\begin{pmatrix} 1 & -2 \\ 1 & 5 \end{pmatrix}\begin{pmatrix} 2 \\ 0 \end{pmatrix} + \begin{pmatrix} -3 \\ 0 \end{pmatrix}\right) = ReLU\begin{pmatrix} -1 \\ 2 \end{pmatrix} = \begin{pmatrix} max(0,-1) \\ max(0,2) \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

$$\hat{y} = \sigma\left(\begin{pmatrix} 1 & 3 \end{pmatrix}\begin{pmatrix} 0 \\ 2 \end{pmatrix} + 0\right) = \sigma(6) = \frac{1}{1+e^{-6}} \approx 0.9975$$