

neural network with two inputs, two hidden layer, two output

~~정의~~: 합성곱: 층이 여러개로 쌓이는 것. 인간의 신경망처럼 망하고 실려 정확도 올라감.

가중치(Weight): input의 값에서 다음 층으로 넘어갈 때, 각기 다르게 곱해서 다른 값이 나온다.

즉, 데이터를 각기 다른 비율로 hidden layer로 전달하기 위해서 weights는 중요도를 다르게 한다.

편향(bias): 하나의 뉴런으로 입력된 모든 값을 모두 더하고 이 값에 더해주는 상수.

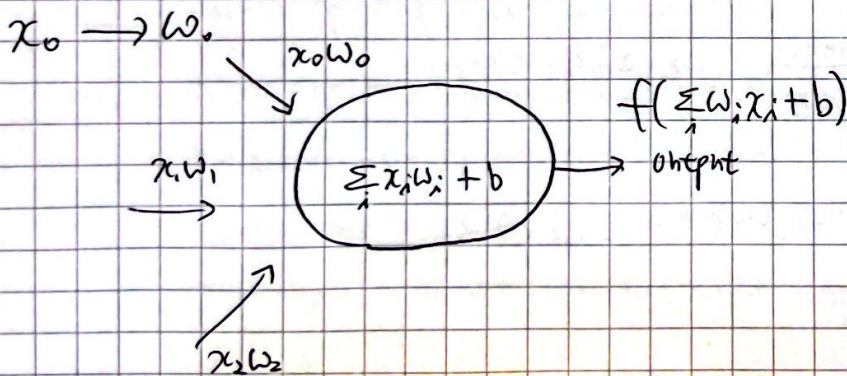
하나의 뉴런의 activation 값을 거쳐 최종적으로 출력되는 값을 출력함.

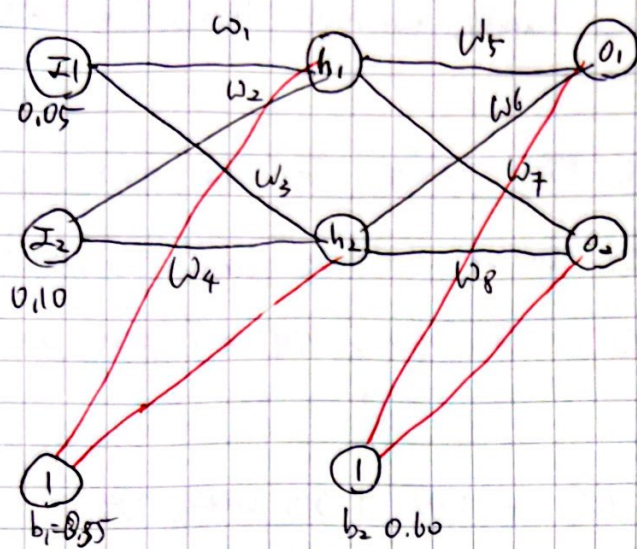
activation function: 외부 자극을 받아서 때문에 어떤 양의 정도를 출력하게 변환을 주는 함수. Sigmoid 사용.

여기서 bias가 양의 정도를 얼마나 쉽게 넘나들게 할지를 조절.

Sigmoid 함수: $S(z) = \frac{1}{1+e^{-z}}$, 0~1까지의 label의 확률을 파악할 예측.

hidden layer: 입력받은 데이터를 activation function을 거쳐 변환해서 전달.





$$\text{out}_{h1} = \frac{1}{1 + e^{-\text{net}_{h1}}}$$

~~$$\text{net}_{h1} = w_1 I_1 + w_2 I_2 + w_3 I_3$$~~

$$\text{net}_{h1} = w_1 I_1 + w_2 I_2 + b_1 x_1$$

$$\text{out}_{h1} = \frac{1}{1 + e^{-\text{net}_{h1}}}$$

Calculating the total error

$$E_{\text{total}} = \sum \frac{1}{2} (\text{target} - \text{output})^2$$

Consider w_5

$$\frac{\partial E_t}{\partial w_5} = \frac{\partial E_t}{\partial \text{out}_{o1}} \frac{\partial \text{out}_{o1}}{\partial \text{net}_{o1}} \frac{\partial \text{net}_{o1}}{\partial w_5} \rightarrow \text{오류 함수 계산}$$

w_5^+ : $\frac{\partial E_t}{\partial w_5}$ 값을 가지고 새로운 weight.

$$w_5^+ = w_5 - \eta \frac{\partial E_t}{\partial w_5} = \sim$$

↓
학습률