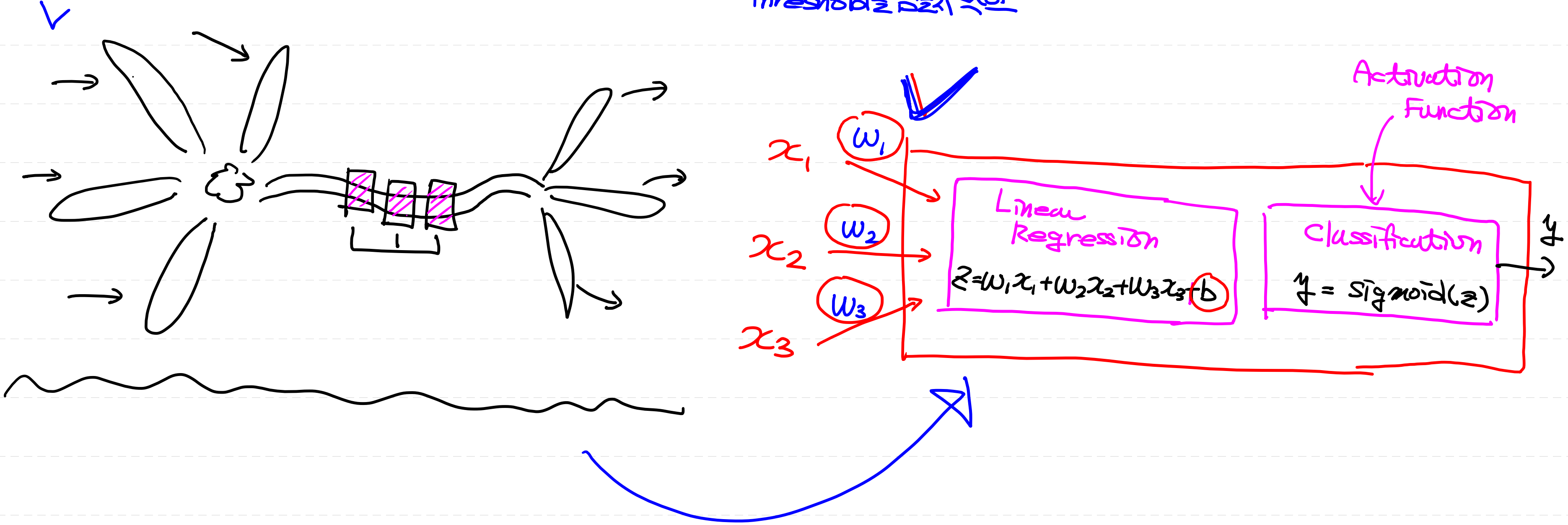
- 03/16 "Neuron"

→ 이전 neuron으로 부터 입력신호를 전달받아서 또 다른 신호를 발생시키는 일을 해요!

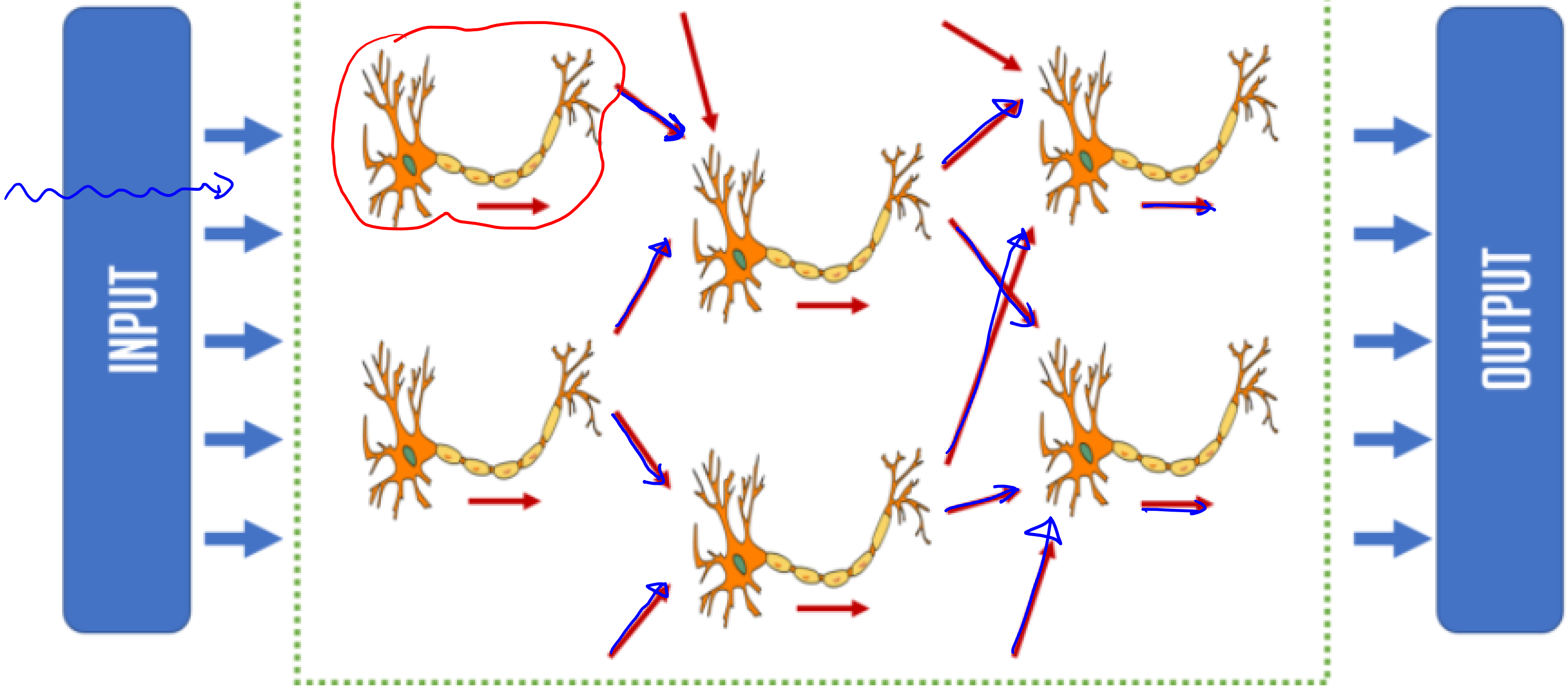
→ 입력값에 비례해서 출력값을 내보내는 형태가 X

→ 입력값에 가중치(w)를 곱한 후 모두 더해서 특정 함수를 이용해 (threshold를 넘는지 확인) 임계점을 넘으면 출력값 발생.!!

Activation Function ⇒ "softmax" "sigmoid"



NEURAL NETWORK (신경망)



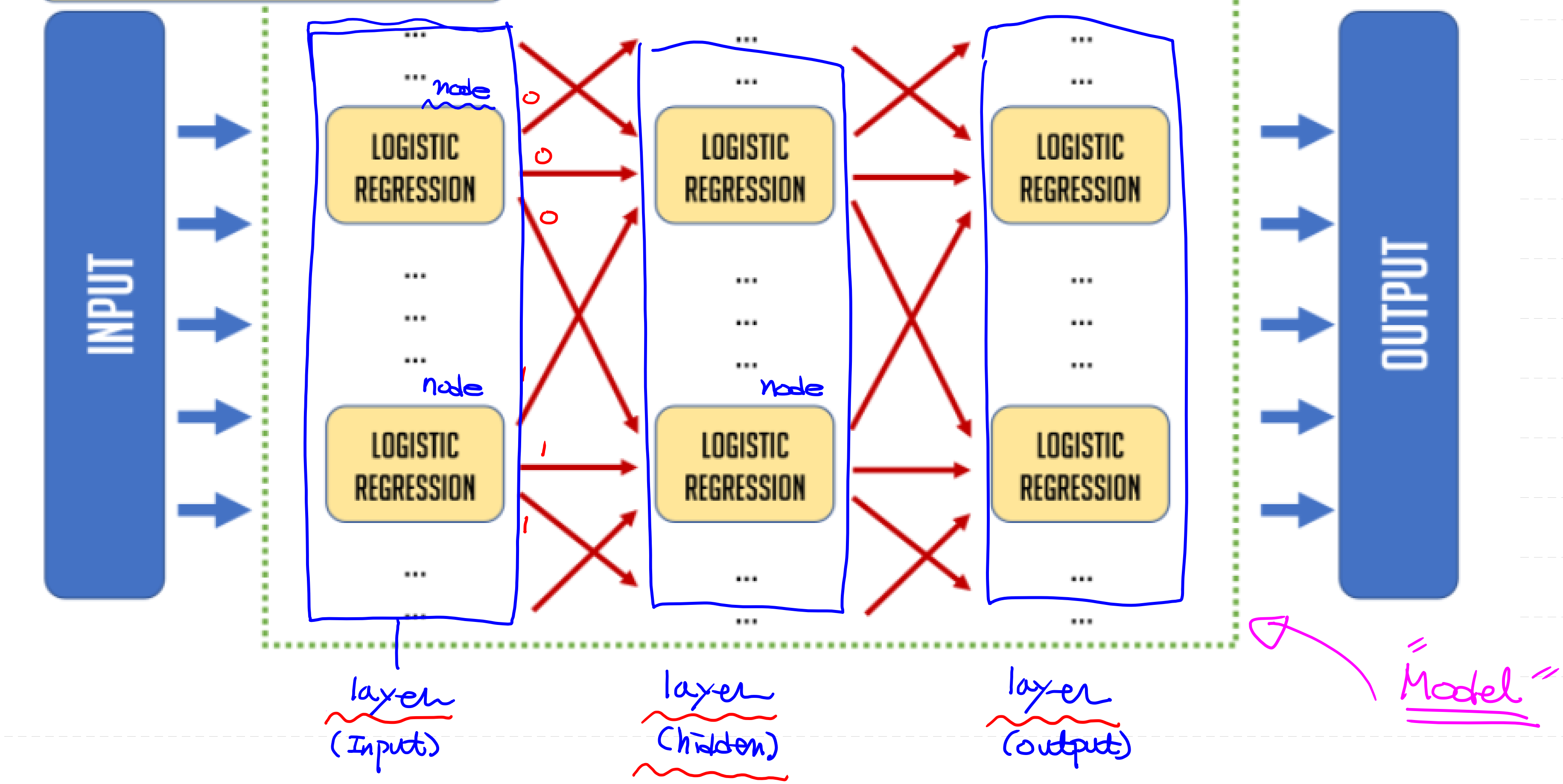
INPUT

OUTPUT

binary classification → 1 layer 안에 logistic regression 이 1개만 존재

multinomial classification → 1 layer 안에 " " 이 여러개 존재

ARTIFICIAL NEURAL NETWORK (인공신경망)



DNN

- Deep Network (Deep Neural Network)

→ Deep Learning

1개의 logistic regression을 나타내는 "node"가 서로 연결되어 있는 신경망 구조를 바탕으로

(Input Layer(입력층) + 1개 이상의 은닉층 (hidden Layer) + Output Layer(출력층)을 구축)

↳ 출력층의 오차를 기반으로 각 node의 가중치를 학습시키는 machine Learning의 한 분야 !!

(1개 이상의 hidden Layer를 이용하면 model의 정확도(accuracy)가 높아져요 ☺)

(hidden Layer를 여러개 사용할 수록 정확도 ↑)

장점 : 정확도가 가장 ↑

단점 : 학습에 들어가는 시간 ↑

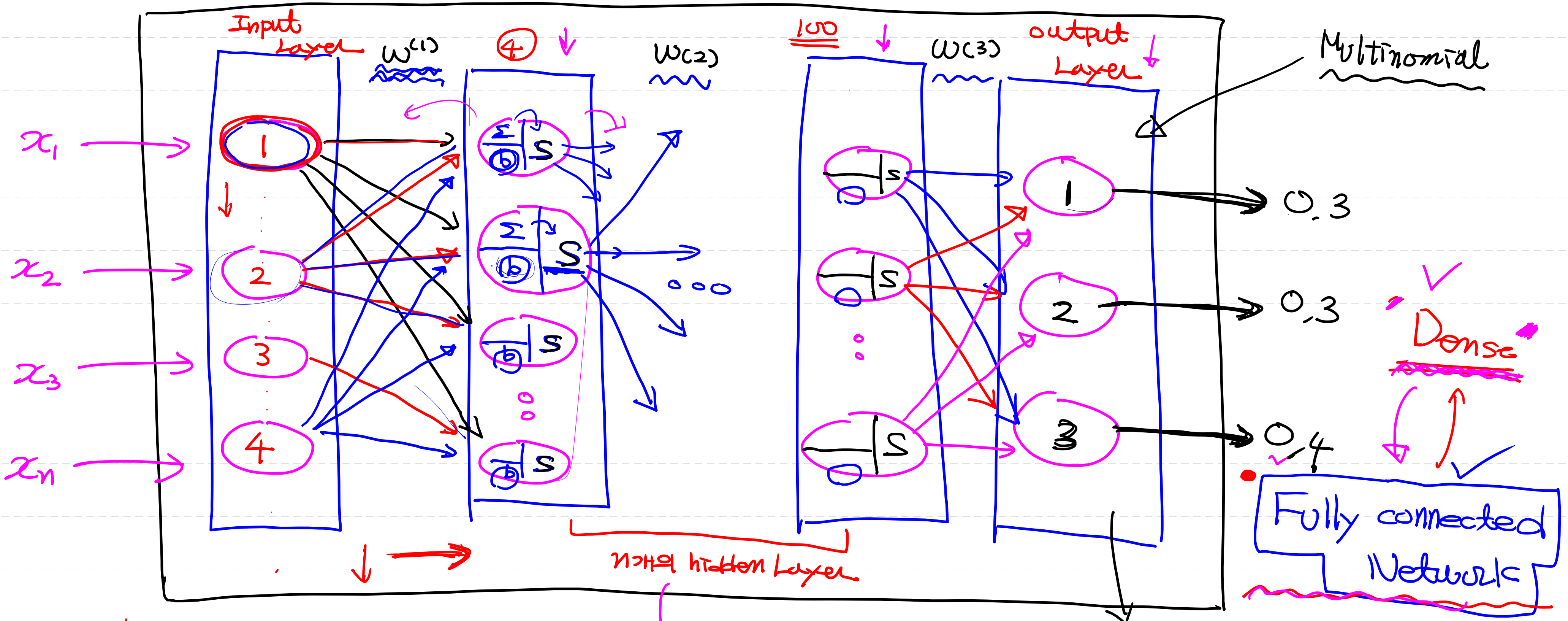
[CPU GPU]

↳ ① 그래픽카드 GPU가 달려있는 경우 (NVIDIA) (x)

② Colab

③ AWS (비용) google Cloud

"Deep Learning"



Softmax

[이전 Layer의 출력의 개수
  이후 Layer의 입력의 개수] "동일"

n의크기에따라 (n=1~3)
정확도와 속도가결정

"data의 흐름" = "Propagation"
(feed forward)

- Code 처리
  (DNN)

Single-Layer perceptron 으로 구현이 X

↳ XOR 진리표 학습 X

↳ Deep Learning 으로 구현 ~~> "학습이 되는지를 확인"

↓

Input Layer 1개 → node의 개수 ⇒ 2개
hidden Layer 2개 → node의 개수 ⇒ 10개, 6개
output Layer 1개 → node의 개수 ⇒ 1개

Tensorflow 1.15
Tensorflow 2.x ~~>