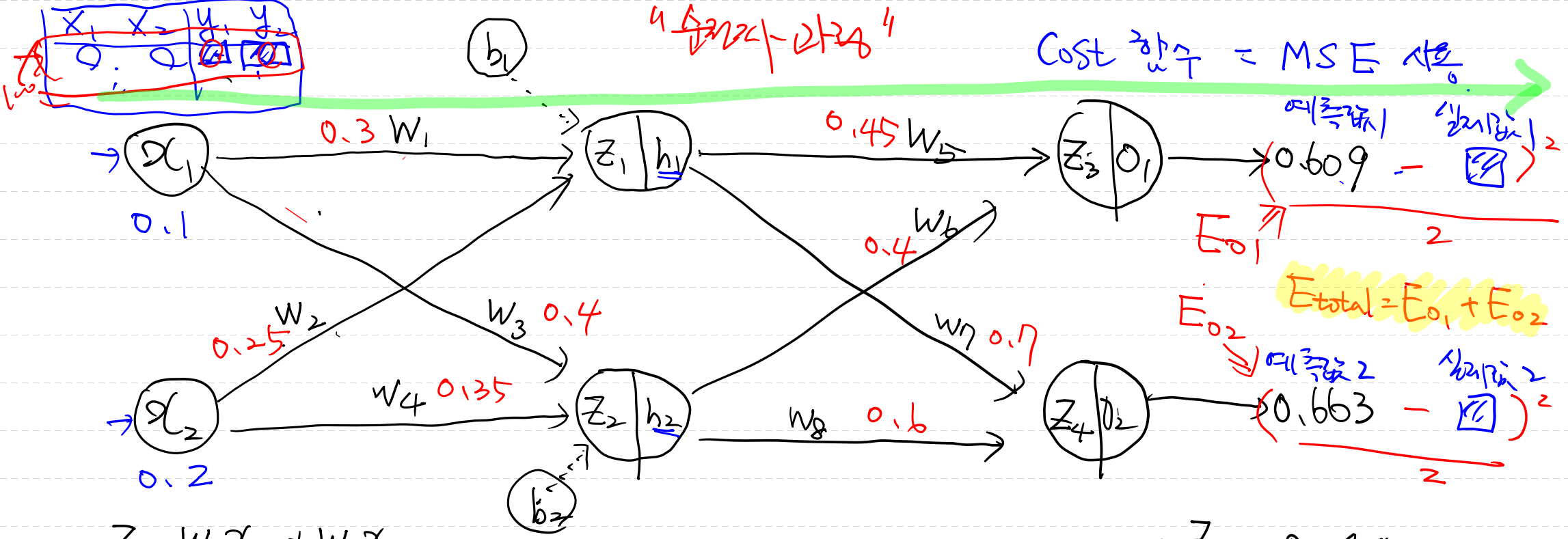


$$\begin{cases} Z_1 = W_1X_1 + W_2X_2 = 0.3*0.1 + 0.25*0.2 = 0.08 \\ Z_2 = W_3X_1 + W_4X_2 = 0.4*0.1 + 0.35*0.2 = 0.11 \end{cases}$$

$$\begin{cases} Z_3 = 0.444 \\ Z_4 = 0.680 \end{cases}$$

$$\begin{cases} h_1 = \text{sigmoid}(Z_1) = \dfrac{1}{1+e^{-0.08}} = 0.519 \\ h_2 = \text{sigmoid}(Z_2) = \dfrac{1}{1+e^{0.11}} = 0.527 \end{cases}$$

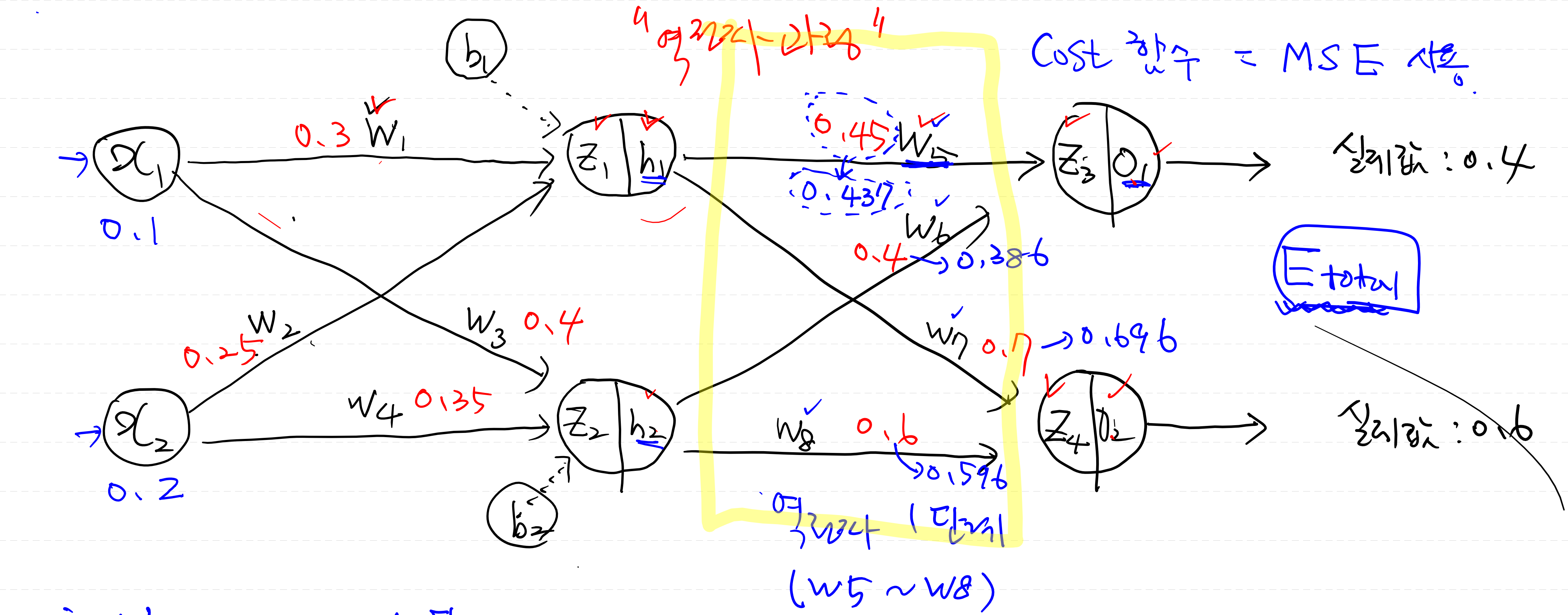
$$\begin{cases} O_1 = \text{sigmoid}(Z_3) = 0.609 \\ O_2 = \text{sigmoid}(Z_4) = 0.663 \end{cases}$$

$$E_{O1} = \frac{1}{2}(\text{예측값1} - \text{실제값1})^2 = 0.021$$

$$E_{O2} = \frac{1}{2}(\text{예측값2} - \text{실제값})^2 = 0.002$$

$$\therefore E_{total} = \boxed{0.023}$$

Cost 함수 = MSE 사용



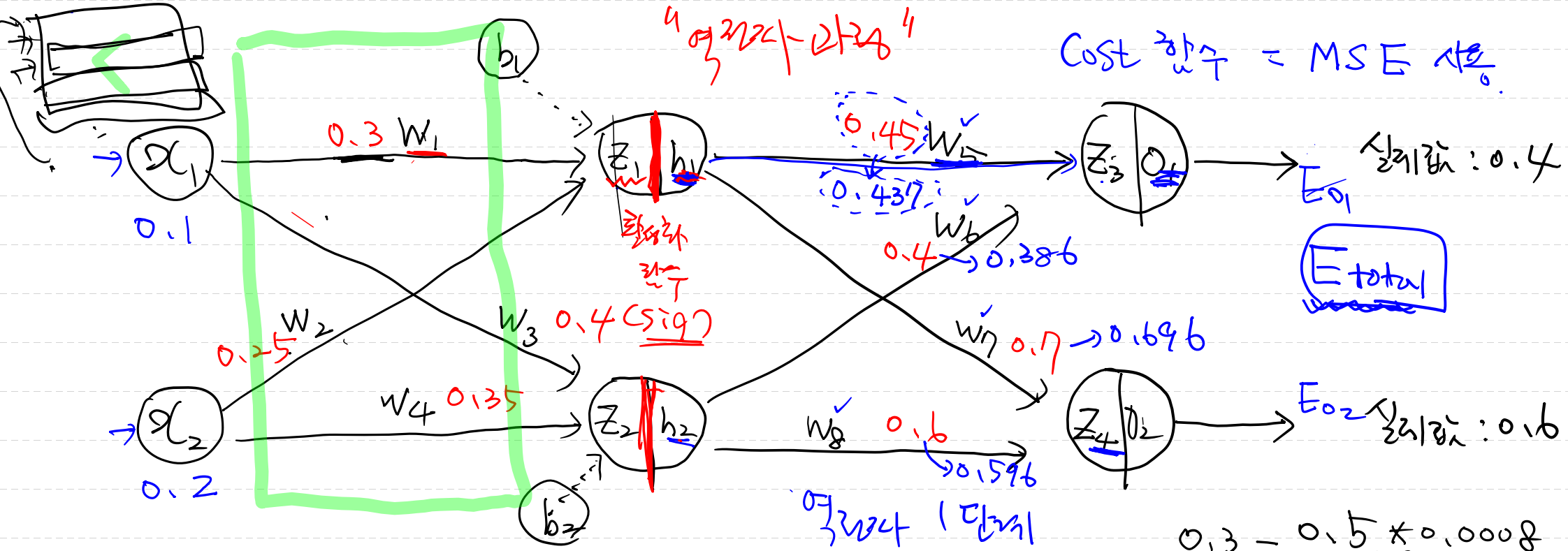
※ W_5 업데이트 (연쇄법칙, 체인룰)

$$\Rightarrow \frac{\partial E_{total}}{\partial W_5} = \frac{\partial E_{total}}{\partial O_1} * \frac{\partial O_1}{\partial Z_3} * \frac{\partial Z_3}{\partial W_5} = 0.0259$$

$$0.209 * 0.238 * 0.519$$

$$W_5^{+} = W_5 - \alpha \cdot \frac{\partial E_{total}}{\partial W_5}$$

$$0.437 = 0.45 - 0.5 \cdot 0.0259$$



※ W_1 업데이트 (역전파법, 체인룰)

$$\frac{\partial E_{total}}{\partial W_1} = \frac{\partial E_{total}}{\partial h_1} * \frac{\partial h_1}{\partial z_1} * \frac{\partial z_1}{\partial w_1}$$

0.031 = 0.250 = 0.1

$$= 0.031 * 0.250 * 0.1 = 0.0008$$

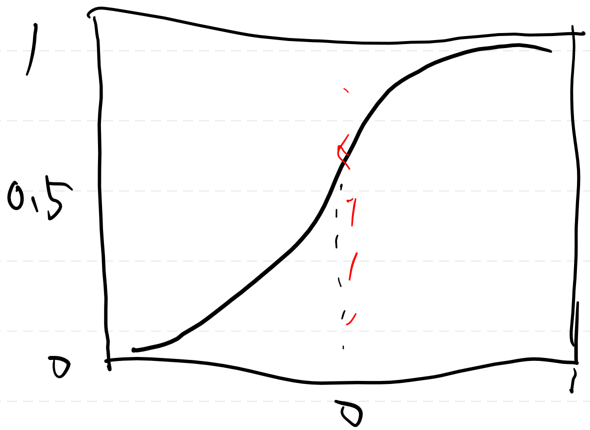
$$W_1^+ = W_1 - \alpha \cdot \frac{\partial E_{total}}{\partial W_1}$$

$0.3 - 0.5 * 0.0008$

$$\frac{\partial E_{total}}{\partial h_1} = \frac{\partial E_{O1}}{\partial h_1} + \frac{\partial E_{O2}}{\partial h_1}$$

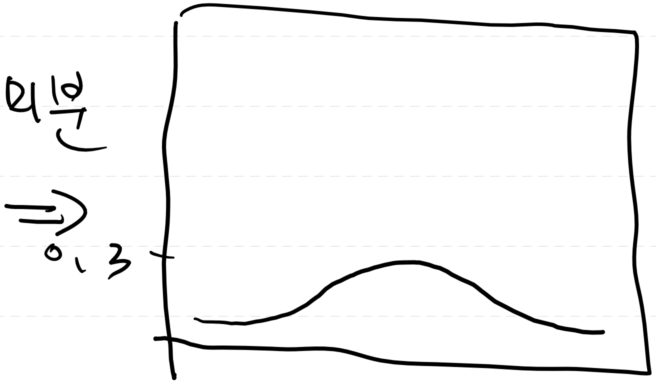
0.031 = 0.022 + 0.009

$$\Rightarrow \begin{cases} \frac{\partial E_{O1}}{\partial h_1} = \frac{\partial E_{O1}}{\partial z_3} * \frac{\partial z_3}{\partial h_1} = 0.022 \\ \frac{\partial E_{O2}}{\partial h_1} = \frac{\partial E_{O2}}{\partial z_4} * \frac{\partial z_4}{\partial h_1} = 0.009 \end{cases}$$



시그모이드

미분 ⇒



미분 후 시그모이드

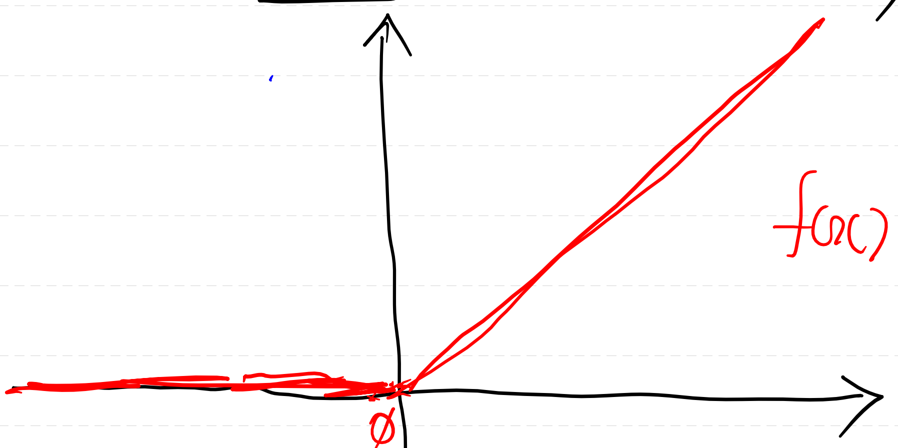
미분 ⇒ .....

기울기가 사라짐 (0 에 수렴)

⇒ vanishing gradient 문제 발생

"결론", 히든 계층에서는 sigmoid를 안 씀.

* relu (변형 relu) 사용

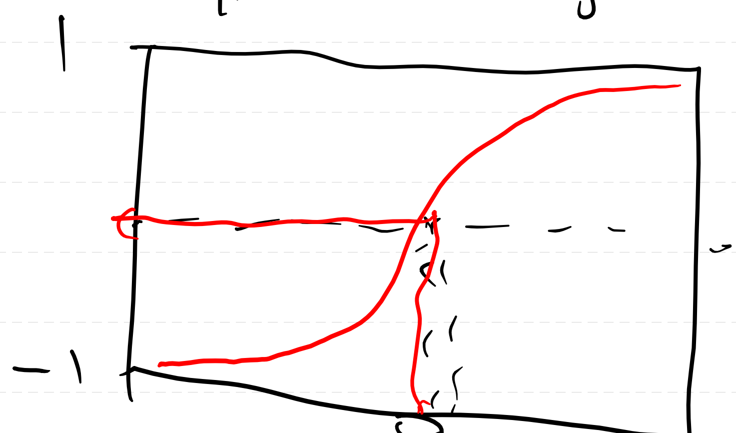


relu

$$f(x) = \begin{cases} x & (x > 0) \\ 0 & (x \le 0) \end{cases}$$

* 하이퍼볼릭 탄젠트 (tanh)

hyperbolic tangent.



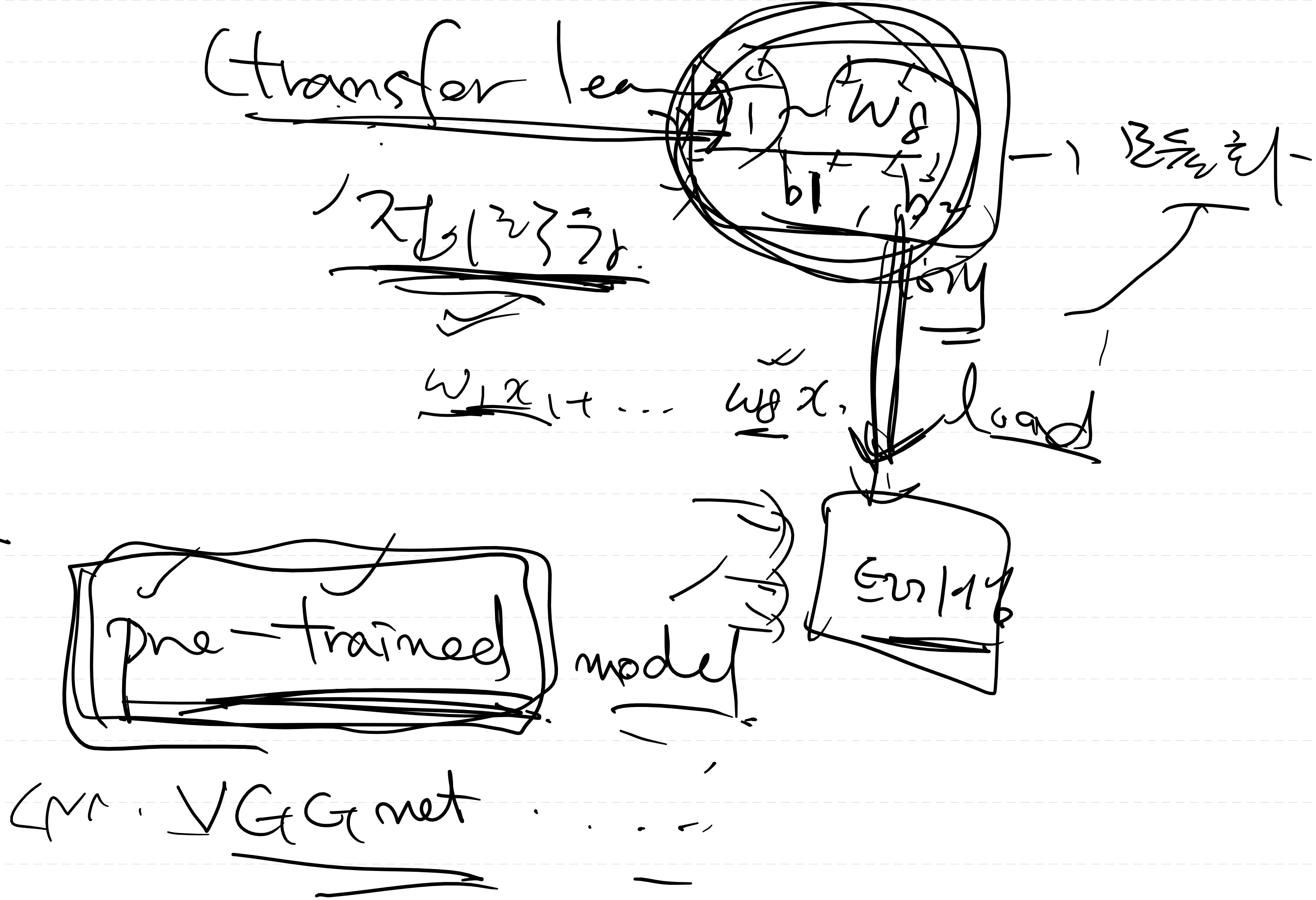
케라스 특징

① 파이썬 기반

② 확장성 우수

③ 간결한 코드

④ 모듈화 용이



Keras : 조립식 구현

Sequential model (100% 조립식 구현)

함수형 api (20% 조립식 80% 구현)

1. 데이터셋 생성 (훈련, 검증, 테스트)

2. 모델 생성 (시퀀스 모델 생성 → 레이어 추가 하여) Sequential 클래스

→ 순차적으로 레이어를 쌓아서 (연결하여) 모델 생성 Dense 클래스

3. (모델) 학습 환경 설정 (cost 함수, SGD ..... (최적화)) compile 함수

4. 모델 학습 (사용) fit 함수

→ 학습 시 훈련, 검증 데이터의 cost, acc 확인

epoch 설정 ??

5. 모델 평가 (사용) evaluate 함수

6. 모델 사용 Predict 함수