



12장

다중 분류 문제

3팀 강호연 김범주 정유찬 송지원 김수환

Contents.

1

다중 분류 문제
이론

2

실습



pre-class quiz
해설

4

추가 예제

다중 분류 문제 이론



01

다중 분류 문제

0과 1 결과값중 하나를 선택하는 문제가 아닌, 3개 이상의 결과값을 예측하는 모델을 다중 분류

원-핫 인코딩

소프트맥스

import <라이브러리>?



프로그램 제작 시 필요한 기능의 모음집

재사용이 필요한 기능을 반복적인 코드 작성을 없애기
위해 언제든지 호출 가능하도록 Class/Function으로
만들어진것



텐서플로우(TensorFlow)

딥러닝 / 머신러닝을 코드로 구현하기 위해 필요한 파이썬 기반 수학적 라이브러리 묶음

- 데이터는 tensor로 표현
- 연산은 graph에서
- graph는 Session내에서 실행

그런데...왜 하필 Python?

- 특정 분야를 위한 **스크립트 언어**의 편리성 多

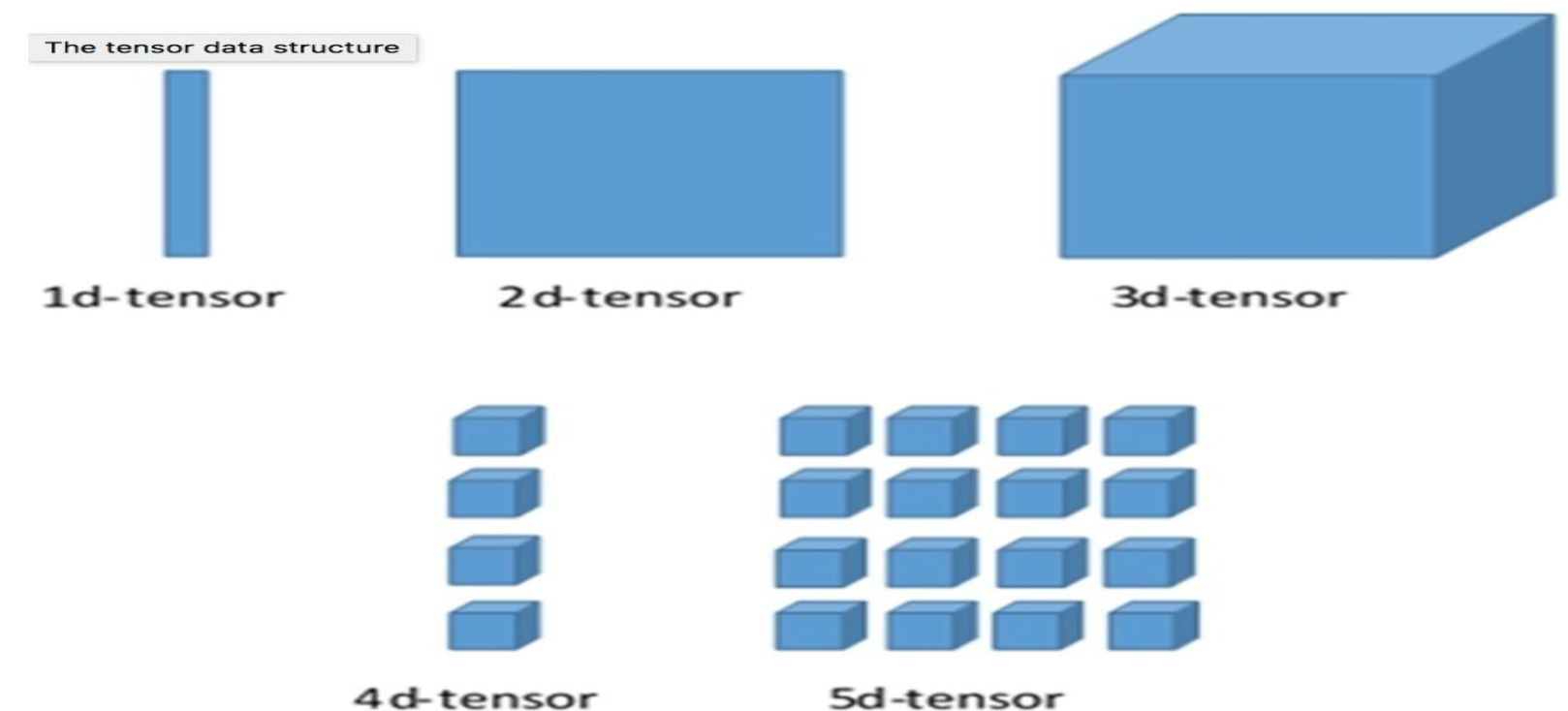
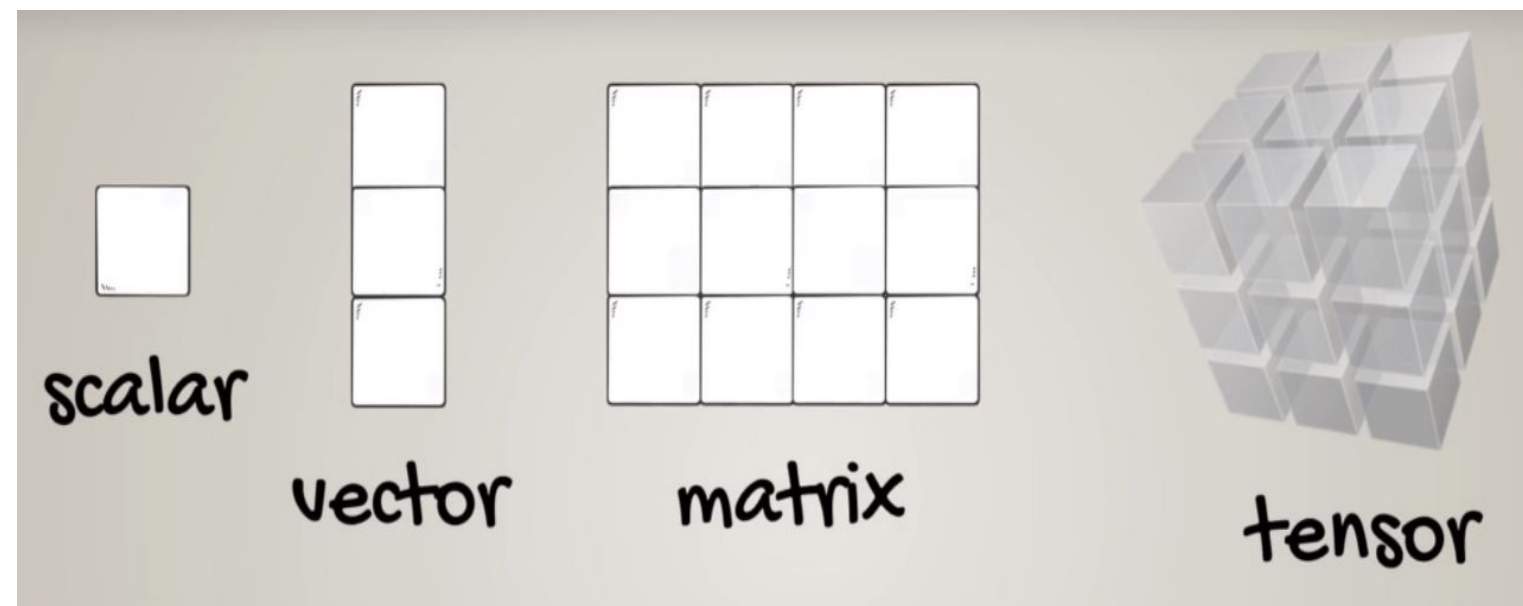
스크립트(script) 언어란?

연극 용어에서 비롯된 스크립트는 **대본**과 같은 의미로
극 안의 상황에서 배우가 어떻게 행동할 것인지 지시해놓은 대본처럼,
소프트웨어를 어떻게 실행할 지 **제어**하는 역할을 하는 언어

파이썬의 Hello world! 출력	C의 Hello world! 출력
<pre>print("Hello world!")</pre>	<pre>#include<stdio.h> int main() { print("Hello world!"); return 0; }</pre>
간단한 출력도 이렇게 차이가 나는데 복잡해지면...?	

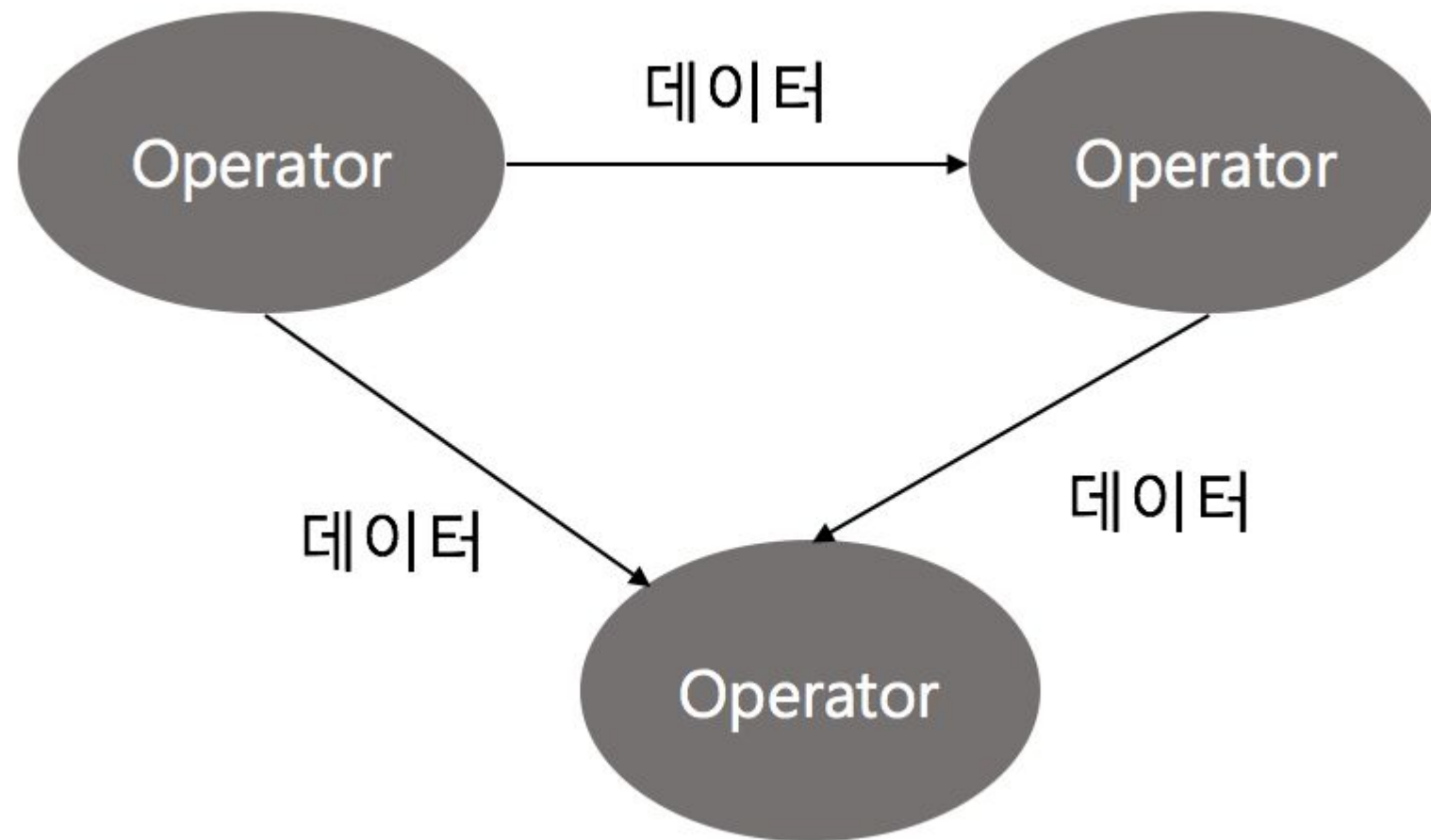
텐서(Tensor) : 정보를 담는 방들의 모임

- 다차원 배열의 일반화
- 텐서플로우에서 사용되는 모든 데이터셋(set)은 텐서
- 텐서의 차원이 커질수록 담을 수 있는 정보(data)가 증가



그래프(Graph) : 설계도

노드(node)/꼭짓점(vertex), 변(edge)으로 구성되어 있는 객체



- ★ 엣지(Edge) = 텐서(Tensor) = 데이터(data)
—> 화살표의 방향은 정보의 흐름을 표현
- ★ 노드(Node) = 연산자(Operator)
—> 간단한 사칙연산부터 고차원의 정보를 처리하는 장치를 나타냄

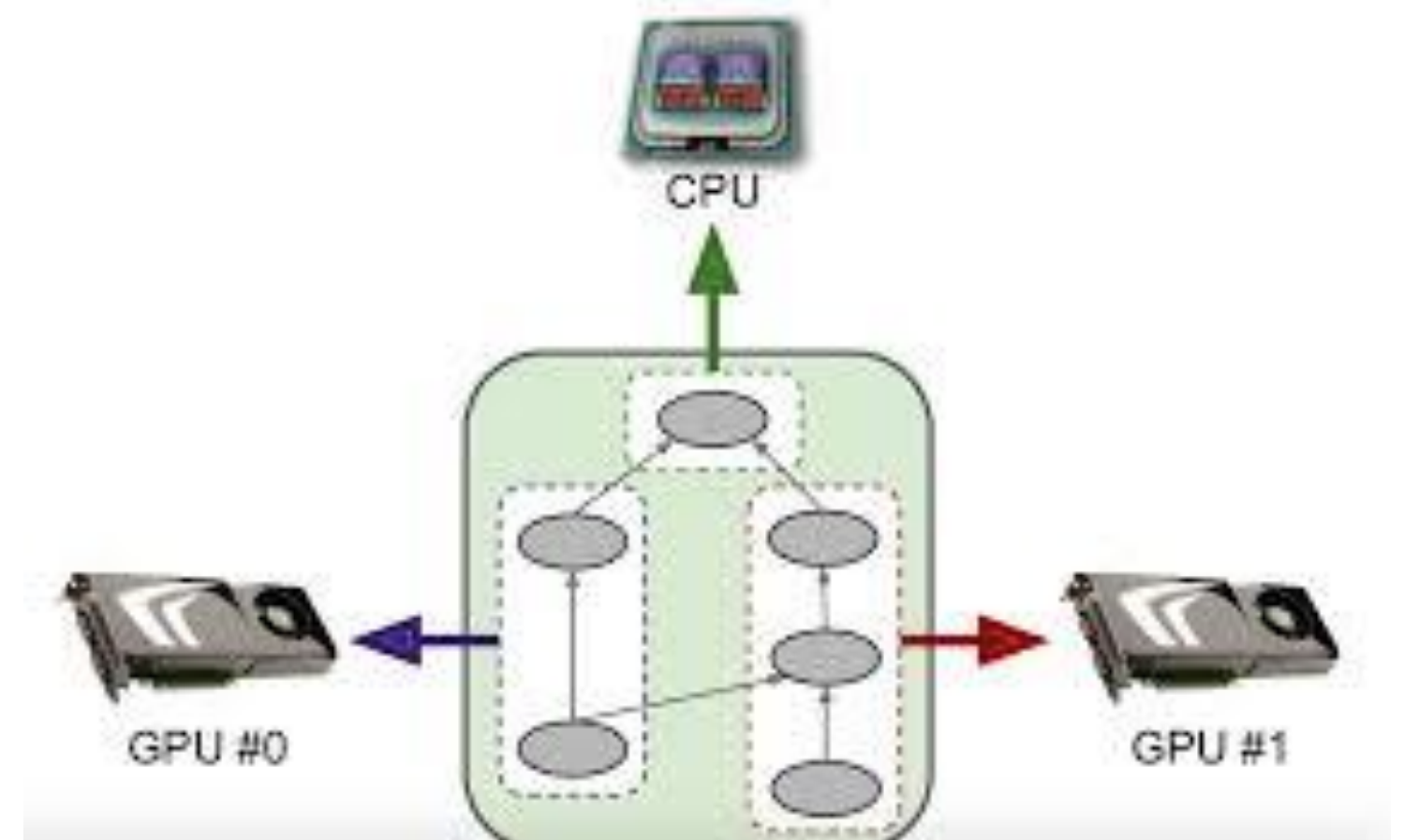
세션(Session) : 실제 구현

<텐서플로우 작동 방식>

1. 모든 것을 파이썬에서 디자인(설계)
2. 연산 담당은 **외부**
—> 구체적으로 **device**(CPUs, GPUs)위에서 연산

- ★ 모델링은 Python, 실제 실행은 C++
—> 구현은 Python이 쉽지만, C/C++ 계열이 하드웨어에서 훨씬 빠르게 연산 가능

텐서플로우 계산이 돌아갈 수 있는 환경을
하드웨어 내부에 만들어주는 코드



세션 선언문

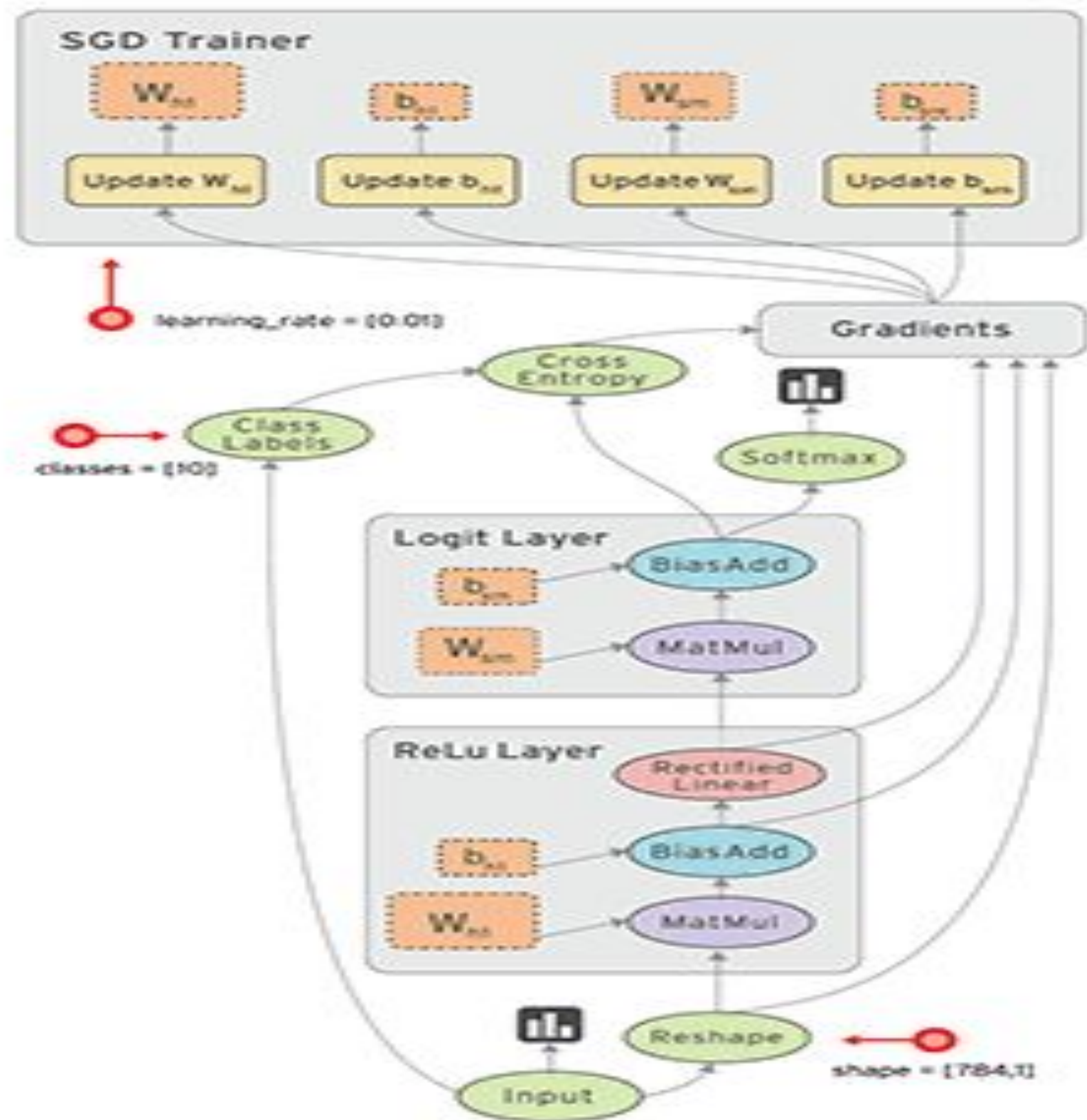
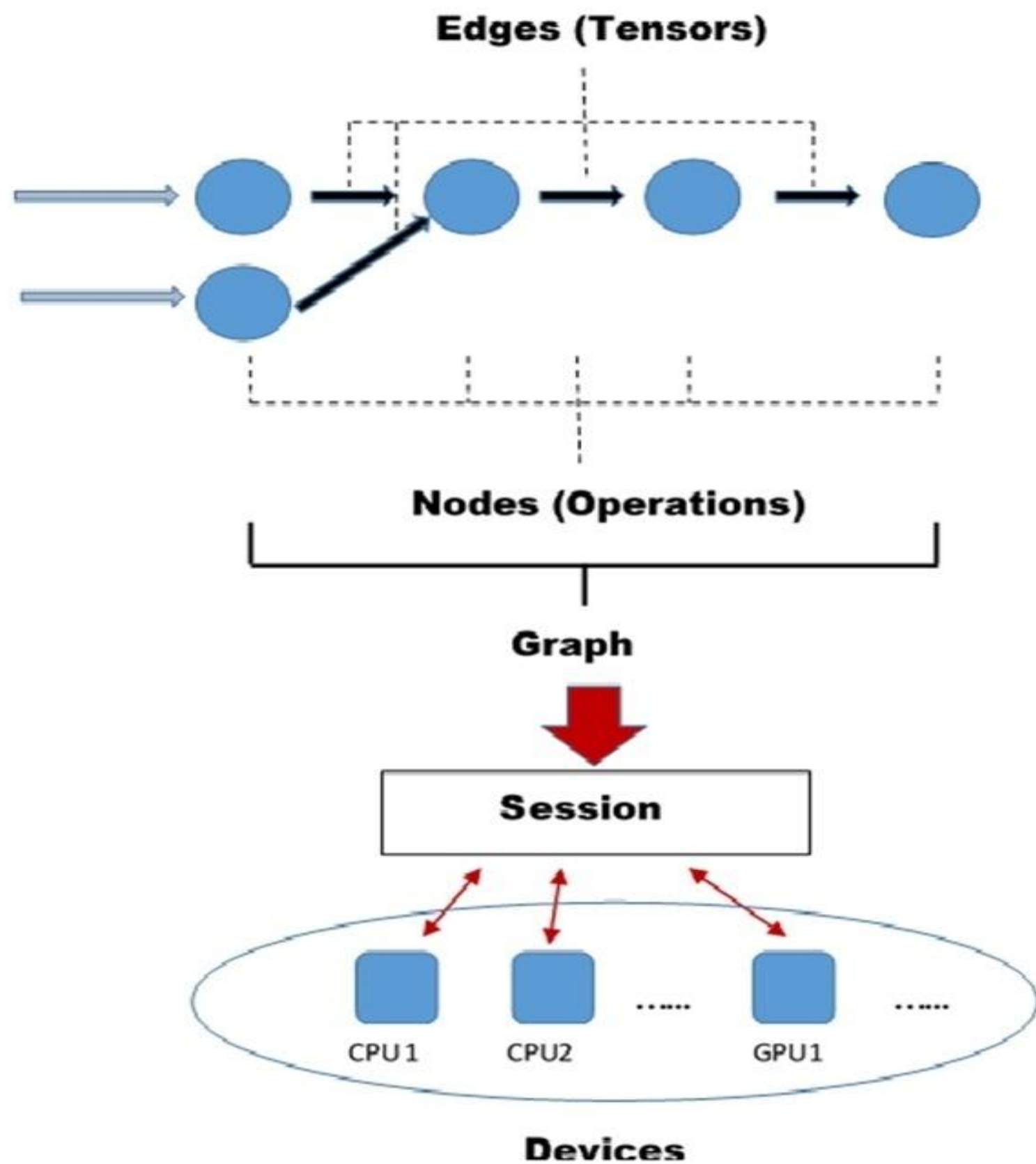
- ① `sess = tf.Session()`
- ② `sess = tf.InteractiveSession()`

세션 실행문

`sess.run(텐서 or 연산)`

세션 종료문

`sess.close()`
(단, with 절에서는 불필요)

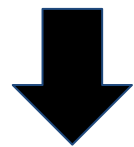


다중 분류 문제 이론

원-핫 인코딩(One-Hot Encoding)

- 컴퓨터는 문자열보다 숫자를 더 잘 처리
- 원-핫 인코딩은 문자를 숫자로 바꿔주는 기법들 중 하나
- 단어 집합의 각 단어들에 고유한 정수를 부여한 후, 표현할 단어의 정수를 인덱스로 인식해 해당 위치에 1, 그 외에는 0을 부여

['dog', 'cat', 'book', 'apple', 'cherry', 'banana']



['dog': 0, 'cat': 1, 'book': 2, 'apple': 3, 'cherry': 4, 'banana': 5]



cherry: [0, 0, 0, 0, 1, 0]

book: [0, 0, 1, 0, 0, 0]

소프트맥스(softmax)

- 다중 분류 문제를 위한 활성화 함수
- 로지스틱 함수의 다차원 일반화 함수(2개의 클래스 분류 ➡ K개의 클래스 분류)
- 소프트맥스의 입력으로 사용되는 벡터의 차원은 분류하고자 하는 클래스의 개수와 동일 해야함

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \text{ for } i = 1, 2, \dots, k$$

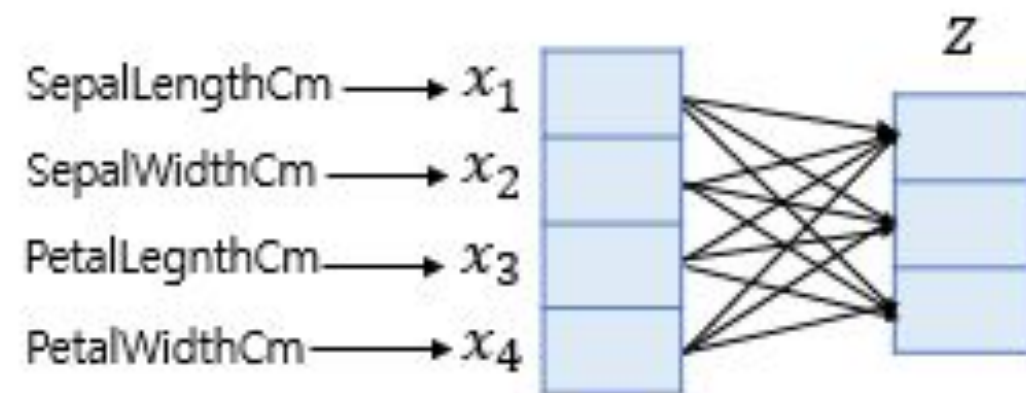
K = 3 ➡ $\text{softmax}(z) = \left[\frac{e^{z_1}}{\sum_{j=1}^3 e^{z_j}} \quad \frac{e^{z_2}}{\sum_{j=1}^3 e^{z_j}} \quad \frac{e^{z_3}}{\sum_{j=1}^3 e^{z_j}} \right] = [p_1, p_2, p_3] = \hat{y} = \text{예측값}$

$$z = [z_1 \ z_2 \ z_3]$$

소프트맥스(softmax)

- 아이리스 품종 예측의 독립변수는 4개(즉, 입력 벡터의 차원이 4차원)

➡ 소프트맥스의 입력 벡터의 차원은 클래스의 개수(3개)와 동일 ➡ 차원 축소



$$\text{softmax} \left(\begin{matrix} & & & \\ & & & \\ & & & \\ & & & \end{matrix} \times \begin{matrix} x \\ \\ \\ \end{matrix} + \begin{matrix} b \\ \\ \end{matrix} \right) = \begin{matrix} \text{예측값} \\ \\ \end{matrix}$$

$3 \times 4 \quad 4 \times 1 \quad 3 \times 1 \quad 3 \times 1$

소프트맥스(softmax)

- 오차 공식: 크로스 엔트로피 함수

$$cost = - \sum_{j=1}^k y_j \log(p_j)$$

$$-\{y_data \log h + (1 - y_data) \log(1 - h)\}$$



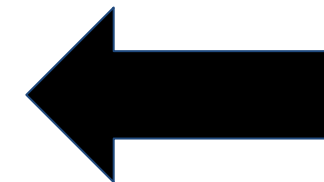
y_data를 y_1 로 치환, $1-y_data$ 를 y_2 로 치환, h 를 p_1 로 치환, $1-h$ 를 p_2 로 치환



$$-(y_1 \log(p_1) + y_2 \log(p_2))$$



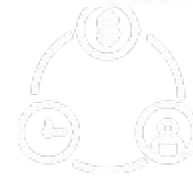
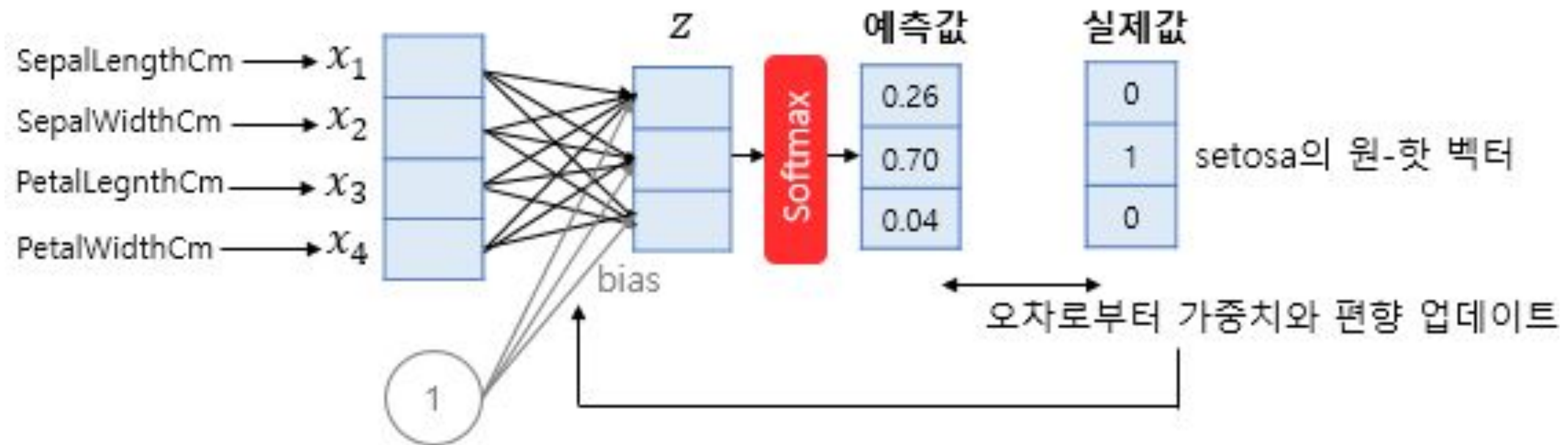
$$-(\sum_{i=1}^2 y_i \log p_i)$$



로지스틱 회귀(이진 분류)에서
사용된 오차 공식과 동일하다는 것을 알수 있다



소프트맥스(softmax)



입력 ➡ 차원 축소 ➡ 소프트맥스 함수 ➡ 예측값 반환 ➡ 오차 계산 ➡ 가중치와 편향 업데이트

: 오차가 최소가 되었을 때, 확률값이 가장 큰 클래스로 분류



실습

—

03 pre - class quiz

해설

4번 문제 해설

Q4. 다음 중 옳지 않은 것을 고르시오. *

- ☐ (1) 다중 분류는 여러 개의 답 중 하나를 고르는 분류 문제다.
- ☐ (2) 활성화 함수를 적용하기 위해서는 Y값이 0과 1로 이루어져 있어야 한다.
- ☐ (3) 소프트맥스는 총합이 1인 형태로 바꿔서 계산해 주는 함수이다.
- ☒ (4) 소프트맥스는 여러 개의 Y값을 0과 1로만 이루어진 형태로 바꾸어 준다.

-> 여러 개의 Y 값을 0과 1의 형태로 바꾸는 기법은 **one-hot 인코딩**이다!

5번 문제 해설

소프트맥스 함수가 적용된 부분을 고르시오. *

(가) 가중합

(나) 소프트맥스:

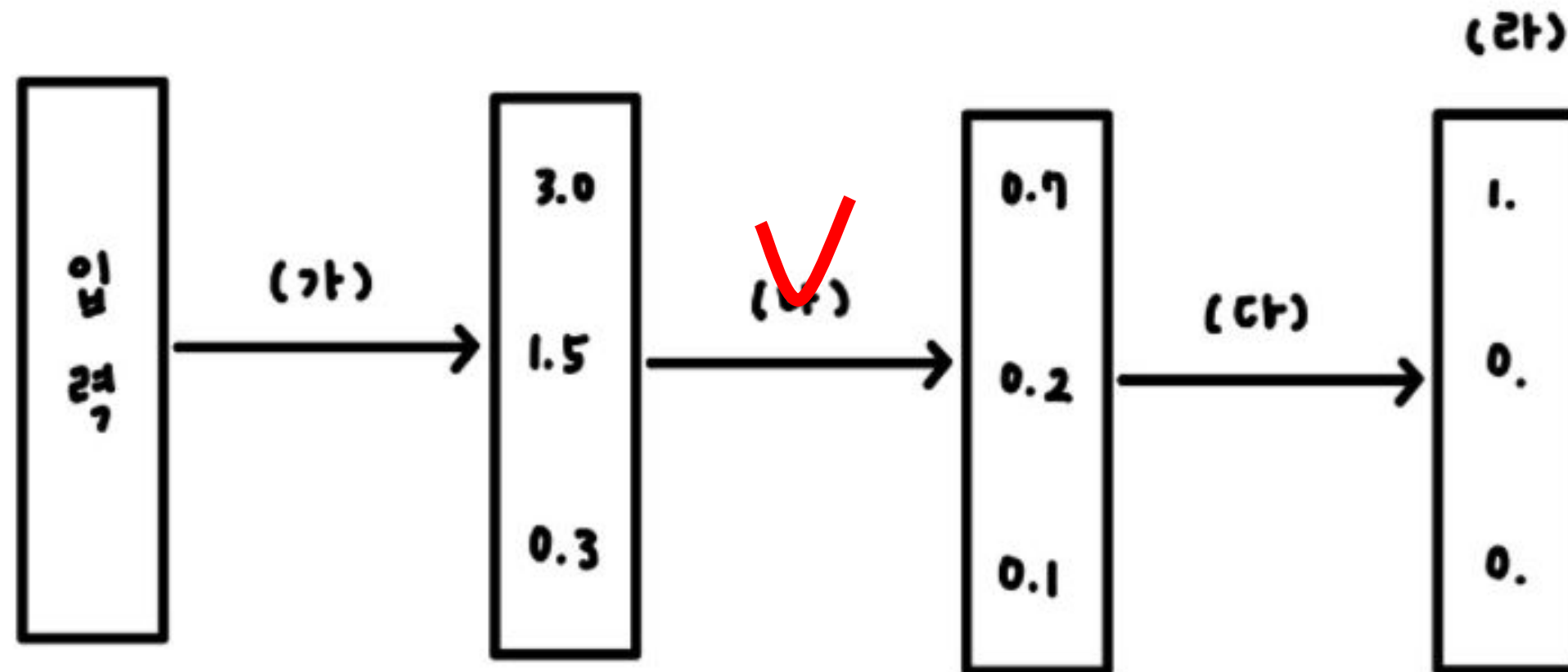
총합이 1인 형태로 바꿔서 계산해 주는 함수

(다) 교차엔트로피손실:

손실함수로 주로 분류 문제에 많이 사용됨

(라) 원-핫 레이블:

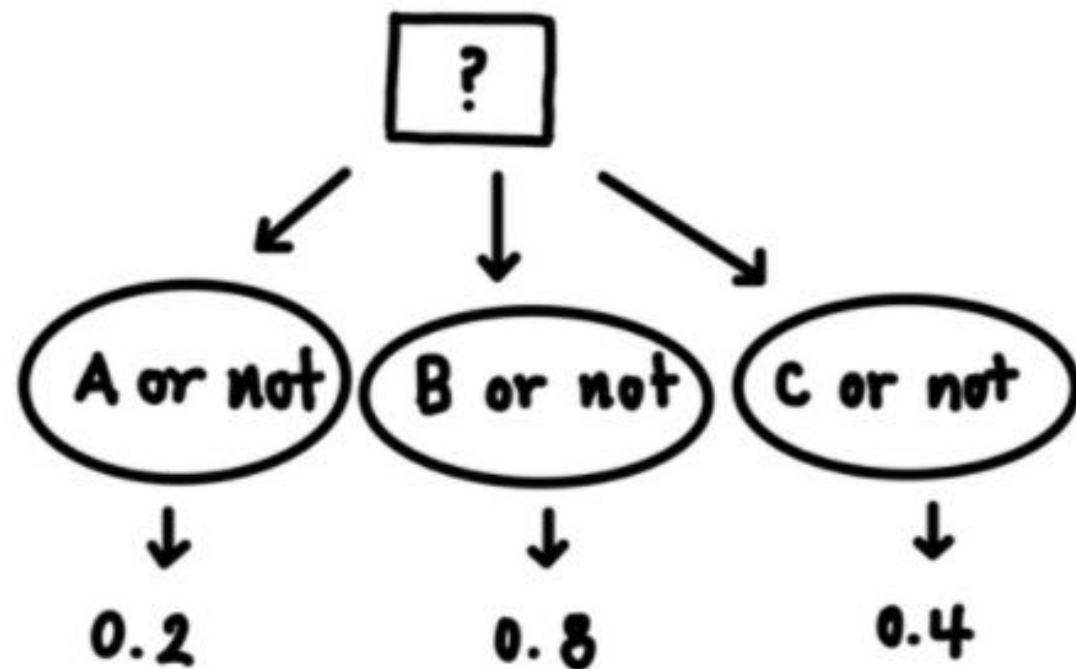
여러 개의 Y 값을 0과 1의 형태로 바꾸는 기법



6번 문제 해설

- 다중 class 예측은 출력값을 확률값으로 바꿔줘야 한다
- 정규화: 전체 data 중 자신의 data 비율로 계산

Q6. 다음 그림은 A,B,C 세가지 클래스를 로지스틱 회귀 방식으로 각각 학습 시킨 후, (?)를 입력하여 나온 클래스 예측 결과값이다. (?)의 클래스를 예측한 결과와 출력값을 확률값으로 바르게 나타낸 것을 고르시오. (계산기 사용가능)



풀이)

$$0.2 / (0.2 + 0.8 + 0.4) = 0.054 \rightarrow 5.4\%$$

$$0.8 / (0.2 + 0.8 + 0.4) = 0.756 \rightarrow 75.6\%$$

$$0.4 / (0.2 + 0.8 + 0.4) = 0.189 \rightarrow 18.9\%$$

- ☐ (1) (?)의 예측 클래스:A, 5% - 80% - 15%
- ☐ (2) (?)의 예측 클래스:B, 20% - 80% - 40%
- ☒ (3) (?)의 예측 클래스:B, 5.4% - 75.6% - 18.9%
- ☐ (4) (?)의 예측 클래스:B, 5% - 80% - 15%
- ☐ (5) (?)의 예측 클래스:C, 5.4% - 75.6% - 18.9%



(?)의 클래스는 B로 예측된다!

04

추가 예제



pytorch를 이용하여 MNIST 손글씨 데이터셋을 분류해보자 (softmax 회귀 활용)

