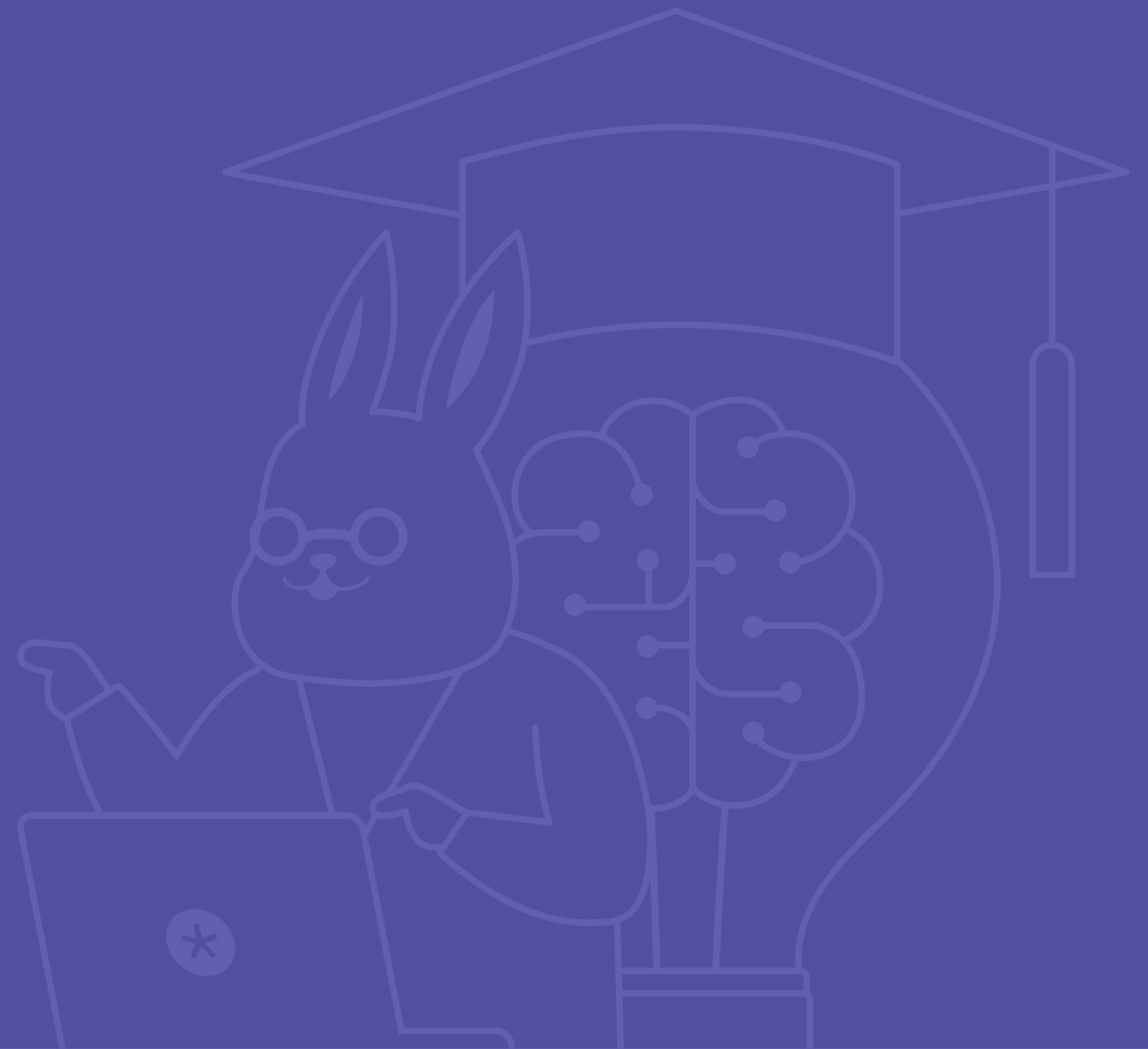




# 딥러닝 기초

## 2장 텐서플로우와 딥러닝 학습 방법

오혜연 교수님



# Contents

- 01. 딥러닝 모델의 학습 방법
- 02. 텐서플로우(TensorFlow)
- 03. 텐서플로우 기초 사용법
- 04. 텐서플로우로 딥러닝 모델 구현하기

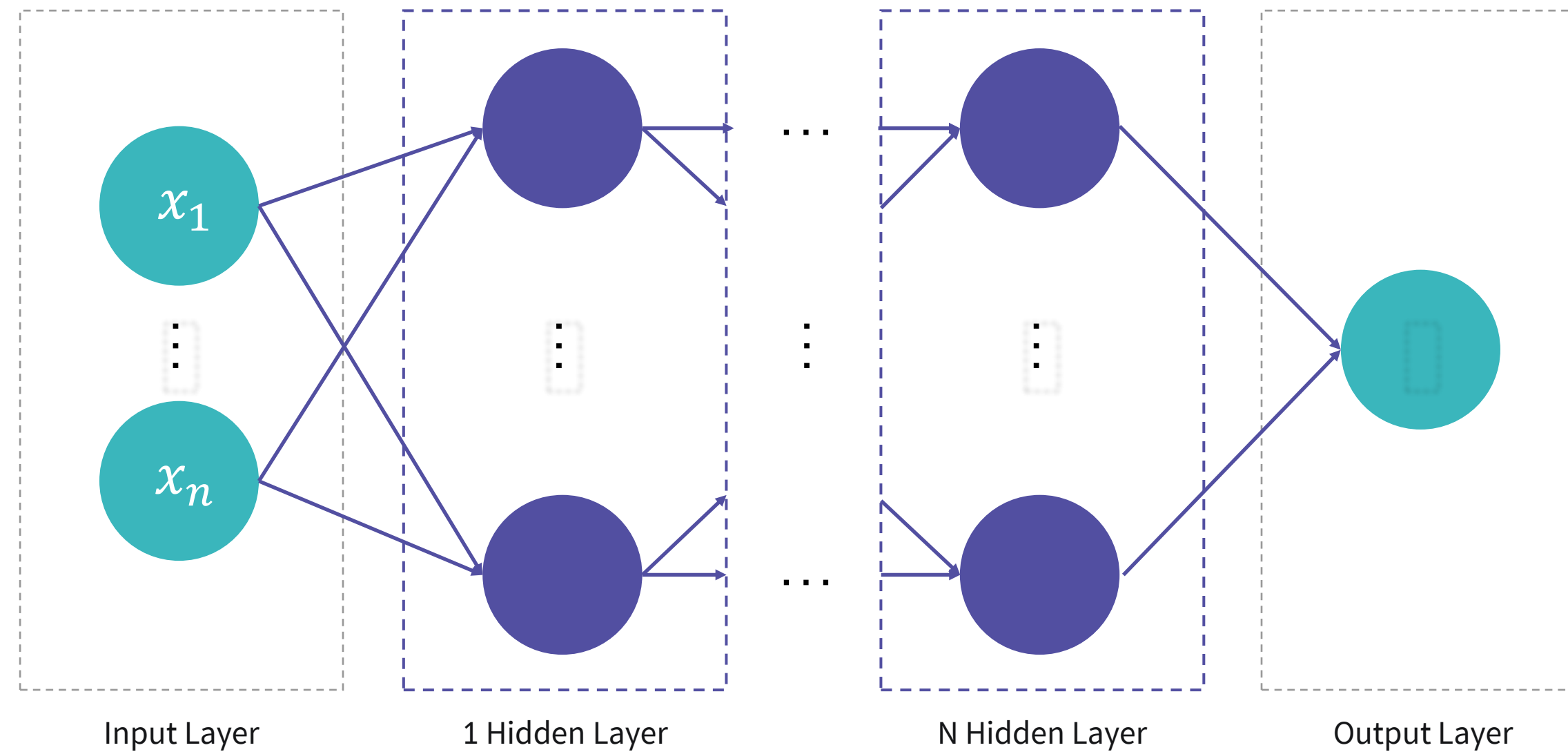
01

# 딥러닝 모델의 학습 방법



# 01 딥러닝 모델의 학습 방법

## ✓ 딥러닝 모델이란



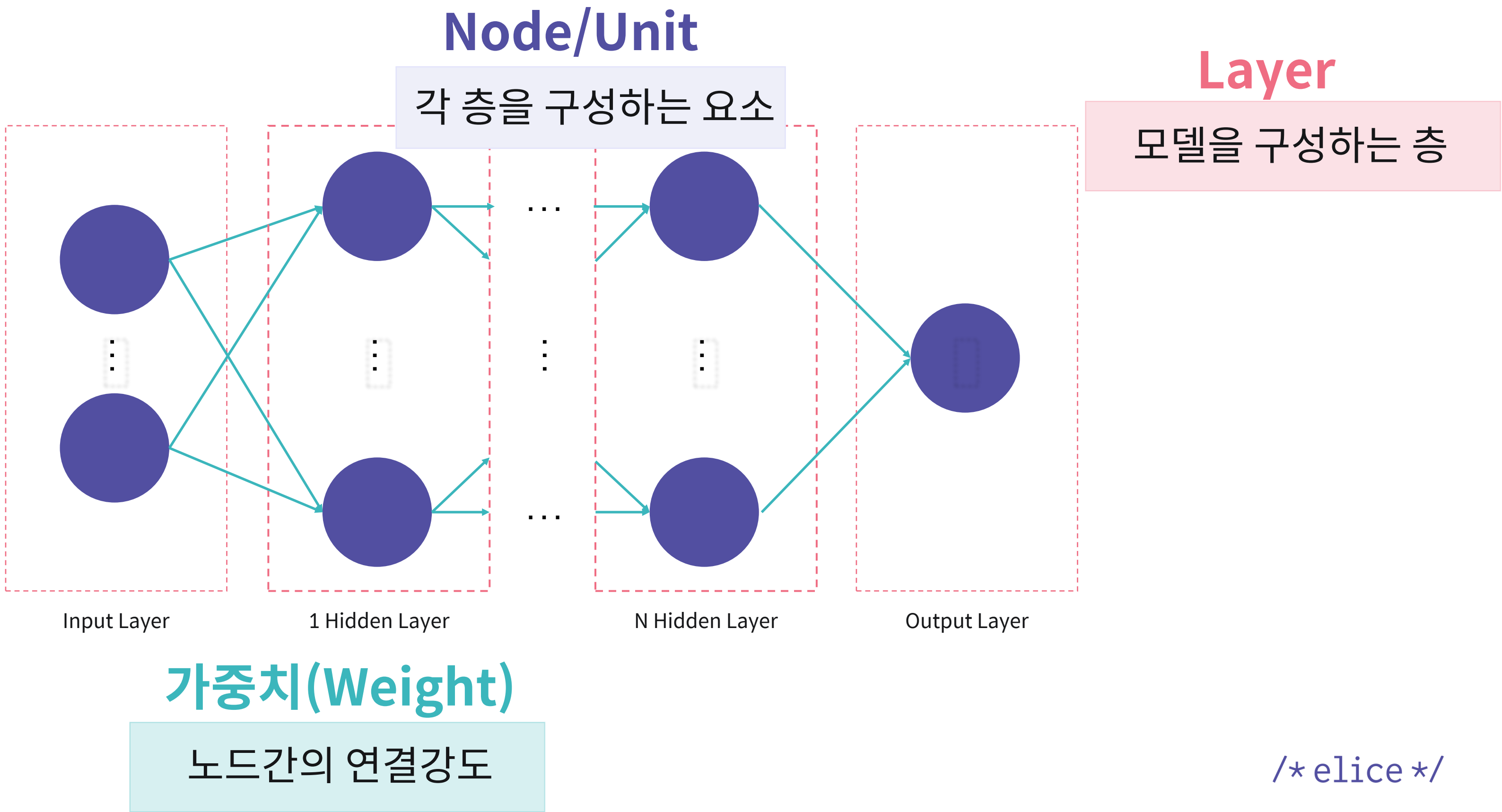
히든층이 3층 이상일 시

깊은 신경망이라는 의미의 Deep Learning 단어 사용

*/\* elice \*/*

# 01 딥러닝 모델의 학습 방법

## ✓ 딥러닝 모델의 구성 요소



# 01 딥러닝 모델의 학습 방법

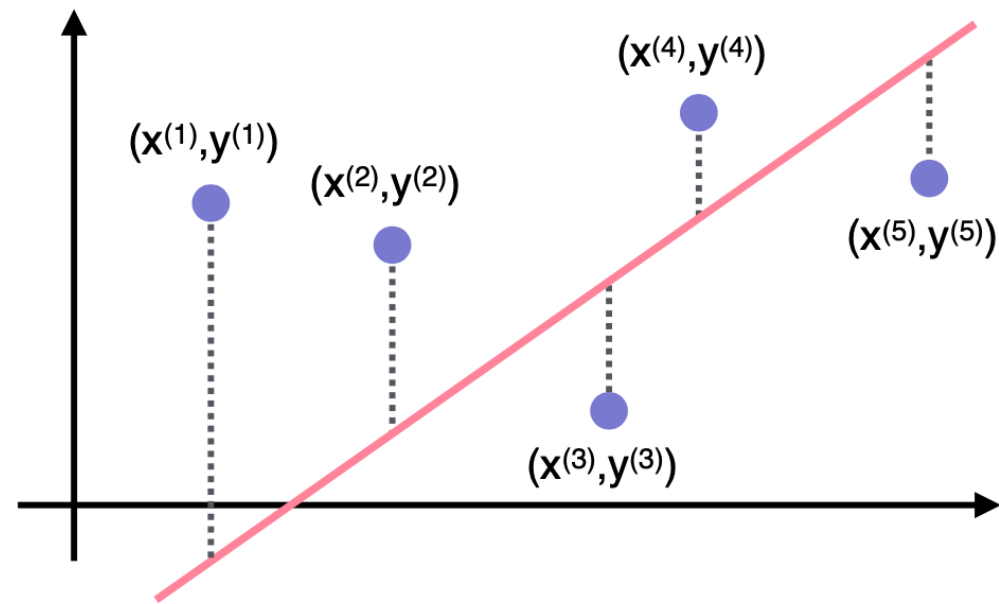
## ✓ 딥러닝 모델의 학습 방법

**Loss function**을 최소화하기 위해 **최적화** 알고리즘을 적용

# 01 딥러닝 모델의 학습 방법

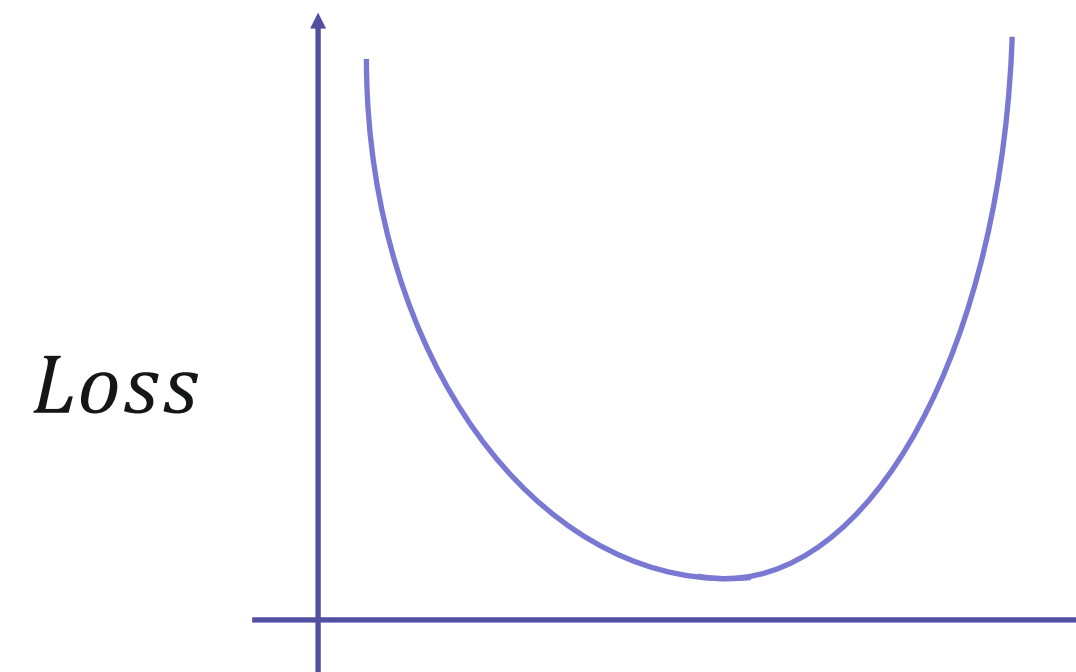
## ✓ 손실 함수(Loss Function)과 최적화(Optimization)

### Loss Function



예측값과 실제값간의 오차값

### Optimization



오차값을 최소화하는  
모델의 인자를 찾는 것

/\* elice \*/

# 01 딥러닝 모델의 학습 방법

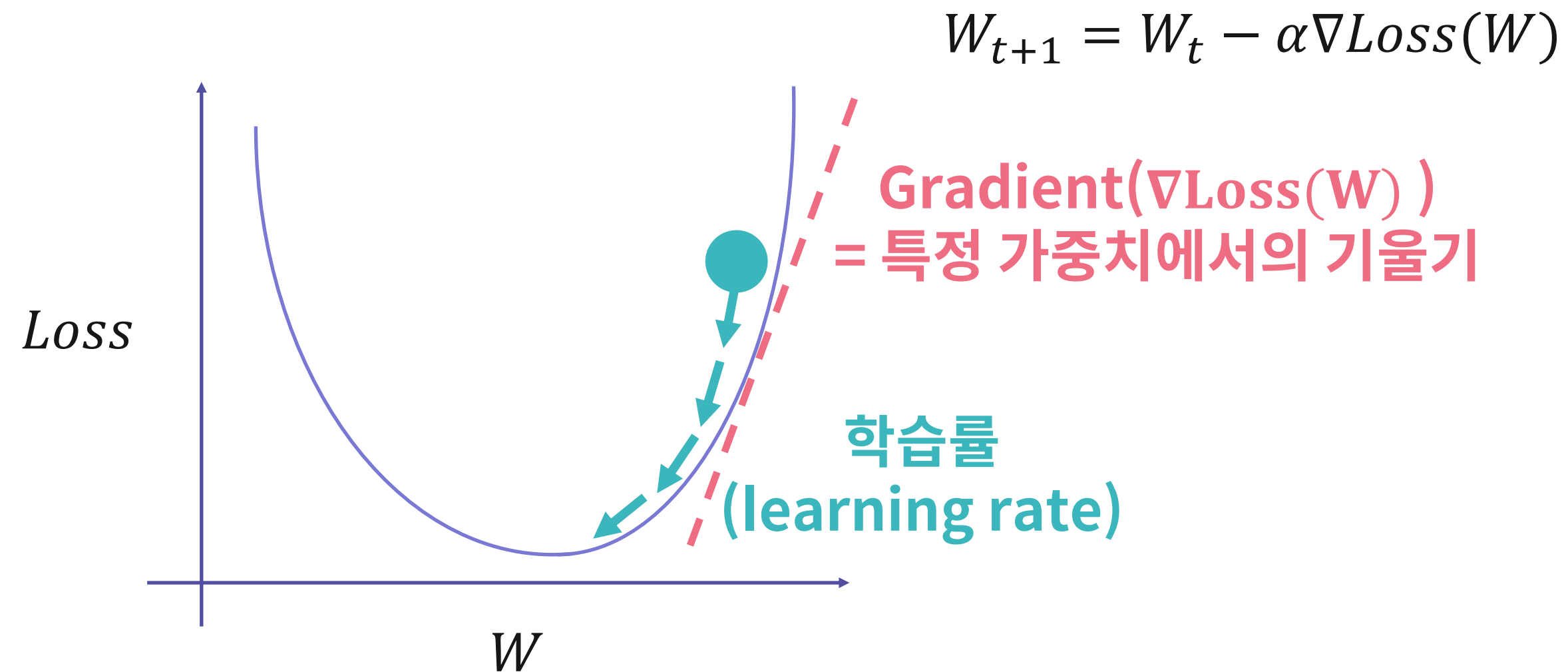
## ✓ 딥러닝 모델의 학습 방법 이해하기

예측값과 실제값 간의 오차값을 최소화하기 위해  
오차값을 최소화하는 모델의 인자를 찾는 알고리즘을 적용



# 01 딥러닝 모델의 학습 방법

## ✓ 가장 기본적인 최적화 알고리즘, Gradient Descent(GD)

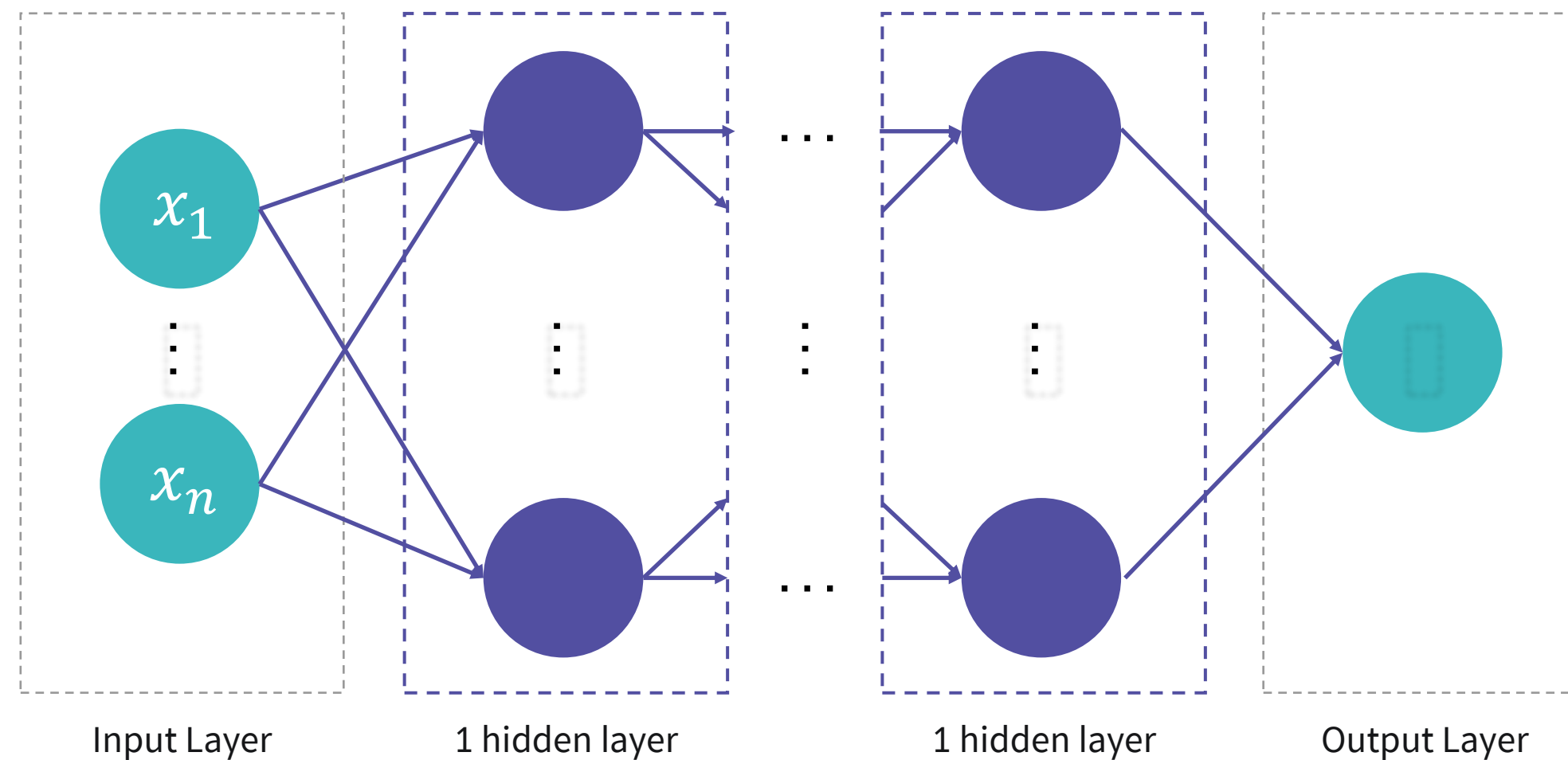


신경망의 가중치들을  $W$ 라고 했을 때,  
손실함수  $\text{Loss}(W)$ 의 값을 최소화하기 위해 기울기  $\nabla \text{Loss}(W)$ 를 이용하는 방법

/\* elice \*/

# 01 딥러닝 모델의 학습 방법

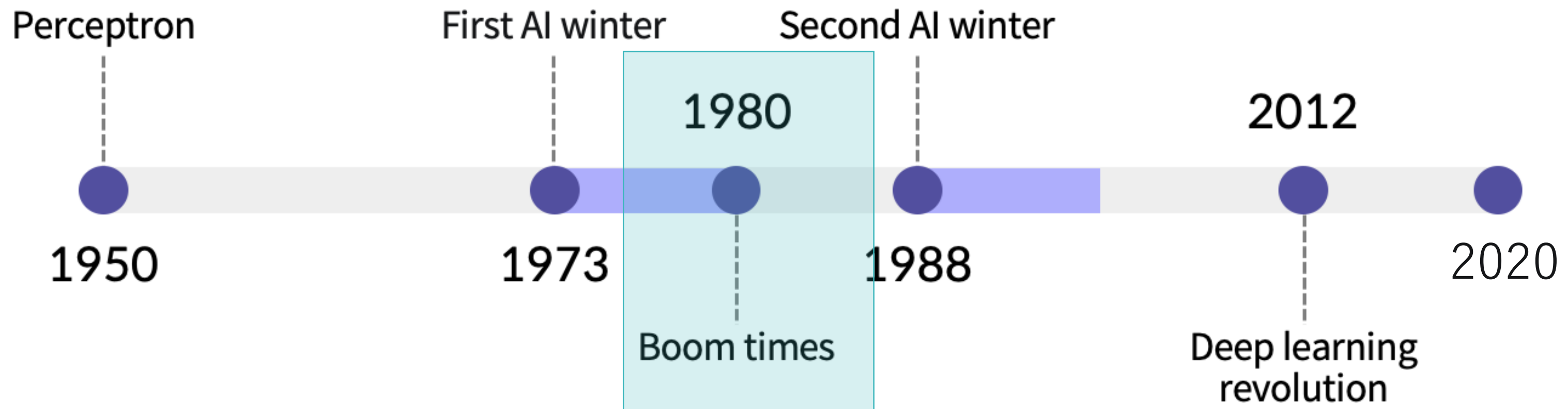
## ✓ 각 가중치들의 기울기를 구하는 방법



딥러닝에서는 **역전파(Backpropagation)**을 통해  
각 가중치들의 기울기를 구할 수 있음

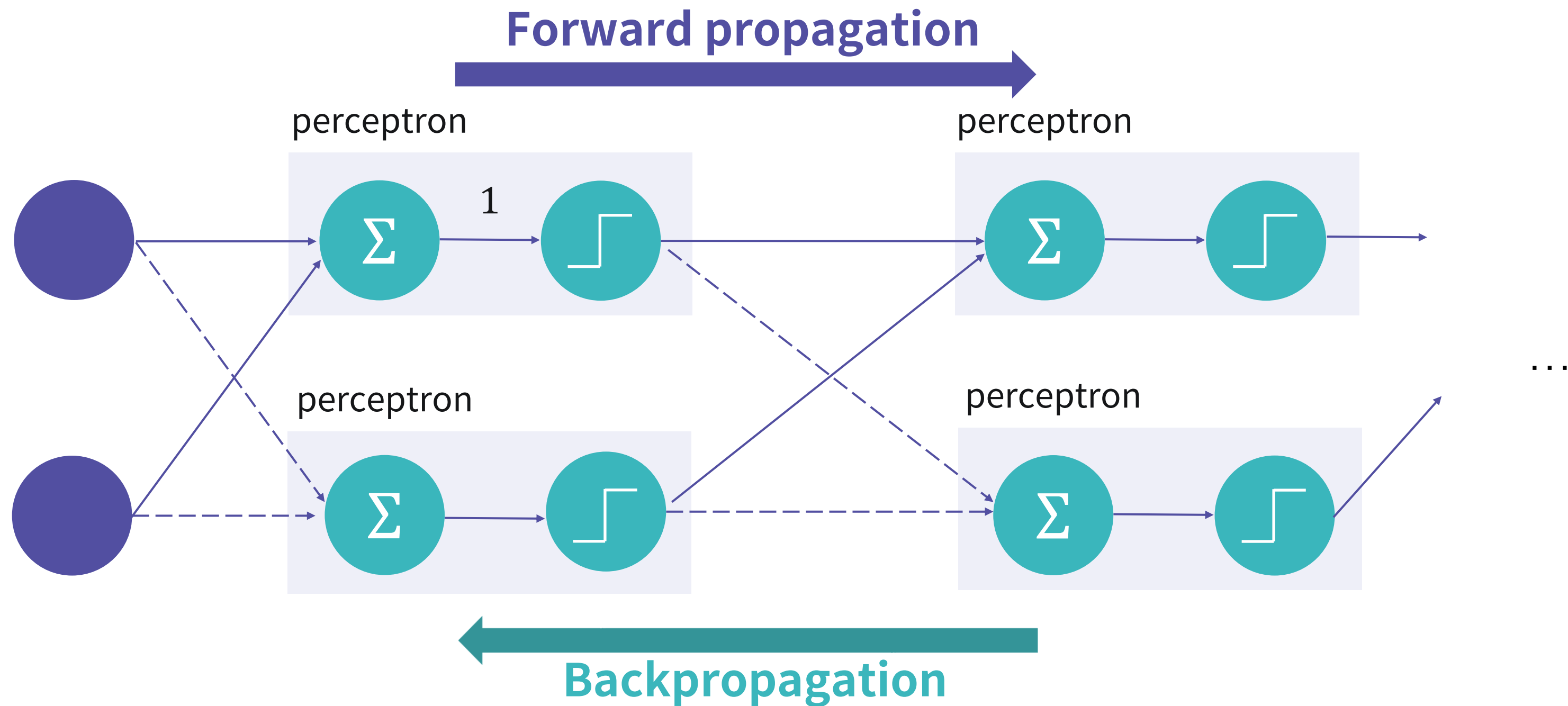
# 01 딥러닝 모델의 학습 방법

## ☑ Boom times의 배경 : 역전파(Backpropagation)



# 01 딥러닝 모델의 학습 방법

## ✓ 역전파(Backpropagation)의 정의

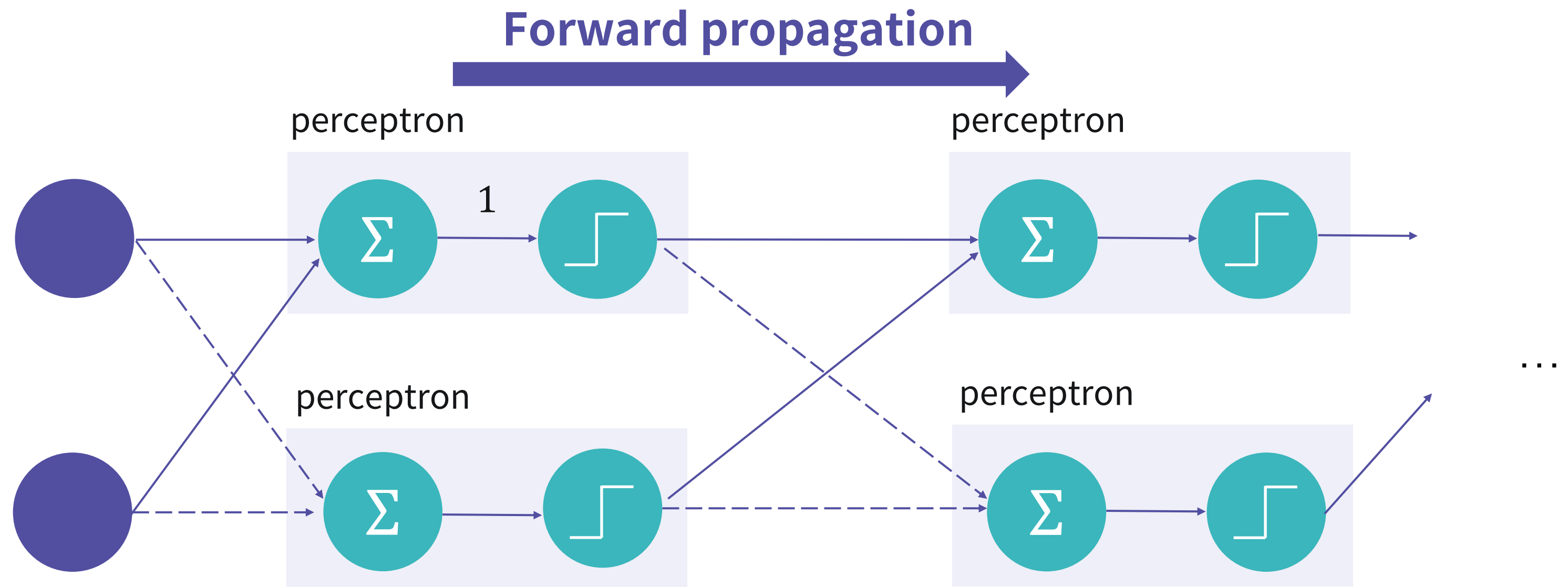


나의 목표 target 값과 실제 모델이 예측한 output 값이 얼마나 차이나는지 구한 후  
오차값을 다시 뒤로 전파해가며 변수들을 갱신하는 알고리즘

`/* elice */`

# 01 딥러닝 모델의 학습 방법

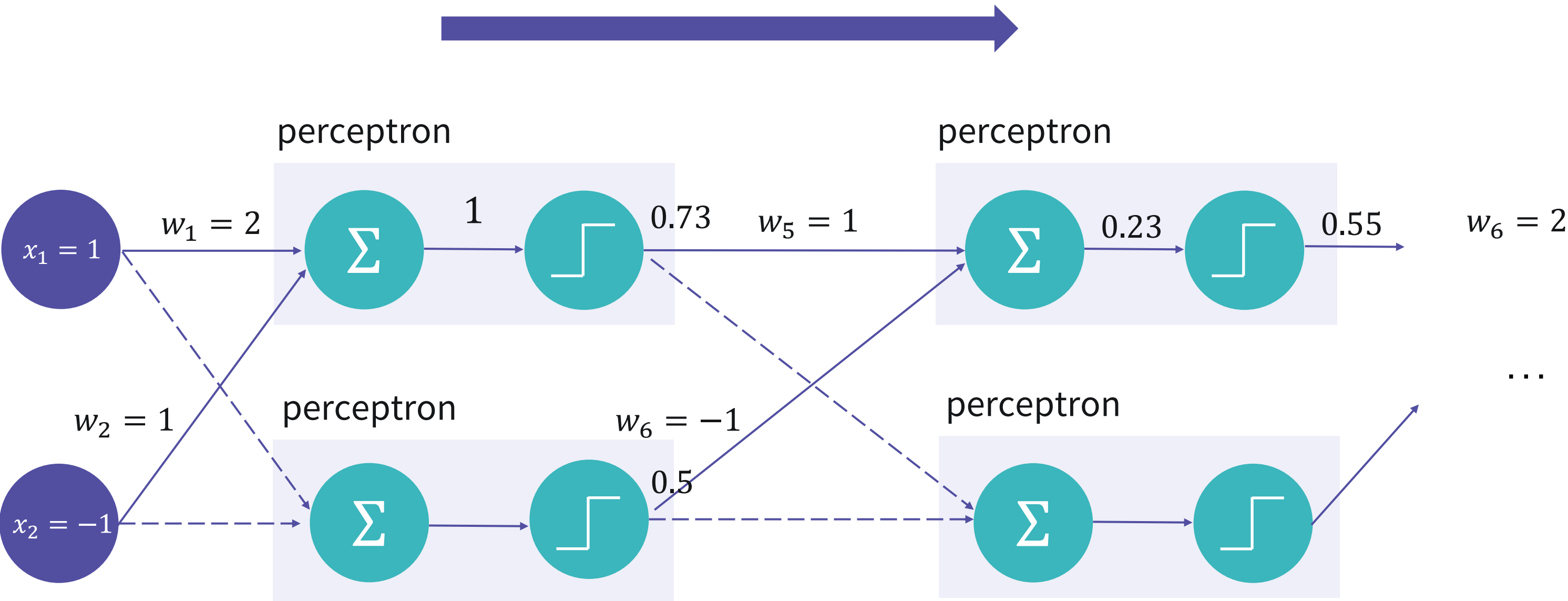
## ✓ 순전파(Forward propagation) 정의



입력 값을 바탕으로 출력 값을 계산하는 과정

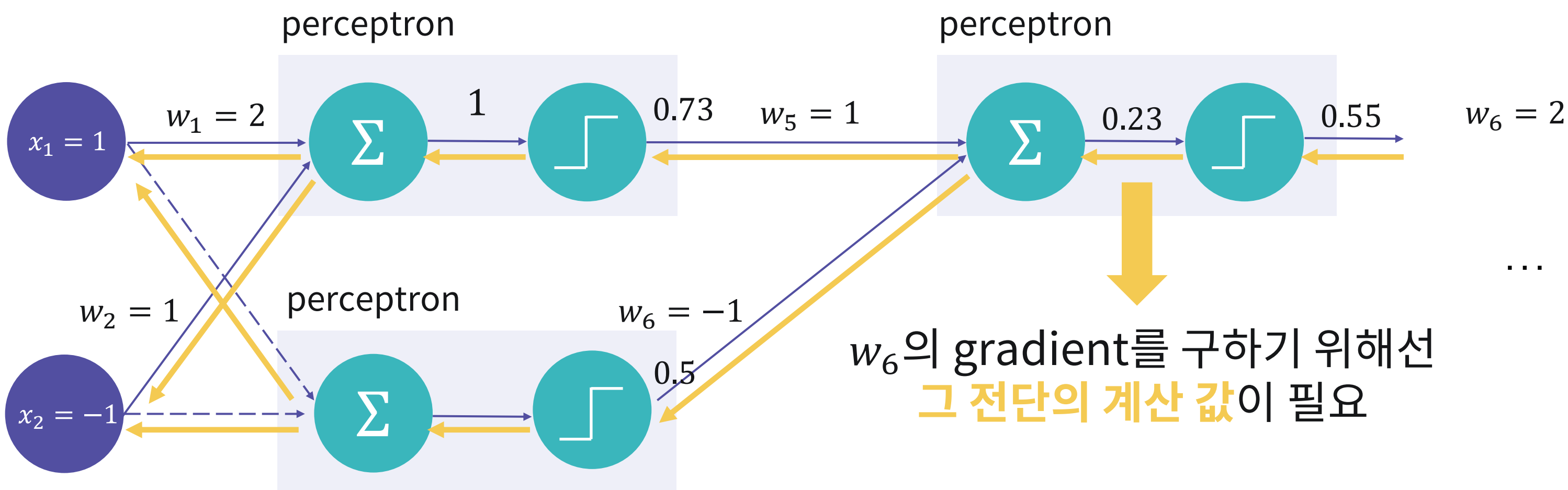
# 01 딥러닝 모델의 학습 방법

## 순전파 예시



# 01 딥러닝 모델의 학습 방법

## ✓ 역전파 예시



Forward propagation의 반대 방향으로 이루어지는 과정

02

# 텐서플로우(TensorFlow)





## 02 텐서플로우(TensorFlow)

### ✓ 딥러닝 모델 구현을 위해 학습해야 할 분야

```
503         message =
504         if not hasattr(self, '_headers_buffer'):
505             self._headers_buffer = []
506         self._headers_buffer.append((" %s %d %s\r\n" %
507             (self.protocol_version, code, message)).encode(
508                 'latin-1', 'strict'))
509
510     def send_header(self, keyword, value):
511         """Send a MIME header to the headers buffer."""
512         if self.request_version != 'HTTP/0.9':
513             if not hasattr(self, '_headers_buffer'):
514                 self._headers_buffer = []
515             self._headers_buffer.append(
516                 ("%s: %s\r\n" % (keyword, value)).encode('latin-1', 'strict'))
517
518         if keyword.lower() == 'connection':
519             if value.lower() == 'close':
520                 self.close_connection = True
521             elif value.lower() == 'keep-alive':
522                 self.close_connection = False
523
```

파이썬



하드웨어

```
17 string sInput;
18 int iLength, iN;
19 double dblTemp;
20 bool again = true;
21
22 while (again) {
23     iN = -1;
24     again = false;
25     getline(cin, sInput);
26     system("cls");
27     stringstream(sInput) >> dblTemp;
28     iLength = sInput.length();
29     if (iLength < 4) {
30         again = true;
31         continue;
32     } else if (sInput[iLength - 3] != '.') {
33         again = true;
34         continue;
35     } while (++iN < iLength) {
36         if (isdigit(sInput[iN])) {
37             continue;
38         } else if (iN == (iLength - 3)) {
39             continue;
40         }
41     }
42 }
```

C/C++

딥러닝 모델(파이썬) + 좋은 연산 장치(하드웨어) + 연산 장치 제어(C/C++)

등등 배울 것이 너무 많음

/\* elice \*/

## 02 텐서플로우(TensorFlow)

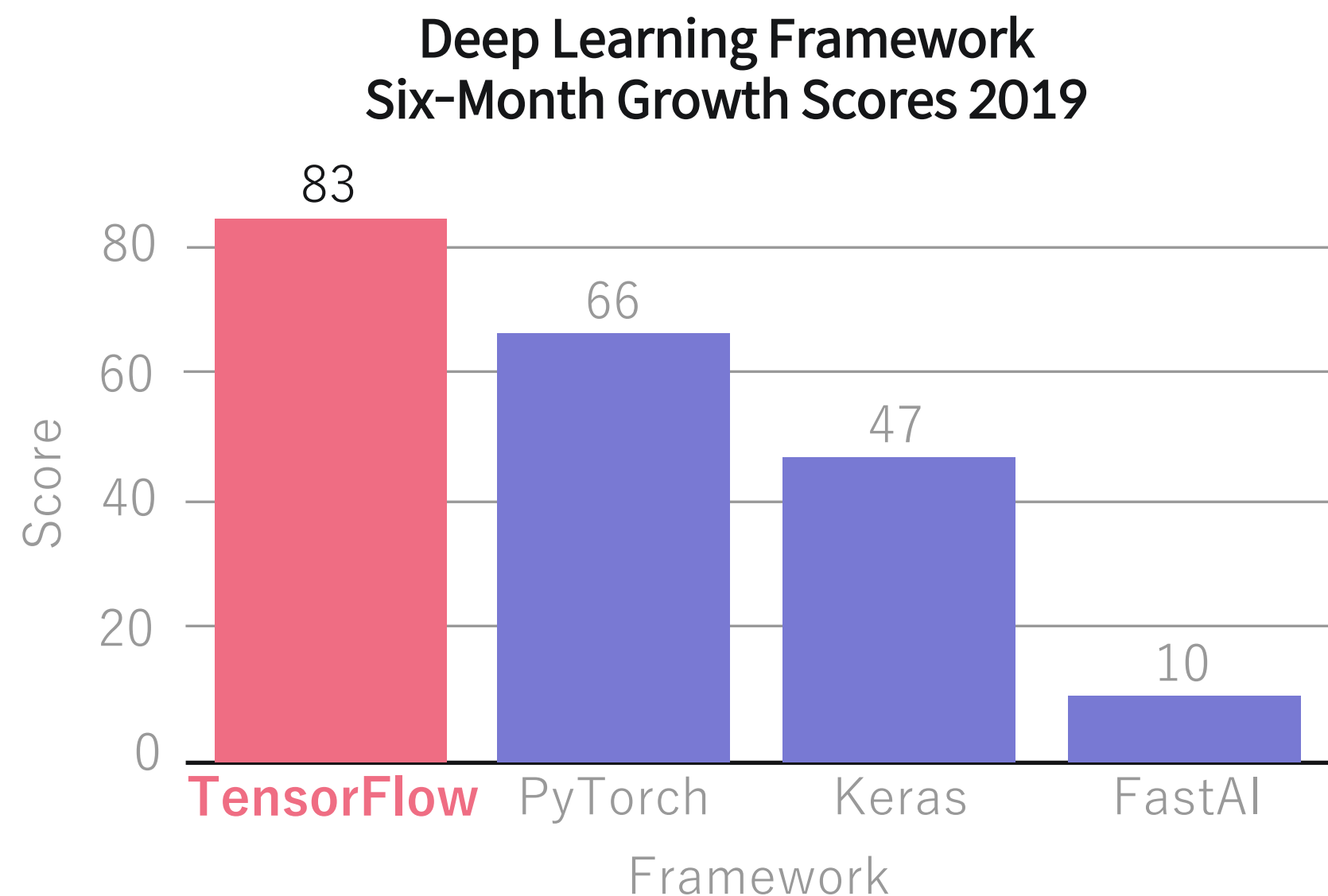
### ✓ 프레임워크를 통한 딥러닝 모델 구현



딥러닝 모델의 **학습**과 **추론**을 위한 프로그램  
딥러닝 모델을 쉽게 구현, 사용가능

## 02 텐서플로우(TensorFlow)

### ✓ 프레임워크 선택하기



가장 많이 사용되고, 빠른 성장율을 가진 프레임워크

**TensorFlow(텐서플로우) 활용**

/\* elice \*/

## 02 텐서플로우(TensorFlow)

### ✓ 텐서플로우(TensorFlow)



# TensorFlow

유연하고, 효율적이며, 확장성 있는 딥러닝 프레임워크  
대형 클러스터 컴퓨터부터 스마트폰까지 다양한 디바이스에서 동작

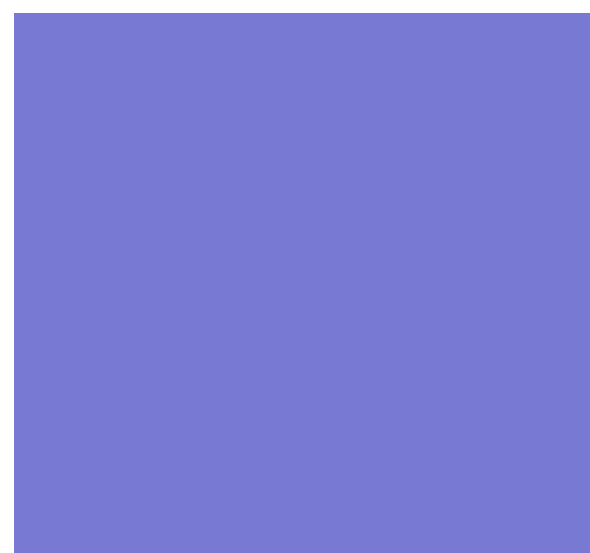
*/\* elice \*/*

## 02 텐서플로우(TensorFlow)

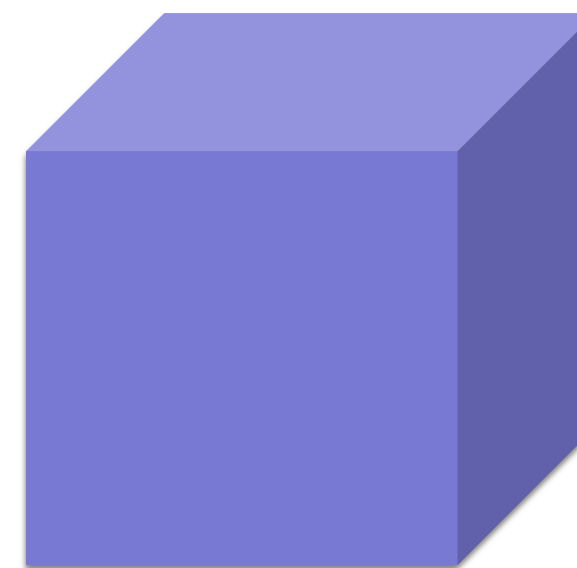
### ✓ 텐서(Tensor)



1-D Tensor



2-D Tensor



3-D Tensor

...

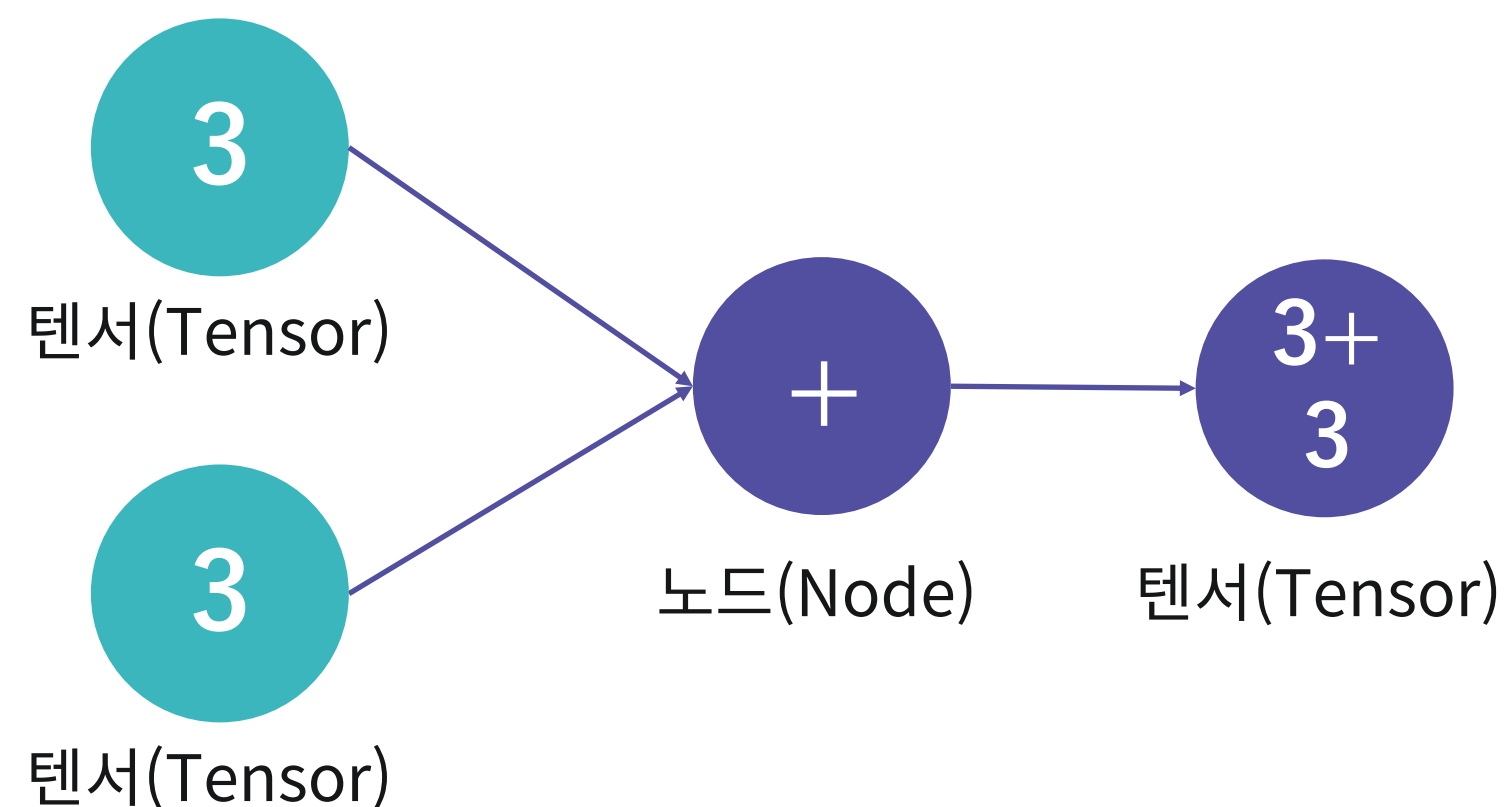
Tensor = Multidimensional Arrays = Data

딥러닝에서 텐서는 다차원 배열로 나타내는 데이터를 의미

*/\* elice \*/*

## 02 텐서플로우(TensorFlow)

### ✓ 플로우(Flow)



플로우는 데이터의 흐름을 의미

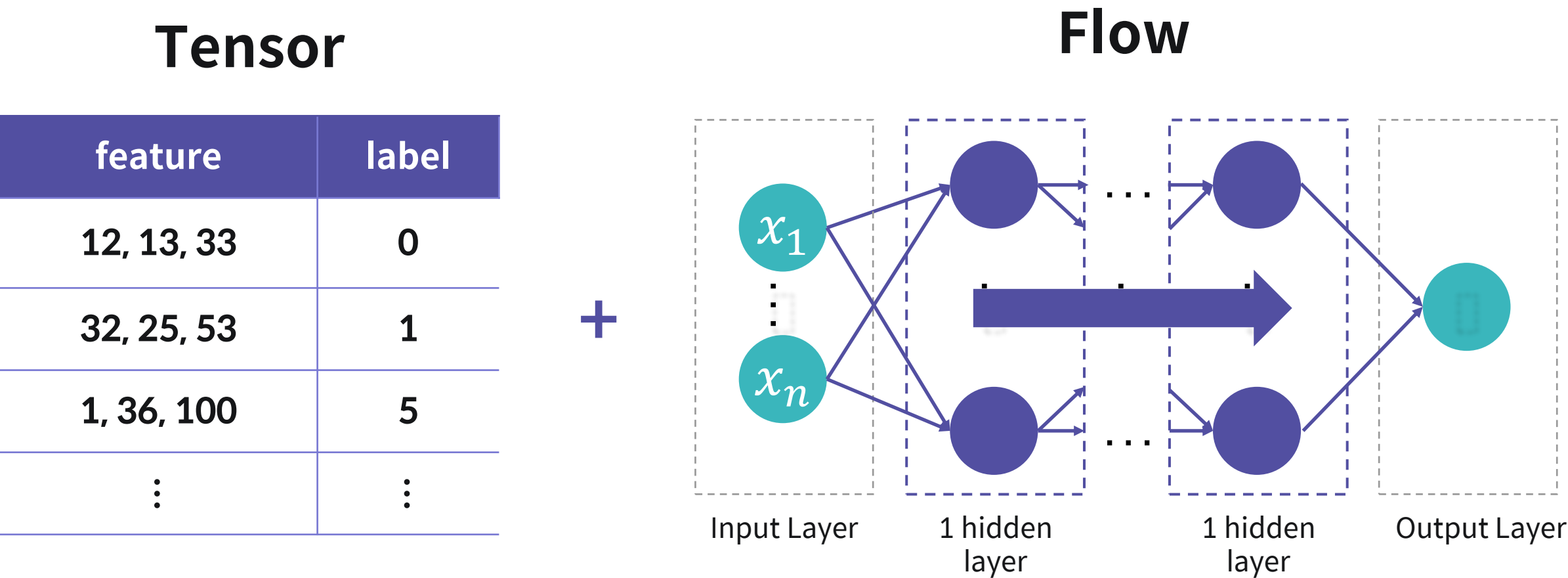
텐서플로우에서 계산은 **데이터 플로우 그래프**로 수행  
그래프를 따라 데이터가 노드를 거쳐 흘러가면서 계산

`/* elice */`



# 02 텐서플로우(TensorFlow)

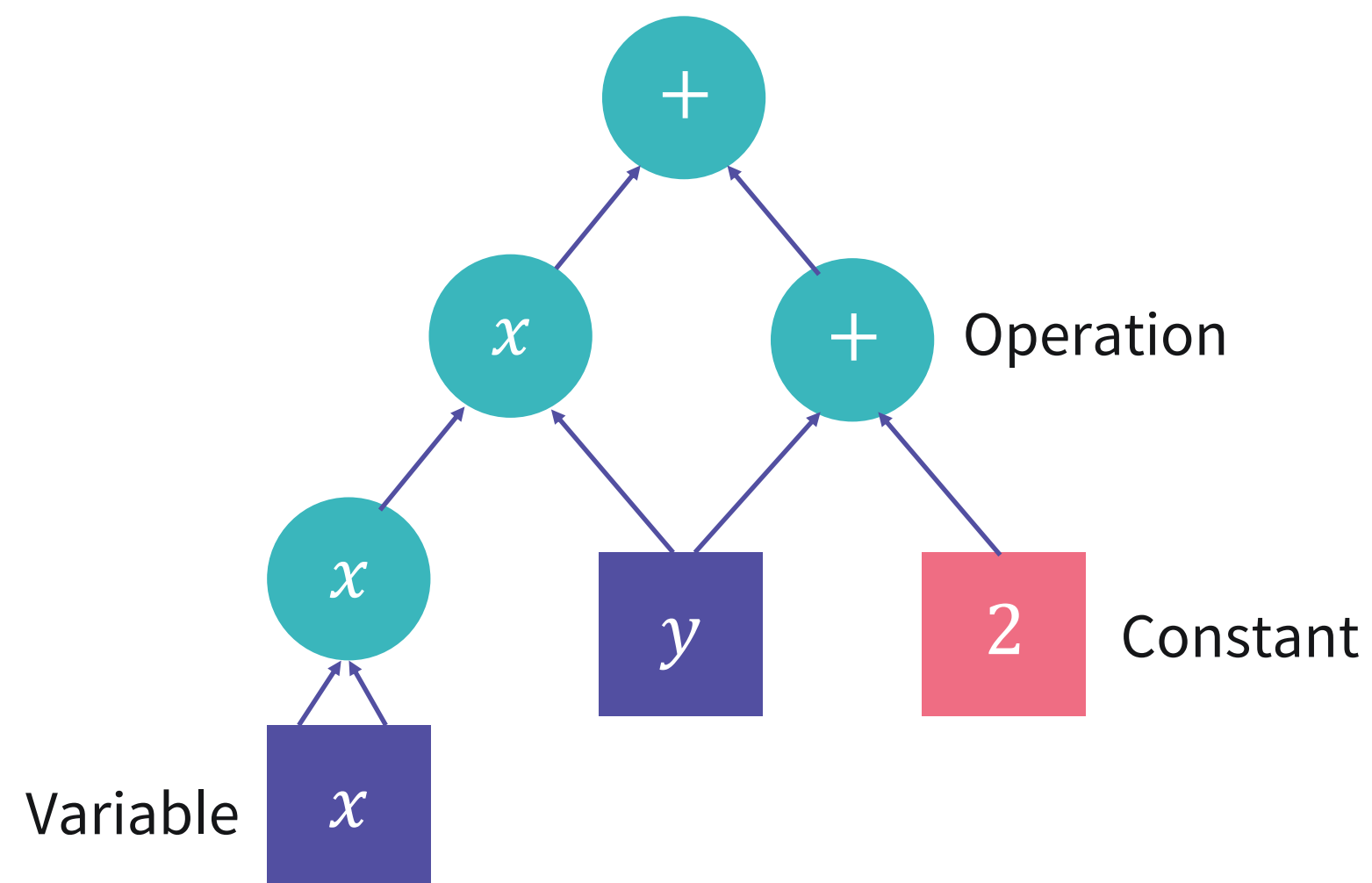
## ✓ 텐서+플로우



딥러닝에서 데이터를 의미하는 **텐서(tensor)**와  
데이터 플로우 그래프를 따라 연산이 수행되는 **형태(Flow)**의 합

## 02 텐서플로우(TensorFlow)

### ✓ 텐서플로우 version.1



이 전 텐서플로우 1.X에서는 계산을 위해  
그래프 및 세션(Session) 생성 필요

`/* elice */`



## 02 텐서플로우(TensorFlow)

- ✓ 직관적인 인터페이스의 텐서플로우 version.2



2019년 9월부터 배포된 2.0 버전 이후부터는 **즉시 실행(Eager Execution)**

기능을 통해 계산 그래프, 세션 생성 없이 실행 가능

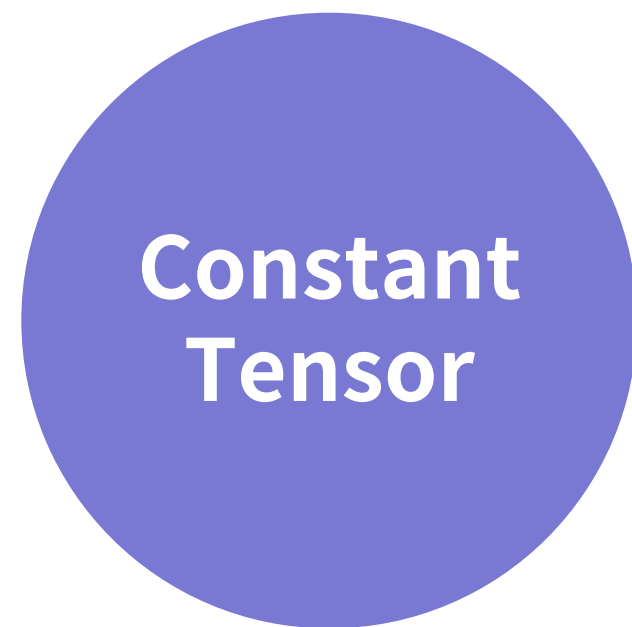
03

# 텐서플로우 기초 사용법

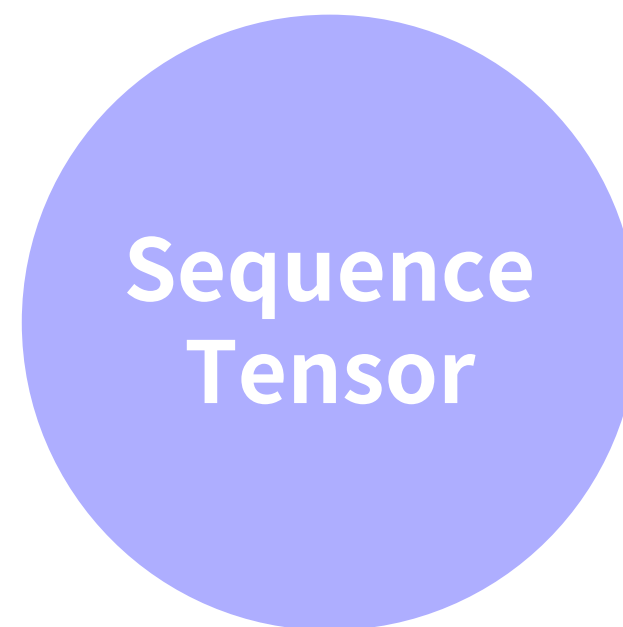


## 03 텐서플로우 기초 사용법

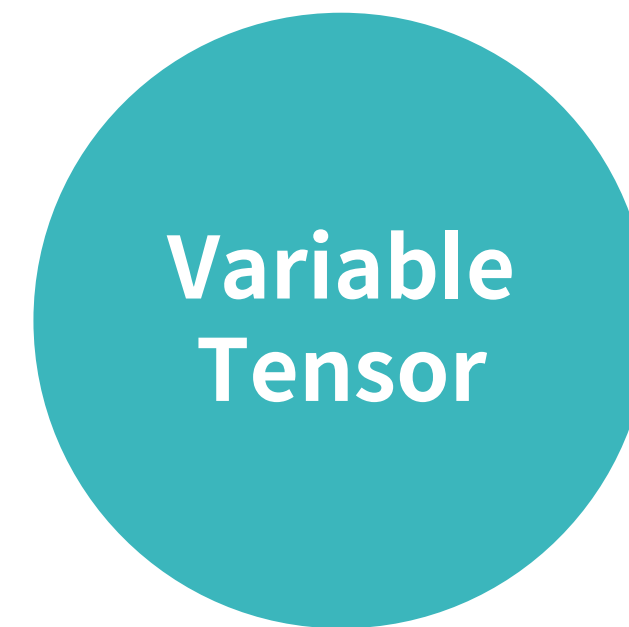
### ✓ 텐서 다뤄보기



상수 텐서



시퀀스 텐서



변수 텐서

다양한 타입의 기본 텐서 알아보기

## 03 텐서플로우 기초 사용법

### ✓ 상수 텐서(Constant Tensor)

#### Example

```
import tensorflow as tf

# 상수형 텐서 생성
tensor_a = tf.constant(value, dtype=None, shape=None, name=None)
```

**value** : 반환되는 상수값   **shape** : Tensor의 차원

**dtype** : 반환되는 Tensor 타입   **name** : 텐서 이름

*/\* elice \*/*

## 03 텐서플로우 기초 사용법

### ✓ 다양한 상수 텐서 생성하기

#### Example

```
import tensorflow as tf

# 모든 원소 값이 0인 텐서 생성
tensor_b = tf.zeros(shape, dtype=tf.float32, name=None)

# 모든 원소 값이 1인 텐서 생성
tensor_c = tf.ones(shape, dtype=tf.float32, name=None)
```

/\* elice \*/

## 03 텐서플로우 기초 사용법

### ✓ 시퀀스 텐서(Sequence Tensor)

#### Example

```
import tensorflow as tf

# start에서 stop까지 증가하는 num 개수 데이터를 가진 텐서 생성
tensor_d = tf.linspace(start, stop, num, name=None)
```

**start** : 시작 값    **stop** : 끝 값

**num** : 생성할 데이터 개수    **name** : 텐서의 이름

*/\* elice \*/*

## 03 텐서플로우 기초 사용법

### ✓ 시퀀스 텐서(Sequence Tensor)

#### Example

```
import tensorflow as tf

# start에서 limit까지 delta씩 증가하는 데이터를 가진 텐서 생성
tensor_e = tf.range(start, limit, delta, name=None)
```

**start** : 시작 값   **limit** : 끝 값

**delta** : 증가량   **name** : 텐서의 이름

`/* elice */`

## 03 텐서플로우 기초 사용법

### ✓ 변수 텐서(Variable Tensor)

#### Example

```
import tensorflow as tf

# 변수형 텐서 생성

tensor_f = tf.Variable(initial_value=None, dtype= None, name= None )
```

**initial\_value** : 초기 값

**dtype** : 반환되는 Tensor 타입    **name** : 텐서의 이름

*/\* elice \*/*



## 03 텐서플로우 기초 사용법

### ✓ 상수 텐서 생성 및 수식 정의

#### Example

```
import tensorflow as tf

# 상수 텐서 생성
a = tf.constant([1,0],dtype=tf.float32)

# 수식 정의
def forward(x):
    return W * x + b
```

/\* elice \*/

## 03 텐서플로우 기초 사용법

### ✓ 정의된 수식을 활용한 연산

#### Example

```
# 텐서 계산 및 출력  
output = forward(a)  
print(output)
```

#### Result

```
tf.Tensor(  
[[1. 0.]  
 [1. 0.]], shape=(2, 2), dtype=float32)
```

/\* elice \*/

04

# 텐서플로우로 딥러닝 모델 구현하기



## 04 텐서플로우로 딥러닝 모델 구현하기

### ✓ 딥러닝 모델 구현 순서

1. 데이터셋 준비하기
2. 딥러닝 모델 구축하기
3. 모델 학습시키기
4. 평가 및 예측하기

## 04 텐서플로우로 딥러닝 모델 구현하기

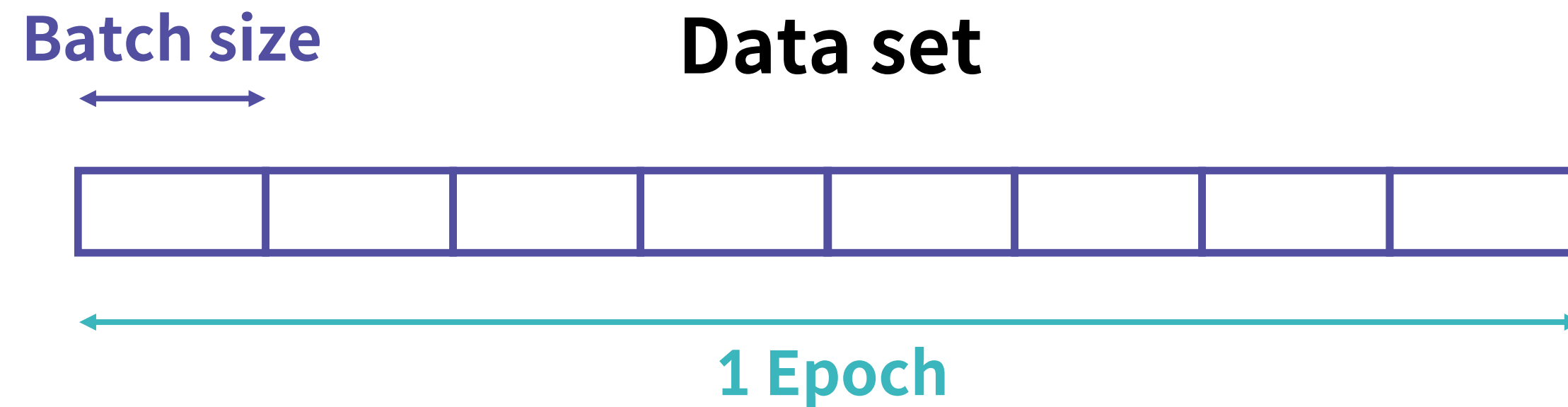
### ✓ 1. 데이터셋 준비하기 : Epoch와 Batch

**Epoch:** 한 번의 epoch는 전체 데이터 셋에 대해 **한 번 학습을 완료한 상태**

**Batch:** 나뉜 데이터 셋 (보통 mini-batch라고 표현)  
**iteration**는 epoch를 나누어서 실행하는 횟수를 의미

## 04 텐서플로우로 딥러닝 모델 구현하기

### ✓ Epoch와 Batch 예시



Ex) 총 데이터가 1000개, Batch size = 100

- 1 iteration = 100개 데이터에 대해서 학습
- 1 epoch =  $1000 / \text{Batch size} = 10$  iteration

## 04 텐서플로우로 딥러닝 모델 구현하기

### ✔ 데이터셋 준비하기 코드 예시

#### Example

```
data = np.random.sample((100,2))
labels = np.random.sample((100,1))

# numpy array로부터 데이터셋 생성
dataset = tf.data.Dataset.from_tensor_slices((data, labels))
dataset = dataset.batch(32)
```

Dataset API를 사용하여 딥러닝 모델 용 데이터 셋을 생성

*/\* elice \*/*

## 04 텐서플로우로 딥러닝 모델 구현하기

### ✓ 2. 딥러닝 모델 구축하기 : 고수준 API 활용



텐서플로우의 패키지로 제공되는 고수준 API  
딥러닝 모델을 간단하고 빠르게 구현 가능



## 04 텐서플로우로 딥러닝 모델 구현하기

### ✓ 딥러닝 모델 구축을 위한 Keras 메소드(1)

모델 클래스 객체 생성

```
tf.keras.models.Sequential()
```

모델의 각 Layer 구성

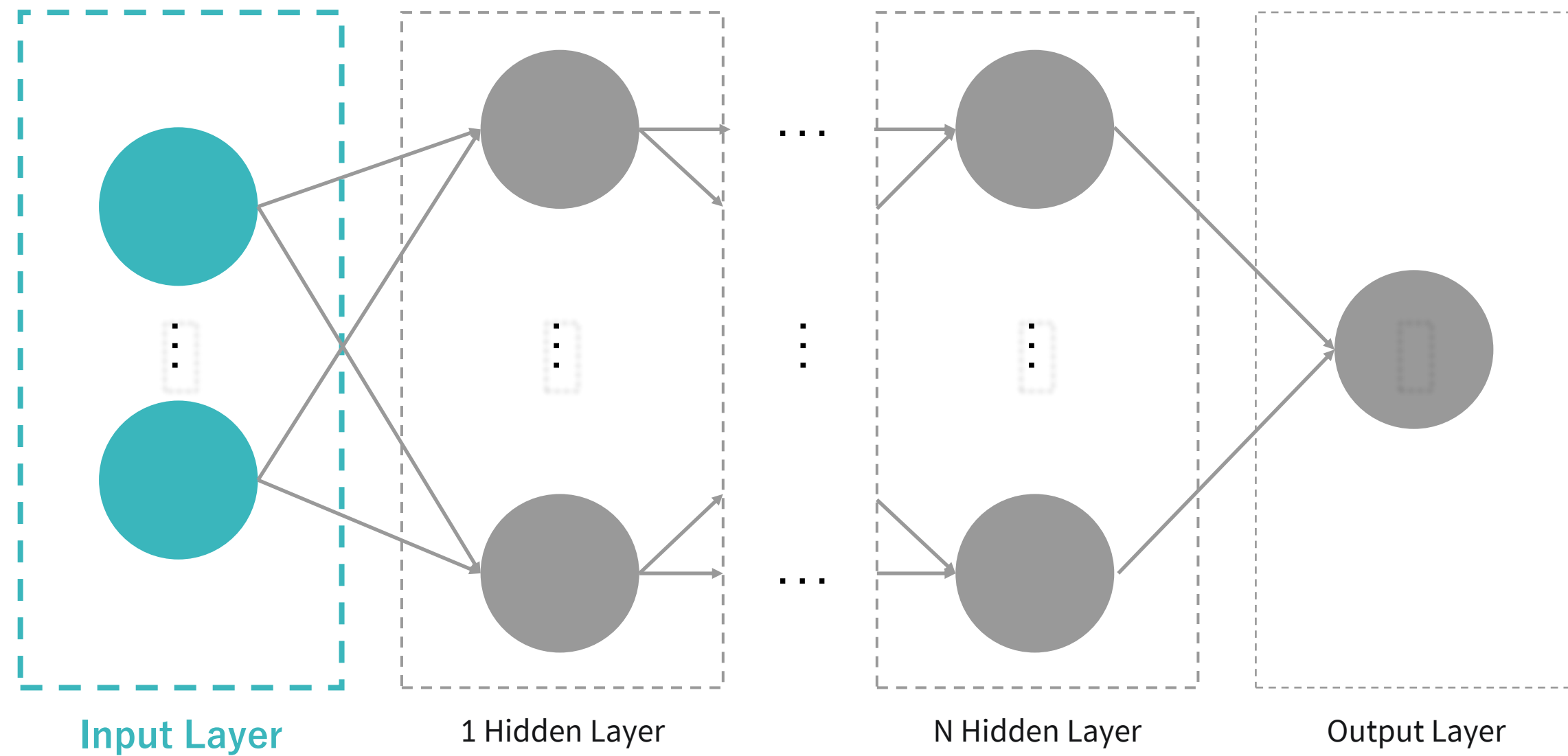
```
tf.keras.layers.Dense(units, activation)
```

- units : 레이어 안의 Node의 수
- activation : 적용할 activation 함수 설정

*/\* elice \*/*

## 04 텐서플로우로 딥러닝 모델 구현하기

### ✓ Input Layer의 입력 형태 지정하기



첫 번째 즉, Input Layer는 입력 형태에 대한 정보를 필요로 함  
**input\_shape** / **input\_dim** 인자 설정하기

*/\* elice \*/*

## 04 텐서플로우로 딥러닝 모델 구현하기

### ✓ 모델 구축하기 코드 예시(1)

#### Example

```
model = tf.keras.models.Sequential([  
    tf.keras.layers.Dense(10, input_dim=2, activation='sigmoid'),  
    tf.keras.layers.Dense(10, activation='sigmoid'),  
    tf.keras.layers.Dense(1, activation='sigmoid'),  
])
```

/\* elice \*/

## 04 텐서플로우로 딥러닝 모델 구현하기

### ✓ 딥러닝 모델 구축을 위한 Keras 메소드(2)

모델에 Layer 추가하기

```
[model].add(tf.keras.layers.Dense(units, activation))
```

- units : 레이어 안의 Node의 수
- activation : 적용할 activation 함수 설정

*/\* elice \*/*

## 04 텐서플로우로 딥러닝 모델 구현하기

### ✓ 모델 구축하기 코드 예시(2)

#### Example

```
model = tf.keras.models.Sequential()

model.add(tf.keras.layers.Dense(10, input_dim=2, activation='sigmoid'))
model.add(tf.keras.layers.Dense(10, activation='sigmoid'))
model.add(tf.keras.layers.Dense(1, activation='sigmoid'))
```

/\* elice \*/

## 04 텐서플로우로 딥러닝 모델 구현하기

### ✓ 3. 딥러닝 모델 학습시키기 : Keras 메소드

모델 학습 방식을 설정하기 위한 함수

```
[model].compile(optimizer, loss)
```

- optimizer : 모델 학습 최적화 방법
- loss : 손실 함수 설정

모델을 학습시키기 위한 함수

```
[model].fit(x, y)
```

- x : 학습 데이터
- y : 학습 데이터의 label

`/* elice */`

## 04 텐서플로우로 딥러닝 모델 구현하기

### ✓ 딥러닝 모델 학습시키기 코드 예시

#### Example

```
model.compile(loss='mean_squared_error', optimizer='SGD')  
model.fit(dataset, epochs=100)
```

`/* elice */`

## 04 텐서플로우로 딥러닝 모델 구현하기

### ✓ 4. 평가 및 예측하기 : Keras 메소드

모델을 평가하기 위한 메소드

```
[model].evaluate(x, y)
```

- x : 테스트 데이터
- y : 테스트 데이터의 label

모델로 예측을 수행하기 위한 함수

```
[model].predict(x)
```

- x : 예측하고자 하는 데이터

*/\* elice \*/*



## 04 텐서플로우로 딥러닝 모델 구현하기

### ✓ 평가 및 예측하기 코드 예시

#### Example

```
# 테스트 데이터 준비하기

dataset_test = tf.data.Dataset.from_tensor_slices((data_test, labels_test))
dataset_test = dataset.batch(32)

# 모델 평가 및 예측하기

model.evaluate(dataset_test)
predicted_labels_test = model.predict(data_test)
```

# Credit

/\* elice \*/

코스 매니저

**임승연**

콘텐츠 제작자

**임승연**

강사

**오혜연 교수님**

감수자

**이해솔**

디자인

**황보영**

# Contact

TEL

070-4633-2015

WEB

<https://elice.io>

E-MAIL

[contact@elice.io](mailto:contact@elice.io)

