

Lecture 2: What is machine learning?

- examples

기계학습개론
박상효

학습목표

- 기계학습의 분류를 이해
- 기계학습의 사례를 경험

핵심용어

- 지도학습
- 비지도학습

지도 방식에 따른 ML 유형

- 지도학습(Supervised learning)
- 비지도학습(Unsupervised learning)
- 준지도학습(Semi-supervised learning)
- 강화학습(Reinforcement learning)

Classes of Learning Problems

Supervised Learning

Data: (x, y)

x is data, y is label

Goal: Learn function to map
 $x \rightarrow y$

Apple example:



This thing is an apple.

Unsupervised Learning

Data: x

x is data, no labels!

Goal: Learn underlying
structure

Apple example:



This thing is like
the other thing.

Reinforcement Learning

Data: state-action pairs

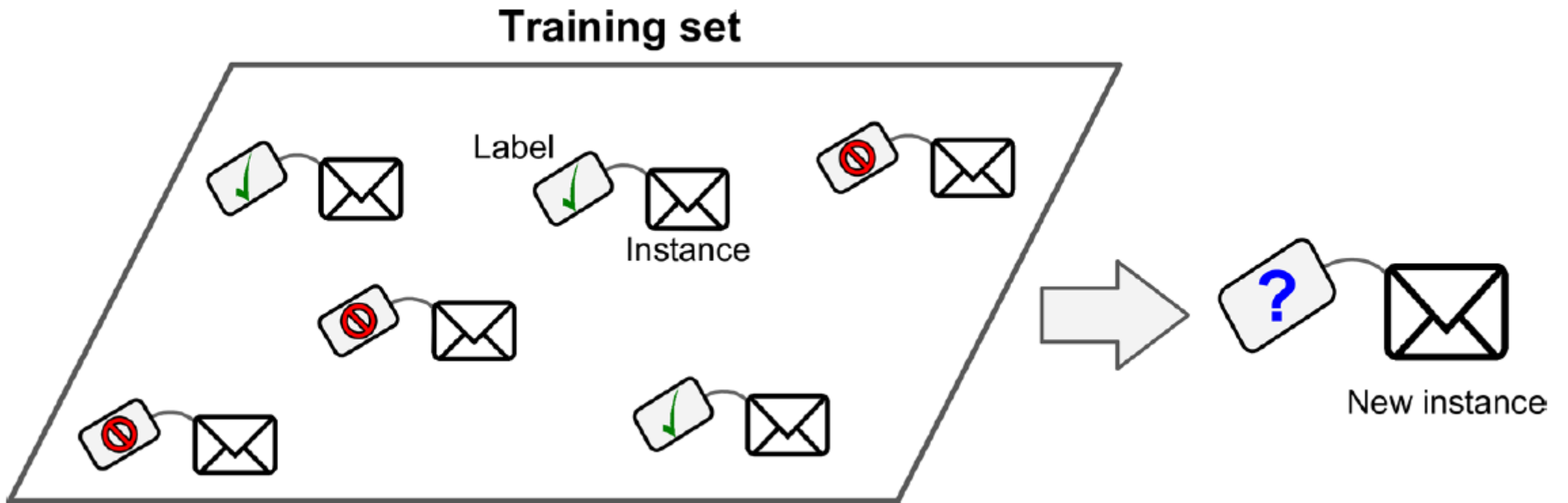
Goal: Maximize future rewards
over many time steps

Apple example:



Eat this thing because it
will keep you alive.

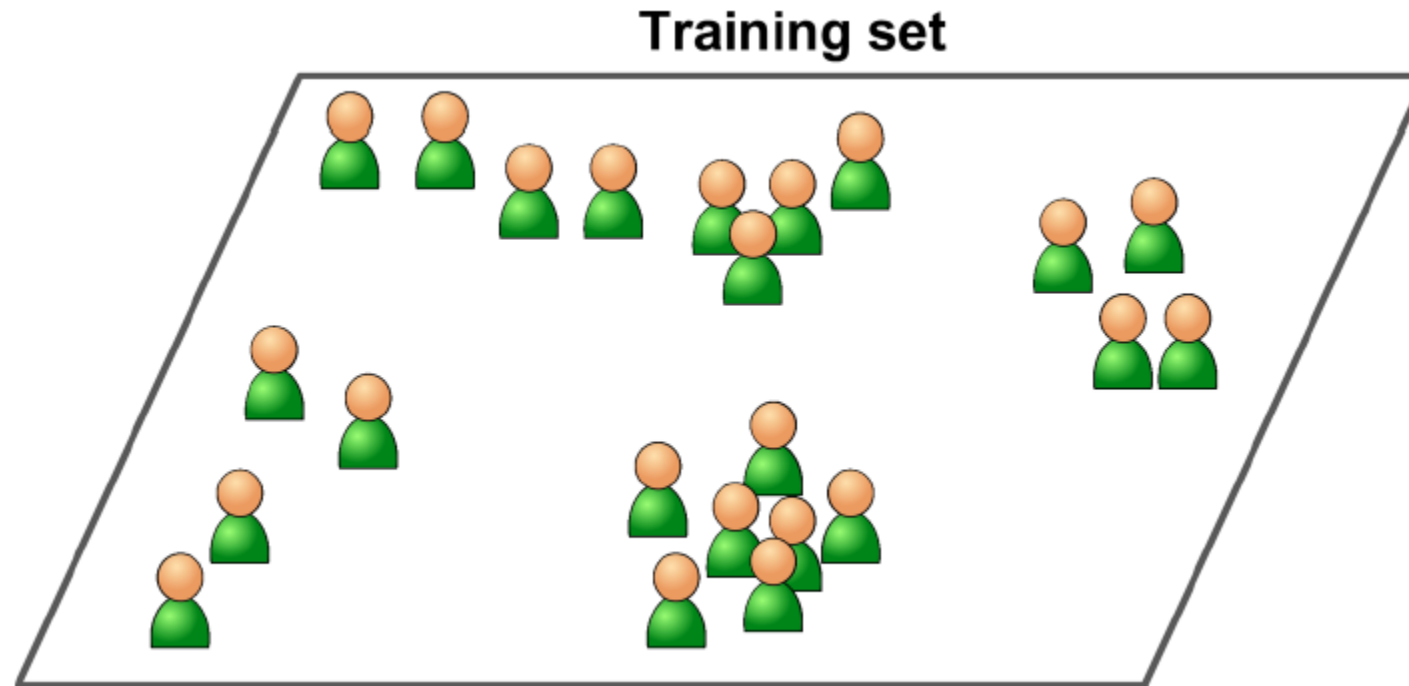
Supervised learning



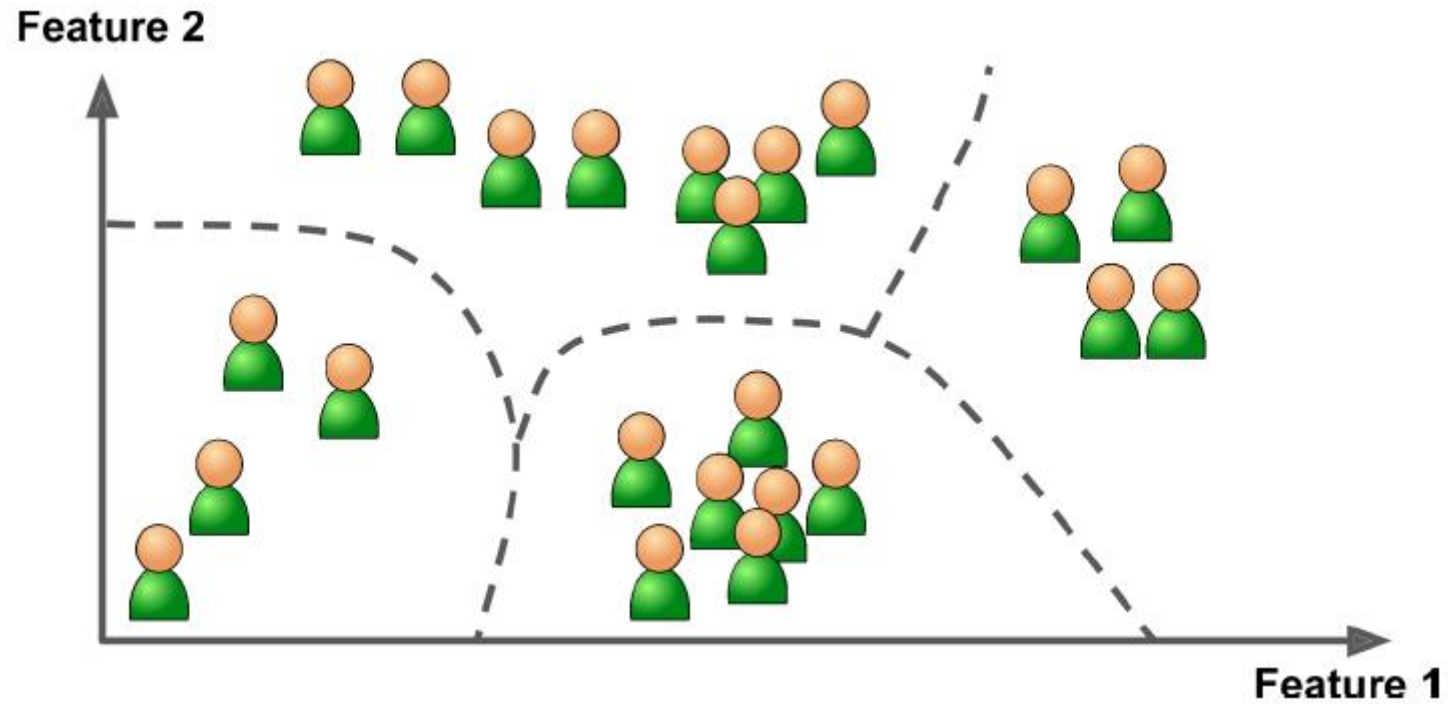
Supervised learning

- 특징 벡터 \mathbb{X} 와 목표값 \mathbb{Y} 가 모두 주어진 상황
- Classification, Regression

Unsupervised learning



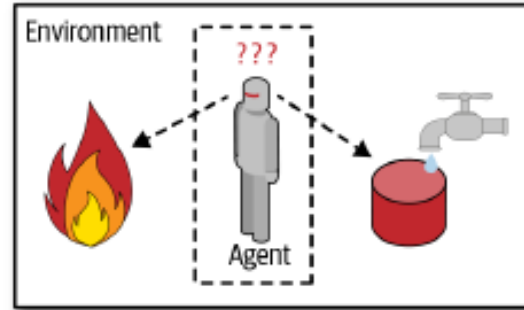
Unsupervised learning



Unsupervised learning

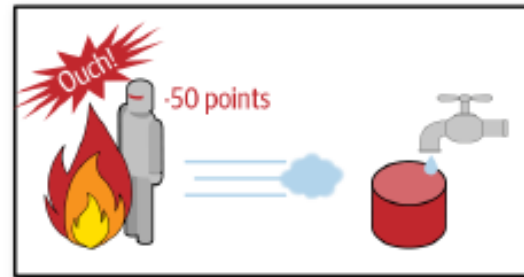
- 특징 벡터 \mathbf{x} 는 주어지는데 목표값 \mathbf{y} 가 주어지지 않는 상황
- Clustering, Dimensionality reduction, Anomaly/Novelty detection

Reinforcement learning



1 Observe

2 Select action using policy



3 Action!

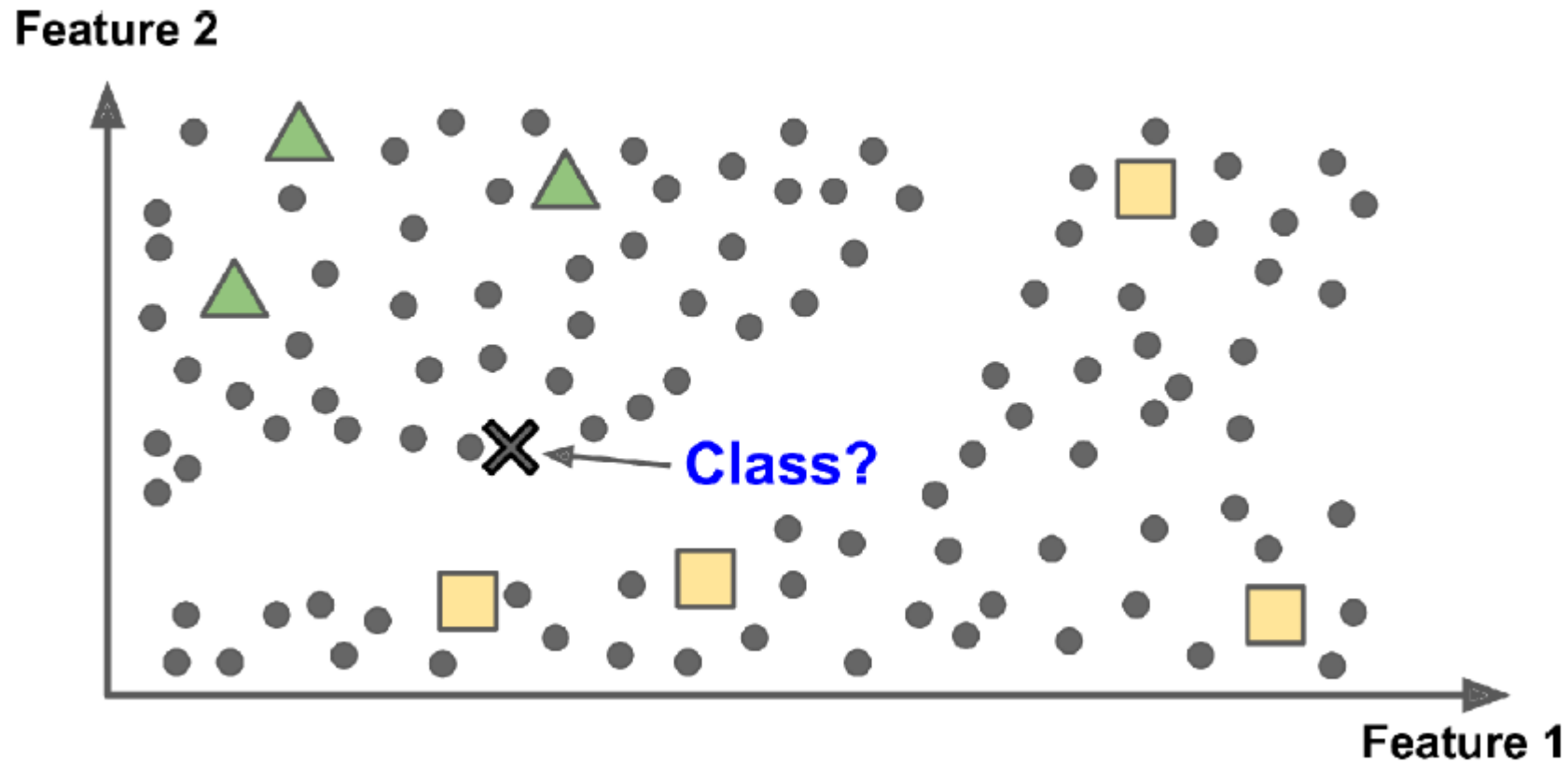
4 Get reward or penalty



5 Update policy (learning step)

6 Iterate until an optimal policy is found

Semi-supervised learning



Semi-supervised learning

- 일부는 \mathbb{X} 와 \mathbb{Y} 를 모두 가지지만, 나머지는 \mathbb{X} 만 가진 상황
- \mathbb{X} 의 수집은 쉽지만, \mathbb{Y} 는 수작업이 필요한 경우 유용함.

다양한 기준에 따른 유형

- 오프라인 학습과 온라인 학습
- 결정론적(deterministic) 학습과 스토캐스틱(stochastic) 학습
 - 결정론적에서는 같은 데이터를 가지고 다시 학습하면 **같은 예측기**가 만들어짐
 - 스토캐스틱 학습은 학습 과정에서 난수를 사용하므로 같은 데이터로 다시 학습하면 **다른 예측기**가 만들어짐.
- 분별(discriminative) 모델과 생성(generative) 모델
 - 분별 모델은 분류 예측에만 관심. 즉 $P(y|\mathbf{x})$ 의 추정에 관심
 - 생성 모델은 $P(\mathbf{x})$ 또는 $P(\mathbf{x}|y)$ 를 추정함

기계학습 예시(수식)

기1.4, 1.7

간단한 기계학습 예

- 선형 회귀 문제
 - 식 (1.2)의 직선 모델을 사용하므로 두 개의 매개변수 $\Theta = (w, b)^T$

$$y = wx + b \quad (1.2)$$

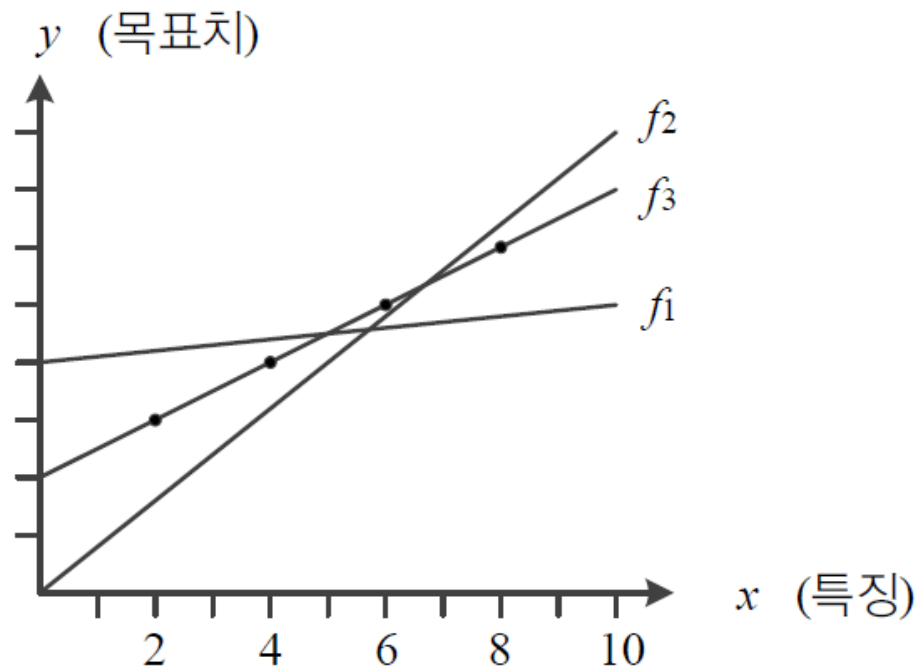


그림 1-4 간단한 기계 학습 예제

간단한 기계학습 예

- 목적 함수 objective function (또는 비용 함수 cost function)
 - 식 (1.8)은 선형 회귀를 위한 목적 함수
 - $f_{\Theta}(\mathbf{x}_i)$ 는 예측함수의 출력, y_i 는 예측함수가 맞추어야 하는 목표값이므로 $f_{\Theta}(\mathbf{x}_i) - y_i$ 는 오차
 - 식 (1.8)을 평균제곱오차 MSE(mean squared error)라 부름

$$J(\Theta) = \frac{1}{n} \sum_{i=1}^n (f_{\Theta}(\mathbf{x}_i) - y_i)^2 \quad (1.8)$$

[예제 1-1]

■ 훈련집합

$$\mathbb{X} = \{x_1 = (2.0), x_2 = (4.0), x_3 = (6.0), x_4 = (8.0)\},$$

$$\mathbb{Y} = \{y_1 = 3.0, y_2 = 4.0, y_3 = 5.0, y_4 = 6.0\}$$

■ 초기 직선의 매개변수 $\theta_1 = (0.1, 4.0)^T$ 라 가정

$$x_1, y_1 \rightarrow (f_{\theta_1}(2.0) - 3.0)^2 = ((0.1 * 2.0 + 4.0) - 3.0)^2 = 1.44$$

$$x_2, y_2 \rightarrow (f_{\theta_1}(4.0) - 4.0)^2 = ((0.1 * 4.0 + 4.0) - 4.0)^2 = 0.16$$

$$x_3, y_3 \rightarrow (f_{\theta_1}(6.0) - 5.0)^2 = ((0.1 * 6.0 + 4.0) - 5.0)^2 = 0.16$$

$$x_4, y_4 \rightarrow (f_{\theta_1}(8.0) - 6.0)^2 = ((0.1 * 8.0 + 4.0) - 6.0)^2 = 1.44$$

$$\longrightarrow J(\theta_1) = 0.8$$

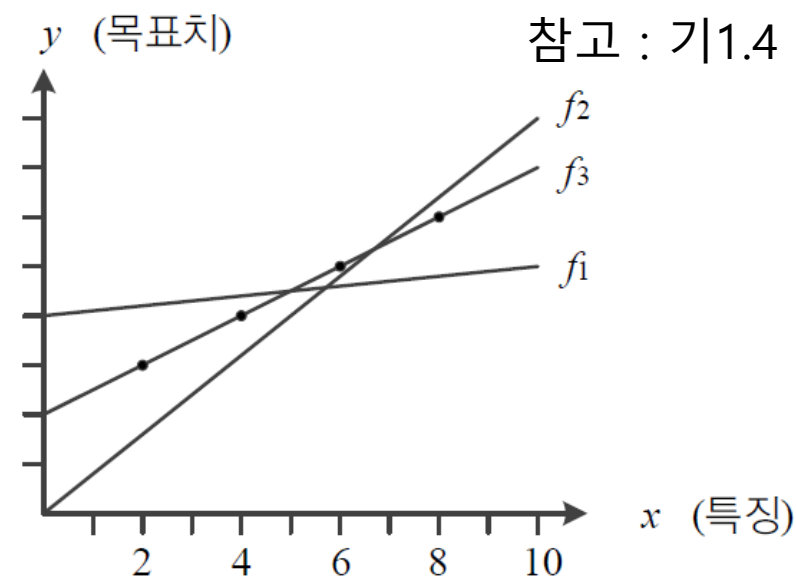


그림 1-4 간단한 기계 학습 예제

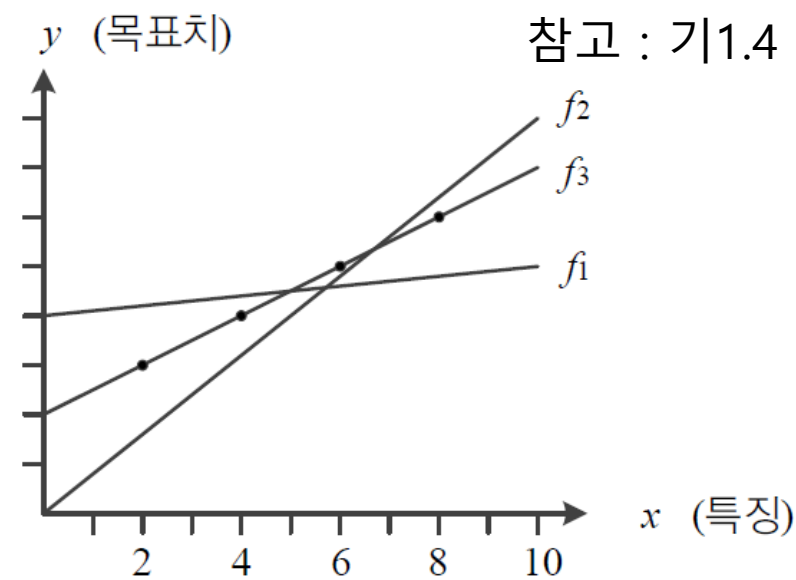


그림 1-4 간단한 기계 학습 예제

[예제 1-1] 훈련집합

- θ_1 을 개선하여 $\theta_2 = (0.8, 0.0)^T$ 가 되었다고 가정

$$\mathbf{x}_1, y_1 \rightarrow (f_{\theta_2}(2.0) - 3.0)^2 = ((0.8 * 2.0 + 0.0) - 3.0)^2 = 1.96$$

$$\mathbf{x}_2, y_2 \rightarrow (f_{\theta_2}(4.0) - 4.0)^2 = ((0.8 * 4.0 + 0.0) - 4.0)^2 = 0.64$$

$$\mathbf{x}_3, y_3 \rightarrow (f_{\theta_2}(6.0) - 5.0)^2 = ((0.8 * 6.0 + 0.0) - 5.0)^2 = 0.04$$

$$\mathbf{x}_4, y_4 \rightarrow (f_{\theta_2}(8.0) - 6.0)^2 = ((0.8 * 8.0 + 0.0) - 6.0)^2 = 0.16$$

$$\longrightarrow J(\theta_2) = 0.7$$

- θ_2 를 개선하여 $\theta_3 = (0.5, 2.0)^T$ 가 되었다고 가정
- 이때 $J(\theta_3) = 0.0$ 이 되어 θ_3 은 **최적값 $\hat{\theta}$** 이 됨

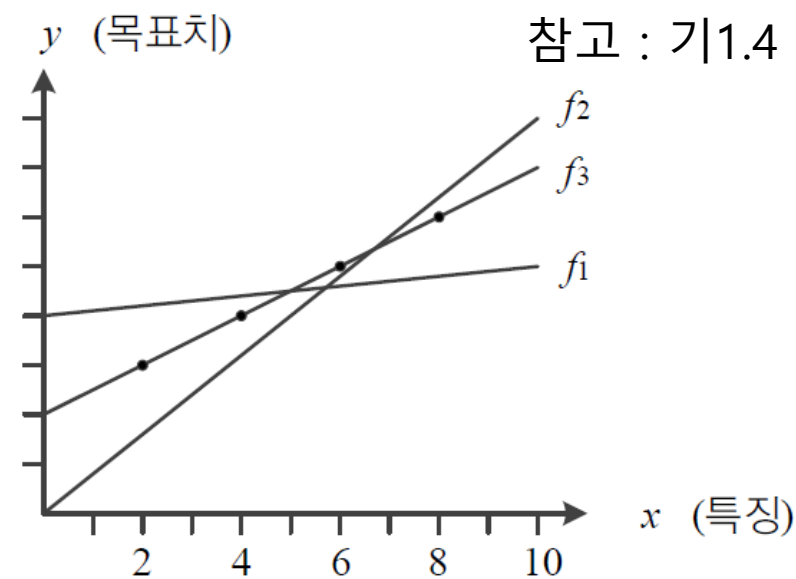
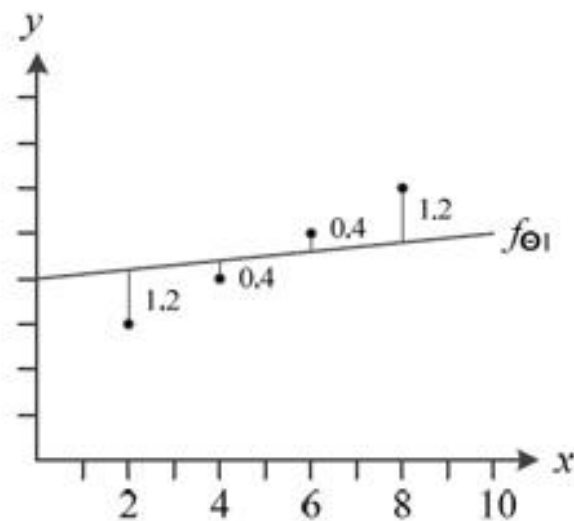
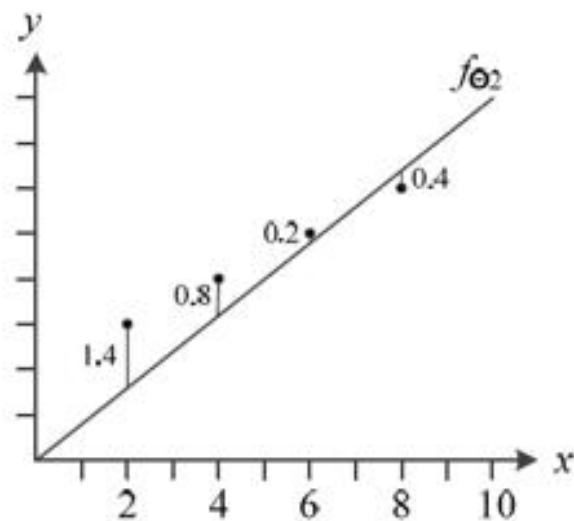


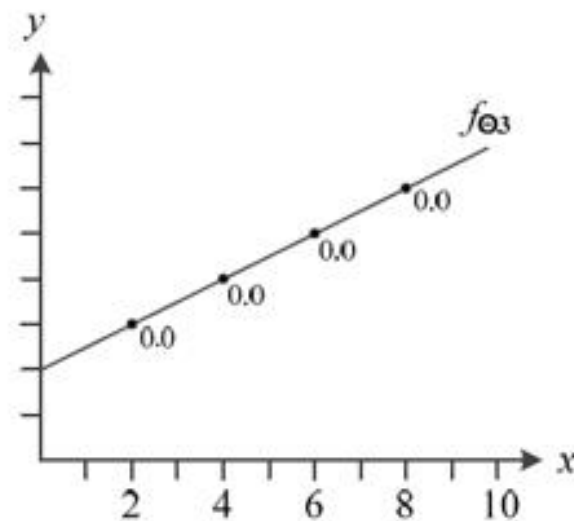
그림 1-4 간단한 기계 학습 예제



(a) 초기 매개변수 θ_1



(b) θ_1 을 개선하여 θ_2 가 됨



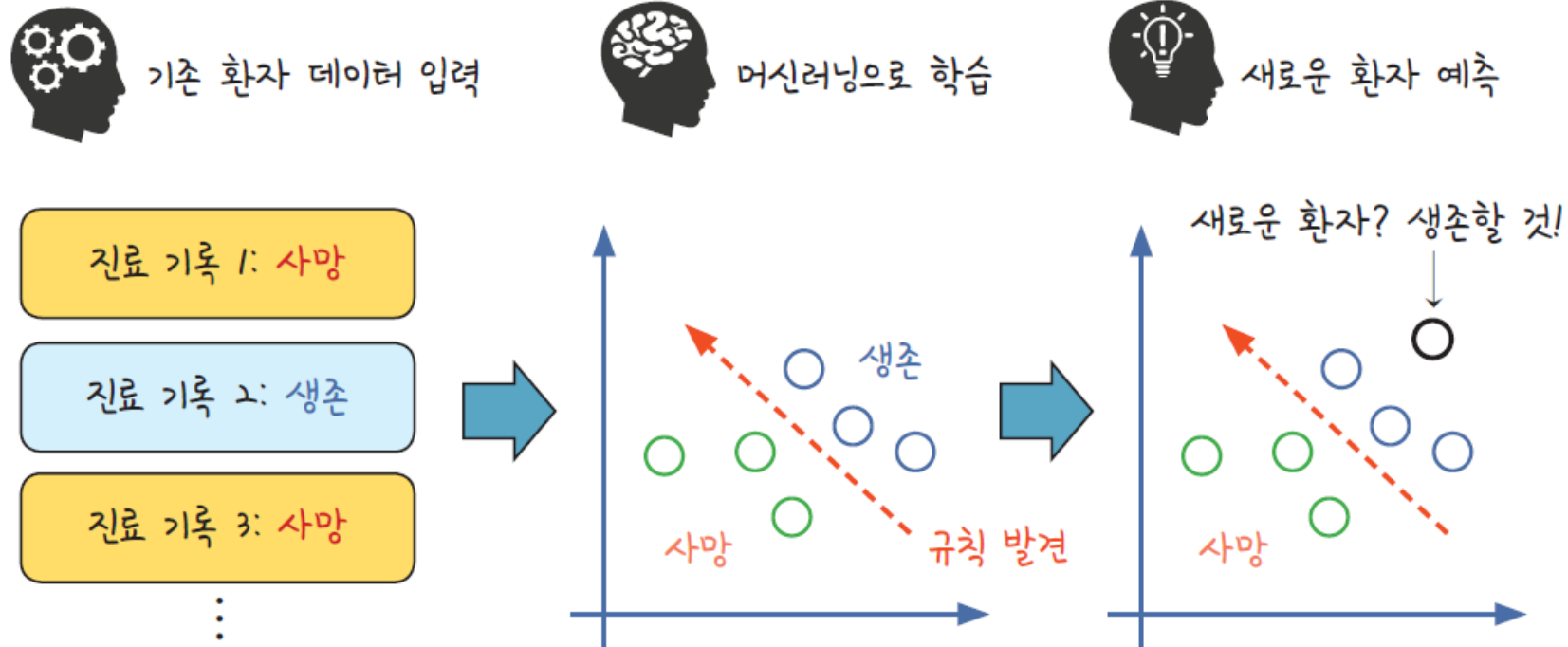
(c) θ_2 를 개선하여 최적의 θ_3 을 찾음

그림 1-11 기계 학습에서 목적함수의 역할

기계학습 예시(코드)

모1

폐암 수술 환자의 생존율 예측하기



폐암 환자 의료 데이터

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	293	1	3.8	2.8	0	0	0	0	0	0	12	0	0	0	1	0	62	0
2	1	2	2.88	2.16	1	0	0	0	1	1	14	0	0	0	1	0	60	0
3	8	2	3.19	2.5	1	0	0	0	1	0	11	0	0	1	1	0	66	1
4	14	2	3.98	3.06	2	0	0	0	1	1	14	0	0	0	1	0	80	1
5	17	2	2.21	1.88	0	0	1	0	0	0	12	0	0	0	1	0	56	0
6	18	2	2.96	1.67	0	0	0	0	0	0	12	0	0	0	1	0	61	0
7	35	2	2.76	2.2	1	0	0	0	1	0	11	0	0	0	0	0	76	0
8	42	2	3.24	2.52	1	0	0	0	1	0	12	0	0	0	1	0	63	1
9	65	2	3.15	2.76	1	0	1	0	1	0	12	0	0	0	1	0	59	0
10	111	2	4.48	4.2	0	0	0	0	0	0	12	0	0	0	1	0	55	0
11	121	2	3.84	2.56	1	0	0	0	1	0	11	0	0	0	0	0	59	0
12	123	2	2.8	2.12	1	0	0	1	1	0	13	0	0	0	1	0	80	0
13	130	2	5.6	4.64	1	0	0	0	1	0	11	0	0	0	1	0	45	0
14	132	2	2.12	1.72	1	0	0	0	0	0	12	0	0	0	1	0	74	0
15	133	2	2.5	71.1	0	0	0	1	0	0	13	0	0	0	1	0	64	1
16	137	2	3.76	3.08	1	0	0	0	1	0	13	0	0	0	1	0	54	0
17	141	2	2.16	1.56	1	0	0	0	1	0	11	0	0	0	1	0	63	0
18	145	2	3.64	2.48	2	0	0	0	1	1	11	0	0	0	1	0	70	0
19	164	2	2.4	1.96	1	0	0	0	1	0	12	0	0	0	0	0	73	0
20	165	2	3	2.4	1	0	0	0	1	0	14	0	0	0	1	0	58	0

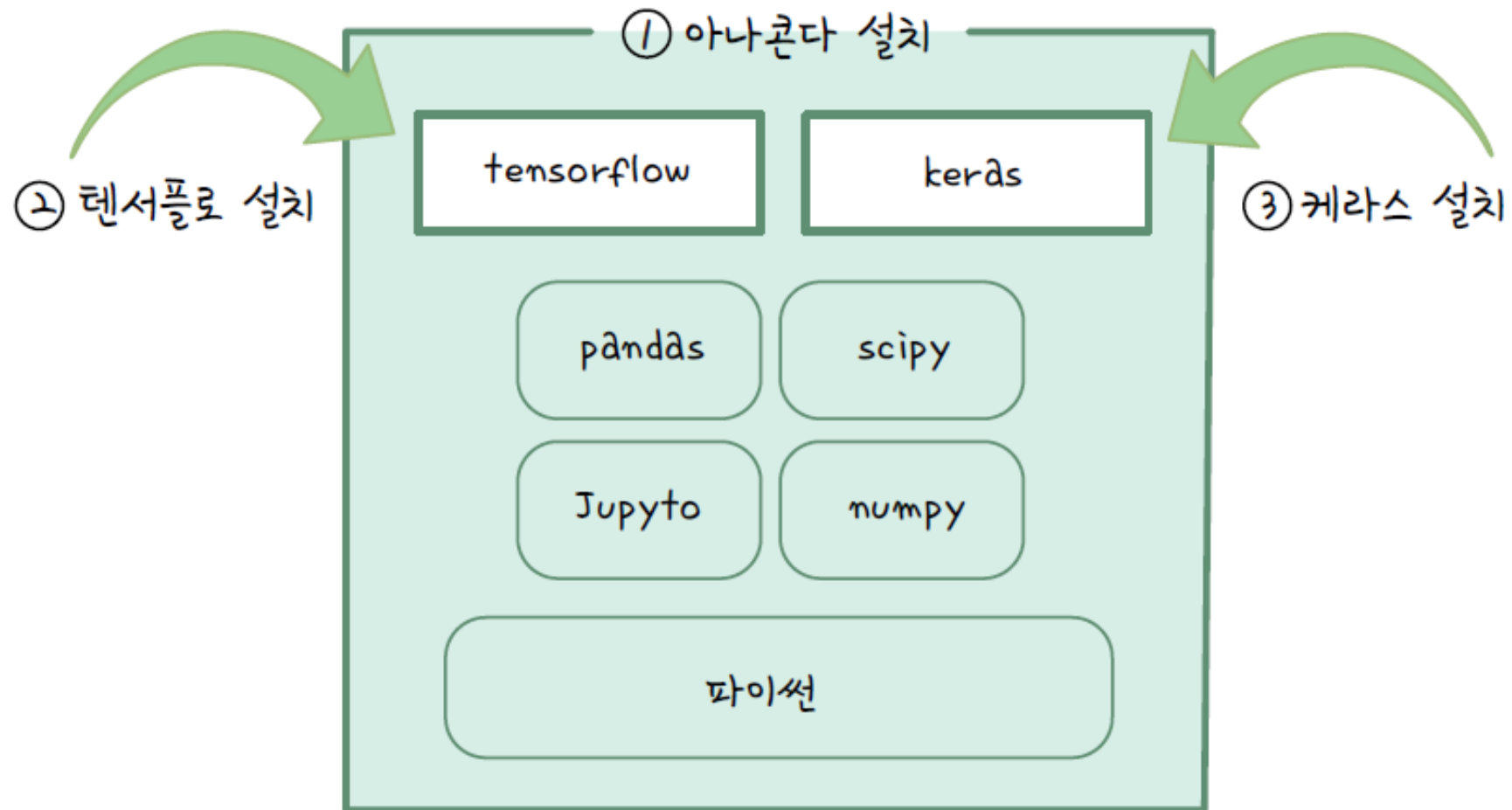
폐암 환자 의료 데이터

줄 항목	속성																	클래스
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	293	1	3.8	2.8	0	0	0	0	0	0	12	0	0	0	1	0	62	0
2	1	2	2.88	2.16	1	0	0	0	1	1	14	0	0	0	1	0	60	0
3	8	2	3.19	2.5	1	0	0	0	1	0	11	0	0	1	1	0	66	1
...
470	447	8	5.2	4.1	0	0	0	0	0	0	12	0	0	0	0	0	49	0

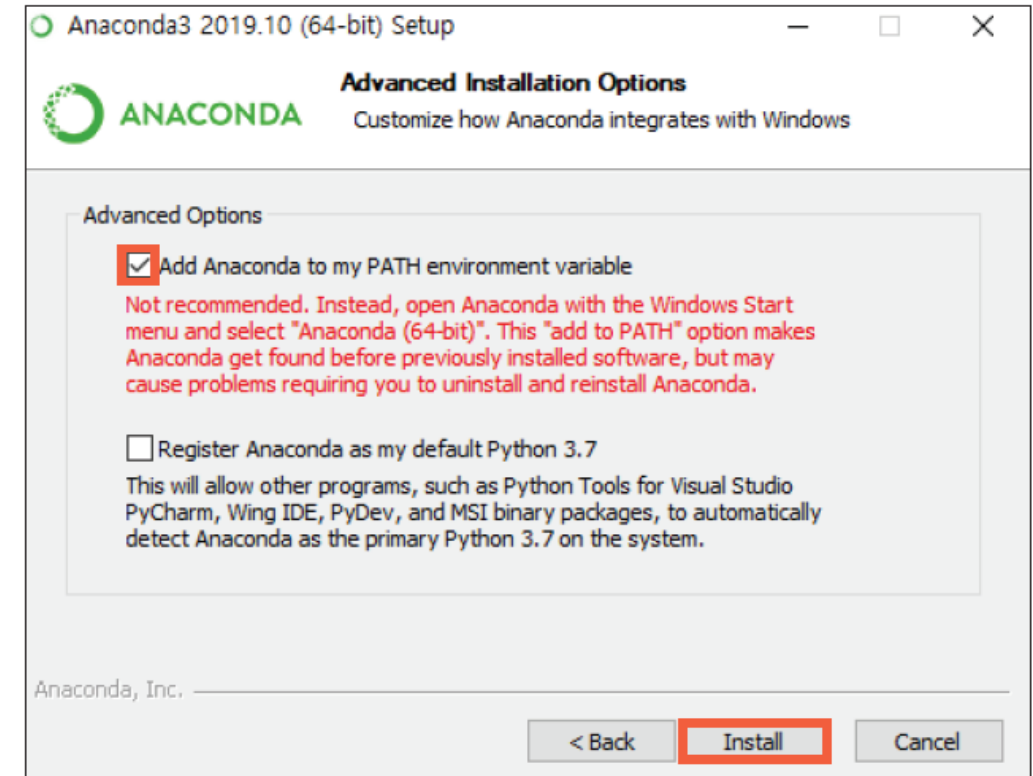
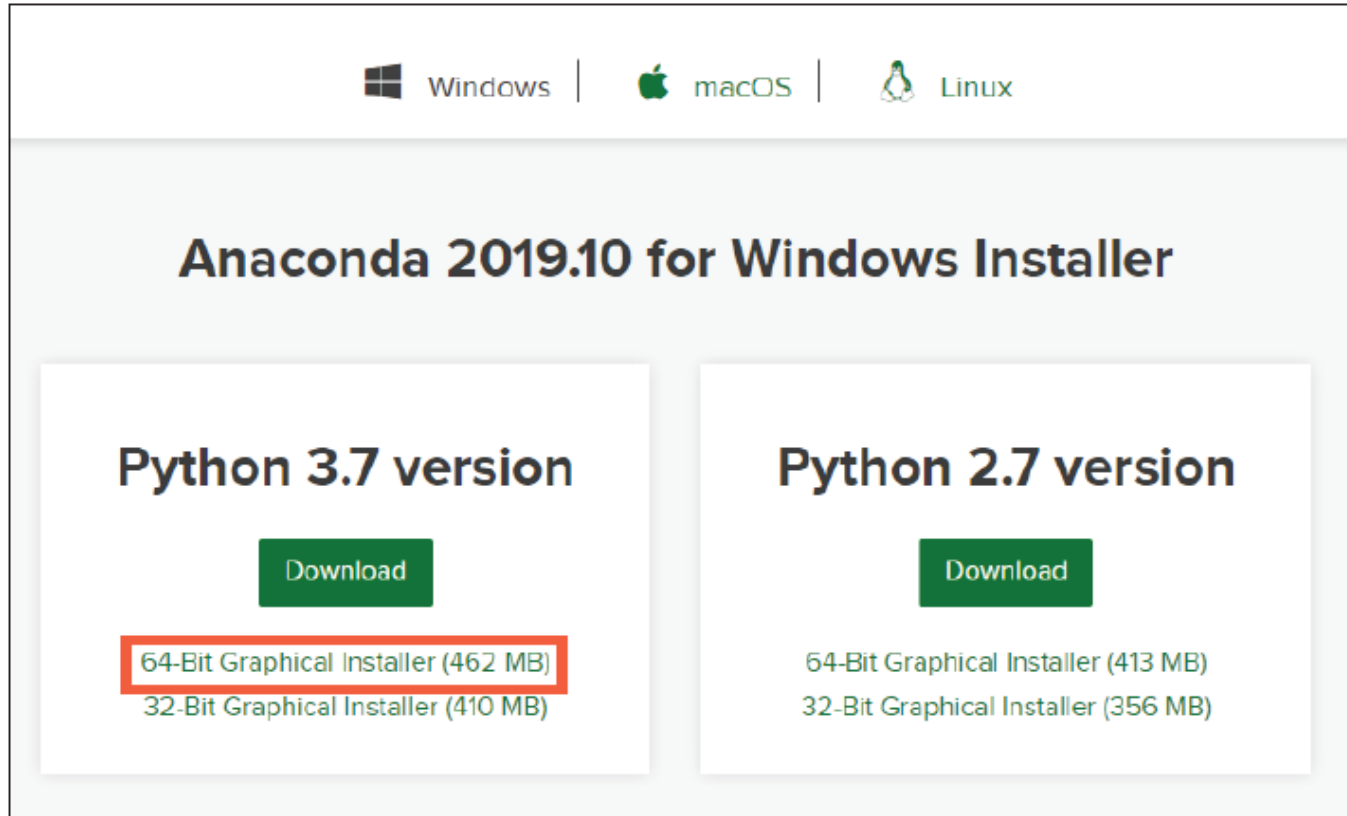
ML/DL 준비사항

- 데이터셋
- 작업환경
 - 본인의 PC/노트북
 - → CPU/GPU 여부에 따라 설치과정이나 성능 차이가 클 수 있음
 - 또는 Google Colab

작업환경 만들기

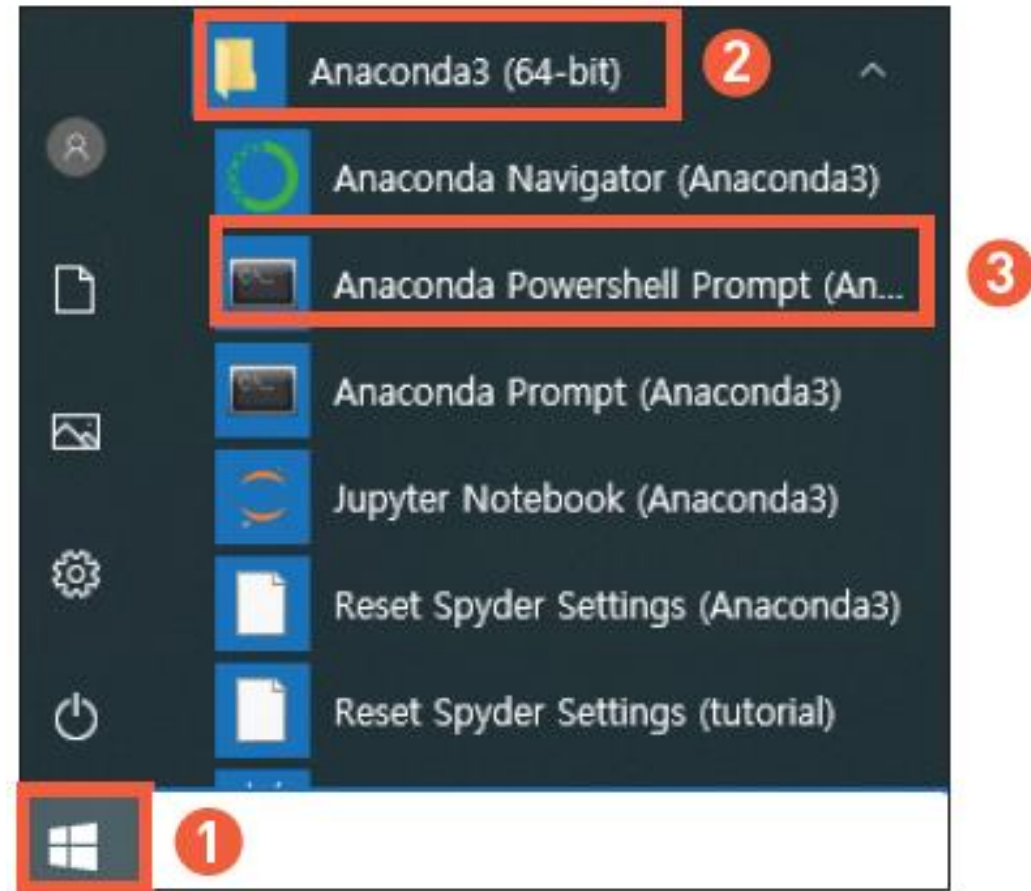


작업환경 만들기



Windows 10 기준

작업환경 만들기



Windows 10 기준

작업환경 만들기

Anaconda Prompt (Anaconda3)

```
(base) C:\Users\it>pip install tensorflow
```

- (base) > conda create -n iml
- (base) > conda activate iml
 - ➔ Anaconda 가상환경이 만들어짐
 - ➔ iml 외에 본인 편한 환경이름으로 바꾸어도 무방함
- (iml) > conda install pip
- (iml) > pip install **tensorflow**
- (iml) > conda install **matplotlib pandas scikit-learn numpy -y**
- (iml) > conda install **ipython notebook -y**

Windows 10 기준

작업환경 만들기

- 설치 확인 필수!

```
Anaconda Prompt (Anaconda3) - python

(base) C:\Users\it>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> print(tf.__version__)
2.0.0
>>> import keras
Using TensorFlow backend.
>>>
```

- Keras를 별도로 설치하지 않은 경우, keras 확인이 다름

```
>>> import tensorflow.keras as keras
>>> print(keras.__version__)
2.4.0
>>>
```

코드/데이터 받기

- <https://github.com/gilbutITbook/080228>
 - 다운 : [deeplearning.zip](#)

gilbutITbook / 080228

Watch 3 Star 20 Fork 38

Code Issues 7 Pull requests 2 Actions Projects Wiki Security Insights

master 080228 / deeplearning / run_project / Go to file Add file

gilbutITbook rmv on 5 Jun History

..		
py	seventh release	3 months ago
test	initial release	7 months ago
train	initial release	7 months ago
01_My_First_DeepLearning.ipynb	sixth update	3 months ago

jupyter 실행

여기에 "[deeplearning.zip](#)" 압축 풀어놓기



- (iml)

```
> jupyter notebook --notebook-dir="C:/Your/Desired/Start/Directory/"
```

jupyter 실행

deep learning / run_project

localhost:8888/tree/deeplearning/run_project

jupyter

Quit Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ↻

<input type="checkbox"/> 0	/ deeplearning / run_project	Name ↓	Last Modified	File size
<input type="checkbox"/>	..		몇 초 전	
<input type="checkbox"/>	py		10분 전	
<input type="checkbox"/>	test		10분 전	
<input type="checkbox"/>	train		10분 전	
<input type="checkbox"/>	01_My_First_Deep learning.ipynb	Running	3분 전	15.2 kB
<input type="checkbox"/>	02_Pima_Indian.ipynb		10분 전	127 kB
<input type="checkbox"/>	03_Iris_Multi_Classfication.ipynb		10분 전	154 kB

deeplearning/run_project/ x 01_My_First_Deeplearning - Jupy x +

localhost:8888/notebooks/deeplearning/run_project/01_My_First_Dee... ☆ ☆ ☆

jupyter 01_My_First_Deeplearning (autosaved) Python 3

File Edit View Insert Cell Kernel Help Not Trusted

Run Cells (Ctrl-Enter)
Run Cells and Insert Below (Alt-Enter)
Run All
Run All Above
Run All Below
Cell Type
Current Outputs
All Output

```
2 # 코드는  
3  
4 # 데이터 불러오기  
5 from tensorflow.keras.preprocessing.text import Tokenizer  
6 from tensorflow.keras.preprocessing.sequence import pad_sequences  
7  
8 # 필요 라이브러리 불러오기  
9 import numpy as np  
10 import tensorflow as tf  
11  
12 # 실행할 때마다 같은 결과를 출력하기 위해 설정하는 부분입니다.  
13 np.random.seed(3)  
14 tf.random.set_seed(3)  
15  
16 # 준비된 수술 환자 데이터를 불러들입니다.  
17 Data_set = np.loadtxt("../dataset/ThoracicSurgery.csv", delimiter=",")  
18  
19 # 환자의 기록과 수술 결과를 X와 Y로 구분하여 저장합니다.  
20 X = Data_set[:, 0:17]  
21 Y = Data_set[:, 17]  
22  
23 # 딥러닝 구조를 결정합니다(모델을 설정하고 실행하는 부분입니다).  
24 model = Sequential()
```

jupyter 실행

- 다음 셀을 추가로 삽입
- 다음 코드를 추가로 입력한 후
- 해당 셀을 실행해보세요

```
In [4]: 1 score = model.evaluate(X,Y, verbose=0)
        2 print('Test loss:', score[0])
        3 print('Test accuracy:', score[1])
```

```
Test loss: 0.3724135458469391
Test accuracy: 0.8553191423416138
```

코드 리뷰

In [1]:

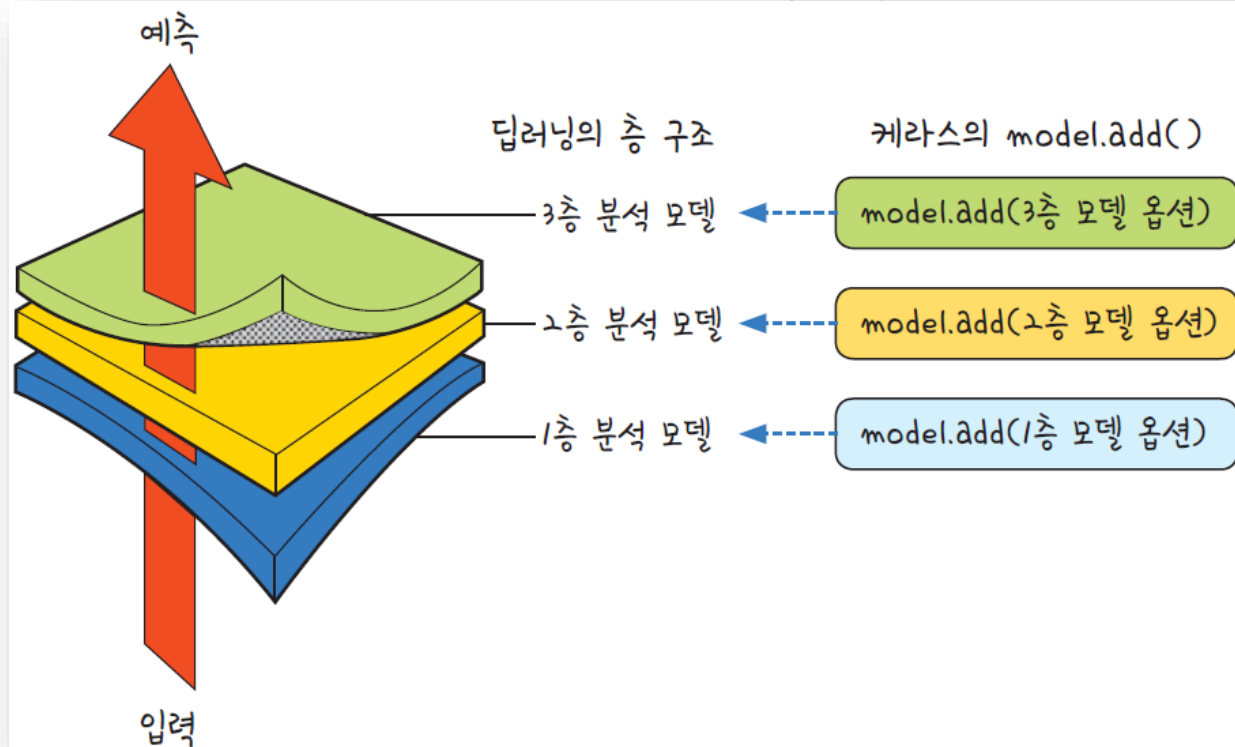
```
1  # -*- coding: utf-8 -*-
2  # 코드 내부에 한글을 사용가능 하게 해주는 부분입니다.
3
4  # 딥러닝을 구동하는 데 필요한 케라스 함수를 불러옵니다.
5  from tensorflow.keras.models import Sequential
6  from tensorflow.keras.layers import Dense
7
8  # 필요한 라이브러리를 불러옵니다.
9  import numpy as np
10 import tensorflow as tf
11
12 # 실행할 때마다 같은 결과를 출력하기 위해 설정하는 부분입니다.
13 np.random.seed(3)
14 tf.random.set_seed(3)
15
16 # 준비된 수술 환자 데이터를 불러들입니다.
17 Data_set = np.loadtxt("../dataset/ThoracicSurgery.csv", delimiter=",")
18
19 # 환자의 기록과 수술 결과를 X와 Y로 구분하여 저장합니다.
20 X = Data_set[:,0:17]
21 Y = Data_set[:,17]
22
23 # 딥러닝 구조를 결정합니다(모델을 설정하고 실행하는 부분입니다).
24 model = Sequential()
25 model.add(Dense(30, input_dim=17, activation='relu'))
26 model.add(Dense(1, activation='sigmoid'))
27
28 # 딥러닝을 실행합니다.
29 model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
30 model.fit(X, Y, epochs=100, batch_size=10)
```

ML:

코드 리뷰

In [1]:

```
1 # -*- coding: utf-8 -*-
2 # 코드 내부에 한글을 사용가능 하게 해주는 부분입니다.
3
4 # 딥러닝을 구동하는 데 필요한 케라스 함수를 불러옵니다.
5 from tensorflow.keras.models import Sequential
```



```
as.layers import Dense
```

를 불러옵니다.

```
tf
```

결과를 출력하기 위해 설정하는 부분입니다.

```
)
```

데이터를 불러들입니다.

```
ct("../dataset/ThoracicSurgery.csv", delimiter=",")
```

결과를 X와 Y로 구분하여 저장합니다.

ML:

```
23 # 딥러닝 구조를 결정합니다(모델을 설정하고 실행하는 부분입니다).
```

```
24 model = Sequential()
```

```
25 model.add(Dense(30, input_dim=17, activation='relu'))
```

```
26 model.add(Dense(1, activation='sigmoid'))
```

```
27
```

```
28 # 딥러닝을 실행합니다.
```

```
29 model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
30 model.fit(X, Y, epochs=100, batch_size=10)
```

```
31
```

코드 리뷰

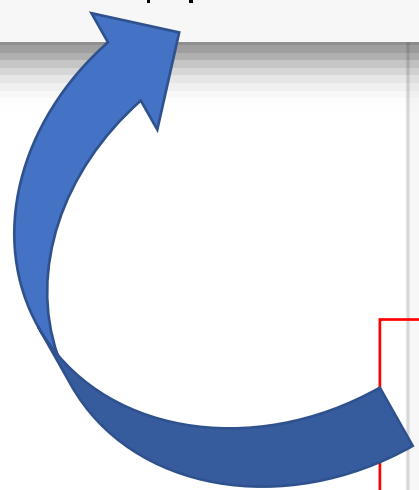
In [1]:

```
1 # -*- coding: utf-8 -*-
2 # 코드 내부에 한글을 사용가능 하게 해주는 부분입니다.
3
4 # 딥러닝을 구동하는 데 필요한 케라스 함수를 불러옵니다.
5 from tensorflow.keras.models import Sequential
6 from tensorflow.keras.layers import Dense
7
```

```
23 # 딥러닝 구조를 결정합니다(모델을 설정하고 실행하는 부분입니다).
24 model = Sequential()
25 model.add(Dense(60, input_dim=17, activation='relu'))
26 model.add(Dense(60, activation='relu'))
27 model.add(Dense(60, activation='relu'))
28 model.add(Dense(1, activation='sigmoid'))
29
```

입니다.

DL:



```
17 Data_set = np.loadtxt("../dataset/ThoracicSurgery.csv", delimiter=",")
18
19 # 환자의 기록과 수술 결과를 X와 Y로 구분하여 저장합니다.
20 X = Data_set[:,0:17]
21 Y = Data_set[:,17]
```

```
23 # 딥러닝 구조를 결정합니다(모델을 설정하고 실행하는 부분입니다).
24 model = Sequential()
25 model.add(Dense(30, input_dim=17, activation='relu'))
26 model.add(Dense(1, activation='sigmoid'))
```

```
27
28 # 딥러닝을 실행합니다.
29 model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
30 model.fit(X, Y, epochs=100, batch_size=10)
```

Summary

- 지도학습(Supervised learning)
- 비지도학습(Unsupervised learning)
- 준지도학습(Semi-supervised learning)
- 강화학습(Reinforcement learning)

- Cost function
 - MSE
- 간단한 Keras 코드 리뷰

In the next lecture...

- 기계학습 이해를 위한 기본 수학 개념
- 모델 훈련 관련 주요 용어
- 퀴즈 #1 (다소 높은 배점)
 - 범위 : 1~2주 개념

참고자료

- 핸1
- 기1
- 모1
- [online] “Introduction to Deep Learning,” MIT, available at <http://introtodeeplearning.com/>
- Github
 - <https://github.com/gilbutITbook/080228>

기 : 기계학습, 오일석, 2017

핸 : 핸즈온머신러닝, 2/E, 2020 (번역)

모 : 모두의 딥러닝, 2/E, 2020

케 : 케라스 창시자에게 배우는..., 2018 (번역)

머 : 머신러닝 도감 그림으로..., 2019 (번역)

파 : Python machine learning, 2/E, 2019 (번역) → “머신러닝 교과서 with 파이썬, ...” 2019