



머신러닝과 사업기획

과정 소개 1.

- 데이터가 많이 쌓이고, 기업의 선제적 의사결정 지원을 위해 Data-Driven Decision이 중요한 화두가 되고 있다.
- 질 좋은, 많은 데이터를 잘 활용하는 것이 중요하다.
- Data Exploration 을 비롯, Machine Learning 기본적인 개념 및 실습을 통해, Internal/External Insight 를 발견하고, 프로세스의 효율적 개선과 기업 경쟁력 향상을 수행하기 위한 역량 향상을 목표로 한다.
- Python을 활용한 기초적인 데이터 분석을 위한 이론과 실습, 그리고 실제 Business에 적용을 위한 모델링을 해 본다.
- Python3, Pandas, Scikit-learn 등을 다룰 예정입니다.

** Python을 처음 다루시는 분들, Computer Science 전공이 아닌 분들도 들을 수 있습니다.

** Python에 대한 허들이 있을 수 있습니다.

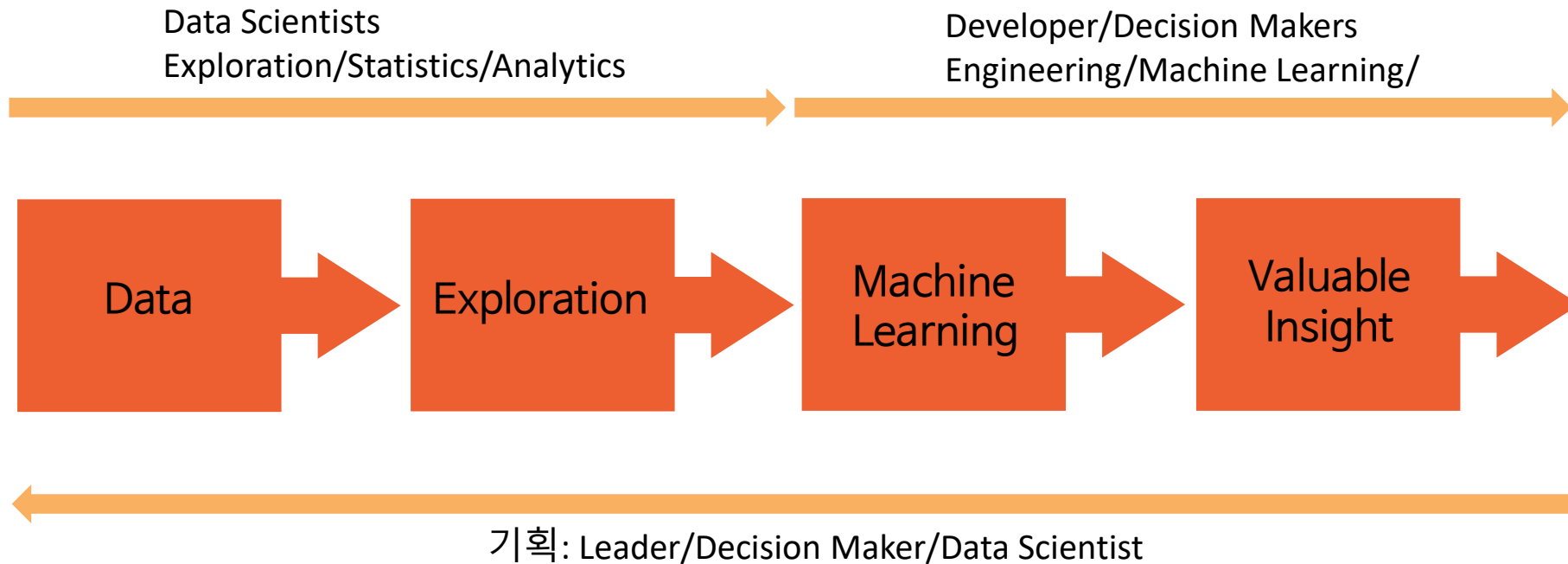
과정 소개 2.

평가기준

1. Term Projects : 50% (Peer-Review)
2. Mid Term : 30%
3. 출석/참여도: 20%

Data Analytics(Machine Learning) Value

- 데이터가 쌓여지고, Data Scientists의 탐색(Exploration) 및 분석(Analysis)를 통해, 데이터를 파악하고,
- 준비된 데이터를 기반으로 Machine Learning 알고리즘을 통해,
- Valuable Insight 로 바뀌어 진다.
- 많은 양의 데이터가 Business Value가 되기 위해서는 데이터는 통해 Insight를 찾아야 된다.



이 강의는 누구에게 필요할까?



Data → Insight → Value



Analysis

- 데이터마이닝
- 머신러닝
- 딥러닝

Action

- 의사결정자
- 엔지니어
- 마케터
- 투자자
- 인사관리자 등

Data → Insight → Value & & Data ← Insight ← Value

- 만약, 의사 결정자가, “Data → Insight → Value” 실행 프로세스를 잘 이해하고 있다면??

Machine Learning
이해하기/실행하기



문제해결



기획

핵심개념, 적용사례 및 방법론을 이론과
실습을 통해 습득
→ 데이터를 보고, 문제를 찾는다.

- 주어진 비즈니스 문제 도출 및 정의
- 데이터 기반 분석으로 문제 해결
- 미래 사업 설계

문제를 기반으로 한 데이터 수집 및 기획
→ 문제를 정의하고,
→ 데이터를 분석하여 해결한다.

Python, Scikit-learn, Tensorflow, Pytorch

- 현업에서 Machine Learning/Big Data Analytics로 해결할 수 있는 문제 도출
- Business Value 및 예상 Insight 도출
- 창의적 사고 방법론, 관련 데이터 인식 및 시각화를 위한 검토

예제로 보기

Machine Learning
이해하기/실행하기



문제해결



기획

핵심개념, 적용사례 및 방법론을 이론과
실습을 통해 습득
→ 데이터를 보고, 문제를 찾는다.

주어진 비즈니스 문제 정의
데이터 기반 분석으로 문제 해결
미래사업 설계(인사이트 도출)

문제를 기반으로 한 데이터 수집 및 기획
→문제를 정의하고, 데이터를 분석하여 해결한다.

1. 집에 재료가 많다.
2. 재료를 활용해서 파티를 해 볼까?
3. 요리할 사람은 있을까?
4. 장소는 어디서 할까?



디너 파티




1. 디너파티를 기획하자.
2. 파티의 목적을 정하자.
3. 필요한 리스트(참석자, 재료 등)를 만들자

데이터를 보고, 기획을 한다.
이전 경험(데이터)에 기반해서 준비한다.

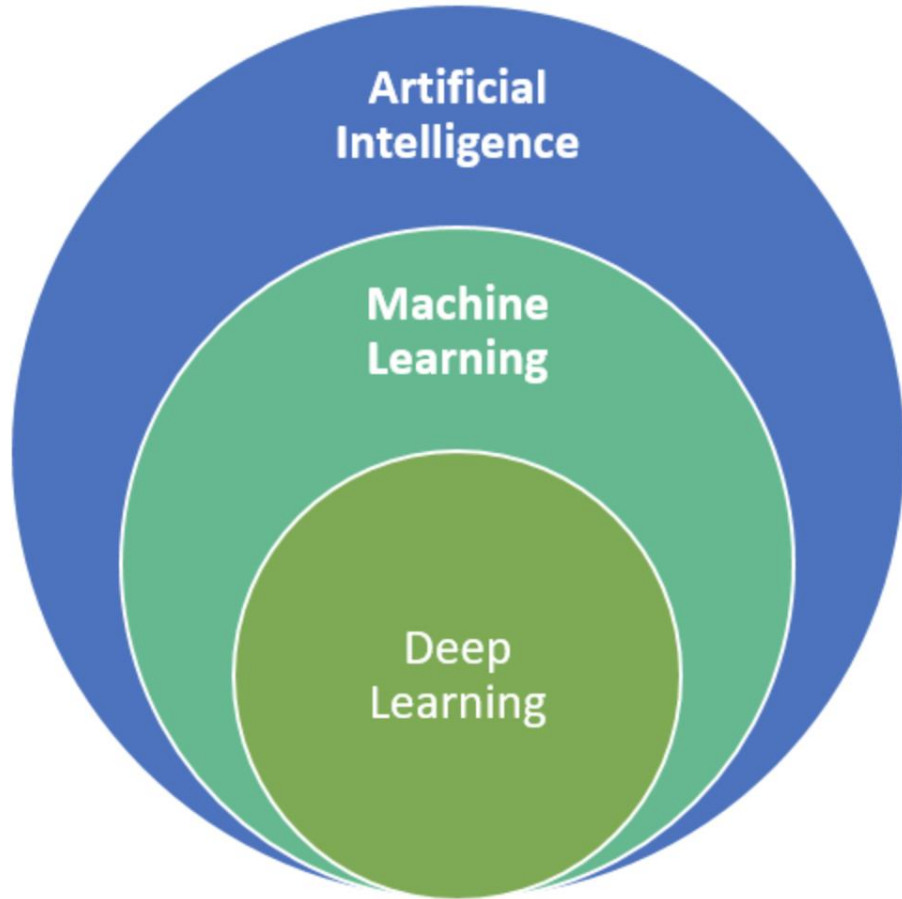


기획 없이 디너파티를 진행하면, 파티 내내
우왕좌왕하고, 즉흥으로, 퀄리티 떨어지는 음식 및
데코레이션을 할 수 밖에 없다. →
임기응변밖에 되지 않는다. 그러나 계획/기획하면,
퀄리티를 높이고 비용도 줄일 수 있다.



AI (Artificial Intelligence) /
ML (Machine Learning) /
DL (Deep Learning)

AI〉 Machine Learning 〉 Deep Learning



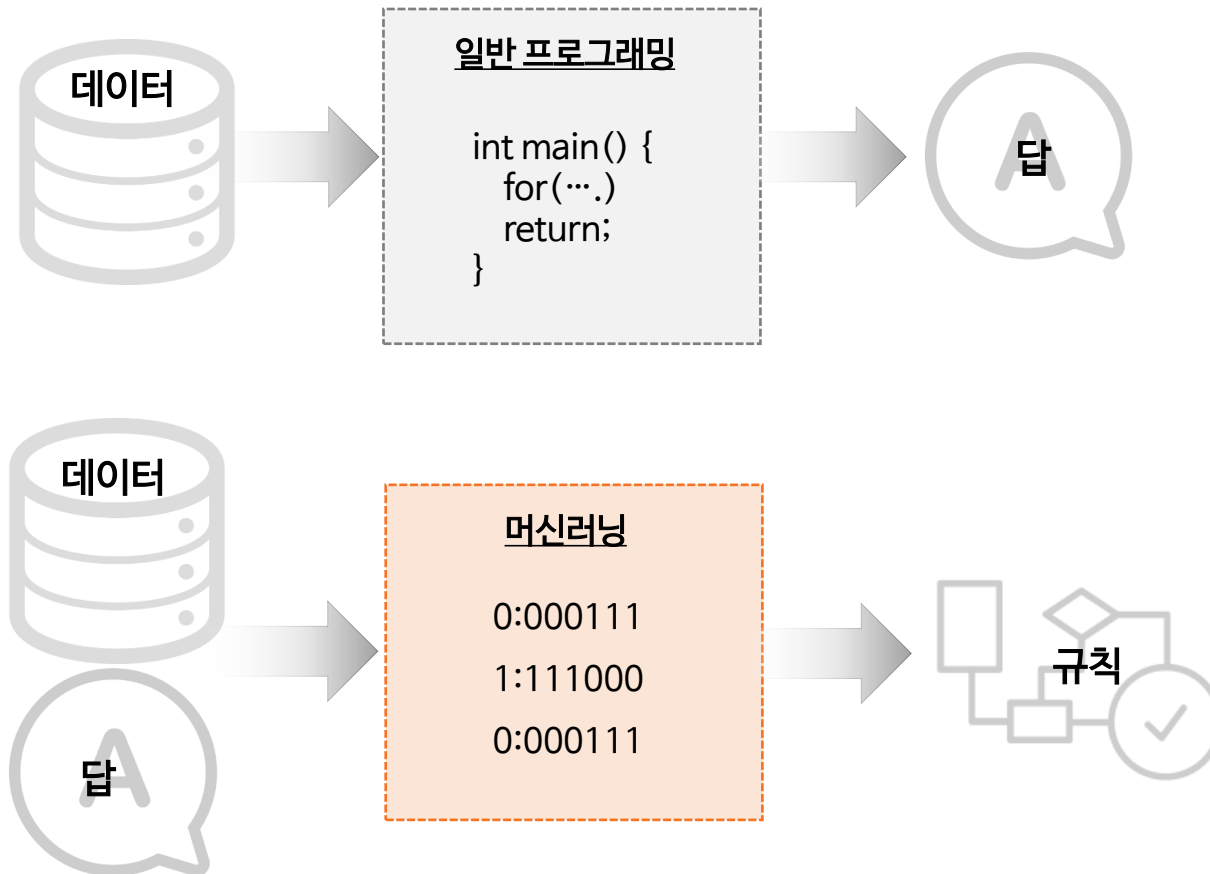
- AI: 가장 큰 범주
 - 큰 범주 안에, 머신러닝이 속하고,
 - 머신러닝의 일부분이 딥러닝
- 사람과 유사한 판단을 컴퓨터가 해 낼 수 있게끔 만드는 것을 인공지능
 - 기존의 데이터를 이용해 앞으로의 일을 예측하는 머신러닝
 - 머신러닝 안에 여러 알고리즘이 있는데,
그 중 한 개가 딥러닝
- 인공지능이 먹을 수 있는 모든 음식이라면,
 - 머신러닝은 그 중 영양가 많은 고기 음식
 - 딥러닝은 그 중에서도 최고급 스테이크 요리

Machine Learning vs. Programming

전통적인 SW(프로그래밍)은 한번만 만들면 된다.

ML(Machine Learning)은 전통적인 SW 개발과 달리, 데이터 학습을 통해, 더 좋은 규칙을 계속 만들어 낸다.

하드웨어 성능이 좋을 수록, 계속 반복 할 수록, 데이터가 많아질 수록 성능도 함께 발전한다.

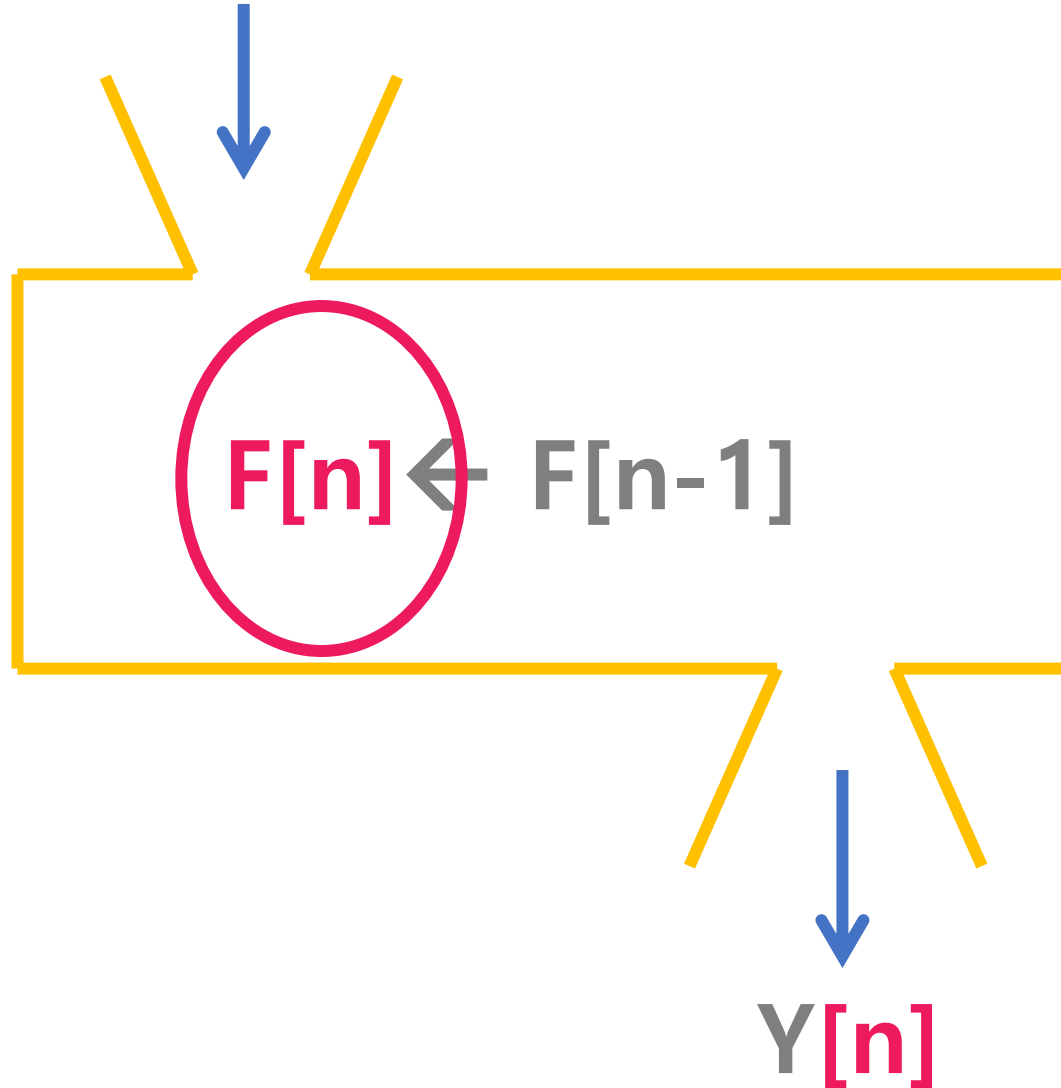




Machine Learning

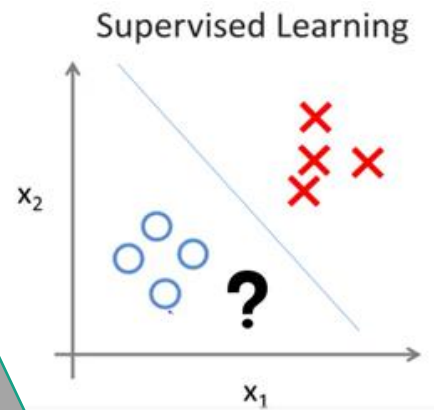
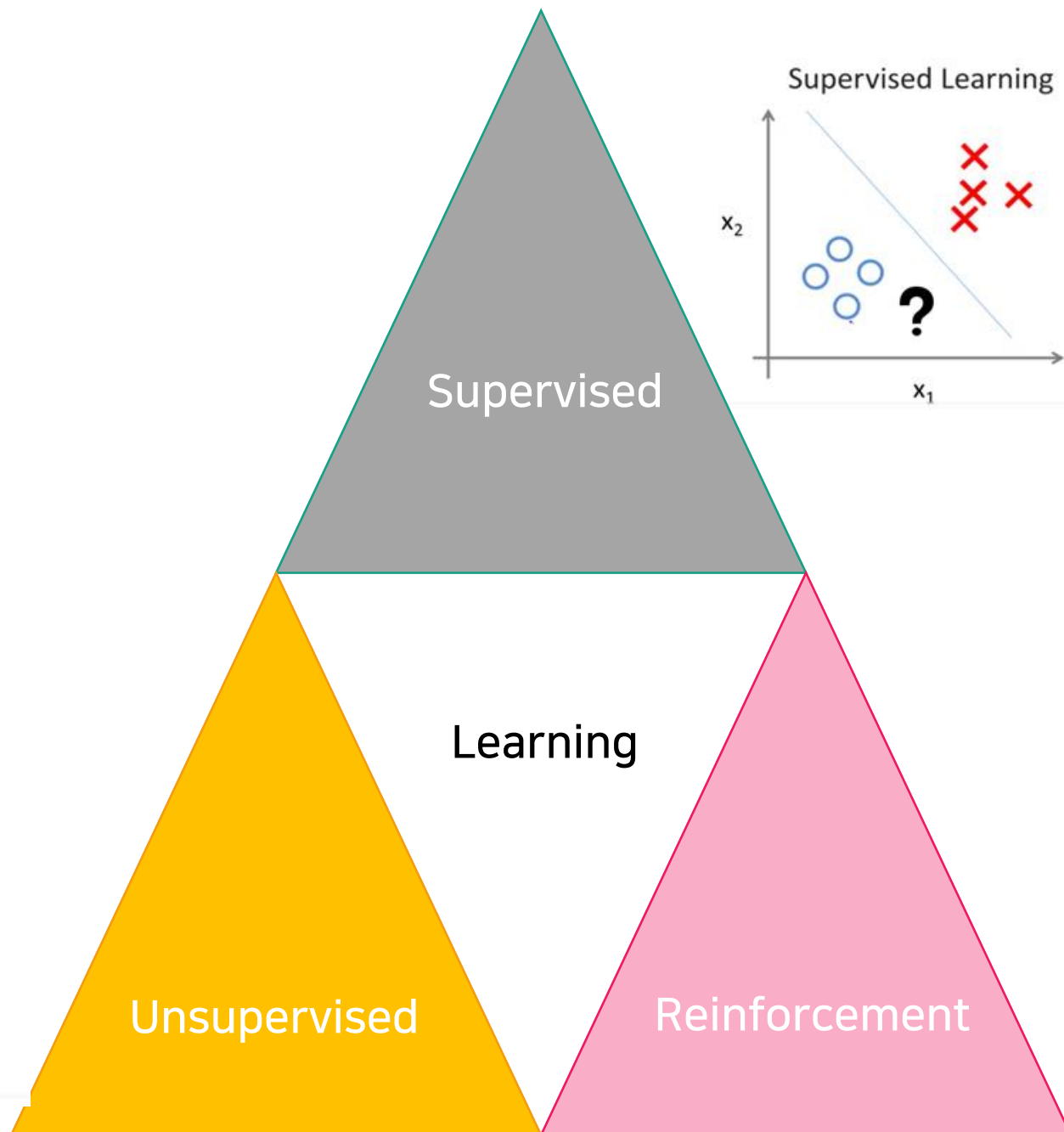
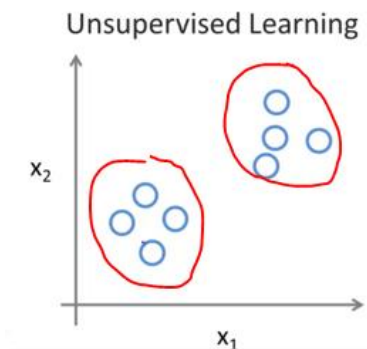
$X[n]$

n : data index

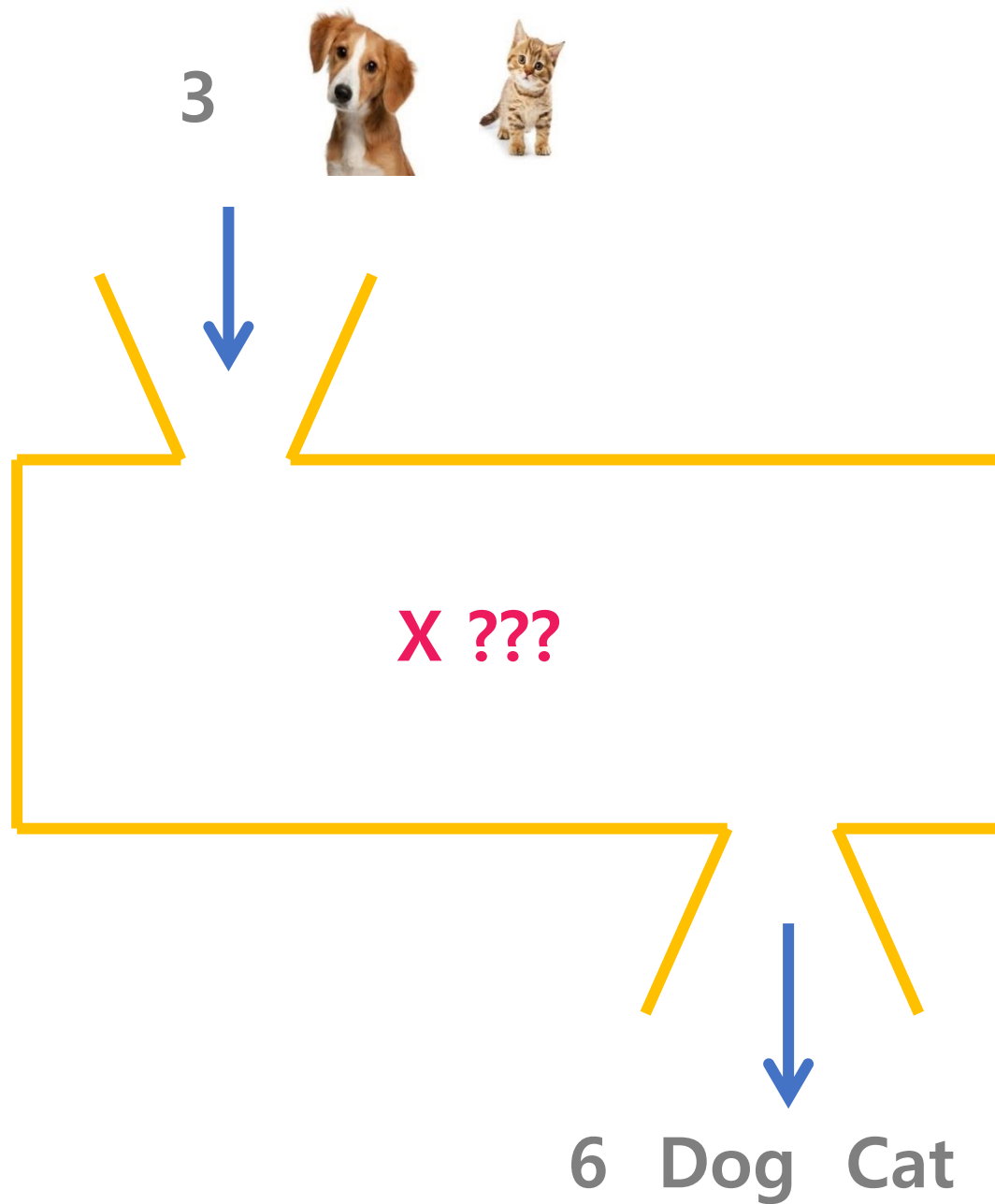


Data가 늘어날수록
알고리즘이 학습(Learning)을 반복한다.
더 똑똑해 진다.

머신러닝의 세가지 타입



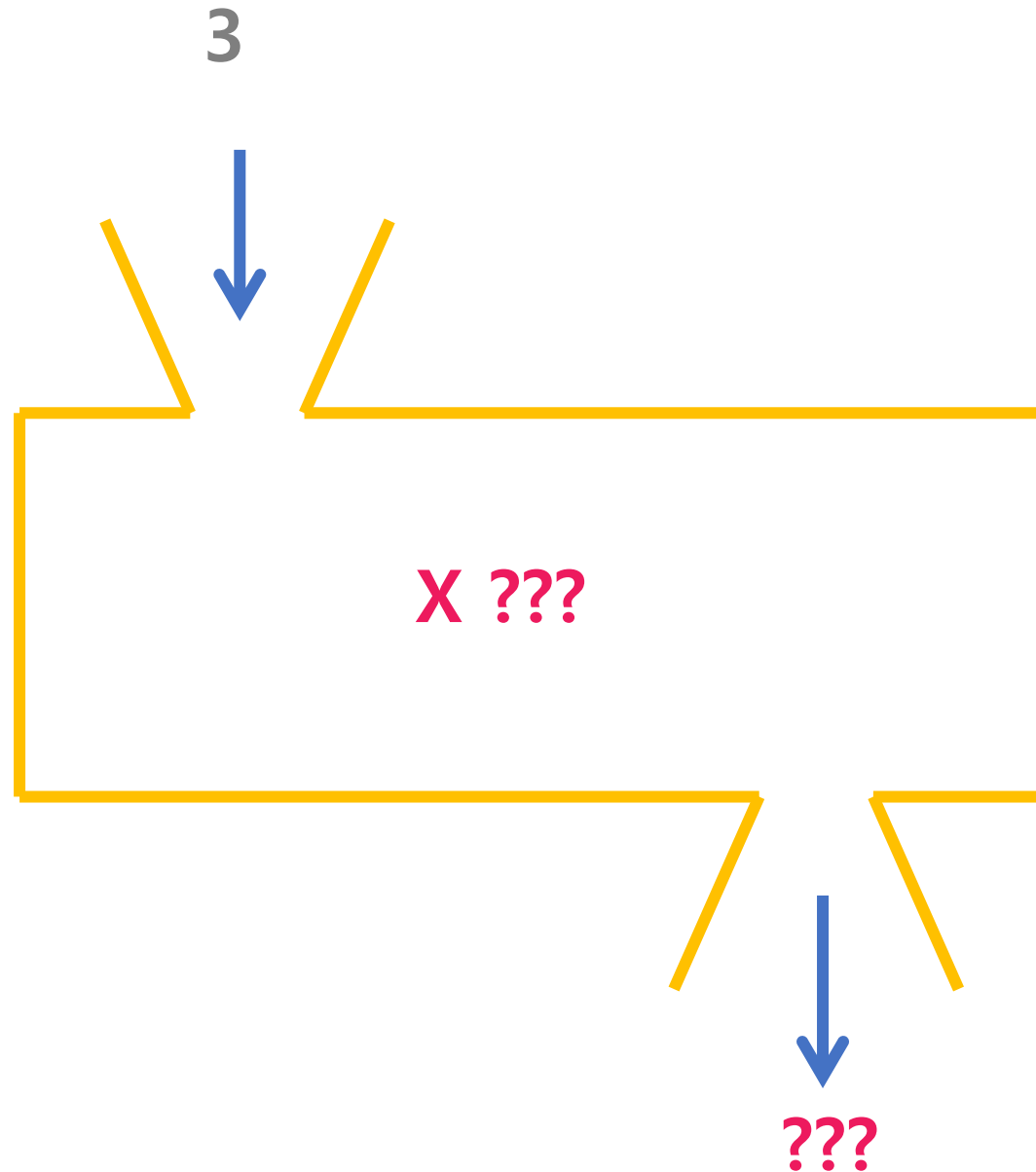
Supervised Learning




- 정답이 주어진다.
- (비교적)문제풀이가 쉽다

UnSupervised Learning

- 미지수 2개, 방정식 1개




$$xy-x-y+1=0,$$

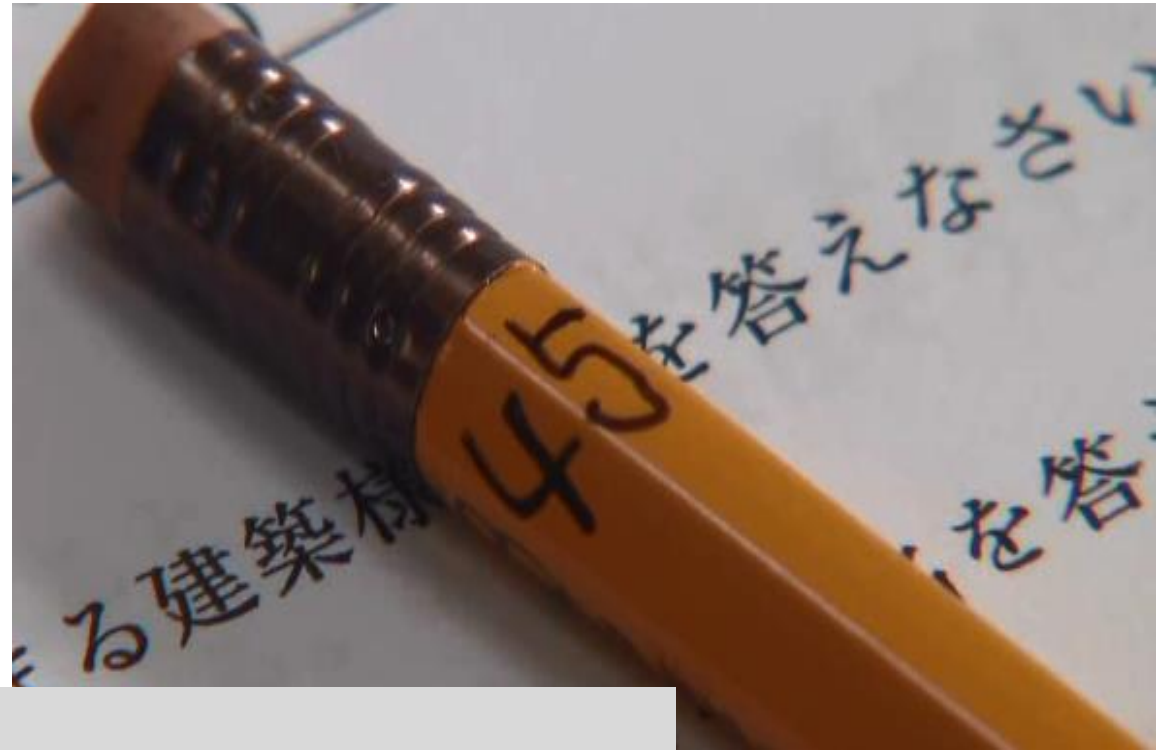
x 와 y 를 구하라.

이것은 풀 수 있나요?

(x, y 는 자연수)

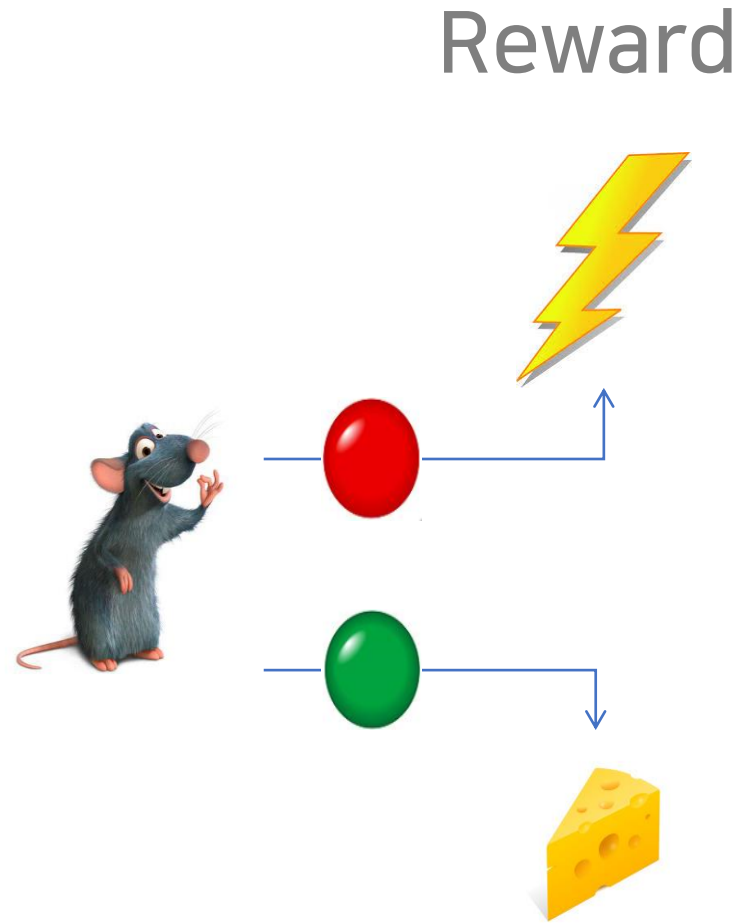
좋은 조건이 주어지거나 잘 찍는 수 밖에...

- 특정 조건이 있을 때만
정답이 주어질 수 있다.
- 기본적으로 문제풀이가 어렵다.
- 정답이 여러 개 일 수 있다.



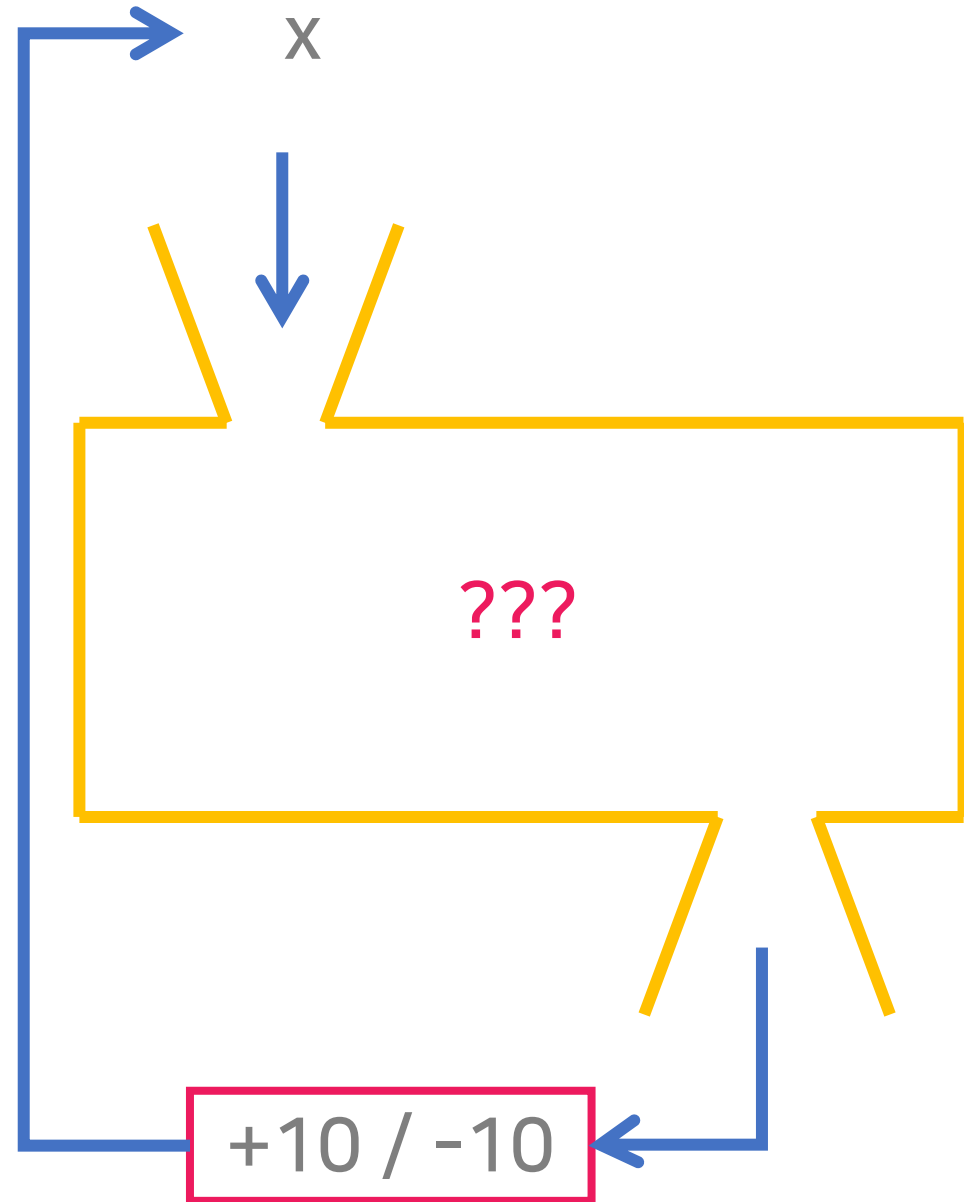
Reinforcement Learning

- 정답이 아닌 REWARD(보상)을 준다.
- 보상을 기반으로 학습한다.



Reinforcement Learning

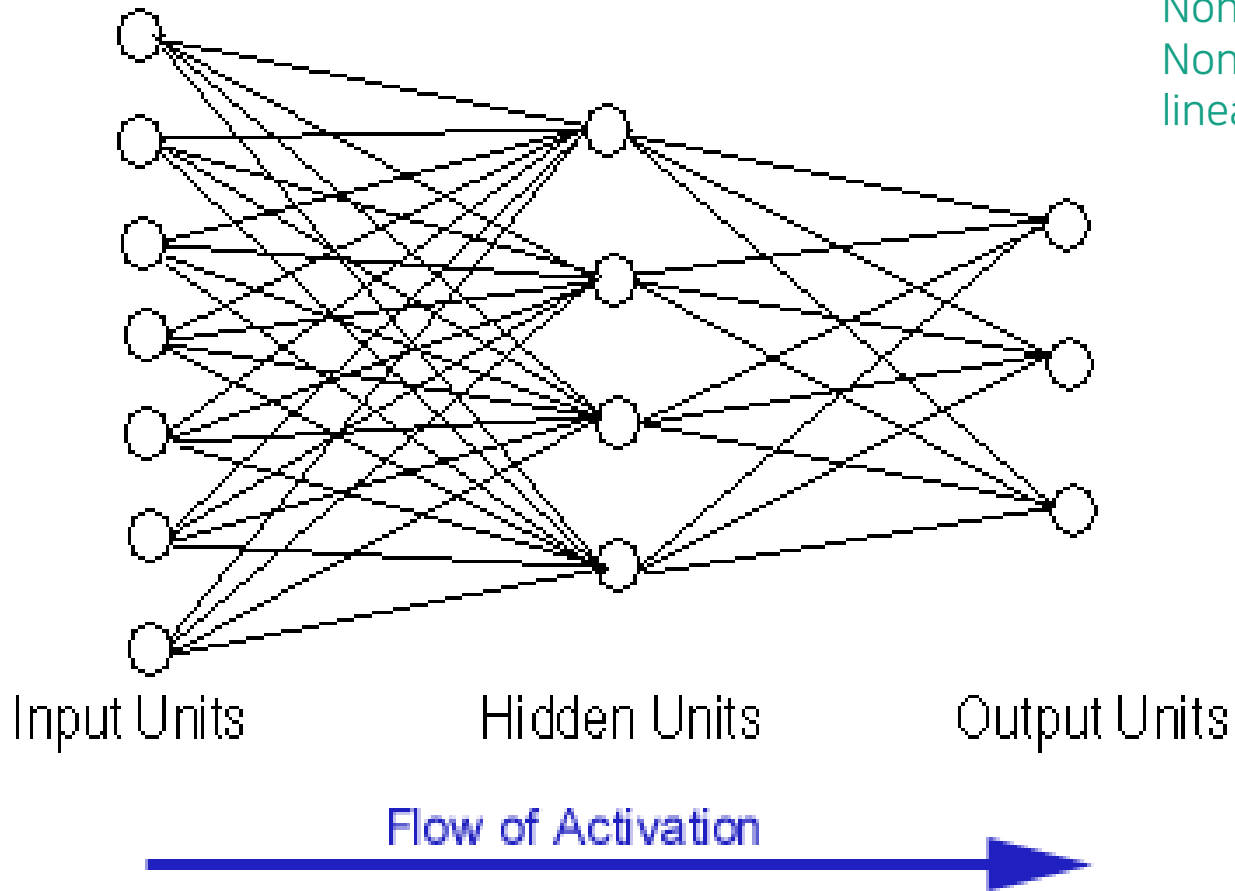
- 정답이 아닌 REWARD(보상)을 준다.
- 보상을 기반으로 학습한다.





Deep Learning은 무엇일까요?

Schematic Diagram of a Neural Network



다른 머신러닝
기법들과의 차이점 2 :

Nonlinear function을 풀기 위해,
Nonlinear function를
linear function의 결합으로

→ 엄청 복잡한
함수(인공지능)을
만들 수 있다.

다층 레이어 (Multiple Layer)

Hidden layer가 2개이상인
NN(Neural Network) 를
Deep Learning이라고 부른다.

쉬어가기 : linear function vs. Nonlinear function

선형회귀 모델은 '회귀계수(regression coefficient)를 선형 결합으로 표현할 수 있는 모델'입니다.

계수들과 변수의 곱셈과 그들의 덧셈과 뺄셈으로만 결합되어 있는 것을 의미합니다.

즉, 독립변수가 일차식인지, 이차식인지, 로그함수인지가 중요한 것이 아니라 추정할 대상인 파라미터가 어떻게 생겼느냐의 문제입니다. 예를 들어 다음 함수는 모두 선형회귀식입니다.

- $y = a_0 + a_1x_1$

- $y = a_0 + a_1x_1 + a_1x_2$

- $y = a_0 + a_1x_1 + a_2x_1^3$

쉬어가기 : linear function vs. Nonlinear function

그렇다면 비선형회귀 모델은 무엇일까요?

비선형 모델은 데이터를 어떻게 변형하더라도 파라미터를 선형 결합 식으로 표현할 수 없는 모델을 의미합니다.

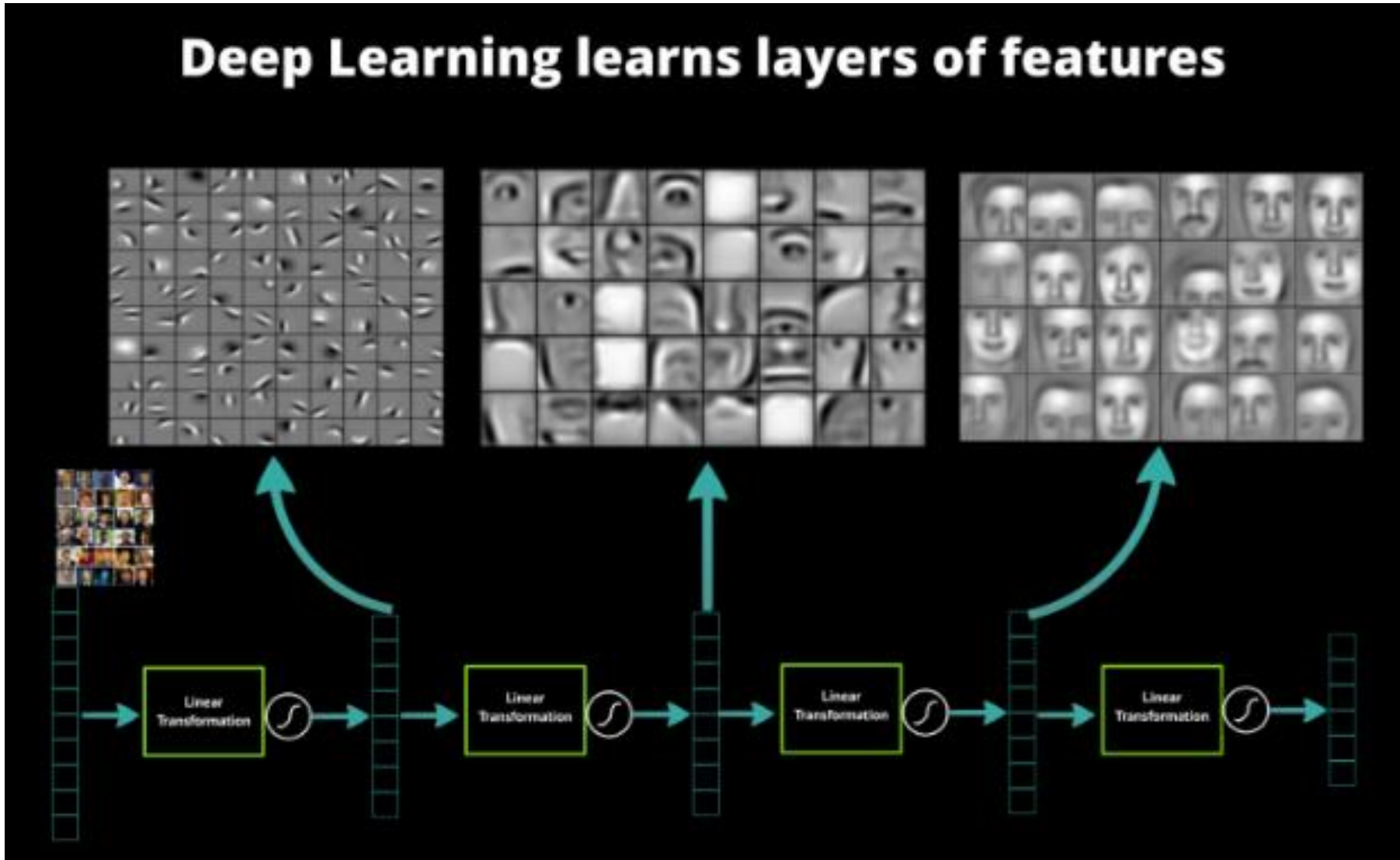
이 수식은 x, y 를 아무리 변경하더라도 파라미터를 선형회귀식으로 표현할 수 없습니다.

$$y = \frac{a_2 x_1}{a_1 + x_2}$$

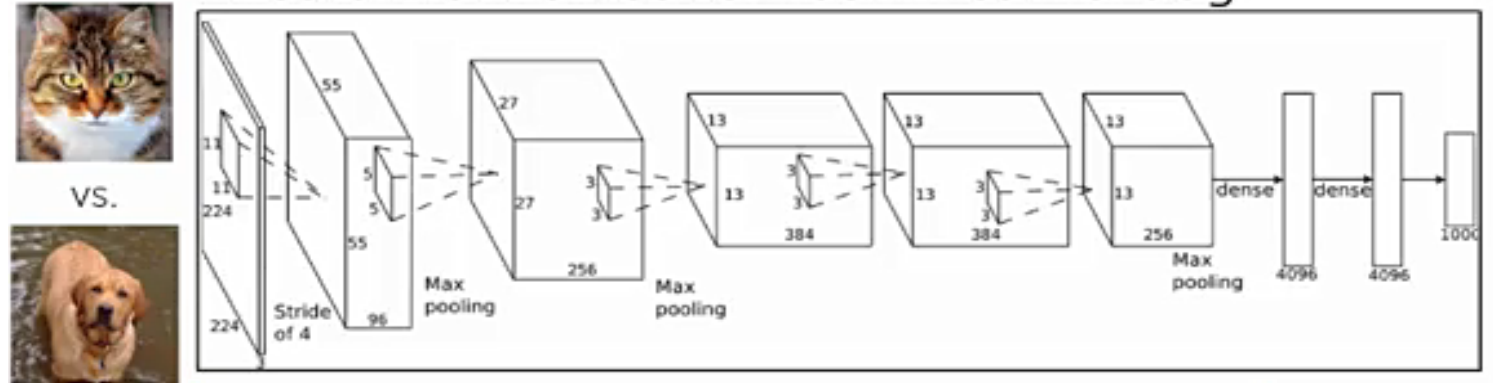
선형회귀 모델은 파라미터 계수에 대한 해석이 단순하지만, 비선형 모델은 형태가 복잡할 경우 해석하기 어렵기 때문에 통계 모델링에서는 비선형회귀 모델을 잘 사용하지 않습니다.

요약: 딥러닝을 한눈에 보기

Deep Learning learns layers of features



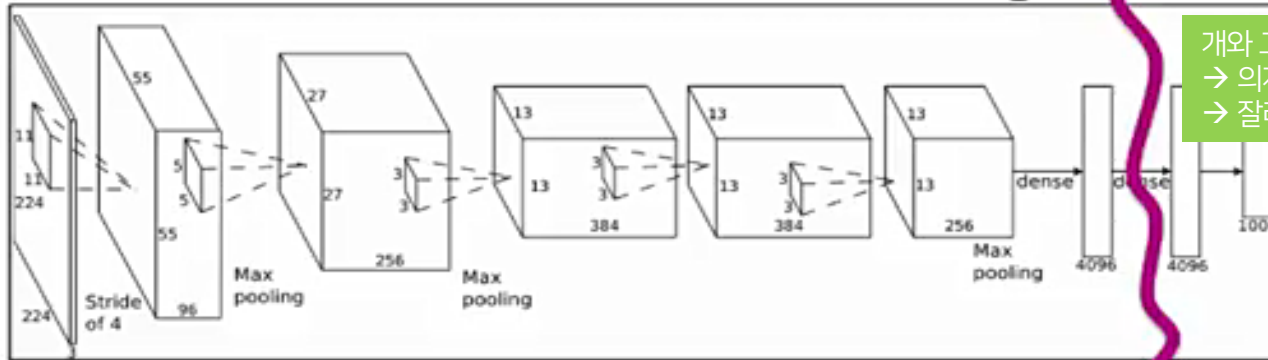
Neural net trained for Task 1: cat vs. dog



중간층은 좀더 일반적입니다. 구석, 선, 원, 구불구불한 패턴 같은 101개의 새 카테고리에도 적용될 수 있는 일반적인 패턴을 표현

고양이와 개에 아주 특화

Neural net trained for Task 1: cat vs. dog




개와 고양이 특화
→ 의자에 도움이 안됨
→ 잘라냄

일반적으로 적용

선형 분류기로 연결해서
의자, 코끼리, 카메라에 대해
조금 있는 데이터를 훈련



Machine Learning은 문제 해결을
위한 함수(f)를 찾는 것이다.


$$Y = f(x)$$

- 종속변수(y)를 독립변수(x)들의 함수 $f(x)$ 로 적합, 즉, $Y=f(x)$
- f (함수)가 무엇일까? 문제를 풀 방법? 분석방법? 을 의미한다.
 - Regression(회귀분석) Algorithm
 - Classification(지도학습) Algorithm
 - Clustering(비지도학습) Algorithm
 - Recommendation(추천) Algorithm
 - Deep Learning(딥러닝) Algorithm

$Y=f(x)$: 예제

- 종속변수(y)를 독립변수(x)들의 함수 $f(x)$ 로 적합, 즉, $Y=f(x)$
- f (함수)가 무엇일까? 문제를 풀 방법? 분석방법? 을 의미한다.

예제 1) Sales Prediction : 특정 고객 → 마케팅 Campaign 반응할 확률

- x : 고객 과거 Data, Campaign 요소들
- y : 반응할 확률 → Campaign을 확인한 사람에게 Sales를 하는 것이 좋다

예제 2) 휴대폰 고객이 향후 6개월 내에 이탈할 확률

- x : 휴대폰 고객, y : 이탈할 확률

예제 3) y : 다음 주 주가상승 여부 = $f(x$: 최근 주가 추이, 환경분석)

머신러닝 모델링 프로세스

1. 문제 정의 및 명확한 목표 설정
2. 데이터 수집 (including Random Sampling)
3. 데이터 탐색, Cleaning, Pre-processing
4. 데이터 분류 및 데이터 세팅
 - Training set, Validation set, Testing set으로 나눈다.

5. 데이터 방법론 선택

- Regression, Classification, Clustering, Recommender 등 선택

6. 구체적인 기술 및 평가 방법 선택

- Linear Regression, Logistic Regression, Decision Tree, Random Forest, KNN, K-means, Matrix Factorization

7. 테스트 및 튜닝

8. 결과 및 모델 비교

9. 모델 선택 및 적용

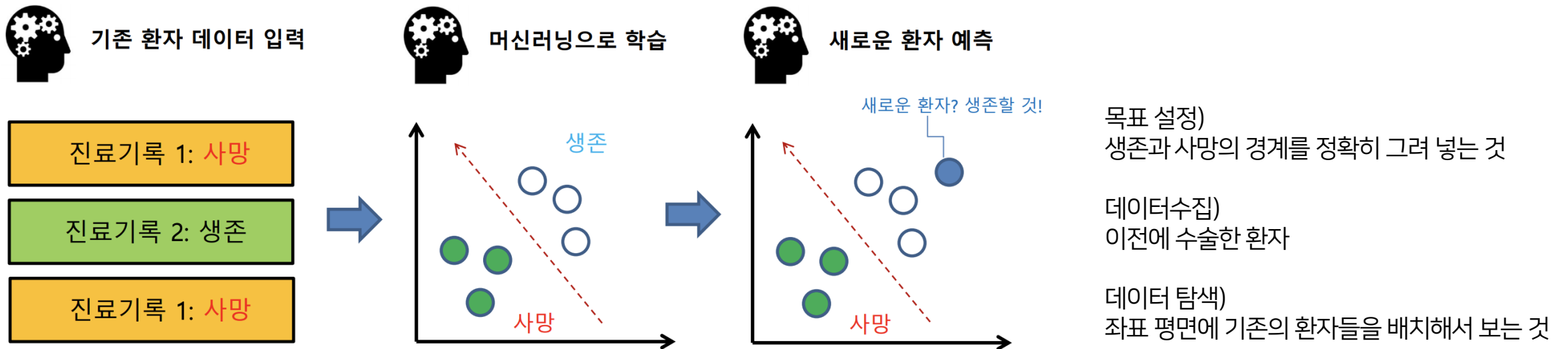
예제) 폐암수술환자의 생존율 예측하기

1. 목표 설정 및 데이터 수집/탐색

1.문제 정의 및 명확한 목표 설정

2.데이터 수집 (including Random Sampling)

3.데이터 탐색(Exploratory Data Analytics), Cleaning, Pre-processing



2. 모델링 하기

5. 데이터 방법론 선택

- Regression, Classification, Clustering, Recommendation, **Deep Learning** 등 선택

6. 구체적인 기술 및 평가 방법 선택

- Linear Regression, Logistic Regression, Decision Tree, Random Forest, KNN, K-means, Matrix Factorization, RNN, CNN 등 선택

딥러닝을 이용해서, 폐암환자 생존율을 예측해 보기 (1) : Overview

```
# -*- coding: utf-8 -*-
# 코드 내부에 한글을 사용가능 하게 해주는 부분입니다.

# 딥러닝을 구동하는 데 필요한 케라스 함수를 불러옵니다.
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# 필요한 라이브러리를 불러옵니다.
import numpy as np
import tensorflow as tf

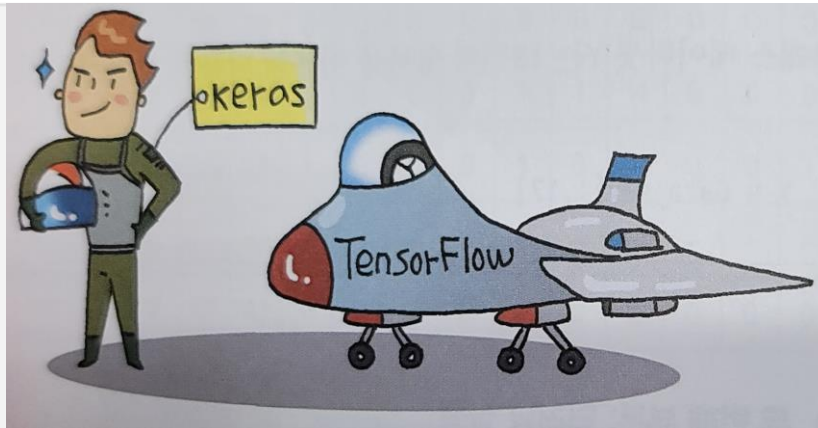
# 실행할 때마다 같은 결과를 출력하기 위해 설정하는 부분입니다.
np.random.seed(3)
tf.random.set_seed(3)

# 준비된 수술 환자 데이터를 불러들입니다.
Data_set = np.loadtxt("./dataset/ThoracicSurgery.csv", delimiter=",")

# 환자의 기록과 수술 결과를 X와 Y로 구분하여 저장합니다.
X = Data_set[:,0:17]
Y = Data_set[:,17]

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

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



딥러닝을 이용해서, 폐암환자 생존율을 예측해 보기(2): 해석하기

```
Epoch 1/100
47/47 [=====] - 1s 1ms/step - loss: 0.6482 - accuracy: 0.8128
Epoch 2/100
47/47 [=====] - 0s 1ms/step - loss: 0.4890 - accuracy: 0.8468
Epoch 3/100
47/47 [=====] - 0s 1ms/step - loss: 0.4416 - accuracy: 0.8511
Epoch 4/100
47/47 [=====] - 0s 1ms/step - loss: 0.4863 - accuracy: 0.8489
Epoch 5/100
47/47 [=====] - 0s 1ms/step - loss: 0.4430 - accuracy: 0.8532
Epoch 6/100
```

- Loss (손실값): 예측이 실패할 확률
- Accuracy (정확도): 예측이 성공할 확률
 - if: accuracy가 1.00이라면, 100% 모두 맞힌다.
 - 위의 결과에서, accuracy가 0.8532이라는 의미는 정확도가 85%를 의미한다. 즉, 예측성공률이 85%
 - 새로운 환자의 정보를 입력하면, 이 환자의 수술 후 생존율을 85%이상 맞힌다는 의미

| 최적화 과정 |

- (Training) 예측 성공률은 데이터를 분석해 데이터를 확장하거나, 딥러닝 구조를 적절하게 바꾸는 등의 노력으로 더 향상가능
- (Testing) 학습에 사용되지 않은 데이터를 따로 모아 테스트를 해보면서, 예측률이 정말로 가능한지 확인 가능

딥러닝을 이용해서, 폐암환자 생존율을 예측해 보기 (3): 데이터 세팅

4. 데이터 분류 및 데이터 세팅

- Training set, Validation set, Testing set으로 나눈다.



쉬어가기. 데이터셋에 대해서 이해하기

- Training data 로 모델링 하고, Training data로 test → 기출 문제를 주고, 그것만 공부한 후, 다시 똑같은 문제를 주고, 시험보는 행위
- Training data로 모델링 하고, Test data로 test → 교과서를 공부하고, 새로운 문제를 받아서, 시험 보는 행위
- Training data로 모델링 하고, Validation data 로 tuning 및 test 하고, Test data로 테스트 하는 것
→ 교과서(training data)로 공부하고, 기출 문제(수능 모의 고사: validation data)로 점검하고,
새로운 문제(수능: test data)로 테스트하는것
- Training data의 일부를 Validation data 으로 나눈다. → training 에서 validation data 는 사용하지 않는다.
- 학습에 사용되지 않은 데이터를 따로 모아 테스트를 해보면서, 예측률이 정말로 가능한지 확인 가능
→ 수능(testing data) 을 잘 보기 위해서 열심히 text book (training data) 공부하는 것!!
→ 실전 모의 고사(validation data)로 테스트 및 검증하면서 실제 수능(test data 기반 test)에서 좋은 성적을 거두는 것

딥러닝을 이용해서, 폐암환자 생존율을 예측해 보기 (4):데이터를 불러와서 코딩해 봅시다.

```
# 필요한 라이브러리를 불러옵니다.
import numpy as np
import tensorflow as tf

# 실행할 때마다 같은 결과를 출력하기 위해 설정하는 부분입니다.
np.random.seed(3)
tf.random.set_seed(3)

# 준비된 수술 환자 데이터를 불러들입니다.
Data_set = np.loadtxt("./dataset/ThoracicSurgery.csv", delimiter=",")

# 환자의 기록과 수술 결과를 X와 Y로 구분하여 저장합니다.
X = Data_set[:,0:17]
Y = Data_set[:,17]
```

```
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
```

딥러닝을 이용해서, 폐암환자 생존율을 예측해 보기 (4):데이터를 불러와서 코딩해 봅시다.

- |라이브러리란| 특정 기능을 담은 작은 프로그램들 (Module)
 - 함수나 클래스를 따로 담아 라이브러리 형태로 공개한 것이 많음
 - 이렇게 미리 만들어진 라이브러리를 불러와서 언제든지 그 기능을 사용하면 됨!!
- `import numpy as np` → numpy 라는 라이브러리를 np 라고 지칭한다는 뜻,
 - numpy는 수치 계산을 위해 만들어진 라이브러리로 데이터 분석에 많이 사용됨.
- Data_set이라는 임시 저장소를 만듦.
- `np.loadtxt()`: numpy 라이브러리 안에 있는 `load.txt()`함수를 사용하여, ThoracicSurgery.csv라는 외부데이터 셋을 불러옴.
- |데이터 관찰| 470개 라인으로 이루어져 있고, 각 라인은 18개 항목으로 이루어짐
 - 0부터 시작해서 17개를 가온다, 즉, 0~16까지는 입력, X,
 - 18번째 항목, index = 17 은 target, Y
- Machine Learning에서 알고리즘이나 좋은 컴퓨터 환경만큼 중요한 것이 바로 데이터를 준비하는 것!!

딥러닝을 이용해서, 폐암환자 생존율을 예측해 보기 (4):데이터를 불러와서 코딩해 봅시다.

Console 1

01_My_First_DeepLearning...ThoracicSurgery.csvThoracicSurgery.csv

Delimiter: ,

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

```

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 18,2,2.96,1.67,0,0,0,0,1,0,0,12,0,0,0,1,0,61,0
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

```

- 환자마다 18개의 정보를 순서에 맞춰 정리 → 각 줄마다 18개의 정보를 갖는다.
 - 앞의 17개는 종양 유형, 폐활량, 호흡곤란여부, 고통 정도, 기침, 흡연, 천식여부 등
 - 마지막 컬럼: 1(true), 0(false) → 18번째 정보는 수술 후 생존결과
- 실습의 목적: 1번째부터 17번째까지의 항목을 분석해서 18번째 항목, 즉 수술 후 생존 또는 사망을 맞추는 것
 - 1번째부터 17번째까지의 항목: 속성(attribute) ← X
 - 18번째 항목: 클래스(class) ← 이론표, 라벨링 ← Y
 - 딥러닝 구동시킬 때,
 - “속성(컬럼, 특성, x값)” 만을 뽑아 데이터 셋을 만들고,
 - “클래스(타겟, y값)” 를 담은 데이터 셋을 따로 만들어 준다.

딥러닝을 이용해서, 폐암환자 생존율을 예측해 보기 (5): 딥러닝 실행하기

딥러닝을 구동하는 데 필요한 케라스 함수를 불러옵니다.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

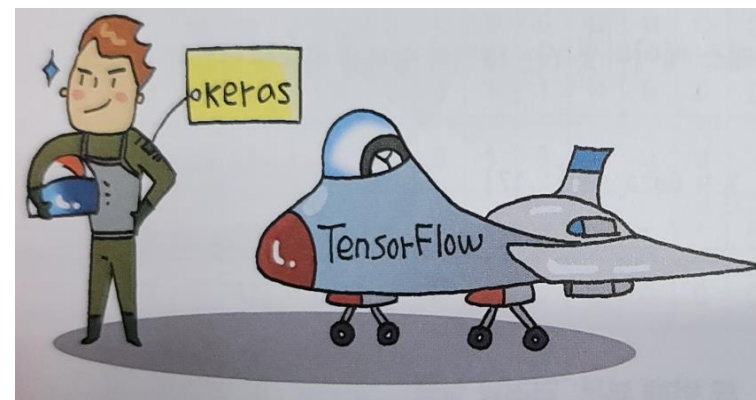
딥러닝 구조를 결정합니다(모델을 설정하고 실행하는 부분입니다).

```
model = Sequential()
model.add(Dense(30, input_dim=17, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

딥러닝을 실행합니다.

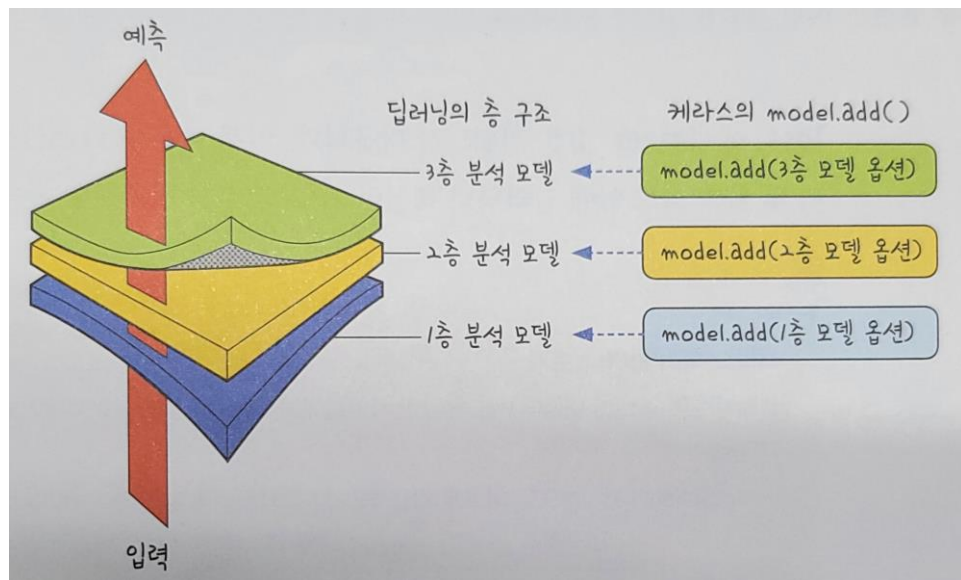
```
model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])
model.fit(X, Y, epochs=100, batch_size=10)
```

- 케라스(keras)를 사용해 딥러닝을 실행시킴.
- 텐서플로(tensorflow)가 미리 설치되어 있어야 함.
- 딥러닝 프로젝트: 여행
- 텐서플로: 목적지를 빠르게 이동시켜 주는 비행기
- 케라스: 비행기의 이륙 및 정확한 지점까지의 도착을 책임지는 '파일럿'
- 딥러닝을 구동시키는 라이브러리에는 텐서플로우 외에도 파이토치(pytorch) 등이 있다.



딥러닝을 이용해서, 폐암환자 생존율을 예측해 보기 (5): 딥러닝 실행하기

```
# 딥러닝을 구동하는 데 필요한 케라스 함수를 불러옵니다.  
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Dense
```



- 설치가 올바르게 되었다면, 설치된 keras library를 불러온다.
- Sequential() 함수와 Dense() 함수를 불러온다.
- Sequential()함수는 딥러닝의 구조를 한층 한층 쉽게 쌓는다.
 - Sequential()함수를 선언하고 나서, model.add()함수를 사용해 필요한 층을 차례로 추가한다.
 - Keras의 가장 큰 장점: model.add()함수를 이용해 필요한 만큼의 층을 빠르고 쉽게 쌓아 올릴 수 있다.
 - Model.add()함수 안에, Dense()함수가 포함되어 있다.
 - Dense()함수는 조밀하게 모여있는 집합의 뜻으로, 각 층이 어떤 특징을 가질지 옵션을 설정할 수 있다.

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

model.add()함수를 이용해 두 개의 층을 쌓아 올림
→ model.add()가 2개!!

딥러닝을 이용해서, 폐암환자 생존율을 예측해 보기 (5): 딥러닝 실행하기

딥러닝을 실행합니다.

```
model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])  
model.fit(X, Y, epochs=100, batch_size=10)
```

- 딥러닝 구조(Sequential)와 층별옵션(Dense)를 정하고 나면 compile()함수를 이용해 실행한다.
 - loss: 한 번 신경망이 실행될 때마다 오차 값을 추적하는 함수
 - optimizer: 오차를 어떻게 줄여나갈지를 정하는 함수
 - activation: 다음 층으로 어떻게 값을 넘길지 결정하는 부분
 - 가장 많이 사용되는 함수(relu(), sigmoid())

	내 용	비 고
1주 (8/29)	강의소개	
2주 (9/5)	Overview	실습준비 자기 소개 section, 조편성 (3인 1조)
3주(9/12)	Regression 이해하기	
4주(9/19)	Regression 실습 및 사례로 배우기	
5주(9/26)	Regression 이해를 점검하기 주제를 선택해서, Regression으로 분석하기 - 주제 선정 - 주제선택이유 (회귀분석이 가능한 이유, 목적 명시) - 분석 - 결과 및 인사이트	Take Home : 중간고사
6주(10/3)	개천절	휴강
7주(10/10)	Midterm Recital	Peer-Review
8주(10/17)	Supervised Learning 이해하기	
9주(10/24)	Supervised Learning / 실습 및 사례로 배우기 Unsupervised Learning 이해하기	
10주(10/31)	Unsupervised Learning 실습하기	
11주 (11/7)	Recommender System 이해하기 / 실습 및 사례로 배우기	
12주(11/14)	프로젝트 설명 및 Proposal 준비시간	프로젝트 설명 및 이전 예제
13주(11/21)	Final Project Proposal	조별 제출 및 발표
14주(11/28)	프로젝트 준비 시간 및 질의/응답시간	
15주(12/5)	Final Presentation	기말고사



Thank You.