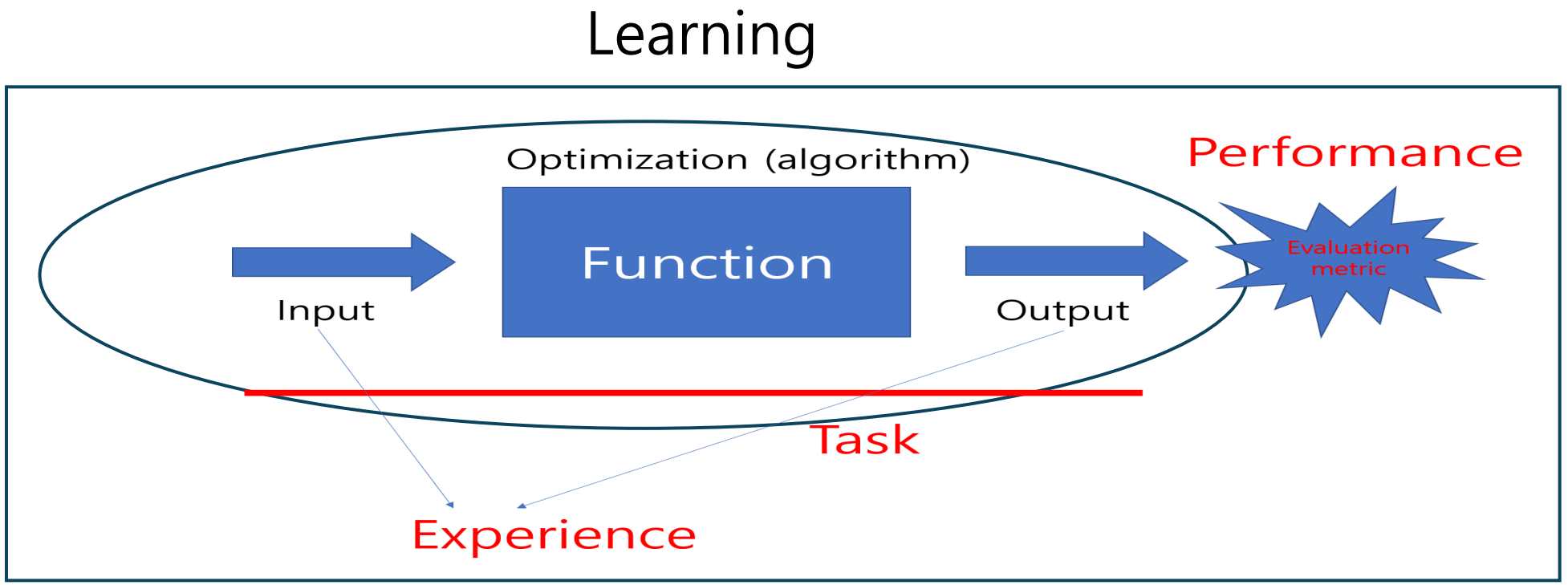
수업 시작 전 준비사항

- 인공지능 수업은 Google Colab으로 실습이 진행 될 것임.
- Colab 사용하는 방법이나, colab 처음 다루는 학생은 문의부탁

AI Goals

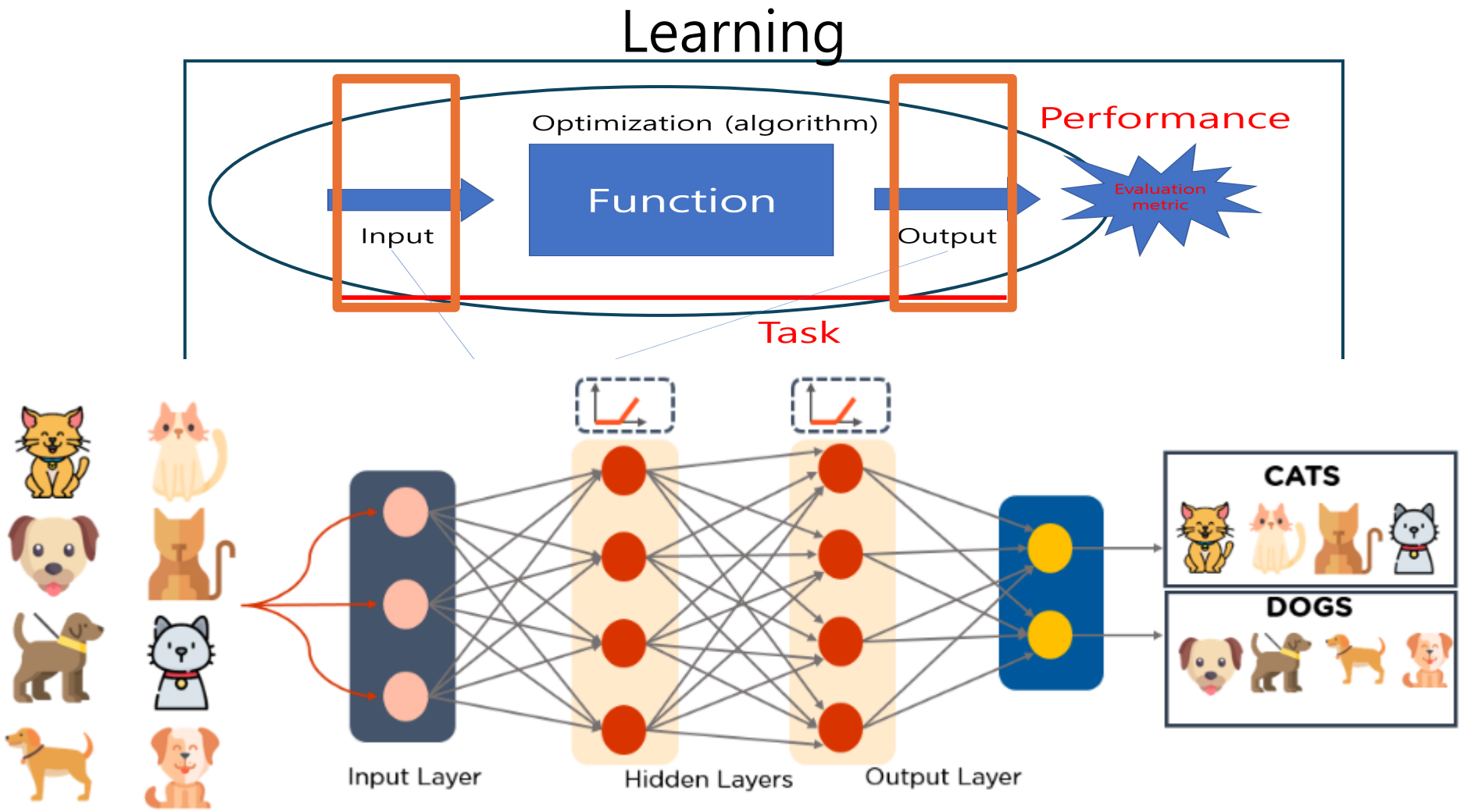


Goal

Function → Target Function

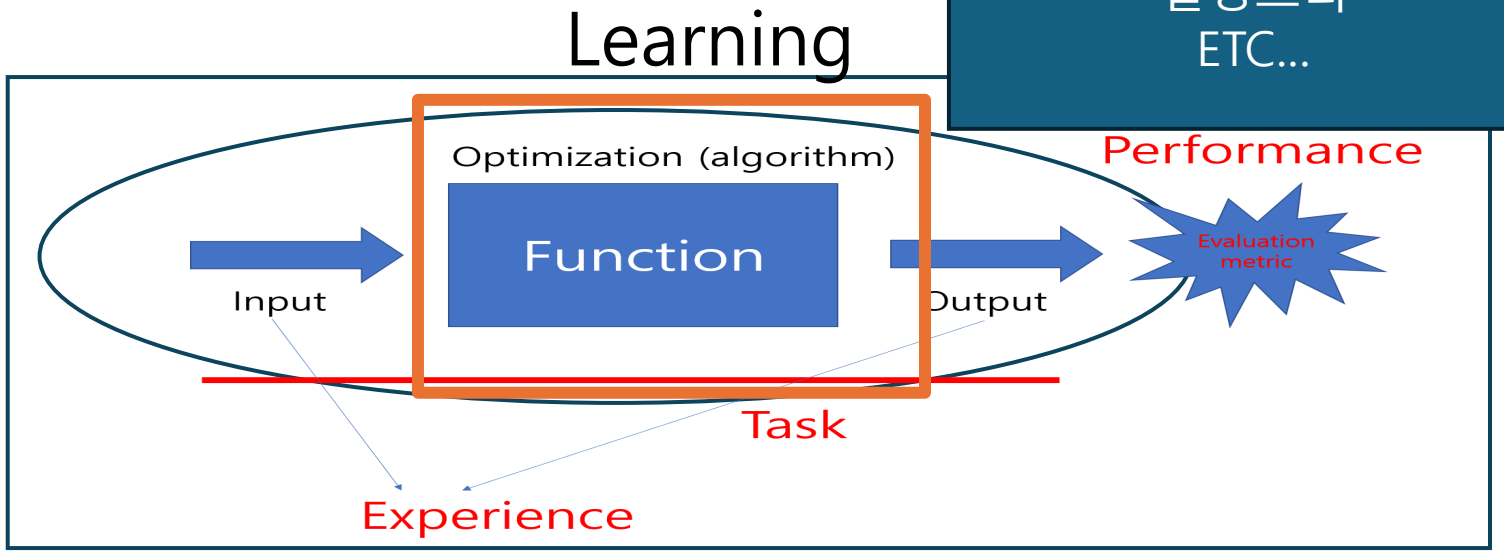
머신러닝으로 근사

AI Goals


Learning
Optimization (algorithm)
Function
Input
Output
Performance
Evaluation metric
Task
CATS
DOGS
Input Layer
Hidden Layers
Output Layer

AI Goals

퍼셉트론,
KNN, 선형회귀
결정트리
ETC…

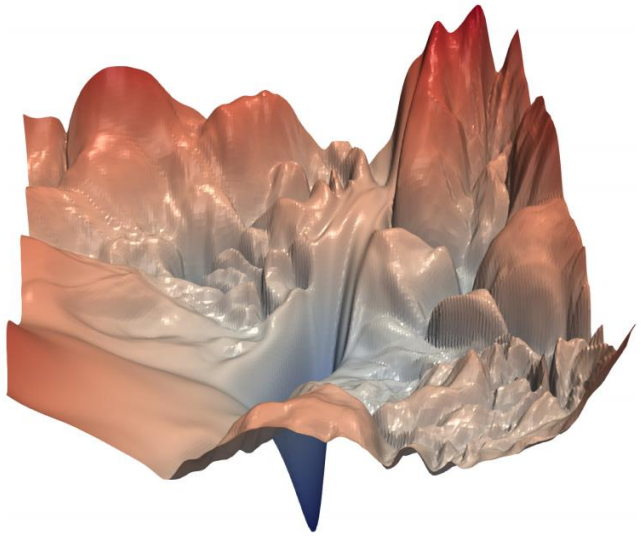


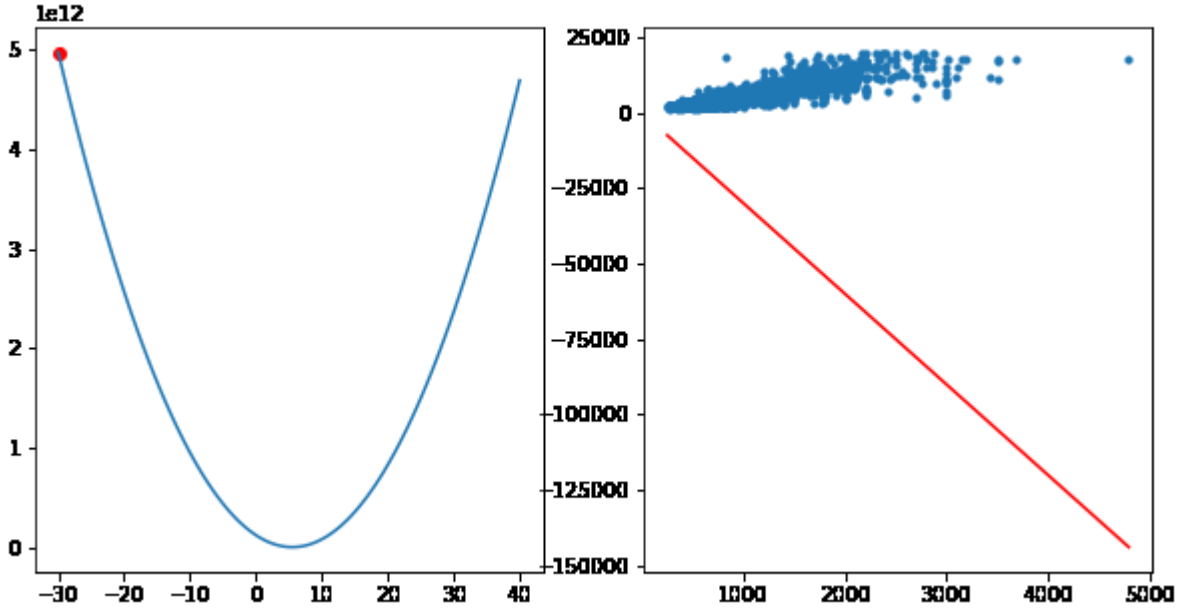
Goal

Function → Target Function

머신러닝으로 근사

AI Goals





Goal

Function

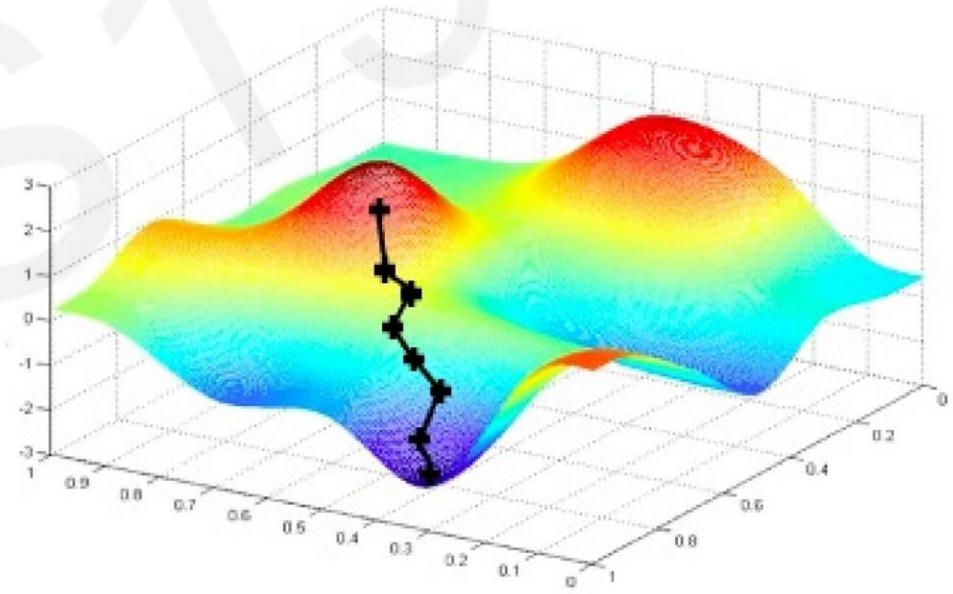
머신러닝으로 근사

Target Function

Gradient Descent

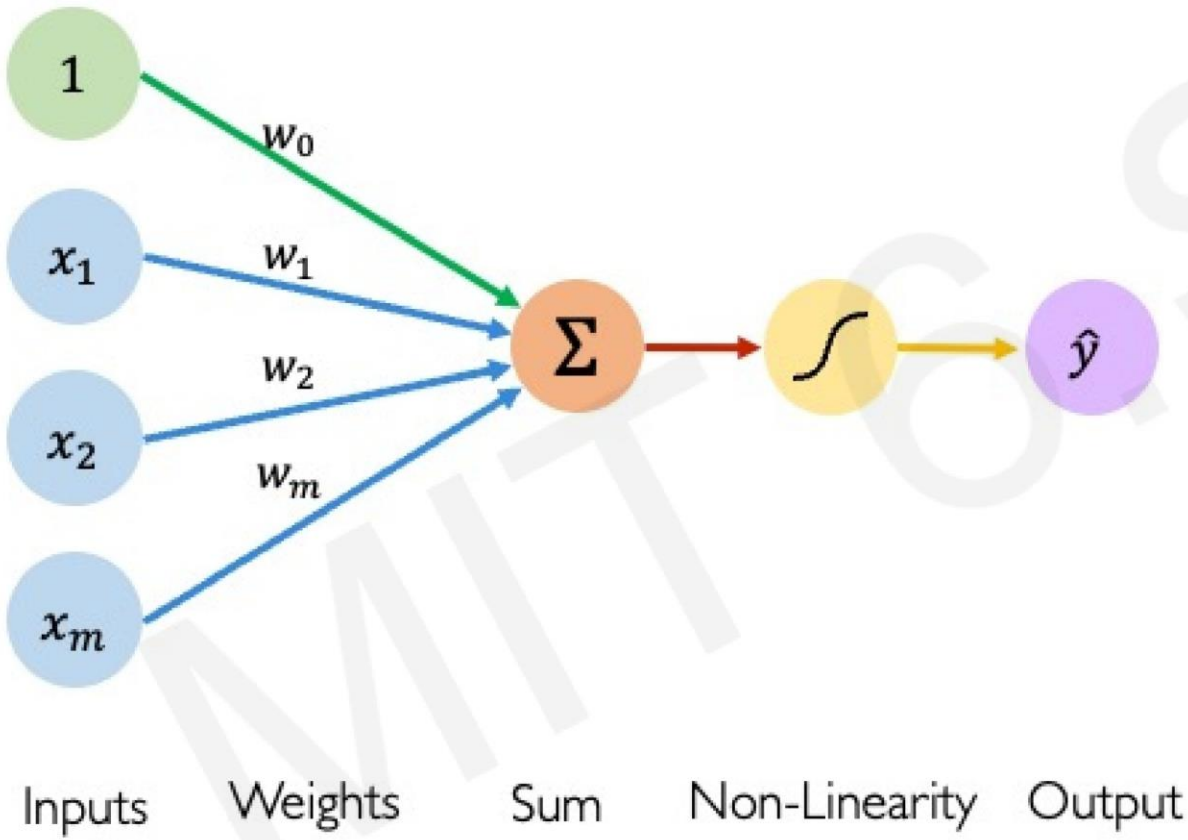
Algorithm

1. Initialize weights randomly $\sim \mathcal{N}(0, \sigma^2)$
2. Loop until convergence:
3. Compute gradient, $\frac{\partial J(W)}{\partial W}$
4. Update weights, $W \leftarrow W - \eta \frac{\partial J(W)}{\partial W}$
5. Return weights



Can be very **computationally intensive** to compute!

The Perceptron: Forward Propagation



Output

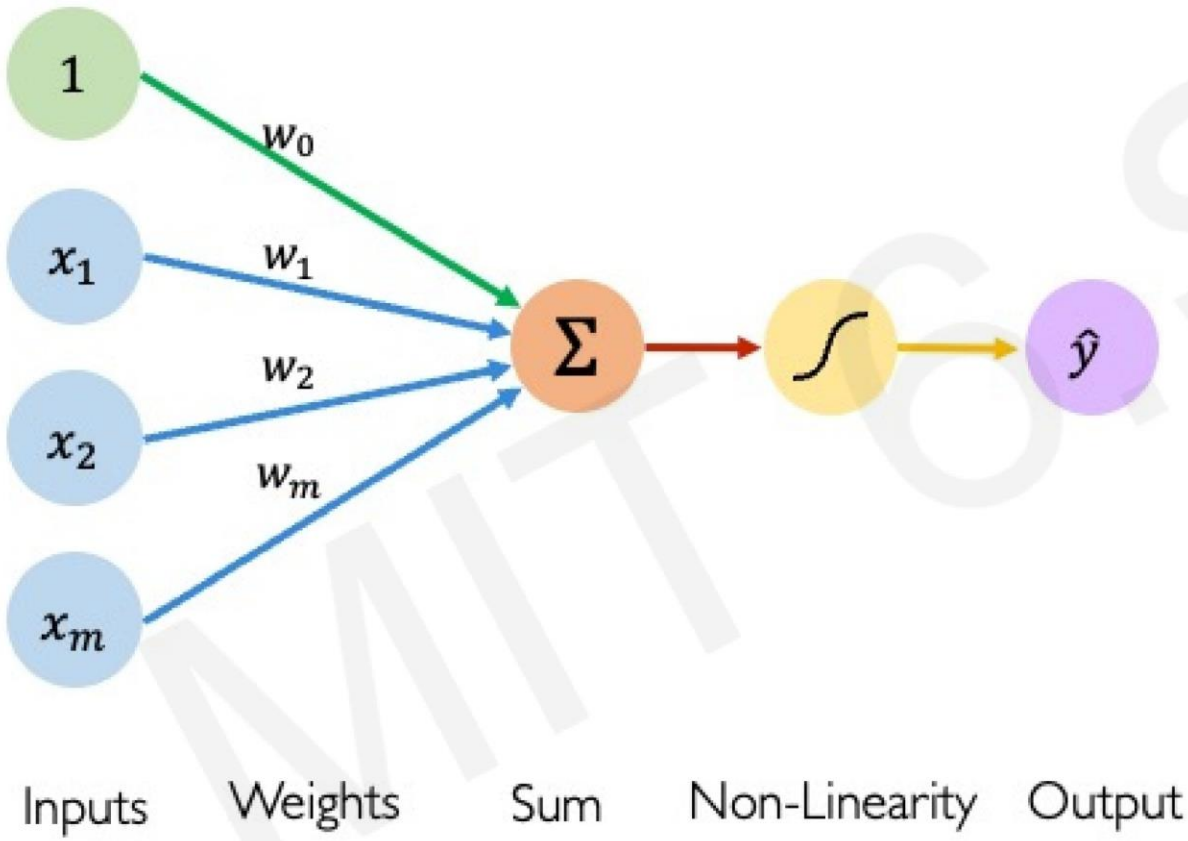
Linear combination of inputs

$$\hat{y} = g\left(w_0 + \sum_{i=1}^{m} x_i\, w_i\right)$$

Non-linear activation function

Bias

The Perceptron: Forward Propagation

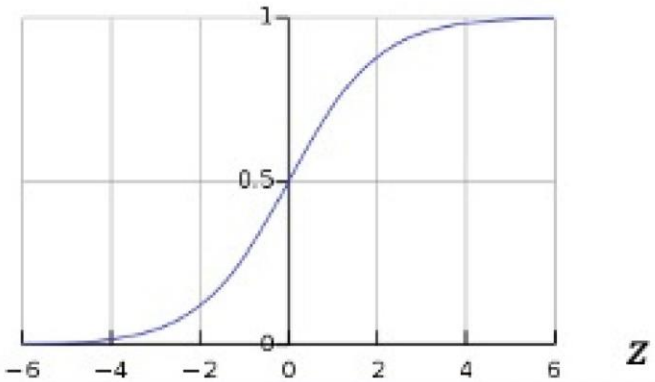


Activation Functions

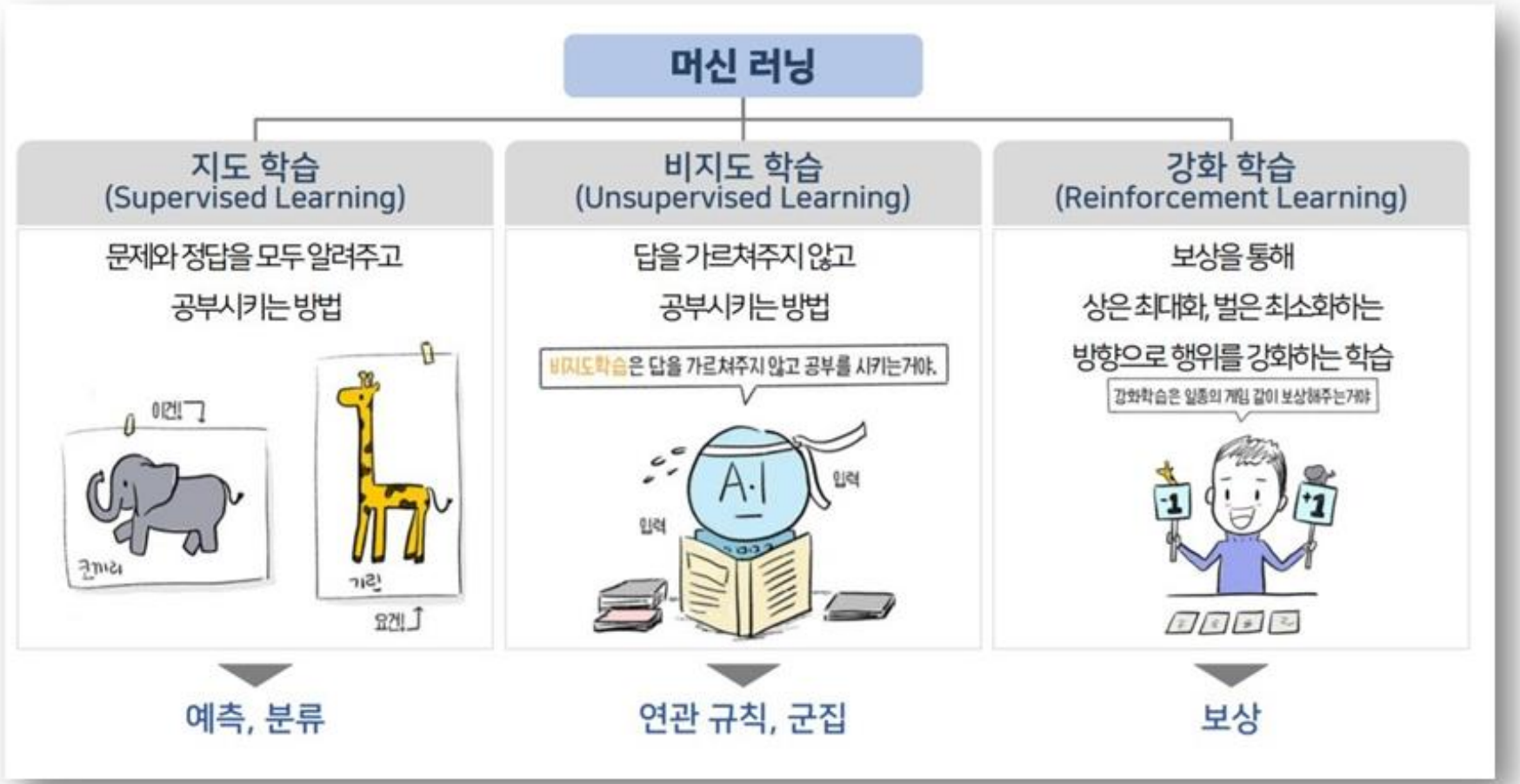
$$\hat{y} = g\left(w_0 + \boldsymbol{X}^T \boldsymbol{W} \right)$$

- Example: sigmoid function

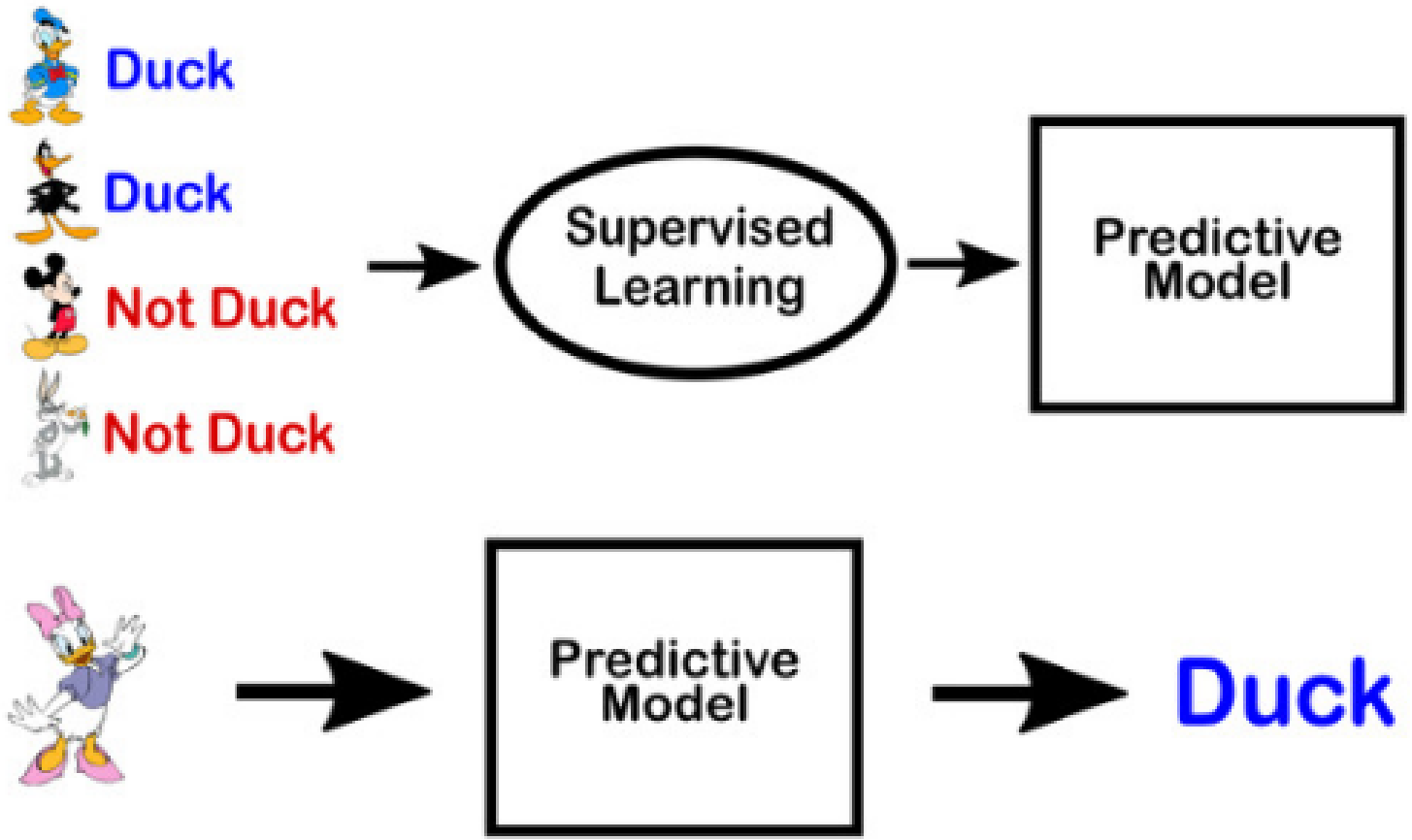
$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$



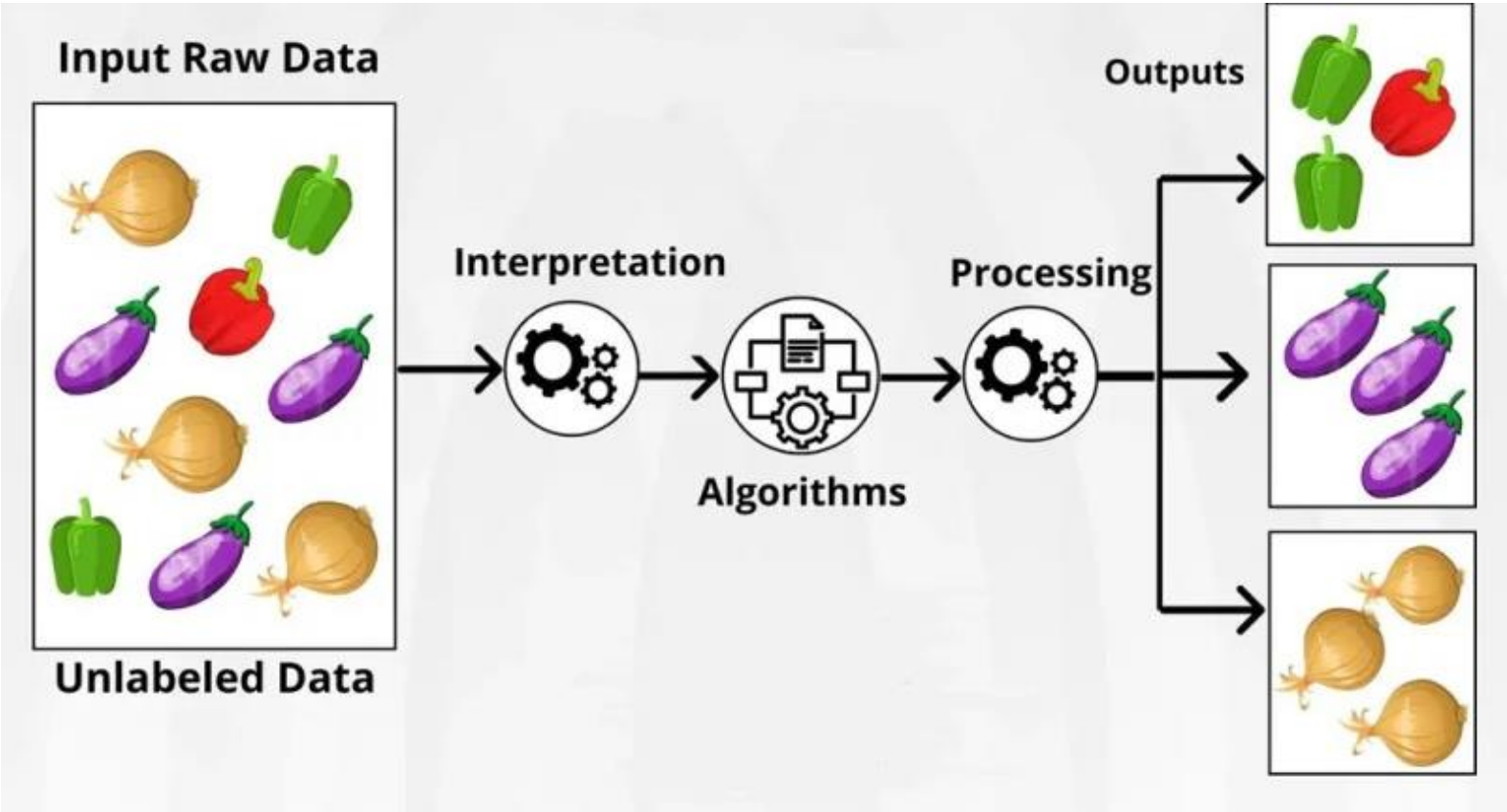
머신러닝 문제 종류


머신 러닝
지도 학습
(Supervised Learning)
문제와 정답을 모두 알려주고
공부시키는 방법
이건 코끼리
이건 기린
비지도 학습
(Unsupervised Learning)
답을 가르쳐주지 않고
공부시키는 방법
비지도학습은 답을 가르쳐주지 않고 공부를 시키는거야.
A·I
입력
입력
강화 학습
(Reinforcement Learning)
보상을 통해
상은 최대화, 벌은 최소화하는
방향으로 행위를 강화하는 학습
강화학습은 일종의 게임 같이 보상해주는거야
-1
+1
예측, 분류
연관 규칙, 군집
보상

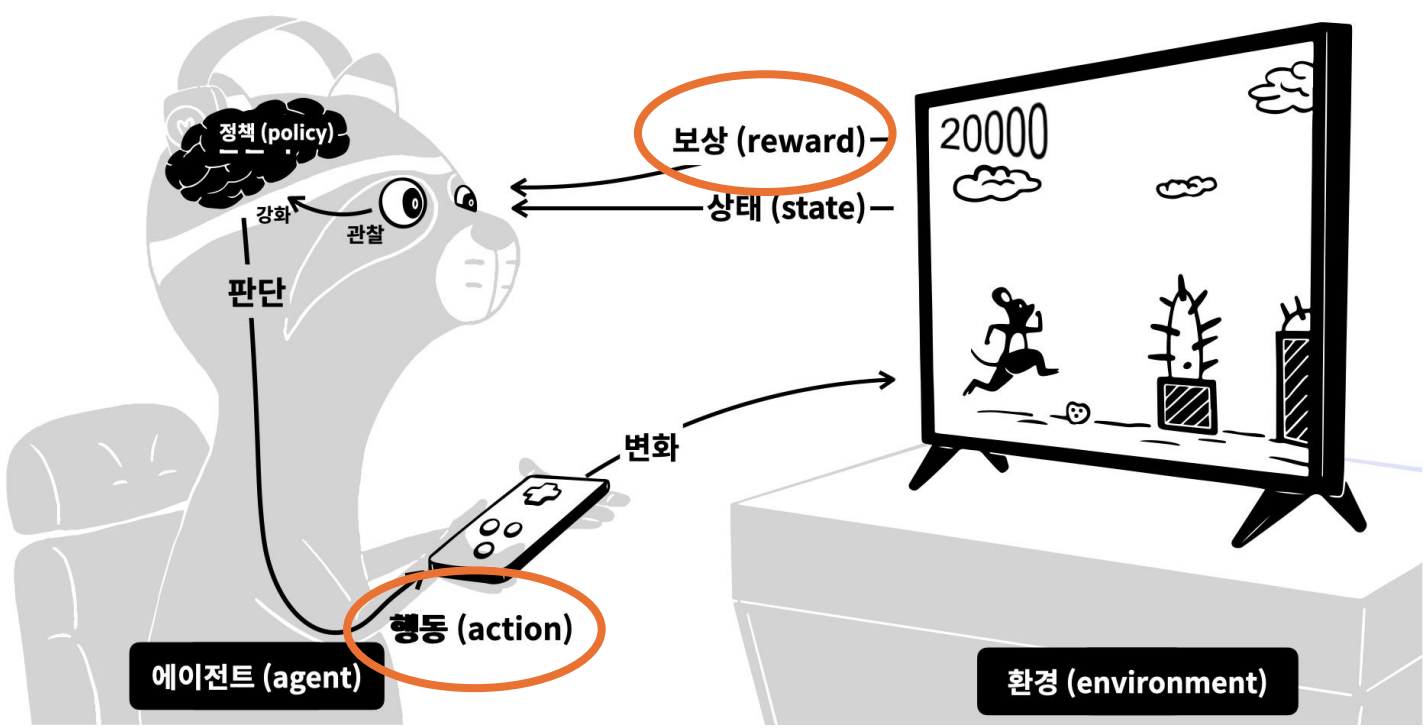
지도학습

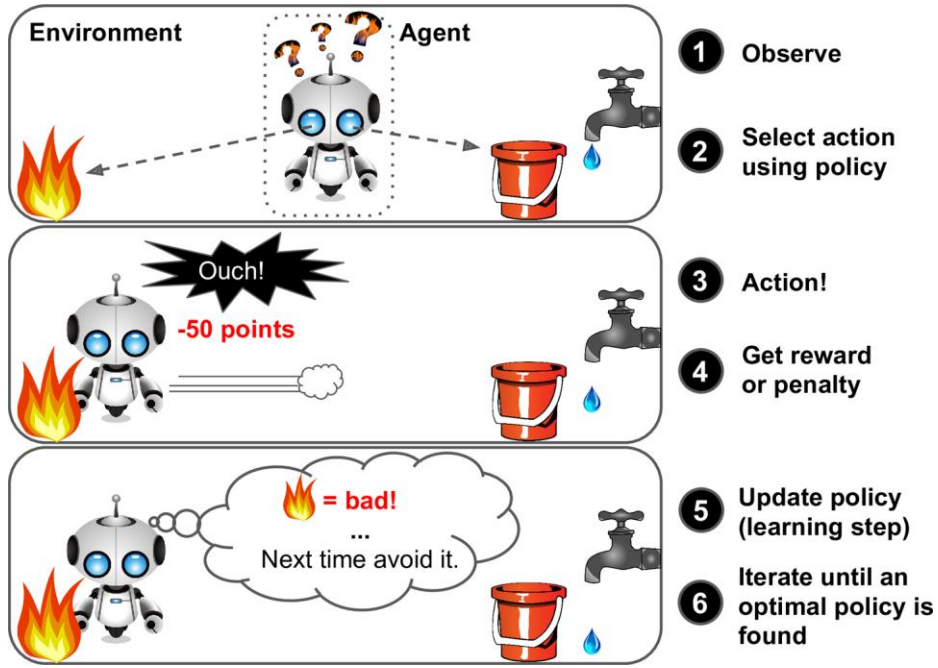

Duck
Duck
Not Duck
Not Duck
Supervised Learning
Predictive Model
Predictive Model
Duck

비지도 학습

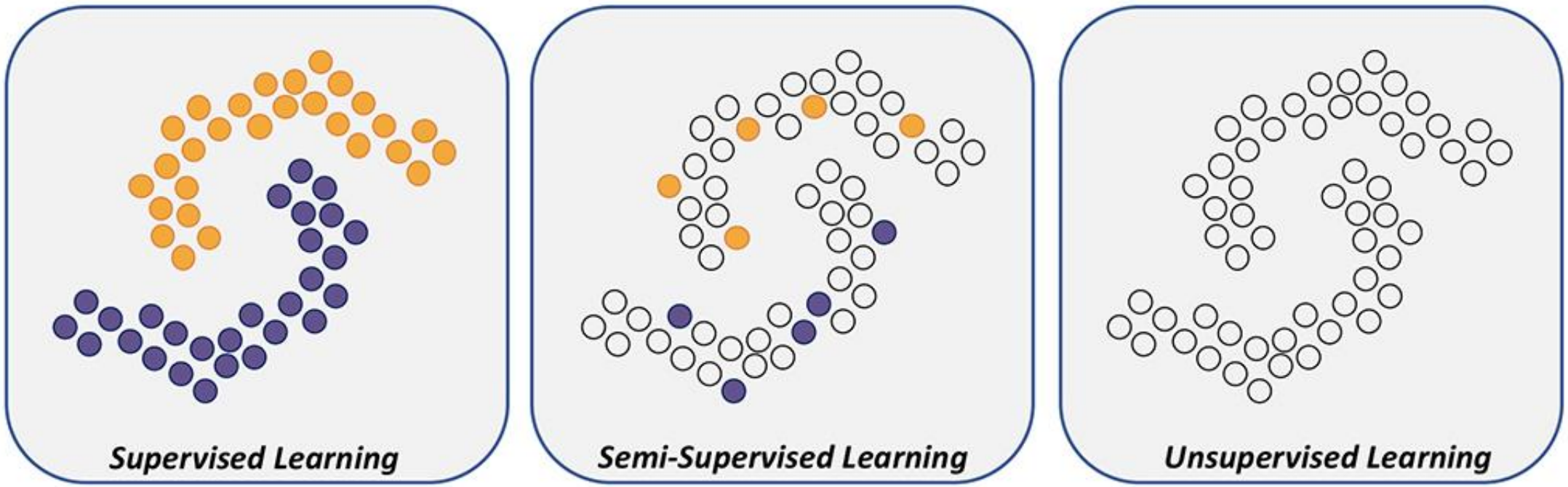

Input Raw Data
Unlabeled Data
Interpretation
Algorithms
Processing
Outputs

강화 학습


정책 (policy)
강화
관찰
판단
보상 (reward)
상태 (state)
변화
행동 (action)
에이전트 (agent)
20000
환경 (environment)

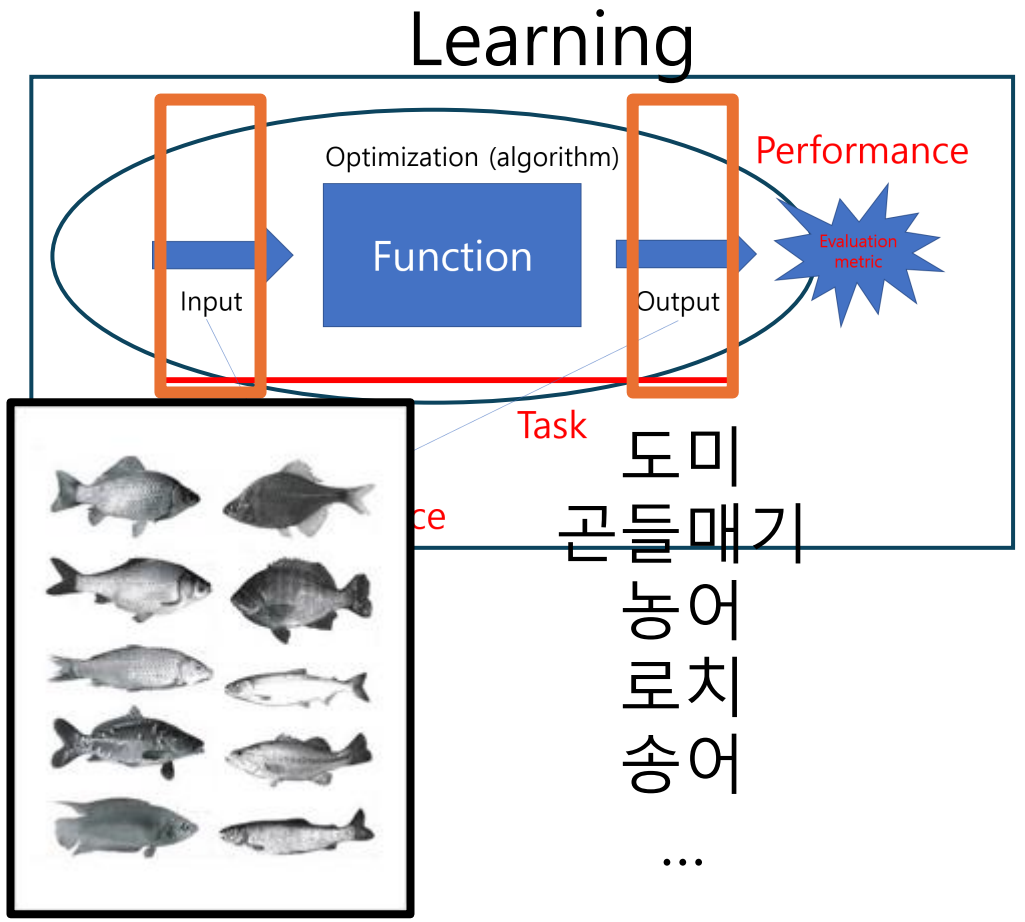

Environment
Agent
Ouch!
-50 points
= bad!
...
Next time avoid it.
1 Observe
2 Select action using policy
3 Action!
4 Get reward or penalty
5 Update policy (learning step)
6 Iterate until an optimal policy is found

준 지도학습


Supervised Learning
Semi-Supervised Learning
Unsupervised Learning

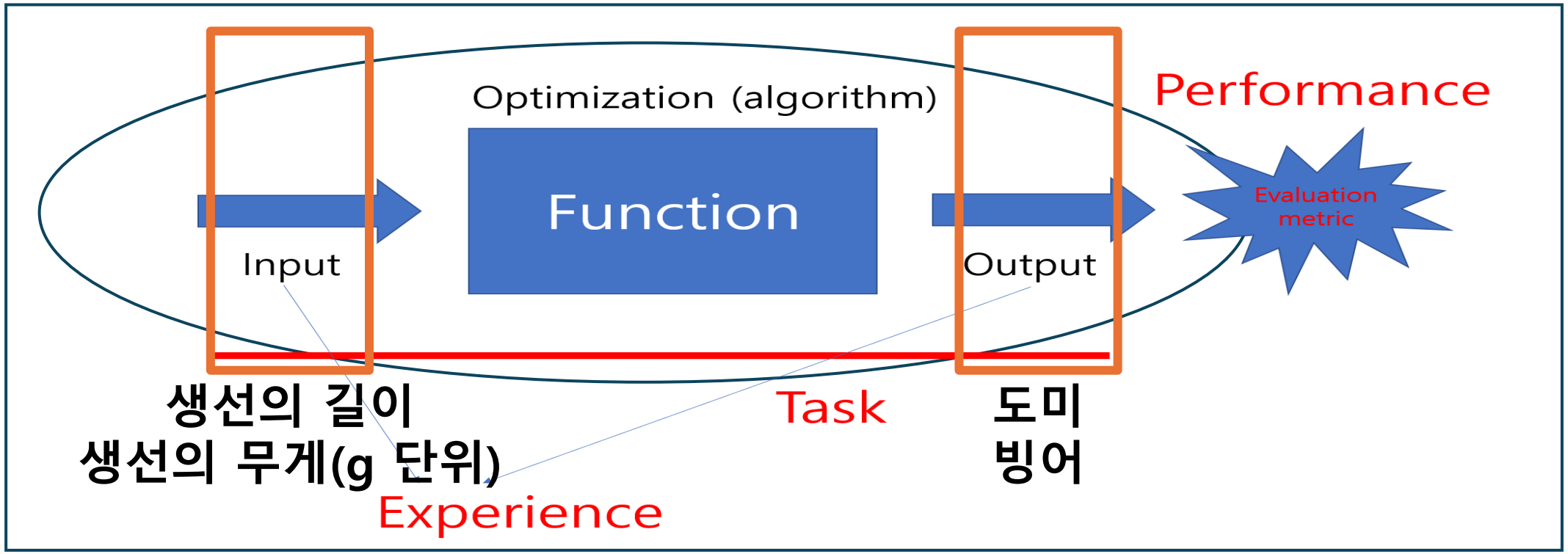
생선 분류 문제 해결해보기

- 여러가지의 생선을 어떤 것을 구분을 해야하는지...
  - Ex) 도미, 곤들매기, 농어, 로치 빙어, 송어 등...
- 여러가지 생선을 구분하기 위해 필요한 특성은 무엇인지...
  - Ex) 각 종류의 이미지, 물고기의 길이, 무게 등...

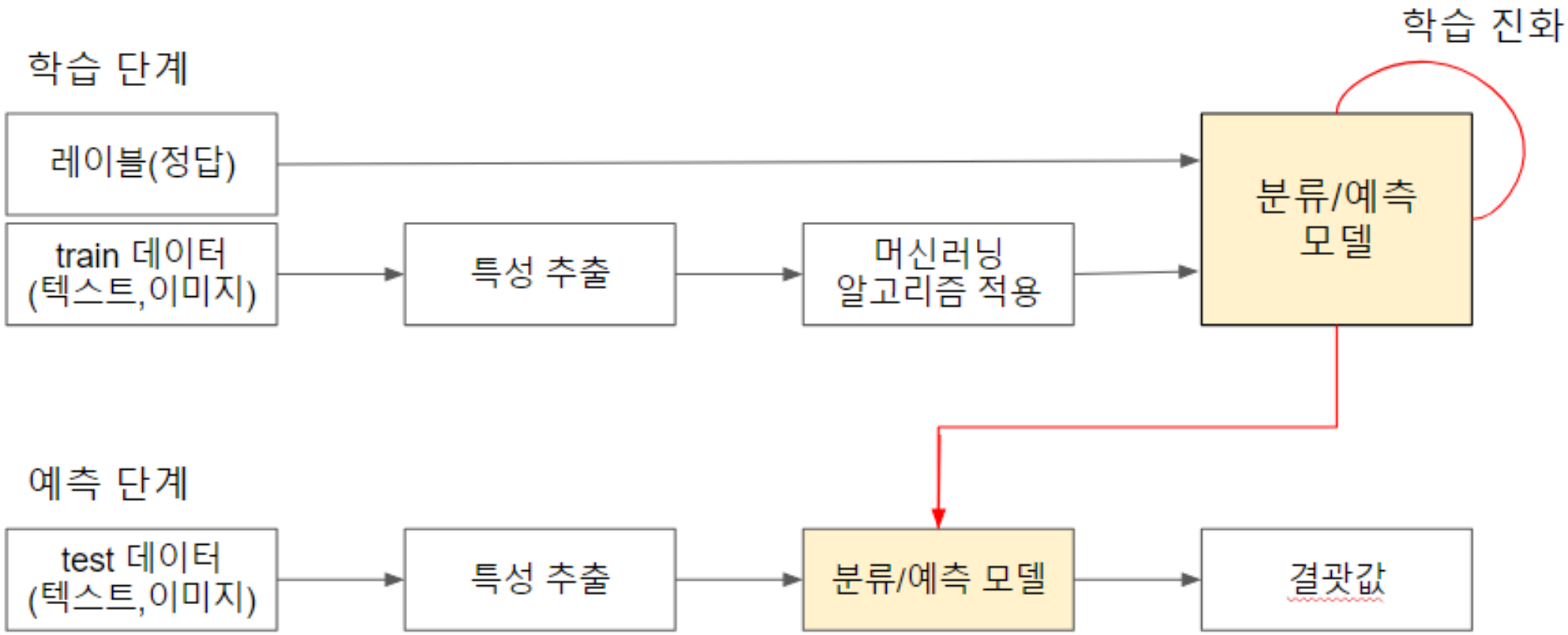


이 책에서 물고기 분류 제안

Learning



인공지능 문제 해결 순서


학습 진화
학습 단계
레이블(정답)
train 데이터
(텍스트,이미지)
특성 추출
머신러닝
알고리즘 적용
분류/예측
모델
예측 단계
test 데이터
(텍스트,이미지)
특성 추출
분류/예측 모델
결괏값

도미 데이터 준비

```
bream_length = [25.4, 26.3, 26.5, 29.0, 29.0, 29.7, 29.7, 30.0, 30.0, 30.7, 31.0, 31.0,
31.5, 32.0, 32.0, 32.0, 33.0, 33.0, 33.5, 33.5, 34.0, 34.0, 34.5, 35.0, 35.0, 35.0, 35.0,
36.0, 36.0, 37.0, 38.5, 38.5, 39.5, 41.0, 41.0]

bream_weight = [242.0, 290.0, 340.0, 363.0, 430.0, 450.0, 500.0, 390.0, 450.0, 500.0,
475.0, 500.0, 500.0, 340.0, 600.0, 600.0, 700.0, 700.0, 610.0, 650.0, 575.0, 685.0,
620.0, 680.0, 700.0, 725.0, 720.0, 714.0, 850.0, 1000.0, 920.0, 955.0, 925.0, 975.0,
950.0]
```

http://bit.ly/bream_list

도미 데이터 시각화

```
import matplotlib.pyplot as plt

plt.scatter(bream_length, bream_weight)
plt.xlabel('length')
plt.ylabel('weight')
plt.show()
```

빙어 데이터 준비

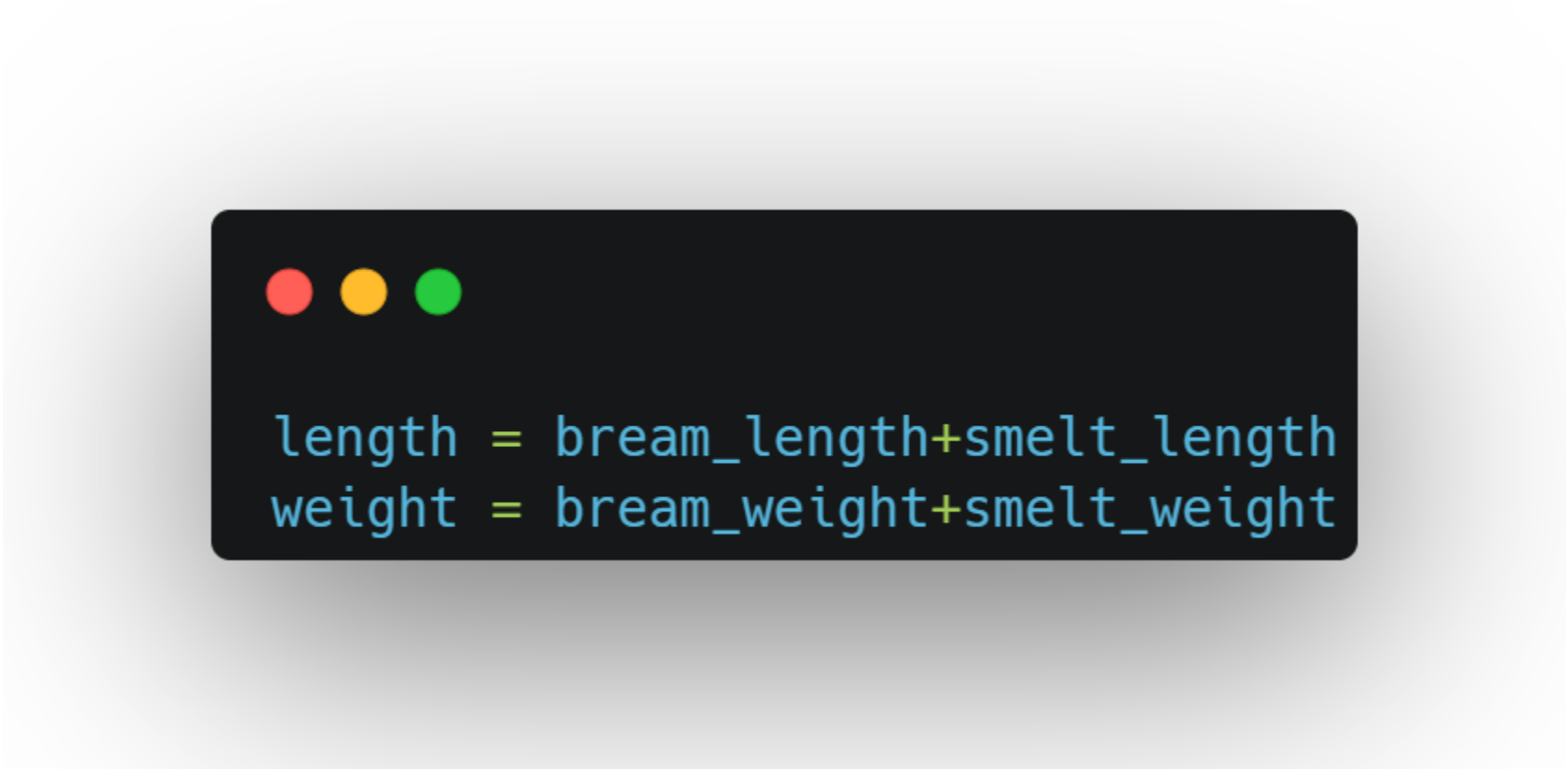
```
smelt_length = [9.8, 10.5, 10.6, 11.0, 11.2, 11.3, 11.8, 11.8, 12.0, 12.2, 12.4, 13.0,
14.3, 15.0]
smelt_weight = [6.7, 7.5, 7.0, 9.7, 9.8, 8.7, 10.0, 9.9, 9.8, 12.2, 13.4, 12.2, 19.7,
19.9]
```

http://bit.ly/smelt_list

도미, 빙어에 해당하는 데이터 시각화

```
plt.scatter(bream_length, bream_weight)
plt.scatter(smelt_length, smelt_weight)
plt.xlabel('length')
plt.ylabel('weight')
plt.show()
```

길이 변수와 무게 변수 합치기



X 변수 데이터 제작

```
fish_data = [[l, w] for l, w in zip(length, weight)]

print(fish_data)
```

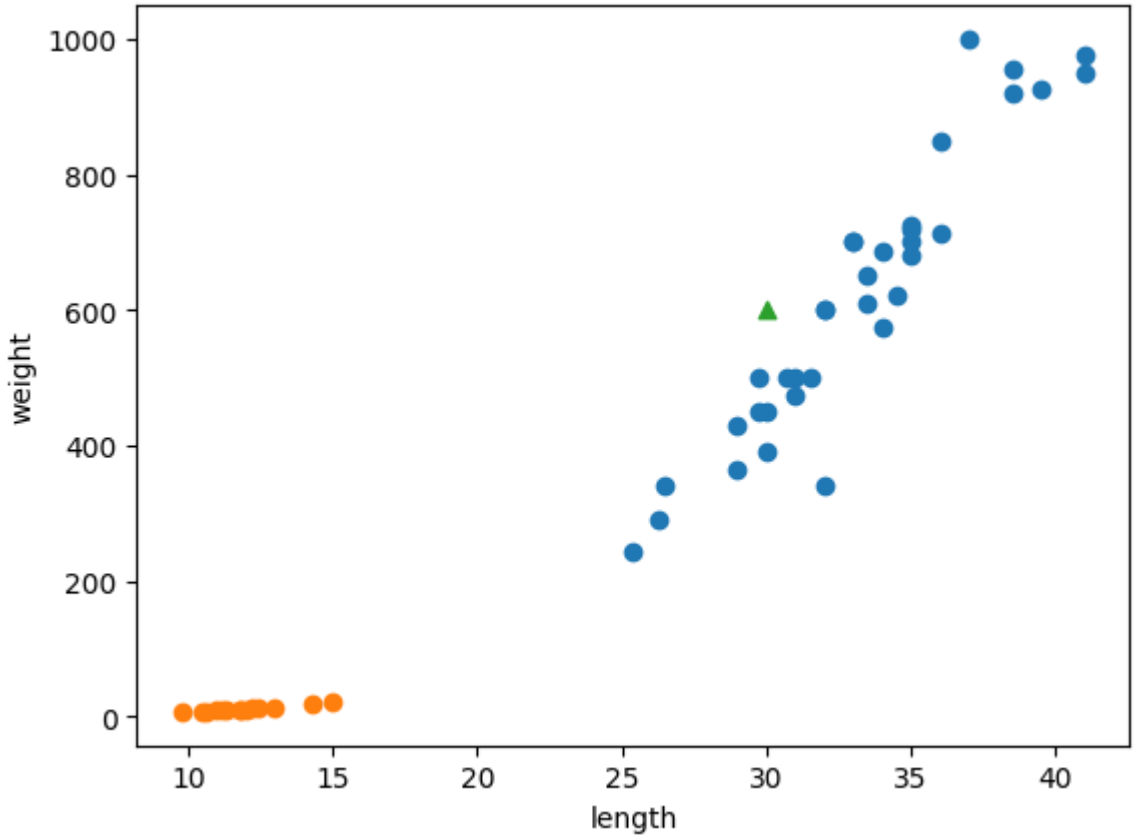
Y 데이터에 선언

```
fish_target = [1]*35 + [0]*14
print(fish_target)
```

KNN 분류 클래스 사용해보기

```
from sklearn.neighbors import KNeighborsClassifier

kn = KNeighborsClassifier()
kn.fit(fish_data, fish_target)
```



길이가 30cm, 무게가 600g인 물고기는?

```
plt.scatter(bream_length, bream_weight)
plt.scatter(smelt_length, smelt_weight)
plt.scatter(30, 600, marker='^')
plt.xlabel('length')
plt.ylabel('weight')
plt.show()
```

길이가 30cm, 무게가 600g인 물고기는?

Knn을 예측하기 위해 필요한 것.

K 개수를 49개로 한다면?

직접 실행해보고, 어떻게 나오는지 확인해보자.