데이터 과학을 위한
Machine Learning & Deep Learning

Made in KYOUNGHEE, LEE

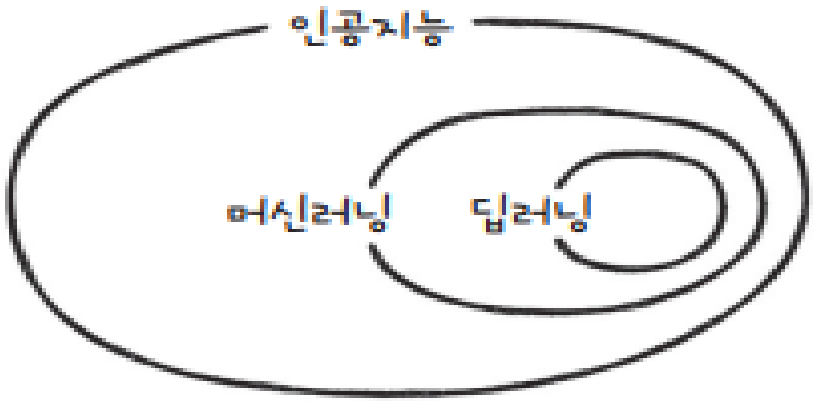
목차

Deep Learning INDEX

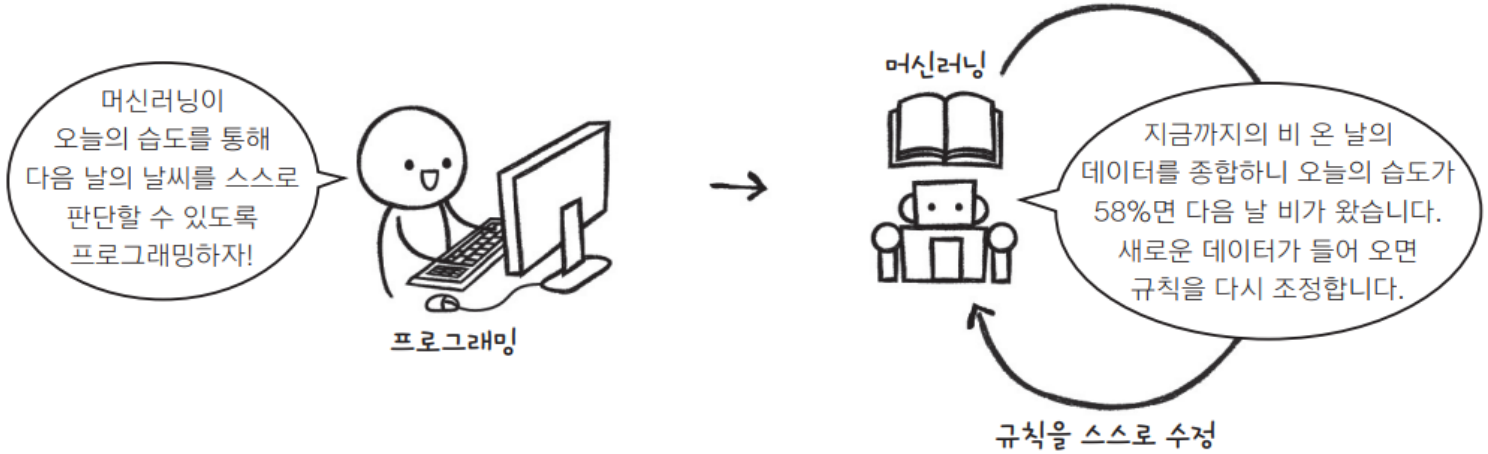
STEP 1
딥러닝(Deep Learning)

- **인공지능(Artificial Intelligence)**
  - 사람의 지능을 만들기 위한 시스템이나 프로그램
  - 강 인공지능(strong AI) : 사람과 구분이 안 될 정도로 강한 성능을 가진 인공지능
  - 약 인공지능(weak AI) : 특정 영역에서 작업을 수행하는 인공지능

- **머신러닝과 딥러닝 그리고 인공지능의 관계**

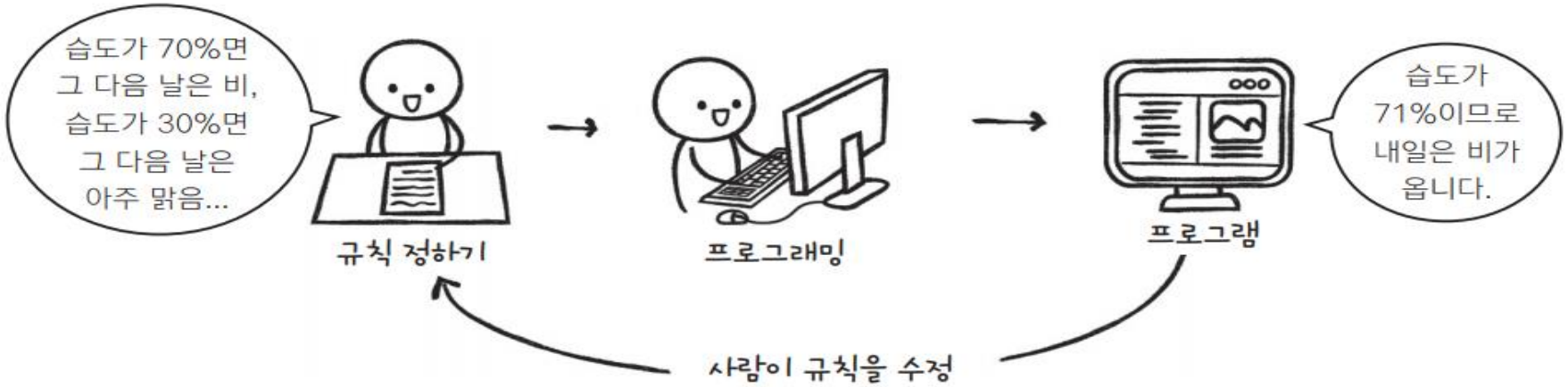


STEP 1
딥러닝(Deep Learning)



- **머신러닝(Machine Learning)**
  - 한글로 풀어 쓰면 '기계 학습' 이다.

- **머신러닝은 스스로 규칙을 수정한다.**
  - 머신러닝과 딥러닝에서 말하는 학습은 데이터의 규칙을 컴퓨터 스스로 찾아내는 것을 말한다.
  - 전통적인 프로그램은 사람이 규칙을 정하여 프로그래밍하고, 사람이 프로그램의 실행 결과를 보며 규칙을 조금씩 수정한다.

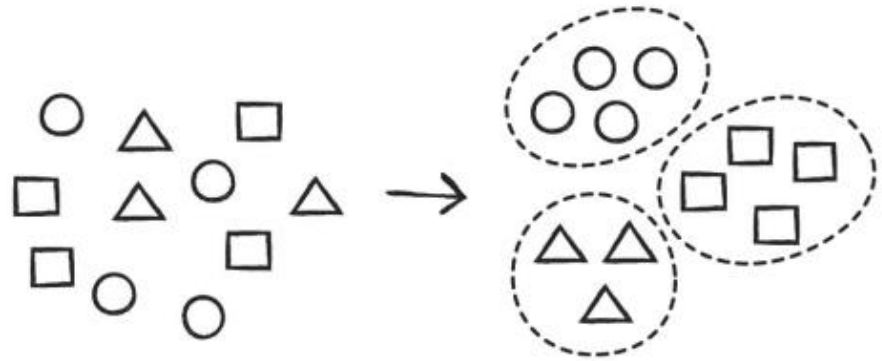


STEP 1
딥러닝(Deep Learning)

- **머신러닝(Machine Learning)의 학습 방식**
  - 지도 학습(supervised learning), 비지도 학습(unsupervised learning), 강화 학습(reinforcement learning)으로 분류
- **지도 학습은 입력과 타깃으로 모델을 훈련시킨다.**

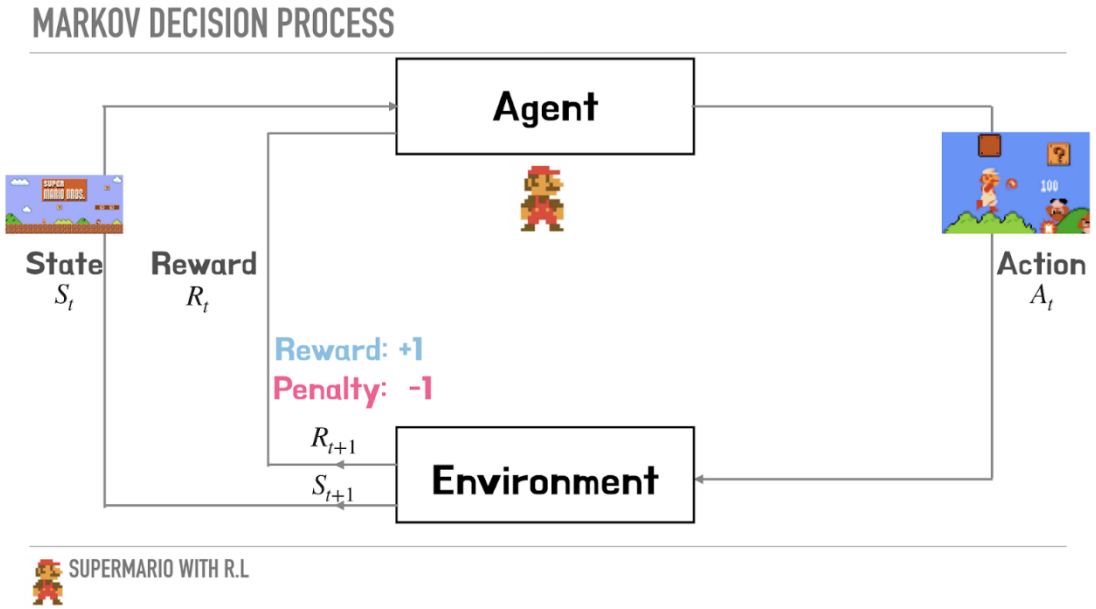
STEP 1
딥러닝(Deep Learning)

- 비지도 학습은 타깃이 없는 데이터를 사용한다.



STEP 1
딥러닝(Deep Learning)

- **강화 학습은 주어진 환경으로부터 피드백을 받아 훈련한다.**
  - 강화 학습은 머신러닝 알고리즘으로 에이전트라는 것을 훈련시킨다.
  - 훈련된 에이전트는 특정 환경에 최적화된 행동을 수행하고 수행에 대한 '보상 ' 과 '현재 상태 ' 를 받는다.
  - 에이전트의 목표는 '최대한 많은 보상을 받는 것 ' 이다.
  - 대표적인 알고리즘 : Q-러닝(Q-learning), SARSA, 인공신경망을 사용한 DQN(Deep Q Network) 등
  - 예 : 딥마인드(DeepMind)의 알파고(AlphaGo)와 같은 게임이나 온라인 광고 등



STEP 1
딥러닝(Deep Learning)

- **지도 학습 유형**

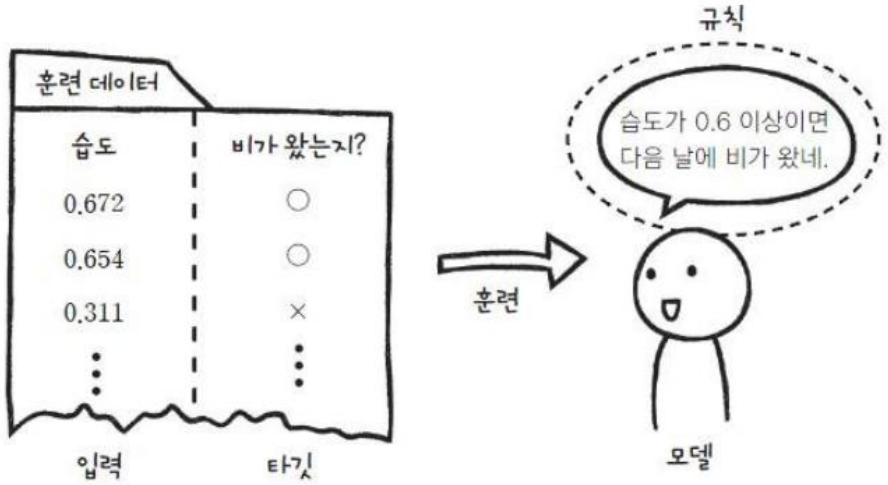
분류	수치예측(혹은 회귀)
- K-최근접 이웃(K-Nearest Neighbors)	- 선형 회귀(Linear Regression)
- 로지스틱 회귀(Logistic Regression)	- 확장된 회귀분석(ex : 다항회귀, 비선형 회귀, 벌점화 회귀 등)
- 인공 신경망 분석(Artificial Neural Network)	- 인공 신경망 분석(Artificial Neural Network)
- 의사결정트리(Decision Tree)	- 의사결정트리(Decision Tree)
- 서포트 벡터 머신(Support Vector Machine)	- 서포트 벡터 머신(회귀) (Support Vector Machine (Regression))
- 나이브 베이즈(Naive Bayes)	- PLS(Partial Least Squares)
- 앙상블 기법(랜덤 포레스트 등)	- 앙상블 기법(랜덤 포레스트 등)

STEP 1
딥러닝(Deep Learning)

- **규칙이란 가중치와 절편을 말한다.**

$$1.5 \times x + 0.1 = y \quad (y\text{가 } 1 \text{ 이상이면 다음 날 비가 온다고 예측})$$

입력, 타깃, 가중치, 절편

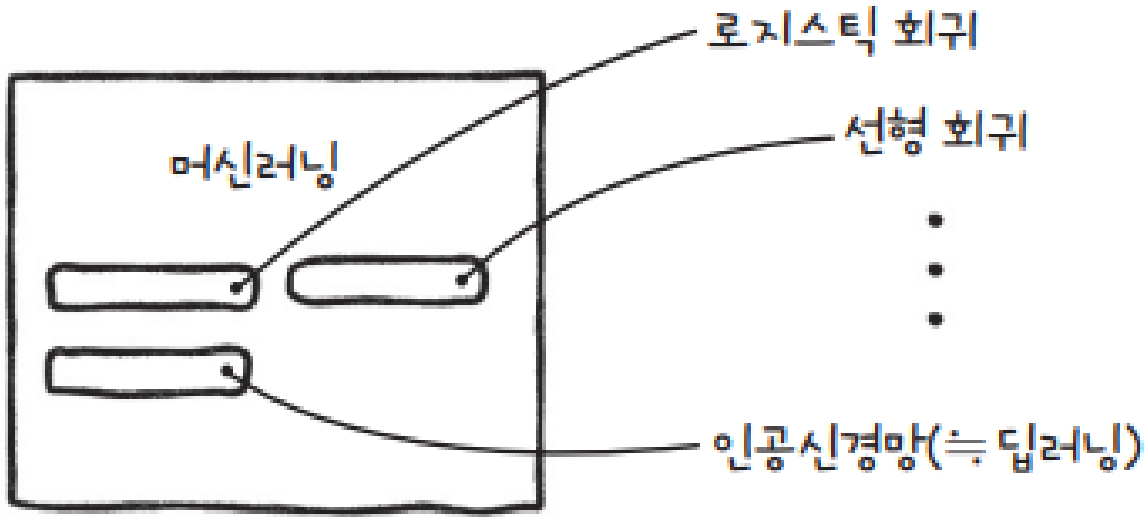


- **모델(Model)은 머신러닝의 수학적 표현이다.**
  - 모델은 훈련데이터로 학습된 머신러닝 알고리즘을 말한다.
  - 가중치와 절편을 합쳐 모델 파라미터(model parameter)라고 부른다.

- **손실 함수로 모델의 규칙을 수정한다.**
  - 모델의 규칙을 수정하는 기준이 되는 함수를 '손실 함수(loss function)'라고 부른다.
  - 손실함수는 모델이 예측한 값과 타깃 값의 차이를 계산하는 함수
  - 최적화 알고리즘으로 손실 함수의 최솟값을 찾는다.

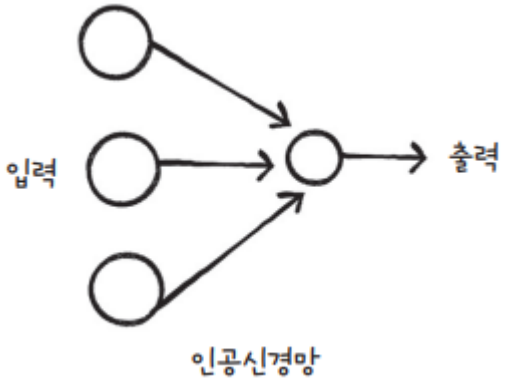
STEP 1
딥러닝(Deep Learning)

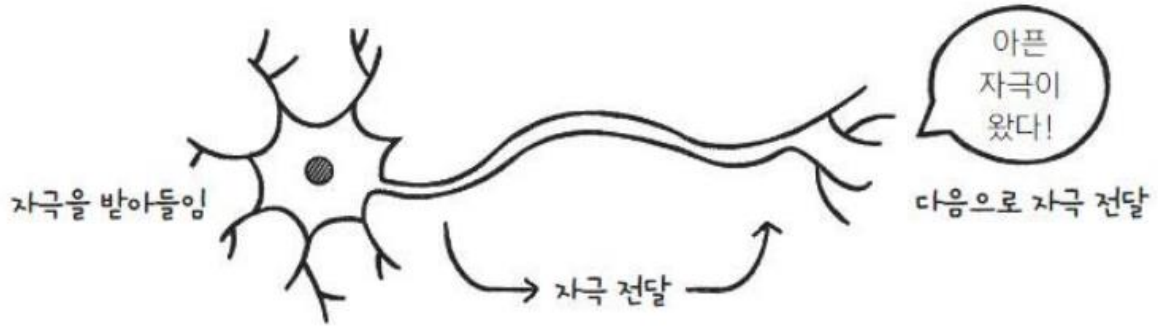
- **딥러닝(Deep Learning)**
  - 머신러닝 알고리즘 중 하나인 인공신경망을 다양하게 쌓은 것이다.
  - 인공신경망을 사용해 만든 것이므로 인공신경망과 딥러닝을 엄밀하게 구분하지 않는다.
  - 복잡한 문제를 해결하기 위해 인공신경망을 다양하게 쌓은 것
  - 머신러닝과 딥러닝은 관계도



STEP 1
딥러닝(Deep Learning)

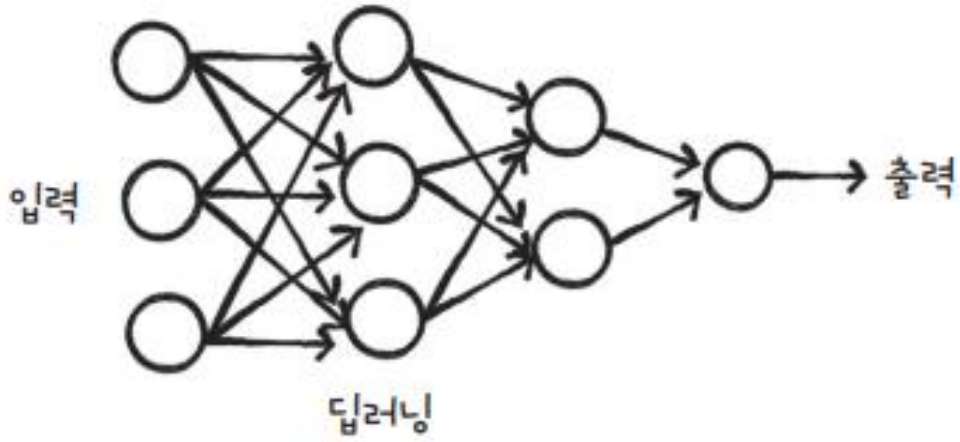
- **딥러닝(Deep Learning)은 머신 러닝 알고리즘 중 하나인 인공신경망(Artificial Neural Network)으로 만든 것**
  - 복잡한 문제를 해결하기 위해 인공신경망을 다양하게 쌓은 것을 딥러닝이라 부른다.





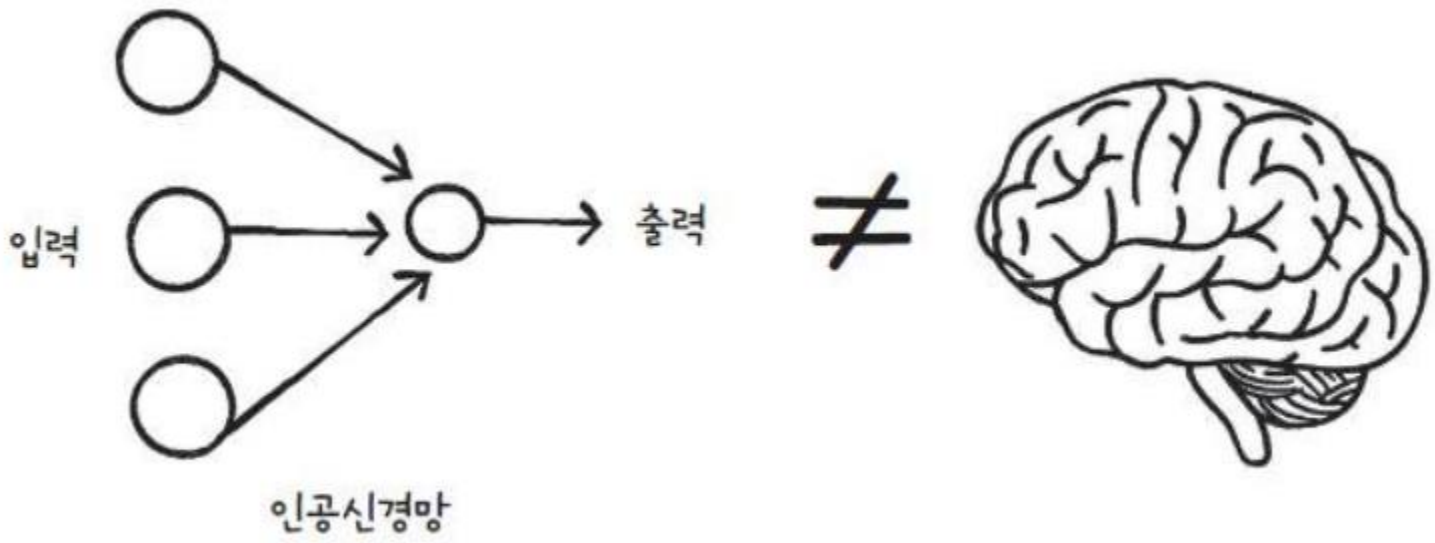
STEP 1
딥러닝(Deep Learning)

- **딥러닝(Deep Learning) 인공신경망으로 구성된다.**
  - 딥러닝이라는 용어는 인공신경망을 여러 겹으로 쌓은 모습에서 유래



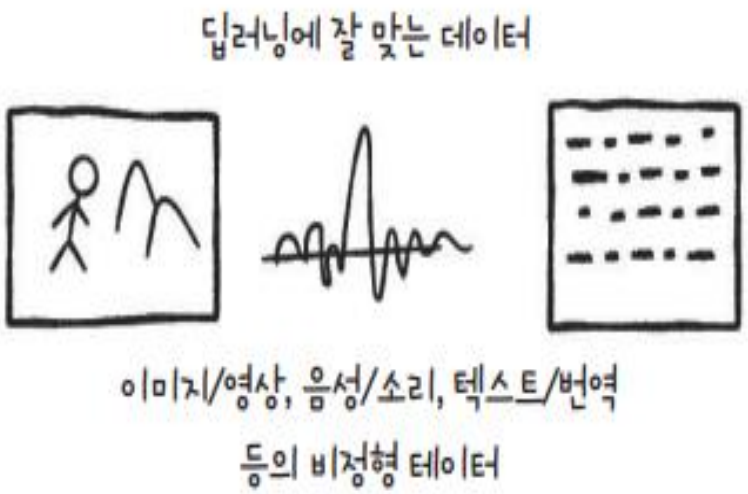
STEP 1
딥러닝(Deep Learning)

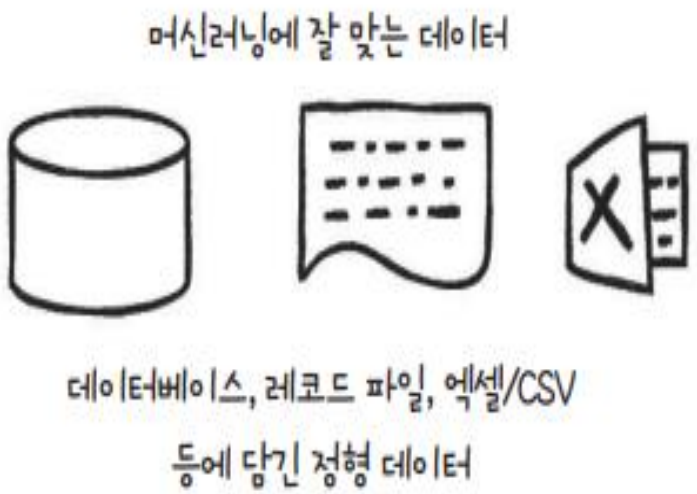
- 딥러닝(Deep Learning)은 사람의 뇌와 많이 다르다.



STEP 1
딥러닝(Deep Learning)

- **딥러닝(Deep Learning) 은 머신 러닝이 처리하기 어려운 데이터를 더 잘 처리한다.**





STEP 1
딥러닝(Deep Learning)

- **딥러닝(Deep Learning) 분야**
  1. 음석 인식(Speech Recognition)
     - Google Assistant
     - Amazon Echo
     - SKT NUGU
  2. 이미지인식(Image Recognition)
     - Google Photo
     - 자율주행 자동차
  3. 자연어처리(Natural Language Processing, NLP)
     - 기계번역(Machine Translation)
     - 챗봇(chatbot)
  4. 게임, 그림, 음악, 로봇, ….

- **손글씨 숫자 인식**
  - MNIST handwritten digits database
  - http://yann.lecun.com/exdb/mnist

STEP 2
수치 예측

- **수치 예측 주요 머신러닝 알고리즘**

종 류	개 념	비 고
회귀분석 (Regresssion Analysis)	관측된 사건들을 정량화해서 독립변수와 종속변수의 관계를 함수식으로 설명하는 방법. 해당 함수식이 모수에 대해 선형일 경우 선형회귀분석이라고 하며, 독립변수가 한 개인 경우 단순선형회귀, 여러 개일 경우 다중선형회귀분석을 적용함	추론통계기반 모형
의사결정트리 (Decision Tree)	목표변수와 가장 연관성이 높은 변수의 순서대로 나무 형태로 가지를 분할하면서 규칙을 만들어내지만, 분류목적의 의사결정트리가 지니계수나 엔트로피를 사용하는 것과는 달리 수치예측 목적일 때는 분산(혹은 표준편차)의 감소량(Variance Reduction)을 최대화하는 기준의 최적분리에 의해 회귀나무를 형성하게 됨	분할 정복기법 (Divide & Conquer)
인공 신경망 분석 (Artificial Neural Network)	인간의 뇌의 뉴런 작용 형태에서 모티브를 얻은 기법으로서, 입력 노드와 은닉 노드, 출력 노드를 구성하여 복잡한 수치예측 문제를 해결할 수 있도록 하는 분석 기법	블랙박스기법
서포트 벡터 머신 (Support Vector Machine, 혹은 Support Vector Regression)	분류문제에서는 서로 다른 분류에 속한 데이터 간의 간격(마진)을 최대화하는 초평면을 찾는 것이라면 수치예측문제에서의 서포트 벡터 머신은 데이터 점들의 분류가 아닌, 데이터 점들을 잘 적합할 수 있도록 가장 많은 데이터 점을 포함하는 튜브를 찾음	선형 및 비선형 (커널트릭)
랜덤 포레스트 (Random Forest)	주어진 데이터로부터 여러 개의 다양한 의사결정트리를 만들어 각 의사결정트리의 예측결과를 평균 내고 그 평균값을 최종결과로 결정하는 앙상블 형태의 기법(분류문제에서는 각 예측 분류결과를 투표하여 과반수 이상인 분류결과를 최종결과로 도출함)	앙상블 모형

STEP 2
수치 예측

- **선형 회귀(Linear Regression) 분석이란?**
  - 독립변수가 종속변수에 영향을 미치는지 알아보고자 할 때 실시하는 분석방법
  - 단순선형회귀분석은 독립변수 X(설명변수)에 대하여 종속변수 Y(반응변수)들 사이의 관계를 수학적 모형을 이용하여 규명하는 것
  - 규명된 함수식을 이용하여 설명변수들의 변화로부터 종속변수의 변화를 예측하는 분석이다.

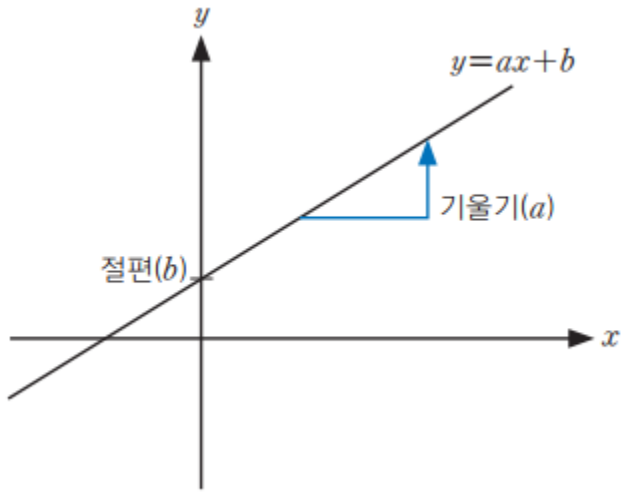
선형 회귀는 아주 간단한 1차 함수로 표현할 수 있다.

선형 회귀의 선형이라는 단어의 의미는 다음 수식을 통해 그려지는 직선 그래프를 보면 쉽게 이해할 수 있다.

$$y = ax + b$$

위 1차 함수의 기울기(slope)는 a이고 절편(intercept)은 b이다.

보통 이런 1차 함수는 2차원 평면에 그리기 쉽다.

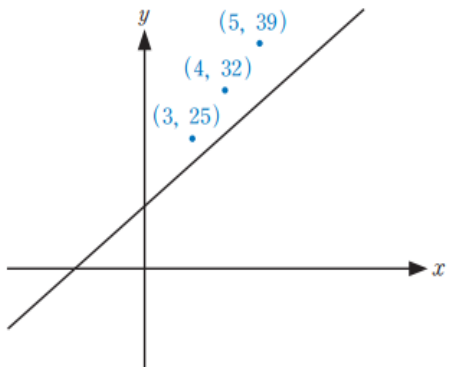


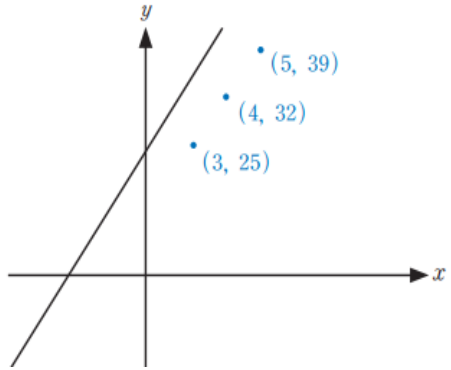
STEP 2
수치 예측

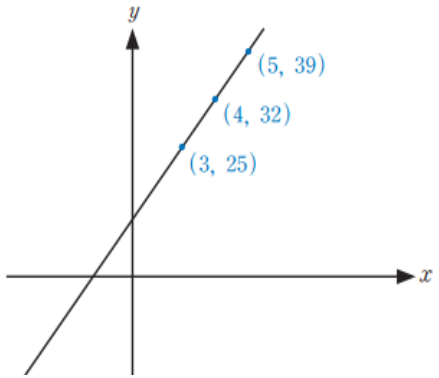
- **선형 회귀는 기울기와 절편을 찾아준다.**
  보통 1차 함수는 x에 따른 y의 값에 하지만 선형 회귀에서는 이와 반대로 x, y가 주어졌을 때 기울기와 절편을 찾는 데 집중한다. 즉, 선형 회귀의 주요 관심사는 절편과 기울기를 찾는 것이다.

문제1. 기울기가 7이고, 절편이 4인 1차 함수 $y = 7x + 4$가 있다. x가 10이면 y는 얼마일까?

문제2. x가 3일 때 y는 25, x가 4일 때 y는 32, x가 5일 때 y는 39라면 기울기와 절편의 값으로 적절한 것은 무엇인가?







STEP 2

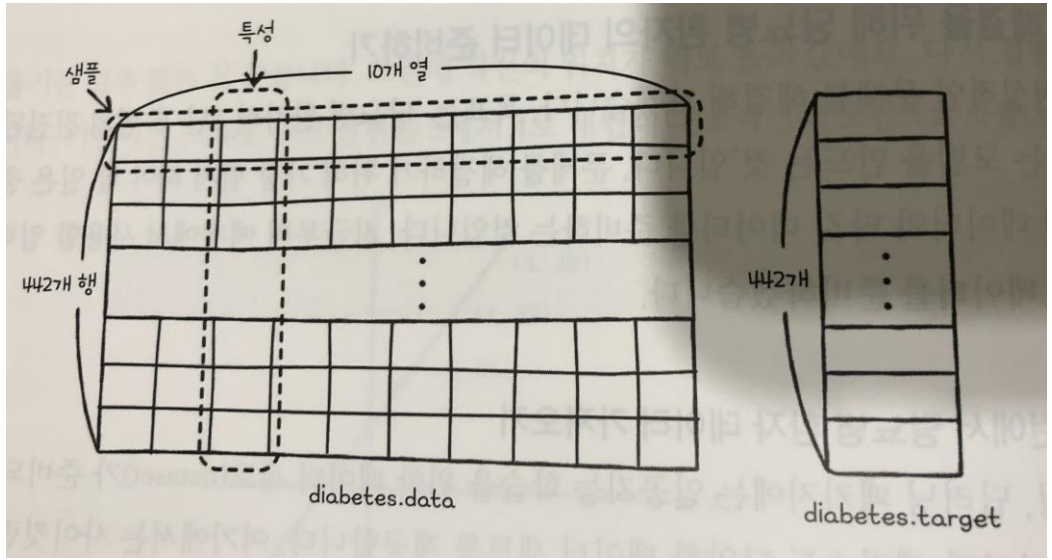
당뇨병 환자 데이터셋(diabetes)

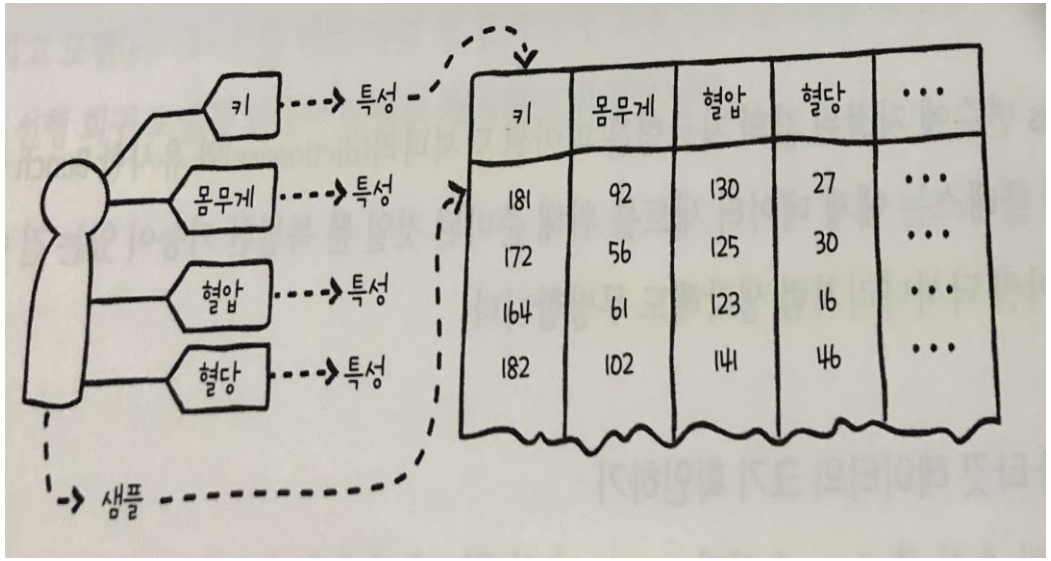
1. load_diabetes( ) 함수로 당뇨병 데이터 준비하기

```
from sklearn.datasets import load_diabetes
diabetes = load_diabetes()
```

2. 입력과 타깃 데이터 크기 확인하기

```
print(disbetes.data.shape, diabetes.target.shape)
```





STEP 2
당뇨병 환자 데이터셋(diabetes)

3. 입력 데이터 자세히 보기

```
diabetes.data[0:3]
array([[ 0.03807591,  0.05068012,  0.06169621,  0.02187235, -0.0442235 ,
        -0.03482076, -0.04340085, -0.00259226,  0.01990842, -0.01764613],
       [-0.00188202, -0.04464164, -0.05147406, -0.02632783, -0.00844872,
        -0.01916334,  0.07441156, -0.03949338, -0.06832974, -0.09220405],
       [ 0.08529891,  0.05068012,  0.04445121, -0.00567061, -0.04559945,
        -0.03419447, -0.03235593, -0.00259226,  0.00286377, -0.02593034]])
```

네 번째 특성의 값입니다.

첫 번째 샘플입니다.

4. 타깃 데이터 자세히 보기

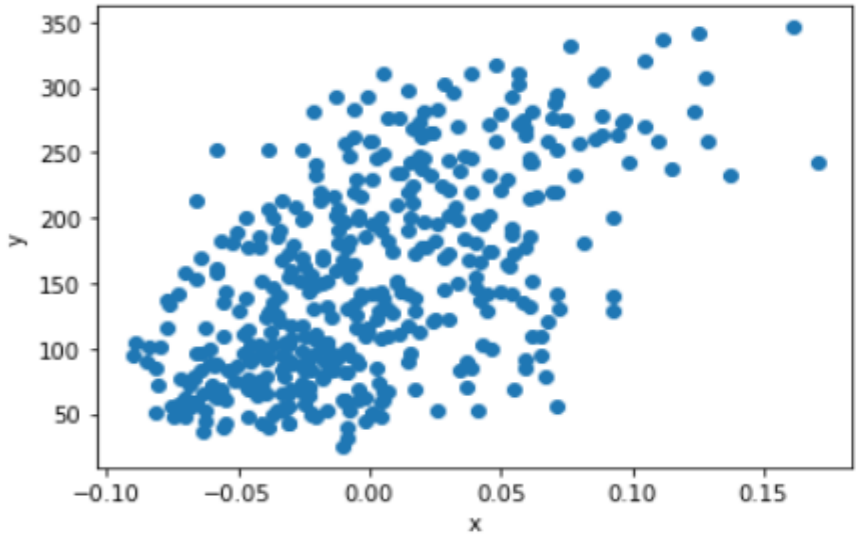
```
In [5]: 1 diabetes.target[:3]
Out[5]: array([151.,  75., 141.])
```

STEP 2

당뇨병 환자 데이터셋(diabetes)

5. 당뇨병 환자 데이터 시각화
   - 세 번째 특성과 타깃 데이터로 산점도 그래프

```
1 plt.scatter(diabetes.data[:, 2], diabetes.target)
2 plt.xlabel('x')
3 plt.ylabel('y')
4 plt.show()
```



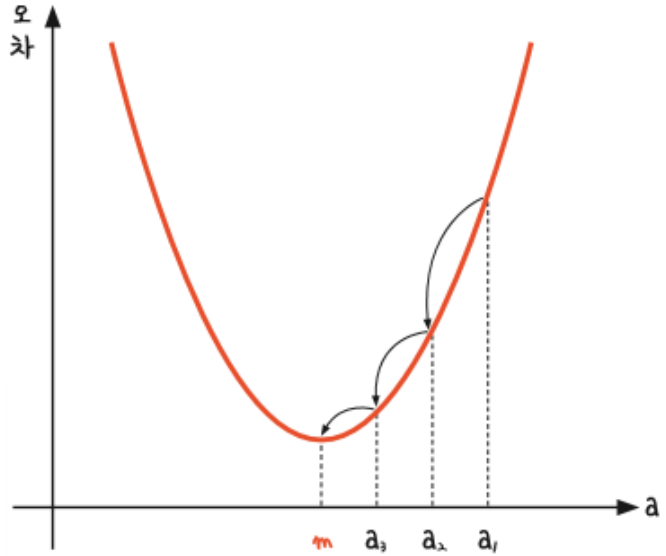
STEP 2
선형 회귀 분석(Linear Regression Analysis)

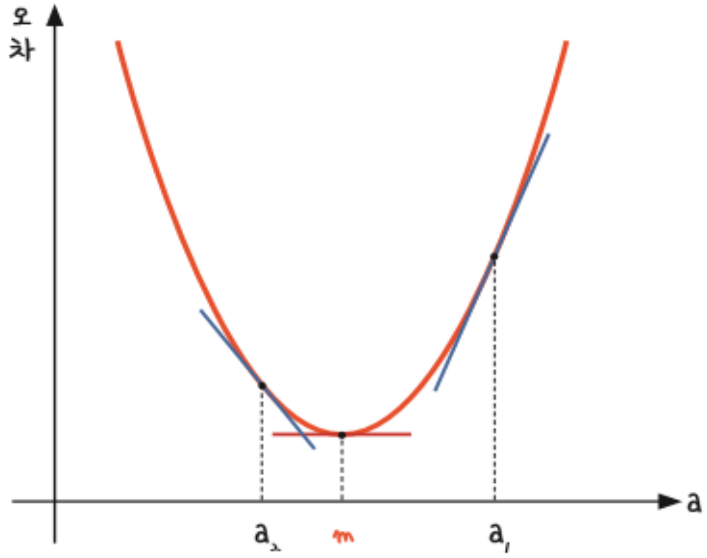
6. 훈련 데이터 준비하기

```
1 x = diabetes.data[:, 2]
2 y = diabetes.target
```

STEP 2
경사 하강법(Gradient Descent)

- **경사 하강법**
  - 미분 기울기를 이용해 오차를 비교하여 가장 작은 방향으로 이동시키는 방법

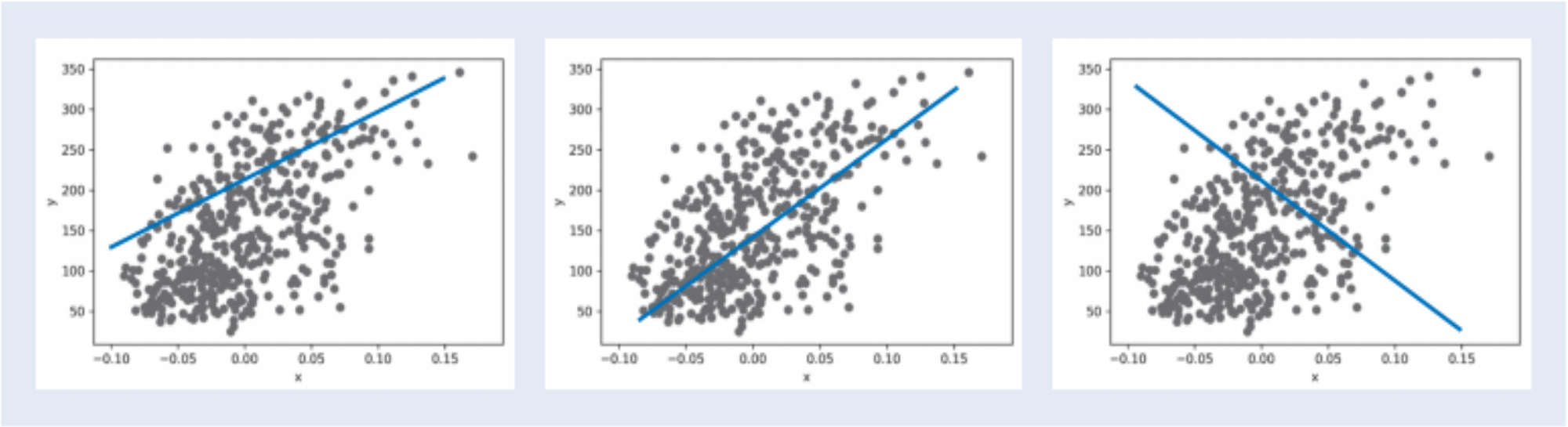




STEP 2

경사 하강법(Gradient Descent)

- **선형 회귀와 경사 하강법의 관계 이해**



경사 하강법은 모델이 데이터를 잘 표현할 수 있도록 기울기(변화율)를 사용하여 모델을 조금씩 조정하는 최적화 알고리즘

STEP 2
경사 하강법(Gradient Descent)

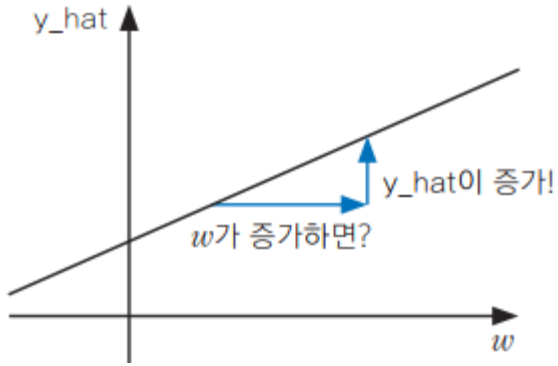
- **예측값과 변화율**
  - 앞으로는 $y = ax + b$로 알고 있던 모델을 $\hat{y} = wx + b$로 이해하기
  - 여기서 가중치 w와 절편 b는 알고리즘이 찾은 규칙을 의미하고, $\hat{y}$ 은 우리가 예측한 값(예측값)을 의미
  - $\hat{y}$는 와이-햇(y-hat)이라고 읽는다.

- **예측값으로 올바른 모델 찾기** : 훈련 데이터에 잘 맞는 w와 b를 찾는 방법

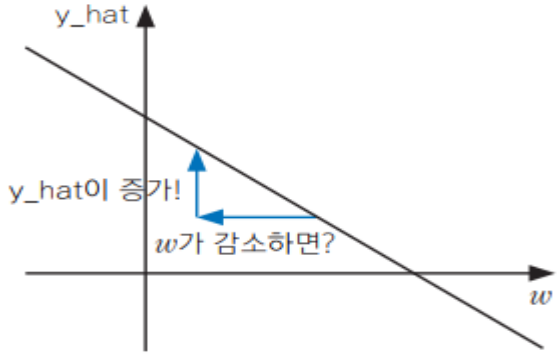
① 무작위로 w와 b를 정합니다(무작위로 모델 만들기).
② x에서 샘플 하나를 선택하여 $\hat{y}$ 을 계산합니다(무작위로 모델 예측하기).
③ $\hat{y}$ 과 선택한 샘플의 진짜 y를 비교합니다(예측한 값과 진짜 정답 비교하기, 틀릴 확률 99%).
④ $\hat{y}$ 이 y와 더 가까워지도록 w, b를 조정합니다(모델 조정하기).
⑤ 모든 샘플을 처리할 때까지 다시 ②~④ 항목을 반복합니다.

STEP 2
경사 하강법(Gradient Descent)

- **변화율로 가중치 업데이트하기**
  - 변화율이 양수일 때 가중치를 업데이트 하는 방법



  - 변화율이 음수일 때 가중치를 업데이트 하는 방법



STEP 2
경사 하강법(Gradient Descent)

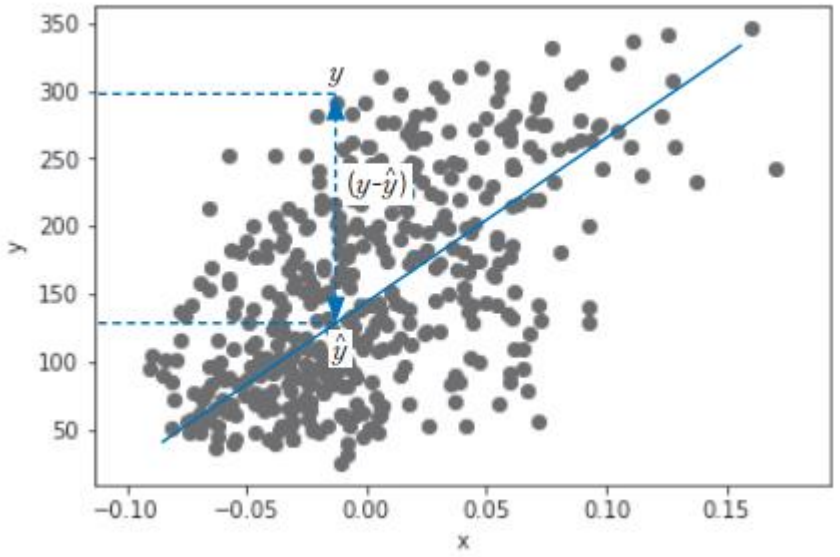
- **변화율로 절편 업데이트하기**
- **오차 역전파로 가중치와 절편을 더 적절하게 업데이트하기**
  - 오차 역전파(backpropagation)는 $\hat{y}$ 과 y의 차이를 이용하여 w와 b를 업데이트한다.
  - 오차 역전파라는 이름에서 알 수 있듯이 이 방법은 오차가 연이어 전파되는 모습으로 수행된다.

STEP 2
손실 함수(Loss Function)

- **손실 함수와 경사 하강법의 관계**
  - 경사 하강법을 기술적으로 표현하면, “어떤 손실 함수가 정의되었을 때 손실 함수의 값이 최소가 되는 지점을 찾아가는 방법 " 이다.
  - 손실 함수란 예상한 값과 실제 타깃값의 차이를 함수로 정의한 것을 말한다.
  - ‘오차를 변화율에 곱하여 가중치와 절편 업데이트하기 ' 는 제곱 오차라는 손실 함수를 미분한 것과 같다.

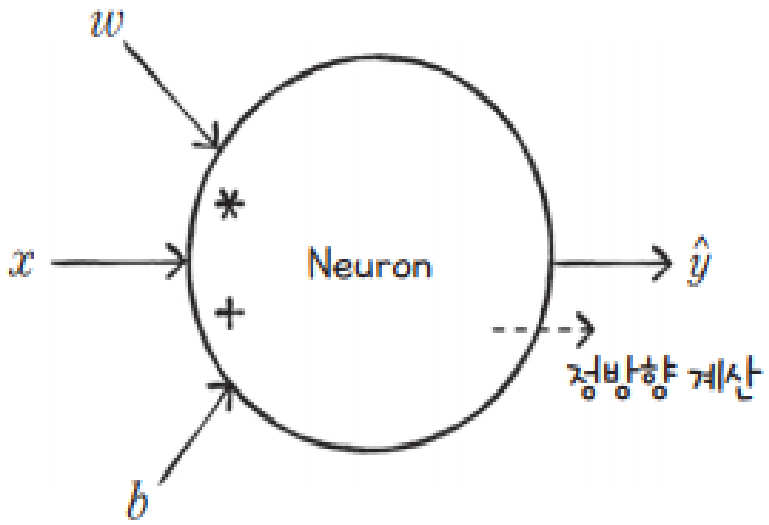
- **제곱 오차(Squared Error)**
  - 타깃값과 예측값을 뺀 다음 제곱한 것
  - 제곱 오차를 수식으로 나타내면

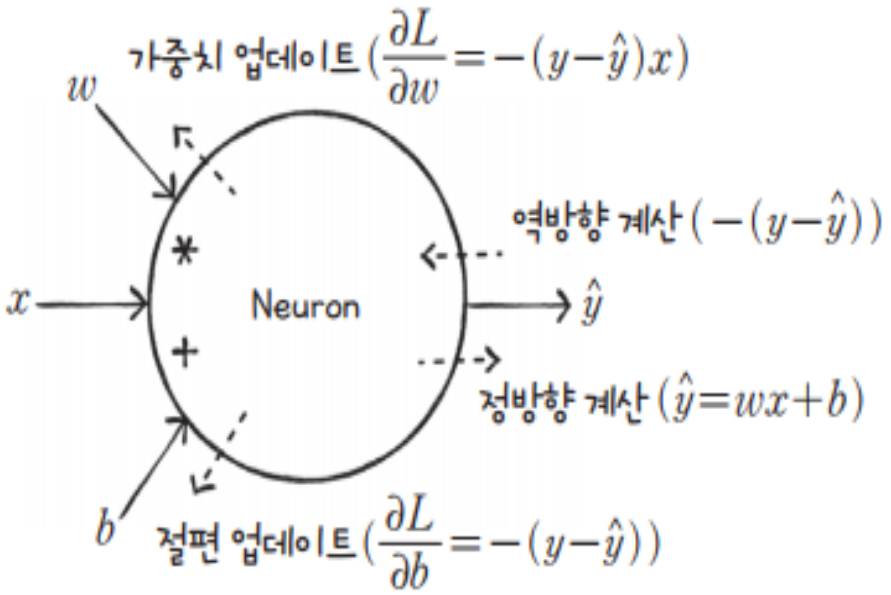
$$SE = (y - \hat{y})^2$$



STEP 2

경사 하강법 알고리즘을 Neuron이라는 이름의 파이썬 클래스 구현





STEP 2

경사 하강법 알고리즘을 Neuron이라는 이름의 파이썬 클래스 구현

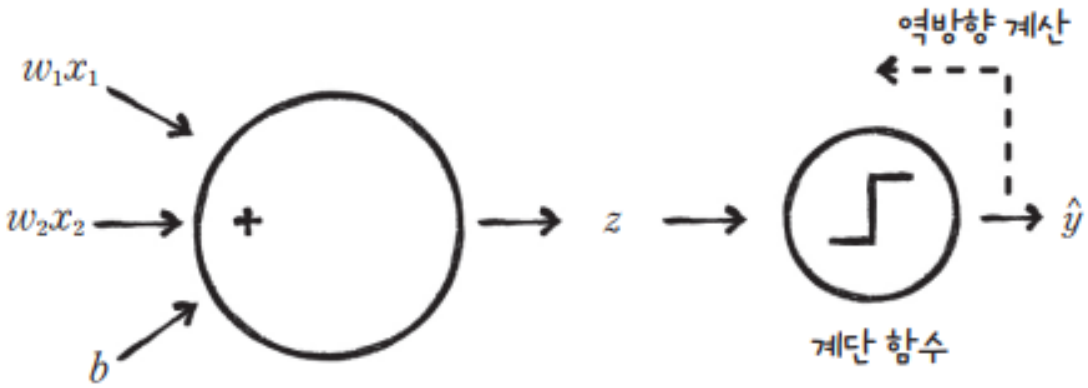
```
class Neuron:

        def __init__(self):
            self.w = 1.0
            self.b = 1.0

        def forpass(self, x):
            y_hat = x * self.w + self.b
            return y_hat

        def backprob(self, x, err):
            w_gred = x * err
            b_gred = 1 * err
            return w_gred, b_gred

        def fit(self, x, y, epochs=100):
            for I in range(epochs):
                 for x_i, y_i in zip(x, y):
                     y_hat = self.forpass(x_i)
                     err = -(y_i - y_hat)
                     w_gred, b_gred = self.backprob(x_i, err)
                     self.w -= w_gred
                     self.b -= b_gred
```

STEP 3

이진 분류(Binary Classification)

- **퍼셉트론 (Perceptron) 알고리즘**
  - 1957년 코넬 항공 연구소(Cornell Aeronautical Lab)의 프랑크 로젠블라트(Frank Rosenblatt) 발표
  - 이진 분류(binary classification)란 임의의 샘플 데이터를 True나 False로 구분하는 문제를 말한다.
  - 예를 들어 과일이라는 샘플 데이터가 있을 때 사과인지(True), 아닌지(False)를 판단하는 것이 이진 분류에 해당

- **퍼셉트론 (Perceptron) 전체 구조**

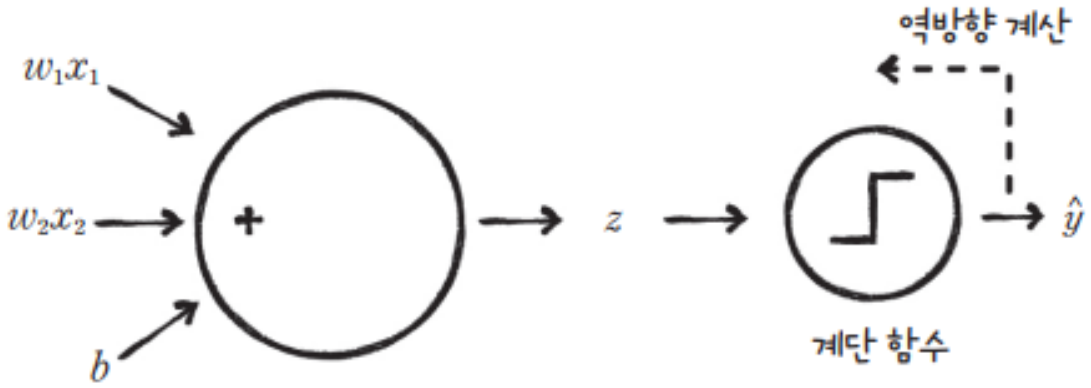


$$w_1x_1 + w_2x_2 + b = z$$

STEP 3

이진 분류(Binary Classification)

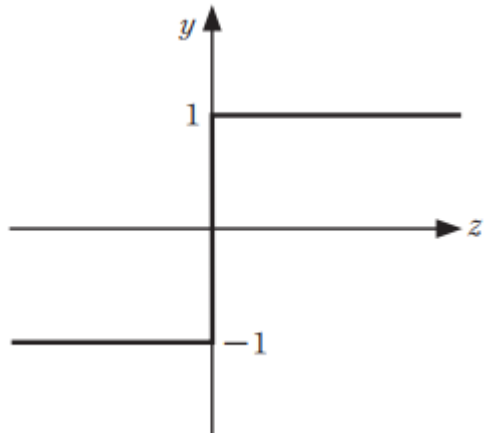
- **퍼셉트론 (Perceptron) 전체 구조**



$$w_1x_1+w_2x_2+b=z$$

$$y=\begin{cases} 1 & (z>0) \\ -1 \end{cases}$$

양성 클래스(positive class)

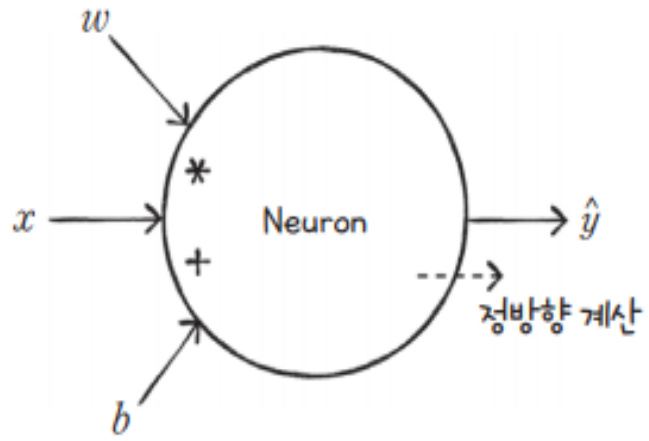


음성 클래스(negative class)

STEP 3

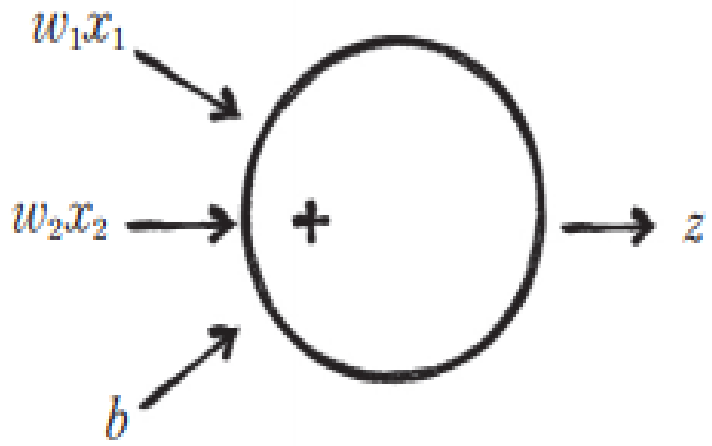
이진 분류(Binary Classification)

여러 개의 특성을 표현하는 방법



$$z = w_1 x_1 + w_2 x_2 + b$$

1번째 특성의 가중치와 입력

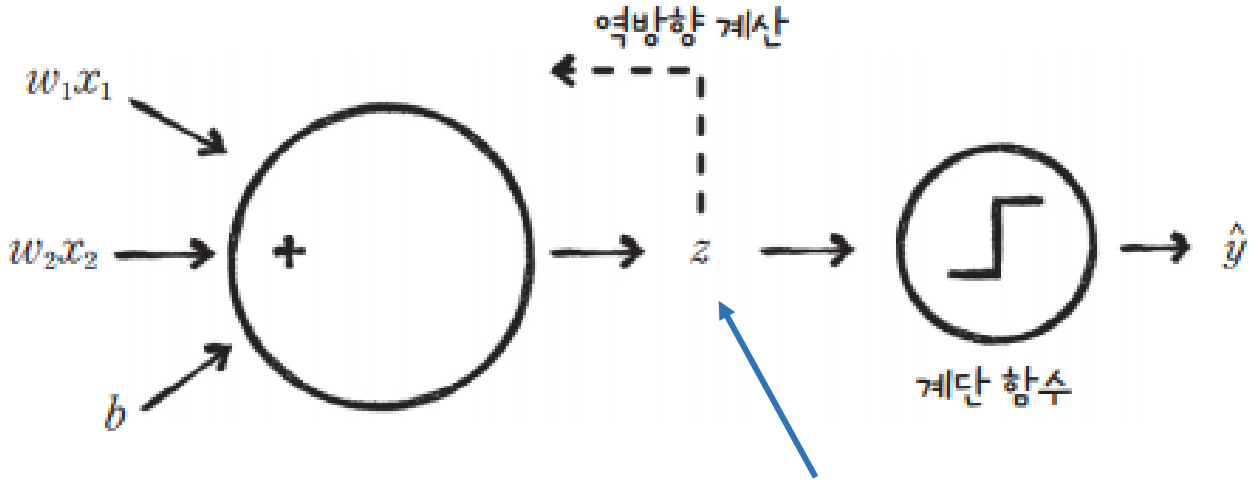


$$z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

$$z = b + \sum_{i=1}^{n} w_i x_i$$

STEP 3
이진 분류(Binary Classification)

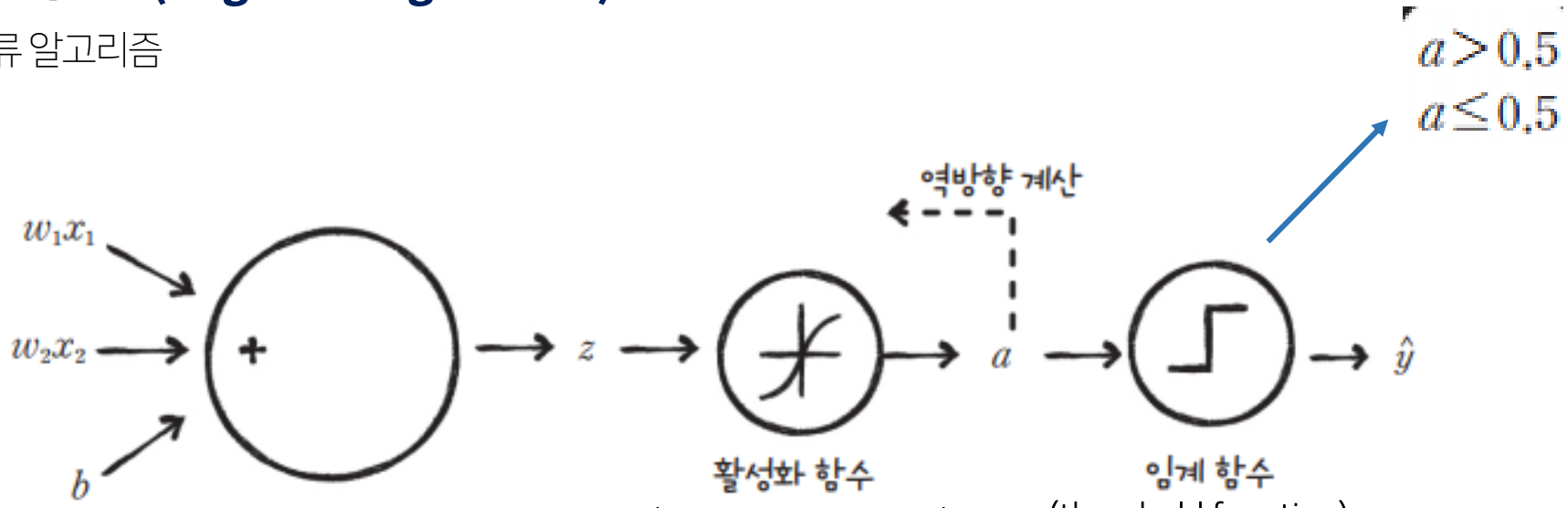
- **아달린(Adaline) 알고리즘**
  - 1960년 스탠포드 대학의 버나드 위드로우(Bernard Widrow)와 테드 호프(Tedd Hoff)가 퍼셉트론을 개선한 적응형 선형 누런(Adaptive Linear Neuron) 발표



역방향 계산이 일어나는 위치가 퍼셉트론과 다름

STEP 3
이진 분류(Binary Classification)

- **로지스틱 회귀(Logistic Regression) 알고리즘**
  - 분류 알고리즘



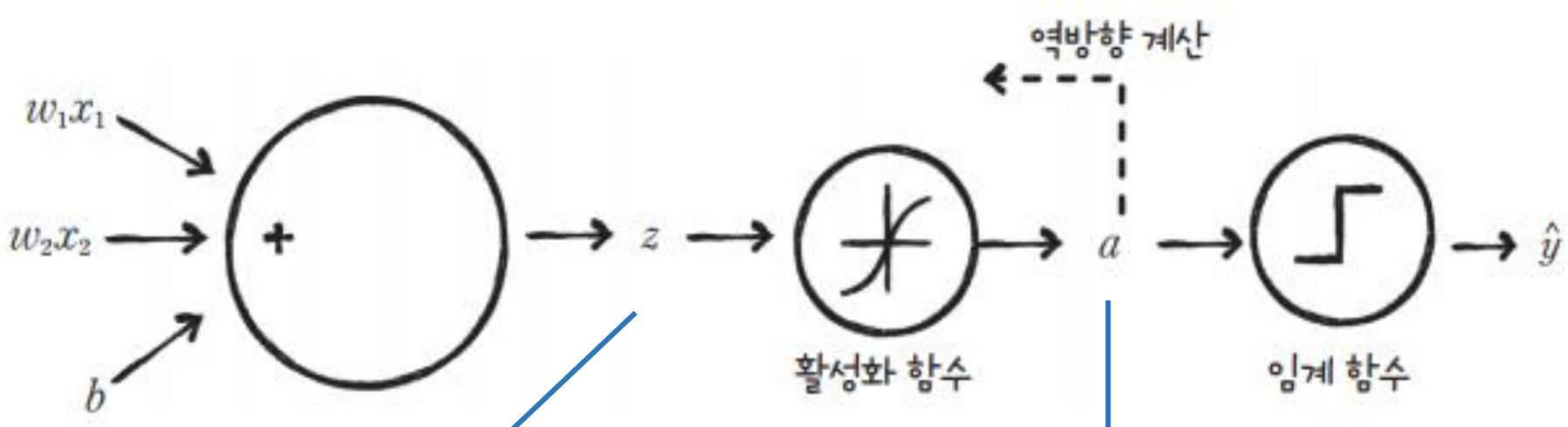
시그모이드 함수
(왜 비선형 함수를 사용할까?)

$$p = \frac{1}{1+e^{-z}}$$

STEP 3

이진 분류(Binary Classification)

- **시그모이드 함수로 확률을 만든다**



$$z = b + \sum_{i=1}^{n} w_i x_i$$

0 ~ 1
a > 0.5 : 양성 클래스
a <= 0.5 : 음성 클래스

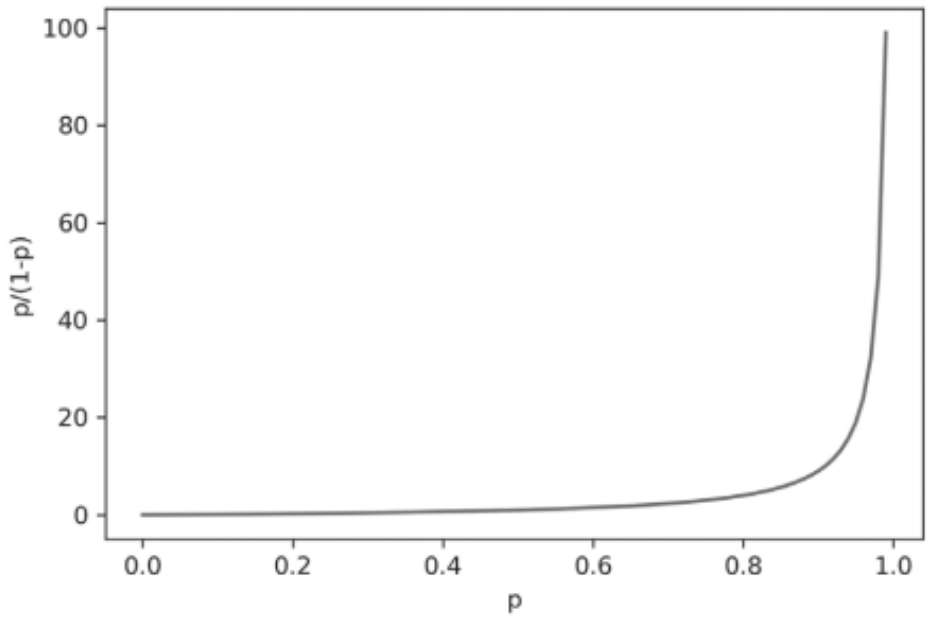
STEP 3

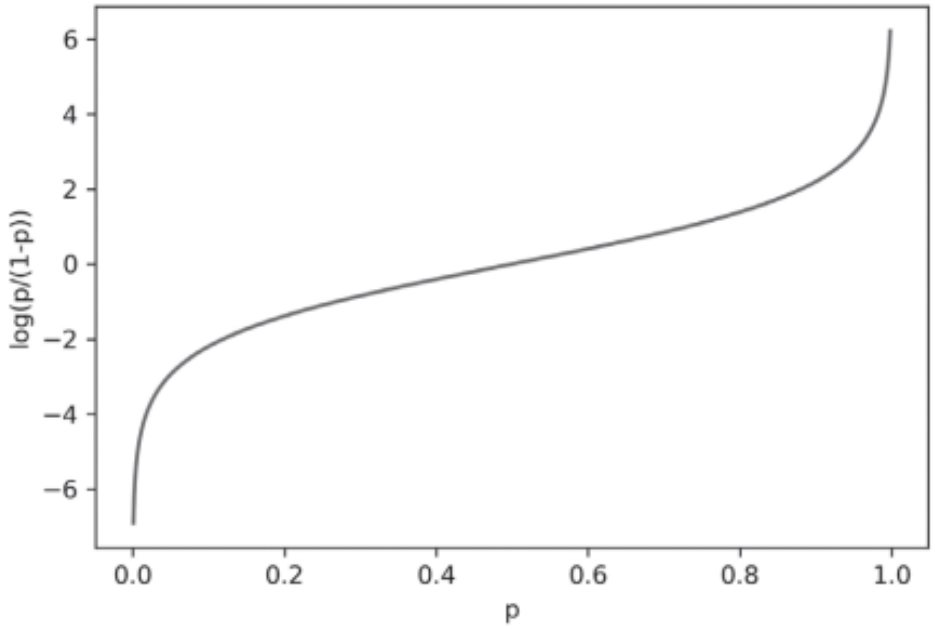
이진 분류(Binary Classification)

- **시그모이드 함수가 만들어지는 과정**
  - 오즈 비(odds ratio) > 로짓 함수(logit function) > 시그모이드 함수

$$OR(odds\ ratio)=\frac{p}{1-p}(p=\text{성공 확률})$$

$$logit(p)=log(\frac{p}{1-p})$$





STEP 3
이진 분류(Binary Classification)

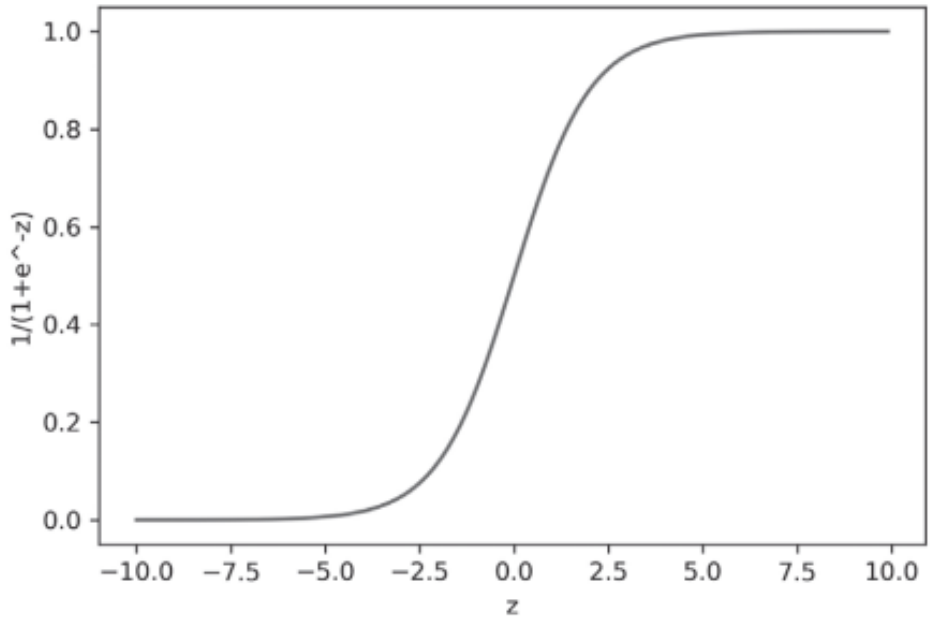
- **로지스틱 함수**
  - 시그모이드(sigmoid) 함수라고도 부른다.

$$log\left(\frac{p}{1-p}\right) = z$$

$$\frac{P}{1-p} = \mathrm{e}^{z}$$

$$p(1+\mathrm{e}^{z}) = \mathrm{e}^{z}$$

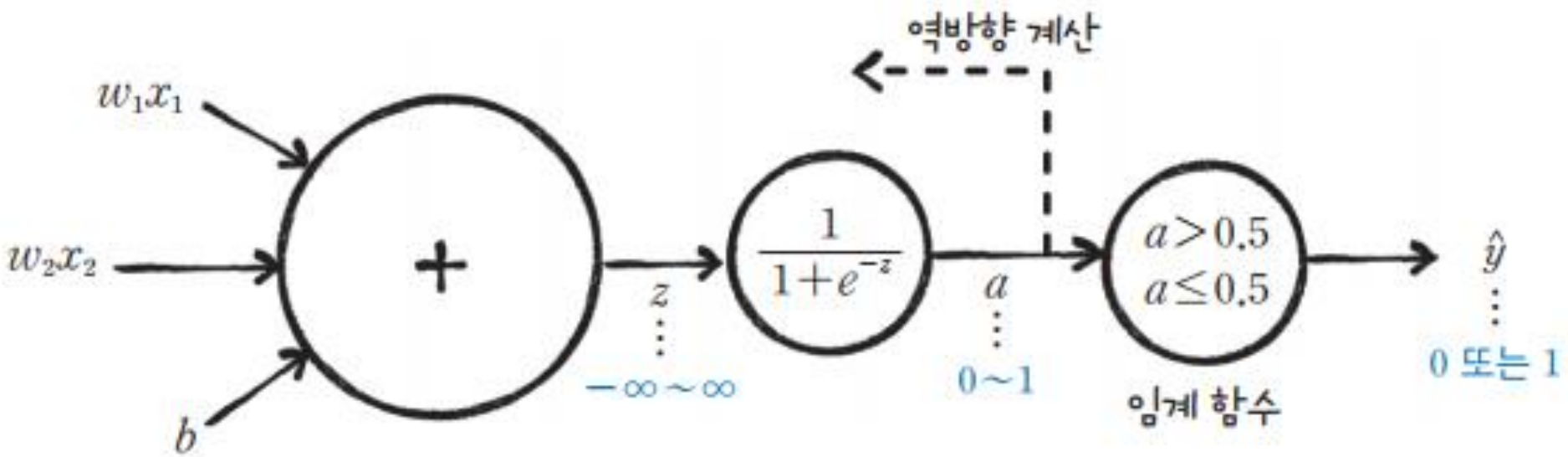
$$p = \frac{\mathrm{e}^{z}}{1+\mathrm{e}^{z}} = \frac{1}{1+e^{-z}}$$



STEP 3

이진 분류(Binary Classification)

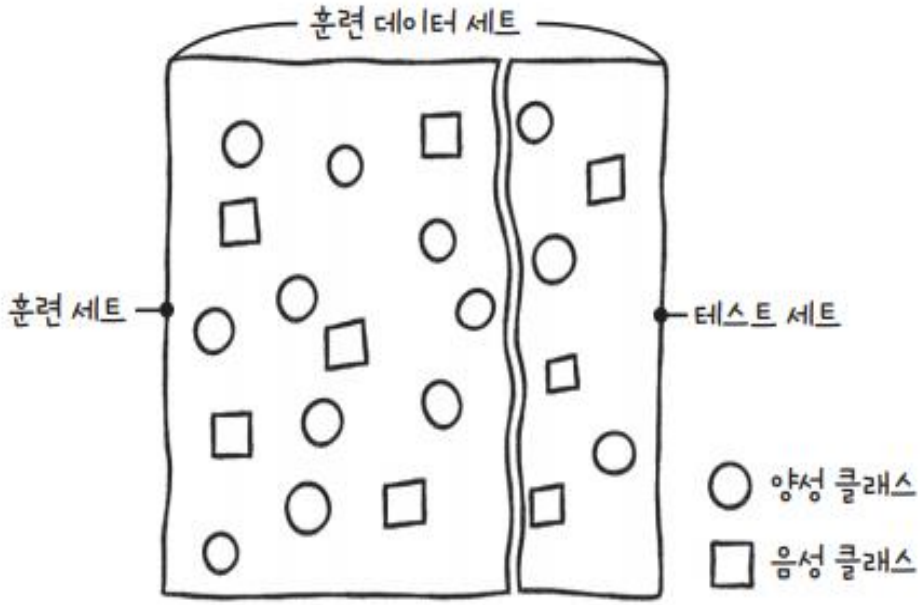
- 로지스틱 회귀 중간 정리

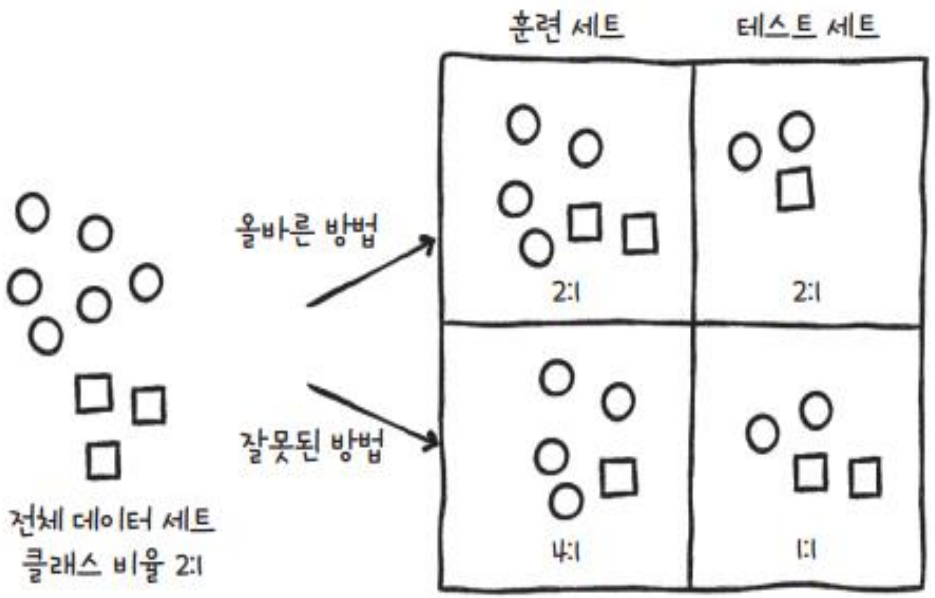


STEP 3

이진 분류(Binary Classification)

- **로지스틱 손실 함수를 경사 하강법에 적용**





STEP 3
이진 분류(Binary Classification)

- **로지스틱 손실 함수를 경사 하강법에 적용**
  - 로지스틱 회귀와 같은 분류의 목표는 올바르게 분류된 샘플 데이터의 비율 자체를 높이는 것이 목표이다.
  - **로지스틱 손실 함수** : 다중 분류를 위한 손실 함수인 크로스 엔트로피(cross entropy) 손실 함수를 이진 분류 버전으로 만든 것

$$L = -(y\log(a) + (1-y)\log(1-a))$$

a : 활성화 함수 출력 값
y : 타깃 값

	L
y가 1인 경우(양성 클래스)	$-\log(a)$
y가 0인 경우(음성 클래스)	$-\log(1-a)$

STEP 3
이진 분류(Binary Classification)

- **로지스틱 손실 함수 미분하기**
  - 가중치와 절편에 대한 로지스틱 손실 함수의 미분 결과

$$\frac{\partial}{\partial w_i}L=-(y-a)x_i$$

$$\frac{\partial}{\partial b}L=-(y-a)1$$

	제곱 오차의 미분	로지스틱 손실 함수의 미분
가중치에 대한 미분	$\frac{\partial SE}{\partial w}=-(y-\hat{y})x$	$\frac{\partial}{\partial w_i}L=-(y-a)x_i$
절편에 대한 미분	$\frac{\partial SE}{\partial b}=-(y-\hat{y})1$	$\frac{\partial}{\partial b}L=-(y-a)1$

STEP 3
이진 분류(Binary Classification)

- **로지스틱 손실 함수 와 연쇄 법칙(Chain Rule)**
  - 연쇄 법칙(Chain Rule) : 미분에서 합성 함수의 도함수(미분한 함수)를 구하기 위한 방법

$$y=f(u),\ u=g(x) \qquad y=f(g(x)) \qquad \frac{\partial y}{\partial x}=\frac{\partial y}{\partial u}\frac{\partial u}{\partial x}$$

$$\frac{\partial L}{\partial a}=\frac{\partial}{\partial a}(-(y\log(a)+(1-y)\log(1-a)))$$

$$=-(y\frac{\partial}{\partial a}\log(a)+(1-y)\frac{\partial}{\partial a}\log(1-a))$$

$$\frac{\partial L}{\partial a}=-(y\frac{1}{a}-(1-y)\frac{1}{1-a})$$

$$\frac{\partial a}{\partial z}=\frac{\partial}{\partial z}(\frac{1}{1+e^{-z}})=\frac{\partial}{\partial z}(1+e^{-z})^{-1}$$

$$=-(1+e^{-z})^{-2}\frac{\partial}{\partial z}(e^{-z})=-(1+e^{-z})^{-2}(-e^{-z})=\frac{e^{-z}}{(1+e^{-z})^2}$$

$$\frac{\partial a}{\partial z}=\frac{1}{1+e^{-z}}\frac{e^{-z}}{1+e^{-z}}=\frac{1}{1+e^{-z}}(1-\frac{1}{1+e^{-z}})=a(1-a)$$

$$\frac{\partial a}{\partial z}=a(1-a)$$

STEP 3
이진 분류(Binary Classification)

- **로지스틱 손실 함수 와 연쇄 법칙(Chain Rule)**
  - 연쇄 법칙(Chain Rule) : 미분에서 합성 함수의 도함수(미분한 함수)를 구하기 위한 방법

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial w_i}$$

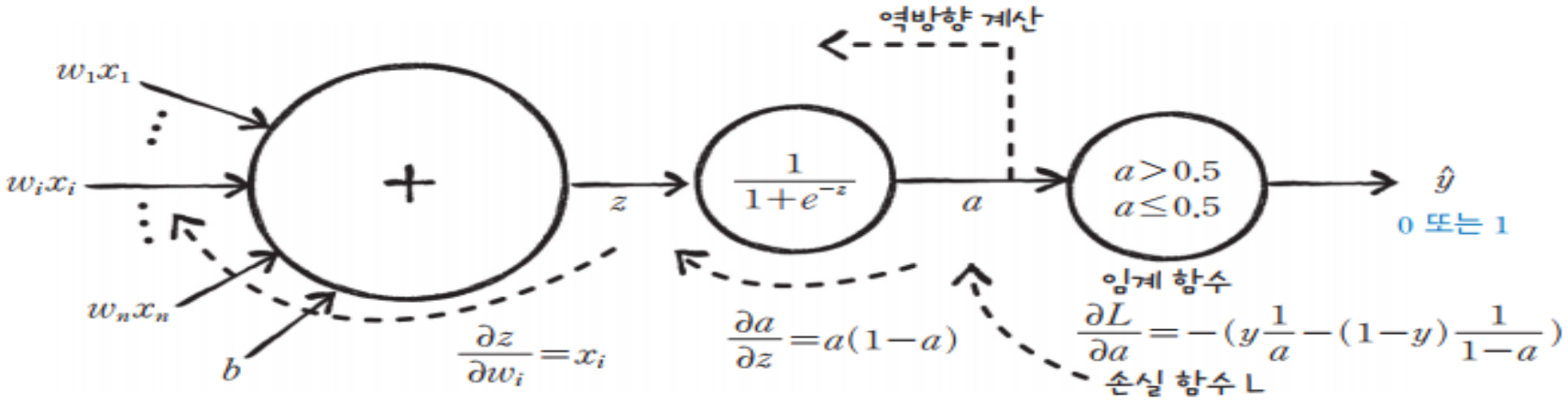
$$= -(y\frac{1}{a} - (1-y)\frac{1}{1-a})a(1-a)x_i = -(y(1-a) - (1-y)a)x_i$$

$$= -(y - ya - a + ya)x_i = -(y-a)x_i$$

STEP 3

이진 분류(Binary Classification)

- **로지스틱 손실 함수의 미분 과정 정리하고 역전파 이해하기**



- 가중치 업데이트 방법 정리하기

$$w_i=w_i-\frac{\partial L}{\partial w_i}=w_i+(y-a)x_i$$

STEP 3

이진 분류(Binary Classification)

- **로지스틱 손실 함수의 미분 과정 정리하고 역전파 이해하기**
  - 절편 업데이트 방법 정리하기

$$\frac{\partial L}{\partial b}=\frac{\partial L}{\partial z}\frac{\partial z}{\partial b}=-(y-a)\frac{\partial}{\partial b}(b+\sum_{i=1}^{n}w_i x_i)=-(y-a)1$$

$$b=b-\frac{\partial L}{\partial b}=b+(y-a)1$$

STEP 3

위스콘신 유방암 데이터 세트(Wisconsin breast cancer dataset)

- **유방암 데이터 세트**
  - 유방암 데이터 세트에는 유방암 세포의 특징 10개에 대하여 평균, 표준 오차, 최대 이상치가 기록되어 있다.
  - 해결할 문제는 유방암 데이터 샘플이 악성 종양(True)인지 혹은 정상 종양(False)인지를 구분하는 이진 분류 문제
  - **주의할 점** : 의학과 이진 분류에서 사용하는 용어가 달라서 착각할 수가 있다.

	의학	이진 분류
좋음	양성 종양(정상 종양)	음성 샘플
나쁨	악성 종양	양성 샘플 (해결 과제)

STEP 3

위스콘신 유방암 데이터 세트(Wisconsin breast cancer dataset)

1. 유방암 데이터 세트 로딩

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_breast_cancer
```

2. 입력 데이터 확인하기

```
print(cancer.data.shape, cancer.target.shape)
```

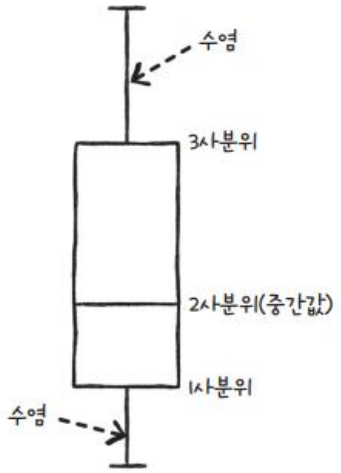
```
cancer.data[:3]
```

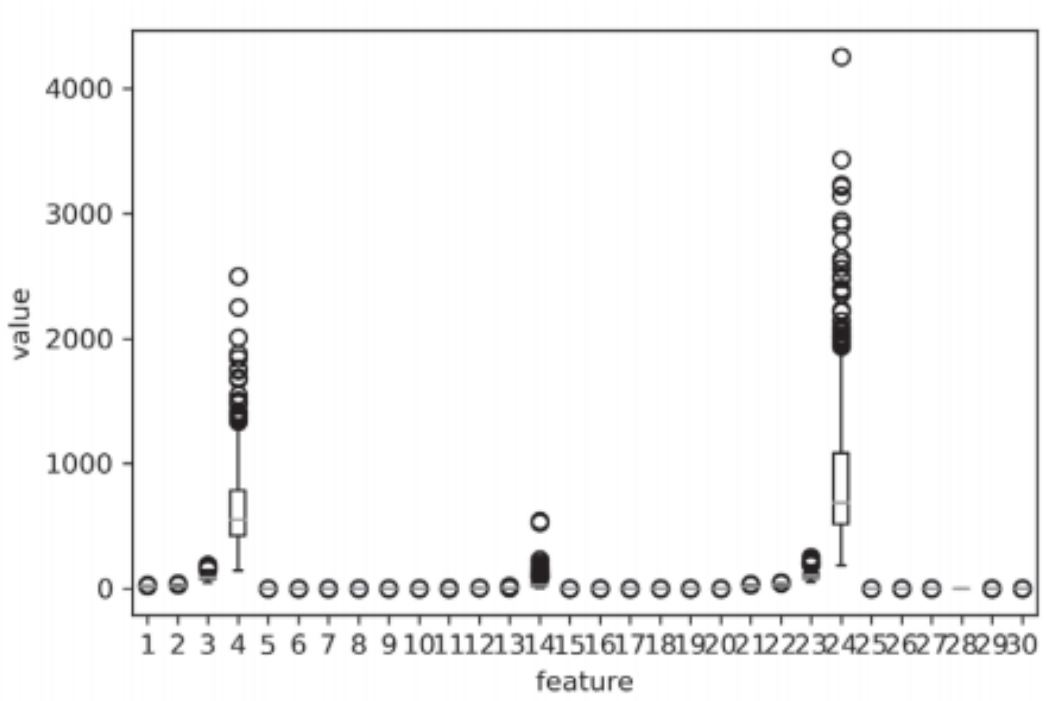
STEP 3

위스콘신 유방암 데이터 세트(Wisconsin breast cancer dataset)

3. 박스 플롯으로 특성의 사분위 수 관찰

```
plt.boxplot(cancer.data)
plt.xlabel('feature')
plt.ylabel('value')
plt.show()
```





STEP 3

위스콘신 유방암 데이터 세트(Wisconsin breast cancer dataset)

4. 눈에 띄는 특성 살펴보기

```
cancer.feature_names[[3, 13, 23]]
```

5. 타깃 데이터 확인하기

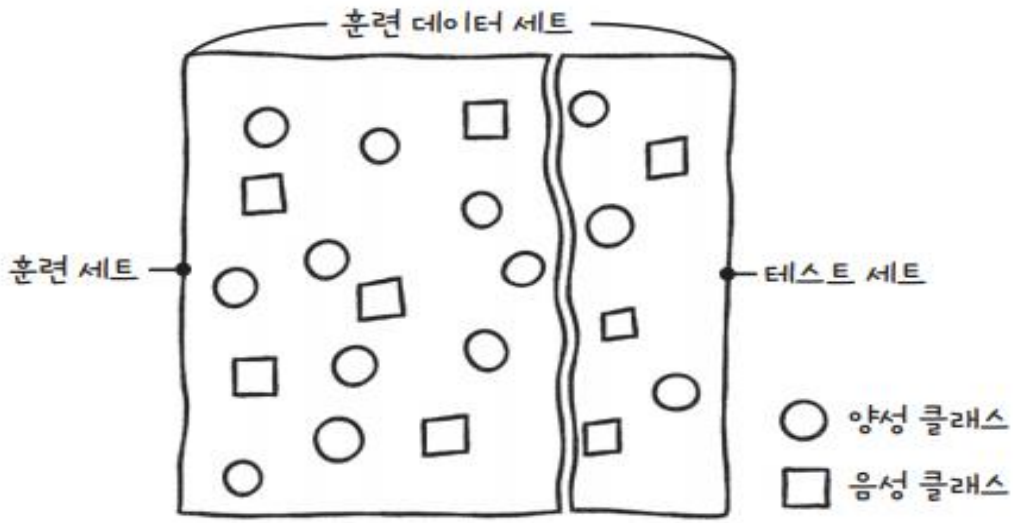
```
np.unique(cancer.target, return_counts=True)
```

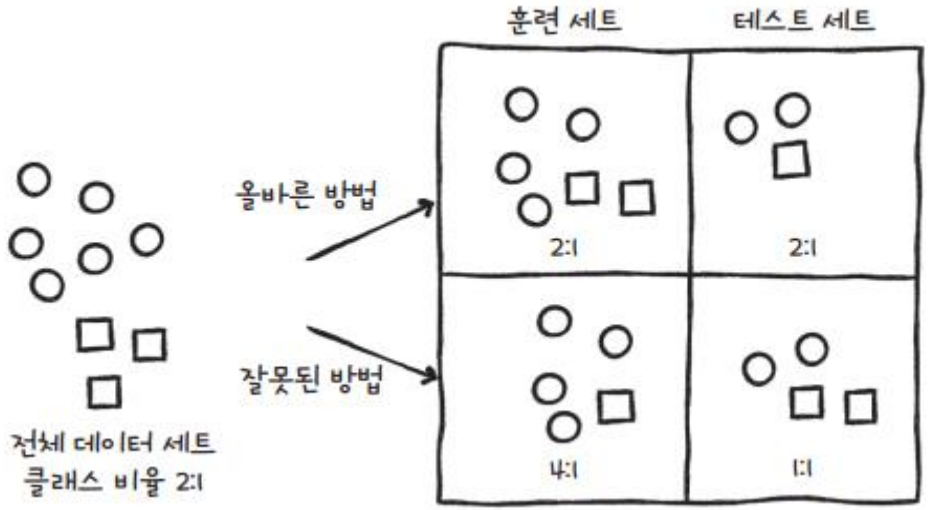
6. 훈련 데이터 세트 저장하기

```
x = cancer.data
y = cancer.target
```

STEP 3
로지스틱 회귀를 위한 뉴런 생성

- **모델의 성능 평가를 위한 훈련 세트와 테스트 세트**
  - 훈련된 모델의 실전 성능을 **일반화 성능(generalization performance)**이라고 부른다.
  - 훈련 데이터 세트를 훈련 세트와 테스트 세트로 나누는 규칙
  - 훈련 데이터 세트를 나눌 때는 테스트 세트보다 훈련 세트가 더 많아야 한다.
  - 훈련 데이터 세트를 나누기 전에 양성, 음성 클래스가 훈련 세트나 테스트 세트의 어느 한쪽에 몰리지 않도록 골고루 섞어야 한다.





STEP 3

로지스틱 회귀를 위한 뉴런 생성

1. train_test_split() 함수로 훈련 데이터 세트 나누기

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, stratify=y,
test_size=0.2, random_state=42)
```

- stratify=y : 훈련 데이터를 나눌 때 클래스 비율을 동일하게 만든다.
- test_size=0.2 : 기본적으로 훈련 데이터 세트를 75:25 비율로 나눈다. 사용자가 원할 경우 비율 조정
- random_state=42 : train_test_split() 함수는 무작위로 데이터 세트를 섞은 다음 나눈다. 섞은 다음 나눈 결과가 항상 일정하도록 난수 초깃값을 지정

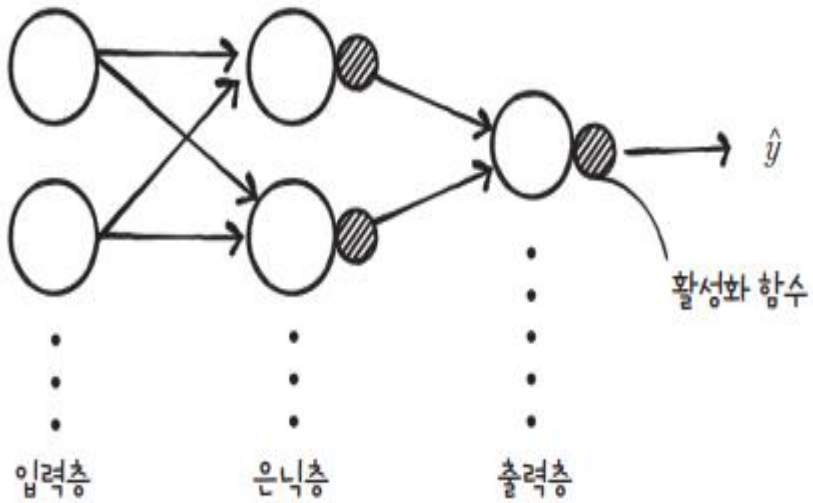
2. 결과 확인하기

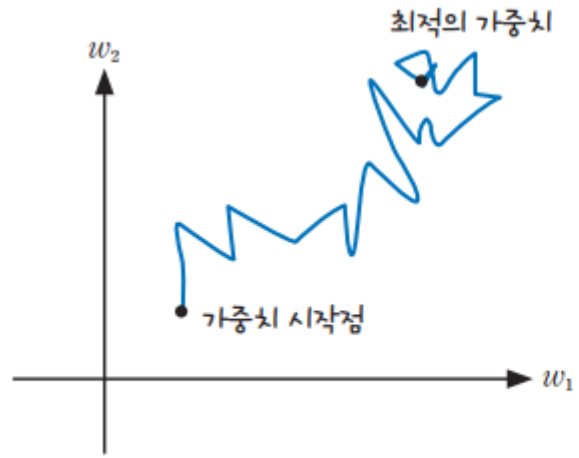
```
print(x_train.shape, x_test.shap)
```

3. unique() 함수로 훈련 세트의 타깃 확인하기

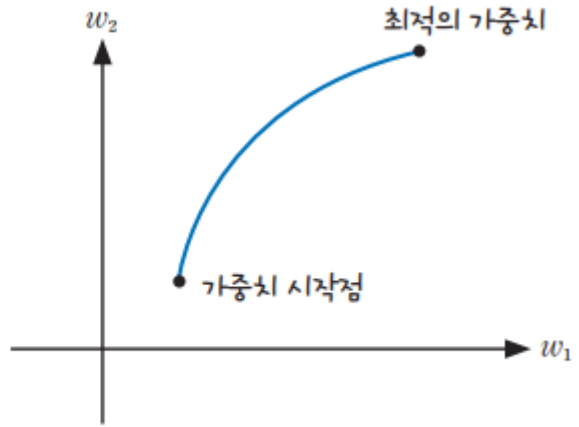
```
np.unique(y_train, return_counts=True)
```

STEP 3
로지스틱 회귀 구현





확률적 경사 하강법



배치 경사 하강법