23 하계 학부연구생 프로그램

ch04. 다양한 분류 알고리즘

인공지능공학과 12223547 박혜민

생선 클래스의 확률 예측 프로그램

01

KNN 분류기

- 작동방식
- 문제점

02

로지스틱 회귀

- 작동방식

03

확률적 경사 하강법

04

손실함수

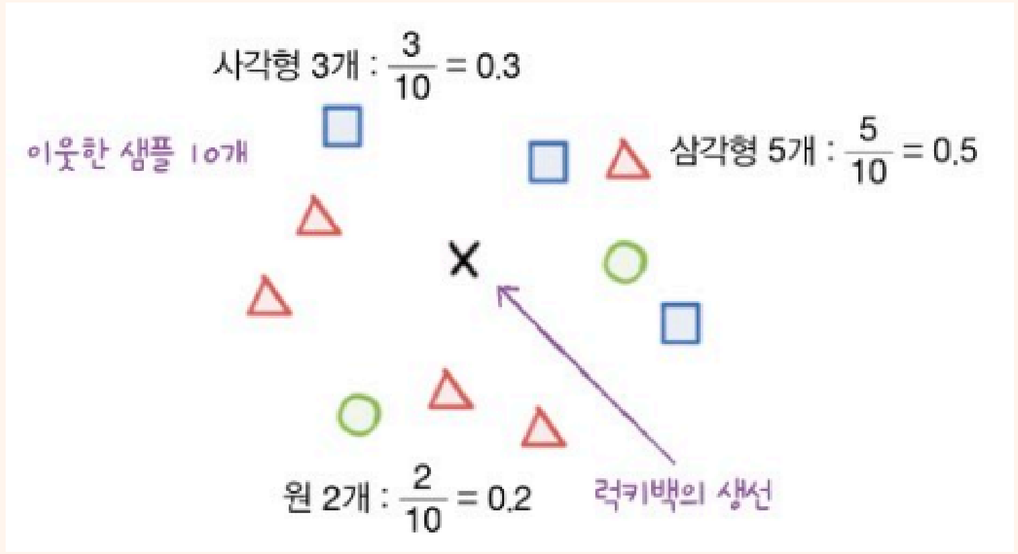
- 확률의 필요성
- 로지스틱 손실함수

KNN 분류기

주변 이웃의 클래스 비율을 확률로 사용

predict_proba() : 클래스별 확률값 계산

k = 10




```
proba = kn.predict_proba(test_scaled[:5])
print(np.round(proba, decimals = 4)) #소수점 4째 자리까지 표기
```

```
[[0.     0.     1.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     1.     0.    ]
 [0.     0.     0.     1.     0.     0.     0.    ]
 [0.     0.     0.6667 0.     0.3333 0.     0.    ]
 [0.     0.     0.6667 0.     0.3333 0.     0.    ]]
```

KNN 분류기의 문제점

확률값의 신뢰성을 높이려면?

<k = 3인 모델>

도출되는 확률 값 : 0, 1/3, 2/3, 1

```
proba = kn.predict_proba(test_scaled[:5])
print(np.round(proba, decimals = 4)) #소수점 4째 자리까지 표기
```

```
[[0.     0.     1.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     1.     0.    ]
 [0.     0.     0.     1.     0.     0.     0.    ]
 [0.     0.     0.6667 0.     0.3333 0.     0.    ]
 [0.     0.     0.6667 0.     0.3333 0.     0.    ]]
```

→ 비율로 접근하여 나올 수 있는 값이 한정적

→ 선형적으로 접근

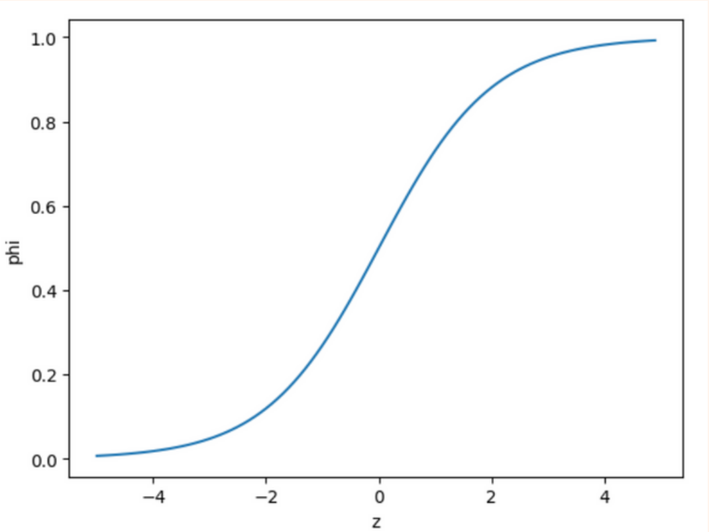
로지스틱 회귀

선형회귀 분류모델

<step>
1. 선형 방정식 학습
2. z값 도출
3. 시그모이드 함수/로지스틱 함수를 사용하여 z를 0~1 값으로 반환
4. 반환 값이 0.5 초과면 양성 클래스, 0.5 이하면 음성 클래스로 분류

$$z = a \times (Weight) + b \times (Length) + c \times (Diagonal) + d \times (Height) + e \times (Width) + f$$

선형 방정식

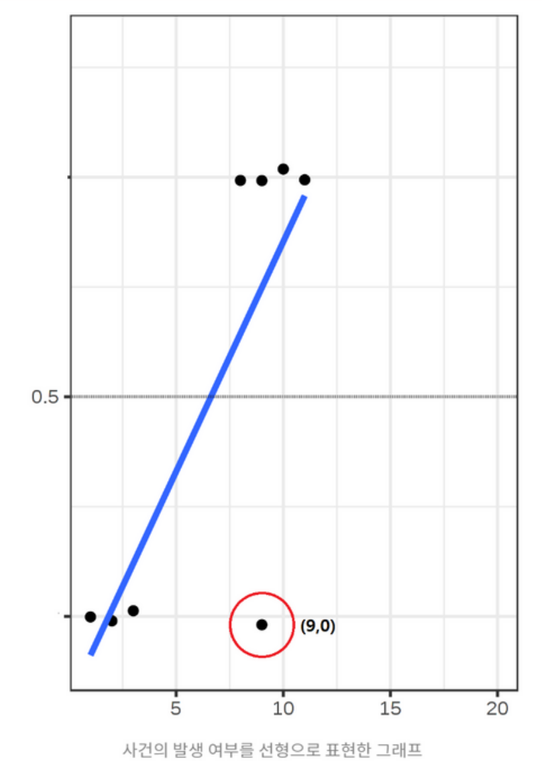


$$\phi = \frac{1}{1+e^{-z}}$$

시그모이드 함수

시그모이드 함수

이 함수를 왜 사용할까?



linear regression

선형 회귀 모델의 문제점

1. 새로운 특이한 데이테의 추가 기존 분류 모델에 큰 영향을 미친다.
2. 출력값의 임계값을 설정할 수 없다.

시그모이드 함수

이 함수를 왜 사용할까?

S자 모양의 그래프로 표현된 시그모이드(Sigmoid) 함

sigmoid

- 구조 : x 값이 매우 크면 1에 수렴. 매우 작으면 0에 수렴
- 비선형적인 데이터의 특성을 잘 반영한다.
- 출력값이 0 ~ 1 사이로 제한된다.

로지스틱 회귀의 다중분류

<특징>

- 반복 알고리즘
  - max_iter 매개변수로 반복횟수 조절 (default = 100)
- L2 규제
  - C 매개변수로 규제정도 조절 (default = 1)
    * alpha 와는 달리 C값 작을수록 규제 강함
- 클래스 각각의 선형방정식 학습
- 소프트맥스 함수

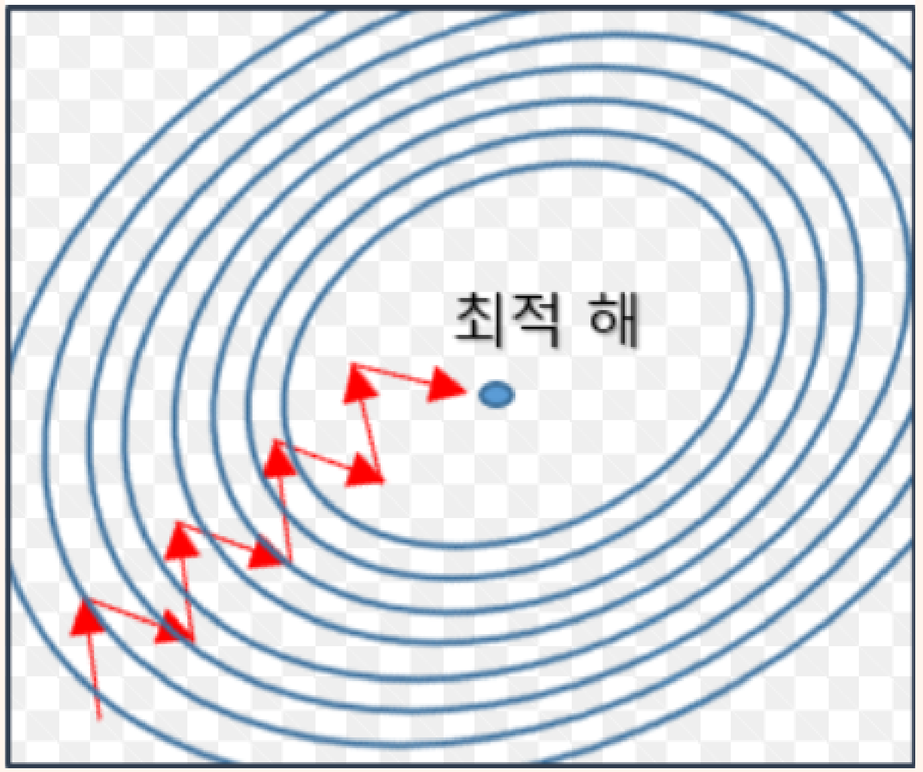
$$s1 = \frac{e^{z1}}{e\_sum},\ s2 = \frac{e^{z2}}{e\_sum},\ \cdots,\ s7 = \frac{e^{z7}}{e\_sum}$$

- z 를 0~1 사이로 반환(소프트맥스)
- 전체 합이 1이 되도록 정규화

확률적 경사 하강법

데이터를 점진적으로 학습하는 방법

훈련세트에서 랜덤하게 하나의 샘플을 선택하여 가파른 경사(손실함수)를 조금씩 내려가는 것을 반복하여 최종적으로 원하는 지점에 도달하는 방법



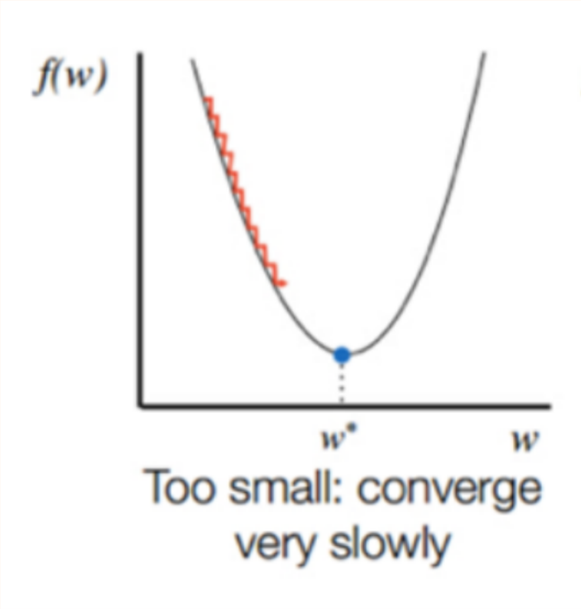
확률적 경사하강법

- eopch : 훈련 세트를 한 번 모두 사용하는 과정
- batch size : 한 번 학습 시 사용하는 샘플의 수
- iteration : 1 epoch의 반복 횟수

확률적 경사하강법

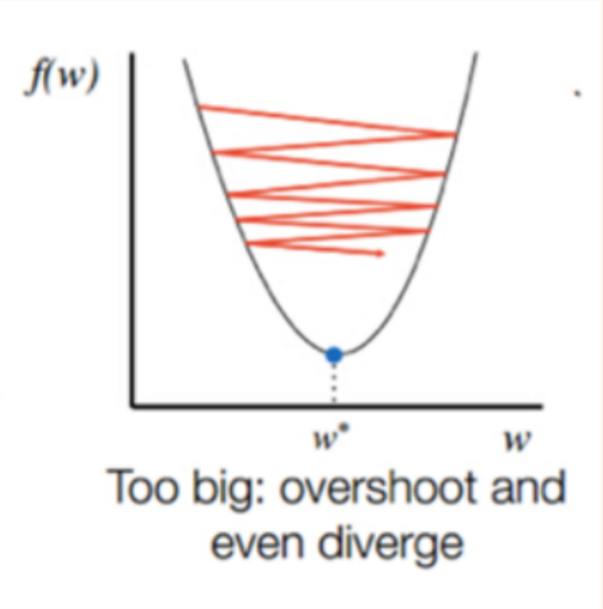
조금씩 이동하는 이유

<step size 가 매우 클 경우>



- 학습시간이 오래걸림.
- 지역 최소값(local minimum)에 수렴할 수 있음.

<step size 가 매우 작을 경우>



- 학습 시간 적게 걸림.
- 전역 최소값(global minimum)을 가로질러 멀어질 수 있음.

손실함수

조건 : 미분 가능(연속적)해야 함

1. 확률을 사용하지 않았을 때

- 값이 이산적이고 듬성듬성 하여 조금씩 움직일 수 없다.

연속적인 개념 '확률' 도입

2. 확률을 사용했을 때

확률 / 예측 / 정답

0.9 / 1 / 1

0.3 / 0 / 1

0.2 / 0 / 0

0.8/ 1 / 0

0.9 x 1 => -0.9 (정확도 높음, 손실 작음)

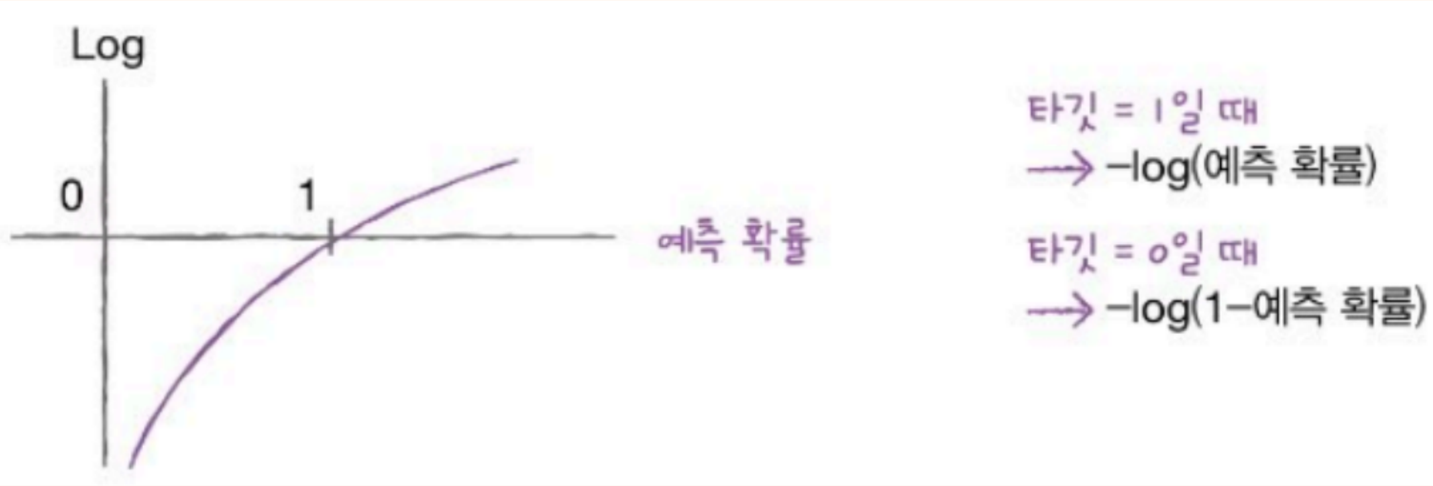
0.3 x 1 => -0.3 (정확도 낮음, 손실 큼)

0.2 -> 0.8 x 1 => -0.8 (정확도 높음, 손실 작음)

0.8 -> 0.2 x 1 => -0.2 (정확도 낮음, 손실 큼)

로지스틱 손실함수

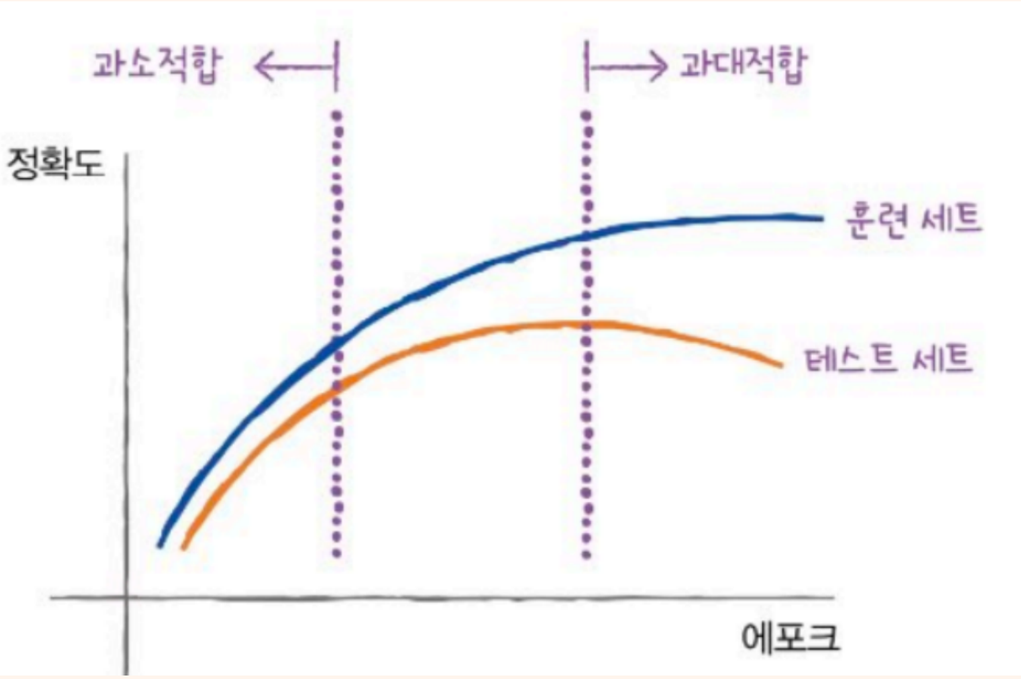
예측확률에 로그함수 적용



- 양수값의 결과로 더 직관적으로 이해 가능
- 0에 가까울 수록 손실 값이 아주 큰 음수가 되어 모델에 큰 영향을 줌

에포크와 과소적합/과대적합

적절한 eopch 지점 찾기



eopch 진행될수록 과대적합 유도

↓

테스트 셋 정확도가 감소하기 시작하는 시점에서 early stopping