

기계학습개론

- 분류 알고리즘 -

교수 이홍로

MP : 010-6611-3896

E-mail : hrlee@cnu.ac.kr

강의 홈페이지 : <https://cyber.hanbat.ac.kr/>



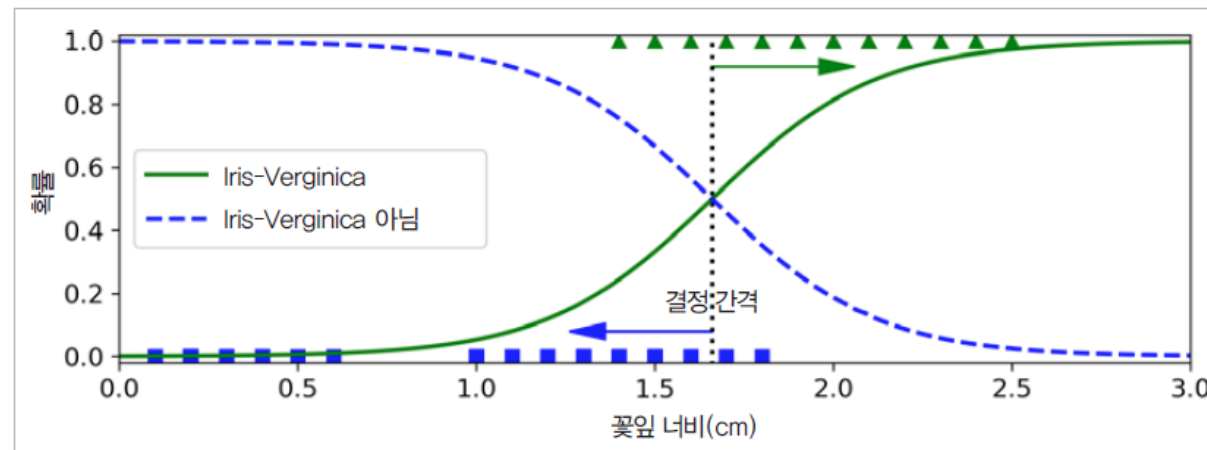
오늘의 강의 목표

- 로지스틱 회귀 알고리즘
- 확률적 경사 하강법

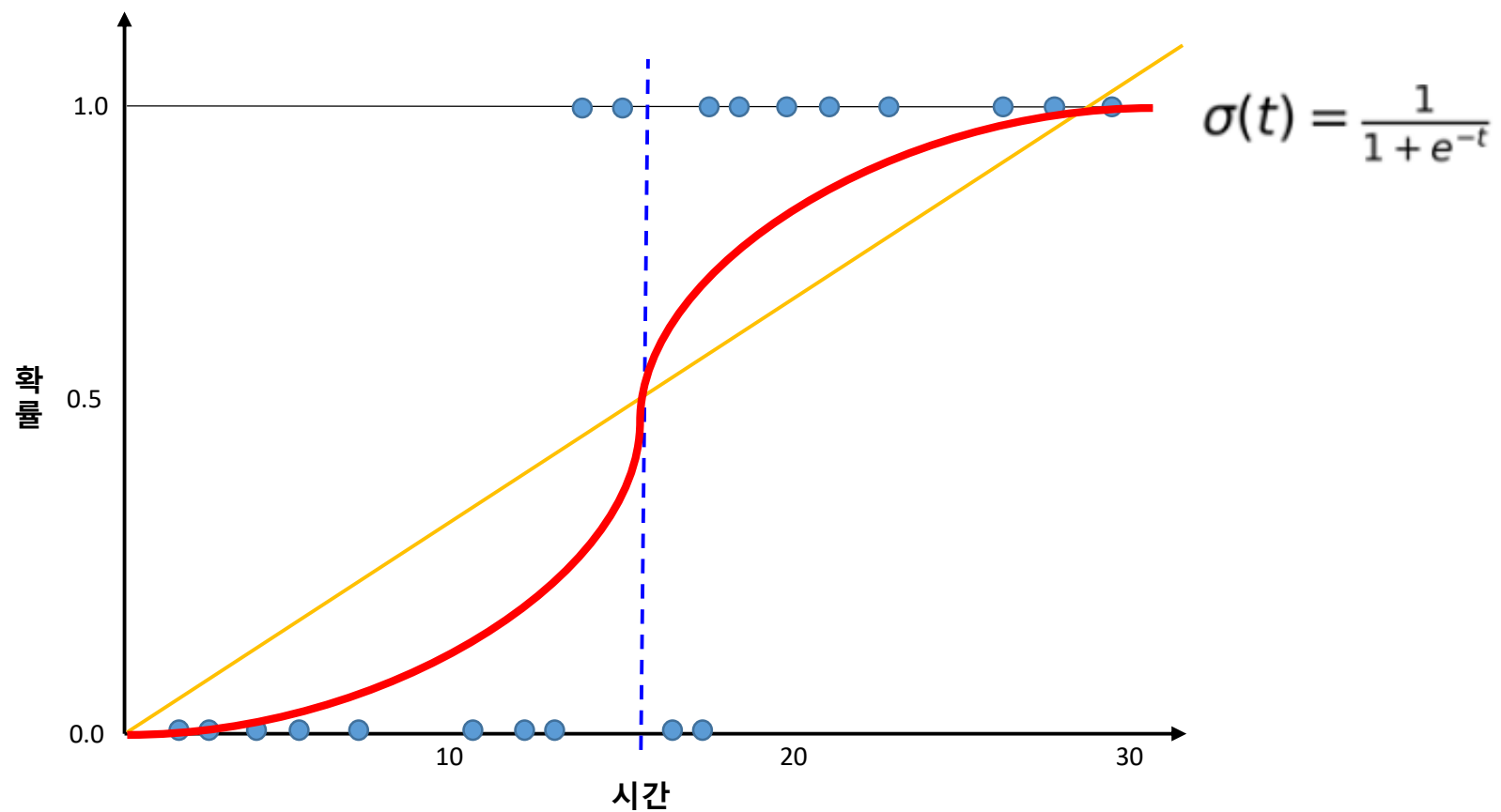
로지스틱 회귀 알고리즘

■ 로지스틱 회귀 알고리즘이란?

- D.R.Cox가 1958년에 제안한 데이터 분류를 위한 지도학습 알고리즘
- 독립 변수의 선형 결합을 이용하여 사건의 발생 가능성을 예측하는 데 사용
- 데이터가 특정 클래스에 속할 확률을 추정하는데 사용
- 사용 예> 암컷/수컷 분류, 이메일의 스팸 여부 분류, 사과의 품질 분류, 숫자 인식 등



로지스틱 회귀 알고리즘



로지스틱 회귀 알고리즘

■ 결정 경계

- 꽃잎의 너비가 0~3cm인 꽃에 대한 모델의 추정 확률을 계산

```
from sklearn.linear_model import LogisticRegression
```

```
log_reg = LogisticRegression()  
log_reg.fit(X, y)
```

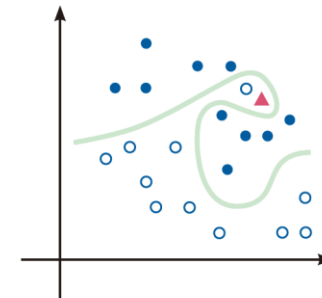
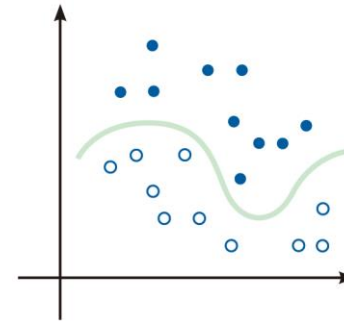
```
X_new = np.linspace(0, 3, 1000).reshape(-1, 1)  
y_proba = log_reg.predict_proba(X_new)  
plt.plot(X_new, y_proba[:, 1], "g-", label="Iris virginica")  
plt.plot(X_new, y_proba[:, 0], "b--", label="Not Iris virginica")  
# 다음은 이미지를 미려하게 출력하기 위한 맷플롯립 코드가 이어집니다.
```



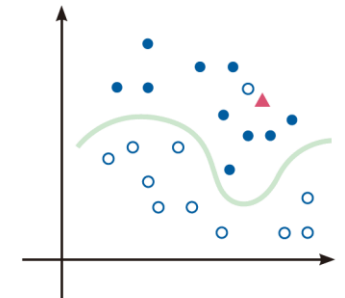
로지스틱 회귀 알고리즘

■ 결정 경계(계속)

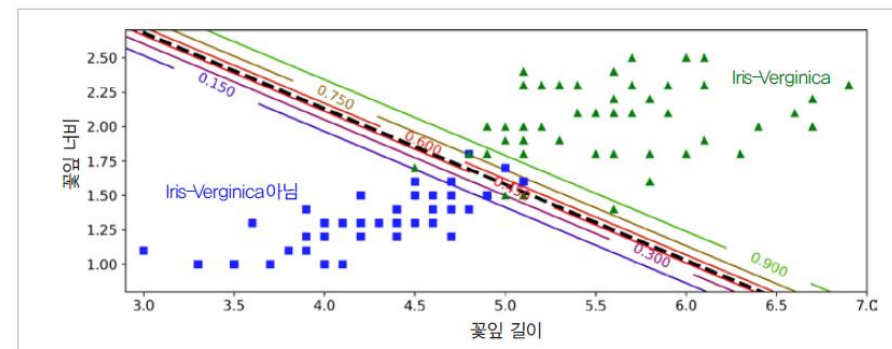
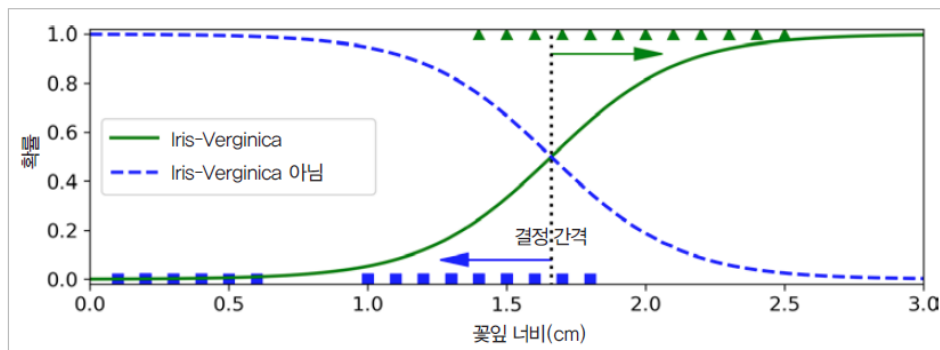
- 비선형 분류기 $\text{nonlinear classifier}$ 사용
- 과잉 적합 overfitting 회피
 - 과잉 적합은 아웃라이어를 맞히려고 과다하게 복잡한 결정 경계를 만드는 현상
 - 훈련 집합에 대한 성능은 높지만 테스트 집합에 대해서는 형편없는 성능(낮은 일반화)



(a) 아웃라이어에 의한 과잉 적합



(b) 아웃라이어를 걸러낸 경우



로지스틱 회귀 알고리즘

■ 소프트맥스 회귀

- 클래스 K에 대한 소프트맥스 점수
- 소프트맥스 함수
- 회귀 분류기의 예측
- 크로스 엔트로피 비용함수
- 클래스 K에 대한 크로스 엔트로피의 그레디언트 벡터

$$s_k(\mathbf{x}) = (\boldsymbol{\theta}^{(k)})^T \mathbf{x}$$

$$\hat{p}_k = \sigma(s(\mathbf{x}))_k = \frac{\exp(s_k(\mathbf{x}))}{\sum_{j=1}^K \exp(s_j(\mathbf{x}))}$$

$$\hat{y} = \operatorname{argmax}_k \sigma(s(\mathbf{x}))_k = \operatorname{argmax}_k s_k(\mathbf{x}) = \operatorname{argmax}_k ((\boldsymbol{\theta}^{(k)})^T \mathbf{x})$$

$$J(\boldsymbol{\Theta}) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \log(\hat{p}_k^{(i)})$$

$$\nabla_{\boldsymbol{\theta}^{(k)}} J(\boldsymbol{\Theta}) = \frac{1}{m} \sum_{i=1}^m (\hat{p}_k^{(i)} - y_k^{(i)}) \mathbf{x}^{(i)}$$

로지스틱 회귀 알고리즘 구현하기

■ 문제

- 다음과 같이 무게, 길이, 대각선, 세로, 가로,의 제품 데이터가 있을 때, A~G까지의 제품을 분류해 보세요.
- 참고 :

	Type	Weight	Length	Diagonal	Height	Width
0	A	242.0	25.4	30.0	11.5200	4.0200
1	A	290.0	26.3	31.2	12.4800	4.3056
2	A	340.0	26.5	31.1	12.3778	4.6961
3	A	363.0	29.0	33.5	12.7300	4.4555
4	A	430.0	29.0	34.0	12.4440	5.1340
...
154	F	12.2	12.2	13.4	2.0904	1.3936
155	F	13.4	12.4	13.5	2.4300	1.2690
156	F	12.2	13.0	13.8	2.2770	1.2558
157	F	19.7	14.3	15.2	2.8728	2.0672
158	F	19.9	15.0	16.2	2.9322	1.8792

159 rows × 6 columns

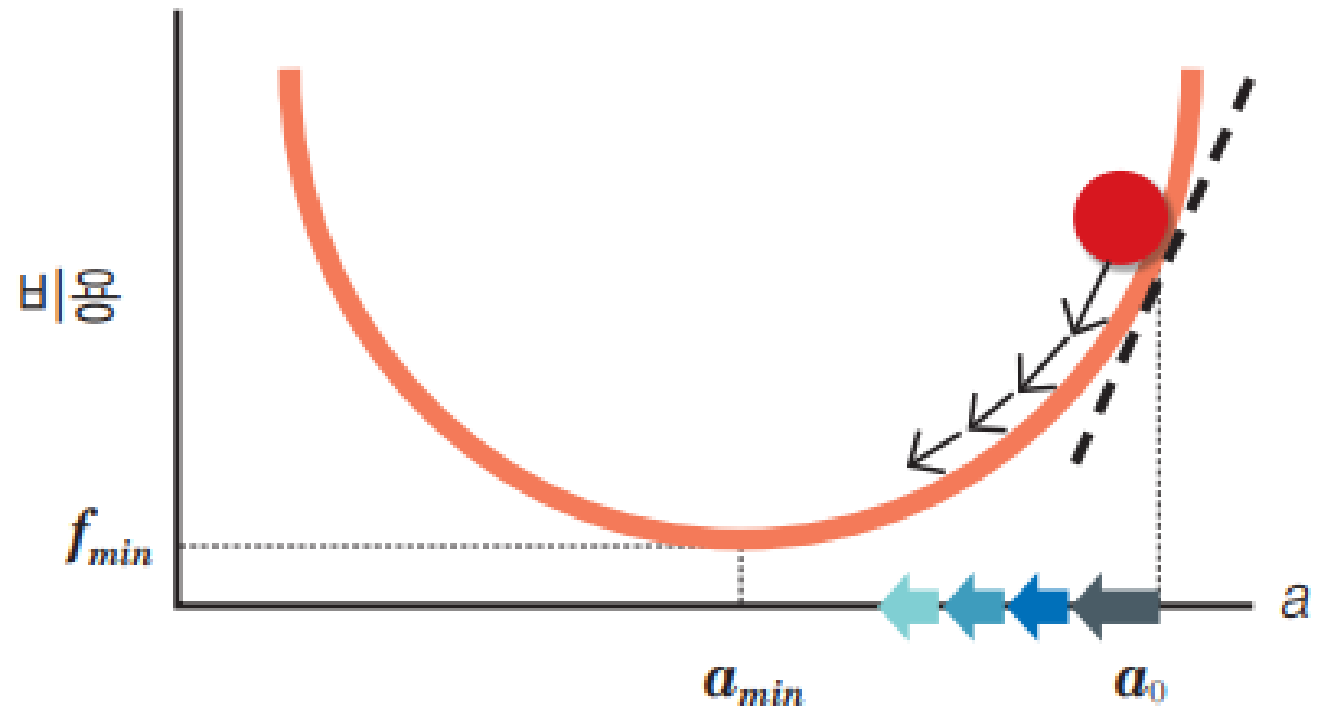
경사 하강법

■ 경사 하강법이란?

- 점진적인 학습을 진행하며 가장 빠른 경사를 따라 목표점에 도달하는 ML 알고리즘

■ 경사 하강법의 특징

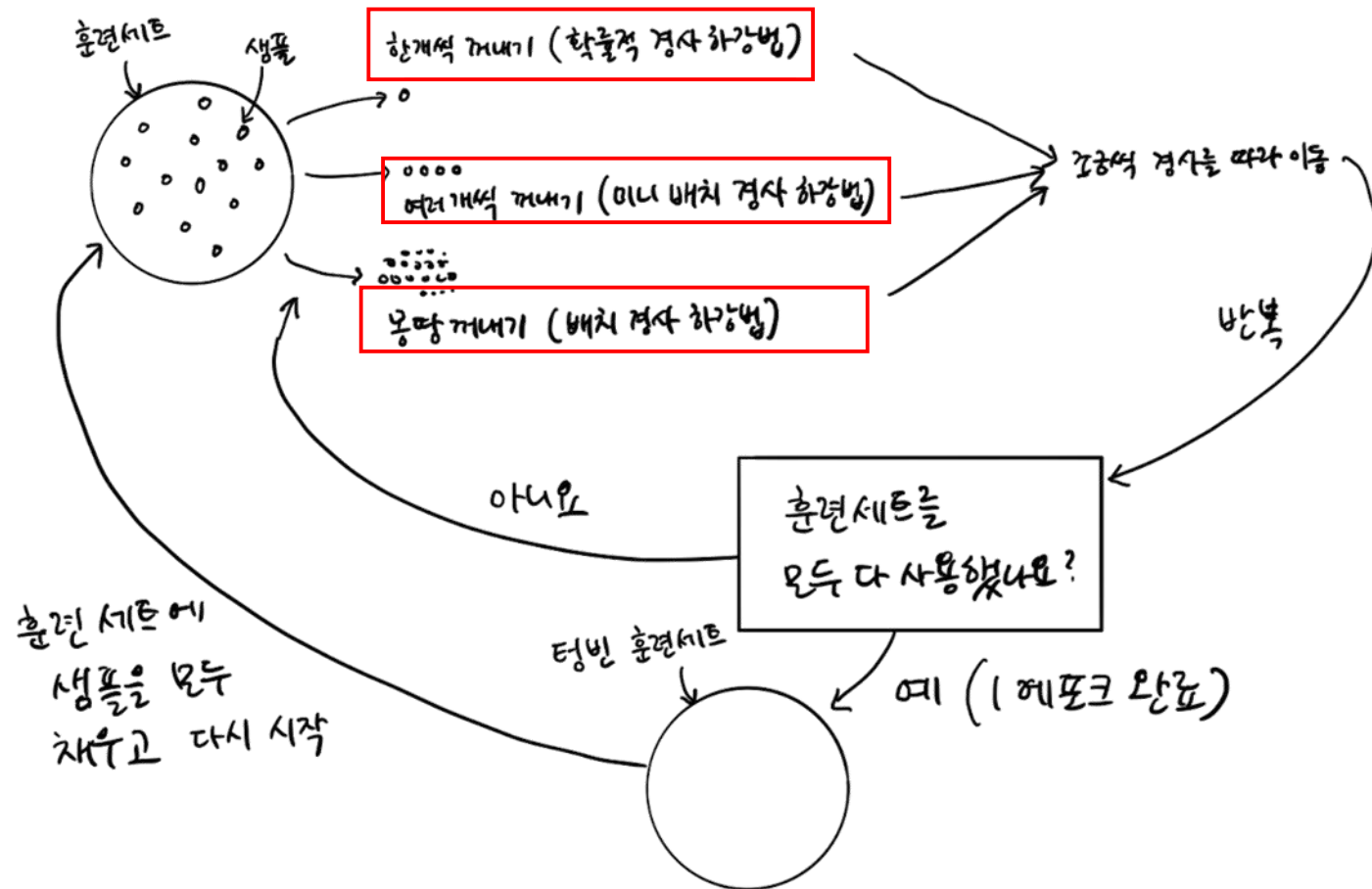
- 많은 데이터를 나누어 처리 가능
- 매일매일 모델을 업데이트하여 서비스에 적용 가능
- 신경망(Deep Learning) 알고리즘에서 활용



경사 하강법

■ 경사하강법의 종류

- 배치 경사 하강법
- 미니배치 경사 하강법
- 확률적 경사 하강법



확률적 경사 하강법

■ 확률적 경사 하강법이란?

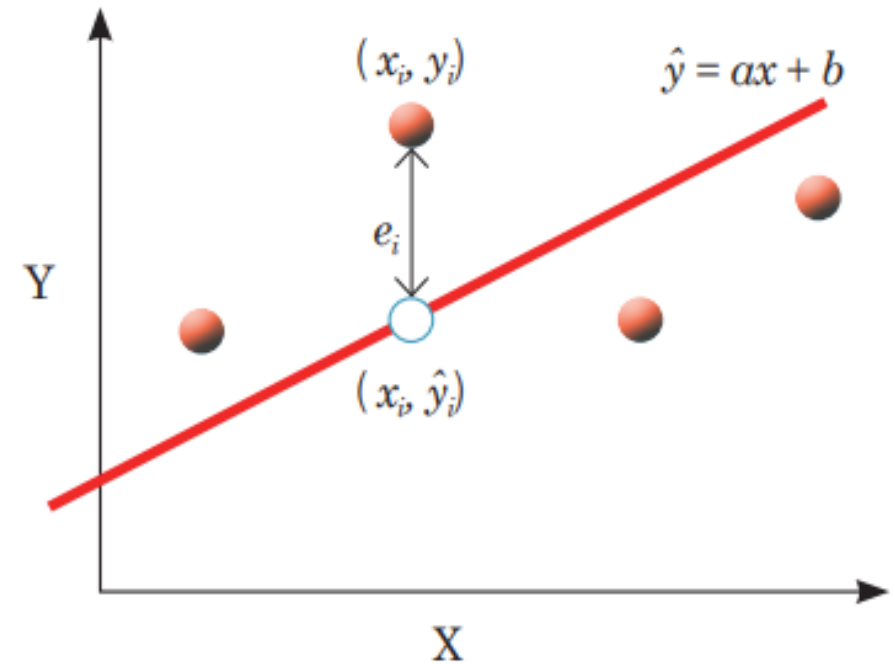
- 훈련 데이터에서 랜덤하게 한 개의 데이터를 선택하여 훈련시키는 경사 하강법

■ 회귀선

$$y = ax + b$$

■ 비용함수(손실함수)

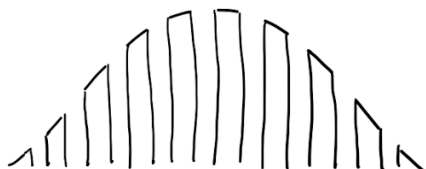
$$f(a,b) = \frac{1}{n} \sum_{i=1}^n e_i^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{n} \sum_{i=1}^n \{y_i - (ax_i + b)\}^2$$



확률적 경사 하강법

로지스틱 손실함수(logistic loss function)

예측	정답(타겟)
1	= 1
0	≠ 1
0	= 0
1	≠ 0



산이 등성등성 하다면
오르거나 내려올수 없습니다

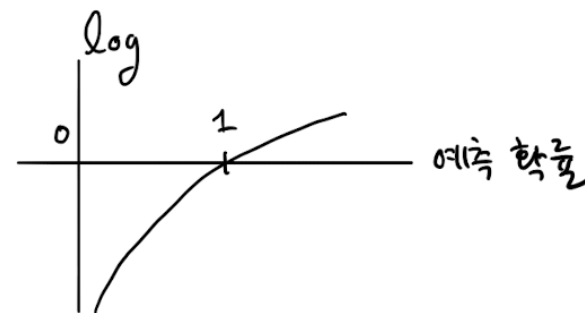


진짜 산은 연속적입니다

예측	정답(타겟)	
0.9	x 1	→ -0.9
0.3	x 1	→ -0.3
0.2 → 0.8	x 1	→ -0.8
0.8 → 0.2	x 1	→ -0.2

낮은 손실

높은 손실



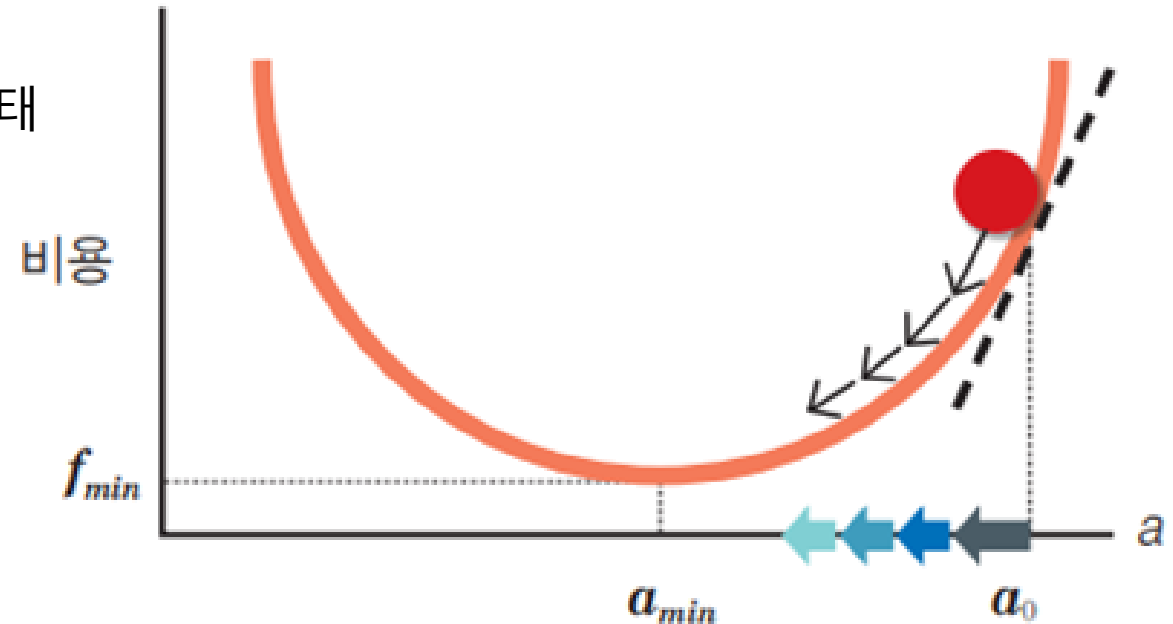
타겟 = 1 일때
→ $-\log(\text{예측 확률})$

타겟 = 0 일때
→ $-\log(1 - \text{예측 확률})$

확률적 경사 하강법

■ 파라미터와 비용함수

- 오차가 커지면 비용은 제곱으로 비례
- 파라미터가 변함에 따라 아래로 볼록한 곡면의 형태
- 임의의 초기값 (a_0, b_0)에서 각각 $\Delta a, \Delta b$ 만큼 이동하면서 비용이 최소(f_{min})가 되는 최적점 (a_{min}, b_{min})을 찾아가는 경로
- $a' = a + \Delta a, b' = b + \Delta b$



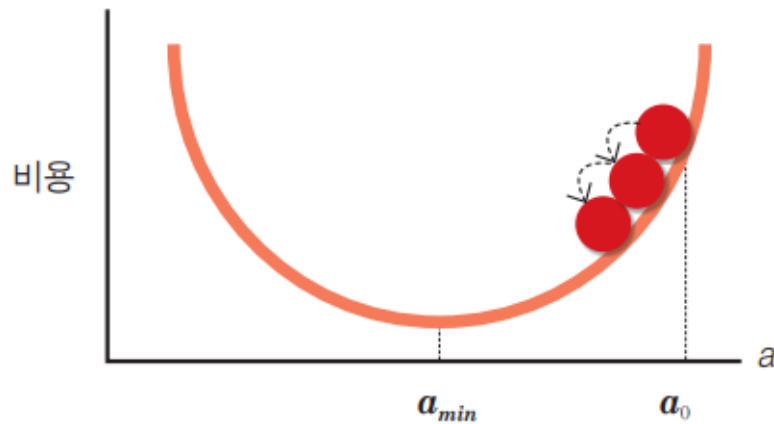
a 관점의 경로

확률적 경사 하강법

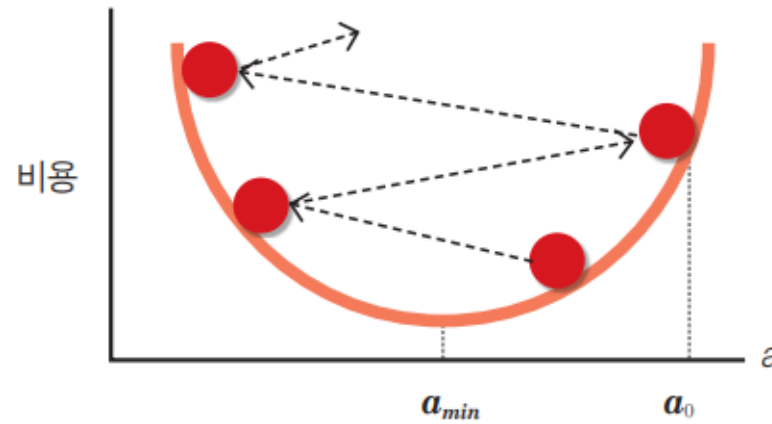
■ 학습률

- 너무 작으면 학습시간이 오래 걸리고,
- 크면 그래프 반대편에 있는 더 높은 비용값의 위치로 이동하면서 발산

■ 작은 학습률에서 점차 증가하면서 효과적 학습 찾음



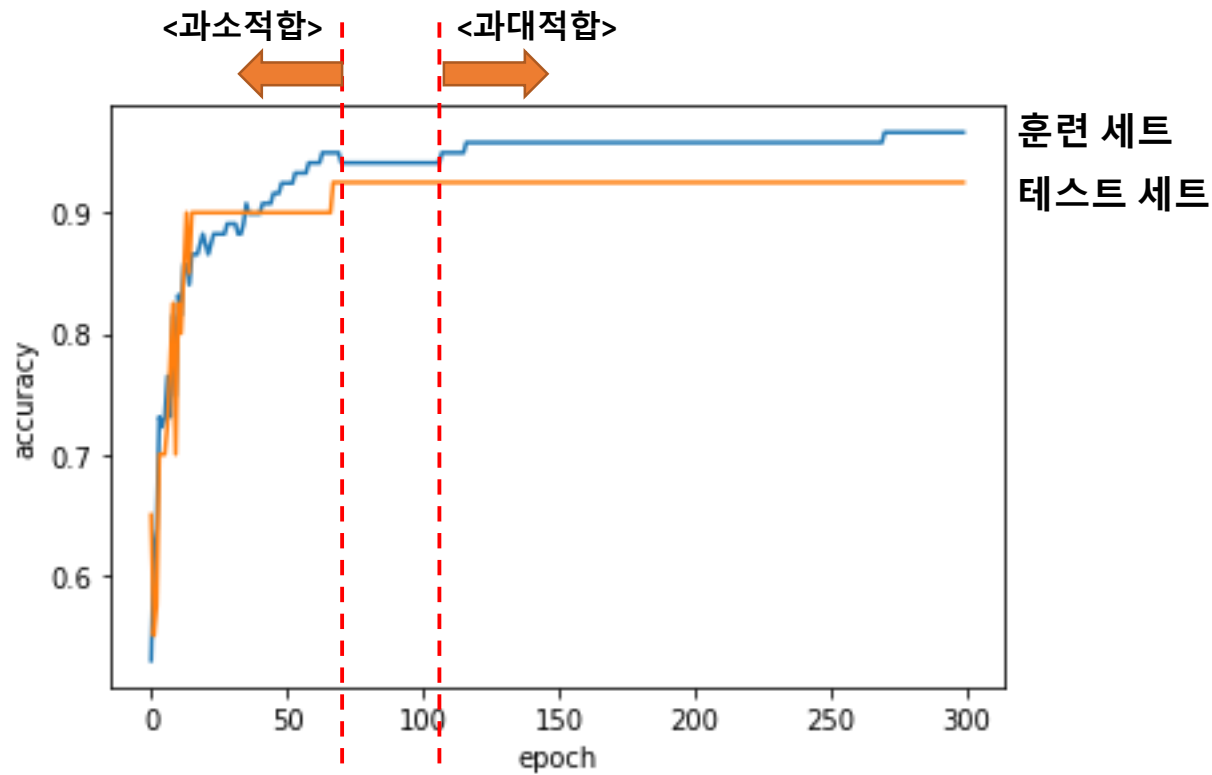
학습률이 너무 작을 때(오랜 학습시간)



학습률이 너무 클 때(발산)

확률적 경사 하강법

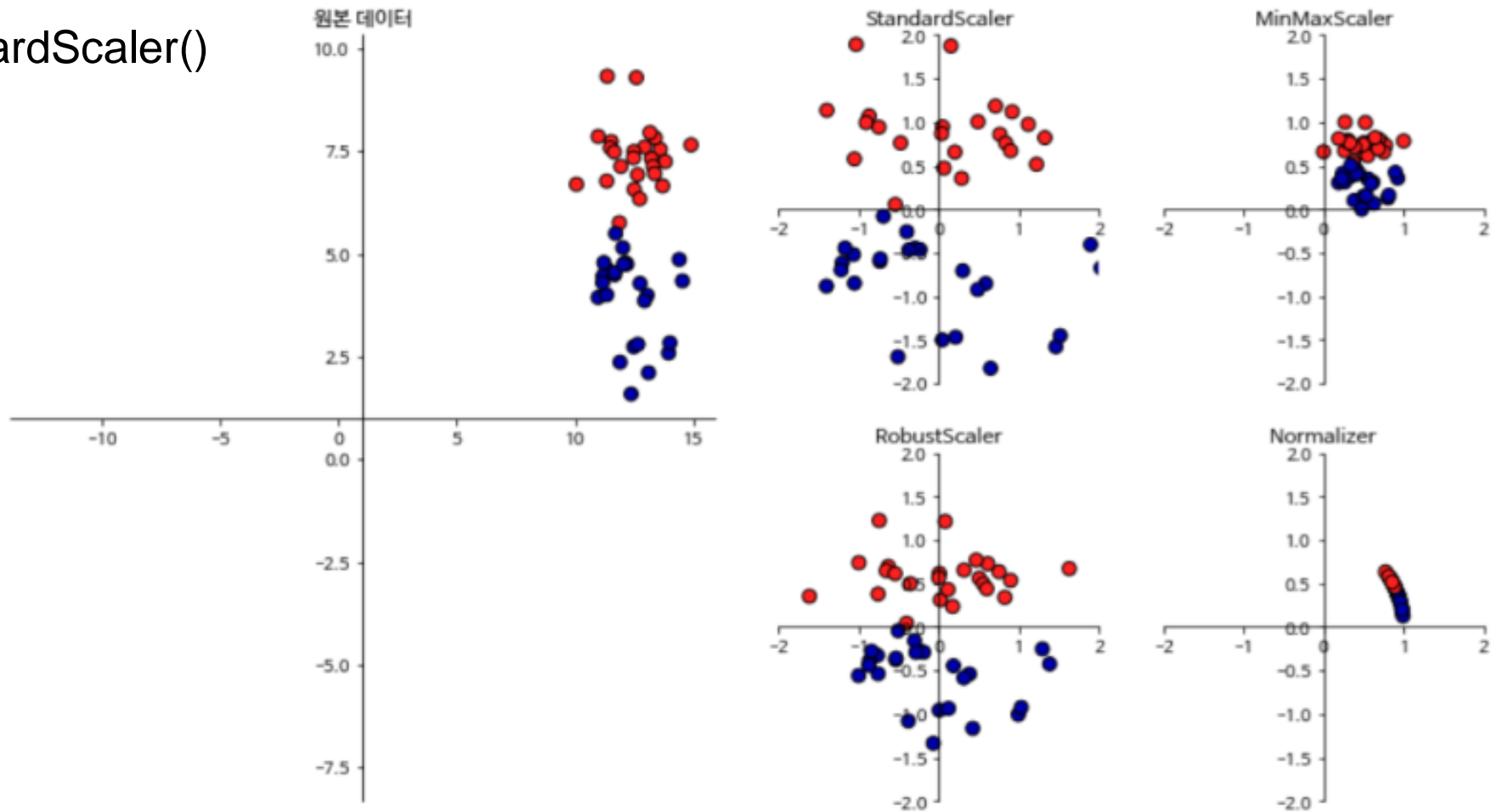
- 에포크와 과대/과소 적합



참고자료

■ 전처리 : 스케일 조정

- StandardScaler()



“어떤 일이 충분히 **중요**하다면
역경이 있더라도
그것을 반드시 **해야만 한다.**”

-엘론 머스크-

