4장 모델 훈련

- 모델 훈련 작동과정 이해
    - 디버깅, 에러 분석 등에 도움
- 딥러닝의 신경망 이해, 구축, 훈련에 필요한 주제 포함

개요

- 선형 회귀 모델
    - 수학적으로 모델 파라미터 구하기
    - 경사하강법 적용 모델 훈련

- 경사하강법 종류
    - 배치 경사하강법
    - 미니배치 경사하강법
    - 확률적 경사하강법(SGD)

- 다항 회귀
    - 비선형 모델 훈련법
- 학습 곡선
    - 과대적합 감지
- 규제 선형 모델
    - 과대적합 위험 감소시키기
- 로지스틱 회귀와 소프트맥스 회귀
    - 회귀 모델을 이용한 분류기로 활용하기

4.1 선형 회귀

선형 회귀 모델 함수

- 한 개의 특성 x_1을 사용하는 i번째 훈련 샘플에 대한 예측값

$$\hat{y}^{(i)} = \theta_0 + \theta_1 \, x_1^{(i)}$$

- $n \geq 1$ 개의 특성을 사용하는 i번째 훈련 샘플에 대한 예측값

$$\hat{y}^{(i)} = \theta_0 + \theta_1 \, x_1^{(i)} + \cdots + \theta_n \, x_n^{(i)}$$

- $n \geq 1$ 개의 특성을 사용하는 i번째 훈련 샘플에 대한 예측값

$$\hat{y}^{(i)} = \theta_0 + \theta_1 \, x_1^{(i)} + \cdots + \theta_n \, x_n^{(i)}$$

$$\hat{y}^{(i)} = \theta_0 + \theta_1 \, x_1^{(i)} + \cdots + \theta_n \, x_n^{(i)}$$

- $\hat{y}^{(i)}$: i 번째 훈련 샘플에 대한 예측값
- $x_k^{(i)}$: i 번째 훈련 샘플의 k 번째 특성값
- θ_0: 편향 파라미터
- θ_k $(k > 0)$: k 번째 특성에 대한 가중치 파라미터

선형 회귀 모델의 행렬 연산 표기법

$$\hat{\mathbf{y}}^T = h_\theta(\mathbf{X}) = \theta^T \mathbf{X}_b^T$$

- 여기서 $\mathbf{x}_b^{(i)}$.

$$= \begin{bmatrix} 1 & x_1^{(i)} & \cdots & x_n^{(i)} \end{bmatrix}$$

- 파이썬 넘파이 2차원 어레이 표현

데이터	행렬 기호	수학 행렬 모양(shape)	넘파이 어레이 모양
레이블, 예측값	$\mathbf{y}, \hat{\mathbf{y}}$	$m \times 1$	(m, 1)
가중치	θ	$(n+1) \times 1$	(n+1, 1)
훈련 세트	$\mathbf{X}$	$m \times n$	(m, n)
훈련 센트(수정)	$\mathbf{X}_b$	$m \times (n+1)$	(m, n+1)

비용함수: 평균 제곱 오차(MSE)

- MSE를 활용한 선형 회귀 모델 성능 평가

$$\text{MSE}(\mathbf{X}, h_\theta) = \frac{1}{m} \sum_{i=0}^{m-1} (\theta^T \, (\mathbf{x}_b^{(i)})^T - y^{(i)})^2$$

- 간단한 표현:

$$\text{MSE}(\theta) := \text{MSE}(\mathbf{X}, h_\theta)$$

목표: 비용함수 최소화

- MSE를 아래와 같이 θ를 입력변수로 갖는 함수로 간주할 수 있음.

$$\text{MSE}(\theta) := \text{MSE}(\mathbf{X}, h_\theta)$$

- 이유: $\mathbf{X}$, m, $\mathbf{x}_b$, $y^{(i)}$은 모두 주어진 상수값들임.

- 목표: $\text{MSE}(\theta)$가 최소가 되도록 하는 θ 찾기

- 방식 1: 정규방정식 또는 특이값 분해(SVD) 활용
    - 드물지만 수학적으로 비용함수를 최소화하는 θ 값을 직접 계산할 수 있는 활용
    - 계산복잡도가 $O(n^2)$ 이상인 행렬 연산을 수행해야 함.
    - 따라서 특성 수(n)이 큰 경우 메모리 관리 및 시간복잡도 문제때문에 비효율적임.

- 방식 2: 경사하강법
    - 특성 수가 매우 크거나 훈련 샘플이 너무 많이 메모리에 모두 담을 수 없을 때 적합
    - 일반적으로 선형 회귀 모델 훈련에 적용되는 기법

4.2 경사 하강법

기본 아이디어

- 훈련 세트를 이용한 훈련 과정 중에 가중치 등과 같은 **파라미터를 조금씩 반복적으로 조정하기**
- 조정 기준: 비용 함수의 크기 줄이기

경사 하강법 관련 주요 개념

최적 학습 모델

- 비용함수를 최소화하는 또는 효용함수를 최대화하는 파라미터를 사용하는 모델
- 예제: 선형 회귀 모델

$$\hat{\mathbf{y}}^T = h_\theta(\mathbf{X}) = \theta^T \mathbf{X}_b^T$$

파라미터

- 예측값을 생성하는 함수로 구현되는 학습 모델에 사용되는 파라미터
- 예제: 선형 회귀 모델에 사용되는 편향과 가중치 파라미터

$$\theta = \theta_0, \theta_1, \ldots, \theta_n$$

비용함수

- 모델이 얼마나 나쁜지를 계산해주는 함수
- 최적 학습 모델은 비용함수가 최소가 되도록 하는 모델
- 예제: 선형 회귀 모델의 평균 제곱 오차(MSE)

$$\mathrm{MSE}(\mathbf{X}, h_\theta) = \frac{1}{m} \sum_{i=0}^{m-1} (\theta^T \, (\mathbf{x}_b^{(i)})^T - y^{(i)})^2$$

전역 최솟값

- 비용함수가 가질 수 있는 최솟값
- 예제: 선형 회귀 모델의 평균 제곱 오차(MSE) 함수가 갖는 최솟값

그레이디언트 벡터

- 다변수 함수의 미분값.
- 여러 개의 값으로 이루어진 벡터로 계산됨.
- (그레이디언트) 벡터는 방향과 크기에 대한 정보 제공
- 그레이디언트가 가리키는 방향의 반대 방향으로 움직여야 가장 빠르게 전역 최솟값에 접근

- 예제: 선형 회귀 MSE의 그레이디언트 벡터 $\nabla_\theta \mathrm{MSE}(\theta)$

$$\nabla_\theta \mathrm{MSE}(\theta) = \begin{bmatrix} \frac{\partial}{\partial \theta_0} \mathrm{MSE}(\theta) \\ \frac{\partial}{\partial \theta_1} \mathrm{MSE}(\theta) \\ \vdots \\ \frac{\partial}{\partial \theta_n} \mathrm{MSE}(\theta) \end{bmatrix} = \frac{2}{m} \mathbf{X}_b^T \left(\mathbf{X}_b \, \theta - \mathbf{y} \right)$$

학습률

- 훈련 과정에서의 비용함수 파라미터 조정 비율

학습률과 모델 학습

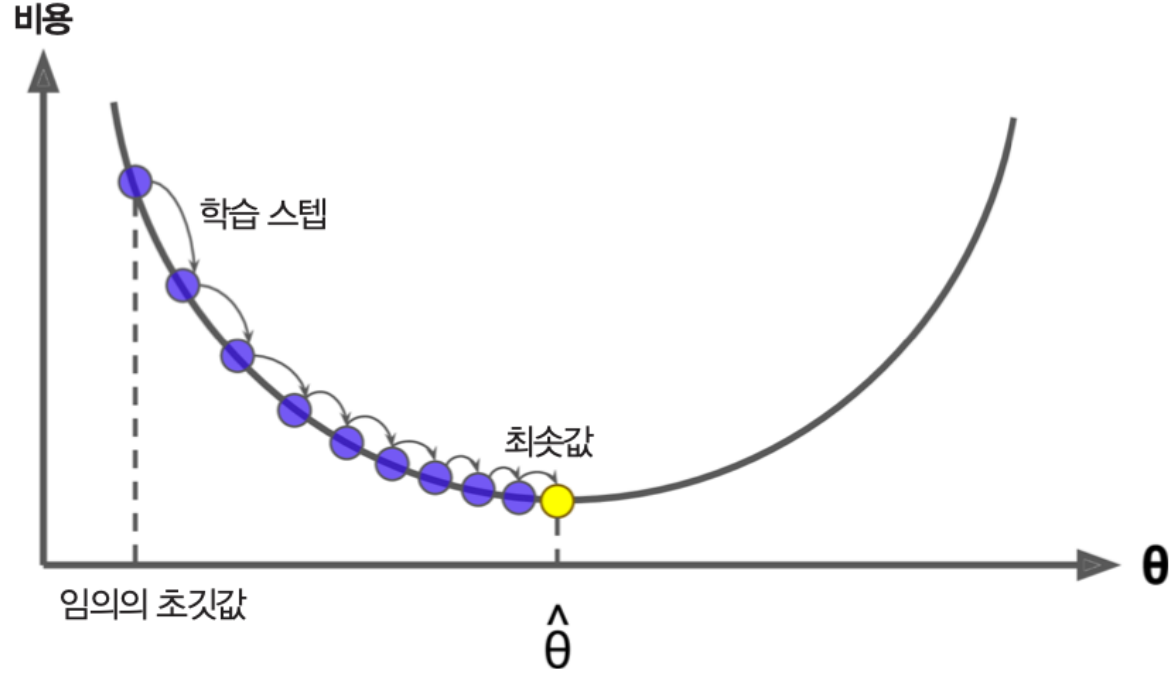
- 예제: 선형회귀 모델 파라미터 조정 과정

- θ를 임의의 값으로 지정한 후 훈련 시작

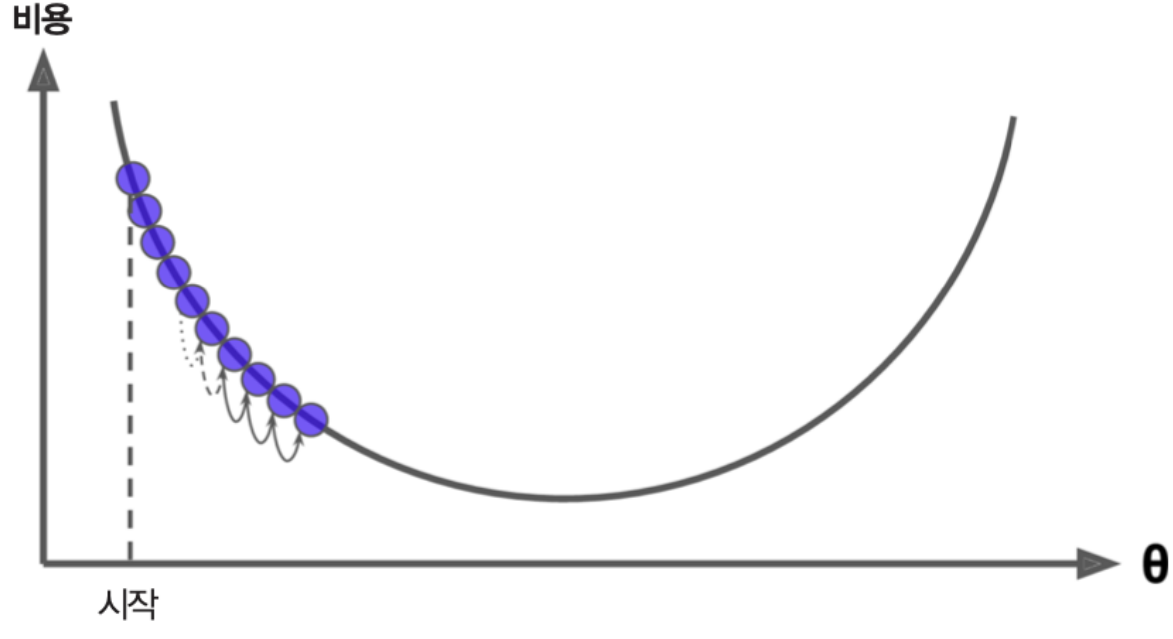
- 아래 단계를 θ가 특정 값에 지정된 오차범위 내로 수렴할 때까지 반복

    1. (배치 크기로) 지정된 수의 훈련 샘플을 이용하여 학습.
    2. 학습 후 $\mathrm{MSE}(\theta)$ 계산.
    3. 이전 θ에서 $\nabla_\theta \mathrm{MSE}(\theta)$과 학습률 η를 곱한 값 빼기.

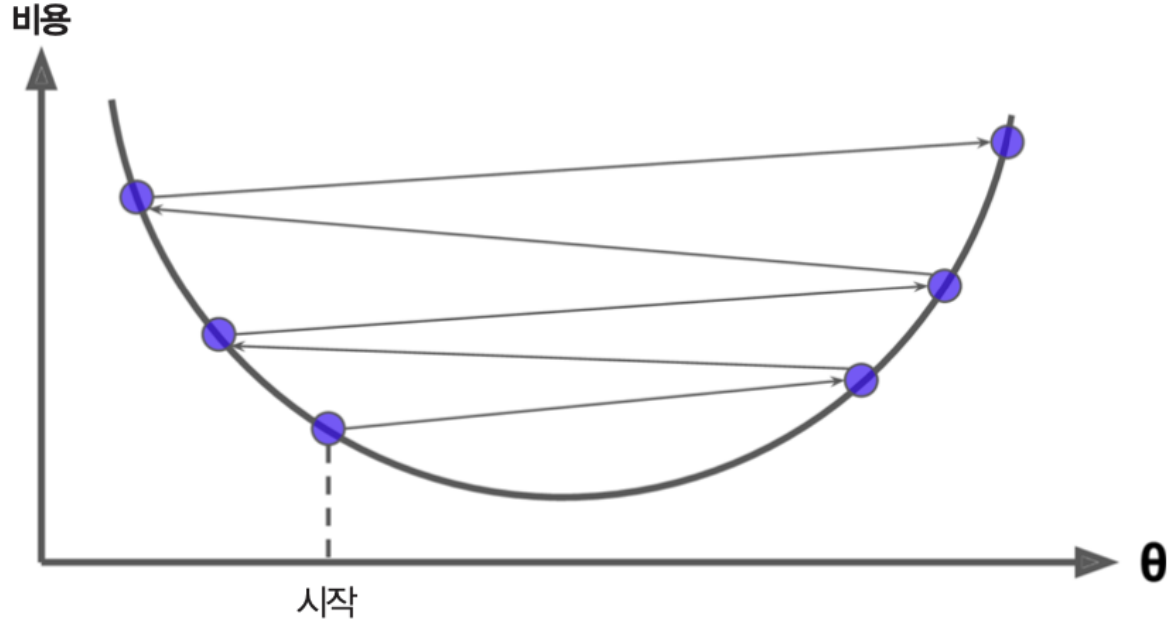
$$\theta^{(\text{new})} = \theta^{(\text{old})} - \eta \cdot \nabla_\theta \text{MSE}(\theta^{(\text{old})})$$



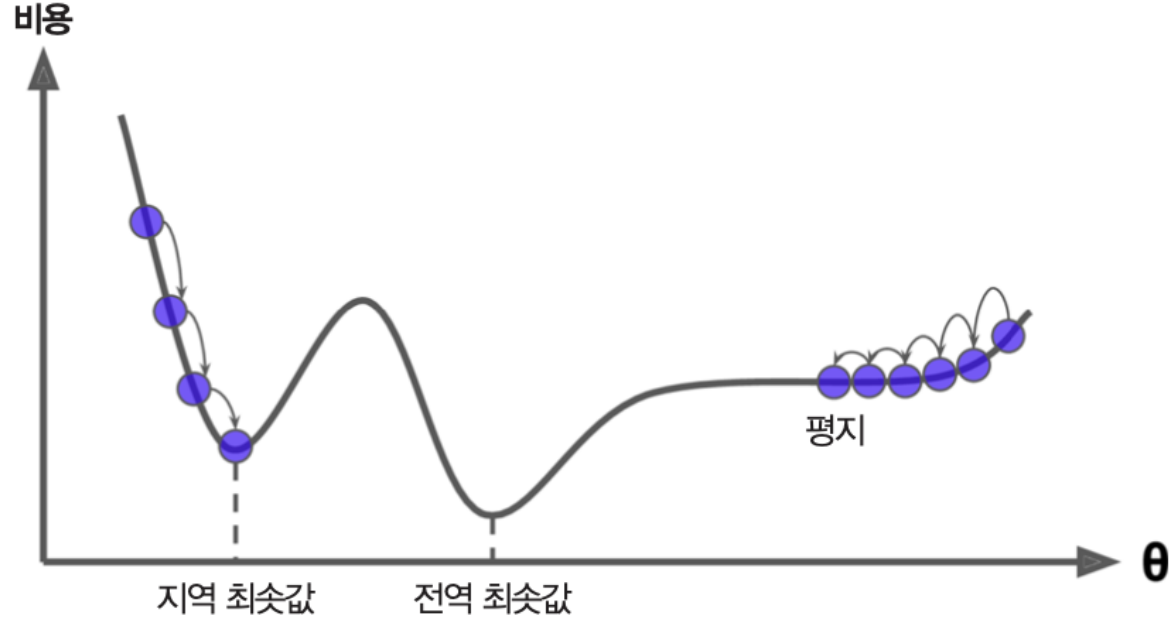
- 학습률이 너무 너무 작은 경우: 비용 함수가 전역 최소값에 너무 느리게 수렴.



- 학습률이 너무 큰 경우: 비용 함수가 수렴하지 않음.



- 시작점에 따라 지역 최솟값에 수렴할 수도 있음.



- 선형 회귀와 학습률
    - 비용함수(MSE)가 볼록 함수. 즉, 지역 최솟값을 갖지 않음
    - 따라서 학습률이 너무 크지 않으면 언젠가는 전역 최솟값에 수렴

하이퍼파라미터(hyperparameter)

- 학습 모델을 지정할 때 사용되는 값
- 예제: 학습률, 배치 크기, 에포크, 허용오차 등
    - 에포크(epoch): 훈련 세트 전체를 대상으로 훈련하는 단계
        - 에포크 수: 전체 훈렌 세트를 몇 번 반복 학습할 지 결정.
    - 배치(batch) 크기: 파라미터를 업데이트하기 위해, 즉 그레이디언트 벡터를 계산하기 위해 필요한 훈련 샘플 수.
    - 허용오차(tolerance): 비용함수의 그레이디언트 벡터의 크기가 허용오차보다 작아지면 학습 종료

스텝(step)

- 지정된 배치 크기의 샘플을 학습한 후에 파라미터를 조정하는 단계
- 예제: 훈련 세트의 크기가 1,000이고 배치 크기가 10이면, 하나의 에포크 기간동안 총 100번의 스텝이 실행됨.

- 스텝 크기 = (훈련 샘플 수) / (배치 크기)

- 경우에 따라 배치 크기 대신에 스텝 크기가 하이퍼파라미터로 주어짐.

배치 크기와 경사 하강법

배치 경사 하강법

- 배치 크기: 전체 훈련 샘플 수
- 즉, 에포크마다 그레이디언트를 계산하여 파라미터 조정
- **주의**: 여기서 사용되는 '배치'의 의미가 '배치 크기'의 '배치'와 다른 의미

확률적 경사 하강법

- 배치 크기: 1
- 즉, 하나의 훈련 샘플을 학습할 때마다 그레이디언트를 계산해서 파라미터 조정

미니배치 경사 하강법

- 배치 크기: 2에서 수백 사이
- 최적 배치 크기: 경우에 따라 다름. 여러 논문이 32 이하 추천

배치 경사 하강법

- 사이킷런은 배치 경사 하강법을 활용한 선형 회귀 미지원
    - 책 176쪽, 표 4-1에서 사이킷런의 SGDRegressor가 배치 경사 하강법을 지원한다고 **잘못** 명시됨.

에포크와 허용오차

- 에포크 수는 크게 설정한 후 허용오차를 지정하여 학습 시간 제한 필요
- 이유: 포물선의 최솟점에 가까워질 수록 그레이디언트 벡터의 크기가 0에 수렴
- 허용오차와 에포크 수는 서로 반비례의 관계
    - 즉, 오차를 1/10로 줄이려면 에포크 수를 10배 늘려야함

단점

- 훈련 세트의 크면 그레이디언트를 계산하는 데에 많은 시간 필요
- 아주 많은 데이터를 저장해야 하는 메모리 문제도 발생 가능

확률적 경사 하강법

장점

- 매우 큰 훈련 세트를 다룰 수 있음
    - 예를 들어, 외부 메모리(out-of-core) 학습을 활용할 수 있음
- 학습 과정이 매우 빠름
- 파라미터 조정이 불안정 할 수 있기 때문에 지역 최솟값에 상대적으로 덜 민감

단점

- 학습 과정에서 파라미터의 동요가 심할 수 있음
- 경우에 따라 전역 최솟값에 수렴하지 못하고 계속해서 발산할 가능성도 높음

학습 스케줄

- 요동치는 파라미터를 제어하기 위해 학습률을 학습 과정 동안 천천히 줄어들게 만들 수 있음

- 주의사항
    - 학습률이 너무 빨리 줄어들면, 지역 최솟값에 갇힐 수 있음
    - 학습률이 너무 느리게 줄어들면 전역 최솟값에 제대로 수렴하지 못하고 맴돌 수 있음

- 학습 스케줄(learning schedule)
    - 에포크, 훈련 샘플 수, 학습되는 샘플의 인덱스에 따른 학습률 지정

사이킷런의 `SGDRegressor`

- 경사 하강법 바로 지원
- 사용되는 하이퍼파라미터
    - `max_iter`: 에포크 수 제한
    - `tol`: 허용 오차
    - `eta0=0,1`: `SGDRegressor`가 사용하는 학습 스케줄 함수에 사용되는 매개 변수. 일종의 학습률.
    - `penalty`: 규제 사용 여부 결정 (추후 설명)

미니배치 경사 하강법

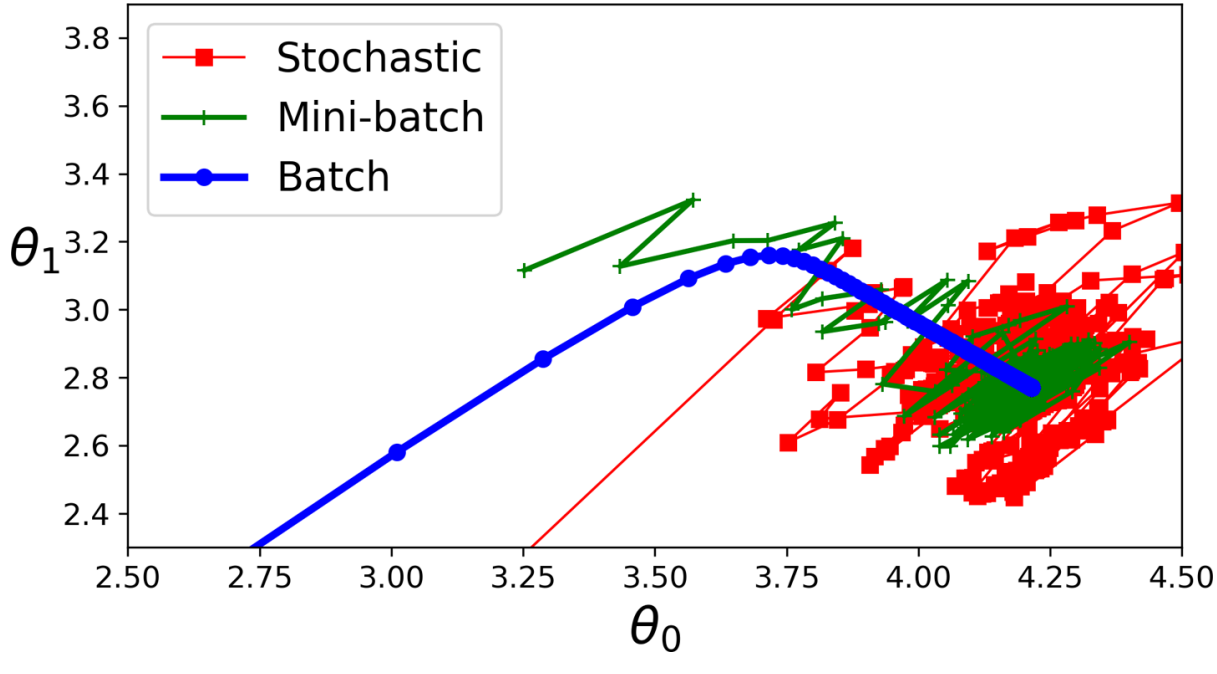
장점

- 배치 크기를 어느 정도 크게 하면 확률적 경사 하강법(SGD) 보다 파라미터의 움직임이 덜 불규칙적이 됨
- 반면에 배치 경사 하강법보다 빠르게 학습
- 학습 스케줄을 사용 가능

단점

- SGD에 비해 지역 최솟값에 수렴할 위험도 커짐.

경사 하강법 비교


Stochastic
Mini-batch
Batch
3.8
3.6
3.4
3.2
3.0
2.8
2.6
2.4
θ_1
2.50
2.75
3.00
3.25
3.50
3.75
4.00
4.25
4.50
θ_0

선형 회귀 알고리즘 비교

알고리즘	많은 샘플 수	외부 메모리 학습	많은 특성 수	하이퍼 파라미터 수	스케일 조정	사이킷런 지원
정규방정식	빠름	지원 안됨	느림	0	불필요	지원 없음
SVD	빠름	지원 안됨	느림	0	불필요	LinearRegression
배치 GD	느림	지원 안됨	빠름	2	필요	LogisticRegression
SGD	빠름	지원	빠름	>= 2	필요	SGDRegressor
미니배치 GD	빠름	지원	빠름	>=2	필요	지원 없음

4.3 다항 회귀

- 선형 회귀를 이용하여 비선형 데이터를 학습하는 기법
- 즉, 비선형 데이터를 학습하는 데 선형 모델 사용을 가능하게 함.

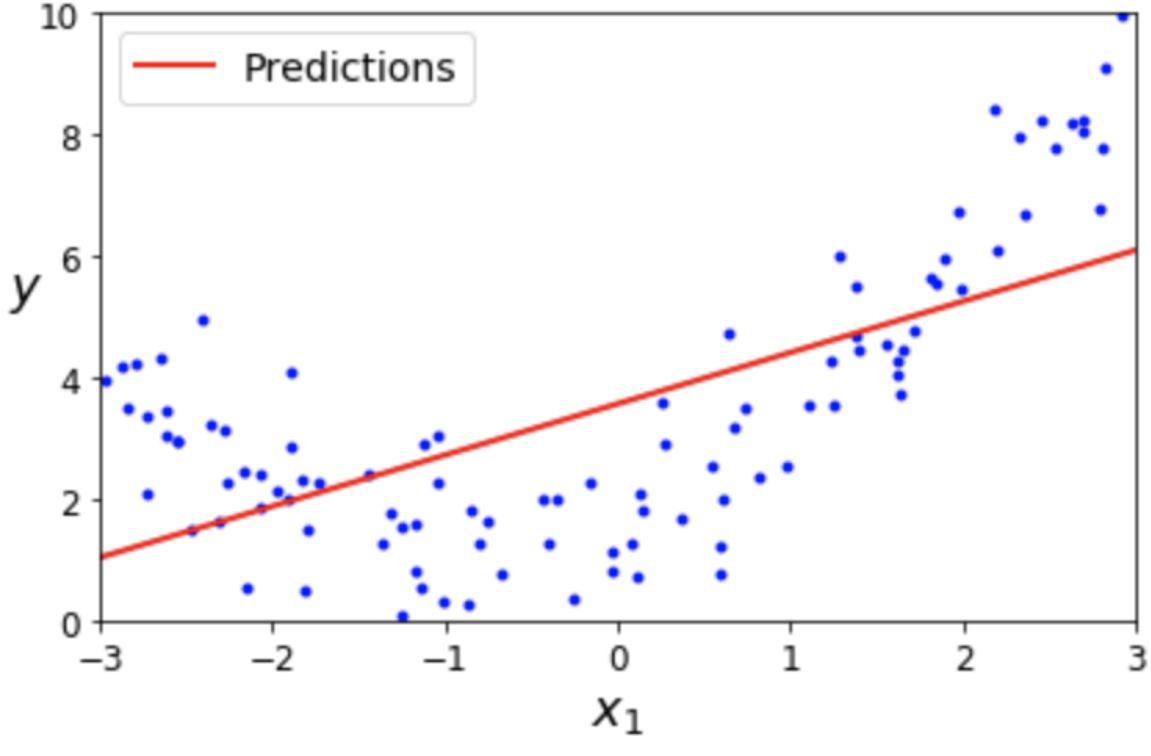
다항 회귀 기본 아이디어

- 특성 조합 활용
- 추가되는 특성은 기존의 특성값들의 거듭제곱, 특성값들 사이의 곱 등으로 이루어짐
- 즉, 특성 변수들의 다항식을 조합 특성으로 추가

선형 회귀 vs. 다항 회귀

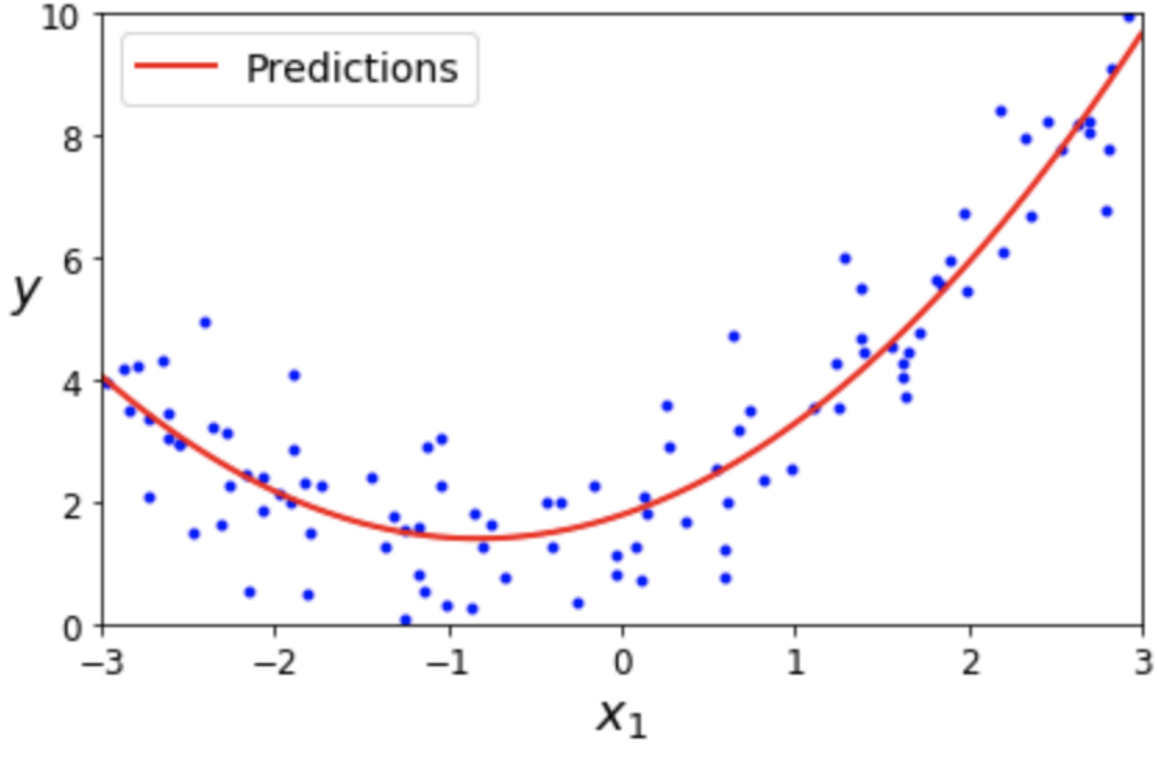
선형 회귀

- 1차 선형 모델: $\hat{y} = \theta_0 + \theta_1 x_1$



다항 회귀

- 2차 다항식 모델: $\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2$



사이킷런의 `PolynomialFeatures` 변환기

- 주어진 특성들의 거듭제곱과 특성들 사이의 곱셈을 실행하여 특성을 추가하는 기능 제공
- `degree=d` 하이퍼파라미터 지정

- 예를 들어 $n = 2, d = 3$인 경우: $(x_1 + x_2)^2 + (x_1 + x_2)^3$ 의 항목에 해당하는 7개 특성 추가

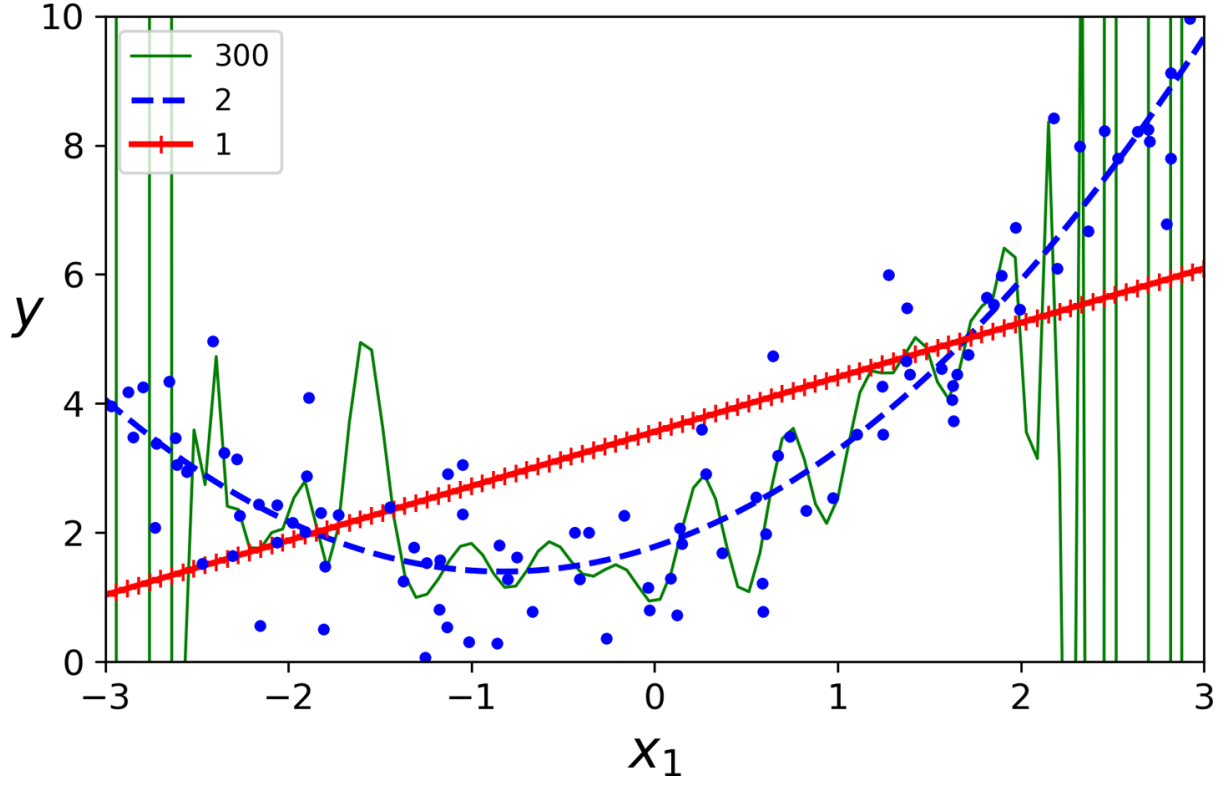
$$x_1^2,\ \ x_1 x_2,\ \ x_2^2,\ \ x_1^3,\ \ x_1^2 x_2,\ \ x_1 x_2^2,\ \ x_2^3$$

- 위 예제에서는 $n = 1, d = 2$ 이기에 x_1^2에 대한 특성 변수만 추가됨.

4.4 학습 곡선

과소적합/과대적합 판정

- 예제: 선형 모델, 2차 다항 회귀 모델, 300차 다항 회귀 모델 비교
- 다항 회귀 모델의 차수에 따라 훈련된 모델이 훈련 세트에 과소 또는 과대 적합할 수 있음.

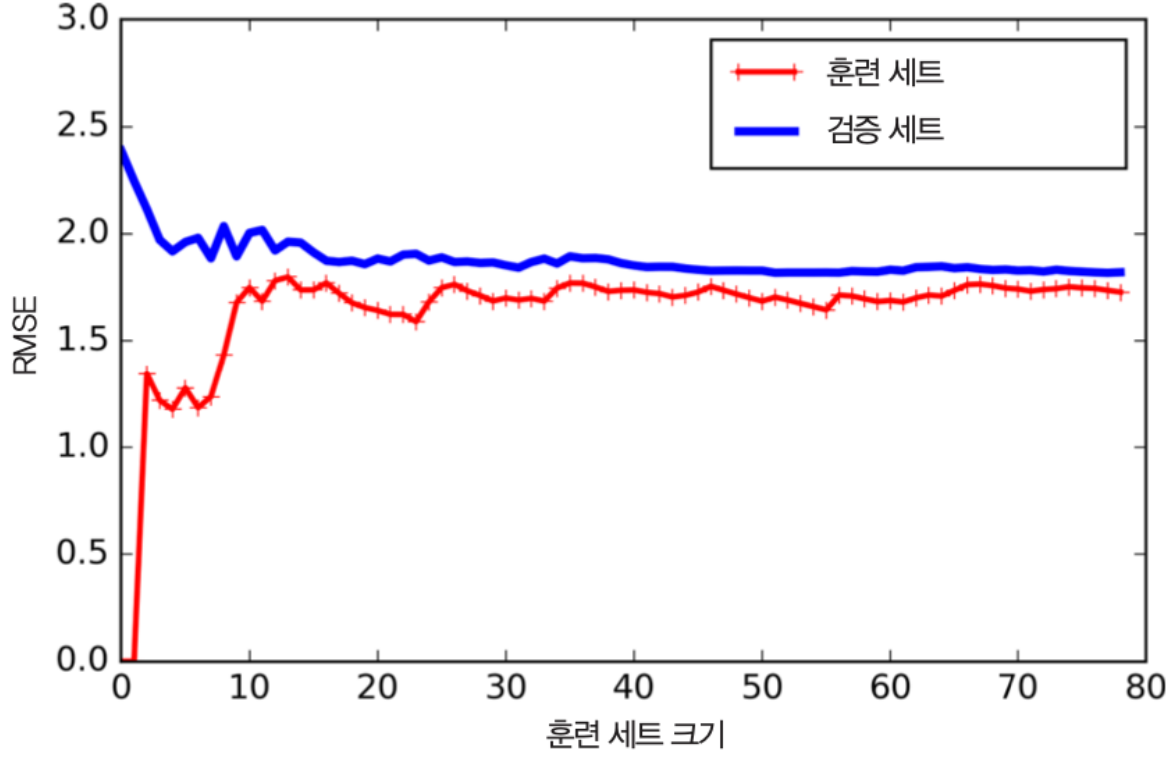


교차 검증 vs. 학습 곡선

- 교차 검증(2장)
    - 과소적합: 훈련 세트와 교차 검증 점수 모두 낮은 경우
    - 과대적합: 훈련 세트에 대한 검증은 우수하지만 교차 검증 점수가 낮은 경우

- 학습 곡선 살피기
    - 학습 속선: 훈련 세트와 검증 세트에 대한 모델 성능을 비교하는 그래프
    - 학습 곡선의 모양에 따라 과소적합/과대적합 판정 가능

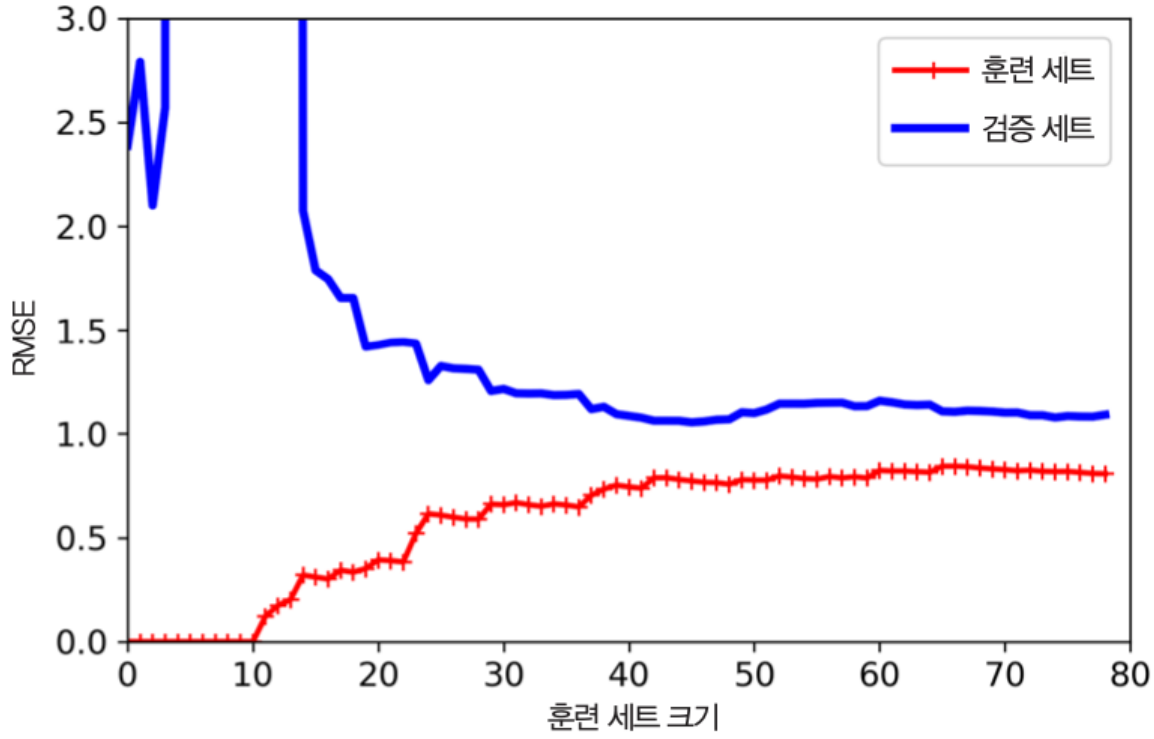
과소적합 모델의 학습 곡선 특징



- 훈련 데이터에 대한 성능
  - 훈련 세트가 커지면서 RMSE(평균 제곱근 오차)가 커짐
  - 훈련 세트가 어느 정도 커지면 더 이상 RMSE가 변하지 않음

- 검증 데이터에 대한 성능
    - 검증 세트에 대한 성능이 훈련 세트에 대한 성능과 거의 비슷해짐

과대적합 모델의 학습 곡선 특징



- 훈련 데이터에 대한 성능
    - 훈련 데이터에 대한 평균 제곱근 오차가 다른 모델보다 훨씬 적다.

- 검증 데이터에 대한 성능
    - 훈련 데이터에 대한 성능과 차이가 크게 벌어진다. 즉, 훈련 데이터에 대한 성능이 훨씬 좋다.

과대적합 모델 개선법

- 검증 데이터에 대한 성능이 훈련 데이터에 대한 성능에 근접할 때가지 훈련 데이터의 크기 늘리기.

4.5 규제 선형 모델

자유도와 규제

- 자유도(degree of freedom): 학습 모델 결정에 영향을 주는 요소(특성)들의 수
    - 단순 선형 회귀의 경우: 특성 수
    - 다항 선형 회귀 경우: 차수

- 규제(regularization): 자유도 제한
    - 단순 선형 회귀 모델에 대한 규제: 가중치 역할 제한
    - 다항 선형 회귀 모델에 대한 규제: 차수 줄이기

가중치 역할 규제 선형 회귀 모델

- 릿지 회귀
- 라쏘 회귀
- 엘라스틱넷

규제 적용 주의사항

규제항은 훈련 과정에만 사용된다. 테스트 과정에는 다른 기준으로 성능을 평가한다.

- 훈련 과정: 비용 최소화 목표
- 테스트 과정: 최종 목표에 따른 성능 평가
    - 예제: 분류기의 경우 재현율/정밀도 기준으로 성능 평가

릿지 회귀

릿지 회귀의 비용함수

$$J(\theta) = \text{MSE}(\theta) + \alpha \frac{1}{2} \sum_{i=1}^{n} \theta_i^2$$

- α(알파): 규제 강도를 지정하는 하이퍼파라미터
- $\alpha = 0$: 단순 선형 회귀
- α가 커질 수록 가중치의 역할이 줄어듦.
    - 비용을 줄이기 위해 가중치를 작게 유지하는 방향으로 학습
- 주의사항: 훈련 세트에 대한 특성 스케일링 전처리 실행 후 적용

라쏘 회귀

라쏘 회귀의 비용함수

$$J(\theta) = \text{MSE}(\theta) + \alpha \sum_{i=1}^{n} | \theta_i |$$

- α(알파)
    - 하이퍼파라미터로 지정됨.
    - 규제 강도 지정
    - $\alpha = 0$이면 규제가 전혀 없는 기본 선형 회귀
- θ_i: 덜 중요한 특성을 무시하기 위해 $|\theta_i|$가 0에 수렴하도록 학습 유도.
- **주의**: θ_0은 규제하지 않음

엘라스틱넷

- 릿지 회귀와 라쏘 회귀를 절충한 모델

엘라스틱넷의 비용함수

$$J(\theta) = \text{MSE}(\theta) + r\,\alpha \sum_{i=1}^{n} \mid \theta_i \mid + \frac{1-r}{2}\,\alpha \sum_{i=1}^{n} \theta_i^2$$

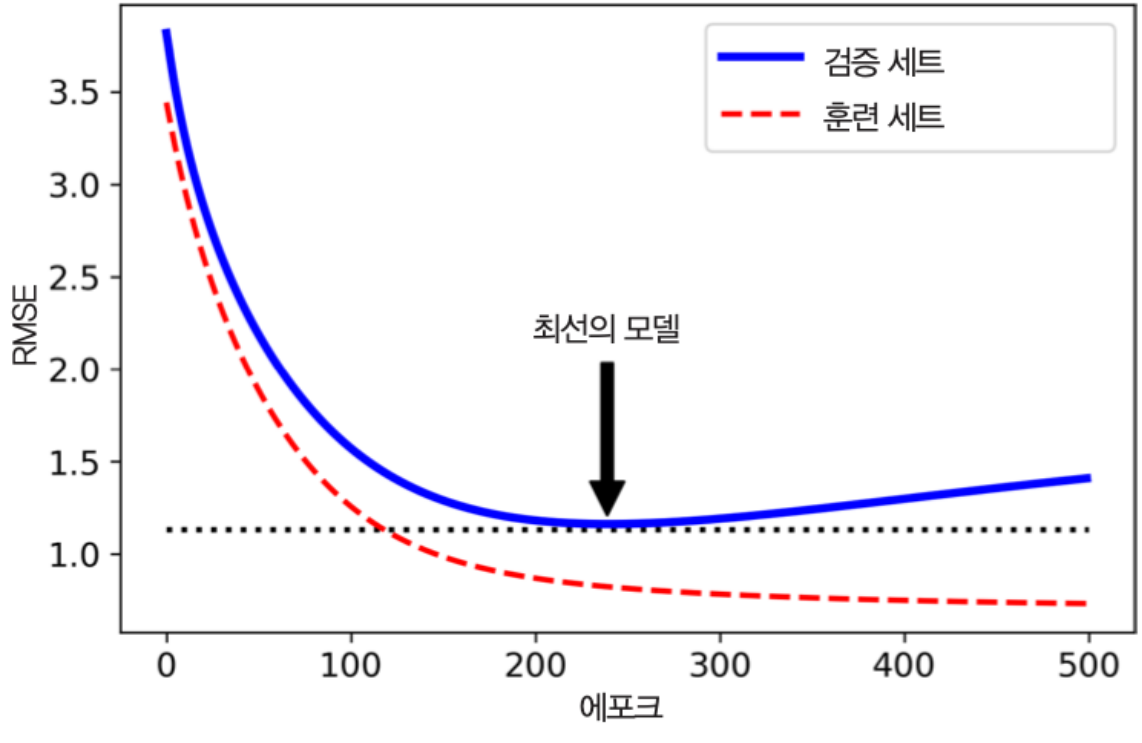
- r을 이용하여 릿지 규제와 라쏘 규제를 적절하게 조절

규제 사용 방법

- 대부분의 경우 약간이라도 규제 사용 추천
- 릿지 규제가 기본
- 유용한 속성이 많지 않다고 판단되는 경우
    - 라쏘 규제나 엘라스틱넷 활용 추천
    - 불필요한 속성의 가중치를 0으로 만들기 때문
- 특성 수가 훈련 샘플 수보다 크거나 특성 몇 개가 강하게 연관되어 있는 경우
    - 라쏘 규제는 적절치 않음.
    - 엘라스틱넷 추천

조기 종료 기법

- 반복 훈련 과정 중에 모델이 훈련 데이터에 점점 더 익숙해져서 과대적합 발생 가능
- 따라서 반복 훈련을 적절한 시기에 종료해야 함
- 반복훈련 종료 기준: 검증 데이터에 대한 손실이 줄어 들다가 다시 커지는 순간
- 조기 종료: 검증 오차가 최소에 다다렀을 때 반복 훈련을 멈추게 하는 기법



- 확률적 경사 하강법, 미니배치 경사 하강법의 경우 손실 곡선이 매끄럽지 않고 진동 발생 가능
- 이런 경우에는 검증 오차가 한동안 최솟값보다 높게 유지될 때 반복 훈련을 멈추고 검증 오차가 최소였을 때의 모델 파라미터 확인

4.6 로지스틱 회귀와 소프트맥스 회귀

로지스틱 회귀와 소프트맥스 회귀를 이용하여 분류 모델 학습 가능

- 이진 분류: 로지스틱 회귀 활용
- 다중 클래스 분류: 소프트맥스 회귀 활용

로지스틱 회귀와 시그모이드 함수

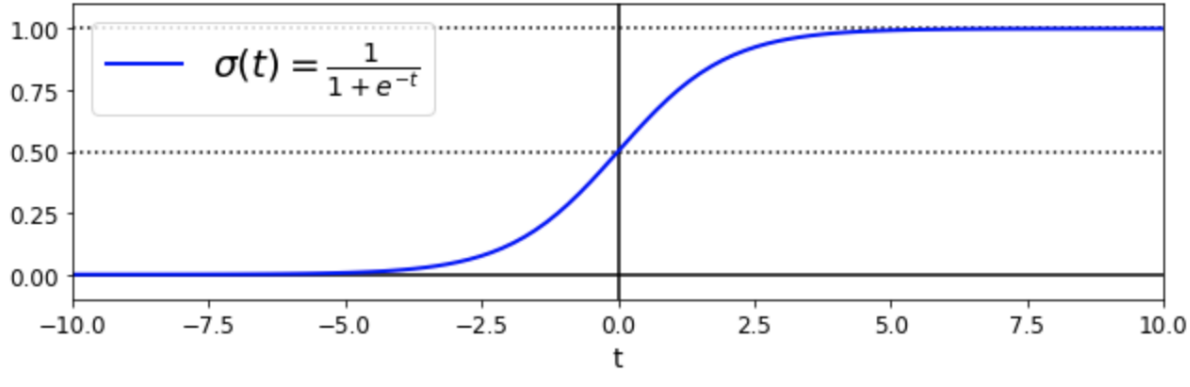
로지스틱 회귀

- 특성과 가중치의 곱한 값들을 더한 결과에 **시그모이드 함수**를 적용한 결과 이용
- 로지스틱 회귀 모델에서 샘플 $\mathbf{x}$에 대한 예측값

$$\hat{p} = h_\theta(\mathbf{x}) = \sigma(\theta^T \mathbf{x}_b^T)$$

시그모이드 함수

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



로지스틱 회귀 모델의 예측값

$$\hat{y} = \begin{cases} 0 & \text{if } \hat{p} < 0.5 \\ 1 & \text{if } \hat{p} \geq 0.5 \end{cases}$$

- $\theta^T \mathbf{x}_b^T \geq 0$ 인 경우: 양성 클래스(1)
- $\theta^T \mathbf{x}_b^T < 0$ 인 경우: 음성 클래스(0)

로지스틱 회귀 모델의 비용함수

- 로지스틱 회귀 모델을 경사하강법을 이용하여 학습

- 비용함수: 로그 손실(log loss) 함수 사용

$$J(\theta) = -\frac{1}{m} \sum_{i=0}^{m-1} [y^{(i)} \ \log(\ \hat{p}^{(i)}\ ) + (1 - y^{(i)})\ \log(\ 1 - \hat{p}^{(i)}\ )]$$

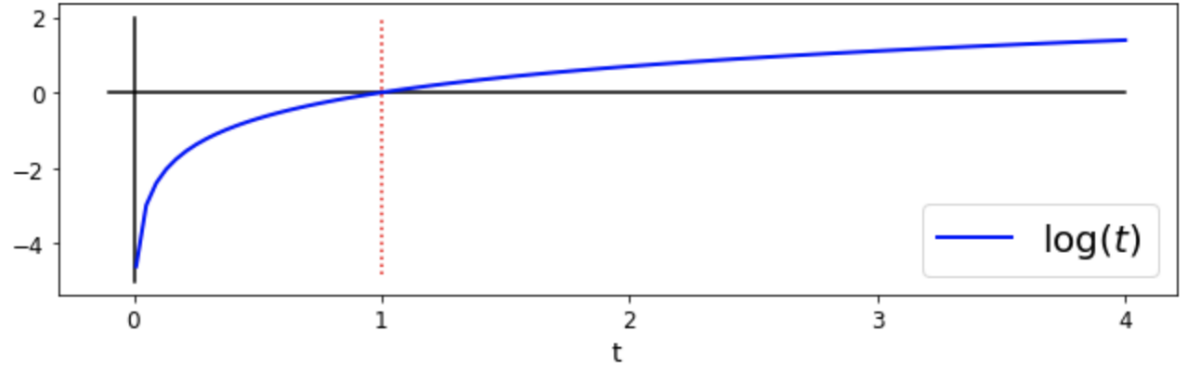
- 이 비용함수에 대해 경사 하강법 적용

- 로그 손실 함수 이해: 하나의 샘플에 대한 아래의 값의 의미 이해 중요

$$-[y^{(i)} \ \log(\ \hat{p}^{(i)}\ ) + (1 - y^{(i)})\ \log(\ 1 - \hat{p}^{(i)}\ )]$$

- 틀린 예측을 하면 값이 커짐.

- log 함수 성질 참조



결정 경계: 로지스틱 회귀 활용 예제

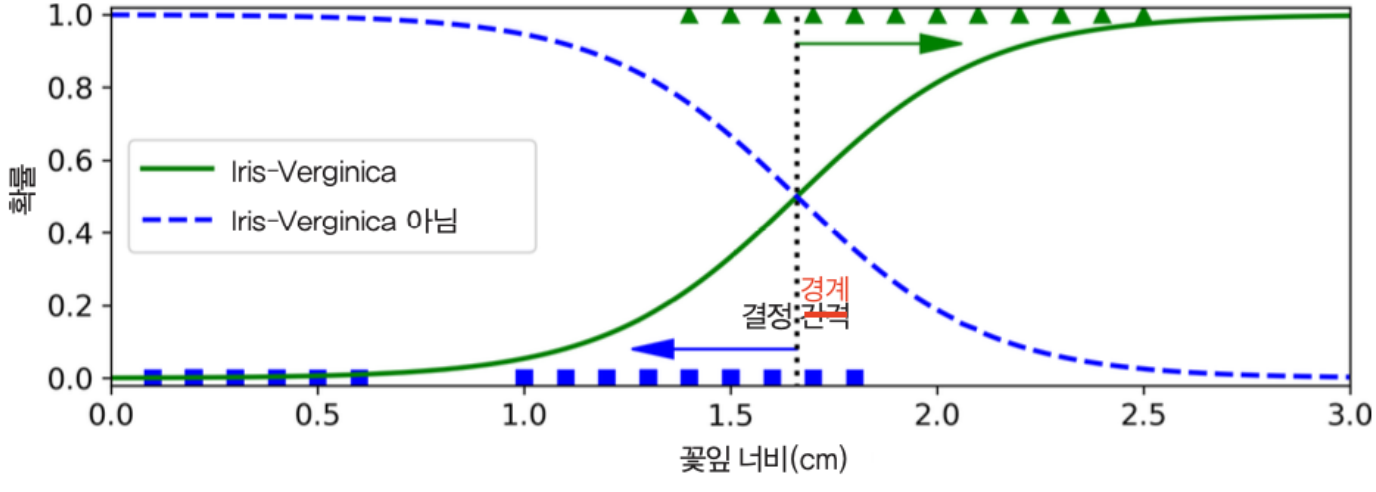
사이킷런에서 제공하는 붓꽃 데이터셋 활용

- 4개의 특성 사용
    - 꽃받침 길이
    - 꽃받침 너비
    - 꽃잎 길이
    - 꽃잎 너비

- 샘플 타깃
    - 0: Iris-Setosa
    - 1: Iris-Versicolor
    - 2: Iris-Virginica

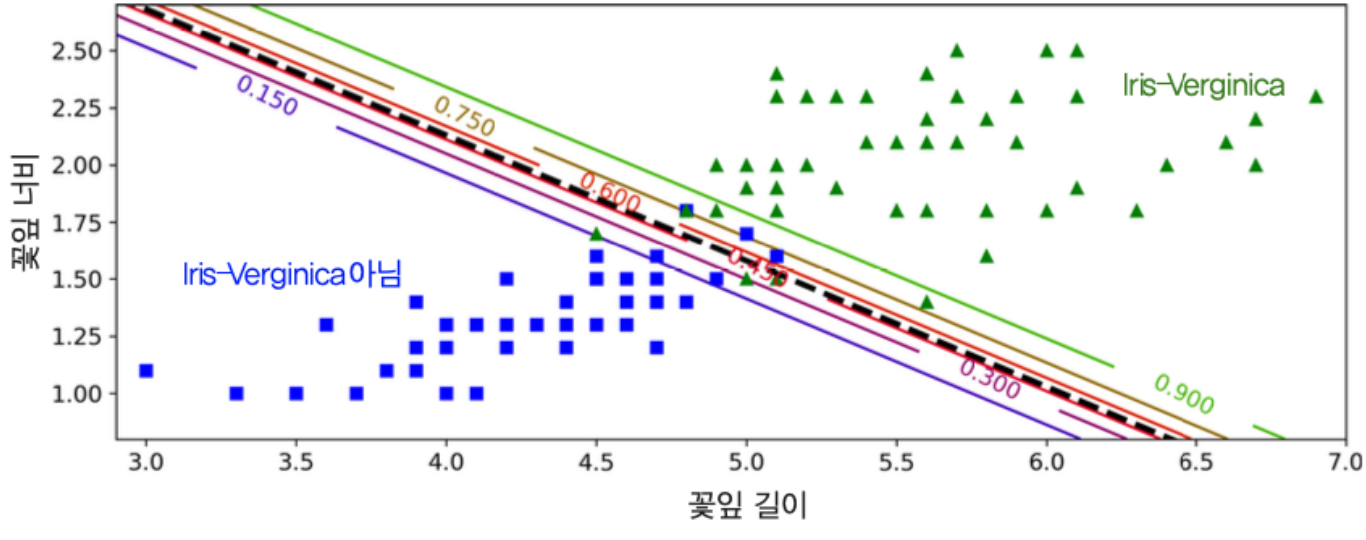
꽃잎의 너비를 기준으로 Iris-Virginica 여부 판정하기

- 결정경계: 약 1.6cm



꽃잎의 너비와 길이를 기준으로 Iris-Virginica 여부 판정하기

- 결정경계: 검정 점선



로지스틱 회귀 규제하기

- 하이퍼파라미터 `penalty`와 `C` 이용

- `penalty`
    - `l1`, `l2`, `elasticnet` 세 개중에 하나 사용.
    - 기본은 `l2`, 즉, ℓ_2 규제를 사용하는 릿지 규제.
    - `elasticnet`을 선택한 경우 `l1_ration` 옵션 값을 지정해서 함께 사용.

- `C`
    - 릿지 규제 정도를 지정하는 α의 역수에 해당.
    - 따라서 0에 가까울 수록 강한 규제 의미.

소프트맥스(softmax) 회귀

- 로지스틱 회귀 모델을 일반화하여 다중 클래스 분류를 지원하도록 한 회귀 모델
- **다항 로지스틱 회귀** 라고도 불림

소프트맥스 회귀 학습 아이디어

- 샘플 $\mathbf{x}$이 주어졌을 때 각각의 분류 클래스 k dp 대한 점수 $s_k(\mathbf{x})$ 계산

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

- 소프트맥스 함수를 이용하여 각 클래스 k에 속할 확률 $\hat{p}_k$ 계산

$$\hat{p}_k = \frac{\exp(s_k(\mathbf{x}))}{\sum_{j=0}^{K-1} \exp(s_j(\mathbf{x}))}$$

- 추정 확률이 가장 높은 클래스 선택

$$\hat{y} = \text{argmax}_k \, s_k(\mathbf{x})$$

주의사항

- 소프트맥스 회귀는 다중 출력 분류 지원 못함.
- 예를 들어, 하나의 사진에서 여러 사람의 얼굴 인식 불가능.

소프트맥스 회귀 비용함수

- 각 분류 클래스 k에 대한 적절한 가중치 벡터 θ_k를 학습해 나가야 함.

- 비용함수: 크로스 엔트로피 비용 함수 사용

$$J(\Theta) = -\frac{1}{m} \sum_{i=0}^{m-1} \sum_{k=0}^{K-1} y_k^{(i)} \ \log(\hat{p}_k^{(i)})$$

- 이 비용함수에 대해 경사 하강법 적용

참조

- $K = 2$이면 로지스틱 회귀의 로그 손실 함수와 정확하게 일치한다.
- 주어진 샘플의 타깃 클래스를 제대로 예측할 경우 높은 확률값 계산
- 크로스 엔트로피 개념은 정보 이론에서 유래하였다. (자세한 설명은 생략)

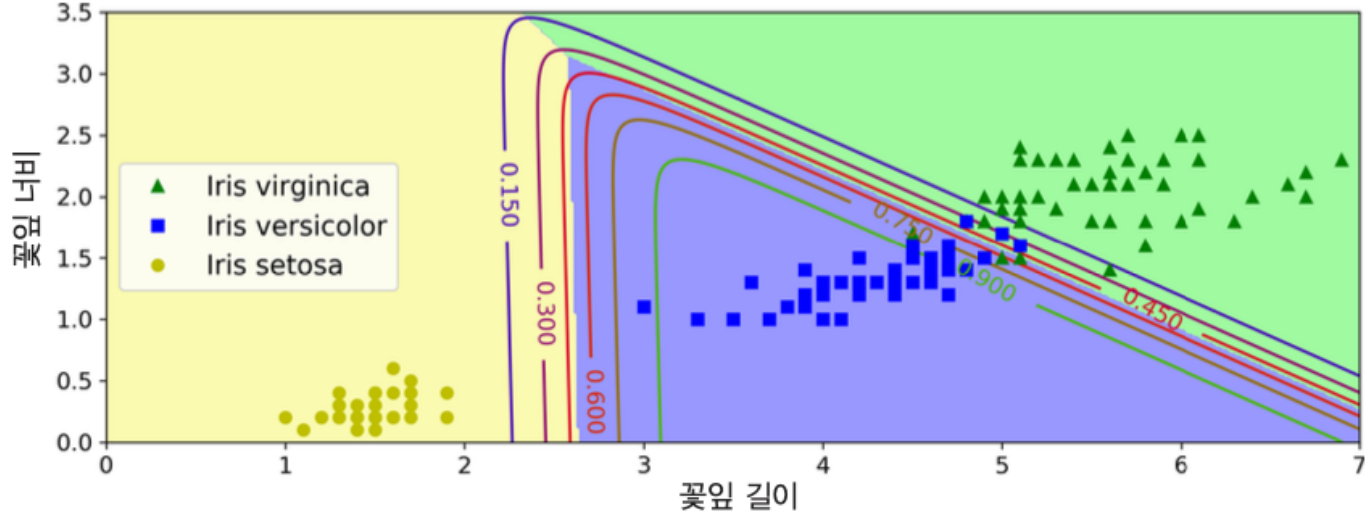
멀티 클래스 분류: 소프트맥스 회귀 활용 예제

- 붓꽃을 세 개의 클래스로 분류하기

- 사이킷런의 `LogisticRegression` 예측기 활용
    - `multi_class` 하이퍼파라미터 값을 `multinomial`로 지정

꽃잎의 너비와 길이를 기준으로 붓꽃 클래스 분류

- 결정경계: 배경색으로 구분
- 곡선: Iris-Versicolor 클래스에 속할 확률



- 예제: 꽃잎 길이 5cm, 너비 2cm 인 붓꽃에 대한 품종 클래스 추정
  - 94.2%의 확률로 Iris-Virginica
  - 또는 5.8%의 확률로 Iris-Versicolor