〈데이터분석과기계학습 2주차〉

간단한 분류 알고리즘

인공지능융합공학부 데이터사이언스전공

곽찬희

앞으로의 강의 방향성

- 겁먹지 맙시다 ☺
- 최대한 수학에 대한 부담감을 줄여드리도록 노력하겠습니다.
- 원리의 이해에 집중하세요!
- 책을 꼭 읽읍시다 + 코드는 반드시 따라합시다.

- 강의는 다음과 같이 구성됩니다.
  1. 그림과 개념 위주의 설명
  2. 수식으로 보충하는 핵심 개념
  3. 코드 (될 수 있으면 line-by-line 으로)

데이터 분석&머신러닝 주제 잡기

1. 내가 관심 갖는 주제/분야에서 출발하기
   - ✓ 금융, 소비자 행동, 스포츠…
   - ✓ (혹은 방법론적으로) 이미지 인식, 음성 인식, 시계열 분석 등

2. 다음의 빈칸을 채워보기
   - ✓ (A) 가 (B)의 예측/분류에 도움이 될까? 만약 그렇다면, 다른 요인들은 어떤 것들이 있을까?
   - ✓ 특정 이벤트 (A) 가 (B) 에 어떤 영향을 끼칠까?

데이터 분석&머신러닝 주제 잡기

- 데이터와 주제를 같이 생각해야 합니다.
  - ✓ 주제는 있는데 해당 주제를 탐구할 데이터가 없으면 꽝
  - ✓ 데이터는 있는데 연구할 주제가 없다면 역시 꽝
  - ✓ 고기가 있어야 스테이크를 만든다.
- 데이터는 어떻게?
  - ✓ 공공데이터
  - ✓ 크롤링
  - ✓ 진행 중인 공모전 등등
- 단일 데이터가 아닌 여러 데이터를 연결하면 새로운 가치를 가진 데이터를 만듦
  - ✓ Ex) 아이스크림 매출 데이터 분석 시 : 판매량 vs. 판매량 + SNS 검색량 + 날씨 + 대체재/보완재 +…

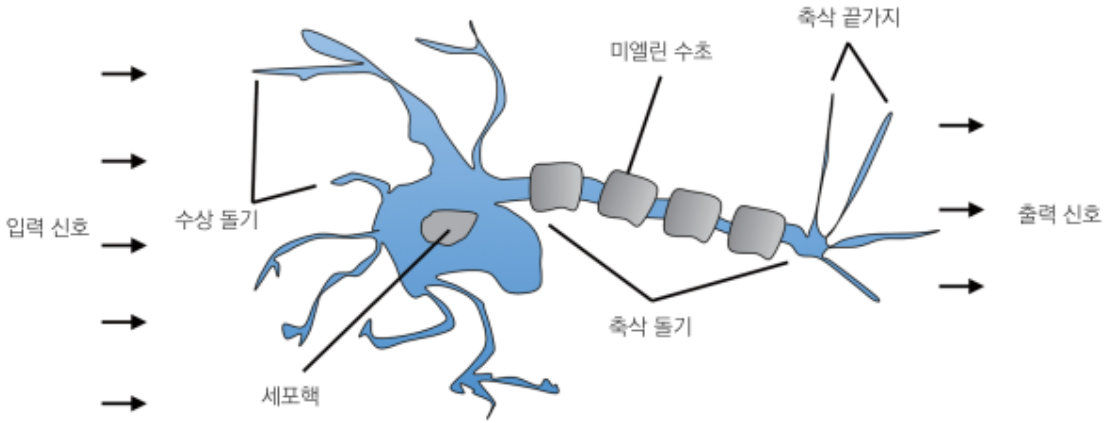
하지 말아야 할 것

- 너무 단순한/당연한 문제
  - ✓ 실적이 좋으면 주가가 좋다, 좋은 뉴스가 나오면 주가가 오를 것이다…
- 너무 단순한 기법
  - ✓ 감성분석을 통한 리뷰 분석 (제가 수도 없이 봐온 주제…)
  - ✓ 시각화만을 통한 분석 (여러분은 2학년이 아닙니다)
  - ✓ 변수가 너무 적은 분석 : $y = ax1 + bx2 + c$
- 데이터가 너무 적은 경우
  - ✓ 단순한 데이터에서는 할 얘기가 없어집니다.

2.1. 인공 뉴런 - 초기 머신 러닝의 간단한 역사

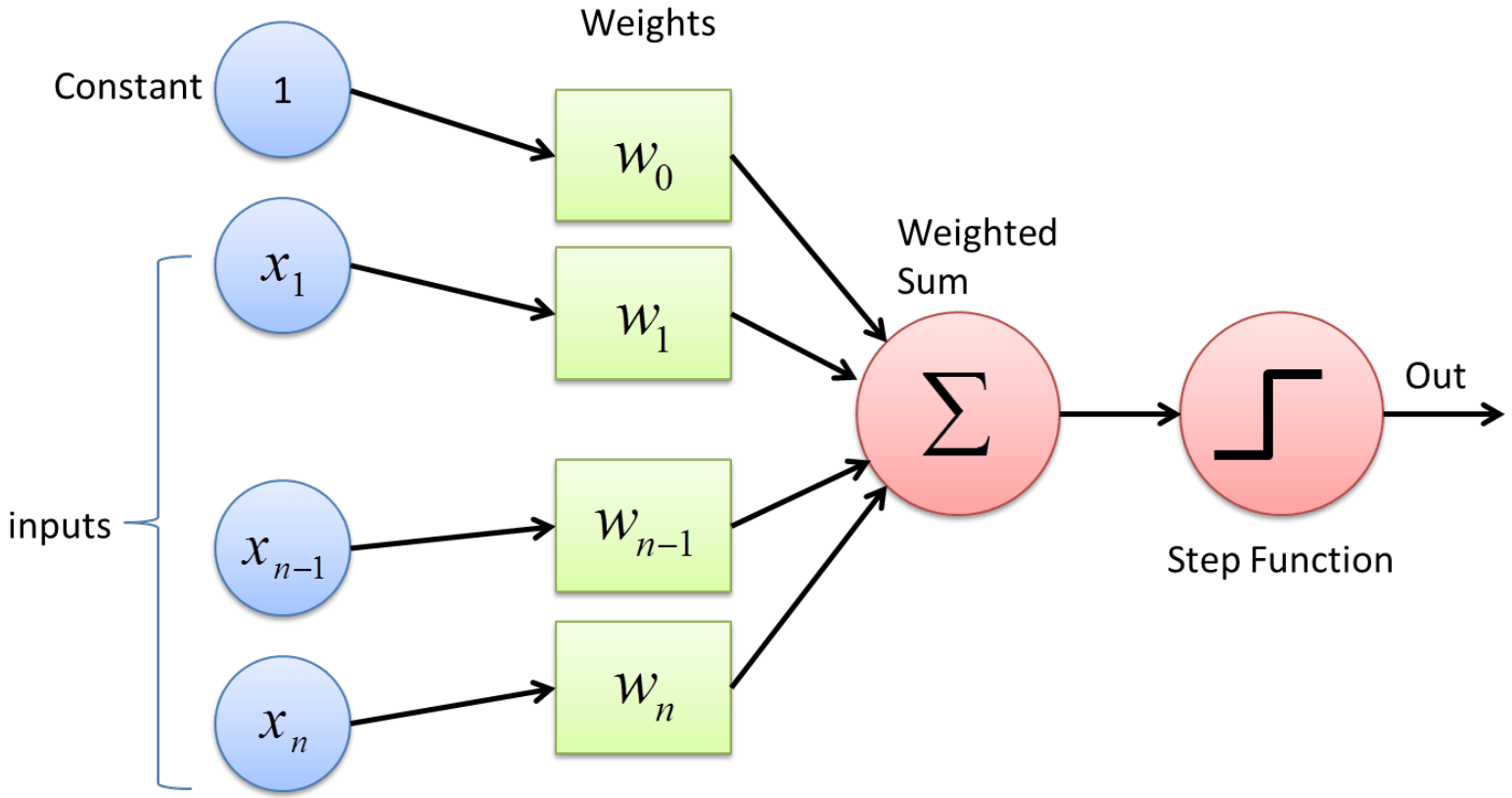
- 생물의 뇌에서 AI 설계의 힌트를 얻음
- 맥컬록-피츠(MCP) 뉴런
  - ✓ 뇌의 화학적, 전기적 신호 처리 구조를 개념화
  - ✓ 특정 임계값을 넘으면 신호 전달

그림 2-1 뇌의 신경 세포



- 퍼셉트론 (Perceptron)
  - ✓ 뉴런 구조를 모방한 학습모델
  - ✓ 퍼셉트론 규칙에서 최적의 가중치를 구하는 알고리즘이 제안됨

2.1.1. 인공 뉴런의 수학적 정의



$$w = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_m \end{bmatrix}, \; x = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{bmatrix}$$

$$w = w^T \cdot x$$

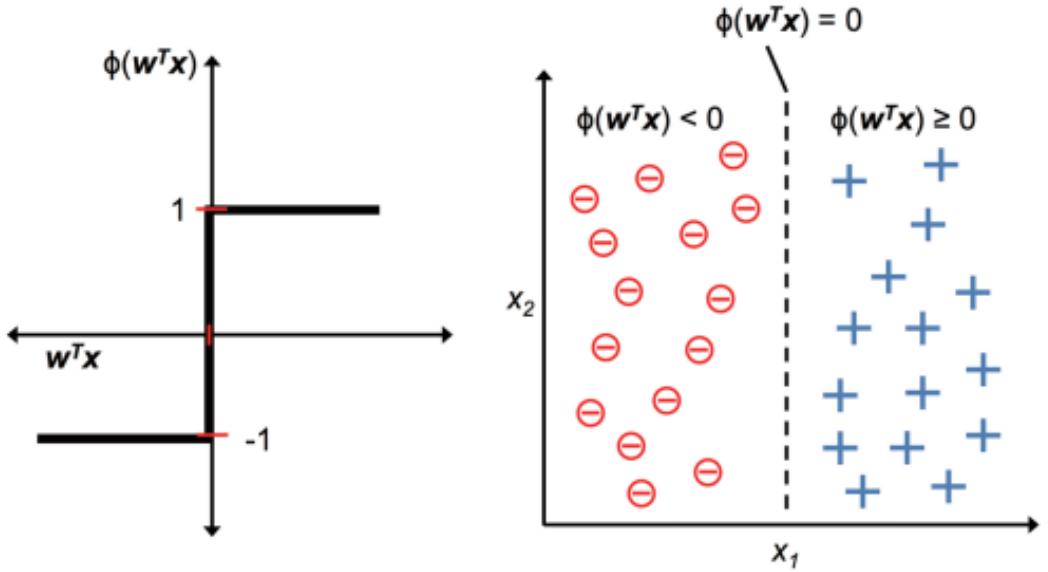
$$\Phi(z) = \begin{cases} 1 \; z \geq 0 \text{일 때} \\ -1 \; \text{그 외} \end{cases}$$

https://miro.medium.com/max/700/1*n6sJ4yZQzwKL9wnF5wnVNg.png

2.1.1. 인공 뉴런의 수학적 정의

- 퍼셉트론에서 결정 함수 ($\Phi(z)$) 는 임계값(θ)을 기준으로 그 이상일 때 클래스1로, 아닐 때 클래스 -1로 예측함
  - ✓ Step Function

그림 2-2 퍼셉트론의 결정 함수와 결정 경계



2.1.2. 퍼셉트론 학습 규칙

- 환원주의(reductionism, 복잡하고 추상적인 개념을 간단하고 명확하게 정의) 접근 방식에 따라 출력을 내거나 내지 않는 경우만 고려
- 위의 접근에 따라 다음의 순서로 퍼셉트론 학습
  1. 가중치를 0 또는 랜덤한 작은 값으로 초기화
  2. 각 훈련 샘플 x^i 에 대해 다음의 작업을 수행
     a. 출력값 $\hat{y}$ 계산 (계단 함수로 예측한 클래스 레이블)
     b. 가중치 업데이트 (η 는 에타로 읽으며 학습률을 의미. 0.0~1.0 사이의 실수)

$$w_j = w_j + \Delta w_j$$

$$\Delta w_j = \eta\left(y^i - \hat{y}^i\right)x_j^i$$

2.1.2. 퍼셉트론 학습 규칙

- **예측이 맞는 경우**

$$\Delta w_j = \eta(-1 - -1)x_j^i = 0$$
$$\Delta w_j = \eta(1 - 1)x_j^i = 0$$

- **예측이 틀린 경우**

$$\Delta w_j = \eta(1 - -1)x_j^i = \eta(2)x_j^i$$
$$\Delta w_j = \eta(-1 - 1)x_j^i = \eta(-2)x_j^i$$

- **예측이 틀렸을 때, 가중치의 변화는 x에 비례**

2.1.2. 퍼셉트론 학습 규칙

- 퍼셉트론은 두 클래스가 선형적으로 구분되고 학습률이 충분히 작을 때만 수렴 보장
- 선형 결정 경계로 나눌 수 없다면, 훈련을 반복할 최대 횟수 (epoch, 에포크)를 지정하고 분류 허용 오차를 지정할 수 있음
  - ✓ 그렇지 않으면 영원히…

그림 2-3 선형적으로 구분되는 데이터셋과 그렇지 못한 데이터셋

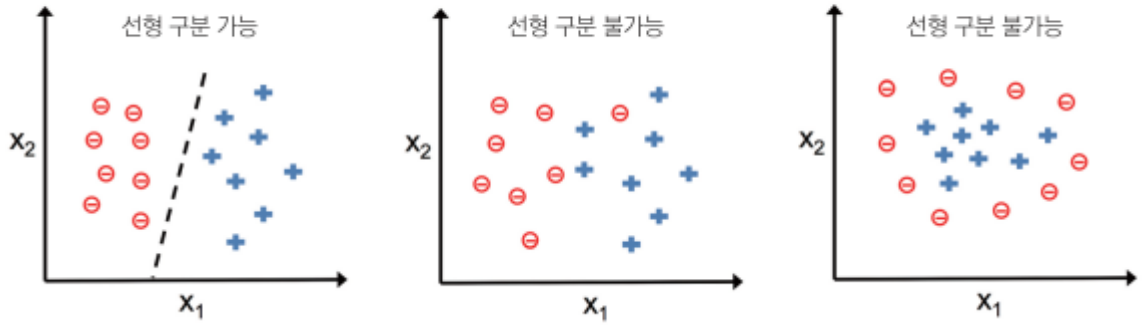
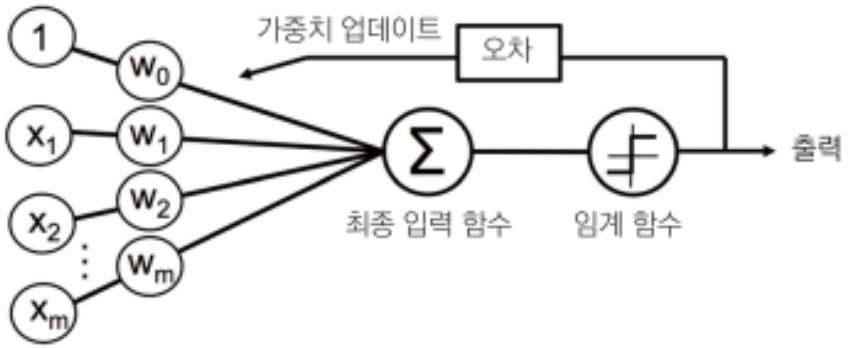


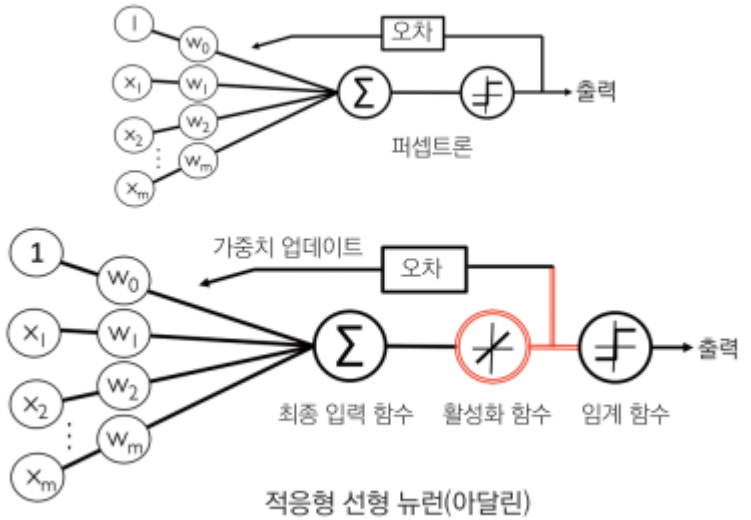
그림 2-4 퍼셉트론 알고리즘



2.3. 적응형 선형 뉴련과 학습의 수렴

- 적응형 선형 뉴런 (ADAptive Linear Neuron, ADALINE, 아달린)
  - ✓ 퍼셉트론의 향상된 버전으로 연속 함수(continuous function)을 비용 함수로 정의하고 최소화하는 핵심 개념을 보여줌
  - ✓ 아달린은 진짜 클래스 레이블과 선형 활성화 함수의 실수 출력 값을 비교
    - 반면 퍼셉트론은 진짜 클래스 레이블과 예측 클래스 레이블을 비교

그림 2-9 퍼셉트론과 아달린 알고리즘의 비교



2.3.1. 경사 하강법으로 비용 함수 최소화

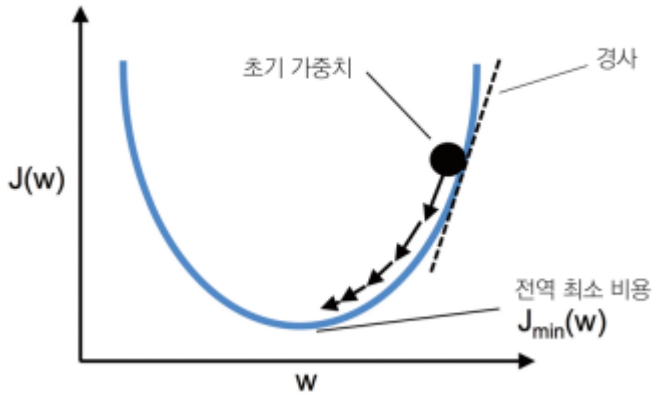
- 목적 함수(object function)
  - ✓ 학습 알고리즘의 과정 동안 최적화를 목표로 하는 함수로, 종종 최소화하려는 비용 함수가 목적 함수가 됨

- 아달린은 계산된 출력과 진짜 클래스 레이블 사이의 제곱 오차합 (Sum of Squared Errors, SSE)로 비용함수 정의

$$J(w) = \frac{1}{2}\sum_i \left(y^i - \phi(z_i)\right)^2$$

  - ✓ Step function 대신 연속 선형 활성화 함수를 쓰는 장점
    - 미분가능!
    - 볼록 함수이므로 전역 최소값 찾기 가능
    - 경사하강법!

그림 2-10 경사 하강법 알고리즘



학습률과 아달린 알고리즘

- 학습률이 너무 크거나 작으면 문제가 생김
  - ✓ 학습률이 너무 클 때: 수렴하지 않음
  - ✓ 학습률이 너무 작을 때: 많은 반복(에포크) 필요

그림 2-11 학습률에 따른 아달린 알고리즘의 수렴

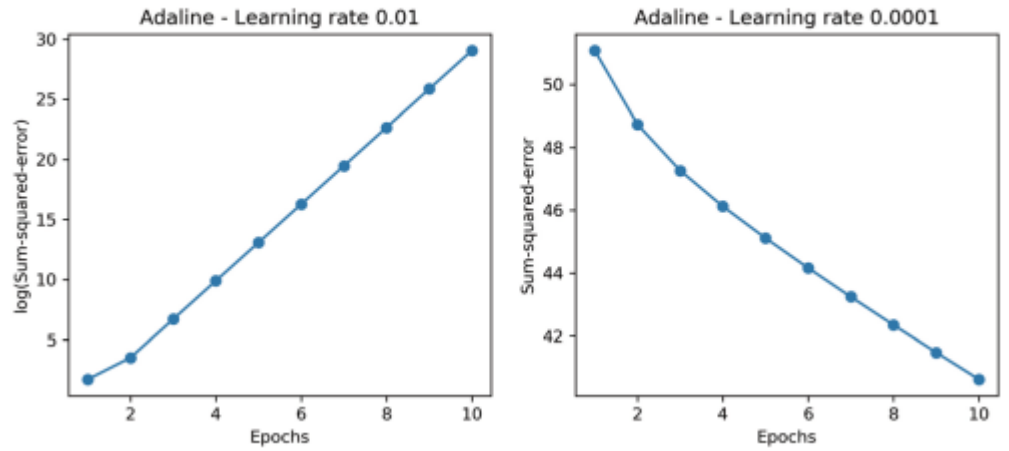
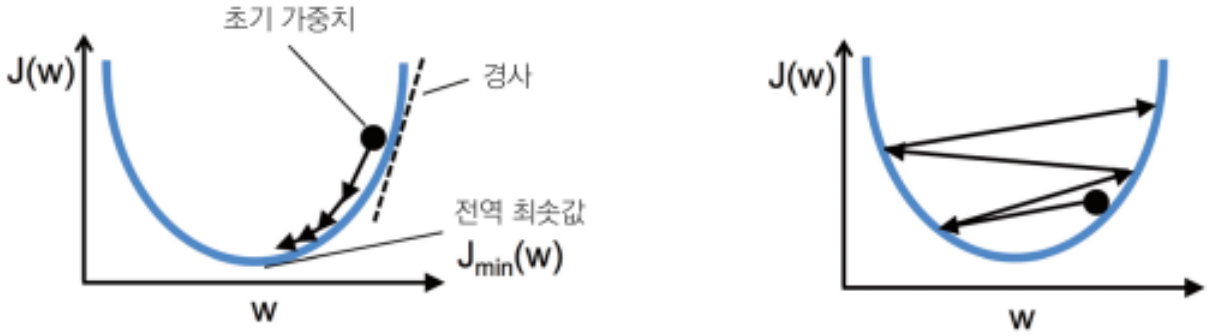


그림 2-12 경사 하강법에서 학습률의 영향

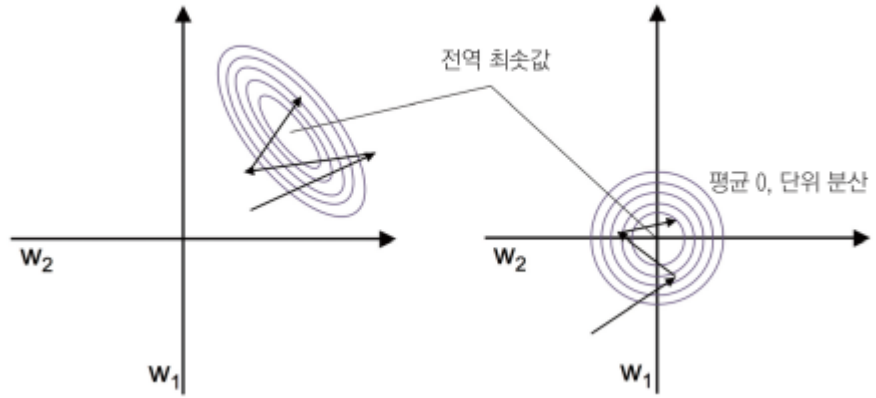


2.3.3. 특성 스케일을 조정하여 경사 하강법 결과 향상

- **표준화(standardization)**
  - ✓ **평균이 0, 표준편차가 1을 갖도록 데이터를 변환**

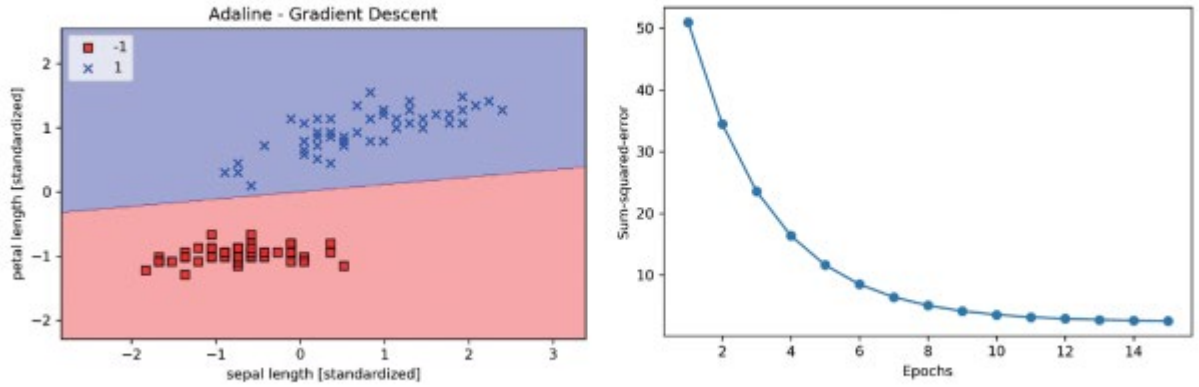
$$x'_j = \frac{x_j - \mu_j}{\sigma_j}$$

그림 2-13 표준화가 비용 함수에 미치는 영향



  - ✓ **표준화 후에는 동일한 학습률에서 더 빠르게 수렴**

그림 2-14 표준화를 적용한 아달린의 결정 경계와 학습 곡선



2.3.4. 대규모 머신 러닝과 확률적 경사 하강법

- **배치 경사 하강법(batch gradient descent)**
  - ✓ 전체 훈련 세트에서 계산된 그래디언트의 반대 방향으로 한 걸음씩 진행하여 비용 함수 최소화
  - ✓ 매번 전체 훈련 데이터를 다시 학습해야하기 때문에 계산비용이 큼

- **확률적 경사 하강법(stochastic gradient descent)**
  - ✓ 배치 경사 하강법의 대안으로, 각 훈련 샘플에 대해 조금씩 가중치를 업데이트
  - ✓ 온라인 학습에 사용 가능 (데이터가 입력 되는대로 훈련에 이용)
- **배치 경사 하강법**: $\Delta w = \eta \sum_i \left(y^i - \phi(z^i)\right)x^i$
- **확률적 경사 하강법**: $\Delta w = \eta\left(y^i - \phi(z^i)\right)x^i$

GD의 친구들

- GD: 모든 데이터를 보고 오류 수정
- SGD: 하나의 데이터를 볼 때마다 오류 수정
- MBGD: 한 챕터(batch)를 보고 오류 수정
  - ✓ 보통 얘를 많이 씀