딥러닝의 통계적이해

2강. 딥러닝과 통계학

1. 통계학과 딥러닝
2. 퍼셉트론과 아달린
3. 선형 회귀모형
4. 최적화방법
5. 로지스틱 회귀모형과 소프트맥스 회귀모형
6. XOR 문제와 다층 신경망

한국방송통신대 이긍희 교수

오늘의 학습목표

1. 딥러닝을 통계학 측면에서 이해한다.
2. 퍼셉트론과 아달린을 이해한다.
3. 선형 회귀모형의 학습을 이해한다.
4. 로지스틱 회귀모형의 학습을 이해한다.
5. 경사하강법을 이해한다.

1. 통계학과 딥러닝

통계학

- 통계학 : 데이터를 통해 배우는 과학
  - 통계적 추론 : 작은 수의 데이터로 모집단 일반화
  - 과학연구, 신약개발, 여론조사, 품질관리 등 성공

- 레이블(답)이 있는 데이터가 많아지고 컴퓨팅 능력이 향상
  - 수학 중심 통계학 → 알고리즘 기반 머신러닝

머신러닝

- 레이블 + 입력 데이터 → 머신러닝 모형 → 적절한 절차
  - 절차를 기반으로 새로운 데이터로 분류 또는 예측

- 인공지능 : 1980년대 후반 이후 통계학의 방법론을 활용
  - 컴퓨팅 능력과 데이터가 부족
    → 딥러닝·머신러닝 : 확률분포, 통계학을 활용
  - 이후 데이터 기반 머신러닝, 딥러닝으로 진화

용어 비교

통계학	딥러닝·머신러닝
모수(parameter)	가중치(weights)
추정(estimation) 적합(fitting)	학습(learning)
회귀 또는 분류	지도학습
군집화, 분포 추정	비지도학습
독립(설명)변수	특징
종속(반응)변수	레이블

통계모형과 딥러닝 모형

- 통계모형 : 확률분포 기반으로 추정과 가설검정을 통해 결과의 원인을 설명하는 것이 주목적
- 딥러닝 모형 : 블랙박스(black box) → 결과의 예측 목적

통계모형과 딥러닝 모형

	전통적 통계모형	딥러닝 모형
데이터 크기	소규모	대규모
모형의 구조	데이터 = 생성구조 + 오차	오차가 주요 문제가 아님
모형의 크기	데이터 생성구조를 저차원 모형으로 파악	데이터 생성구조가 복잡 → 고차원 모형(블랙박스)
모형의 평가	적합도, 유의성	예측력
모형의 작성	생성구조를 오차에서 분리	학습을 통해 데이터의 복잡한 특성을 이해

확률과 정보량

- 확률(probability) : 사건이 일어날 가능성을 0과 1사이 숫자로 표현

- 사건 B 의 정보량 : $I(B) = -\log[P(B)]$

확률변수

- 확률변수 : 표본공간을 숫자로 바꿔 주는 함수
  - 이산형 확률변수와 연속형 확률변수로 구분

- 확률변수의 기댓값과 산포

$$\mathrm{E}(X) = \begin{cases} \sum_i x_i f(x_i), & X\text{는 이산형} \\ \int_{-\infty}^{\infty} x f(x) dx, & X\text{는 연속형} \end{cases}$$

$$Var(X) = E[(X-\mu)^2]$$

엔트로피

◆ 정보량의 기댓값 : 엔트로피(entropy)

$$H(X) = E[I(X)] = \begin{cases} -\sum_i \log[f(x_i)] f(x_i), & X\text{가 이산형} \\ -\int_{-\infty}^{\infty} \log[f(x)] f(x) dx, & X\text{가 연속형} \end{cases}$$

베르누이 분포

- 시행결과가 둘 중 하나가 되는 확률변수의 분포 :
  $X \sim Ber(p)$
  - 2개의 범주를 분류하는 머신러닝 과제에 활용

베르누이 분포

- $f(x) = P(X = x) = p^x(1-p)^{1-x}, \quad x = 0,1$
  - $E(X) = p, \quad Var(X) = p(1-p)$

멀티누이 분포

- 확률변수가 K개 범주 가질 때의 분포
- $P(X_1 = x_1, X_2 = x_2, \ldots, X_K = x_k) = \prod_{i=1}^{K} p_i^{x_i}, \quad x_i = 0,1$
  - $p_k = P(X_K = k), \ \sum_{i=1}^{K} p_i = 1, \ \sum_{i=1}^{K} x_i = 1$

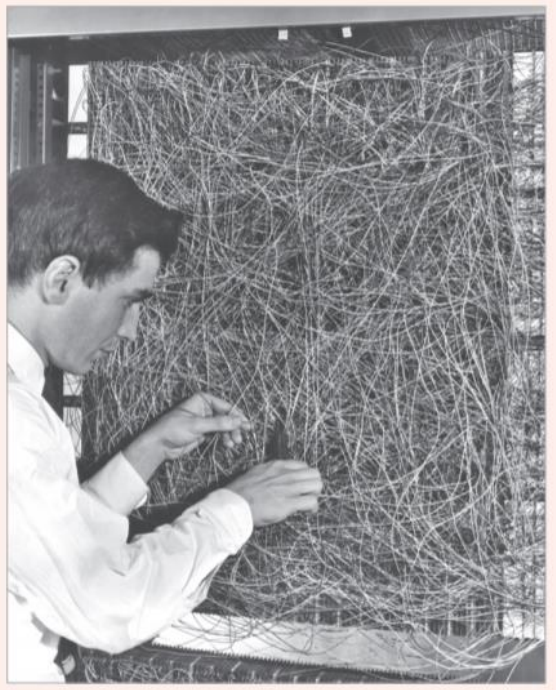
정규 분포

- $X \sim N(\mu, \sigma^2)$: 연속형 변수 예측 과제에 활용
- $f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], \quad -\infty < x < \infty$
- 엔트로피 : $H(X) = \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2}$

2. 퍼셉트론과 아달린

퍼셉트론

- 1958년 로젠블랫(F. Rosenblatt) : 이미지 인식 기계인 퍼셉트론

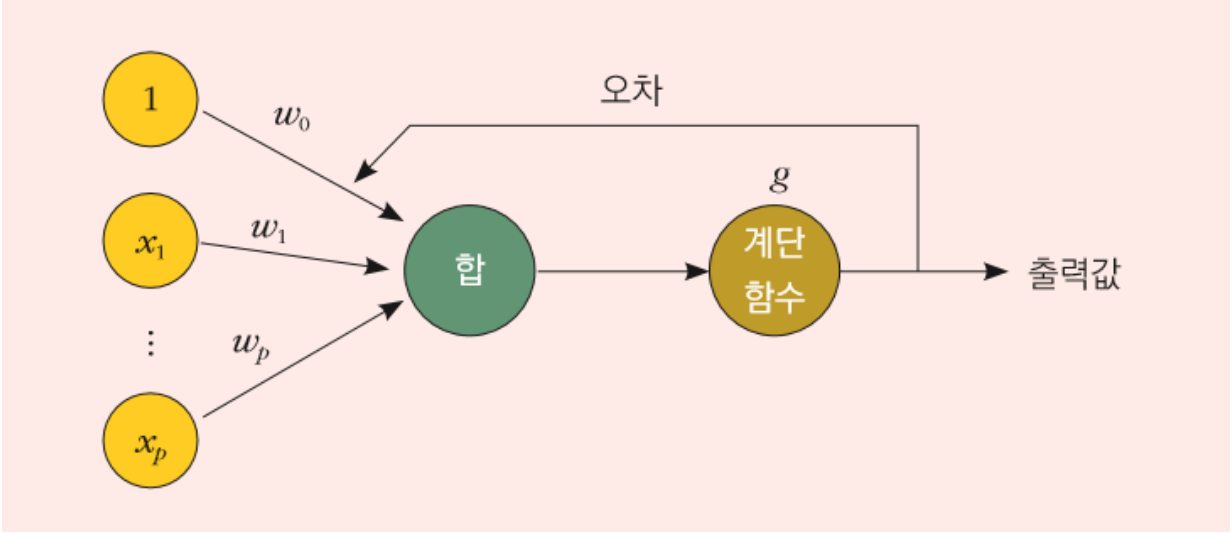


출처 : getty images

퍼셉트론

- 퍼셉트론 : 선형 이진(binary) 분류기

$$\hat{y} = g\left(\sum_{j=0}^{p} w_j x_j\right) = g(\boldsymbol{x}^T \boldsymbol{w}), \quad g(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$$



퍼셉트론의 학습

- 퍼셉트론을 손실함수를 정하지 않고 학습

→ 가중치 $w_0, w_1, \ldots, w_p$의 값 구함

- 오차 $e^{(i)} = \left(y^{(i)} - \hat{y}^{(i)}\right)$로 가중치 갱신($\eta$:학습률)

$$w_j := w_j + \Delta w_j$$

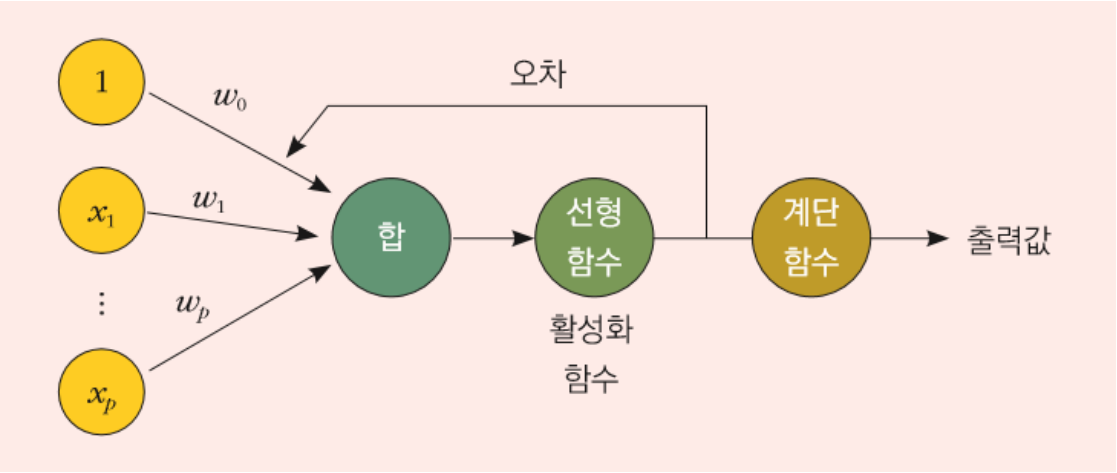
$$\Delta w_j = \eta e^{(i)} x_j^{(i)}, \quad j = 0,1,\cdots,p$$

아달린

- 1960년 위드로(B. Widrow)와 호프(M. Hoff)
  - 아달린(Adaline, Adaptive Linear Neuron) 기계 개발

아달린

- 아달린 : 퍼셉트론과 유사, 차이는 선형함수 적용 후 계단함수 이용
  - $\hat{y} = g\left(\sum_{j=0}^{p} w_j x_j\right) = g(\boldsymbol{x}^T \boldsymbol{w})$



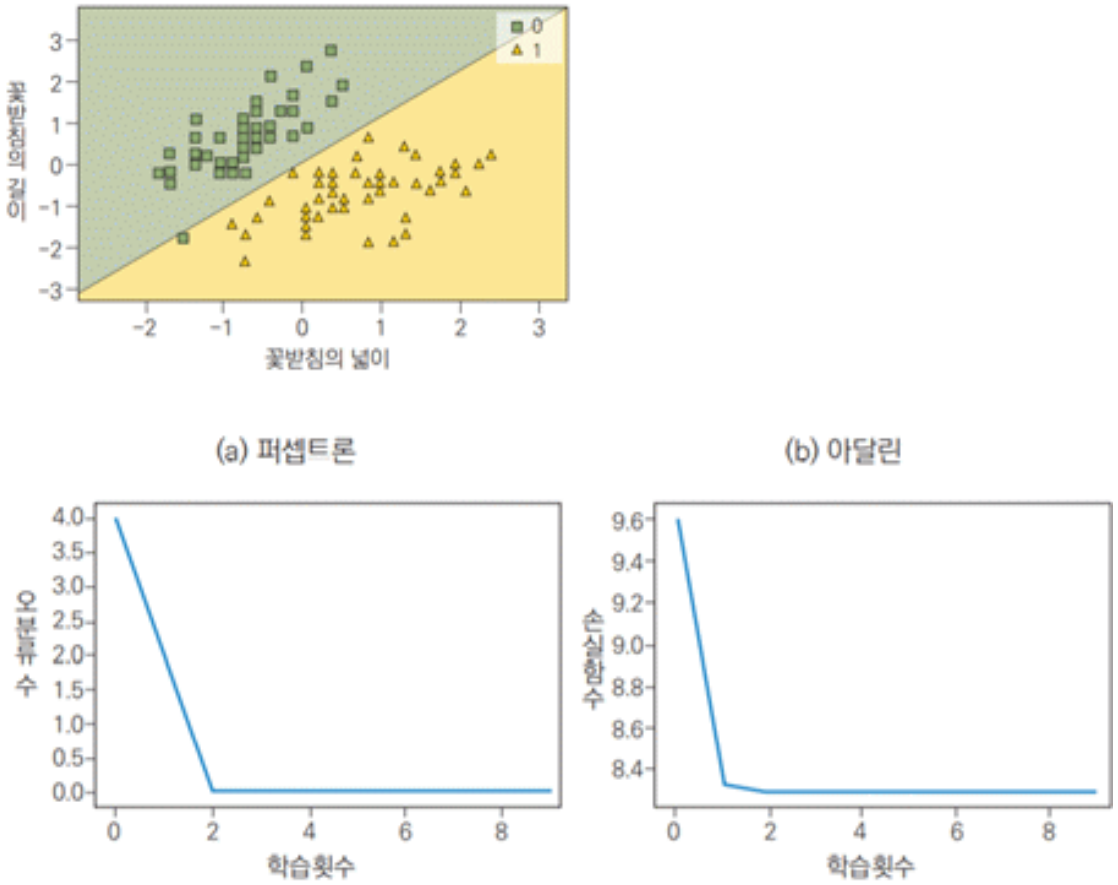
아달린

- 위드로-호프의 학습규칙
  - 손실함수 : $J(w) = \frac{1}{2}\sum_i \left(y^{(i)} - \hat{y}^{(i)}\right)^2 = \frac{1}{2}\sum_i \left[e^{(i)}\right]^2$
  - $w_j := w_j + \Delta w_j$

    $\Delta w_j = \eta e^{(i)} x_j^{(i)}, \quad j = 0,1,\cdots,p$

- 아달린은 오차가 커질수록, 입력값이 커질수록 가중치가 크게 수정 → 최소제곱법과 동일

IRIS 종 분류



3. 선형 회귀모형의 학습

선형 회귀모형

- 설명변수들로 종속변수를 선형적으로 예측하는 모형

  - $y_i = w_0 + w_1 x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)$
  - w_0, w_1 : 최소제곱법과 최대가능도추정법으로 추정

최소제곱법

- 손실함수 최소화하는 w_0, w_1를 구함
  - 손실 함수 : $J(w) = \frac{1}{2n}\sum_{i=1}^{n}(w_0 + w_1 x_i - y_i)^2$
  - w_0와 w_1에 대해 미분

$$\rightarrow \hat{w}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \quad \hat{w}_0 = \bar{y} - \hat{w}_1 \bar{x}$$

최대가능도법

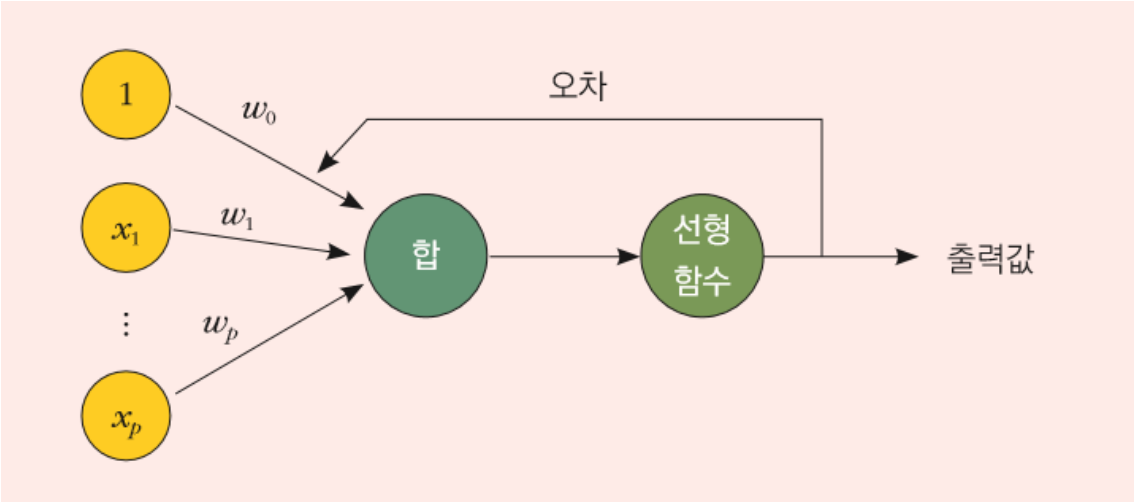
- 가능도 함수를 최대화하는 w_0, w_1를 구함

  - 가능도 함수 : $L(w) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(y_i - w_0 - w_1 x_i)^2}{2\sigma^2}\right]$

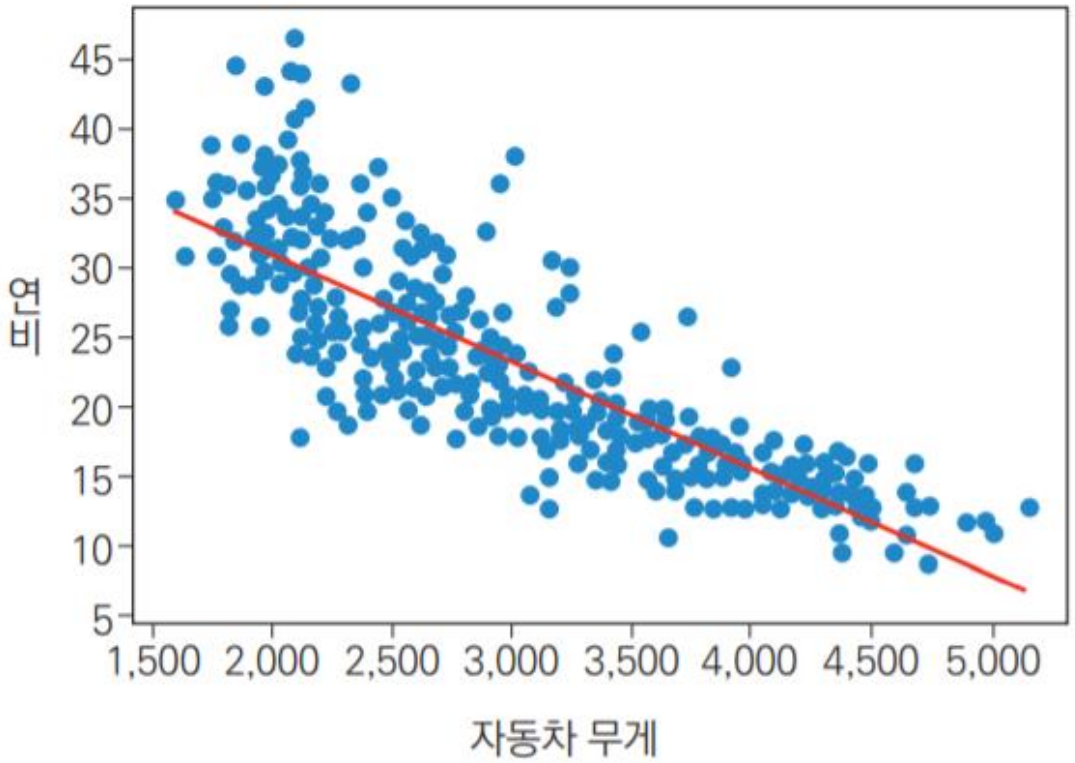
  - 최대가능도추정법 추정결과 : 최소제곱법과 동일

수치해석적 방법

- 선형 회귀모형은 g가 선형인 신경망으로 표현
  - $\hat{y} = g\left(\sum_{j=0}^{p} w_j x_j\right) = g(\boldsymbol{x}^T \boldsymbol{w})$



선형회귀분석의 예


45
40
35
30
25
20
15
10
5
연비
1,500
2,000
2,500
3,000
3,500
4,000
4,500
5,000
자동차 무게

4. 최적화 방법

머신러닝과 최적화

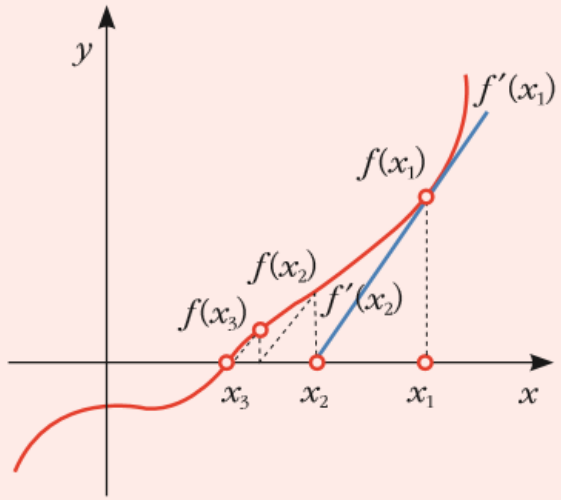
- 머신러닝 : 손실함수를 최소화하여 모형의 모수를 정함
  → 최적화(optimization) 문제

- 손실함수 미분해서 최적해를 구하기 어려운 경우
  - 수치해석 최적화 방법 : 뉴턴(Newton – Raphson) 방법과 경사하강법(Gradient Descent Algorithm)

뉴턴의 방법

◆ $f(x) = 0$ 만족 x를 찾는 방법 : $f(x) = g'(x)$

$$x := x - \frac{f(x)}{f'(x)} \leftrightarrow x^{(i+1)} = x^{(i)} - \frac{f\left(x^{(i)}\right)}{f'\left(x^{(i)}\right)}$$



뉴턴의 방법

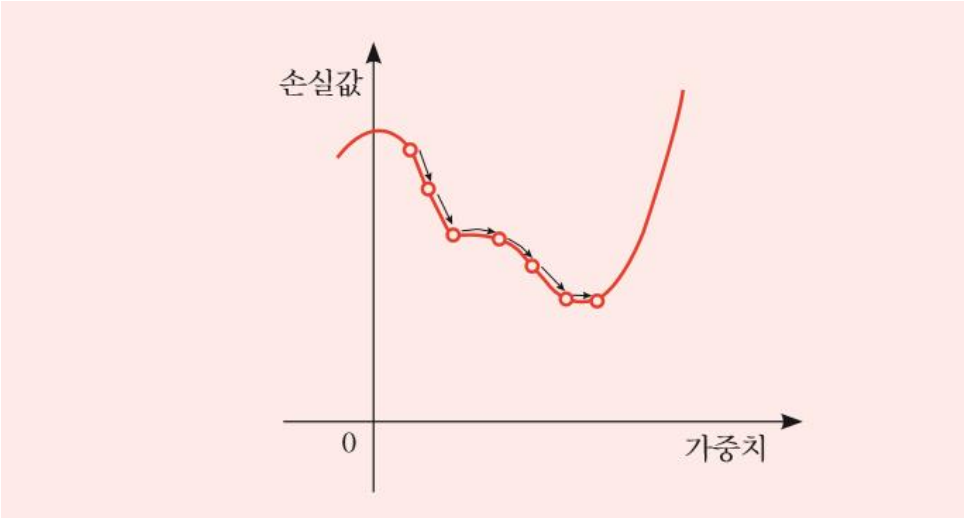
- 손실함수 $J(w)$: 2차 미분가능, 일변량 함수

$$w := w - \frac{J'(w)}{J''(w)}$$

- 뉴턴의 방법 : 계산량 적고, 학습률(η)을 구할 필요 없음
  - $J(w)$ 2차 미분 가능 함수, 변수 수가 작아야 함

경사하강법

- 함수 1차 미분 가능, 볼록 함수가 아닌 복잡한 모양
  - 함수의 현재 위치에서 조금씩 이동 → 최솟값에 접근
  - 손실함수를 줄이는 경사를 따라 조금씩 가중치 갱신



경사하강법

- 손실함수 $J(w)$ 최소점으로 가는 w를 찾는 법

$$w := w - \eta \frac{\partial}{\partial w} J(w)$$

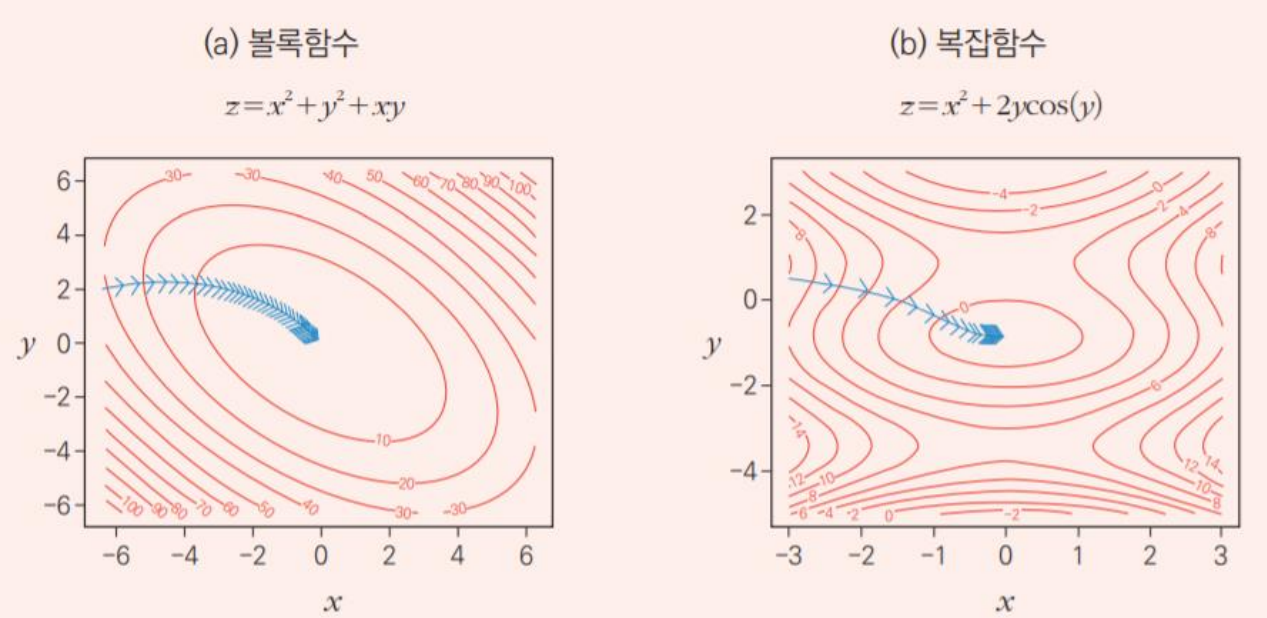
- 선형 회귀모형

$$w_0 := w_0 - \eta \frac{1}{n} \sum_{i=1}^{n} (w_0 + w_1 x_i - y_i)$$

$$w_1 := w_1 - \eta \frac{1}{n} \sum_{i=1}^{n} (w_0 + w_1 x_i - y_i) x_i$$

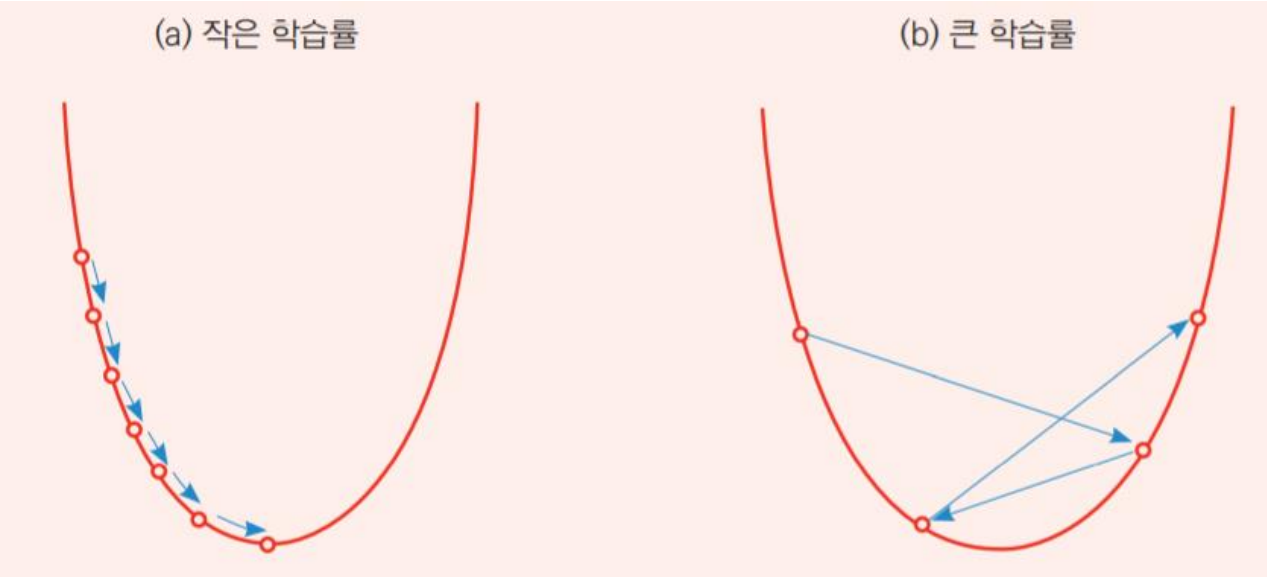
경사하강법의 경로

- 손실함수가 복잡한 형태 → 국지 최솟값



경사하강법

- 경사하강법에서 학습률 η에 따라 수렴속도가 달라짐



확률적 경사하강법(SGD)

- 데이터 임의로 한 개 선택 → $\frac{\partial}{\partial w} J(w)$ 구하고, 가중치 갱신
- 미니배치 경사하강법 : 일부 데이터를 이용하여 경사하강법 적용

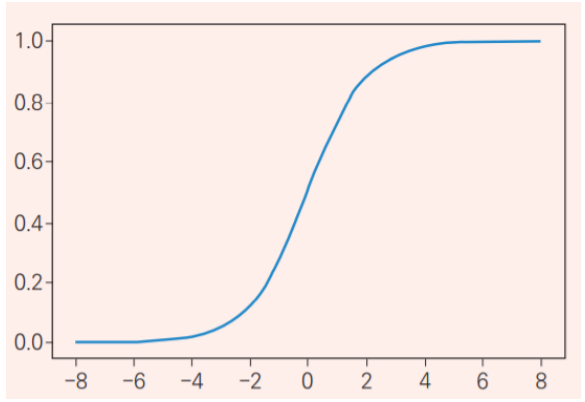
에포크(epoch)

- 1 에포크(epoch) : 훈련 데이터 전체를 학습하는 것
  - 배치(batch)는 전체 훈련데이터를 몇 개로 나눈 것
  - 에포크(epoch)의 수는 전체 데이터를 학습한 횟수

5. 로지스틱 회귀모형과 소프트맥스 회귀모형

로지스틱 회귀모형

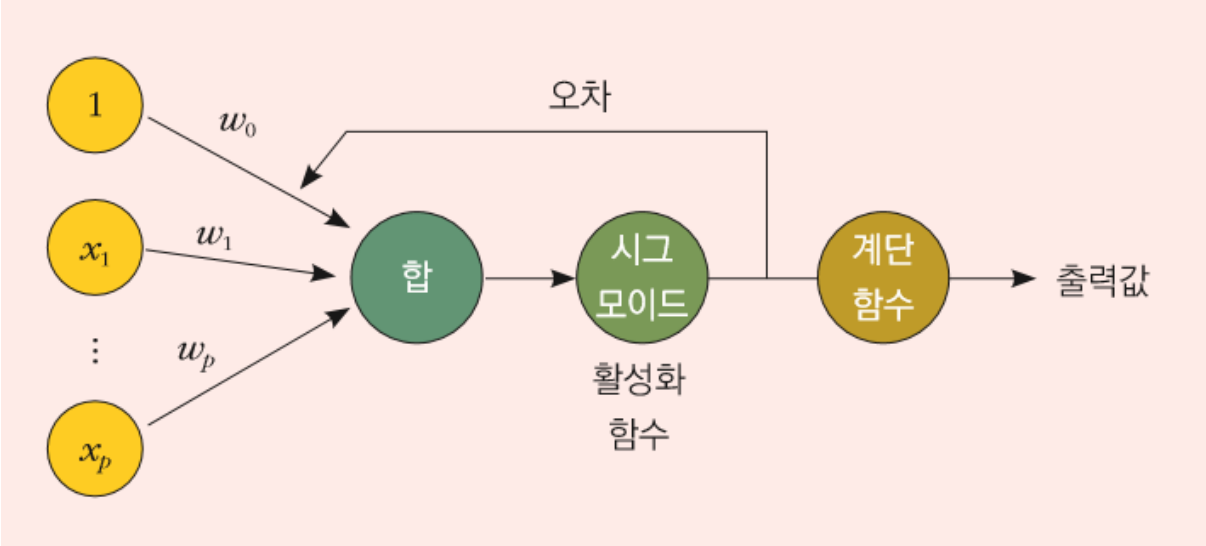
- 출력값은 2개 범주
  - 활성화 함수 : $g(x) = \frac{1}{1+e^{-x}}$



  - $\pi(\boldsymbol{x}) = P(y = 1|\boldsymbol{x})$
    - $\pi(\boldsymbol{x}) > 0.5 \rightarrow y = 1$로 판단

로지스틱 회귀모형

- $\pi(\boldsymbol{x}) = g\left(\sum_{i=0}^{p} w_i x_i\right) = \dfrac{1}{1+\exp\left(-\sum_{i=0}^{p} w_i x_i\right)}$



로지스틱 회귀모형

- $\pi_i = \pi_i(\boldsymbol{x})$인 베르누이 분포

  - 가능도 함수 : $L(w) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i}$

  - 로그가능도 함수 :

$$\log[L(w)] = \sum_{i=1}^{n}[y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i)]$$

  → 경사하강법으로 w를 구함

로지스틱 회귀모형

- 로그가능도함수 최대화 = 손실함수 최소화

$$J(w) = -\log[L(w)] = \sum_{i=1}^{n} J_i(w)$$

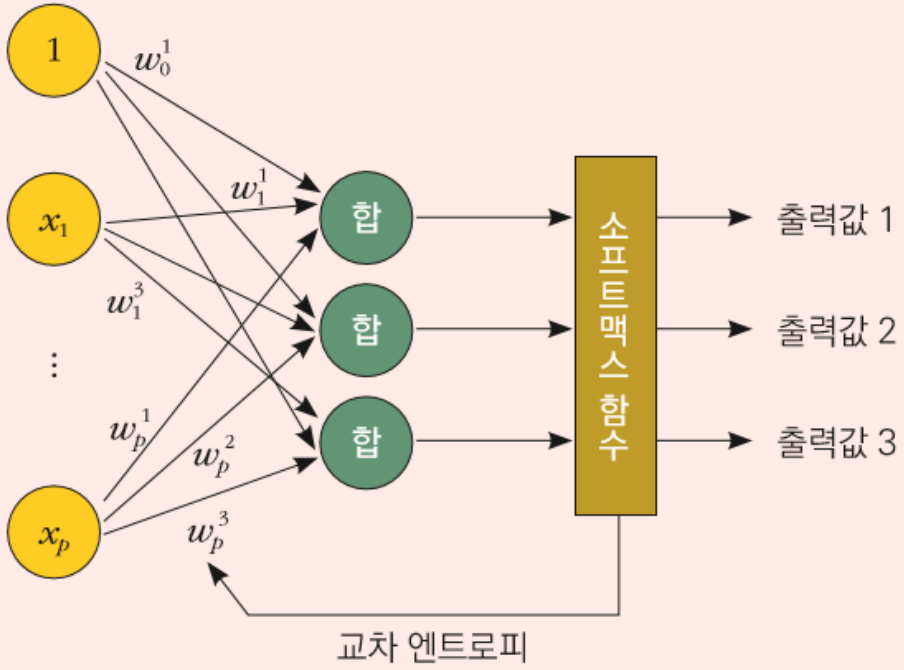
$$J_i(w) = -[y_i \log(\pi_i) + (1 - y_i) \log(1 - \pi_i)]$$

- 경사하강법 : $w_j := w_j - \eta \frac{\partial}{\partial w_j} J(w)$

소프트맥스 회귀모형

- 여러 개중 하나를 분류할 때의 모형
  - 소프트맥스 함수 : $g(x_i) = \frac{\exp(x_i)}{\sum_i \exp(x_i)}$
  - $\pi_i^k = P(y_i = k|\boldsymbol{x}, \boldsymbol{w}^k) = \frac{\exp(\boldsymbol{x}\boldsymbol{w}^k)}{\sum_{j=1}^K \exp(\boldsymbol{x}\boldsymbol{w}^j)}$

소프트맥스 회귀모형


1
x_1
x_p
w_0^1
w_1^1
w_1^3
w_p^1
w_p^2
w_p^3
합
합
합
소프트맥스 함수
출력값 1
출력값 2
출력값 3
교차 엔트로피

소프트맥스 회귀모형

- 멀티누이분포 → 손실함수 : 교차 엔트로피

$$J(w) = -\sum_i \sum_k y_i^k \log\left(\pi_i^k\right)$$

- 경사하강법 : 가중치 갱신

$$w_j^k := w_j^k - \eta \frac{\partial J(w)}{\partial w_j^k}$$

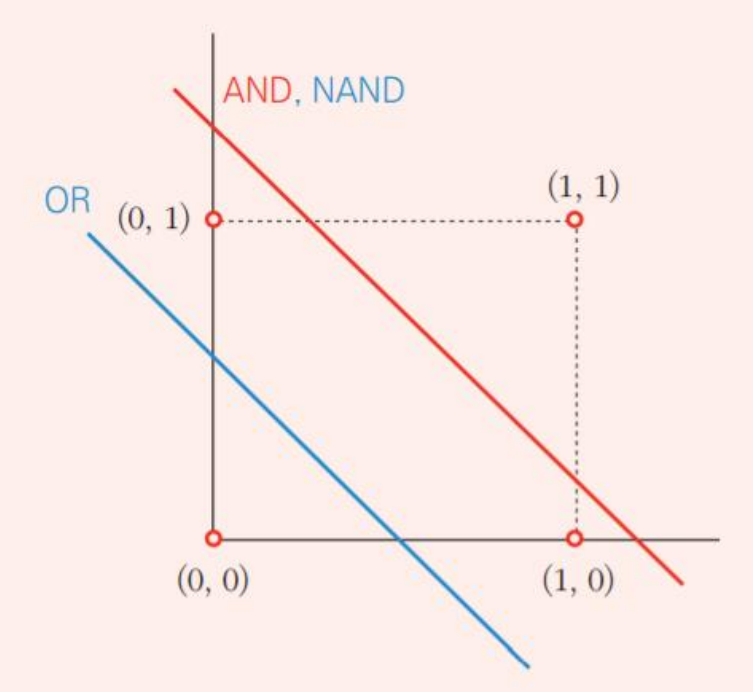
활성화 함수와 손실함수

	출력변수	활성화 함수	손실함수
이산형	베르누이 분포	시그모이드	이진 엔트로피
이산형	멀티누이 분포	소프트맥스	교차 엔트로피
연속형	정규분포	선형	평균제곱오차

6. XOR 문제와 다층 신경망

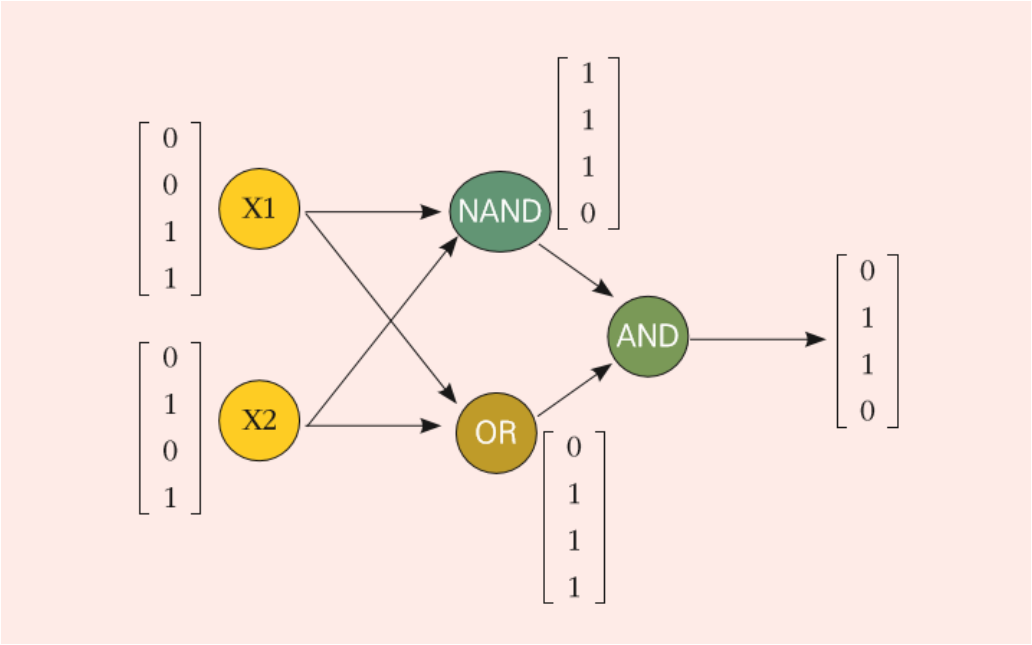
XOR 문제

- 민스키와 페퍼트 : 1개 층 신경망은 선형적 분류만 가능, XOR 문제 해결 불가능 → 인공지능의 겨울



다층신경망

- NAND 연산과 OR 연산 + AND 연산 → XOR 연산이 가능
  - 중간층을 추가한 다층 신경망으로 비선형 문제를 해결 가능



학습정리

- ✓ 퍼셉트론과 아달린은 초기 신경망 모형인데, 오차수정법으로 학습한다.
- ✓ 경사하강법은 손실함수를 최소화하는 대표적 알고리즘이다.

학습정리

- ✓ 선형 회귀모형, 로지스틱 회귀모형과 소프트맥스 회귀모형을 경사하강법으로 추정할 수 있다.
- ✓ 퍼셉트론으로는 XOR 문제와 같은 비선형 분류 문제를 해결할 수 없다.

딥러닝의 통계적이해

다음시간안내

3강.
딥러닝 모형의 구조와 학습(1)